Statistical Approaches for Entity Resolution under Uncertainty

Neil G. Marchant

ORCID: 0000-0001-5713-4235

Doctor of Philosophy

May 2021

School of Computing and Information Systems
Melbourne School of Engineering
The University of Melbourne

Submitted in total fulfilment of the requirements
for the degree of Doctor of Philosophy

Abstract

When real-world entities are referenced in data, their identities are often obscured. This presents a problem for data cleaning and integration, as references to an entity may be scattered across multiple records or sources, without a means to identify and consolidate them. Entity resolution (ER; also known as record linkage and deduplication) seeks to address this problem by linking references to the same entity, based on imprecise information. It has diverse applications: from resolving references to individuals in administrative data for public health research, to resolving product listings on the web for a shopping aggregation service. While many methods have been developed to automate the ER process, it can be difficult to guarantee accurate results for a number of reasons, such as poor data quality, heterogeneity across data sources, and lack of ground truth. It is therefore essential to recognise and account for sources of uncertainty throughout the ER process. In this thesis, I explore statistical approaches for managing uncertainty—both in quantifying the uncertainty of ER predictions, and in evaluating the accuracy of ER to high precision. In doing so, I focus on methods that require minimal input from humans as a source of ground truth. This is important, as many ER methods require vast quantities of human-labelled data to achieve sufficient accuracy.

In the first part of this thesis, I focus on Bayesian models for ER, owing to their ability to capture uncertainty, and their robustness in settings where labelled training data is limited. I identify scalability as a major obstacle to the use of Bayesian ER models in practice, and propose a suite of methods aimed at improving the scalability of an ER model proposed by Steorts (2015). These methods include an auxiliary variable scheme for probabilistic blocking, a distributed partially-collapsed Gibbs sampler, and fast algorithms for performing Gibbs updates. I also propose modelling refinements, aimed at improving ER accuracy and reducing sensitivity to hyperparameters. These refinements include the use of Ewens-Pitman random partitions as a prior on the linkage structure, corrections to logic in the record distortion model and an additional level of priors to improve flexibility.

I then turn to the problem of ER evaluation, which is particularly challenging due to the fact that coreferent pairs of records (which refer to the same entity) are extremely rare. As a result, estimates of ER performance typically exhibit high levels of statistical uncertainty, as they are most sensitive to the rare coreferent (and predicted coreferent) pairs of records. In order to address this challenge, I propose a framework for online supervised evaluation based on adaptive importance sampling. Given a target performance measure and set of ER systems to evaluate, the framework adaptively selects pairs of records to label in order to approximately minimise statistical uncertainty. Under verifiable conditions on the performance measure and adaptive policy, I establish strong consistency and a central limit theorem for the resulting performance estimates. I conduct empirical studies, which demonstrate that the framework can yield dramatic reductions in labelling requirements when estimating ER performance to a fixed precision.

Declaration

This is to certify that

(i) the thesis comprises only my original work towards the Doctor of Philosophy degree except where indicated in the preface;

(ii) due acknowledgement has been made in the text to all other material used; and

(iii) the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Neil G. Marchant

Preface

This thesis comprises original work which was conducted solely during my PhD candidature and has not been submitted for any other qualifications. Several chapters incorporate material from publications of which I was the primary author.

Chapter 3 is based on the following publication:

> N. G. Marchant, A. Kaplan, D. N. Elazar, B. I. P. Rubinstein and R. C. Steorts. "d-blink: Distributed End-to-End Bayesian Entity Resolution". In: *Journal of Computational and Graphical Statistics* (2021). DOI: 10.1080/10618600.2020.1825451.

I developed and implemented the ideas, conducted most of the experiments, and was the lead author. A. Kaplan contributed to software testing. R. C. Steorts conducted experiments for the case study on U.S. Census and Social Security Administration data presented in Section 3.8. Permission was granted by the U.S. Census Bureau Disclosure Review Board to release limited experimental results in this section (DRB#: CBDRB-FY20-309). R. C. Steorts, B. I. P. Rubinstein and D. N. Elazar jointly supervised the work. All authors assisted with editing.

Chapters 5 and 6 incorporate material from the following publication:

> N. G. Marchant and B. I. P. Rubinstein. "Needle in a Haystack: Label-Efficient Evaluation under Extreme Class Imbalance". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '21. Virtual Event, Singapore: ACM, 2021. DOI: 10.1145/3447548.3467435. Accepted.

I developed most of the ideas, conducted all of the experiments and was responsible for the writing. B. I. P. Rubinstein guided the work and assisted with editing.

Chapter 6 incorporates material from the following publication:

> N. G. Marchant and B. I. P. Rubinstein. "In Search of an Entity Resolution OASIS: Optimal Asymptotic Sequential Importance Sampling". In: *Proc. VLDB Endow.* 10.11 (2017), pp. 1322-1333. DOI: 10.14778/3137628.3137642.

I developed most of the ideas, conducted all of the experiments and was responsible for the writing. B. I. P. Rubinstein guided the work and assisted with editing.

During my PhD candidature, I was financially supported by an Australian Government Research Training Program Scholarship, Australian Bureau of Statistics project ABS2018.363, and the AMSIIntern program hosted by the Australian Bureau of Statistics.

Acknowledgements

I would like to thank all of those who supported me in the completion of this thesis. I am especially indebted to my principal supervisor Ben Rubinstein, who has been an unwavering source of support and encouragement during my candidature. Ben's enthusiasm for research, openness to new ideas and warm personality have made working with him an absolute joy. I am also immensely grateful to have been co-supervised by Rebecca (Beka) Steorts, who welcomed me into her research group midway through my candidature. Beka has been extremely generous with her time, despite living on the other side of the world, and I have benefitted greatly from her mentorship. My thanks also go to the other members of my PhD advisory committee: Aurore Delaigle and Reeva Lederman.

During my candidature I was fortunate to complete an internship at the Australian Bureau of Statistics (ABS). I would like to thank Daniel Elazar, Kelly Chiu and colleagues for welcoming me into the Methodology Division at the ABS, and providing me with a valuable experience. The practical knowledge I gained at the ABS has shaped my understanding of some aspects of the research presented in this thesis. I was also fortunate to visit Duke University, and would like to thank Beka Steorts, Andee Kaplan and postdocs in the Department of Statistical Science for being kind and generous hosts.

Finally, I would like to thank my family for their enduring love and support. This thesis would not have been possible without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data has become a central feature of the socio-economic landscape. As a society, we are collecting, storing, sharing and processing data at an unprecedented rate and scale [CML14]. This growth has been fuelled in part by technological advances in data storage, networking, cloud computing and smart devices [Kit14]. With the increasing availability of data comes opportunities to accelerate and automate knowledge discovery and decision making. Indeed data-driven innovation has been recognised as a key pillar of 21st century growth, with the "potential to significantly enhance productivity, resource efficiency, economic competitiveness and social well-being" [OEC15]. However, one of the major barriers to data-driven innovation is the difficulty in extracting useful and reliable information from raw data, which may be of variable quality and may be spread across multiple sources [SS14]. As a result, there has been increasing interest in automated solutions for data cleaning [Chu+16] and data integration [DS15], which aim to coerce data into a unified, consistent format.

This thesis focuses on a fundamental task that arises in the context of data cleaning and data integration called *entity resolution*. Many data sources contain information about real-world entities, such as people, businesses, products, etc. However, when entities are represented in data, their identities are often recorded imprecisely and/or inconsistently. This makes it difficult to consolidate records that refer to the same entity, as there is no unique identifier (e.g. a social security number) which can serve as a key for a database join. Entity resolution (ER) seeks to address this problem by *inferring* which records refer to the same entity based on observed data. It facilitates the removal of duplicate records, and the integration of records from different data sources—both key steps to achieving clean integrated data. ER is also known under a variety of other names, including *record linkage*, *data matching*, *deduplication* and *merge/purge* [Chr12c; GM12].

ER has been applied to solve practical problems in a variety of domains, including linking administrative records for public health research [JRB11], producing accurate statistics on human rights violations [Sad18], deduplicating citation databases [BG07] and identifying suspicious individuals for counterterrorism purposes [GL04]. A motivating application of ER is presented in Figure 1.1. It illustrates how ER can be used to build a comparison shopping service that integrates data from multiple retailers. In this application, ER is used to identify listings from different retailers that relate to the same product. This allows the service to maintain a unified product catalogue, where products in the catalogue are linked to retailer listings. Without ER, the catalogue would contain

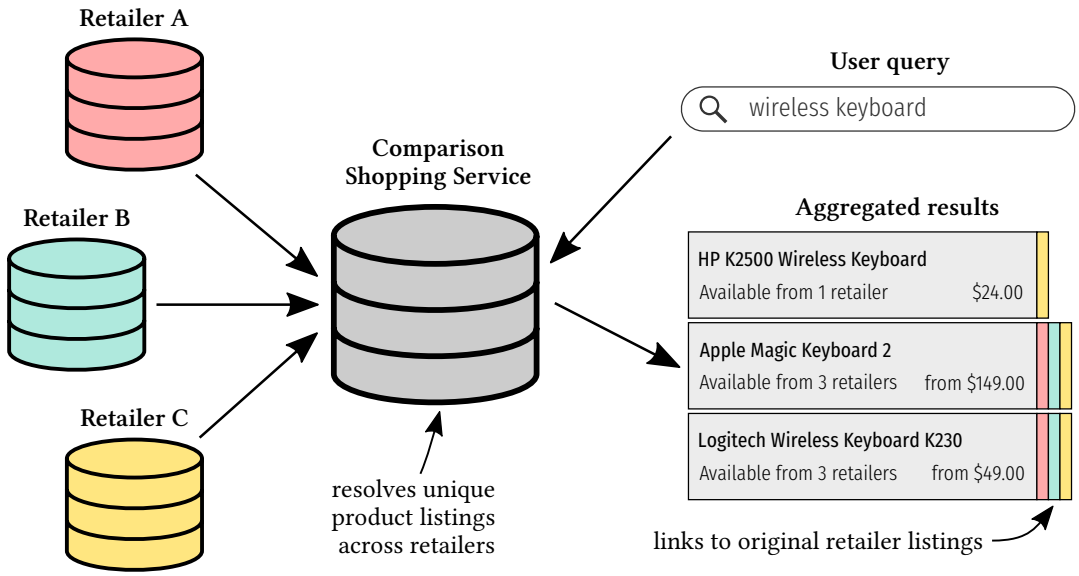


Figure 1.1: Schematic of a comparison shopping service that uses entity resolution to maintain a unified product catalogue based on listings from multiple retailers. Listings that relate to the same product are grouped together in the catalogue, providing a user-friendly experience.

duplicate product entries, leading to a poor user experience.

1.1 Challenges for effective entity resolution

Early applications of ER date back to the 1950s, when simple matching rules were applied to resolve individuals in vital public records [New+59]. Significant progress has been made since then, including the development of statistical models, the application of machine learning methods, as well as practical techniques for improving scalability [GM12]. In this section, we discuss several challenges for effective ER, which serve as motivation for the research directions explored in this thesis.

1.1.1 Open world assumption

ER is a difficult task because it must often be performed under the assumption of an *open world* [SS14]. This means that the input data is effectively unconstrained in the worst case. In an open world, ER methods must be robust to heterogeneity in data representations, as well as broader data quality issues, such as data entry errors, corruption and missing values. Figure 1.2 illustrates some of these issues using real data from a comparison shopping service application. It contains two listings for the same product from different retailers. However since the data comes from an open world, there are many inconsistencies which makes it difficult to determine whether the listings refer to the same product or not. Firstly, there are differences in the schemas which makes it impossible to directly compare most of the attributes. For example, the listing from retailer A has a “Dimensions” attribute, while the listing from retailer B has “Product

Product Listing from Retailer A

Name:	Microsoft Surface Wireless Keyboard
Manufacturer:	Microsoft
Type:	Keyboard
Colour:	Silver
Weight:	419.3 g
Dimensions:	420.9 × 112.60 × 19.30 mm
Warranty:	1 year
Price:	$158

Product Listing from Retailer B

Product Name:	Microsoft Surface Bluetooth Keyboard
Brand:	Microsoft Surface
Product Type:	Keyboards
Colour:	Grey
Weight (kg):	419.3
Product width (cm):	11.26
Product Depth (cm):	1.93
Manufacturer's warranty (months):	12
Price:	$158

Figure 1.2: A realistic example of semantic heterogeneity and data quality issues encountered when performing entity resolution of product listings. Differences between the schemas and attribute values in the two listings make it difficult to determine whether the listings refer to the same product algorithmically. Even humans may have difficulty without access to the product image.

Width (cm)" and "Product Depth (cm)" attributes. Secondly, even if the schemas were aligned, there are still differences in the semantic representation of attribute values. For instance, retailer A lists the colour as "Silver" while retailer B lists the colour as "Grey". There is also a data entry error: retailer B lists the weight as 419.3 kg. When combined, these issues make ER difficult to automate—particularly when using traditional methods based on hard-coded rules. Recent ER methods based on deep learning have shown promise when the input data is dirty and/or unstructured, however they rely heavily on human-labelled data for training [Mud+18; Ebr+18; Kas+19].

1.1.2 Reliance on human input

Given the immense challenge of performing ER in an open world, it is often necessary for humans to be involved in the ER process in order to ensure performance standards are met. Common tasks performed by humans include: model selection, manual parameter tuning, and preparation of labelled data for training and evaluation. Of these tasks, the most demanding is arguably the preparation of labelled data, as large quantities are typically required due to severe imbalance between coreferent and non-coreferent pairs of records [Bel+12]. As a result, it is important to study label-efficient methods for ER in order to ease the burden on humans and improve cost-effectiveness. There may also be security or privacy considerations which make it difficult for humans to provide labels, especially when sensitive personal data is involved.

Previous work on label efficiency has largely focused on reducing the amount of labelled data required for training, by leveraging unsupervised models [RC04; BG06; Ste15; Sad17], active learning [SB02; AGK10; Bel+12; Kas+19], and transfer learning

[NRG12; Kas+19]. We discuss these approaches in further detail in Section 2.2. To our knowledge, there has been no work on label-efficient evaluation in the context of ER.

1.1.3 Quantification of uncertainty

Another shortcoming of commonly used ER methods, is their limited ability to quantify and propagate uncertainty. The majority of methods simply return the "most likely" solution, even if there are feasible alternatives. For example, when applying ER to product listings, as illustrated in Figure 1.2, most methods would return a point prediction for the match status of the two listings. However, in this example it is difficult to say for certain whether the listings refer to the same product based on the textual attributes—perhaps there is a 60% chance that the listings match and a 40% chance that they don't. It is important to quantify and propagate this uncertainty, as it may improve the accuracy of other parts of the data cleaning or data integration process. Uncertainty may also be useful to the end-user—in fact it is sometimes more important than the prediction itself. For example, when performing statistical analyses on integrated data, it may be important to account for ER uncertainty as a source of error [TL15; KBS18]. While there have been some attempts to handle uncertainty using Bayesian methods, most current approaches do not scale to realistic problems [SHF16].

1.1.4 Computational efficiency and scalability

A prevailing challenge for ER applications at scale, is the need to balance computational efficiency without compromising accuracy (specifically recall of coreferent records). Formal treatments of ER have shown that the problem of obtaining a globally-optimal solution is NP hard [CKM00]. Most common ER methods instead solve the problem locally, without paying attention to global consistency. However, this is also computationally challenging, as it is necessary in general to compare each record with every other record to determine whether they are coreferent or not—a task that scales quadratically in the number of input records [Chr12a]. As a result, approximations are often used to avoid performing a quadratic number of comparisons, some of which are discussed in Section 2.4.4. While approximations can be effective, it is important to ensure that there are no severe consequences in terms of accuracy and robustness.

1.1.5 Statistically-sound evaluation

Evaluation is an essential part of the ER process, since there is a risk that automated methods may fail to achieve an acceptable level of accuracy. The relative scarcity of coreferent pairs of records presents a unique challenge for ER evaluation. Commonly used performance measures for ER are sensitive to coreferent (and predicted coreferent) pairs of records, and therefore exhibit high variance when they are estimated using an unbiased sample of labelled data. This makes standard unbiased sampling impractical in most circumstances, as vast quantities of labelled data are required to drive down the variance.

Ad-hoc sampling approaches are sometimes used as a practical alternative to unbiased sampling [Fu+12; Rah+14], however the resulting performance estimates may be misleading due to statistical bias. It is also common to avoid evaluation altogether

[MAS14; CBW17] or only consider precision, which is easier to estimate than recall [Xu+13]. There is an apparent need to develop methods for ER evaluation, which improve upon the efficiency of unbiased sampling methods, while ensuring that the performance estimates remain unbiased. This problem has received little attention in the ER literature to date, although there has been some related work in the broader context of machine learning [BC10; DM11; SLS10].

1.2 Research questions and contributions

In this thesis, we investigate several research questions motivated by the challenges for ER outlined above. We focus primarily on statistical solutions, given the inherent need to measure and account for uncertainty, and employ ideas from computer science to achieve favourable computational efficiency.

The first two research questions relate to Bayesian models as an attractive solution for performing ER. In contrast to discriminative models and rule-based approaches, Bayesian models naturally support uncertainty propagation, and they allow for the incorporation of prior knowledge. They are particularly effective in unsupervised or semi-supervised settings, as the priors have a regularising effect and sources of uncertainty are ideally reflected in the posterior predictions. However, inference for Bayesian models is generally computationally expensive, especially in the case of ER due to the quadratic scaling mentioned previously. As a result, recent Bayesian ER models [Ste15; SHF16] have seen limited adoption as they are not readily scalable. While deterministic blocking has been proposed as a solution by some [TL11; Sad14; SHF16], it may compromise recall and does not naturally fit within a Bayesian framework. This leads us to consider the following research question:

> **(RQ1)** *Can we develop more scalable and efficient inference algorithms for Bayesian ER models, without severely compromising accuracy?*

We explore this question in Chapter 3 using the `blink` ER model [Ste15] as a foundation. Specifically, we propose an auxiliary variable sampling scheme that effectively performs probabilistic blocking, while jointly inferring the other model parameters. Thus we obtain the benefits of blocking—reducing comparisons between records that are unlikely to be coreferent—without compromising correctness of the posterior approximation asymptotically.

When applying Bayesian models, model misspecification and sensitivity are important considerations. Ideally, the model should reflect reality as closely as possible and the predictions from the model should not depend sensitively on prior assumptions [BIR00]. In the Bayesian ER literature, there has been much debate about appropriate priors for the linkage or coreference structure, which describes how records are clustered into groups that are mutually coreferent (referring to the same entity). Some priors are known to be overly informative (e.g. the ones used in [BS14; Ste15; SHF16]), while commonly used priors for nonparametric mixture models are expected to exhibit asymptotic behaviour that is misspecified for ER [Mil+15]. While there has been some recent work on priors specifically designed for ER [Mil+15; Zan+16], little is known empirically about the performance of various priors. Related to this issue, is the fact that some ER models are

known to be sensitive to variations in hyperparameters [Sad14; Ste15]. Thus we consider the following research questions:

> **(RQ2)** *Are standard nonparametric clustering priors suitable for ER models? Moreover, how can we reduce the sensitivity of ER models to misspecified priors?*

We study these questions in Chapter 4 using the ER model from Chapter 3 as a basis. Specifically, we consider a broad family of clustering priors which correspond to Ewens-Pitman random partitions [Pit06]. We also propose changes to logic in the record distortion model and introduce an additional level of priors to improve model flexibility. Finally, we assess the impact of our proposed modelling changes by conducting a comprehensive empirical study.

The second set of research questions we consider are related to ER evaluation. In order for evaluation to be useful, it is important that practitioners can be confident in the results. However, standard unbiased methods require an impractically large quantity of labelled examples to produce precise estimates of performance. On the other hand, ad-hoc methods may yield more precise estimates using fewer labels, at the risk of injecting significant bias. It is clear that alternative methods are required to produce accurate and precise estimates of ER performance, without requiring an unreasonable quantity of labelled examples. There has been some work in this area in a machine learning context [BC10; DM11; SLS10], however existing methods have several limitations—e.g. they only support a limited class of performance measures, and some achieve limited gains in efficiency. This leads us to consider the following research question:

> **(RQ3)** *Can we design a framework for evaluating ER which is easy to use, label-efficient and backed by theoretical guarantees?*

We investigate this question in Chapter 5. Specifically, we propose a solution based on *adaptive importance sampling (AIS)* [Bug+17] which supports a broad family of performance measures (corresponding to transformations of vector-valued risk functionals). It manages the class imbalance associated with ER by selecting items to label using a biased, adaptive sampling policy. We prove that the estimates produced by our framework satisfy strong consistency, which guarantees that performance estimates converge to the unknown population performance asymptotically. In addition, we establish a central limit theorem which can be used to assess asymptotic efficiency and to compute approximate confidence regions.

Following on from this work, we consider the following research questions in Chapter 5:

> **(RQ4)** *How efficient is adaptive importance sampling compared to alternative methods for evaluating ER results? What improvements can be expected in terms of statistical precision/labelling budgets?*

We address these questions by conducting a thorough empirical study for several realistic evaluation tasks. We consider multiple instantiations of our adaptive importance sampling framework, and compare against static importance sampling [SLS10; Sch+16] and stratified sampling [DM11] baselines.

1.3 Thesis structure

The remainder of this thesis is structured as follows:

- Chapter 2 provides background material on ER, and reviews related work on automated methods for ER, scalability, and evaluation.
- Chapter 3 presents a principled approach to scaling and distributing inference for a Bayesian ER model called `blink` [Ste15].
- Chapter 4 investigates modelling refinements for the `blink` ER model, aimed at reducing sensitivity and improving goodness of fit.
- Chapter 5 presents a label-efficient framework for supervised evaluation based on adaptive importance sampling.
- Chapter 6 instantiates the framework proposed in Chapter 5 with Bayesian adaptive labelling policies, and presents empirical results on the expected efficiency gains for ER evaluation tasks.
- Chapter 7 concludes the thesis with a summary of the contributions and ideas for future work.

Chapter 2

Background and related work

The idea of entity resolution (or record linkage) can be traced back to the 1940s and 50s, when researchers began to explore the feasibility of linking vital records for public health research [Dun46; New+59]. Many advancements have been made since then, and entity resolution (ER) continues to be a problem of immense interest, in order to facilitate data cleaning and data integration in an increasingly data-driven society. In this chapter, we provide background information on the data integration problem (Section 2.1), showing how ER arises as a fundamental sub-problem. We then formally define ER in Section 2.2, before surveying automated ER methods in Section 2.4. In Section 2.5 we review crowdsourcing as a tool for acquiring ground truth for ER. Then in Section 2.6 we review current practices for evaluating ER systems. All of the concepts and related work discussed in this chapter are of general relevance to the topic of this thesis. We defer discussion of specialised concepts and related work to individual chapters.

2.1 Data integration

When data is spread across multiple disparate sources, extracting valuable information becomes challenging, due to differences in storage formats, differences in semantic representations, differences in source quality, and redundancies across sources. Data integration (also known as information integration) aims to resolve these issues, by providing unified access to integrated data under a common schema [DHI12]. Applications of data integration include, construction of knowledge bases in the life sciences [GS08], integration of data silos within large organisations [BH08], data sharing between government agencies [Amb+02], and integration of data from the web [CHK09].

Data integration has a long history as a research field, with some pioneering work dating to the early 1980s [Smi+81]. As a result, common architectures have emerged for designing modular data integration systems. Figure 2.1 illustrates a pipelined architecture for integrating multi-source structured data, which is pervasive in the literature [BN09; DS15]. It assumes that the sources describe entities from the *same domain*, with some overlap between the schemas. The data integration process typically proceeds in three stages:

(i) *Schema alignment.* The purpose of this stage is to detect semantically equivalent schema elements (attributes) in the data sources. The output is typically a

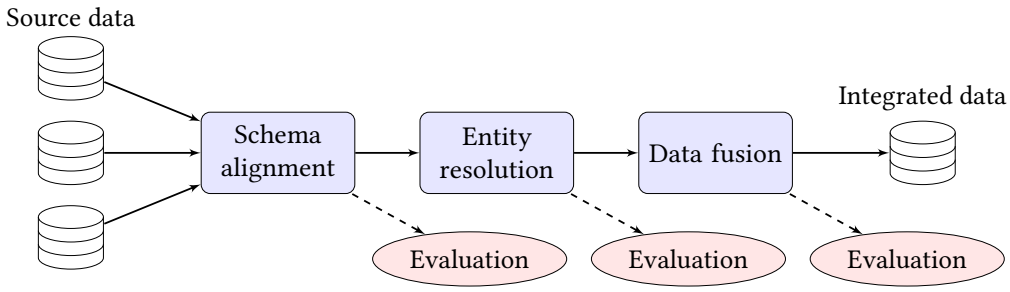


Figure 2.1: A data integration pipeline. Adapted from Dong and Srivastava [DS15].

set of schema mappings, which map the source schemas to a mediated schema. Due to heterogeneity of schema representations, this task is often non-trivial to automate. In simple cases, there is a one-to-one mapping between attributes in different sources—e.g. `phone_number` in one source may be semantically equivalent to `contact_number` in another source. More complex mappings may also be required—e.g. it may be necessary to concatenate `first_name` and `last_name` in one source in order to match `full_name` in another source.

(ii) *Entity resolution.* The objective of this stage is to identify records that refer to the same entity. The output can be represented as a partition of the records into clusters that are mutually *coreferent* (refer to the same entity). This task may be challenging for several reasons: (i) the attributes in the mediated schema may be insufficient to uniquely identify the entity represented in each record; (ii) there may be data errors and/or differences in semantic representation which are difficult to resolve automatically; and (iii) it may be computationally infeasible to compare all records to determine whether they are coreferent or not.

(iii) *Data fusion.* This stage aims to merge clusters of coreferent records, while resolving conflicting attribute values. Conflicts may arise for a multitude of reasons, including data entry errors, differences in semantic representations, and differences due to temporal variation. The output of this stage is a set of representative records—one for each entity—which conform to the mediated schema.

Each of these stages are challenging problems to solve in their own right, especially with minimal human intervention. When the data sources are assumed to come from an open-world, as is common in practice, it is difficult to make guarantees about the reliability of automated solutions. Evaluation is therefore essential, in order to ensure outputs at each stage of the pipeline meet acceptable standards of quality. This is often done by comparing outputs to human-generated ground truth.

There is a significant body of research covering data integration, and the specific stages defined above. The primary focus of this thesis is on methods for performing and evaluating entity resolution, which we review in the upcoming sections. A broad introduction to the field of data integration is provided in books [DHI12; DS15] and surveys articles [BH08; HRO06]. We encourage readers who are interested in schema alignment and data fusion to consult survey articles by Bernstein et al. [BMR11] and Bleiholder and Naumann [BN09].

Remark 2.1 (Data integration terminology). *Since data integration research is conducted independently in different communities (e.g. databases, statistics, natural language processing), various terms are used to refer to the same concepts. Schema alignment is closely related to* schema matching, schema mapping *and* schema integration *[BMR11]. Entity resolution is largely synonymous with* data matching, record linkage, merge/purge, deduplication *[NH10; Chr12c] and* identity uncertainty *[Pas+02]; and is closely related to* coreference resolution *[HK07]. Data fusion is also known as* record canonicalisation *[Cul+07],* data merging, conflict resolution *[DN09], and* truth-finding *[Zha+12].*

2.2 Entity resolution

When real-world entities are referenced in data, their identities are often obscured. This occurs whenever a reference to an entity is not accompanied by a *unique identifier (UID)* that is consistent across all data sources of interest. Instead, entities are often referenced by *quasi-identifiers (QIDs)*—pieces of information that are correlated with identity which don't satisfy uniqueness guarantees [Dal86]. For example, when a person is recorded in a database, they may only provide basic personal details, such as name, date of birth and zip code. While these details serve as QIDs, they are may be unreliable for identification purposes. This is because the QIDs may vary with time (e.g. if the person moves to a different zip code or changes name), they may be subject to errors or semantic variation (e.g. typographical errors), and/or they may coincidentally match the details of other people (e.g. for a very common name).

In order to clean and integrate data from multiple sources, it is important to deal with these ambiguities, so that all data related to the same entity can be consolidated. This task—of identifying the entities represented in data—is a key step in the data integration process (see Figure 2.1), which we refer to as *entity resolution (ER).* However, it is also known under other names, including *record linkage* and *deduplication* (see Remark 2.1). Below we provide a formal definition of the ER problem.

Definition 2.1 (Entity resolution). *Consider a set of sources $\mathcal{S}$ providing a set of records (or entity-mentions) $\mathcal{R}$. Let $\mathcal{P}$ denote a homogenous relation on the set $\mathcal{R}$ such that:*

- *$(r, r') \in \mathcal{P}$ for any pair of records $r \neq r' \in \mathcal{R}$ that are* coreferent *(referring to the same entity), and*
- *$(r, r') \notin \mathcal{P}$ for any pair of records $r \neq r' \in \mathcal{R}$ that are* non-coreferent *(referring to distinct entities).*

The entity resolution (ER) *problem is to approximate the true coreference relation $\mathcal{P}$ (assumed unknown) with a predicted relation $\hat{\mathcal{P}}$.*

The ER problem is trivial to solve when UIDs are included with each record or entity mention—one can simply perform a database join on the UID attribute. However, as noted previously, there are many practical cases where globally consistent UIDs are unavailable, and one must exploit patterns in the data (e.g. agreement between QIDs) in order to approximately solve the problem. Automated methods for performing ER have been studied since the 1950s [New+59], and it continues to be an active area of research in the statistics, database, natural language processing and machine learning communities. We

provide a comprehensive survey of ER methodology in Section 2.4. In the remainder of this section, we review a pipelined architecture for solving ER as a classification problem that is widely used in practice.

Remark 2.2 (Privacy considerations). *ER can be applied, intentionally or unintentionally, as an attack on privacy [NS08; AG09; CRT17]. Linkage attacks occur when a data set is released with (sometimes very subtle) QIDs and sensitive attributes such as health information, under the false assumption that UIDs cannot be recovered. Linking such "deidentified" datasets to a public dataset comprising both UIDs and matching QIDs reidentifies individuals and maps their identities to sensitive attributes. It is important that the techniques in this thesis are not used to intentionally breach privacy of individuals, and that releases of sensitive data be aware of the risk of reidentification.*

2.2.1 Entity resolution as a pairwise classification problem

Although ER is most naturally cast as a clustering problem, it can be formulated as a binary classification problem, albeit with some caveats. To do this, we treat pairs of records from the product space $\mathscr{R} \times \mathscr{R}$ as observations and attempt to classify them as *matches* (referring to the same entity) or *non-matches* (referring to distinct entities). There are three important benefits of this formulation. Firstly, it allows ER to be solved using black-box supervised classifiers—e.g. neural networks [Mud+18] or support vector machines [BM02]. Secondly, it is amenable to parallelisation as each pair of records can be classified independently [CCH04]. Thirdly, it integrates nicely with blocking, which improves computational tractability by ignoring pairs of records that are unlikely matches [Chr12b].

However, there are also downsides which result from treating the pairs of records independently [DM05]. Firstly, the pairs are not independent and identically distributed, which means standard theoretical guarantees do not hold. Secondly, it is not possible to exploit patterns at the entity-level when making predictions. Thirdly, there may be conflicts among the predictions for each pair. For example, if the classifier predicts records A and B are a *match*, records B and C are a *match*, but records A and C are a *non-match*, then the predicted relation $\hat{\mathscr{P}}$ is intransitive. These conflicts can be resolved in a post-processing step, which we refer to as *clustering* (following [DS15]).

2.2.2 Entity resolution pipeline

Many ER systems follow a standard pipelined architecture that is well-suited to the pairwise formulation described above [NH10; Chr12c; DS15]. The input to the pipeline is a collection of data sources, along with mappings from the source schemas to a mediated schema (see *schema alignment* in Section 2.1). The source data then flows through the pipeline in a sequence of four stages, as illustrated in Figure 2.2 and described below:

(i) *Pre-processing.* Prior to ER, the source data may reside on different physical systems and/or be stored in different formats. In this step, an extract-transform-load (ETL) process is executed to load the data into a single database or data warehouse [NH10]. During the ETL process, the data sources are mapped to the mediated schema, and cleaning and standardisation may be applied to reduce noise and semantic heterogeneity.

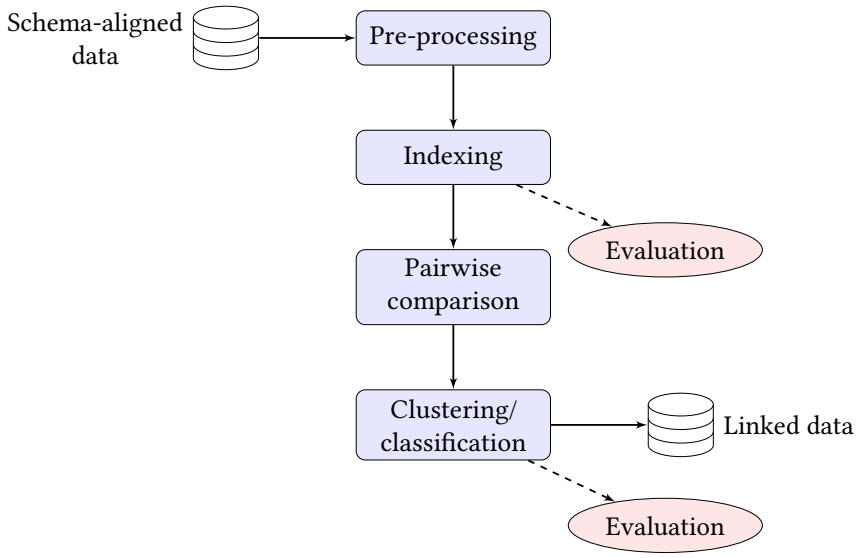


Figure 2.2: A typical entity resolution pipeline. Adapted from Christen [Chr12c].

(ii) *Indexing/blocking.* This step is included to improve computational tractability, particularly when there is a large volume of source data. Ideally, we would like to compare all pairs of records in the product space $\mathscr{R} \times \mathscr{R}$ to classify them as *matches* or *non-matches*, however this quickly becomes infeasible as the number of records $|\mathscr{R}|$ grows. Since the vast majority of record pairs are likely to be obvious non-matches, it is in some sense wasteful to compare them. The purpose of this step is to efficiently filter out the obvious non-matches, leaving behind a set of candidate record pairs that can be classified more carefully in the next step. Various methods can be used to produce the set of candidates, such as blocking and locality sensitive hashing [Ste+14]. We review some of these in Section 2.4.4.

(iii) *Pairwise comparison.* In this step, each pair of records in the candidate set is compared to produce a real-valued score, that is ideally correlated with the match likelihood. This is often done by comparing the values for each attribute—e.g. by applying a similarity or distance measure. The resulting attribute similarities or distances form a feature vector, which is then fed into a binary classifier to produce a real-valued score.

(iv) *Clustering/classification.* In this step, the match scores for the record pairs are processed to produce a predicted relation $\hat{\mathscr{P}}$. While it is common to compute $\hat{\mathscr{P}}$ by applying a threshold to the scores, the resulting relation may be intransitive. Various methods can be applied to produce a transitive predicted relation—e.g. taking the transitive closure [HS98] or applying hierarchical clustering algorithms based on the match scores [CR02]. Constraints on the relation can also be imposed at this stage. For example, if the sources are known to be free of duplicates, it may be desirable to enforce a one-to-one matching constraint, so that each mutually coreferent cluster of records contains at most one record from each source [ZRG15].

Evaluation is also a crucial step in the pipeline, as indicated in Figure 2.2. It is common

to evaluate blocking separately, as it may have a large impact on recall and computational efficiency [CG07]. It is also essential to evaluate the quality of the final output: the predicted coreference relation $\hat{\mathcal{P}}$. We review evaluation of ER in Section 2.6.

2.3 Measures for comparing attribute values

A fundamental sub-problem for entity resolution is determining whether a pair of attribute values are semantically equivalent [ME96]. This can be challenging due to differences in representations and data entry errors. A common instance of this problem occurs when comparing names of people, which may be abbreviated or recorded with spelling or typographical errors. For example, a person named "Nathaneal" may also use the abbreviated name "Nate", or their name may be erroneously recorded as "Nathaniel". These variations can be identified and matched using string similarity/distance measures. Various measures are used in ER systems, depending on the types of values to be compared, and the expected variations/errors [ME96; CRF03].

The generative ER models we present in Chapters 3 and 4 leverage string similarity/distance measures to model attribute-level distortions. In this section, we provide background information on three categories of string measures—character-level measures, token-level measures and hybrid measures—some of which are used in experiments in Chapters 3 and 4. We let u and v denote the strings to be compared and use the notation:

- $|u|$ to denote the length of the string;
- u_i to denote the character at the i-th position in string u; and
- $u_{i:j}$ to denote the sub-string of u from the i-th to the j-th position inclusive.

We define a string similarity/distance measure to be a function that maps a pair of strings to a non-negative real-valued number. In an ER context, a high similarity (large distance) means the pair of strings are likely (unlikely) to have the same semantic meaning.

2.3.1 Character-level measures

In natural language contexts, character-level measures are commonly used to compare short strings or individual words. They are generally not suited for comparing long multi-word strings, as they are highly sensitive to word-level differences [BM03]. We consider two example measures below.

Levenshtein distance. A popular class of character-level measures are based on the concept of edit distance. These measures compute the minimum cost of edit operations to convert one string into another. The simplest edit distance metric is the *Levenshtein distance*, for which the allowed operations are character insertions, character deletions and character substitutions. It assigns a unit cost to each operation. The distance can be

defined recursively as follows [Lap00]:

$$\mathrm{dist}_{\mathrm{Ed}}(u, v) = \begin{cases} \max(|u|, |v|), & \text{if } \min(|u|, |v|) = 0, \\ \min \begin{cases} \mathrm{dist}_{\mathrm{Ed}}(u_{1:(|u|-1)}, v) + 1, \\ \mathrm{dist}_{\mathrm{Ed}}(u, v_{1:(|v|-1)}) + 1, \\ \mathrm{dist}_{\mathrm{Ed}}(u_{1:(|u|-1)}, v_{1:(|v|-1)}) + \mathbb{I}\left[u_{|u|} \neq v_{|v|}\right], \end{cases} & \text{otherwise.} \end{cases}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. The distance can be bounded below by $\mathrm{abs}(|u| - |v|)$ and bounded above by $\max\{|u|, |v|\}$.

Normalised Levenshtein distance. The range of the Levenshtein distance varies depending on the length of the input strings. However, in some circumstances it is desirable to compare the strings over a fixed range, that is independent of the input strings. The *normalised Levenshtein distance* can be used for this purpose. Yujian and Bo [YB07] notes that there are numerous ways of performing the normalisation, however they advocate the following variant which satisfies the axioms for a distance metric:

$$\mathrm{dist}_{\mathrm{nEd}}(u, v) = \begin{cases} 0, & \text{if } u = v, \\ \frac{2\mathrm{dist}_{\mathrm{Ed}}(u,v)}{|u|+|v|+\mathrm{dist}_{\mathrm{Ed}}(u,v)} & \text{otherwise.} \end{cases} \tag{2.1}$$

The range of the distance is $[0, 1]$. The normalised Levenshtein similarity is defined as $\mathrm{sim}_{\mathrm{nEd}}(u, v) = 1 - \mathrm{dist}_{\mathrm{nEd}}(u, v)$.

Jaro-Winkler similarity. Jaro [Jar89] proposed a string similarity measure which accounts for common types of human errors when recording names. A variation of the measure was proposed by Winkler [Win90], which is known as the *Jaro-Winkler similarity*. It places more weight on matching characters at the beginning of the string, in order to better capture similarities between abbreviated names, variations of the same name and typographical errors (which are more likely to occur at the end of the string).

2.3.2 Token-level measures

Another class of measures is based on a tokenised representation of the strings. Natural language strings are commonly tokenised by splitting the string on white space (i.e. into words) or using character n-grams. For example, when the string "North Carolina" is tokenised using white space as a delimiter, the result is ["North", "Carolina"]. When it is tokenised using tri-grams the result is ["Nor", "ort", "rth", 'th_', "h_C", "_Ca", "Car", "aro", "oli", "lin", "ina"], where "_" denotes a space. The resulting tokenised representations can be compared in various ways. We consider two examples below, both of which are insensitive to the token order.

Jaccard similarity. A simple way of comparing the tokenised strings is to compute the fraction of overlapping tokens. Let U denote the set of (unique) tokens from string u and V denote the set of (unique) tokens from string v. The *Jaccard similarity* measures the

similarity of the two sets U and V as follows:

$$\mathrm{sim}_{\mathrm{Jac}}(U, V) = \begin{cases} 1, & \text{if } U = V = \emptyset \\ \frac{|U \cap V|}{|U \cup V|}, & \text{otherwise.} \end{cases} \tag{2.2}$$

The range of the similarity is $[0, 1]$.

Cosine similarity. Another way of measuring the similarity between the tokenised strings is to encode the strings in a vector space. Various vector space model representations are used in natural language processing, with the term frequency-inverse document frequency (TF-IDF) model [SWY75] being a classic example. It represents a string in a vector space of dimension equal to the vocabulary size, where the term weights are computed as products of the word (token) frequency in the string and the inverse frequency in the document (all strings). Assuming the tokenised strings u, v are represented as vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, the *cosine similarity* is defined as:

$$\mathrm{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} \mathbf{u}_i \mathbf{v}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{u}_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{v}_i^2}}.$$

The range of the similarity is $[0, 1]$.

2.3.3 Hybrid measures

While token-level measures can be used to compare multi-word strings, they do not allow for approximate matches between the tokens. Hybrid measures address this issue, by comparing the strings at both the token- and character-level.

Monge-Elkan measure. Monge and Elkan [ME96] proposed a hybrid similarity measure for comparing multi-word strings (tokenised into words). It relies on an *inner* similarity measure $\mathrm{sim}_{\mathrm{inner}}$ to measure the character-level similarity between pairs of tokens. Letting $\mathbf{u}$ denote the ordered list of tokens extracted from u (and similarly for v), the *Monge-Elkan measure* is defined as:

$$\mathrm{sim}_{\mathrm{ME}}(\mathbf{u}, \mathbf{v}) = \frac{1}{|\mathbf{u}|} \sum_{i=1}^{|\mathbf{u}|} \max_{j \in \{1,\ldots,|\mathbf{v}|\}} \mathrm{sim}_{\mathrm{inner}}(\mathbf{u}_i, \mathbf{v}_j), \tag{2.3}$$

where $\mathbf{u}_i$ denotes the i-th token in the list $\mathbf{u}$. In words, this measure finds the closest matching token in $\mathbf{v}$ for each token in $\mathbf{u}$ based on the inner similarity measure. The inner similarities for the closest matches are then averaged by taking the arithmetic mean. As a result, the range of the measure is the same as the range of the inner similarity function.

2.3.4 Learning measures under supervision

We have seen that a variety of measures are available for comparing strings, however some measures are known to perform better than others in different scenarios [CRF03]. In addition, several of the measures depend on tunable parameters—e.g. the cost of edit operations for Levenshtein distance, or the vector space model for cosine similarity. While

domain experts may be able to select appropriate measures, an alternative approach is to automatically learn a suitable measure under supervision [RY98; BM02; BM03].

Ristad and Yianilos [RY98] proposed a model for corrupted strings based on a memoryless stochastic transduction. Their model incorporates a set of edit operations (insertion, deletion and substitution), each of which has a different probability of occurring. These probabilities can be inferred from a corpus of examples using the expectation-maximisation algorithm. The similarity between a pair of strings can then be measured based on the likelihood under the inferred parameters. A similar model was studied by Bilenko and Mooney [BM03], which added affine gaps (misalignments) to the set of edit operations. It was found to outperform regular edit distance in ER experiments.

Bilenko and Mooney [BM03] also proposed a discriminative method for learning token-based string similarity measures. They formulated the learning task as a binary classification problem, where the input is the element-wise product of a pair of token vectors, and the output is a binary match/non-match label. After training a kernelised support vector machine (SVM), they defined the token-based similarity to be the normalised distance from the SVM decision boundary.

2.4 Entity resolution methods

Automated methods for performing ER date back to the late 1950s, when early computers became capable of performing limited data processing tasks [New+59]. Much progress has been made since then, and a variety of methods are now in use, including manual rule-based methods, statistical models and generic supervised machine learning algorithms. In order to provide context for the generative ER models explored in Chapters 3 and 4, we review automated ER methods across two broad categories: discriminative approaches (Sections 2.4.1 and 2.4.2) and generative approaches (Section 2.4.3). We refer the reader to surveys by Winkler [Win06] and Getoor and Machanavajjhala [GM12] for further coverage of ER methods. Finally, in Section 2.4.4 we review methods for managing scalability of ER, which are relevant to our work in Chapter 3.

2.4.1 Discriminative classification approaches

Many ER solutions are based on discriminative classification models under the pairwise formulation of ER (see Section 2.2.1). These models learn to discriminate between matching and non-matching record pairs by generalising from labelled examples. Conventionally, discriminative models are trained on a set of labelled examples that is prepared in advance called a *training set* (examples in ER include [GSR96; TKM01; BM02; Wil11; Kon+16; Mud+18; Ebr+18]). While this approach is generally effective—particularly if labelled examples are already available—large training sets are often required to achieve good performance. In an effort to reduce labelling requirements, researchers have explored active learning [SB02; Bel+12; AGK10; QPS17; Kas+19] and transfer learning [NRG12; Kas+19]. We review work in each of these settings below.

Feature-based methods. Many supervised ML algorithms rely on feature engineering in order to achieve good performance. In the pairwise ER formulation, informative features are often generated using attribute-level similarity/distance scores, which are

selected based on domain knowledge. Various discriminative classifiers have been used with similarity/distance-based features in the literature, including decision trees [GSR96; TKM01], support vector machines [BM02] and single-layer perceptrons [Wil11]. The feature engineering proposed by Wilson [Wil11] is unique among these works, as it allows for missing values and more complex interactions between attributes. Bilenko and Mooney [BM02] proposed an additional level of supervision, by learning custom distance measures for each attribute. Most discriminative models require significant quantities of training data—e.g. Wilson [Wil11] used approximately 50,000 hand-labelled pairs to train an ER system for genealogical records. More recently, Konda et al. [Kon+16] designed an integrated system called `Magellan` for performing supervised ER. It aims to streamline ER workflows by supporting all steps of a typical ER pipeline (see Section 2.2.2) in an interactive environment. It includes support for automated model selection using classifiers implemented in scikit-learn [Ped+11].

Deep learning. Recent works automate feature engineering for ER using deep neural networks [Mud+18; Ebr+18; Kas+19]. Mudgal et al. [Mud+18] proposed a neural network architecture called `DeepMatcher` consisting of three modules: (i) an attribute embedding module, (ii) an attribute similarity representation module, and (iii) a classifier module. The first two modules are responsible for learning a distributed representation [GBC16] of the record pair, so that the pair can be classified in the third module using a simple logistic regression layer. The authors discussed design choices for each module, and tested four variations empirically. Their deep learning approach achieved similar ER accuracies as `Magellan` [Kon+16] for structured data, while taking significantly longer to train. However, they observed performance gains on semi-structured data with textual attributes. This aligns with performance gains enjoyed by deep learners more broadly in natural language processing applications.

Ebraheem et al. [Ebr+18] proposed a deep learning model with a similar architecture as [Mud+18]. Their model—called `DeepER`—relies on pre-trained word embeddings, and a bi-directional recurrent neural network with long-term short-term memory units for learning distributed representations. Their empirical observations were similar to [Mud+18]: they observed little to no improvement in performance compared to `Magellan` for structured data [Kon+16], but significant gains for dirty data with longer text attributes.

Kasai et al. [Kas+19] argued that deep learning models for ER are overly data-hungry. They proposed a solution that incorporates transfer learning and active learning, which are both aimed at reducing requirements for labelled data (see upcoming paragraphs). In order to incorporate transfer learning effectively, they designed an architecture that encourages data source-independent distributed representations. This is necessary to guard against idiosyncratic representations that tend to arise within individual sources. Their empirical studies demonstrated competitive performance on three data sets (compared to `DeepMatcher` [Mud+18] and an SVM) using an order of magnitude fewer labels.

Active learning. When training data is unavailable in advance, it must be collected manually by querying human annotators. In typical applications of supervised ML, training data is prepared by selecting examples to label uniformly at random. However, this is problematic for ER due to severe class imbalance between matches and non-

matches [QPS17]. In order to deal with the imbalance, numerous authors have proposed ER methods based on active learning, where the learner is responsible for selecting informative examples to label [SB02; TKM01; de +10; IB12; QPS17; Kas+19]. This can result in reduced sample complexity (improved label efficiency) compared to conventional supervised ML.

Many active learning strategies/algorithms have been proposed (see survey [Set09]). The query by committee strategy has been used to design several ER methods based on active learning [SB02; TKM01; de +10; IB12]. Under this strategy, the learner maintains a "committee" of models, and examples are selected for labelling for which the committee's predictions are in maximum disagreement. However, [AGK10] argued that the approaches presented in [SB02; TKM01] are limited, as they use a 0-1 loss which is ill-suited for imbalanced problems. To deal with this limitation, they proposed a method that incorporates precision and recall in the objective function, which is tailored for linear classifiers and decision trees. Their objective function maximises recall while requiring that the precision exceeds a user-specified threshold, thereby giving the user more control over the precision-recall trade-off.

Bellare et al. [Bel+12] designed an active learning method with the same objective function as [AGK10], however their method is as a wrapper around black-box active learners. They derive an upper bound on the label complexity, which is at most $O(\log^2 N)$ times the label complexity of the black box. When combined with the Importance-Weighted Active Learning (IWAL) algorithm [BDL09], their method achieves sublinear label complexity. In addition, their empirical results demonstrate reductions in computational complexity for large numbers of attributes, and better satisfaction of the objective compared to [AGK10].

More recently, Qian et al. [QPS17] proposed an active learning system for ER with a vastly different architecture. Rather than learning a single classifier, they instead aim to learn a collection of rules, each of which returns high-precision predicted matches. At each stage of the learning process, their system proposes a new rule (based on the existing labelled data) and decides whether to accept or reject the rule based on newly labelled examples. When collecting labels, their system preferentially selects likely false positives and false negatives using a heuristic method. They compare empirically with [AGK10], and observe improved recall with less variance in the results.

Active learning can be viewed as an analogue of the evaluation problem we consider in Chapters 5 and 6. Both problem settings are concerned with improving label efficiency by allowing the learner/evaluator to actively select examples to label. However, in the case of evaluation, additional care is needed to ensure that the active selection of examples does not bias the results.

Transfer learning. Another strategy for reducing labelling requirements is to rely on knowledge gained from solving a related ER task. This strategy is known as *transfer learning* in the ML community [TS10]. Negahban et al. [NRG12] leveraged transfer learning to design a method for performing ER on multiple data sources from the same domain. Under a 1-1 match constraint, their method learns a separate linear classifier for matching records across each pair of sources. They transfer knowledge across the pairs of sources by learning the classifiers jointly, assuming a decomposition of the classifier weights into three components: one that captures general patterns, one that captures source-specific patterns and one that captures pairwise deviations. Their experiments

on movie listings from multiple sources, demonstrated improvements in label efficiency compared with a baseline approach that learns each classifier separately.

Kasai et al. [Kas+19] applied transfer learning in combination with active learning, to improve the accuracy of deep learning models for ER in low-resource settings. Their neural network architecture encourages distributed representations of record pairs that are source-independent—i.e. that ignore differences in semantic representation across sources. They consider the problem of performing ER on a new source using a neural network that was pre-trained on data for similar sources from the same domain. They find that the source-independent representation yields a small improvement in ER accuracy in some cases, however ultimately they conclude that adaptation is necessary using data collected for the new source pairs. As summarised previously, after combining transfer learning with active learning for adaptation, their approach achieves accuracies that are competitive with `DeepMatcher` [Mud+18] using an order of magnitude fewer labels.

2.4.2 Discriminative clustering approaches

While the classification approaches surveyed in the previous section are convenient, they make the simplifying assumption that matching pairs of records are independent. This is clearly not true, as each entity may be associated with multiple matching pairs. Since this dependence is not accounted for, classification approaches may produce intransitive coreference relations—i.e. where the pairwise coreference predictions are in conflict with one another. In Section 2.2.2, we noted that this issue may be addressed by applying a discriminative clustering algorithm to pairwise classification scores as a post-processing step. We review some of these discriminative clustering algorithms below, noting that many originated in a broader context (outside ER). In addition, we review discriminative clustering models that are specially designed for ER, all of which are based on probabilistic models.

Distance-based clustering. When similarity or distance scores are available between pairs of records, ER can be solved in an ad-hoc manner using hierarchical clustering algorithms. The similarity or distance scores may be provided by a trained classifier, or a hand-crafted similarity or distance measure, such as the ones presented in Section 2.3. Several works have explored the use of greedy agglomerative clustering algorithms using different linkage criteria [MNU00; BBS05; CGM05]. Bilenko et al. [BBS05] concluded that the single-linkage criterion is ill-suited for ER, as it results in over-linkage, while the complete-linkage criterion yielded good results. On the other hand, Chaudhuri et al. [CGM05] argued that specialised linkage criteria are required for ER, owing to the smaller cluster sizes. They proposed two criteria based on the intuition that coreferent records are expected to be mutual nearest neighbours, and that the local neighbourhood of a coreferent cluster is empty or sparse. An alternative to hierarchical clustering is correlation clustering, where the goal is to find a clustering that globally maximises agreements based on similarity scores [BBC04]. While correlation clustering is NP-hard [ACN08], effective heuristic methods have been proposed for coreference resolution [SNL01; NC02; EC08].

Probabilistic models. Discriminatively-trained undirected probabilistic graphical models have been proposed as a principled alternative to post-hoc clustering based on classifier scores [Wel+04; CM05; MW04; DHM05; SD06]. These models avoid making pairwise coreference predictions as an intermediate step, and instead make predictions for the coreference relation, conditional on observed data. Several discriminative models have been proposed which are based on conditional random fields (CRFs) [Wel+04; MW04; CM05]. McCallum and Wellner [MW04] noted that CRFs allow for richer dependence structures than generative models (reviewed next), as the practitioner is not required to explicitly model the dependence structure. However, unlike generative models, CRFs require labelled training data. In some cases, inference for CRF ER models can be cast as a weighted graph partitioning problem. Singla and Domingos [SD06] applied a different class of probabilistic models to ER called Markov logic networks (MLNs), which combine probabilistic graphical models with first-order logic. These models incorporate soft constraints expressed as formulas in first-order logic, which can be specified manually or learnt from data. While they are arguably more expressive than CRFs, inference is more complicated. Singla and Domingos propose an approximate MAP inference method incorporating a weighted satisfiability solver and gradient ascent.

2.4.3 Generative approaches

An alternative paradigm to discriminative models are *generative models*, which attempt to model the data generation process in order to make predictions. Since labelled examples are generally not required to train generative models, they are particularly attractive in unsupervised and semi-supervised settings. While the reduced need for supervision is advantageous, generative models are not without downsides. In particular, it may be challenging to accurately model the generative process (especially for complex dirty data), and training tends to be more computationally demanding than for discriminative models. Our work in Chapters 3 and 4 contributes to these challenges—by improving the scalability and accuracy of a generative ER model proposed by [Ste15].

Fellegi-Sunter (FS) model. Fellegi and Sunter [FS69] proposed an influential probabilistic framework for ER/record linkage across a pair of data sources, building on earlier work by Newcombe et al. [New+59]. Under the pairwise formulation of ER (see Section 2.2.1), they derived optimal decision rules for predicting matching/uncertain/non-matching record pairs, based on a generative model for comparison vectors associated with each pair. They defined comparison vectors $\gamma(r_a, r_b) = [\gamma_1(r_a, r_b), \ldots, \gamma_K(r_a, r_b)]$ as functions of the K attributes for each record r_a, r_b in the pair. In the simplest case, a comparison vector can be viewed as a binary vector, where the i-th entry $\gamma_i(r_a, r_b)$ represents agreement/disagreement on the i-th attribute.

Letting M and U denote the unknown sets of matching/non-matching record pairs, Fellegi and Sunter defined the so-called m- and u-probabilities for each record pair (r_a, r_b):

$$m(\gamma(r_a, r_b)) = P(\gamma_i(r_a, r_b) | (r_a, r_b) \in M) \quad \text{and} \quad u(\gamma(r_a, r_b)) = P(\gamma_i(r_a, r_b) | (r_a, r_b) \in U).$$

These probabilities can be used to classify the pair as follows:

$$y(r_a, r_b) = \begin{cases} \text{match}, & \text{if } \frac{m(\gamma(r_a,r_b))}{u(\gamma(r_a,r_b))} > \tau_u, \\ \text{uncertain match}, & \text{if } \tau_l \leq \frac{m(\gamma(r_a,r_b))}{u(\gamma(r_a,r_b))} \leq \tau_u, \\ \text{non-match}, & \text{if } \frac{m(\gamma(r_a,r_b))}{u(\gamma(r_a,r_b))} > \tau_l, \end{cases}$$

for some thresholds $0 \leq \tau_l \leq \tau_u < \infty$. While Fellegi and Sunter show that this rule is optimal in terms of statistical power, the result is based on several assumptions that do not hold in practice (see [TL11]).

We discuss two practical issues here. First, we note that there is no established method for selecting the decision thresholds τ_l and τ_u, which are important for balancing precision and recall. While unsupervised [BR95] and heuristic methods [SBP11] have been proposed, more reliable results are likely to be obtained if training data is available. Second, we note that it is often necessary to make simplifying assumptions about the form of the conditional distributions for the comparison vectors. In practice, the components of the comparison vectors are usually assumed to be independent conditional on the match/non-match status.[1] It is then possible to infer maximum likelihood estimates for the model parameters using the expectation-maximisation (EM) algorithm in an unsupervised setting [Win00]. Inference methods have also been proposed for more general dependence structures (see for e.g. [Win89; LR01; RC04]).

Extensions to the FS model. Winkler [Win06] surveys a long line of research building on the FS model. We highlight two recent extensions here. The first is a generalisation of the FS model to multiple data sources [SF13]. While the original FS model can be applied to link multiple data sources in a pairwise fashion, it produces intransitive coreference relations, which must be corrected in a post-processing step. The generalisation proposed by Sadinle and Fienberg [SF13] does not suffer from this problem, as it operates on the product space of S-tuples (assuming S data sources), thereby guaranteeing transitive relations. The decision rules and inference algorithms are natural generalisations of those used for the original FS model, however scalability is a concern since the product space grows exponentially as a function of S.

Sadinle [Sad14] proposed another extension to the FS model which is tailored for deduplication of a single source. It combines the FS likelihood with a transitivity constraint on the links between records. Unlike most variants of the FS model, the pairwise comparison vectors are permitted to encode multiple levels of agreements—not only binary agreement/disagreement. In addition, priors are placed on the model parameters—a uniform prior on the linkage structure and truncated beta priors on the m- and u-probabilities. Due to the additional complexity, maximum likelihood estimation of the model parameters is not possible using the EM algorithm. Instead, Sadinle proposes a Markov chain Monte Carlo algorithm, handling scalability through blocking (see Section 2.4.4).

Bayesian models. Bayesian models (also known as *Bayesian networks* and *directed graphical models*) provide a formalism for reasoning under uncertainty based on a generative model of the problem of interest [PR03]. Most Bayesian models in the ER literature

[1] In this case, the model is equivalent to Naïve Bayes [Wil11].

follow a similar high-level structure: the entities are modelled as latent objects, which are randomly selected to be represented in data sources. The representations are often determined from latent entity attributes, which may be subject to corruption.

Pasula et al. [Pas+02] proposed one of the first Bayesian models for ER, which is tailored for unstructured citation data. The citations are assumed to be generated by sampling publications uniformly at random from a latent "population" of publications, which are in turn generated from a latent population of authors. Both populations are assumed to be finite, with a vague prior placed on the population sizes. Publications and authors are selected according to a uniform prior. The unstructured nature of the data presents challenges for segmentation of publication attributes, which are incorporated in the generative process. The priors on textual attributes are based on bigram models, which facilitate modelling of character-level distortions. The model was shown to yield a significant improvement in ER accuracy compared to hierarchical clustering in an unsupervised setting.

Daumé and Marcu [DM05] and Bhattacharya and Getoor [BG06] also proposed specialised Bayesian ER models for citation data. The model proposed by Daumé and Marcu [DM05] can be viewed as a generalisation of [Pas+02], which incorporates a nonparametric Dirichlet Process prior on the entity population, while also allowing for supervised training. Bhattacharya and Getoor [BG06] extended Latent Dirichlet Allocation (originally formulated for topic modelling) to perform ER of authors in citation data. They assume that the authors (entities) are related through a latent group structure. Their experiments demonstrate that the latent group structure helps to resolve ambiguous author identities in an unsupervised setting, although the improvement over hierarchical clustering is marginal in some cases.

Various Bayesian models have been proposed for ER of structured data [Lar05; TL11; Sad14; Ste15; SHF16; STL18]. Tancredi and Liseo [TL11] proposed a model for performing ER on a pair of data sources with categorical attributes, under the assumption that there are no duplicates within each source (a 1-1 matching constraint). It can be viewed as an alternative to the Fellegi-Sunter (FS) model [FS69], which describes the generative process for the raw data, rather than pairwise comparison data. The model incorporates a finite population of entities of unknown size, and assumes each data source is generated by sampling entities without replacement from the population. The record attributes are assumed to be copied from latent entity attributes according to a hit-miss distortion model [CH90]. The model was observed to achieve lower bias for various parameters of interest, compared to the FS model, and superior coverage probabilities for interval estimates.

Steorts et al. [SHF16] proposed a model called `SMERED` for ER of multiple data sources with categorical attributes. It is similar in spirit to the model proposed by Tancredi and Liseo [TL11], however the 1-1 matching constraint between sources is optional and the size of the latent population of entities is assumed to be known. Steorts [Ste15] proposed an extension of `SMERED` called `blink` that incorporates distance functions in the distortion model. This improves the flexibility of the distortion model—e.g. allowing for modelling of character-level distortions based on edit distance. The model was observed to perform well on personal data (with names) in an unsupervised setting, achieving lower error rates than baseline supervised methods. More recently, another variant of `SMERED` was proposed that removes the assumption of a fixed (finite) population size, by employing a

nonparametric Pitman-Yor process prior [STL18].

Microclustering priors. When specifying a Bayesian model for ER, it is necessary to specify a prior distribution on the coreference relation, which can alternatively be represented as a partition of the records. Some Bayesian ER models assume that links from records to entities are sampled according to a uniform prior over a finite population of entities [Pas+02; TL11; Ste15; SHF16], while others adopt a nonparametric approach [DM05; BG06; STL18]. Miller et al. [Mil+15] pointed out that both of these classes of priors exhibit asymptotic behaviour that is ill-suited for ER applications. In particular, the priors implicitly assume that the number of records linked to each entity grows *linearly* with the total number of records, while empirical observations call for *sub-linear* growth. They introduced the term *microclustering* to refer to priors that exhibit sub-linear growth.

Several microclustering priors have been proposed with varying behaviours and trade-offs [Zan+16; KJ16; BCT17]. Zanella et al. [Zan+16] proposed priors that are related to finite Gibbs partitions [GP06; De +15]. The resulting priors are exchangeable, however they do not satisfy Kolmogorov consistency. Benedetto et al. [BCT17] proposed non-exchangeable priors using a construction based on completely random measures and a Poisson embedding of the random partition. Their priors exhibit sub-linear growth, and incorporate a hyperparameter that controls the growth rate. However, it is unclear whether their priors can be applied in situations where the observations are unordered. Klami and Jitta [KJ16] also proposed a class of priors for microclustering by placing constrained priors on the cluster sizes. While their approach allows for a great degree of control over the cluster sizes, inference is challenging, as the cluster assignments must be updated jointly in order to satisfy the combinatorial constraints.

2.4.4 Methods for scaling entity resolution

Scalability and computational efficiency are important considerations when designing entity resolution (ER) systems. Practical ER systems must be able to scale to data sources containing millions of records, while in some cases providing update-to-date results in near real-time [Pap+20]. Formal treatments of the ER problem have shown that finding a globally-optimal solution is NP-hard [CKM00; ZRG15]. Even finding a locally-optimal solution presents a challenge, as it is necessary to perform all-to-all pairwise comparisons between records, which scales quadratically in the number of records.

In order to improve scalability, many ER systems incorporate *blocking* to reduce the number of pairwise comparisons (see Section 2.2.2). Blocking methods efficiently filter out comparisons between pairs of records that are likely non-matches, leaving a much smaller set of candidate pairs to be compared in later stages. While blocking can significantly improve scalability and efficiency, it may increase the rate of false negative errors—matching pairs that are misclassified as non-matches. Effective blocking methods should ideally satisfy the following properties:

- *High recall.* All matching pairs should appear in the set of candidate pairs, to avoid false negative errors.
- *High reduction ratio.* The set of candidate pairs should be small in order to improve computational efficiency/scalability.

- *Linear complexity.* The algorithm should ideally scale linearly in the number of records.

There is a vast body of literature on blocking, which is also known as *indexing* and *filtering*. We refer the reader to Christen [Chr12b] and Papadakis et al. [Pap+20] for surveys and Papadakis et al. [Pap+16] for an empirical comparison of various methods. In the remainder of this section, we review some of the most common methods.

Traditional blocking. This method partitions the records into disjoint blocks and outputs comparisons between pairs of records that fall within the *same* block. Conventionally, the blocks are formed by partitioning the records according to their values for a selected attribute [FS69; New88]. While this approach can be effective, it is likely to perform poorly in terms of recall if the attribute selected for blocking is unreliable. For example, if person-related records are blocked on `year_of_birth`, any record pairs with a discrepancy on `year_of_birth` will not be considered for matching. In addition, there is a limited capacity to control block sizes, which are determined by the distribution of the selected attribute.

Blocking functions. Traditional blocking can be generalised by introducing *blocking functions*, which are responsible for assigning records to blocks [BKM06; Das+12]. Formally, a blocking function is a 1-1 mapping from a tuple of record attributes to a key, which uniquely identifies the assigned block. A blocking function may combine and transform information from multiple attributes, thereby providing greater flexibility compared to traditional blocking. For example, a blocking function designed for person-related records might output keys containing the first three digits of a `phone_number` attribute, concatenated with the first character of a `surname` attribute. There is a functional resemblance between blocking functions and locality-sensitive hashing (LSH) methods, which map similar inputs to the same hash values with high probability. Steorts et al. [Ste+14] proposed two blocking methods based on LSH. The methods were observed to perform similarly to manually-derived blocking functions, albeit with increased computational complexity. Like manually-specified blocking functions, the LSH methods required tuning to achieve good performance.

Multi-pass blocking. While blocking functions allow for control of the blocking assignments, there is still a risk of discarding matching pairs of records which are assigned to distinct blocks. This risk can be minimised by performing *multiple* blocking passes, using a different blocking function in each pass [Kel84; Jar89; HS95; Wha+09]. Various methods have been proposed to refine the candidate pairs produced in each pass. Hernández and Stolfo [HS95] recommended taking the union of the candidate pairs produced in all passes, before taking the transitive closure. Whang et al. [Wha+09] proposed an iterative approach, where the blocking passes are performed in succession, with the matching pairs propagating to subsequent passes. While both of these methods aim to improve coverage, Papadakis et al. [Pap+14] proposed a method called *meta-blocking* that leverages multiple passes to refine the set of candidate pairs. It exploits similarity information encapsulated in the multiple blocking passes to form a graph representation of the records, where edges between records are weighted according to the frequency of

occurrence in the blocking passes. The number of candidate pairs can then be reduced by pruning low-weight edges. The framework has been extended to a distributed setting to further improve scalability [Eft+17].

Automatic blocking. The effectiveness of blocking is often highly dependent on the choice of blocking functions and characteristics of the source data. In practice, blocking functions are typically selected manually based on trial and error or domain knowledge [Win05]. Several works have explored the idea of learning blocking functions automatically from labelled data [BKM06; MK06; Das+12]. Bilenko et al. [BKM06] proposed a framework to learn a class of blocking functions that can be expressed as disjunctions of blocking predicates. They proved that the problem of learning the optimal blocking function is NP-hard, however they proposed effective approximate algorithms. Das Sarma et al. [Das+12] extended the work of Bilenko et al. [BKM06] by considering a larger family of tree-structured blocking functions, while incorporating user-specified constraints on the block sizes, disjointness and coverage/efficiency trade-off.

Sorted neighbourhood method. Hernández and Stolfo [HS95] proposed an alternative to conventional blocking known as the *sorted neighbourhood method (SNM)*. It constructs blocks by passing a sliding window over sorted records. The sorting may be performed with respect to the raw attributes, or after applying transformations. If the transformations and sorting are designed so that similar records appear close together, then the resulting blocks are likely to contain a large fraction of the matching pairs. As with conventional blocking, multiple passes of SNM can be performed to improve coverage. An important parameter of SNM is the size of the sliding window, which can be used to trade-off recall and efficiency. Yan et al. [Yan+07] proposed a method for adapting the size of the sliding window based on the similarity of records within the window.

Canopy clustering. McCallum et al. [MNU00] proposed a generic method for improving the scalability of clustering algorithms called *canopy clustering*. It divides a data set into overlapping subsets called *canopies* using a distance measure that can efficiently answer range queries. Given such a distance measure and two distance thresholds $d_{\text{loose}} > d_{\text{tight}}$, a canopy is constructed as follows: (i) a record is randomly selected from the data set to serve as a canopy centre; (ii) records within distance d_{loose} from the canopy centre are copied into the canopy; (iii) records within distance d_{tight} from the canopy centre are removed from the data set. Steps (i)–(iii) are repeated until the data set is empty. The quality of the resulting canopies is dependent on the distance measure and distance thresholds, with smaller distance thresholds yielding smaller canopies at the potential cost of reduced recall. Typically, inverted-index based distance functions are used, such as the Jaccard index or TF-IDF cosine distance [BKM06] (see Section 2.3.2).

2.5 Crowdsourcing and entity resolution

While automated methods are quite capable for many ER applications, it is often necessary to acquire human-labelled ground truth for quality control purposes. An accessible and budget-friendly source of human input is provided by crowdsourcing platforms, which

employ workers to perform discrete on-demand tasks [Vau17]. We review some of this work below, as it is relevant to the online evaluation framework we develop for ER in Chapters 5 and 6.

2.5.1 Hybrid human-machine methods

Recent work has explored *hybrid human-machine methods* for ER, which combine predictions from automated ER algorithms with crowdsourcing as an error-correction layer [Wan+12; Wan+13; WMG13; VBD14; VG15; FSS16; VGP17; MS17; Gal+18].

Wang et al. [Wan+12] introduced one of the first hybrid human-machine methods for ER called `CrowdER`. It uses an automated method to compute similarity scores for pairs of records. Records that are sufficiently similar are then scheduled for verification by crowd workers. The authors prove that optimal task scheduling is NP-hard, however they claim that verification of pairs in decreasing order of similarity performs well in practice. Later, Vesdapunt et al. [VBD14] showed that the heuristic approach for scheduling proposed by Wang et al. [Wan+12] can be $\Omega(N)$ worse than optimal, where N is the number of records in the database. They analysed two alternative scheduling strategies: one where pairs are selected uniformly at random, and another where the pairs are prioritisation based on the expected number of duplicates. They find that both strategies are at most $O(K)$ worse than optimal where K is the expected number of entity clusters. Information theoretic lower bounds for three heuristic scheduling approaches proposed in [Wan+12] and [VBD14] were obtained by Mazumdar and Saha [MS17].

When scheduling tasks for workers, one can opt to show a *pair* of records in each task (as in [Wan+13; VBD14; VG15; FSS16]) or *multiple* records (as in [Wan+12; VGP17]). In the pairwise case, the worker answers whether a pair of records is coreferent or not. In the multi-record case, the worker groups the records into coreferent clusters. [VGP17] studied the optimal scheduling of tasks in pairwise and multi-record format. They found that ambiguous cases are best resolved through pairwise tasks, while obvious cases were more efficiently resolved through multi-record tasks.

2.5.2 Crowdsourcing for evaluation

Crowdsourcing is increasingly used to evaluate and debug models [Vau17]. This is especially true in unsupervised scenarios where there is no objective notion of ground truth. While crowdsourcing has not been applied to evaluate ER systems, there are many examples in other domains, such as topic modelling [Cha+09; NBB11], image segment classification [SG19], extreme classification [Sun+14], information retrieval systems [ARS08] and machine translation [Cal09]. Some authors have noted a high level of agreement between evaluation results obtained via crowdsourcing and those obtained via domain experts [Cal09; SG19]. However, Lease [Lea11] emphasises the importance of quality control when relying on crowdsourcing—e.g. due to worker biases, lack of effort or lack of domain knowledge.

2.6 Evaluation of entity resolution

The very circumstances that give rise to entity resolution (ER)—lack of unique identifiers, heterogeneity across data sources, and poor data quality—explain the crucial role of evaluation in ER workflows. Evaluation can be used for a variety of purposes, including parameter tuning, benchmarking, and providing quality guarantees prior to deployment. In an ER context, the target of evaluation is typically the predicted coreference relation $\hat{\mathcal{P}}$ (see Definition 2.1). Ideally, $\hat{\mathcal{P}}$ should be identical to the unknown true relation $\mathcal{P}$, however a perfect prediction is rarely achievable. Supervised evaluation methods estimate the extent to which the predicted relation $\hat{\mathcal{P}}$ deviates from the unknown true relation $\mathcal{P}$, based on a sample of the ground truth encoded in $\mathcal{P}$. A variety of performance measures can be used to quantify differences between $\hat{\mathcal{P}}$ and $\mathcal{P}$ [Bar15], which we review in Sections 2.6.1 and 2.6.2. We also review performance measures for evaluating blocking in isolation in Section 2.6.3.

2.6.1 Pairwise performance measures

In Section 2.2.1 we noted that ER is often formulated as a binary classification problem on the product space of record pairs $\mathcal{R} \times \mathcal{R}$. Under this formulation, it is common to use standard methods for evaluating binary classifiers. Suppose the binary labels *match*/*non-match* are encoded as 1/0 respectively. Let $X = \{x_1 = (r_{i_1}, r_{j_1}), \ldots, x_n = (r_{i_n}, r_{j_n})\}$ be a sample of record pairs from the product space $\mathcal{R} \times \mathcal{R}$ and let $\hat{y}(x_i) \in \{0, 1\}$ denote the predicted label for record pair $x_i \in \mathcal{R} \times \mathcal{R}$. Suppose true labels $Y = \{y_1, \ldots, y_n\}$ have been acquired for each record pair in X. We define the number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) as follows:

$$\begin{aligned} \text{TP} &= \sum_{i=1}^{n} \hat{y}(x_i) \cdot y_i, \quad \text{FP} = \sum_{i=1}^{n} \hat{y}(x_i) \cdot (1 - y_i), \\ \text{FN} &= \sum_{i=1}^{n} (1 - \hat{y}(x_i)) \cdot y_i, \quad \text{and} \quad \text{TN} = \sum_{i=1}^{n} (1 - \hat{y}(x_i)) \cdot (1 - y_i). \end{aligned} \tag{2.4}$$

TP and TN count record pairs that are correctly classified, while FP and FN count record pairs that are incorrectly classified (type I and II errors respectively).

In ER there is severe imbalance between matches and non-matches. As a result, TP typically scales linearly in $|\mathcal{R}|$, while TN scales quadratically in $|\mathcal{R}|$ (assuming low error rates). This means any ER system can achieve high accuracy by predicting that all record pairs are non-matches. It is therefore essential to select performance measures that are robust under class imbalance.

Pairwise precision and recall. Precision and recall are suitable for the pairwise formulation of ER, as they are insensitive to TN:

$$\text{pPr} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad \text{pRe} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

In words, pairwise precision pPr is the fraction of true matches among the predicted matches, and pairwise recall pRe is the fraction of predicted matches among the true

matches. Both measure take values on the unit interval (excluding pathological cases), where higher values indicate better performance. Often there is a trade-off to be made between precision and recall, depending on whether exactness (precision) or completeness (recall) of the matches is more important for a given application.

Pairwise F-measure. In some circumstances it is desirable to summarise the precision and recall in a single scalar measure. The weighted F-measure can be used for this purpose. It is defined as the weighted harmonic mean of the precision and recall:

$$F_\beta = (1 + \beta^2)\frac{\text{pPr} \cdot \text{pRe}}{\beta^2\text{pPr} + \text{pRe}}$$

where the weight $\beta \in [0, \infty)$ relates to the importance that the user attaches to recall over precision. When $\beta = 1$, precision and recall are weighted equally and the measure is known as the *balanced F-measure* or *F1-score*.

2.6.2 Clustering performance measures

The pairwise performance measures outlined in the previous section tend to focus on local agreements between the predicted and true relations, and do not penalise violations of transitivity. However, when transitivity is enforced, ER can be cast as a clustering problem and evaluated using clustering performance measures. Let $\mathcal{R}' = \{r_1, \ldots, r_n\}$ denote a sample of records from $\mathcal{R}$. When the predicted and true coreference relations $\hat{\mathcal{P}}$ and $\mathcal{P}$ are transitive, they induce clusterings of the records in $\mathcal{R}'$. Specifically, each cluster corresponds to an equivalence class under the relation. In the definitions below, we let P and $\hat{P}$ denote the *true* and *predicted* clusterings of $\mathcal{R}'$, respectively.

Cluster-level precision and recall. Cluster-level precision and recall are defined in terms of exact cluster matches [HEG06; Wel+04]:

$$\text{cPr} = \frac{|\hat{P} \cap P|}{|\hat{P}|} \quad \text{and} \quad \text{cRe} = \frac{|\hat{P} \cap P|}{|P|}.$$

In words, the cluster-level precision (cPr) is the fraction of true clusters that also appear in the predicted clusters, and the cluster-level recall (cRe) is the fraction of predicted clusters that also appear in the true clusters. This measure is likely to be too strict when the clusters are large, as a single error in a cluster will result in a mismatch. The cluster-level F-measure $\text{c}F_\beta$ is defined analogously to the pairwise F-measure, as the weighted harmonic mean of cPr and cRe.

Closest cluster-level precision and recall. A less strict alternative to the cluster-level precision and recall was used for evaluation by Benjelloun et al. [Ben+09] (see also [Bar15]). Rather than counting exact cluster matches between P and $\hat{P}$, it instead allows for fuzzy matches by determining the closest match based on the Jaccard similarity. Under this relaxed notion of cluster matches, the closest cluster-level precision (ccPr) and recall (ccRe) are defined as follows:

$$\text{ccPr} = \frac{\sum_{\hat{p} \in \hat{P}} \max_{p \in P} \text{sim}_{\text{Jac}}(\hat{p}, p)}{|\hat{P}|} \quad \text{and} \quad \text{ccRe} = \frac{\sum_{p \in P} \max_{\hat{p} \in \hat{P}} \text{sim}_{\text{Jac}}(\hat{p}, p)}{|P|}.$$

where $\text{sim}_{\text{Jac}}(\cdot,\cdot)$ is the Jaccard similarity as defined in (2.2).

Homogeneity, completeness and V-measure. Rosenberg and Hirschberg [RH07] proposed entropy-based clustering measures called *homogeneity* and *completeness*, which are somewhat analogous to precision and recall respectively. A predicted clustering satisfies *homogeneity* if all of the clusters contain data points (records) from the same class (related to the same entity). It satisfies *completeness* if all the data points (records) of a given class (related to a particular entity) are assigned to the same cluster. The two measures are defined as follows:

$$\text{Ho} = \begin{cases} 1 & \text{if } H(P,\hat{P}) = 0, \\ 1 - \frac{H(P|\hat{P})}{H(P)} & \text{otherwise.} \end{cases}$$

$$\text{Co} = \begin{cases} 1 & \text{if } H(\hat{P},P) = 0, \\ 1 - \frac{H(\hat{P}|P)}{H(\hat{P})} & \text{otherwise.} \end{cases}$$

where the entropy and conditional entropies are defined as:

$$H(P|\hat{P}) = -\frac{1}{n}\sum_{\hat{p}\in\hat{P}}\sum_{p\in P}|p\cap\hat{p}|\log\frac{|p\cap\hat{p}|}{\sum_{p\in P}|p\cap\hat{p}|}$$

$$H(P) = -\frac{1}{|P|}\sum_{p\in P}\sum_{\hat{p}\in\hat{P}}|p\cap\hat{p}|\log\frac{\sum_{\hat{p}\in\hat{P}}|p\cap\hat{p}|}{|P|}.$$

Both homogeneity (Ho) and completeness (Co) take on values in the unit interval, where higher values indicate better performance.

Rosenberg and Hirschberg also defined the *V-measure* as an analogue to F-measure:

$$V_\beta = (1+\beta^2)\frac{\text{Ho}\cdot\text{Co}}{\beta^2\cdot\text{Ho}+\text{Co}},$$

where the weight $\beta \in [0,\infty)$ relates to the importance that the user attaches to completeness over homogeneity. Becker [Bec11] proved that the balanced V-measure (with $\beta = 1$) is equivalent to the normalised mutual information when the arithmetic mean is used as the aggregation function. An advantage of homogeneity and completeness is that they allow assessment of different types of clustering errors, in a similar vein to precision and recall.

Adjusted Rand index. Cluster measures can also be defined in terms of pairwise agreements, similar to the pairwise measures listed in Section 2.6.1. Given a sample of records $\mathcal{R}'$ of size n, there are $\binom{n}{2}$ pairs of records to consider. Each pair may be classified as a *match* (assigned to the same cluster) or a *non-match* (assigned to distinct clusters). We can then define true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) as in Section 2.6.1, using the true clustering P as a reference.

The *Rand index (RI)* [Ran71] measures clustering similarity as the fraction of pairwise decisions on which the two clusterings agree:

$$\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}.$$

It is equivalent to classification accuracy and is difficult to interpret for ER applications owing to the high proportion of true negatives (TN). Hubert and Arabie [HA85] defined a corrected-for-chance version of the Rand index, known as the *adjusted Rand index (ARI)*:

$$\text{ARI} = \frac{2(\text{TN} \cdot \text{TP} - \text{FP} \cdot \text{FN})}{(\text{TN} + \text{FP})(\text{FP} + \text{TP}) + (\text{TN} + \text{FN})(\text{FN} + \text{TP})}.$$

It compares the observed agreements to expected agreements under a random model. The range of the ARI is $[-1, 1]$, where a value of 0 corresponds to a completely random prediction, a value of 1 corresponds to a perfect match, and a value of -1 correspond to predictions that are anti-correlated with the true clustering. Gates and Ahn [GA17] note that the choice of random model has a strong impact on the way similarity is measured. The ARI approaches the pairwise F1-score as $\text{TN} \longrightarrow \infty$.

Generalised merge distance. Menestrina et al. [MWG10] proposed a distance measure for clustering evaluation called *generalised merge distance (GMD)*, which is inspired by string edit distance measures. The GMD between clusterings $\hat{P}$ and P is the minimum legal path cost required to convert $\hat{P}$ to P using split and merge operations. The costs of split and merge operations are determined by user-defined operation-order-independence cost functions. For particular choices of the cost functions, the GMD reduces to other measures—e.g. pairwise precision, recall and F1-score. In general, the GMD may be difficult to compare across data sets, as the range varies depending on the size of the data set.

2.6.3 Performance measures for blocking

When a pipelined architecture is used to perform ER (see Section 2.2.2), it is common to evaluate steps in the pipeline separately in order to isolate potential issues. The blocking step has a strong bearing on the final output, as it acts as a filter on the product space of record pairs. If the filtering is too aggressive, matches may be missed in the final output. On the other hand, if the filtering is too lenient, there may be too many pairs to compare in the next stage, resulting in poor computational efficiency. The trade-off between these two factors is often measured in terms of the *pair completeness* and *reduction ratio* [EVE02].

Pair completeness. Let $\mathcal{X}$ denote the complete product space and let $\mathscr{C} \subset \mathcal{X}$ denote the set of candidate record pairs output in the blocking step. The *pair completeness (PC)* measures the quality of $\mathscr{C}$, ignoring considerations of computational efficiency:

$$\text{PC} = \frac{|\mathscr{C} \cap \mathscr{P}|}{|\mathscr{P}|},$$

where $\mathscr{P}$ is the unknown true relation. It is equivalent to recall, and takes on values in the unit interval, with larger values indicating better performance. In practice, one could estimate PC using a sample of record pairs as described in Section 2.6.1.

Reduction ratio. The set of candidate record pairs $\mathscr{C}$ can be assessed in terms of computational efficiency through the *reduction ratio (RR)*:

$$\mathrm{RR} = 1 - \frac{|\mathscr{C}|}{|\mathfrak{X}|}.$$

In words, RR is the relative reduction in the size of the product space. A larger value of RR corresponds to more aggressive filtering, and improved computational efficiency of ER. It is important to note that RR does not directly measure ER quality, although larger values of RR are typically associated with smaller values of PC. Since RR does not depend on the unknown true relation, it can be computed exactly without resorting to sampling.

Chapter 3

Scalable unsupervised Bayesian entity resolution

Bayesian models provide a natural framework for reasoning under uncertainty, and are therefore an appealing tool for solving entity resolution (ER) tasks. While various Bayesian ER models have been proposed in the literature, their use has been limited in practice due to poor scalability of inference. In this chapter, we propose methods for improving the scalability and statistical efficiency of inference for the `blink` ER model [Ste15]. Our solution, called distributed `blink` or `d-blink`, integrates probabilistic blocking, a distributed partially-collapsed Gibbs sampling algorithm, and fast algorithms for performing Gibbs updates. Empirical studies on six data sets—including an application to U.S. Census and administrative data—demonstrates the vastly improved efficiency of `d-blink` compared to existing approaches.

3.1 Introduction

A recent development in entity resolution methodology has been the application of Bayesian models [Pas+02; BG06; TL11; FLS15; Ste15; SHF16; Zan+16]. In an ER context, a Bayesian model typically assumes that records in a database arise as references to latent (unobserved) entities. Prior knowledge and assumptions about the data-generating process can be encoded in the model, thereby improving robustness when labelled training data is scarce or unavailable. This is in contrast to generic models from statistics and machine learning (e.g. deep neural networks), which may require significant training data to achieve competitive ER accuracy [Ste15]. A further advantage of the Bayesian paradigm, is its ability to naturally account for uncertainty. This is important as there is a considerable degree of uncertainty in many ER applications, and accounting for uncertainty may substantially improve the accuracy and rigour of post-ER tasks [Wic+08; KBS18].

This chapter incorporates material from the following publication:

N. G. Marchant, A. Kaplan, D. N. Elazar, B. I. P. Rubinstein and R. C. Steorts. "d-blink: Distributed End-to-End Bayesian Entity Resolution". In: *Journal of Computational and Graphical Statistics* (2021). DOI: 10.1080/10618600.2020.1825451.

Approval was obtained from the U.S. Census Bureau Disclosure Review Board to publish results in Section 3.8 (DRB#: CBDRB-FY20-309).

Despite the benefits of Bayesian models for ER, current approaches are limited in practice due to poor scalability. Most approaches rely on Markov chain Monte Carlo (MCMC) to infer the unknown entity references (also known as the *linkage structure* or *coreference structure*). The computation time for a single step of the Markov chain is generally dominated by all-to-all comparisons between the records (or records and entities). This is compounded by the fact that a large number of steps may be required before the chain converges. Some prior work avoids the issue of scalability, by focusing on applications to small data sets with only several hundred records [Ste15; Zan+16; Sad17]. Others manage scalability in an ad-hoc manner, by applying conventional blocking [Chr12a] as a pre-processing step [For+01; Lar05; Lar12; TL11; GAZ13; Sad14; SHF16]. For instance, Steorts et al. [SHF16] partition the records into disjoint blocks and fit the ER model independently on each block. While this can improve scalability, it may severely compromise the accuracy of the posterior. Since the blocks are fixed a priori, records are forbidden from referring to the same entity if they reside in different blocks—the model has no ability to recover from a poor blocking design. Moreover, when the model is fit independently on each block, the model parameters are effectively replaced by block-level approximations.

In this chapter, we advocate a principled approach to scaling Bayesian ER models, which does not suffer from the aforementioned limitations. Our approach integrates auxiliary blocks into the model, so that blocking is performed automatically during MCMC. By doing this in a careful way, we are able to ensure that the marginal posterior is preserved, so that the inferred model parameters are asymptotically independent of the chosen blocking design. We focus specifically on scaling the `blink` ER model [Ste15], as it supports ER of multiple structured databases, while other models are specialised to one or two databases. In addition to integrating blocking, we propose several ideas aimed at improving scalability:

(i) we distribute/parallelise inference at the block level;

(ii) we propose a blocking function based on k-d trees which achieves proper load balancing;

(iii) we design a partially-collapsed Gibbs sampler with improved mixing properties;

(iv) we propose a sub-quadratic algorithm for updating entity assignments which leverages indexing; and

(v) we propose a novel algorithm for efficiently updating entity attributes based on perturbation sampling.

Our scalable extension of `blink`, which incorporates all of these ideas, is called "distributed `blink`" or `d-blink` for short.

We implement `d-blink` as an Apache Spark [Zah+16] package and conduct an empirical evaluation using five data sets. Where `blink` fails to scale beyond a few thousand records, we find that `d-blink` readily scales to several hundred thousand records in a distributed setting. To illustrate the effectiveness of our approach for realistic ER tasks, we present a case study using Census and administrative data from the U.S. state of Wyoming.

Chapter outline. We review related work in Section 3.2. Section 3.3 formulates ER in a Bayesian setting, and presents the `d-blink` model with integrated blocking. We provide guidelines for selecting blocking functions in Section 3.4. We then discuss inference and propose a distributed partially-collapsed Gibbs sampler in Section 3.5. We suggest additional methods for improving computational efficiency of inference in Section 3.6. Section 3.7 presents a comprehensive empirical evaluation, and Section 3.8 presents a case study to U.S. Census and administrative data. We make closing remarks in Section 3.9.

3.2 Related work

The related work most closely connected to this chapter spans three key areas: probabilistic ER methods, inference for Bayesian ER models, and distributed Markov chain Monte Carlo (MCMC). We refer the reader to Section 2.4 for a comprehensive review of other ER methods.

Probabilistic ER methods. The first probabilistic approach to ER was due to Newcombe et al. [New+59], who applied matching rules to pairs of records. This idea was later formalised in a seminal paper by Fellegi and Sunter [FS69] within a decision-theoretic framework. Many variations of the Fellegi-Sunter (FS) approach have been proposed (for surveys, see [Win06; Win14]), including a generalisation to multiple databases [SF13]. Others have addressed scalability of FS-type approaches using blocking/indexing methods (see [Chr12c; Ste+14] for surveys) and efficient data structures [EFI19]. However, traditional FS approaches do not naturally support propagation of ER uncertainty, and existing methods for scaling make approximations that sacrifice accuracy.

While the FS approach has been highly influential, it has also been criticised due to its lack of support for duplicates within databases; misspecified independence assumptions; and its dependence on subjective thresholds [TL11]. These limitations have prompted development of more sophisticated Bayesian models, including models for bipartite matching [For+01; Lar05; Lar12; TL11; GAZ13; Sad17; MSM19], deduplication [Sad14; TSL20] and matching across multiple databases [Ste15; SHF16]. Several of these models operate on attribute-level comparisons between pairs of records in a similar vein as the FS approach [Lar05; Lar12; GAZ13; Sad14; Sad17; MSM19]. This contrasts with entity-centric generative models which assume the records arise as distortions to some latent entity attributes [TL11; Ste15; SHF16; TSL20].

In scenarios where training data is scarce or unavailable, Bayesian generative models tend to be more robust than discriminative or likelihood-based methods, as the priors have a regularising effect. Bayesian generative models are also amenable to theoretical analysis: recent work has obtained lower bounds on the probability of misclassifying the entity associated with a record [SBN17]. However, a major downside of Bayesian ER models is the computational cost of performing inference, as we discuss next.

Inference for Bayesian ER models. Most prior work on Bayesian generative models for ER [e.g. TL11; GAZ13; Ste15] has relied on Gibbs sampling for inference. Compared to other Markov chain Monte Carlo (MCMC) algorithms, Gibbs sampling is relatively easy to implement, however it may suffer from slow convergence and poor mixing owing

to its highly local moves [Liu04]. Scalability is also a challenge, as a naïve Gibbs update for the linkage structure requires all-to-all comparisons between records (or between records and entities for entity-centric models). This issue is often managed by applying deterministic blocking prior to Gibbs sampling, thereby sacrificing accuracy and proper treatment of uncertainty [Lar05; Lar12; TL11; GAZ13; Sad14].

In the broader context of clustering models, the *split-merge algorithm* [JN04] has been proposed as an alternative to Gibbs sampling. It is a Metropolis-Hastings algorithm, which traverses the space of clusterings via proposals that split individual clusters or merge pairs of clusters. Since multiple cluster items are updated in a single move, it is less susceptible to becoming trapped in local modes. Steorts et al. [SHF16] applied this algorithm, in combination with deterministic blocking, to update the linkage structure in an ER model similar to `blink`. A close relative of the split-merge algorithm is the *chaperones algorithm*, which was proposed for inference in microclustering models [Zan+16]. The chaperones algorithm is expected to be more efficient, as it preferentially focuses on more likely cluster reassignments, through a user-specified biased distribution on the product space of cluster items. However, the biased distribution must be designed so that random item pairs can be drawn efficiently, without explicitly constructing the product space.

More recently, Zanella [Zan20] proposed a general framework for designing informative proposals in a Metropolis-Hastings setting, which is suited for discrete spaces (e.g. the space of possible linkage structures). They show that *locally-balanced proposals* are asymptotically-optimal within the class of pointwise informative proposals, and demonstrate significant improvements in efficiency when compared to a split-merge-type algorithm. However, computing a locally-balanced proposal for the linkage structure is computationally challenging due to quadratic scaling. This can be mitigated to some extent by running locally-balanced updates within randomly-selected sub-blocks of records. However to avoid poor mixing, care must be taken to ensure that randomly-selected sub-blocks contain likely matching records.

In contrast to much of the literature on Bayesian ER models, McVeigh et al. [MSM19] proposed a method that combines deterministic blocking and restricted MCMC (based on earlier work by [MM17]). They balance approximation error by performing coarse-grained deterministic blocking/indexing as an initial step, followed by data-dependent post-hoc blocking. During inference, the linkage structure is updated using locally-balanced proposals, restricted to the post-hoc blocks. They demonstrate improved scalability—to data sets with several hundred thousand of record—with minimal risk of approximation error. However, their approach is not directly compatible with distributed inference (see below) and may require modification for use with an entity-centric model.

Parallel/distributed MCMC. Recent literature has focused on using parallel and distributed computing to scale up MCMC algorithms, where applications have included Bayesian topic models [New+09; SN10; ASW14] and mixture models [WDX13; CF13; Lov+13; Ge+15]. We review the application to mixture models, as they are conceptually similar to ER models.

Existing work has concentrated on Dirichlet process (DP) mixture models and hierarchical DP mixture models. The key to enabling distributed inference for these models is the realisation that a DP mixture model can be reparameterised as a mixture of DPs. Put simply, the reparameterised model induces a *partitioning* of the clusters into blocks, such

that clusters assigned to *distinct blocks* are conditionally independent. As a result, variables within blocks can be updated in parallel. Williamson et al. [WDX13] exploited this idea at the thread level to parallelise inference for a DP mixture model. Chang and Fisher [CF13] followed a similar approach, but included an additional level of parallelisation within blocks using a parallelised version of the split-merge algorithm. Others [Lov+13; Ge+15] have developed distributed implementations in the MapReduce framework.

A disadvantage of the aforementioned approaches is the need for synchronisation and potentially significant data transfer between compute nodes at the end of each MCMC step. Recent work has looked to reduce communication between nodes by running local simulations in parallel on potentially overlapping blocks, then merging the results [Zua+19; Ni+20; SWD20]. However these approaches do not come with strong guarantees on the quality of the approximation to the posterior clustering. In particular, these methods optimize a local loss function at one or more steps, which does not guarantee a good global solution and inhibits uncertainty propagation. The method by Song et al. [SWD20] attempts to minimize this effect by only optimizing local losses in one part of their algorithm.

Since the `blink` model does not rely on a DP or HDP prior for the linkage structure, we cannot directly apply existing approaches for distributed/parallel MCMC.[1] However we do borrow the reparameterisation idea, albeit with a more flexible partition specification which permits similar entities to be co-blocked, while facilitating load balancing. It would be interesting to see whether similar ideas can be applied to microclustering models, which are thought to be well-suited for entity resolution [Zan+16].

3.3 A scalable model for Bayesian ER

We now present our extension to the `blink` model for Bayesian ER [Ste15], which incorporates auxiliary blocks, support for missing values, and generic attribute similarity functions. We describe notation and assumptions in Section 3.3.1, before outlining the generative process and posterior distribution in Sections 3.3.2 and 3.3.3. Attribute similarity measures are defined in Section 3.3.4, including a truncation approximation which improves scalability. In Section 3.3.5, we demonstrate that the marginal posterior of `d-blink` is equivalent to `blink` under certain conditions.

3.3.1 Notation and assumptions

We consider entity resolution of structured data from one or more data sources. Table 3.1 summarises our notation, including model-specific parameters which will be introduced in Section 3.3.2. Let $s \in \{1, \dots, S\}$ be an index over sources and $i \in \{1, \dots, N\}$ be an index over records, which is unique across all sources. The source of the i-th record is denoted by $s_i \in \{1, \dots, S\}$ and the record's attribute values are represented as a tuple $\mathbf{x}_i = (x_{i1}, \dots, x_{iA})$ indexed by $a \in \{1, \dots, A\}$. We assume $x_{ia} \in \mathscr{D}_a$ for all i and a, where the domain $\mathscr{D}_a$ of the a-th attribute is a finite set of strings. We allow for the fact that some

[1] A DP prior is thought to be ill-suited for entity resolution because it assumes the number of entity "clusters" grows logarithmically in the number of records, while empirical observations call for near-linear growth [Zan+16].

Table 3.1: Summary of notation.

Notation	Description
$s \in 1 \dots S$	index over sources
$i \in 1 \dots N$	index over records
$e \in 1 \dots E$	index over entities
$a \in 1 \dots A$	index over attributes
$b \in 1 \dots B$	index over blocks
$\mathscr{D}_a$	domain of attribute a
$v \in 1 \dots \lvert\mathscr{D}_a\rvert$	index over domain of attribute a
$\mathbf{x}_i = (x_{i1}, \dots, x_{iA})$	attribute values for record i
$\mathbf{z}_i = (z_{i1}, \dots, z_{iA})$	distortion indicators for record i
$\mathbf{o}_a = (o_{i1}, \dots, o_{iA})$	observation indicators for record i
$\mathbf{y}_e = (y_{e1}, \dots, y_{eA})$	attribute values for for entity e
γ_i	assigned block for record i
λ_i	assigned entity for record i
`PartFn`$(\cdot)$	block assignment function
$\mathscr{R}_e$	set of records assigned to entity e
$\mathscr{E}_b$	set of entities assigned to block b
θ_{sa}	distortion probability for attribute a in source s
$\beta_{sa}^{(0)}, \beta_{sa}^{(1)}$	hyperparameters for prior on θ_{sa}
η_{sa}	observation probability for attribute a in source s
$\phi_a(\cdot)$	distribution over domain of attribute a
$\mathrm{sim}_a(\cdot, \cdot)$	similarity measure for attribute a

attributes x_{ia} may be missing completely at random through a corresponding indicator variable [LR02, p. 12]:

$$o_{ia} = \begin{cases} 1, & \text{if } x_{ia} \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases}$$

For compactness of notation, we refer to the set of all index combinations for a variable using a boldface capital, e.g. $\mathbf{S} = \{s_i\}_{i=1\dots N}$ and $\mathbf{X} = \{x_{ia}\}_{i=1\dots N; a=1\dots A}$. We also define notation to separate the record attributes $\mathbf{X}$ into an observed part $\mathbf{X}^{(o)}$ (those x_{ia}'s for which $o_{ia} = 1$) and a missing part $\mathbf{X}^{(m)}$ (those x_{ia}'s for which $o_{ia} = 0$).

We assume that there exists a finite population of entities, indexed by $e \in \{1, \dots, E\}$, which are represented in the records. Each entity e is described by a tuple of attribute values $\mathbf{y}_e = (y_{e1}, \dots, y_{eA})$, which may appear distorted in the records. The entity represented in the i-th record is denoted by $\lambda_i \in \{1, \dots, E\}$. The complete set of entity references $\mathbf{\Lambda} = \{\lambda_1, \dots, \lambda_N\}$ is referred to as the *linkage structure*. If the entities and records are viewed as independent vertex sets of a bipartite graph, $\mathbf{\Lambda}$ denotes the links (undirected edges) between entities and records. We place no constraints on the links, apart from the fact that each record must be linked to exactly one entity—i.e. all record vertices in the graph have degree 1. In particular, we permit duplicate records within sources and allow for arbitrary links across sources.

For computational convenience, we assume the entities are partitioned into B blocks indexed by $b \in \{1, \dots, B\}$. The partition is determined by a user-specified blocking function `PartFn` $: \bigotimes_{a=1}^{A} \mathscr{D}_a \to \{1, \dots, B\}$, which maps entities to blocks according to their attribute values. We also assume that records are assigned to blocks—we let $\gamma_i \in \{1, \dots, B\}$ denote the assigned block for the i-th record.

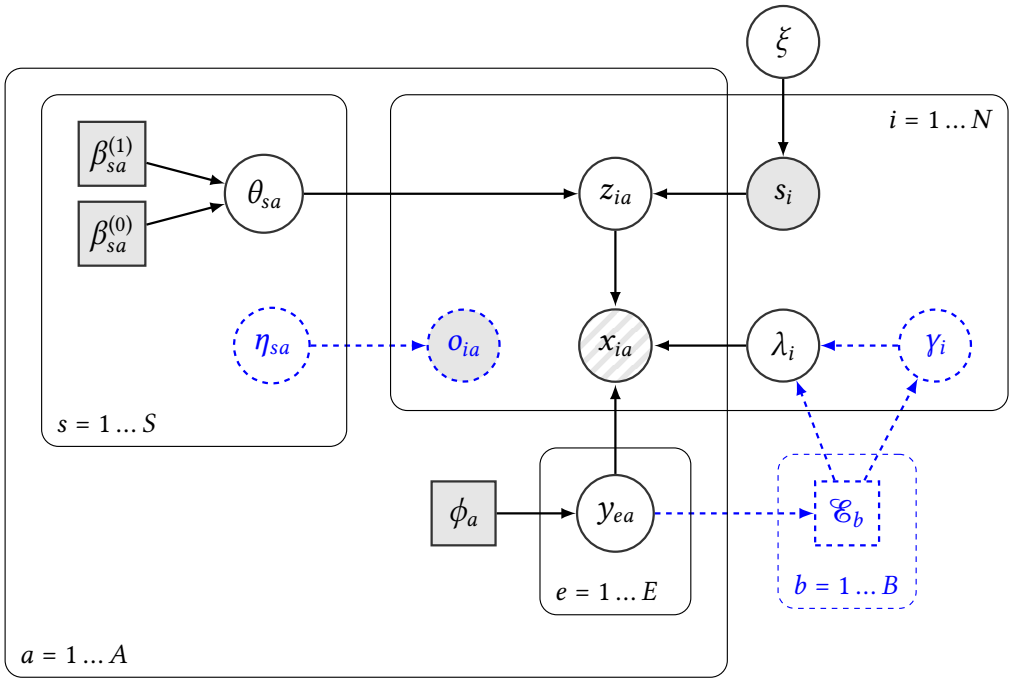


Figure 3.1: Plate diagram for `d-blink`. Extensions to `blink` are highlighted in a dashed blue line style. Circular nodes represent random variables; square nodes represent deterministic variables; (un)shaded nodes represent (un)observed variables; arrows represent conditional dependence; and plates represent replication over an index.

After specifying a generative model (see next section), we perform ER by inferring the *joint* posterior distribution over:

- the block assignments $\mathbf{\Gamma} = \{\gamma_i\}_{i=1\ldots N}$,
- the linkage structure $\mathbf{\Lambda} = \{\lambda_i\}_{i=1\ldots N}$, and
- the true entity attribute values $\mathbf{Y} = \{y_{ea}\}_{e=1\ldots E;a=1\ldots A}$,

conditional on the observed record attribute values $\mathbf{X}^{(o)}$ and sources $\mathbf{S}$. Note that we operate in a fully unsupervised setting, since we do not condition on ground truth for the links or entities. Inferring $\mathbf{\Gamma}$ is equivalent to the *blocking* stage of ER, where the records are partitioning into blocks to limit the comparison space. Inferring $\mathbf{\Lambda}$ is equivalent to the *matching/linking* stage of ER, where records that refer to the same entities are identified. Inferring $\mathbf{Y}$ is equivalent to the *merging* stage, where linked records are combined to produce a single representative record. By inferring $\mathbf{\Gamma}$, $\mathbf{\Lambda}$ and $\mathbf{Y}$ jointly, we are able to propagate uncertainty between the three stages.

3.3.2 Model specification

We now describe the generative process for `d-blink`. We provide a visual representation of the model in Figure 3.1, with key differences from `blink` highlighted in a dashed blue line style.

Entity model. Each entity e in the population is associated with a tuple of "true" attribute values $\mathbf{y}_e = (y_{e1}, \ldots, y_{eA})$. The value of the a-th attribute y_{ea} is assumed to be drawn independently from a distribution ϕ_a over the attribute domain $\mathcal{D}_a$:

$$y_{ea} \overset{\text{ind.}}{\sim} \text{Discrete}[\phi_a].$$

Following the `blink` model, we set the entity population size E and the distributions over the attribute domains ϕ_a empirically. Recommendations for setting these parameters are provided in Section 3.7.2.

Auxiliary blocks. The parameter space associated with the entities $\bigotimes_{a=1}^{A} \mathcal{D}_a$ is partitioned into B disjoint blocks. The partition is parameterised through a deterministic *blocking function*:

$$\texttt{PartFn} : \bigotimes_{a} \mathcal{D}_a \longrightarrow \{1, \ldots, B\}, \tag{3.1}$$

which is a free parameter and may be selected for inferential convenience. We motivate the auxiliary blocks in Section 3.4.1 and provide recommendations for selecting the blocking function in Section 3.4.

We shall often need to refer to the entities assigned to a particular block. To do this concisely, we introduce the notation $\mathcal{E}_b(\mathbf{Y}) = \{e : \texttt{PartFn}(\mathbf{y}_e) = b\}$ to denote the set of entities assigned to block b. It is important to note that this set is random due to the dependence on $\mathbf{Y}$, however we shall often omit the dependence for brevity.

Linkage model. Following `blink`, we assume records are instantiated by selecting an entity from the population uniformly at random. In order to achieve this behaviour while incorporating auxiliary blocks, we assume each record i is first assigned to a block γ_i with probability proportional to the block sizes:

$$\gamma_i | \mathbf{Y} \overset{\text{ind.}}{\sim} \text{Discrete}_{b \in \{1 \ldots B\}}[|\mathcal{E}_b|/E].$$

Then, an entity λ_i is selected uniformly at random from the assigned block:

$$\lambda_i | \gamma_i, \mathbf{Y} \overset{\text{ind.}}{\sim} \text{DiscreteUniform}[\mathcal{E}_{\gamma_i}].$$

Source model. Once a record is instantiated, it must be associated with one of the data sources. We assume the source for the i-th record is drawn independently from a distribution ξ:

$$s_i | \xi \overset{\text{iid.}}{\sim} \text{Discrete}[\xi].$$

There is no need to specify ξ in practice, since it has no bearing on inference. This is because the s_i's are fully observed and conditionally independent of the other model parameters.

Distortion model. We assume the record attribute values are generated by copying the attribute values from the linked entity, subject to distortion. Following `blink`, we introduce a distortion probability θ_{sa} associated with each source s and attribute a. We assume

$$\theta_{sa} | \beta_{sa}^{(0)}, \beta_{sa}^{(1)} \overset{\text{ind.}}{\sim} \text{Beta}[\beta_{sa}^{(0)}, \beta_{sa}^{(1)}],$$

where $\beta_{sa}^{(0)}$ and $\beta_{sa}^{(1)}$ are hyperparameters. We provide recommendations for setting these hyperparameters in Section 3.7.2. The distortion probabilities feed into the generative process for the attribute values of the i-th record, as outlined below.

(i) For each attribute a, draw a distortion indicator z_{ia}:

$$z_{ia}|\theta_{s_i a}, s_i \overset{\text{ind.}}{\sim} \text{Bernoulli}[\theta_{s_i a}].$$

(ii) For each attribute a, draw a value x_{ia}:

$$x_{ia}|z_{ia}, y_{\lambda_i a} \overset{\text{ind.}}{\sim} (1 - z_{ia})\delta(y_{\lambda_i a}) + z_{ia}\,\text{Discrete}_{v\in\mathcal{D}_a}[\psi_a(v|y_{\lambda_i a})].$$

where $\delta(y)$ represents a point mass at y. If $z_{ia} = 0$, x_{ia} is copied directly from the entity. Otherwise, x_{ia} is drawn from the domain $\mathcal{D}_a$ according to the distortion distribution ψ_a. In the literature, this is known as a hit-miss model [CH90].

(iii) For each attribute a, draw an observation indicator o_{ia}:

$$o_{ia} \overset{\text{ind.}}{\sim} \text{Bernoulli}[\eta_{sa}].$$

If $o_{ia} = 1$, x_{ia} is observed, otherwise it is missing. There is no need to specify η_{sa} since the o_{ia}'s are fully observed and conditionally independent of the other model parameters. Note that this is equivalent to assuming record attribute values are *missing completely at random*. This is likely an overly simplistic assumption for real ER data sets, however it allows us to incorporate missingness with a minimal cost to model complexity.

Distortion distribution. The distribution $\psi_a(\cdot|w)$ chooses a distorted value for attribute a conditional on the true value w. In our parameterisation of the model, it is defined as

$$\psi_a(v|w) = h_a(w)\phi_a(v)\text{e}^{\text{sim}_a(v,w)}, \tag{3.2}$$

where $h_a(w) = 1/\sum_{v\in\mathcal{D}_a}\phi_a(v)\text{e}^{\text{sim}_a(v,w)}$ is a normalisation constant and sim_a is the similarity measure for attribute a (see Section 3.3.4). Intuitively, this distribution chooses values in proportion to their empirical frequency, while placing more weight on those that are "similar" to w. This reflects the notion that distorted values are likely to be close to the truth, as is the case when modelling typographical errors.

3.3.3 Posterior distribution

By reading the conditional dependence structure off the plate diagram (Figure 3.1), we obtain the following expression for the posterior distribution over the model parameters:

$$p(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{Y}, \mathbf{Z}, \mathbf{\Theta}, \mathbf{X}^{(m)}|\mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto \prod_{e,a} p(y_{ea}|\phi_a) \times \prod_{s,a} p(\theta_{sa}|\beta_{sa}^{(0)}, \beta_{sa}^{(1)})$$
$$\times \prod_i \left\{ p(\gamma_i|\mathbf{Y})p(\lambda_i|\gamma_i, \mathbf{Y}) \prod_a p(z_{ia}|\theta_{s_i a}) \right\} \times \prod_{i,a} p(x_{ia}|z_{ia}, y_{\lambda_i a}).$$

Ideally, we'd like to marginalise out all variables that are not of interest (e.g. $\boldsymbol{\Theta}$ and $\mathbf{Z}$), however this is not tractable analytically. Fortunately, we can marginalise out the missing record attributes $\mathbf{X}^{(m)}$ which yields the following:

$$\begin{aligned} p(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Theta} | \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto & \prod_{e,a} p(y_{ea} | \phi_a) \times \prod_{s,a} p(\theta_{sa} | \beta_{sa}^{(0)}, \beta_{sa}^{(1)}) \\ & \times \prod_i \left\{ p(\gamma_i | \mathbf{Y}) p(\lambda_i | \gamma_i, \mathbf{Y}) \prod_a p(z_{ia} | \theta_{s_i a}) \right\} \times \prod_{\substack{i,a \\ o_{ia}=1}} p(x_{ia} | z_{ia}, y_{\lambda_i a}). \end{aligned} \tag{3.3}$$

We can further expand this expression by substituting the conditional distributions given in Section 3.3.2:

$$\begin{aligned} p(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Theta} | \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto & \prod_{e,a} \phi_a(y_{ea}) \times \prod_{s,a} \theta_{sa}^{\beta_{sa}^{(0)}-1} (1-\theta_{sa})^{\beta_{sa}^{(1)}-1} \times \prod_i \mathbb{I}\left[\lambda_i \in \mathscr{C}_{\gamma_i}(\mathbf{Y})\right] \\ & \times \prod_{i,a} \theta_{s_i a}^{z_{ia}} (1-\theta_{s_i a})^{1-z_{ia}} \times \prod_{\substack{i,a \\ o_{ia}=1}} \left\{ (1-z_{ia}) \mathbb{I}[x_{ia} = y_{\lambda_i a}] + z_{ia}\, \psi_a(x_{ia} | y_{\lambda_i a}) \right\}. \end{aligned} \tag{3.4}$$

3.3.4 Attribute similarity measures

We now discuss the attribute similarity measures that appear in the distortion distribution of (3.2). The purpose of these measures is to quantify the propensity that some value v in the attribute domain is chosen as a distorted alternative to the true value w.

Definition 3.1 (Attribute similarity measure). *Let $\mathscr{D}$ be the domain of an attribute. An* attribute similarity measure *on $\mathscr{D}$ is a function* $\mathrm{sim} : \mathscr{D} \times \mathscr{D} \longrightarrow [0, s_{\max}]$ *that satisfies* $0 \leq s_{\max} < \infty$ *and* $\mathrm{sim}(v, w) = \mathrm{sim}(w, v)$ *for all* $v, w \in \mathscr{D}$.

Note that this parameterisation in terms of attribute *similarity* measures differs from `blink`, which uses *distance* measures. By changing parameterisation, we are able to make use of a more efficient sampling method, as described in Section 3.6.3. The next proposition states that the two parameterisations are in fact equivalent, so long as the distance measure is bounded and symmetric.

Proposition 3.2. *Let* $\mathrm{dist}_a : \mathscr{D} \times \mathscr{D} \longrightarrow [0, d_{\max;a}]$ *be the attribute distance measure that appears in* `blink`, *and assume that* $0 \leq d_{\max;a} < \infty$ *and* $\mathrm{dist}_a(v, w) = \mathrm{dist}_a(w, v)$ *for all* $v, w \in \mathscr{D}$. *Define the corresponding attribute similarity measure for* `d-blink` *as*

$$\mathrm{sim}_a(v, w) := d_{\max;a} - \mathrm{dist}_a(v, w). \tag{3.5}$$

Then the parameterisation of ψ_a used in `d-blink` *is equivalent to* `blink`.

Proof. It is straightforward to show that sim as defined in (3.5) satisfies the requirements of Definition 3.1. All that remains is to show that the two parameterisations of the distortion distribution ψ_a are equivalent. Beginning with ψ_a as parameterised in `blink`, we substitute (3.5) and observe that

$$\psi_a(v | w) \propto \phi_a(v) \mathrm{e}^{-\mathrm{dist}_a(v,w)} = \phi_a(v) \mathrm{e}^{d_{\max;a} + \mathrm{sim}_a(v,w)} \propto \phi_a(v) \mathrm{e}^{\mathrm{sim}_a(v,w)}.$$

This is identical to our parameterisation in (3.2). □

In this chapter, we restrict our attention to the following similarity measures for simplicity:

- *Constant similarity measure.* This measure is appropriate for categorical attributes when there is no reason to believe one value is more likely than any other as a distortion to the true value w. Without loss of generality, it may be defined as $\text{sim}_{\text{const}}(v, w) = s_{\max}$ for all $v, w \in \mathcal{D}$.
- *Normalised Levenshtein similarity measure.* This measure is appropriate for modelling character-level distortions, such as character insertions, character deletions or character substitutions. It is based on a normalised variant of the Levenshtein (edit) distance, proposed by Yujian and Bo [YB07], as defined in (2.1).

Ideally, one should select attribute similarity measures according to the data at hand. Section 2.3 reviews some commonly-used measures and provides recommendations depending on the types of expected corruptions.

3.3.5 Model equivalence

We have purposely designed `d-blink` so that it reduces to `blink` under certain conditions. When the record attributes are fully observed, the posterior distribution of `d-blink` as specified in (3.4) is similar to `blink`. The difference lies in the factors involving the block assignments γ_i and the entity assignments λ_i. However, if one marginalises out the auxiliary block assignments—as is done automatically in Markov chain Monte Carlo—the posterior distributions are identical. This statement is made precise below.

Proposition 3.3. *Suppose the conditions of Proposition 3.2 hold and that $\beta_{sa}^{(0)} = \beta^{(0)}$ and $\beta_{sa}^{(1)} = \beta^{(1)}$ for all sources s and attributes a. Assume furthermore that all record attributes are observed, i.e. $o_{ia} = 1$ for all i, a. Then the marginal posterior of $\boldsymbol{\Lambda}$, $\mathbf{Y}$, $\mathbf{Z}$ and $\boldsymbol{\Theta}$ for* `d-blink` *(i.e. marginalised over $\boldsymbol{\Gamma} = [\gamma_i]_{i=1\ldots N}$) is identical to the posterior for* `blink`.

Proof. Under the stated conditions, the only factor in the posterior (3.3) that differs from `blink` is:

$$\prod_i p(\lambda_i|\gamma_i, \mathbf{Y})p(\gamma_i|\mathbf{Y}). \tag{3.6}$$

Substituting the density for the conditional distributions for a single i factor yields:

$$p(\lambda_i|\gamma_i, \mathbf{Y})p(\gamma_i|\mathbf{Y}) = \frac{\mathbb{I}\left[\lambda_i \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right]}{|\mathscr{E}_{\gamma_i}(\mathbf{Y})|} \times \frac{|\mathscr{E}_{\gamma_i}(\mathbf{Y})|}{E} = \frac{1}{E}\mathbb{I}\left[\lambda_i \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right].$$

Putting this in (3.6) and marginalising over $\boldsymbol{\Gamma}$ we obtain:

$$\prod_i \sum_{\gamma_i=1}^{B} p(\lambda_i|\gamma_i, \mathbf{Y})p(\gamma_i|\mathbf{Y}) = \prod_i \frac{1}{E} \sum_{\gamma_i=1}^{B} \mathbb{I}\left[\lambda_i \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right] = \prod_i \frac{1}{E}\mathbb{I}[\lambda_i \in \{1, \ldots, E\}],$$

which is the factor that appears in the posterior for `blink`. □

This is an important result, as it shows our inferences for the parameters of interest ($\boldsymbol{\Lambda}$ and $\mathbf{Y}$) are the same as we would obtain from `blink`. Thus we are able to apply blocking to scale the model, without compromising the correctness of the posterior distribution.

3.4 Blocking functions

In Section 3.3.2 we introduced a blocking function (Equation 3.1) that is responsible for assigning entities to blocks. This function may be regarded as a free parameter, since it has no bearing on model equivalence according to Proposition 3.3. However, from a practical perspective the blocking function ought to be chosen carefully, as it can impact inferential efficiency—both in terms of computational and mixing time. We suggest some guidelines for choosing a blocking function in Section 3.4.1, before presenting an example based on k-d trees in Section 3.4.2.

3.4.1 Interpretation and guidelines

Recall that the blocking function assigns an entity to a block according to its attributes $\mathbf{y}_e = [y_{ea}]_{a=1\ldots A}$. Since $\mathbf{y}_e$ is *unobserved*, it must be treated as a random variable over the space of possible attributes $\mathscr{D}_\otimes := \bigotimes_{a=1}^{A} \mathscr{D}_a$. This means the blocking function should not be interpreted as partitioning the entities directly. Rather, it should be interpreted as partitioning the space $\mathscr{D}_\otimes$ in which the entities reside, while taking the distribution over $\mathscr{D}_\otimes$ into account. With this interpretation in mind, we argue that the blocking function should ideally satisfy the following properties:

(i) *Balanced weight.* The blocks should have equal weight (probability mass) under the distribution over $\mathscr{D}_\otimes$, thereby ensuring the entities are distributed evenly (in expectation) among the blocks. This is a desirable property, as it ensures proper load balancing for our distributed inference algorithm (see Section 3.5.2).

(ii) *Entity separation.* A pair of entities drawn at random from the same block should have a high degree of similarity, while entities drawn from different blocks should have a low degree of similarity. This improves the likelihood that similar records will end up in the same block, and allows them to more readily form likely entities.

These properties need not be satisfied strictly: the extent to which they are satisfied is merely expected to improve the efficiency of the inference. For example, satisfying the first property requires knowledge of the marginal posterior distribution over $\mathbf{y}_e$, which is infeasible to calculate. We note that there is likely to be tension between the two properties, so that a balance must be struck between them.

3.4.2 *k*-d tree blocking function

We now describe a practical blocking function based on k-d trees, which is used in our experiments in Section 3.7.

Background. A k-d tree is a binary tree that recursively partitions a k-dimensional affine space [Ben75; FBF77]. In the standard set-up, each node of the tree is associated with a data point that implicitly splits the input space into two half-spaces along a particular dimension. Owing to its ability to hierarchically group nearby points, it is commonly used to speed up nearest-neighbour search. This makes a k-d tree a good candidate for a blocking function, since it can be balanced while grouping similar points.

Setup. Our setup differs from a standard k-d tree in several aspects. Firstly, we are considering a discrete space $\mathcal{D}_\otimes$ (not an affine space), where the "k dimensions" are the A attributes. Secondly, we do not store data points in the tree. We only require that the tree implicitly stores the boundaries of the blocks, so that it can assign an arbitrary $\mathbf{y} \in \mathcal{D}_\otimes$ to the correct block (a leaf node). Thirdly, since we are working in a discrete space, the input space to a node is a countable set. The node must split the input set into two parts based on the values of one of the attributes.

Fitting the tree. Since it is infeasible to calculate the marginal posterior distribution over $\mathbf{y}_e$ exactly, we use the empirical distribution from the records as an approximation. In other words, we treat the records as a sample from the distribution over $\mathbf{y}_e$, and fit the tree so that it remains balanced with respect to this sample. The depth of the tree d determines the number of blocks (2^d).

Achieving balanced splits. When fitting the tree, each node receives an input set of samples, and a rule must be found that splits the set into two roughly equal (balanced) parts based on an attribute. In ordinary k-d trees, the median is often used for this purpose, however it is not appropriate for the discrete input sets that we encounter. As a result, we propose the following alternative splitting rules:

(i) *Ordered median.* This rule is appropriate if the set of input attribute values is large and/or has a natural ordering. If there is no natural ordering, an artificial ordering must be applied (e.g. lexicographic ordering). The splitting rule is determined by sorting the input values and finding the median, accounting for the frequency of each value. Attribute values ordered before (after) the median are passed to the left (right) child node.

(ii) *Reference set.* This rule is appropriate if the set of input attribute values is small with no natural ordering. The splitting rule is determined by using a first-fit bin-packing algorithm to split the values into two roughly equal-sized bins, accounting for the frequency of each value. One of these bins is then labelled the "reference set". Attribute values (not) in the reference set are passed to the left (right) child node.

We allow the user to specify an ordered list of attributes to be used for splitting. To ensure balanced splits, we recommend selecting attributes with a large domain. If possible, we recommend preferencing attributes which are known a priori to be reliable (low distortion), as this will reduce the shuffling of entities/records between blocks. In principle, it is possible to automate the process of fitting a tree: one could grow several trees with randomly-selected splits and use the one that is most balanced. We examine balance empirically in Section 3.7.3.

3.5 Inference

We now turn to approximating the full joint posterior distribution over the unobserved variables $\mathbf{Z}$, $\mathbf{Y}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$, as given in (3.4). Since it is infeasible to sample from this distribution directly, we design MCMC algorithms based on partially-collapsed Gibbs

(PCG) sampling [DP08]. In addition, we show how to exploit the conditional independence induced by the blocks to distribute the PCG sampling across multiple threads or machines.

3.5.1 Partially-collapsed Gibbs sampling

Following Steorts [Ste15], we initially experimented with regular Gibbs sampling.[2] However, the resulting Markov chains exhibited slow convergence and poor mixing. This is a known shortcoming of Gibbs sampling which may be remedied by collapsing variables and/or updating correlated variables in groups [Liu04]. These ideas form the basis for a framework called *partially-collapsed Gibbs (PCG) sampling*—a generalisation of Gibbs sampling that generally has better convergence properties [DP08].

Under the PCG framework, variables are updated in groups by sampling from their conditional distributions. These conditional distributions may be taken with respect to the joint posterior (like regular Gibbs), or with respect to *marginal distributions* of the joint posterior (unique to PCG). The latter case is called *trimming* and must be handled with care so as not to alter the stationary distribution of the Markov chain.

In applying PCG sampling to `d-blink`, we must decide how to apply the three tools: *marginalisation* (equivalent to grouping), *permutation* (changing the order of the updates) and *trimming* (removing marginalised variables). In theory, the convergence rate should improve with more marginalisation and trimming, however this must be balanced with the following: (i) whether the resulting conditionals can be sampled from efficiently, and (ii) whether the resulting dependence structure is compatible with our distributed set-up (see Section 3.5.2). We consider two samplers, PCG-I and PCG-II, described below. Of the two, we recommend PCG-I as it is more efficient in our empirical evaluations (see Section 3.7.3). We include PCG-II, as one would expect PCG-II to perform better than PCG-I in terms of mixing, however when computational efficiency is taken into account the performance is worse (see Figure 3.6).

PCG-I sampler

The PCG-I sampler uses regular Gibbs updates for θ_{sa}, λ_i and z_{ia} for all s, i and a. The conditional distributions for these updates are listed in Appendix A. When updating the entity attributes y_{ea} and the block assignments γ_i, marginalisation and trimming are used. Specifically, we apply marginalisation by jointly updating $\mathbf{y}_e$ and $\{\gamma_i, \mathbf{z}_i\}_{i \in \mathcal{R}_e}$ (the set of γ_i's and $\mathbf{z}_i$'s for records i linked to entity e). We then trim (analytically integrate over) $\{\mathbf{z}_i\}_{i \in \mathcal{R}_e}$.

We shall now derive this update. From (3.3), the joint posterior of $\mathbf{y}_e$, $\{\gamma_i, \mathbf{z}_i\}_{i \in \mathcal{R}_e}$ conditioned on the other parameters has the form

$$p\Big(\mathbf{y}_e, \{\gamma_i, \mathbf{z}_i\}_{i \in \mathcal{R}_e} \Big| \mathbf{Z}^{\neg \mathcal{R}_e}, \boldsymbol{\Gamma}^{\neg \mathcal{R}_e}, \boldsymbol{\Theta}, \boldsymbol{\Lambda}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}\Big) \propto$$
$$\prod_a \Big\{ p(y_{ea}|\phi_a) \times \prod_{i \in \mathcal{R}_e} p(\gamma_i|\Upsilon) p(\lambda_i|\gamma_i, \Upsilon) p(z_{ia}|\theta_{s_i a}) \times \prod_{\substack{i \in \mathcal{R}_e \\ o_{ia}=1}} p(x_{ia}|z_{ia}, \lambda_i, y_{ea}) \Big\},$$

[2] We define *regular* Gibbs sampling as the most basic variation where variables are updated iteratively one-at-a-time by sampling from their conditional distributions.

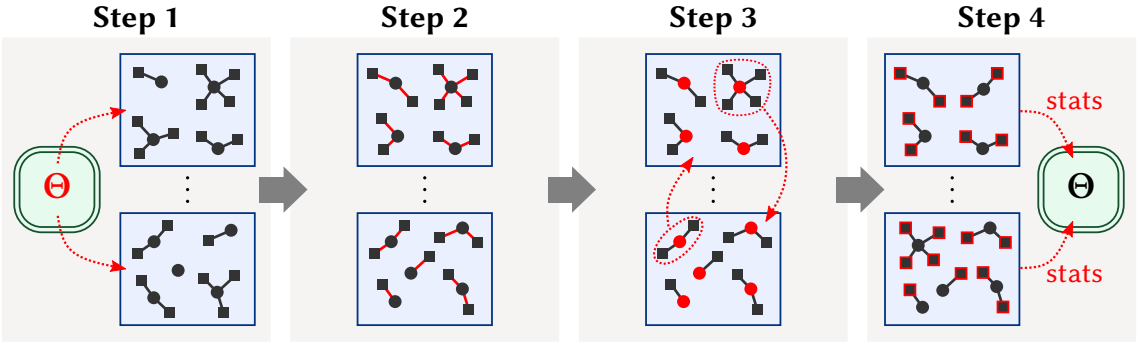


Figure 3.2: Schematic depicting a single iteration of distributed PCG sampling. The entity attributes (Y—circular nodes), record attributes and their distortion indicators (X, Z—square nodes), and links from records to entities (Λ—node connectors) are distributed across the workers (blue rectangular plates) according to their assigned blocks. The distortion probabilities (Θ) reside on the manager (green rounded-rectangular plate).

where superscript $\neg\mathcal{R}_e$ denotes exclusion of any records in $\mathcal{R}_e$ (those currently linked to entity e). Substituting the distributions and trimming $\{\mathbf{z}_i\}_{i\in\mathcal{R}_e}$ yields

$$p\left(\mathbf{y}_e, \{\gamma_i\}_{i\in\mathcal{R}_e} \middle| \mathbf{Z}^{\neg\mathcal{R}_e}, \mathbf{\Gamma}^{\neg\mathcal{R}_e}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}\right) = p(\{\gamma_i\}_{i\in\mathcal{R}_e}|\mathcal{R}_e, \mathbf{y}_e) \prod_a p(y_{ea}|\mathcal{R}_e, \mathbf{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \tag{3.7}$$

where

$$p(y_{ea}|\mathcal{R}_e, \mathbf{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto \phi_a(y_{ea}) \prod_{\substack{i\in\mathcal{R}_e \\ o_{ia}=1}} \{(1-\theta_{s_i a})\mathbb{I}[x_{ia} = y_{ea}] + \theta_{s_i a}\psi_a(x_{ia}|y_{ea})\}$$

$$\text{and} \quad p(\{\gamma_i\}_{i\in\mathcal{R}_e}|\mathcal{R}_e, \mathbf{y}_e) \propto \prod_{i\in\mathcal{R}_e} \mathbb{I}[\gamma_i = \texttt{PartFn}(\mathbf{y}_e)]\,.$$

Note that the update for $\{\gamma_i\}_{\mathcal{R}_e}$ is deterministic, conditional on $\mathbf{y}_e$ and $\mathcal{R}_e$.

Since we have applied trimming, we must permute the updates so that the trimmed variables Z are not conditioned on in later updates. This means the updates for $\mathbf{y}_e$ and $\{\gamma_i, \mathbf{z}_i\}_{\mathcal{R}_e}$ must come *after* the updates for θ_{sa} and λ_i, but *before* the updates for z_{ia}.

PCG-II sampler

The PCG-II sampler is identical to PCG-I, except that it replaces the regular Gibbs update for λ_i with an update that marginalises and trims $\mathbf{z}_i$. To derive the distribution for this update, we first consider the joint posterior of λ_i and $\mathbf{z}_i$ conditioned on the other parameters:

$$p(\lambda_i, \mathbf{z}_i|\mathbf{\Gamma}, \mathbf{Y}, \mathbf{\Theta}, \mathbf{Z}^{\neg i}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto p(\lambda_i|\gamma_i, \mathbf{Y}) \times \prod_a p(z_{ia}|\theta_{s_i a}) \times \prod_{\substack{a \\ o_{ia}=1}} p(x_{ia}|z_{ia}, \lambda_i, y_{\lambda_i a})$$

Table 3.2: Dependencies for the conditional updates used in the PCG-I sampler.

Update variables	Dependencies
θ_{sa}	$\sum_{i:s_i=s} z_{ia}$
λ_i	$\mathbf{z}_i$, $\mathbf{x}_i$, γ_i, $\mathscr{E}_{\gamma_i}$, $\{\mathbf{y}_e\}_{e\in\mathscr{E}_{\gamma_i}}$
y_{ea}, $\{\gamma_i, z_{ia}\}_{i\in\mathscr{R}_e}$	$\mathscr{R}_e$, $\{x_{ia}, s_i\}_{i\in\mathscr{R}_e}$, $\boldsymbol{\Theta}$
z_{ia}	x_{ia}, s_i, λ_i, $y_{\lambda_i a}$, $\theta_{s_i a}$

where superscript $\neg i$ denotes exclusion of record i. Substituting the distributions and trimming $\mathbf{z}_i$ yields

$$p(\lambda_i|\boldsymbol{\Gamma}, \mathbf{Y}, \boldsymbol{\Theta}, \mathbf{Z}^{\neg i}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto$$
$$\mathbb{I}\left[\lambda_i \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right] \times \prod_{\substack{a \\ o_{ia}=1}} \left\{(1-\theta_{s_i a})\mathbb{I}[x_{ia} = y_{\lambda_i a}] + \theta_{s_i a}\psi_a(x_{ia}|y_{\lambda_i a})\right\}.$$

3.5.2 Distributing the sampling

By examining the conditional distributions derived in the previous section and those listed in Appendix A, one can show that the updates for the variables associated with entities and records (z_{ia}, λ_i, γ_i and y_{ea}) only depend on variables associated with entities and records assigned to the *same* block (excluding $\boldsymbol{\Theta}$). These dependencies are summarised in Table 3.2 for the PCG-I sampler. The distortion probability θ_{sa} is an exception—it is not associated with any block and may depend on z_{ia}'s across *all* blocks.

This dependence structure—in particular, the conditional independence of entities and records across blocks—makes the PCG sampling amenable to distributed computing. As such, we propose a manager-worker architecture where:

- the *manager* is responsible for storing and updating variables *not* associated with any block (i.e. $\boldsymbol{\Theta}$); and
- each *worker* represents a block, and is responsible for storing and updating variables associated with the entities and records assigned to it.

The manager/workers may be processes running on a single machine or on machines in a cluster. If using a cluster, we recommend that the nodes be tightly coupled, as frequent communication between them is required.

Figure 3.2 depicts a single iteration of PCG sampling using our proposed manager-worker architecture. Of the four steps depicted, Steps 2 and 3—where the links, entity attributes and block assignments are updated—are the most computationally intensive. We therefore expect to achieve a significant speed-up by distributing these steps across the workers.

To ensure good load balancing of these steps it is important that the blocks are well-balanced (see Section 3.4.1), otherwise workers responsible for smaller blocks must wait idly for other workers to finish before the next iteration can begin. This is because Step 1 requires global synchronisation of state across the workers. The blocks also have an effect on communication costs, which are most significant in Step 3, where the entities and linked records are shuffled to their newly-assigned blocks. A well-chosen blocking function can minimise this cost, by ensuring similar records/entities are co-blocked.

3.6 Achieving fast Gibbs updates

We now outline several ideas for improving the computational efficiency of the Gibbs updates. Section 3.6.1 presents a sub-quadratic algorithm for updating the linkage structure $\mathbf{\Lambda}$. Sections 3.6.2 and 3.6.3 demonstrate a fast method for updating the entity attributes $\mathbf{Y}$.

3.6.1 Efficient pruning of candidate links

In this section, we describe a trick that is aimed at improving the computational efficiency of the Gibbs update for λ_i (used in the Gibbs and PCG-I samplers). This particular trick is incompatible with the joint PCG update for λ_i and $\mathbf{z}_i$ (used in the PCG-II sampler).

Consider the conditional distribution for the λ_i update listed in Appendix A:

$$p(\lambda_i = e|\mathbf{\Gamma}, \mathbf{Y}, \mathbf{Z}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto \mathbb{I}\left[e \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right] \times \prod_{\substack{a \\ o_{ia}=1}} \left\{(1 - z_{ia})\mathbb{I}[x_{ia} = y_{ea}] + z_{ia}\psi_a(x_{ia}|y_{ea})\right\}. \tag{3.8}$$

The support of this distribution is the *set of candidate links* for record i, which we denote by $\mathscr{L}_i$. Looking at the first indicator function above, we see that $\mathscr{L}_i \subseteq \mathscr{E}_{\gamma_i}$, i.e. the candidate links are restricted to the entities in the *same block* as record i. Thus, a naïve sampling approach for this distribution takes $O(|\mathscr{E}_{\gamma_i}|)$ time.

We can improve upon the naïve approach by exploiting the fact that $\mathscr{L}_i$ is often considerably smaller than $\mathscr{E}_{\gamma_i}$. To see why this is the case, note that the second indicator function in (3.8) further restricts $\mathscr{L}_i$ if any of the distortion indicators for the observed record attributes are zero. Specifically, if $z_{ia} = 0$ and $o_{ia} = 1$, $\mathscr{L}_i$ cannot contain any entity whose a-th attribute y_{ea} does not match the record's a-th attribute x_{ia}. This implies $\mathscr{L}_i$ is likely to be small in the case of low distortion.

Putting aside the computation of $\mathscr{L}_i$ for the moment, this means we can reduce the time required to update λ_i to $O(|\mathscr{L}_i|)$. To compute $\mathscr{L}_i$ efficiently, we propose maintaining an inverted index over the entity attributes within each block. Specifically, the index for the a-th attribute in block b should accept a query value $v \in \mathscr{D}_a$ and return the set of entities that match on v:

$$\mathscr{M}_{pa}(v) = \{n \in \mathscr{E}_p \,:\, y_{ea} = v\}. \tag{3.9}$$

Once the index is constructed, we can efficiently retrieve the set of candidate links for record i by computing a multiple set intersection:

$$\mathscr{L}_i = \bigcap_{\{a\,:\,z_{ia}=0 \wedge o_{ia}=1\}} \mathscr{M}_{\gamma_i a}(x_{ia}). \tag{3.10}$$

This assumes at least one of the observed record attributes is not distorted. Otherwise $\mathscr{L}_i = \mathscr{E}_{\gamma_i}$.

Since the sizes of the sets $\mathscr{M}_{\gamma_i a}(x_{ia})$ are likely to vary significantly, we advise computing the intersection iteratively in increasing order of size. That is, we begin with the smallest set and retain the elements that are also in the next largest set, and so on. With a hash-based set implementation, this scales linearly in the size of the first (smallest) set.

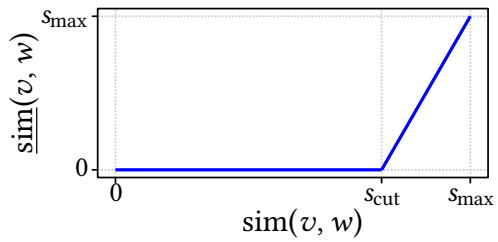


Figure 3.3: Transformation from a raw similarity function (sim) to a truncated similarity function ($\underline{\mathrm{sim}}$).

3.6.2 Caching and truncation of attribute similarities

We have not yet emphasised that the updates for $\mathbf{\Lambda}$, $\mathbf{Y}$ and $\mathbf{\Gamma}$ depend on the attribute similarities between pairs of values in the attribute domains. Specifically, for each attribute a, we need to access the indexed set of attribute similarities $\mathcal{S}_a = \{\mathrm{sim}_a(v, w) : v, w \in \mathcal{D}_a \times \mathcal{D}_a\}$. These similarities may be expensive to evaluate on-the-fly, so we cache the results in memory on the workers.

To manage the quadratic scaling of $\mathcal{S}_a$, and in anticipation of another trick introduced in Section 3.6.3, we transform the similarities so that those *below* a cut-off $s_{\mathrm{cut};a}$ are regarded as completely disagreeing. We achieve this by applying the following truncation transformation to the raw attribute similarity $\mathrm{sim}_a(v, w)$:

$$\underline{\mathrm{sim}}_a(v, w) = \max\left(0, \frac{\mathrm{sim}_a(v, w) - s_{\mathrm{cut};a}}{1 - s_{\mathrm{cut};a}/s_{\max;a}}\right). \tag{3.11}$$

as illustrated in Figure 3.3. Whenever a raw attribute similarity is called for, we replace it with this truncated version. Only pairs of values with positive truncated similarity are stored in the cache—those not stored in the cache have a truncated similarity of zero by default. Note that attributes with a constant similarity function $\mathrm{sim}_{\mathrm{const}}$ are treated specially—there is no need to cache the index set of similarities, since they are all identical.

It is important to acknowledge that the truncated similarities are an approximation to the original model. We claim that the approximation is reasonable on the following grounds:

- *Low loss of information.* Below a certain cut-off, the attribute similarity function is unlikely to encode much useful information for modelling the distortion process. For example, the fact that the normalised edit similarity of "Smith" and "Chiu" is 0.385, whereas the similarity of "Smith" and "Chen" is 0.286, doesn't necessarily imply that "Chiu" is more likely than "Chen" as a distorted alternative to "Smith".

- *Precedent.* In the record linkage literature, value pairs with similarities below a cut-off are regarded as completely disagreeing [Win02; EFI17].

- *Efficiency gains.* As we shall soon see in Section 3.6.3, we can perform the combined $\mathbf{Y}$, $\mathbf{\Gamma}$, $\mathbf{Z}$ update more efficiently by eliminating pairs below the cut-off from consideration.

3.6.3 Fast updates of entity attributes using perturbation sampling

We now present a novel sampling algorithm that allows us to efficiently perform the PCG update for y_{ea} and $\{\gamma_i, z_{ia}\}_{\mathcal{R}_e}$. The algorithm relies on the observation that the conditional distribution for y_{ea} can be expressed as a mixture of two components:

(i) a *base distribution* over $\mathcal{D}_a$ which is ideally constant for all entities; and

(ii) a *perturbation distribution* which varies for each entity, but has a much smaller support than $\mathcal{D}_a$.

With this representation, we can avoid computing and sampling from the full distribution over $\mathcal{D}_a$, which varies for each y_{ea} update. Rather, we only need to compute the perturbation distribution over a much smaller support, and then sample from the mixture, which can be done efficiently using the Vose-Alias method [Vos91]. We refer to this algorithm as *perturbation sampling*.

Perturbation sampling

Although we're interested in applying perturbation sampling to a specific conditional distribution, we shall describe the idea in generality.

Consider a target probability mass function (pmf) $p(x|\omega)$ with finite support $\mathcal{X}$, which varies as a function of parameters $\omega \in \Omega$. In general, one must recompute the probability tables to draw a new variate whenever ω changes—a computation that takes $O(|\mathcal{X}|)$ time. However, if the dependence on ω is of a certain restricted form, we show that it is possible to achieve better scalability by expressing the target as a mixture. This is made precise in the following result.

Proposition 3.4. *Let $p(x|\omega)$ be a pmf with finite support $\mathcal{X}$, which depends on parameters $\omega \in \Omega$. Suppose there exists a "base" pmf $q(x)$ over $\mathcal{X}$ which is independent of ω and a non-negative bounded perturbation term $\epsilon(x|\omega)$, such that $p(x|\omega)$ can be factorised as $p(x|\omega) \propto q(x)(1 + \epsilon(x|\omega))$. Then $p(x|\omega)$ can be expressed as a mixture over the base pmf $q(x)$ and a "perturbation" pmf $v(x|\omega) := c\, q(x)\epsilon(x|\omega)$ over $\mathcal{X}^\star = \{x \in \mathcal{X} : \epsilon(x|\omega) > 0\}$ as follows:*

$$p(x|\omega) = \frac{c}{1+c} q(x) + \frac{1}{1+c} v(x|\omega) \tag{3.12}$$

where $c^{-1} := \sum_{x \in \mathcal{X}^\star} q(x)\epsilon(x|\omega)$.

Proof. The result is straightforward to verify by substitution. □

Algorithm 3.1 shows how to apply this result to draw random variates from a target pmf. Briefly, it consists of three steps: (i) the perturbation pmf v and its normalisation constant c are computed; (ii) a biased coin is tossed to choose between the base pmf q and the perturbation pmf v; and (iii) a random variate is drawn from the selected pmf. If q is selected, a pre-initialised Alias sampler is used to draw the random variate (reused for all ω). Otherwise if v is selected, a new Alias sampler is instantiated. The result below states the time complexity of this algorithm.

Proposition 3.5. *Algorithm 3.1 returns a random variate from the target pmf $p(x|\omega)$ for any $\omega \in \Omega$ in $O(|\mathcal{X}^\star|)$ time.*

Algorithm 3.1 Perturbation sampling for $p(x|\omega)$

Input: map from $x, \omega \in \mathcal{X}^\star \times \Omega \longrightarrow \epsilon(x|\omega)$; map from $x \in \mathcal{X} \longrightarrow q(x)$; pre-initialised Alias sampler for q.

1: $v \longleftarrow \emptyset$ ▷ empty map
2: **for** $x \in \mathcal{X}^\star$ **do**
3: $\quad v(x) \longleftarrow q(x)\epsilon(x|\omega)$
4: **end for**
5: $c \longleftarrow 1/\sum_{x\in\mathcal{X}^\star} v(x)$ ▷ normalisation
6: $s \sim \text{Bernoulli}\left[\frac{c}{1+c}\right]$
7: **if** $s = 1$ **then**
8: $\quad$ **Return**: $x \sim q(\cdot)$ ▷ using input Alias sampler
9: **else**
10: $\quad v \longleftarrow c \times v$
11: $\quad$ **Return**: $x \sim v(\cdot)$ ▷ using new Alias sampler
12: **end if**

Proof. Lines 2–6 are $O(|\mathcal{X}^\star|)$. By properties of the Alias sampler [Vos91], line 8 is $O(1)$ and line 11 is $O(|\mathcal{X}^\star|)$. Thus the overall complexity is $O(|\mathcal{X}^\star|)$. □

This is a promising result, since the size of the perturbation support $|\mathcal{X}^\star|$ is typically of order 10 for our application, while the size of the full support $|\mathcal{X}|$ may be as large as 10^5 for some of the data sets we considered in Section 3.7. Hence, we expect a significant speed-up over the naïve approach.

Application of perturbation sampling

We now return to our original objective: performing the joint PCG update for y_{ea} and $\{\gamma_i, z_{ia}\}_{\mathcal{R}_e}$. Referring to (3.7), we can express the conditional distribution for y_{ea} (i.e. the target distribution) as

$$p(y_{ea} = v|\mathcal{R}_e, \boldsymbol{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto q_a(v|\mathcal{R}_e, \mathbf{O}) \left(1 + \epsilon_a(v|\mathcal{R}_e, \boldsymbol{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S})\right).$$

The base distribution is given by

$$q_a(v|\mathcal{R}_e, \mathbf{O}) \propto \phi_a(v) \left(h_a(v)\right)^{n_a(\mathcal{R}_e, \mathbf{O})} \tag{3.13}$$

where $n_a(\mathcal{R}_e, \mathbf{O}) = |\{i \in \mathcal{R}_e \,:\, o_{ia} = 1\}|$ is the number of records linked to entity e with observed values for attribute a; and the perturbation term is given by

$$\epsilon_a(v|\mathcal{R}_e, \boldsymbol{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) = \prod_{\substack{i\in\mathcal{R}_e \\ o_{ia}=1}} \left\{ e^{\underline{\text{sim}}_a(x_{ia}, v)} + \frac{(\theta_{s_i a}^{-1} - 1)\, \mathbb{I}[x_{ia} = v]}{\phi_a(x_{ia}) h_a(x_{ia})} \right\} - 1.$$

The full support of the target pmf is $\mathcal{D}_a$, while the perturbation support is given by

$$\{x_{ia} \,:\, i \in \mathcal{R}_e \wedge o_{ia} = 1\} \;\cup\; \{v \in \mathcal{D}_a \,:\, \underline{\text{sim}}_a(v, x_{ia}) > 0 \wedge o_{ia} = 1 \text{ for any } i \in \mathcal{R}_e\}.$$

In words, this set consists of the observed values for attribute a in the records linked to entity e, plus any sufficiently similar values from the attribute domain (for which the

Table 3.3: Summary of data sets used in the empirical study. Those marked with a '⋆' are synthetic.

Data set	# records (N)	# sources (S)	# entities	# attributes (A)	
				categorical	string
⋆ ABSEmployee	660,000	3	400,000	4	0
NCVR	448,134	2	296,433	3	3
NLTCS	57,077	3	34,945	6	0
SHIW0810	39,743	2	28,584	8	0
⋆ RLdata10000	10,000	1	9,000	2	3

truncated similarity is non-zero). The size of the perturbation set will vary depending on the cut-off used for the truncation transformation—the higher the cut-off, the smaller the set. This implies that there is a trade-off between efficiency (small perturbation set) and accuracy (lower cut-off).

Remark 3.1. *The astute reader may have noticed that the base distribution q_a given in (3.13) is not completely independent of the conditioned parameters, as is required by Proposition 3.4. In particular, q_a depends on $n_a(\mathcal{R}_e, \mathbf{O})$, which is the size of entity e when all record attribute values are observed. Fortunately, we expect the range of regularly encountered entity sizes to be small, so we sacrifice some memory by instantiating multiple Alias samplers for each $n_a(\mathcal{R}_e, \mathbf{O})$ over some expected range. In the worst case, when a value is encountered outside the expected range and the base distribution is required (unlikely since the weight on the base component is typically small), we instantiate the base distribution on-the-fly, which has the same asymptotic cost as the naïve approach.*

3.7 Empirical evaluation

We shall now conduct an empirical evaluation of `d-blink`, to assess its scalability and accuracy. Section 3.7.1 provides a summary of the five data sets used in our experiments. Section 3.7.2 details the experimental setup, including details of the hardware, implementation and parameter settings. We present the results in three sections. Section 3.7.3 focuses on scalability and computational efficiency, Section 3.7.4 compares the ER predictions against baselines and Section 3.7.5 provides a sensitivity analysis.

3.7.1 Data sets

We experiment with three synthetic and two real data sets, as summarised in Table 3.3. All data sets come with ground truth entity identifiers (of varying reliability), which we use for evaluation purposes. A summary of each data set is provided below.

- `ABSEmployee`. A synthetic data set used internally for linkage experiments at the Australian Bureau of Statistics. It simulates an employment census and two supplementary surveys (it is not derived from any real data sources). We used four categorical attributes: `MB`, `BDAY`, `BYEAR` and `SEX`.

- `NCVR`. Two snapshots from the North Carolina Voter Registration database taken two months apart [Chr14]. The snapshots are filtered to include only those voters whose details changed over the two-month period. We used `first_name`, `middle_name` and `last_name` as string-type attributes. `age`, `gender` and `zip_code` were used as categorical attributes. Although unique voter identifiers are included, they are known to contain some errors [Chr14].
- `NLTCS`. A subset of the U.S. National Long-Term Care Survey [Man10] comprising the 1982, 1989 and 1994 waves. It was necessary to use a subset, as race was sub-sampled in the other three years, making it unsuitable for ER. We used four categorical attributes: `SEX`, `DOB`, `STATE` and `REGOFF`. The included unique identifiers are known to be of high quality.
- `SHIW0810`. A subset from the Bank of Italy's Survey on Household Income and Wealth [Ban] comprising the 2008 and 2010 waves. We used eight categorical attributes: `IREG`, `SESSO`, `ANASC`, `STUDIO`, `PAR`, `STACIV`, `PERC` and `CFDIC`. Unique identifiers were inferred using a deterministic algorithm, and are of unknown quality.[3]
- `RLdata10000`. A synthetic data set distributed with the `RecordLinkage` R package [SB10]. We used `fname_c1` and `lname_c1` as string-type attributes and `bd`, `bm`, `by` as categorical attributes. The `fname_c2` and `lname_c2` were excluded as they have a high fraction of missing values.

3.7.2 Setup

We ran all experiments using an implementation of `d-blink` built on the Apache Spark distributed computing framework [Zah+16]. Since `d-blink` requires control over the partitioning (entities and records *must* reside on their assigned partitions), we used the RDD API with a custom partitioner.

Hardware. Most results presented here were obtained using a local server running Spark 2.3.1 in local (pseudo-cluster) mode. The server had 2× 28-core Intel Xeon Platinum 8180M CPUs, solid state storage and 2 TB of RAM. We used a maximum of 64 threads for our experiments and allocated 128 GB of RAM on the driver. In order to test the effect of increased communication costs between worker (executor) nodes on commodity cloud hardware, we also replicated some of the results on a Spark YARN cluster running on the Amazon Elastic MapReduce platform (release 5.17.0). We used a m4.large instance (4 vCores, 8 GB memory, 32 GB storage) for the master node and m5.xlarge instances (4 vCores, 16 GB memory, 32 GB storage) for the task nodes.

Hyperparameter settings. We used the following hyperparameter settings for all experiments unless otherwise specified.

- We set the distortion hyperparameters to $\beta_{sa}^{(0)} = \frac{R}{1000}$ and $\beta_{sa}^{(1)} = \frac{R}{10}$. This corresponds to a prior mean distortion probability of approximately 1%, with the strength varying in proportion to the total number of records R.

[3] Further information and open-source code is provided at `http://github.com/ngmarchant/shiw`

- The size of the latent entity population E was set to R as recommended by Steorts [Ste15]. This corresponds to a prior mean number of observed entities of $(1-e^{-1})R \approx 0.63R$ [SHF16]. It is important not to set E too low, as it places an upper bound on the number of entities present in the data.
- The entity attribute distributions $\{\phi_a\}$ were set empirically based on the observed record attributes. Specifically, we set

$$\phi_a(v) = \frac{\sum_{i:\, o_{ia}=1} \mathbb{I}[x_{ia} = v]}{|\{i : o_{ia} = 1\}|} \quad \text{for all } a.$$

- For simplicity, we treated all attributes as either "categorical-type" with similarity function $\text{sim}_{\text{const}}$ or "string-type" with similarity function $10.0 \times \text{sim}_{\text{nEd}}$ (see Section 3.3.4). The similarity cut-off for string-type attributes was set to 7.0, following advice in the `RecordLinkage` R package [SB10].
- We used the k-d tree blocking function as defined in Section 3.4.2. The *reference set* splitting rule was used for input sets with 30 or fewer elements, and the *ordered median* splitting rule was used otherwise.

Initialisation and MCMC. To initialise the Markov chain, we linked each record to a unique entity and copied the record attributes into the entity attributes, assuming no distortion. Any entity attributes that were missing after this process (due to missing record attributes) were filled by drawing an attribute value from the empirical distribution. We set the thinning interval to 10—i.e. we only saved every tenth step along the chain. This increases the effective sample size for a given storage budget.

3.7.3 Computational and sampling efficiency

Following [TVP16], we measured the efficiency using the rate of effective samples produced per unit time (ESS rate), which balances sampling efficiency (related to mixing/autocorrelation) and computational efficiency. We used the `mcmcse` R package [Fle+17] to compute the effective sample size (ESS), which implements a multivariate method proposed by Vats et al. [VFJ19].

Since the number of variables in the model is unwieldy (there are at least $(E+R+T)A+R$ unobserved variables) we computed the ESS for the following summary statistics:

- the number of observed entities (scalar);
- the aggregate distortion for each attribute (vector); and
- the cluster size distribution (vector containing frequency of 0-clusters, 1-clusters, 2-clusters, etc.).

`d-blink` versus `blink`. We compared `d-blink` (using the PCG-I sampler) to our own implementation of `blink` (i.e. a Gibbs sampler without any of the tricks described in Section 3.6). For a fair comparison, we switched off blocking in `d-blink`. We used the smallest data set (`RLdata10000`), as `blink` cannot cope with larger data sets. Figure 3.4

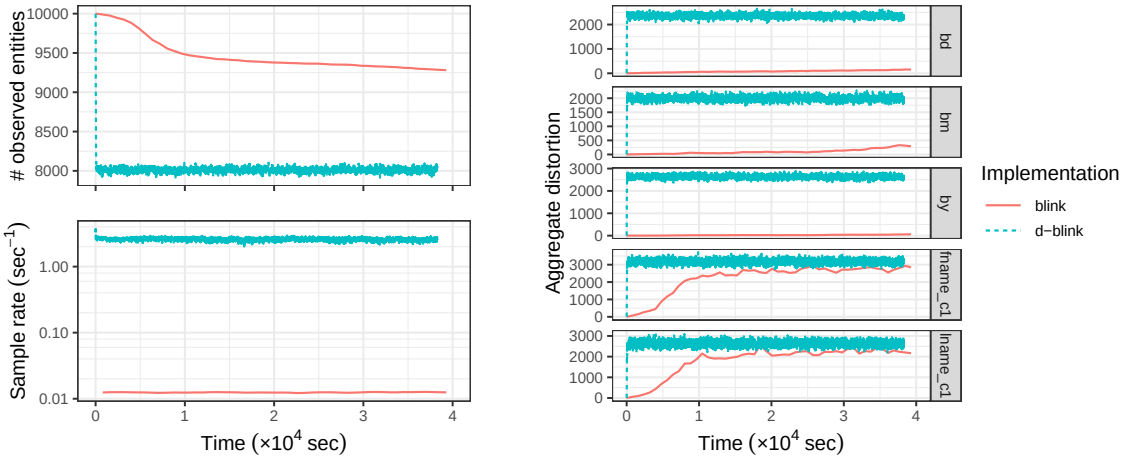


Figure 3.4: Comparison of convergence rates for `d-blink` and `blink`. The summary statistics for `d-blink` (top left and right panels) rapidly converge to equilibrium, while those for `blink` fail to converge within 11 hours. The number of samples generated per second is also substantially lower for `blink` (lower left panel).

contains trace plots for two summary statistics as a function of running time. It is evident that `blink` has not converged to the equilibrium distribution within the allotted time of 11 hours, while `d-blink` converges to equilibrium in 100 seconds. Looking solely at the time per iteration, `d-blink` is at least 200× faster than `blink`.

Blocking and efficiency. We tested the effect of varying the number of blocks B on the efficiency of `d-blink`. For each value of B, we computed the ESS rate averaged over 3000 iterations. We used the `NLTCS` data set and the PCG-I sampler. Figure 3.5 presents the results in terms of the speed-up relative to the ESS rate for $B = 1$. On our local server we observe a near-linear speed-up in B, with the exception of $B = 32$. The speed-up is less pronounced when run on the cloud, The speed-up is expected to taper off with increasing numbers of blocks, as parallel gains in efficiency are overcome by losses due to communication costs and/or poorer mixing. This tipping point seems difficult to predict for a given set up, as it depends on complex factors such as the data distribution, the splitting rules used, and the hardware characteristics.

Sampling methods and efficiency. We evaluated the efficiency of the three samplers introduced in Section 3.5.1 (Gibbs, PCG-I and PCG-II) using the ESS rate, averaged over 3000 iterations. We set $B = 16$ and used the `NLTCS` data set. The results, shown in Figure 3.6, indicate that the PCG-I sampler is considerably more efficient (by a factor of 20–100×) than the baseline Gibbs sampler on this data set. The efficiency of the PCG-II sampler is similar to the Gibbs sampler, despite the fact that the PCG-II sampler is expected to have the best mixing properties. This is likely due to the increased computational complexity of the linkage structure update for the PCG-II sampler, which scales quadratically, unlike the sub-quadratic update used in the PCG-I and Gibbs samplers (see Section 3.6.1).

Load balancing. In Section 3.4.2, we proposed a blocking function based on k-d trees, and argued that it could yield balanced blocks with good entity separation. While running

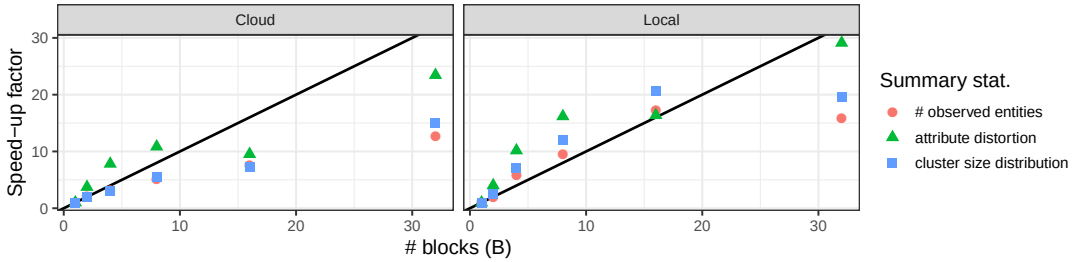


Figure 3.5: Efficiency gain of `d-blink` as a function of the number of blocks. Results are presented for experiments run on the AWS cloud (left panel) and our local server (right panel) for various summary statistics of interest (coloured markers). The speed-up measures the ESS rate relative to the ESS rate for $B = 1$ (no blocking) for the `NLTCS` data set.

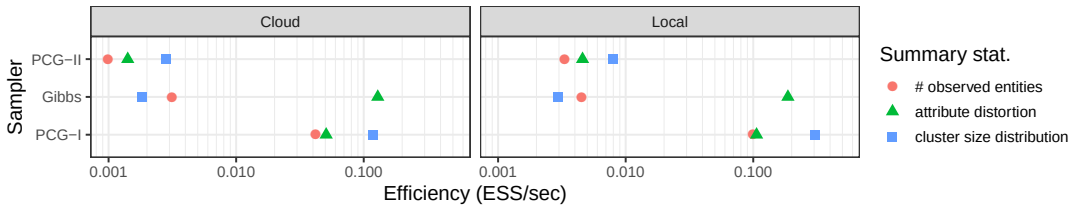


Figure 3.6: Efficiency of `d-blink` as a function of the sampling method. Results are presented for experiments run on the AWS cloud (left panel) and our local server (right panel) for various summary statistics of interest (coloured markers). All measurements are for the NLTCS data set with $B = 16$.

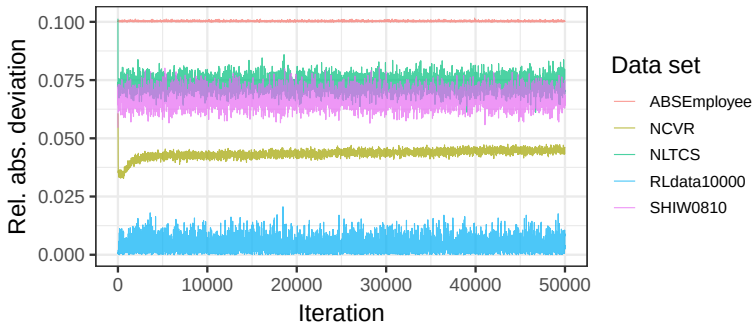


Figure 3.7: Imbalance of the block sizes for a single run on our local server. The imbalance is measured in terms of the relative absolute deviation from the perfectly balanced configuration. The number of blocks for each data set in the order listed in the legend is $B = 64, 64, 16, 2, 8$.

`d-blink` with the k-d tree blocking function, we recorded the size of the blocks ($|\mathcal{E}_b|$ for all b) to assess whether they were well-balanced. Figure 3.7 illustrates the results in terms of the relative absolute deviation from the perfectly balanced configuration (where the entities are divided equally among the blocks). We can see that the k-d tree partitioner is functioning quite well—the deviation from the perfectly balanced configuration is no more than 10% for all data sets.

3.7.4 Linkage quality

Though not our primary focus, we assessed the performance of `d-blink` in terms of its predictions for the linkage structure for the data sets in Table 3.3. This was not previously possible with `blink`, as it could not scale to large data sets.

Point estimate methodology. To evaluate the performance of `d-blink` with respect to the ground truth, we extracted a point estimate of the linkage structure from the approximate posterior samples using the *shared most probable maximal matching sets (sMPMMS)* method [SHF16]. This method circumvents the problem of label switching [JHS05]—where the identities of the entities do not remain constant along the Markov chain.

The sMPMMS method involves two main steps. In the first step, the most probable entity cluster is computed for each record based on the posterior samples. In general, these entity clusters will conflict with one another—e.g. the most probable entity cluster for r_1 might be (r_1, r_2) while for r_2 it could be (r_1, r_2, r_3). The second step resolves these conflicts by assigning precedence to links between records and their most probable entity clusters. The result is a globally-consistent estimate of the linkage structure that satisfies transitivity.

We distributed the computation of the sMPMMS method in the Spark framework. We used 9000 approximate posterior samples which were derived from a Markov chain of length 10^5 by discarding the first 10^4 iterations as burn-in[4] and applying a thinning interval of 10. These parameters were chosen by inspection of trace plots.

[4]We applied a burn-in of 60k iterations for `NCVR` as it was slow to converge.

Baseline methods. We compared `d-blink` with three baseline methods as described below. We focus on (scalable) unsupervised methods as we assumed very little to no labelled data was available for training.

- *Exact Matching.* Links records that match on all A attributes. It is unsupervised and ensures transitivity.
- *Near Matching.* Links records that match on at least $L - 1$ attributes. It is unsupervised, but does not guarantee transitivity.
- *Fellegi-Sunter.* Links records according to a pairwise match score that is a weighted sum of attribute-level dis/agreements. The weights are specified by the Fellegi-Sunter model [FS69] and were estimated using the expectation-maximisation algorithm, as implemented in the `RecordLinkage` R package [SB10]. We chose the threshold on the match score to optimise the F1-score using a small amount of training data (size 10 and 100). This makes the method semi-supervised. Note that the training data was sampled in a biased manner to deal with the imbalance between the matches and non-matches (half with match scores above zero and half below). The method does not guarantee transitivity.

Results. Table 3.4 presents performance measures categorised by data set and method. The pairwise performance measures (precision, recall and F1-score) are provided for all methods, however the cluster performance measures (adjusted Rand Index, see [VEB10], and percentage error in the number of clusters) are only valid for methods that guarantee transitivity of closure (`d-blink` and Exact Matching). Despite being fully unsupervised, `d-blink` achieves competitive performance when compared to the semi-supervised Fellegi-Sunter method. The two simple baselines, Near Matching and Exact Matching, are acceptable for data sets with low noise but perform poorly otherwise (e.g. `NCVR` and `RLdata10000`).

Uncertainty measures. `d-blink` allows for measures of uncertainty to be reported, unlike the baseline methods, since we have access to an approximation of the posterior distribution. For example, in Figure 3.8 we compute posterior estimates for the number of entities present in each data set, with 95 per cent Bayesian credible intervals. Note that the posterior estimates are typically quite sharp. This seems to confirm arguments by Steorts et al. [SHF16] regarding the informativeness of the prior for the linkage structure in `blink`. We examine more flexible priors in Chapter 4.

3.7.5 Sensitivity analysis

We conducted an empirical sensitivity analysis for `d-blink` with respect to variations in the following hyperparameters:

- $\beta_{sa}^{(0)}, \beta_{sa}^{(1)}$: the shape parameters for the Beta prior on the distortion probabilities. We used the same values for all s, a.
- E: the size of the latent population.

Table 3.4: Evaluation of ER quality for `d-blink` and baseline methods. "ARI" stands for adjusted Rand index and "Err. # clust." is the percentage error in the number of clusters.

Data set	Method	Pairwise measures			Cluster measures	
		Precision	Recall	F1-score	ARI	Err. # clust.
ABSEmployee	d-blink	0.9763	0.8530	**0.9105**	**0.9105**	**+1.667%**
	Fellegi-Sunter (10)	**0.9963**	0.8346	0.9083	—	—
	Fellegi-Sunter (100)	**0.9963**	0.8346	0.9083	—	—
	Near Matching	0.0378	**0.9930**	0.0728	—	—
	Exact Matching	0.9939	0.8346	0.9074	0.9074	+9.661%
NCVR	d-blink	0.9146	**0.9654**	**0.9393**	**0.9392**	**−3.587%**
	Fellegi-Sunter (10)	0.9868	0.7874	0.9083	—	—
	Fellegi-Sunter (100)	0.9868	0.7874	0.9083	—	—
	Near Matching	0.9899	0.7443	0.8497	—	—
	Exact Matching	**0.9925**	0.0017	0.0034	0.0034	+51.09%
NLTCS	d-blink	0.8319	0.9103	0.8693	0.8693	−22.09%
	Fellegi-Sunter (10)	**0.9094**	0.9087	**0.9090**	—	—
	Fellegi-Sunter (100)	**0.9094**	0.9087	**0.9090**	—	—
	Near Matching	0.0600	**0.9563**	0.1129	—	—
	Exact Matching	0.8995	0.9087	0.9040	**0.9040**	**+2.026%**
SHIW0810	d-blink	**0.2514**	0.5396	**0.3430**	**0.3429**	−37.65%
	Fellegi-Sunter (10)	0.0028	0.9050	0.0056	—	—
	Fellegi-Sunter (100)	0.0025	**0.9161**	0.0050	—	—
	Near Matching	0.0043	0.9111	0.0086	—	—
	Exact Matching	0.1263	0.7608	0.2166	0.2166	**−37.40%**
RLdata10000	d-blink	0.6334	**0.9970**	0.7747	**0.7747**	**−10.97%**
	Fellegi-Sunter (10)	0.9957	0.6174	0.7622	—	—
	Fellegi-Sunter (100)	0.9364	0.8734	0.9038	—	—
	Near Matching	0.9176	0.9690	**0.9426**	—	—
	Exact Matching	**1.0000**	0.0080	0.0159	0.0159	+11.02%

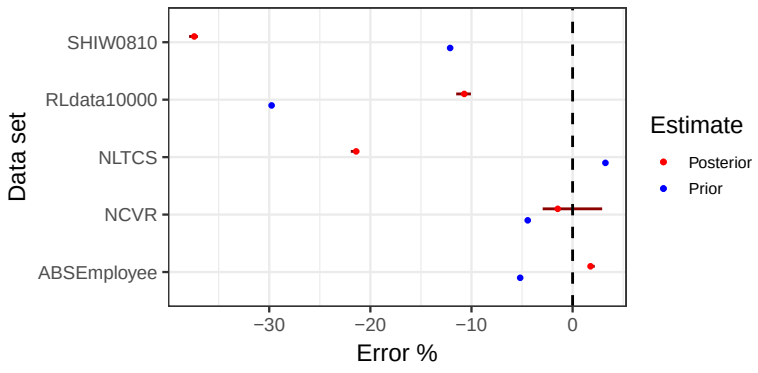


Figure 3.8: Percentage error in the posterior/prior estimates for the number of observed entities for `d-blink`. The posterior estimates are generally sharp and tend to underestimate the true number of entities represented in the data.

Table 3.5: Sensitivity analysis for various hyperparameter combinations using `RLdata10000`. The first group of rows tests the effect of varying the *strength* of the Beta prior, the second group tests the effect of varying the *mean* of the Beta prior, the third group tests the effect of varying the population size, and the fourth group tests the effect of varying the scaling factor for the similarity function.

Distortion		Pop. size	Max. sim.	Pairwise measures			Cluster measures	
$\beta^{(0)}$	$\beta^{(1)}$	E	$s_{\max}$	Precision	Recall	F1-score	ARI	Err. # clust.
0.1	**10.0**	10000	10.0	0.5342	0.9990	0.6962	0.6962	−17.47%
1.0	**100.0**	10000	10.0	0.5435	0.9990	0.7040	0.7040	−16.58%
10.0	**1000.0**	10000	10.0	0.6334	0.9970	0.7747	0.7747	−10.97%
100.0	**10000.0**	10000	10.0	0.9180	0.9850	0.9503	0.9503	−1.595%
10.0	**1000.0**	10000	10.0	0.6334	0.9970	0.7747	0.7747	−10.97%
50.5	**959.5**	10000	10.0	0.6132	0.9970	0.7593	0.7593	−11.90%
101.0	**909.0**	10000	10.0	0.5992	0.9970	0.7485	0.7485	−12.90%
10.0	1000.0	**9000**	10.0	0.5306	0.9970	0.6926	0.6926	−15.65%
10.0	1000.0	**10000**	10.0	0.6334	0.9970	0.7747	0.7747	−10.97%
10.0	1000.0	**11000**	10.0	0.6999	0.9960	0.8221	0.8221	−7.365%
10.0	1000.0	10000	**5.0**	0.6927	0.9940	0.8164	0.8164	−22.12%
10.0	1000.0	10000	**10.0**	0.6334	0.9970	0.7747	0.7747	−10.97%
10.0	1000.0	10000	**50.0**	0.2112	0.3920	0.2745	0.2745	−12.50%

- $s_{\max}$: the scaling factor for the similarity function. This controls the inverse temperature of the softmax distribution for the distorted attribute values.

We used the `RLdata10000` data set, as its relatively small size meant that inference could be run quickly for various hyperparameter combinations.

We varied each of the hyperparameters in turn, while holding all other hyperparameters fixed. For the Beta prior on the distortion probabilities, we first varied the strength while fixing the prior mean to ~ 1%, then we varied the mean (1%, 5% and 10%) while fixing $\beta^{(0)} + \beta^{(1)}$ (related to the strength). Table 3.5 presents the evaluation measures for each combination of parameters. The results indicate that the inferred linkage structure is relatively sensitive to all of the parameters, however sensitivity is in general predictable, following clear and intuitive trends. Of particular interest is the fact that the model performs best when the Beta prior on the distortion probabilities is sharply peaked near zero. It seems that the model has a tendency to overestimate the amount of distortion, particularly in the absence of ground truth.

3.8 Application to the 2010 U.S. Decennial Census[5]

National statistics agencies frequently need to link inter- or intra-agency data sets, for a number of purposes such as quality control. One critical problem in the United States (U.S.) occurs every ten years, when the U.S. Census Bureau must enumerate the population in each State as mandated under the U.S. Constitution, Article I, Section 2. The enumeration is used to apportion the representation of legislators, and to allocate resources for housing,

[5]R. C. Steorts conducted the experiments reported in this section.

highways, schools, assistance programs, and other projects that are vital to the prosperity, welfare, and economic growth of the U.S. As the country grows and becomes more diverse, it becomes more challenging to produce an accurate enumeration. Many individuals elect not to fill out census forms, which results in them not being directly counted in the enumeration. Other individuals may be counted multiple times due to duplicate responses. For example, students attending universities or private schools (living in group quarters) are often double counted as they are legally required to be counted by their university/school, while also being counted by their parents/guardians as part of a household.

Motivated by these data duplication issues, we apply `d-blink` to conduct an enumeration in the state of Wyoming. In order to improve coverage, we combine records from the 2010 Decennial Census with administrative records from the Social Security Administration's Numerical Identification System (Numident).[6] In total, we consider 1,050,000 records representing the population of Wyoming: a subset of 494,000 records from the 2010 Decennial Census and 556,000 records from the Numident.[7] Our goal is to recover the unique individuals represented in these records using unsupervised ER.

We apply `d-blink` using overlapping attributes from the Census and the Numident: first and last name, date of birth, gender, and zip code. We treat first and last name as string-type attributes and the remaining attributes as categorical. To manage scalability, we utilize the k-d tree blocking function outlined in Section 3.4.2, splitting recursively on gender and birth year at each level of the tree. We ran inference for 15,000 iterations using the PCG-I sampler. After removing 5,000 iterations as burn-in and applying thinning with an interval of 10, we obtained 1,000 approximate samples from the posterior. Convergence diagnostics are consistent with those reported for the other data sets, and are complicated to release due to the fact that the data is protected under Title 13.

After performing ER using `d-blink`, we are able to provide a posterior estimate of the total number of unique individuals represented in both data sets. Table 3.6 reports a point estimate based on the mean. The standard error is quite narrow, which is consistent with knowledge of the uniform prior [SHF16]. We find that our estimate is significantly larger than the unadjusted count of 563,626 reported by Rastogi et al. [Ras+12]. The difference may be explained by several factors. Firstly, our approach may capture individuals who are not represented in the Census, but who are represented in the Numident (assuming they have a Social Security number). Indeed, the participation rate for the Census is known to be lower in Wyoming than for other states [Uni]. Secondly, there may be some double-counting for records that cannot be reliably linked—e.g. due to missing or unreliable attribute values. Thirdly, there may be minor differences in the Census data—e.g. whether blank forms are discarded or not.

To assess the reliability of ER, we report pairwise evaluation measures (precision, recall and F1-score) in Table 3.6. These measures are computed using ground truth identifiers, which are available for a limited subset of the records. To our knowledge, these are the first performance measures that have been published for ER of Census and administrative data at the state-level. However, we note that the measures should be interpreted with caution, as the limited ground truth may not be representative of all

[6] The Numident is the Social Security Administration's computer database file of an abstract of the information contained in an application for a U.S. Social Security number.

[7] These figures have been rounded to the nearest thousand as they are protected under Title 13.

Table 3.6: Results for ER of 2010 Census and Numident data in Wyoming. Pairwise evaluation measures are computed using ground truth identifiers available for a subset of the records.

Pairwise measures			Posterior population size	
Precision	Recall	F1-score	Mean	Std. error
0.97	0.84	0.90	616,000	5,000

records (hence the need for unsupervised methods).

We believe that `d-blink` shows promise in producing enumerations at the state-level, while accounting for ER uncertainty. Moving forward, it would be beneficial to study the accuracy and scalability of `d-blink` in other states, to further assess the reliability of our methodology for conducting linkage tasks within national statistical agencies.

3.9 Concluding remarks

In this chapter, we have developed a scalable and distributed method for performing unsupervised ER in a Bayesian framework. Our method, called `d-blink`, extends the `blink` ER model [Ste15], by incorporating blocking for improved scalability, adding support for missing values, and allowing for user-specified attribute similarity functions. We were able to incorporate blocking without compromising the correctness of inference asymptotically, via an auxiliary variable representation. Specifically, we introduced an auxiliary partition of the latent entity space into blocks, and auxiliary block assignments for each record, which are inferred during Markov chain Monte Carlo. This stands in contrast with much of the literature, which assumes the block assignments are fixed a priori.

In addition to blocking, we proposed several ideas for improving the computational and statistical efficiency of inference. These ideas included a partially-collapsed Gibbs sampling algorithm which can be distributed/parallelised at the block-level, as well as computational tricks for speeding up the Gibbs updates. We conducted an empirical study to assess the impact of blocking, and our other ideas on inferential efficiency. Our results showed that all of our ideas lead to substantial gains in efficiency, each by a factor of 10–100×. This allowed us to apply `d-blink` to large data sets of around one million records, including an application to population enumeration using Census and administrative data.

While our approach to scaling was effective, we found that the `blink` model was sensitive to hyperparameters, which is generally regarded as undesirable [BIR00]. In the next chapter, we explore refinements to the `blink` model aimed at reducing sensitivity and improving ER accuracy. Apart from modelling improvements, there are several interesting directions for future work. One could explore *variational inference* [BKM17] as an alternative to Markov chain Monte Carlo. However, while variational inference is generally less computationally demanding, selecting a class of variational distributions which can accurately approximate the posterior may be challenging. Another direction which we did not explore is the integration of `d-blink` with post-ER tasks, such as canonicalisation [MW04] and regression [KBS18]. It would be interesting to see whether

issues arise for large data sets, particularly when the approximate posterior samples become too large to store in memory.

Chapter 4

A flexible model for unsupervised Bayesian entity resolution

This chapter continues our work on Bayesian approaches to entity resolution (ER). Motivated by issues with the `blink` ER model [Ste15] encountered in the previous chapter, we propose refinements aimed at improving accuracy and reducing sensitivity to hyperparameters. Our refined model incorporates: (i) a more flexible class of priors on the linkage structure corresponding to the Ewens-Pitman family of random partitions [Pit06]; (ii) corrections to logic in the distortion model; and (iii) priors on parameters that were held fixed in `blink`. We assess the impact of our refinements empirically, and observe improved performance and robustness when compared to `blink` and an unsupervised ER model by Sadinle [Sad14].

4.1 Introduction

In the previous chapter, we discussed the advantages of Bayesian models for solving entity resolution (ER) tasks, particularly in scenarios where labelled training data is scarce, unavailable or difficult to acquire. We identified poor scalability as a barrier to the adoption of Bayesian ER models, and proposed an effective solution for the `blink` model [Ste15]. Since our focus was on improving the scalability of `blink` without altering the joint posterior, we regarded modelling innovations as being out-of-scope. However, our empirical studies uncovered potential opportunities to improve the `blink` model—particularly in reducing sensitivity to hyperparameters. We therefore explore modelling refinements in this chapter, with the aim of improving goodness of fit and robustness to misspecified priors. While `blink` serves as the primary foundation for this work, we also compare to related Bayesian ER models by Steorts et al. [SHF16] and Steorts et al. [STL18].

Our refinements are focused on three key areas:

(i) we consider a more flexible and general class of priors on the linkage structure;

(ii) we correct logic in the distortion model; and

(iii) we deepen the model, placing priors on parameters that were previously held fixed to improve flexibility.

In making these changes, we are careful to ensure that inference remains computationally tractable, without introducing significant overhead compared to `blink`. We note that refinements (ii) and (iii) are compatible with the blocked/distributed approach to inference presented in Chapter 3. However refinement (i) is not directly compatible, and would require adaptation of the auxiliary blocking scheme. We do not consider auxiliary blocking in this chapter, as the added complexity distracts from our focus on modelling.

A fundamental assumption in the `blink` model, which we retain in our refined model, is *exchangeability* [p. 253 Dur19] of the observed records. Roughly speaking, this means that the order in which records are observed has no bearing on the model parameters. When combined with a consistency assumption, exchangeability implies that the permitted priors on the linkage structure fall within the family of two-parameter Ewens-Pitman (EP) random partitions [p. 62 Pit06]. We consider three parameter regimes from this family, which are related to finite population models, Dirichlet and Pitman-Yor processes. In order to improve flexibility, we place hyperpriors on the EP parameters. To our knowledge, these EP parameter regimes have not been thoroughly tested in the context of ER models, especially when combined with hyperpriors.

To assess the impact of our modelling refinements, we conduct an empirical study using four ER data sets. We compare our refined model to the original `blink` model, and a model proposed by Sadinle [Sad14] which is closely related to the Fellegi-Sunter model [FS69]. Our refined model clearly outperforms `blink`, and achieves the best F1 score on three of the four data sets. In addition, we assess the impact of our refinements to the distortion model and linkage structure prior *separately*, and find that both contribute to the improved performance. Moreover, we show that our refined model is relatively insensitive to the EP parameter regime selected for the linkage structure prior. This is an interesting result, given discussion in the literature about the importance of selecting appropriate clustering priors for ER models [BS14; Mil+15; Zan+16].

Chapter outline. We review related work in Section 4.2 and provide background material on exchangeable random partitions in Section 4.3. In Section 4.4 we present our refined model and provide suggestions for setting hyperparameters. We design an MCMC inference algorithm in Section 4.5 and conduct an empirical evaluation in Section 4.6. In Section 4.7, we summarise our contributions and suggest directions for future work.

4.2 Related work

Our work is most closely related to Bayesian clustering models for ER, which allow for duplicates across and within multiple data sources, while preserving transitivity [Sad14; BS14; Ste15; SHF16; Zan+16; STL18]. Most of these models are generative, assuming records arise as distortions to a set of latent entity attributes. The model by Sadinle [Sad14] is an exception: it combines the Fellegi-Sunter model [FS69] for comparison vectors with a prior on the linkage structure that obeys transitivity constraints. We discuss how the generative models differ from our own in Section 4.4.

Another body of work focuses on relational ER, where the entities/records have a network structure. Pasula et al. [Pas+02] proposed a probabilistic relational ER model for citation matching. Other generative approaches include an application of latent Dirichlet allocation to author ER in citation databases [BG06], and an idealised generative model

for relational data [FLS15]. McCallum and Wellner [MW04] proposed a discriminative ER model based on conditional random fields, which more readily accounts for complex dependencies. However, it is not formulated for structured data and requires labelled training data.

When ER is formulated as a clustering problem, the linkage structure can be interpreted as *partitioning* the records into groups (or clusters) that are linked to the same entity. Thus suitable priors on the linkage structure (partition) can be borrowed from the literature on random partitions. Pitman [Pit95] studied infinitely exchangeable random partitions, which were originally motivated by applications in population genetics [Kin78a; Kin78b]. These are related to the Dirichlet and Pitman-Yor processes, which have been used as priors in ER models [BG06; FLS15; STL18], and the coupon-collector's process used in the ER models of Steorts [Ste15] and Steorts et al. [SHF16].

A recent development is the idea of microclustering: random partitions whose block sizes grow sublinearly in N [Mil+15]. This behaviour is thought to be desirable for ER, however it requires abandoning consistency or exchangeability. Zanella et al. [Zan+16] propose two random partition models based on Gibbs partitions that abandon consistency. Benedetto et al. [BCT17] propose a model that abandons exchangeability. However, they assume observations are ordered, which is not the case in our application. Further work in this area includes the non-exchangeable uniform process [Wal+10] and microclustering priors with bounds on the cluster sizes [KJ16].

4.3 Exchangeable random partitions

In many applications of ER, the observed records have no natural ordering—i.e. we don't know whether record i_1 was generated before or after record i_2. In these circumstances, it is reasonable to assume the records are *exchangeable* [Ald85]. This means our model must be invariant under permutations of the record indices $i \in \{1, \dots, N\}$, which has implications for the allowed priors on the linkage structure. The most general class of priors which satisfy exchangeability, while being consistent as N varies, are *infinite exchangeable random partitions* [Pit06, p. 43]. This class of random partitions is covered by the two-parameter Ewens-Pitman (EP) family [Pit06, p. 62].

We now provide a constructive definition of the EP family through a generalised Chinese restaurant process. An illustration is provided in Figure 4.1. Let (Π_N) denote a sequence of exchangeable random partitions, where Π_N is an exchangeable random partition of the finite set of integers $[N] = \{1, \dots, N\}$. Beginning with $\Pi_1 = \{1\}$, we describe how to generate the elements of the sequence, such that one obtains an infinite exchangeable partition Π_∞ in the limit $N \longrightarrow \infty$. In the first step, we interpret Π_1 as a single customer seated at a table in a restaurant. At the N-th step, a new customer is seated at:

- an occupied table with probability $\frac{N_k - \sigma}{N + \alpha}$, or
- a new table with probability $\frac{\alpha + K\sigma}{N + \alpha}$,

where K is the number of occupied tables at step $N - 1$, N_k is the number of customers seated at table k at step $N - 1$, and (σ, α) are the EP parameters. In our application, tables correspond to entities and customers correspond to records.

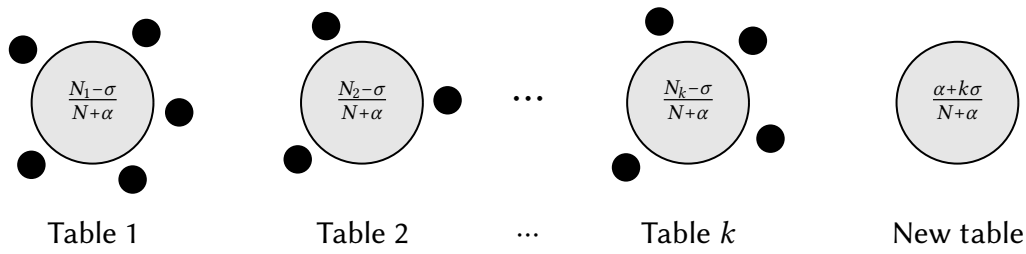


Figure 4.1: Chinese restaurant process construction for a Ewens-Pitman random partition.

The allowable values of the EP parameters fall into two regimes depending on the sign of σ:

- $\sigma < 0$ and $\alpha = -m\sigma$ for some $m \in \mathbb{N}$. We refer to this regime as the *generalised coupon partitions*, since they are closely related to the coupon-collector's partition [Pit06, p. 46]. These partitions are generated by sampling with replacement from a finite population of size m, where the mixing proportions are drawn from a symmetric Dirichlet distribution with concentration parameter $-\sigma$. The coupon-collector's partition is obtained in the limit $\sigma \longrightarrow -\infty$.
- $0 \leq \sigma \leq 1$ with $\alpha > -\sigma$. We refer to this regime as *Pitman-Yor partitions* after Pitman and Yor [PY97]. These partitions are generated by sampling with replacement from an infinite population [see Kin78b], and the resulting partitions demonstrate preferential attachment behaviour. The special case $\sigma = 0$ corresponds to the Ewens partition [Kin78a].

To illustrate the varying behaviour of the random partitions as a function of σ, one can examine the asymptotic number of blocks K_N as $N \longrightarrow \infty$. Pitman [Pit06, p. 70] shows that

$$K_N \overset{a.s.}{\asymp} \begin{cases} m, & \sigma < 0 \text{ and } \alpha = -m\sigma \text{ for } m \in \mathbb{N} \\ \alpha \log N, & \sigma = 0 \text{ and } \alpha > 0, \\ S_\sigma N^\sigma, & 0 < \sigma < 1 \text{ and } \alpha > -\sigma, \end{cases}$$

where S_σ is a strictly positive random variable. Thus by varying σ, we can encode a prior belief that K_N is asymptotically constant, logarithmic or sublinear in N.

4.4 A refined model for ER

In this section, we propose several refinements to the `blink` ER model [Ste15]. We begin by reviewing the problem setting and notation in Section 4.4.1. Then in Section 4.4.2, we review each component of the model, showing how our proposed changes address potential deficiencies with `blink`. Finally, in Section 4.4.4 we provide recommendations for setting hyperparameters.

4.4.1 Problem formulation and notation

We build on the formulation for Bayesian ER that was previously used for `d-blink` in Chapter 3. Since there is significant overlap, we only provide a brief summary here and

Table 4.1: Summary of new notation introduced in this chapter. We continue to use notation from the previous chapter (see Table 3.1) with some redefinitions noted below.

Symbol	Description
$\pi = (\pi_1, \pi_2, \ldots)$	mixing proportions for the entities
σ, α	Ewens-Pitman parameters
$\kappa = -\sigma > 0, m = \frac{\alpha}{\kappa} \in \mathbb{N}$	alternative Ewens-Pitman parameters (finite population)
$\chi^{(0)}, \chi^{(1)}$	hyperparameters for the prior on κ or α
$\zeta^{(0)}, \zeta^{(1)}$	hyperparameters for the prior on σ
r, ν	hyperparameters for the prior on m
$\mathbf{G} = (G_1, \ldots, G_A)$	distributions over attribute domains
ϕ_a	(redefined) base distribution for DP prior on G_a
v_a	concentration parameter for DP prior on G_a
$\mathbf{H}_e = (H_{e1}, \ldots, H_{eA})$	distortion distributions for entity e
$\psi_a(x\|y)$	(redefined) base distribution for prior on H_{ea}
ρ_a	concentration parameter for DP prior on H_{ea}
$\text{dist}_a(\cdot, \cdot)$	distance measure for attribute a
ω_{ia}	distortion propensity for attribute a of record i

refer the reader to Section 3.3.1 for further details. Where possible, we reuse notation from Chapter 3. New notation is summarised in Table 4.1 and introduced in the next section.

We consider performing ER on structured data from one or more data sources. Suppose we observe N records indexed by $i \in \{1, \ldots, N\}$ from data sources indexed by $s \in \{1, \ldots, S\}$. Each record i is associated with a data source $s_i \in \{1, \ldots, S\}$, and is described by a tuple of A attribute values $\mathbf{x}_i = (x_{i1}, \ldots, x_{iA})$ indexed by $a \in \{1, \ldots, A\}$. We assume that the records represent distorted observations of entities from a potentially infinite population indexed by $e \in \mathbb{N}$.[1] Each entity e is described by a tuple of "true" attribute values $\mathbf{y}_e = (y_{e1}, \ldots, y_{eA})$, which may appear distorted in the records. The unobserved relationships between records and entities are represented by the linkage structure $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_N)$, where $\lambda_i \in \mathbb{N}$ denotes the entity linked to record i.

We are interested in the fully unsupervised setting, where ground truth information about the linkage structure and entities is unavailable. Our goal is to infer the unknown linkage structure $\boldsymbol{\Lambda}$ based solely on the observed record attributes $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and source indicators $\mathbf{S} = \{s_1, \ldots, s_N\}$. Since we are interested in uncertainty of inferred linkage structure, we seek an approximation to the posterior—not merely a point estimate.

4.4.2 Model specification

Our refined model retains the fundamental structure of `blink`. Concretely, we model a population of entities and their true attributes, and assume records are instantiated by sampling entities from the population. Once a record is instantiated, it inherits its attribute values from the "true" entity attributes, subject to distortion. We now review each component of the model, while motivating and explaining our proposed changes.

[1] Here we deviate from the formulation in Section 3.3.1, which assumes that the population of entities is finite.

Entity model. We assume the "true" attribute values associated with each entity are generated i.i.d. according to an unknown distribution $\mathbf{G}$ with support on the product space of attribute domains $\mathscr{D} = \prod_a \mathscr{D}_a$. To improve tractability, we assume correlations between attributes are negligible, and place independent Dirichlet Process priors on each component of $\mathbf{G} = (G_1, \ldots, G_A)$, viz.

$$\begin{aligned} G_a &\overset{\text{ind.}}{\sim} \text{DP}(\upsilon_a, \phi_a)\,, && \forall a, \\ y_{ea}|G_a &\overset{\text{ind.}}{\sim} \text{Discrete}(G_a), && \forall e, a, \end{aligned}$$

where $\upsilon_a > 0$ is a concentration parameter and ϕ_a is a base distribution on $\mathscr{D}_a$.

In the limit $\upsilon_a \longrightarrow \infty$, G_a converges in distribution to the base distribution ϕ_a. If in addition ϕ_a is set empirically, based on the relative frequencies of values observed in the records, then the above model for y_{ea} reduces to the one used in `blink`. Our model is likely to be more flexible than `blink`, as G_a is treated as an unknown (random) distribution, rather than a hyperparameter. While the empirical approach used in `blink` is likely to yield a decent approximation, it may break down with increasing levels of distortion. This is because a distorted value v, such as a typographical error, may appear with a relatively high probability in the records, but a correspondingly low probability in the entities—i.e. we have $\phi_a(v) \gg G_a(v)$. By treating G_a as a separate unknown distribution, we can more accurately account for this scenario.

Linkage model. We assume each record i is instantiated by linking to an entity λ_i drawn randomly from the population with replacement, according to unknown mixing proportions $\pi = (\pi_1, \pi_2, \ldots)$. This is a more flexible approach than `blink`, which assumes the mixing proportions π are uniform over a finite population of fixed size E. We can view the random process of linking records to entities as inducing a *random partition* of the records into groups, such that each group is mutually linked to the same entity. Following the discussion about exchangeability in Section 4.3, we assume the random partition is drawn from the Ewens-Pitman (EP) family with parameters (σ, α). The corresponding distribution on the mixing proportions π differs depending on the sign of σ, or equivalently, whether the population of entities is finite or infinite.

For the finite regime (generalised coupon partitions) we let $\sigma = -\kappa < 0$ and $\alpha = m\kappa$ for some $m \in \mathbb{N}$. Our model with hyperpriors on m and κ is as follows:

$$\begin{aligned} \kappa &\sim \text{Gamma}(\chi^{(0)}, \chi^{(1)}), \\ m &\sim \text{NegativeBinomial}(r, \nu) + 1, \\ \pi|\kappa, m &\sim \text{Dirichlet}(\kappa), \\ \lambda_i|\pi &\overset{\text{iid.}}{\sim} \text{Categorical}(\pi), && \forall i, \end{aligned} \tag{4.1}$$

where $\chi^{(0)}, \chi^{(1)}, r > 0$ and $0 < \nu \leq 1$ are hyperparameters and κ is a vector of length m with identical entries κ.[2] Note that the hyperprior on m is a shifted negative binomial distribution with density

$$p(m|r, \nu) = \begin{cases} \frac{(m+r-2)!}{(r-1)!(m-1)!}\, \nu^r (1-\nu)^{m-1}, & m \in \{1, 2, \ldots\}, \\ 0, & \text{otherwise.} \end{cases}$$

[2] The case $\kappa \longrightarrow \infty$ with m fixed corresponds to the linkage model considered in Chapter 3.

In the infinite regime (Pitman-Yor partitions) the mixing proportions are drawn from a two-parameter Poisson-Dirichlet (PD) distribution[3] [PY97]. Our model with hyperpriors on σ and α is as follows:

$$\begin{aligned} \sigma &\sim \text{Beta}(\zeta^{(0)}, \zeta^{(1)}), \\ \alpha &\sim \text{Gamma}(\chi^{(0)}, \chi^{(1)}), \\ \pi|\sigma, \alpha &\sim \text{PoissonDirichlet}(\sigma, \alpha), \\ \lambda_i|\pi &\overset{\text{iid.}}{\sim} \text{Categorical}(\pi), \qquad \forall i, \end{aligned} \tag{4.2}$$

where $\chi^{(0)}, \chi^{(1)}, \zeta^{(0)}, \zeta^{(1)} > 0$ are hyperparameters. Here we assume $\alpha > 0$ and $0 < \sigma < 1$, which is a subset of the admissible parameter space: $0 \leq \sigma \leq 1$ and $\alpha > -\sigma$. We also consider the case where $\sigma = 0$, which corresponds to the Ewens partition.

Due to the inclusion of hyperpriors on the EP parameters, the above priors for the linkage structure are expected to be more flexible than priors used previously in ER models. For example, the coupon-collector's partition has been used as prior for the linkage structure in the `blink` [Ste15] and `SMERED` [SHF16] ER models, although it is recognised as being overly informative [SHF16]. It corresponds to a generalised coupon partition with $\kappa \longrightarrow \infty$ and m fixed. The Pitman-Yor partition has also been used as a prior on the linkage structure for an ER model, however the hyperparameters σ and α were fixed [STL18].

Source model. We reuse the source model from `blink` without any changes. It assumes the source s_i associated with record i is drawn from a distribution ξ over the sources:

$$s_i|\xi \overset{\text{iid.}}{\sim} \text{Discrete}(\xi), \qquad \forall i.$$

There is no need to specify ξ since it is independent of the other model parameters, and the source indicators s_i are fully observed.

Distortion model. We now specify how record attribute values are generated by copying the linked entity attribute values subject to distortion. Following `blink`, we assume the distortion process occurs independently for each attribute. Our proposed model differs from `blink` in three respects:

(i) When deciding whether a record attribute x_{ia} should be copied from the linked entity attribute $y_{\lambda_i a}$ with distortion, we introduce a dependence on $y_{\lambda_i e}$ through a distortion propensity variable ω_{ia}. This accounts for the fact that some entity attribute values are more likely to be distorted than others.

(ii) We model the *distortion distribution*—which selects distorted record values for entity attribute y_{ea}—as a random distribution H_{ea} with a Dirichlet Process prior. This contrasts with `blink`, where the distortion distribution is a hyperparameter, set empirically based on the observed records and specified distance measure.

[3] The Poisson-Dirichlet distribution gives the rank-ordered mixing proportions in decreasing order of frequency.

(iii) We exclude the "true" entity attribute value y_{ea} from the support of the distortion distribution H_{ea}. This means a record attribute value x_{ia} can only be distorted if it differs from the "true" entity attribute value $y_{\lambda_i a}$. In `blink`, x_{ia} can be distorted while agreeing exactly with $y_{\lambda_i a}$. This obscures the meaning of "distortion" and seems to encourage high distortion modes (see Figure 4.3).

First, we detail the model for the binary distortion indicator z_{ia}, which determines whether the entity attribute value $y_{\lambda_i a}$ is copied into x_{ia} with or without distortion. We assume each z_{ia} is drawn independently from a Bernoulli distribution, where the probability of distortion depends conditionally on a distortion propensity ω_{ia} and a source/attribute-level factor $\theta_{s_i a}$. We place a Beta prior on $\theta_{s_i a}$ and assume the distortion propensity ω_{ia} is deterministic conditional on the entity attribute value $y_{\lambda_i a}$. Specifically, we assume

$$\theta_{sa} \overset{\text{ind.}}{\sim} \text{Beta}\left(\beta_{sa}^{(0)}, \beta_{sa}^{(1)}\right) \qquad \forall s, a \tag{4.3}$$

$$\omega_{ia} | y_{\lambda_i a} = \max_{x \in \mathcal{D}_a \setminus \{y_{\lambda_i a}\}} \mathrm{e}^{-\text{dist}_a(y,x)} \qquad \forall i, a$$

$$z_{ia} | \theta_{s_i a}, \omega_{ia} \overset{\text{ind.}}{\sim} \text{Bernoulli}(\theta_{s_i a} \omega_{ia}) \qquad \forall i, a \tag{4.4}$$

where $\beta_{sa}^{(0)}, \beta_{sa}^{(0)} > 0$ are hyperparameters.

As noted previously, the distortion propensity ω_{ia} accounts for the fact that some entity attribute values $y_{\lambda_i a} \in \mathcal{D}_a$ are more likely to be distorted than others. In modelling ω_{ia}, we make use of prior information encoded in the user-specified attribute distance measure dist_a (see Section 4.4.3). If $y_{\lambda_i a}$ is not close to any other values in the domain as measured by dist_a, it is unlikely to be distorted and ω_{ia} approaches zero. On the other hand, if $y_{\lambda_i a}$ is close to at least one other value in the domain, ω_{ia} approaches 1 and distortion can occur. Current entity-centric ER models [Ste15; SHF16; STL18] do not include this logic, and effectively assume that $\omega_{ia} = 1$ for all i, a.

Once the distortion indicator z_{ia} is generated, we can begin the copying process. If $z_{ia} = 0$ then the entity attribute value $y_{\lambda_i a}$ is copied directly into the record attribute value x_{ia}. However, if $z_{ia} = 1$ then an alternative value for x_{ia} is drawn from a distortion distribution $H_{\lambda_i a}$. Mathematically, we write

$$x_{ia} | z_{ia}, y_{\lambda_i a}, H_{\lambda_i a} \overset{\text{ind.}}{\sim} \begin{cases} \delta(y_{\lambda_i a}), & \text{if } z_{ia} = 0, \\ \text{Discrete}(H_{\lambda_i a}), & \text{if } z_{ia} = 1, \end{cases} \qquad \forall i, a \tag{4.5}$$

where $\delta(y)$ denotes a point mass at y. This is known as a *hit-miss model* in the ER literature [CH90].

Distortion distribution. We deviate from `blink` in our specification for the prior on the distortion distribution H_{ea}. Specifically, we assume H_{ea} is drawn from a Dirichlet Process:

$$H_{ea} | y_{ea} \overset{\text{ind.}}{\sim} \text{DP}(\rho_a; \psi_a(y_{ea})) \qquad \forall e, a, \tag{4.6}$$

where $\rho_a > 0$ is a concentration hyperparameter and $\psi_a(y_{ea})$ is a base distribution with support on a subset of $\mathcal{D}_a \setminus \{y_{\lambda_i a}\}$.

By treating H_{ea} as an unknown distribution, rather than a hyperparameter as in [Ste15; SHF16; STL18], we are able to improve flexibility. The Dirichlet Process is a natural fit for modelling H_{ea}, as we expect H_{ea} to be concentrated on a relatively small number of alternative values in realistic scenarios. For example, if y_{ea} corresponds to the first name "Carrie", we might expect H_{ea} to be concentrated on common misspellings, such as "Carey", "Karrie", "Kerry", or "Kari". Note that we can recover the distortion distribution used in `blink` by letting $\rho_a \longrightarrow \infty$ and setting the base distribution $\phi_a(y)$ empirically as detailed in (3.2).

The other important difference in our specification of H_{ea} is the exclusion of y_{ea} from the support. We believe this results in a more appropriate formulation of the *hit-miss model* in cases where H_{ea} is atomic. Since the original hit-miss model [CH90] was formulated for non-atomic H_{ea}, the probability of drawing y_{ea} from H_{ea} was zero. By explicitly excluding y_{ea} from the support of H_{ea}, we are able to replicate this behaviour, thereby avoiding illogical cases where the record attribute is distorted ($z_{ia} = 1$) while at the same time matching the non-distorted entity attribute exactly ($x_{ia} = y_{\lambda_i a}$). Apart from the modelling advantages (see Figure 4.3), excluding y_{ea} from the support of H_{ea} also makes inference more tractable, as we are able to collapse H_{ea} (see Section 4.5).

4.4.3 Attribute distance measures

Our proposed distortion model is parameterised by a set of distance measures $\{\text{dist}_a\}$, one for each attribute $a \in \{1, \dots, A\}$. These distance measures are used to encode prior knowledge about the likelihood that a record attribute value x appears as a distorted alternative to an entity attribute value y. The larger the distance $\text{dist}_a(y, x)$, the less likely x is considered to be a distortion of y. Since the likelihood of distorting x to y may not be the same as the likelihood of distorting y to x, we do not require that the distance measures are symmetric. Our assumptions are detailed below.

Definition 4.1 (Attribute distance measure). *Let $\mathcal{D}$ be the domain of an attribute. An attribute distance measure on $\mathcal{D}$ is a function* $\text{dist} : \mathcal{D} \times \mathcal{D} \longrightarrow [0, \infty)$. *The measure is not assumed to be a distance metric: in particular, we do not require* $\text{dist}(y, x) = \text{dist}(x, y)$.

By adopting attribute distance measures, we are deviating from our approach in the previous chapter, where we used a parameterisation in terms of attribute *similarity* measures (see Section 3.3.4). Recall that our motivation for using similarity measures was purely based on computational efficiency. Specifically, we were able to develop an efficient algorithm for updating the entity attributes based on perturbation sampling (see Sections 3.6.2 and 3.6.3). Unfortunately this algorithm is incompatible with the distortion model proposed here, so we revert to using distance measures instead. This is in line with the `blink` model, which used edit distance measures to model typographical errors.

We recommend selecting the attribute distance measures carefully (see Section 2.3), leveraging prior knowledge about the distortion process where possible. Below we discuss two attribute distance measures which are used later in our empirical evaluation.

Constant distance measure. We recommend selecting a constant distance measure $\text{dist}(y, x) \equiv const.$ for categorical attributes. This encodes a prior belief that all values in the domain $x \in \mathcal{D}$ are equally likely as a distorted alternative to the entity attribute

value y. In our empirical study (see Section 4.6), we use a constant distance measure for attributes such as *date of birth* and *sex*.

Hybrid distance measure. When an attribute domain contains medium-length strings with multiple words in each string, we recommend using a hybrid distance measure (see Section 2.3.3). A hybrid distance measure accounts for differences between words (tokens), while allowing for fuzzy matching between words. The measure we describe here draws inspiration from a hybrid similarity measure proposed by Monge and Elkan [ME96] for attribute matching, as defined in (2.3). However, unlike Monge and Elkan we attempt to match the tokens in each string while incorporating penalties for "missing" tokens.

Suppose we would like to compare a pair of multi-token strings x and y. As a running example, we consider x = "University of California, San Diego" and y = "Univ. Calif., San Diego". Given a separator character (e.g. a space), we can map each string to a set of tokens. For example, string x from our running example would be mapped to

$$X = \{\text{“California,”}, \text{“Diego”}, \text{“of”}, \text{“San”}, \text{“University”}\}.$$

Note that we have used capital X to denote the token set[4] representation of string x—a convention we adopt in the remainder of this section. Note that X is a lossy representation of x, as it discards information about the token order. This is desirable for the running example, as permuting the tokens does not significantly change the meaning of the strings.

We propose to measure the distance from x to y via a generalized edit distance on the token sets X and Y. We consider three elementary edit operations:

- *token insertions* where a token b is appended to the input set;
- *token deletions* where a token a is removed from the input set; and
- *token substitutions* where a token a in the input set is replaced by a token $b \neq a$.

Each elementary operation takes an input set Q to an output set Q', which we write as $Q \longrightarrow Q'$, and has an associated cost $c(Q \longrightarrow Q') \geq 0$. We let

$$c(Q \longrightarrow Q') = \begin{cases} d_i \, \mathrm{dist}_{\mathrm{inner}}(\lambda, b), & \text{if } Q = Q' \setminus \{b\} \text{ (insertion)}, \\ d_d \, \mathrm{dist}_{\mathrm{inner}}(a, \lambda), & \text{if } Q \setminus \{a\} = Q' \text{ (deletion)}, \\ d_s \, \mathrm{dist}_{\mathrm{inner}}(a, b), & \text{if } Q \setminus \{a\} = Q' \setminus \{b\} \text{ (substitution)}, \end{cases}$$

where d_i, d_d and d_s are non-negative weights; λ is the null string; and $\mathrm{dist}_{\mathrm{inner}}(\cdot, \cdot)$ is an *inner distance measure* on tokens (strings). We then define the *hybrid distance* between x and y as the minimum average cost of transforming X into Y via a sequence of elementary edit operations $T_{X,Y} = (X \longrightarrow Q_1, Q_1 \longrightarrow Q_2, \ldots, Q_{l-1} \longrightarrow Y)$. Symbolically, we write

$$\mathrm{dist}_{\mathrm{hybrid}}(x, y) = \min_{T_{X,Y}} \frac{1}{|T_{X,Y}|} \sum_{(Q \longrightarrow Q') \in T_{X,Y}} c(Q \longrightarrow Q').$$

[4] Technically we consider a multi-set, since we allow tokens to appear multiple times.

The above cost minimization problem can be solved using an off-the-shelf linear sum assignment problem (LSAP) solver [Cro16]. In order to do so, we need to add null string tokens to X and Y to account for all possible insertion and deletion operations. Concretely, we add $|Y|$ null tokens to X to allow for insertions and $|X|$ null tokens to Y to allow deletions. We then construct a pairwise cost matrix by applying $\text{dist}_{\text{inner}}$ to all pairs of tokens in (the amended) X and Y. The resulting matrix is then passed to the LSAP solver, which returns the optimal set of edit operations and their cost.

Returning to our running example, if we set $\text{dist}_{\text{inner}}$ to the Levenshtein distance, the solution to the LSAP is

$$\{(\text{“University”} \leftrightarrow \text{“Univ.”}, 5), (\text{“of”} \leftrightarrow \lambda, 2), (\text{“California,”} \leftrightarrow \text{“Calif.,”}, 6),$$
$$(\text{“San”} \leftrightarrow \text{“San”}, 0), (\text{“Diego”} \leftrightarrow \text{“Diego”}, 0), (\lambda \leftrightarrow \lambda, 0), (\lambda \leftrightarrow \lambda, 0),$$
$$(\lambda \leftrightarrow \lambda, 0), (\lambda \leftrightarrow \lambda, 0)\}.$$

Hence we conclude that $\text{dist}_{\text{hybrid}}(x, y) = \frac{5+2+6+0+0}{5} = 2.6$. This distance reflects the semantic closeness between x and y better than the Levenshtein distance, which gives a larger value of 14 when evaluated directly on x and y.

4.4.4 Hyperparameter specification

We now make recommendations for the configuration of hyperparameters in our refined model. Most of our recommendations differ from those provided for `blink` [Ste15].

Distortion base distribution. We recommend using the attribute distance measure to set the base distribution $\psi_a(y_{ea})$ for the distortion distribution introduced in (4.6). Specifically, we recommend a softmax distribution

$$\psi_a(x|y_{ea}) \propto \mathbb{I}[x \neq y_{ea}] \exp(-\text{dist}_a(y_{ea}, x)), \tag{4.7}$$

where the temperature parameter is absorbed in the definition of the distance measure, and the indicator function excludes y_{ea} from the support. This puts more weight on values in the domain closer to y_{ea} and less weight on values further away. Unlike `blink`, we do not include a factor proportional to the empirical frequency of x in the observed records. This is because we don't expect the empirical frequency across all values (distorted and non-distorted) to accurately reflect the frequency of the distorted values, which are typically rare (e.g. typographical errors). For a categorical attribute with $\text{dist}_a(y, x) \equiv 0$, (4.7) reduces to the uniform distribution. In this case, it may make sense to incorporate the empirical frequencies. This can be done by setting

$$\psi_a(x|y_{ea}) \propto \mathbb{I}[x \neq y_{ea}] \sum_{i=1}^{N} \mathbb{I}[x_{ia} = x].$$

Other hyperparameters. In the absence of prior knowledge, we recommend choosing hyperparameters that yield vague or non-informative priors. For example, one can set the beta priors to be uniform, by setting $\beta_{sa}^{(0)} = \beta_{sa}^{(1)} = 1$ for all s, a, and $\zeta^{(0)} = \zeta^{(1)} = 1$. To obtain a vague Gamma prior, we recommend setting the shape parameter $\chi^{(0)} = 1$ and the rate parameter $\chi^{(1)}$ to be small (e.g. 10^{-2}). For the shifted negative binomial prior,

we obtain a vague prior by setting $r \approx N$ and v to be small (e.g. 10^{-4}). For the prior on the entity attribute values, we recommend setting $v_a = 1$ and using a uniform base distribution $\phi_a = \left[|\mathcal{D}_a|^{-1}, \ldots, |\mathcal{D}_a|^{-1}\right]$ for all a. Finally, for the concentration parameters ρ_a associated with the distortion distributions, we recommend setting $\rho_a = 1$ for all a. If the distorted alternatives for attribute a are expected to be more (or less) concentrated ρ_a can be reduced (or increased).

4.5 Inference

Now that we have proposed a refined ER model, we turn to the problem of designing an efficient method for inference. In the previous chapter, we performed approximate inference using a Markov chain Monte Carlo (MCMC) algorithm based on partially-collapsed Gibbs (PCG) sampling [DP08]. We take the same approach here, however we are required to make significant changes due to the increased complexity of the new model compared to `blink` (and `d-blink`).

Recall that PCG sampling allows for groups of variables to be updated jointly (known as *marginalisation*), while also allowing variables within groups to be collapsed (known as *trimming*). Generally it is desirable to perform as much marginalisation and trimming as possible, in order to reduce autocorrelation of the Markov chain and increase the rate of convergence to equilibrium. However, this desire must be balanced with computational and mathematical constraints. In our proposed sampling scheme, we *fully collapse* the mixing proportions π and the distortion distributions H_{ea}, and *partially-collapse* the distortion indicators z_{ia} in a joint update for the entity attributes. By collapsing the mixing proportions, we obtain an urn-based scheme for updating the linkage structure, which is related to urn-based schemes used in the context of non-parametric mixture models [Nea00].[5]

Since this chapter is primarily focused on modelling, rather than inference and sampling algorithms, we provide technical details for the PCG sampler in Appendix B. However, we highlight some of the key design considerations in the remainder of this section.

4.5.1 Nonconjugacy

While we attempted to maintain conjugacy in our refined model, we were unable to avoid nonconjugate priors in some cases. This complicates inference, as the posterior conditional distributions used in Gibbs sampling are no longer of a standard form. There are several well-established methods for dealing with nonconjugacy, including Metropolis-Hastings algorithms [CG95], rejection sampling [GW92] and auxiliary variable methods [DWW99]. We opt to use auxiliary variable methods owing to their simplicity, as there is no need to design proposals or monitor acceptance rates.

There are two sets of parameters in our refined model where non-conjugacy is an issue. The first are the distortion probabilities θ_{sa} defined in (4.3), where the incorporation of the distortion propensities ω_{ia} breaks the conjugacy of the beta prior. We propose

[5] While collapsing the mixing proportions improves statistical efficiency, it introduces dependencies between the entity assignments $\lambda = (\lambda_1, \ldots, \lambda_N)$. This makes it incompatible with the distributed approach considered in Chapter 3.

an auxiliary variable sampling scheme to update θ_{sa} in Appendix B.1. The second set of problematic parameters are the EP parameters: α and σ, or κ and m depending on the regime. We use an auxiliary variable scheme proposed by Teh [Teh06], to update α and σ under a gamma and beta prior, as summarised in Appendix B.4.1. We extend this idea to design an auxiliary variable update for κ and m under a gamma and shifted negative binomial prior in Appendix B.4.2.

4.5.2 Collapsing the distortion indicators

When designing a PCG sampler for `d-blink` in the previous chapter, we opted to jointly update the entity attribute y_{ea} and linked distortion indicators $\{z_{ia}\}_{i:\lambda_i=e}$, while collapsing the linked distortion indicators. This yielded a significant improvement in statistical efficiency in our empirical study (see results for PCG-I in Section 3.7.3). Fortunately, we are able to apply the same idea to our refined model.

The posterior factors involving z_{ia} factorise over i and a, so that collapsing z_{ia} yields:

$$\begin{aligned} P(x_{ia}|\theta_{s_ia}, \omega_{ia}, y_{\lambda_i a}, H_{\lambda_i a}) &\propto \sum_{z_{ia}=0}^{1} P(x_{ia}|z_{ia}, y_{\lambda_i a}, H_{\lambda_i a}) P(z_{ia}|\theta_{s_ia}, \omega_{ia}) \\ &\propto (1 - \theta_{s_ia}\omega_{ia})\mathbb{I}[x_{ia} = y_{\lambda_i a}] + \theta_{s_ia}\omega_{ia} H_{\lambda_i a}(x_{ia}). \end{aligned} \tag{4.8}$$

This result is used to derive a partially-collapsed joint update for the entity attribute y_{ea}, linked distortion indicators $\{z_{ia}\}_{i:\lambda_i=e}$, and distortion distribution H_{ea} in Appendix B.2. While it is also possible to partially-collapse the distortion indicators in a joint update for the linkage structure $\boldsymbol{\Lambda}$, we opt not to do so, since conditioning on the distortion indicators allows us to reduce the computational complexity via indexing (see Appendix B.3). This trade-off between computational and statistical efficiency was studied empirically in our experiments with `d-blink` (see results for PCG-II in Section 3.7.3).

4.5.3 Collapsing the distortion distributions

So long as the entity attribute y_{ea} is excluded from the support of the distortion distribution H_{ea}, we are able to fully collapse H_{ea} by relying on conjugacy. The posterior factors which involve the distortion distribution H_{ea} are as follows:[6]

$$P(H_{ea}|y_{ea}) \prod_{i:\lambda_i=e} P(x_{ia}|z_{ia}, H_{ea}, y_{ea}).$$

Since we assumed H_{ea} does not contain y_{ea} in its support, we can rewrite the above expression as:

$$P(H_{ea}|y_{ea}) \prod_{\substack{i:\lambda_i=e \\ x_{ia}\neq y_{ea}}} H_{ea}(x_{ia}) \prod_{\substack{i:\lambda_i=e \\ x_{ia}=y_{ea}}} (1 - z_{ia}) \prod_{\substack{i:\lambda_i=e \\ x_{ia}\neq y_{ea}}} z_{ia}. \tag{4.9}$$

In doing so, we have isolated the factors involving $H_{ea}(x_{ia})$, which means the first line in the above display is proportional to a Dirichlet-Multinomial likelihood. This relies on the fact that the Dirichlet Process can be expressed as a Dirichlet distribution when the base distribution is atomic [BH10], viz. $\mathrm{DP}(\rho_a; \psi_a(y_{ea})) = \mathrm{Dirichlet}(\rho_a \psi_a(y_{ea}))$.

[6] If z_{ia} is collapsed as in (4.8), the following results can be adapted by making the replacement $z_{ia} \longrightarrow \theta_{s_ia}\omega_{ia}$.

Thus we can marginalise out H_{ea} in (4.9) to obtain the expression

$$\frac{n_{ea}^{\neg}(y_{ea})\mathrm{B}(\rho_a; n_{ea}^{\neg}(y_{ea}))}{\prod_{v \in \mathcal{D}_{ea} \setminus \{y_{ea}\}} n_{ea}(v)\mathrm{B}(\rho_a \psi_a(v|y_{ea}); n_{ea}(v))} \times \prod_{\substack{i:\lambda_i=e \\ x_{ia}=y_{ea}}} (1 - z_{ia}) \prod_{\substack{i:\lambda_i=e \\ x_{ia}\neq y_{ea}}} z_{ia},$$

where B is the Beta function, $n_{ea}(v) = \sum_{i:\lambda_i=e} \mathbb{I}[x_{ia} = v]$ and $n_{ea}^{\neg}(v) = \sum_{i:\lambda_i=e} \mathbb{I}[x_{ia} \neq v]$. This result is used in Appendices B.3 and B.2 to derive updates for the linkage structure and entity attributes.

4.5.4 Computational considerations

We now discuss considerations for improving the computational complexity of our sampling scheme. The main bottleneck is the update for the linkage structure which scales naïvely as $O(N \cdot E)$ where E is the number of instantiated entities. The update for the entity attributes may also be problematic for large domains $\mathcal{D}_a$ as it scales as $O(E \cdot |\mathcal{D}_a|)$ for the a-th attribute.

We are able to reduce the computational complexity of the linkage structure update by exploiting constraints imposed by the distortion model. Close inspection of the update for the entity linked to record i (see Appendix B.3) reveals that some entities can be immediately excluded from consideration. Specifically, only those entities whose attributes match the corresponding *non-distorted* record attributes (x_{ia} with $z_{ia} = 0$) may be linked to record i. In order to efficiently query this set of entities, we maintain inverted indexes that map an attribute value $x \in \mathcal{D}_a$ to the set of entities instantiated with that value $\{e : x = y_{ea}\}$. This approach is considerably more efficient than iterating over all entities sequentially, so long as the level of distortion is relatively low. However it is important to note that it relies crucially on *not* collapsing the distortion indicators, as explained in Section 3.6.1.

To improve the complexity of the entity attribute update, we impose a cut-off on the distance measures.[7] Concretely, for attribute a we replace the "raw" distance measure dist_a by

$$\underline{\mathrm{dist}}_a(y, x) = \begin{cases} \mathrm{dist}_a(y, x), & \text{if } \mathrm{dist}_a(y, x) \leq d_a^{(\mathrm{cut})}, \\ \infty, & \text{otherwise,} \end{cases}$$

where $d_a^{(\mathrm{cut})} \in (0, \infty)$ is a configurable cut-off. This approximation eliminates the need to consider unlikely distortions from entity attribute y to record attribute x, for which $\mathrm{dist}(y, x) > d_a^{(\mathrm{cut})}$. It plays a similar role to blocking in the record linkage literature [Ste+14] and resembles the approach proposed for `d-blink` in Section 3.6.2. In order to make use of this approximation, we must maintain indices from record attribute values $x \in \mathcal{D}_a$ to entity attribute values which fall below the cut-off $\{y \in \mathcal{D}_a | \mathrm{dist}_a(y, x) \leq d_a^{(\mathrm{cut})}\}$.

4.6 Empirical evaluation

We conduct an empirical evaluation using four ER data sets from different domains. The data sets are introduced in Section 4.6.1 and we provide details of the experimental setup

[7] It is not possible to apply perturbation sampling as in Section 3.6.3 due to the more complex form of the update for y_{ea} (see Appendix B.2).

Table 4.2: Summary of data sets used for the empirical evaluation.

Data set	Entity type	# records (N)	# entities	# attributes (A) categorical	string
`RLdata`	People	10,000	9,000	2	3
`nltcs`	People	5,359	3,307	5	0
`cora`	Citations	1,295	125	0	4
`rest`	Restaurants	864	752	0	4

in Section 4.6.2. In Section 4.6.3, we attempt to isolate the effects of the proposed changes to the linkage structure and distortion model by performing an exhaustive comparison. Finally, in Section 4.6.4 we compare our refined model with the original `blink` model and an extension of the Fellegi-Sunter model proposed by Sadinle [Sad14].

4.6.1 Data sets

Table 4.2 provides a summary of the ER data sets used in our experiments. All data sets come with ground truth entity identifiers, which are used to evaluate the quality of ER predictions. The identifiers are not used during inference, which is entirely unsupervised. We do not consider some of the larger data sets studied in the previous chapter, as our approach to inference in this chapter is less scalable (we do not apply blocking and distributed inference). Below, we provide more detailed information about each data set and the attributes used for ER.

- `RLdata` is an artificial person data set included with the RecordLinkage R package [SB10]. We previously used this data set to evaluate `d-blink`, where it was referred to as `RLdata10000`. We model `fname_c1` and `lname_c1` (first and last name) as string attributes using the normalised Levenshtein distance measure. The attributes related to date of birth—`bd`, `bm` and `by`—are modelled as categorical attributes with a constant distance measure.

- `nltcs` is derived from the National Long Term Care Survey [Man10]. It includes respondent records across the 1982, 1989 & 1994 waves from the U.S. state of Alabama. We note that this version of the data set is smaller than the one used to evaluate `d-blink`, as it excludes records from other states. We model all of the available attributes—`DOB_YEAR`, `DOB_MONTH`, `DOB_DAY`, `REGOFF` and `SEX`—as categorical with a constant distance measure.

- `cora` is a collection of computer science citation records hosted on the RIDDLE repository [Bil]. It is significantly "dirtier" than the above two data sets, and is expected to present a challenge for our ER model. As a pre-processing step, we separated hyphenated words and removed punctuation. We also corrected several erroneous ground truth labels. The `title`, `venue` and `authors` attributes generally contain multiple words with semantic and character-level variations, and are therefore modelled using the hybrid distance measure introduced in Section 4.4.3. The `year` attribute is modelled using normalised Levenshtein distance.

- `rest` is a collection of restaurant records from the Fodor and Zagat restaurant guides hosted on the RIDDLE repository [Bil]. It is not as "dirty" as `cora`, but expected to be more challenging than `RLdata` and `nltcs` owing to semantic variations. We applied the same pre-processing steps as for `cora`. The `name` and `addr` attributes generally contain multiple words and are therefore modelled using the same hybrid distance measure as for `cora`. The `city` and `type` (cuisine) attributes are modelled as categorical with a constant distance measure.

4.6.2 Experimental setup

Implementation and hardware. All experiments are conducted in R version 3.4.4, running on a local server fitted with two 28-core Intel Xeon Platinum 8180M CPUs and 12 TB of RAM.[8] Our refined model and the `blink` model [Ste15] are implemented in an open-source R package called `exchanger`[9]. We also developed our own implementation of the model proposed by Sadinle [Sad14] in an open-source R package called `BDD`[10], since there is no publicly-available implementation. For efficiency reasons, inference algorithms for all models are implemented in C++ using the Rcpp interface [EF11].

Hyperparameter settings. We generally follow the recommendations in Section 4.4.4 for setting vague/non-informative priors. However, in order to encode a slight bias towards low precision modes, we replace the recommended uniform prior on the distortion probability with a weakly-skewed prior by setting $\beta^{(0)} = 1$ and $\beta^{(1)} = 4$. This corresponds to a prior mean distortion probability of 20% with a standard deviation of 16%. We set $\rho_a = 1$ and $\upsilon_a = 1$ for most attributes with non-constant distance measures, in order to encourage concentration of the distortion distribution and entity attribute distribution. However for categorical attributes (with constant distance measures), we set $\rho_a \longrightarrow \infty$ and $\upsilon_a \longrightarrow \infty$ as we do not expect the distributions to be concentrated.

When setting hyperparameters for the two baseline models, we attempt to follow the recommendations of the authors. For `blink`, we set $m = N$ for the coupon-collector's prior and $\beta_{sa}^{(0)} = N/1000$ and $\beta_{sa}^{(1)} = N/10$ for the Beta prior on the distortion probabilities (here N is the total number of records). For the model by [Sad14], we set the agreement levels by inspecting the distribution of distances for each attribute. We use uniform priors on the $m^\star$ and $u^\star$ probabilities, as suggested by the author. We do not truncate the priors on the $m^\star$ probabilities, as we encountered convergence issues and observed a high degree of sensitivity to the truncation points.

Initialisation and MCMC. We use the same initialisation as for `d-blink`—linking each record to a unique entity and copying the record attributes into the entity attributes, assuming no distortion. The Ewens-Pitman parameters are initialised using the prior mean. For our model and `blink`, we run Markov chain Monte Carlo (MCMC) for 10^5 iterations, discarding the first 10^4 iterations as burn-in, and applying thinning with an interval of 10.[11] This yields 9000 approximate posterior samples. Since the model by

[8] R scripts are published at github.com/cleanzr/exchanger-experiments.
[9] Source code published at github.com/cleanzr/exchanger.
[10] Source code published at github.com/cleanzr/BDD.
[11] The chain is slower to converge for the `cora` data set, so we increase the burn-in interval to 5×10^4.

Sadinle [Sad14] is less complex, we generally observe more rapid convergence. We therefore collect 1000 approximate posterior samples using a thinning interval of 10 and a shorter burn-in interval of 1000.

4.6.3 Effects of the proposed changes

We are interested in studying the effect of our proposed changes to the linkage structure prior and distortion model in isolation. By doing so, we hope to determine whether each change is beneficial in its own right, and/or whether one change is more beneficial than the other. We consider two distortion models:

- Ours: the refined distortion model presented in Section 4.4.2; and
- `blink`: the distortion model used in `blink` and `d-blink`;

and four linkage structure priors:

- PY: a Pitman-Yor partition with $\sigma \in (0, 1)$ and hyperpriors on σ, α as detailed in (4.2);
- Ewens: a Ewens partition with $\sigma = 0$ and a hyperprior on α as detailed in (4.2);
- GenCoupon: a generalised coupon partition with $\sigma = -\kappa < 0$ and hyperpriors on κ, m as detailed in (4.1); and
- Coupon: coupon collector's partition used in `blink` and `d-blink`, which is an instance of the generalised coupon partition with fixed parameters ($\kappa \longrightarrow \infty$ and $m = N$).

Thus we have eight model variants to test on each data set. Pairwise evaluation measures for each model variant and data set are presented in Figure 4.2.

Distortion model. Referring to the evaluation results in Figure 4.2, we see that our refined distortion model achieves the highest F1 score for all but one of the data sets and linkage structure priors. The exception is for `cora` under the Coupon linkage structure prior, where the `blink` distortion model has a slight edge. The improvement for our refined distortion model is most pronounced for `RLdata`, where we see a gain of ~0.5 in the F1 score. We observe very little difference in the results for `nltcs`, which is to be expected since the distortion models are similar for categorical attributes with small domains.

We can gain further insight into the differences between the two distortion models by examining the inferred level of distortion under each model, as depicted in Figure 4.3. In general, the `blink` model has a tendency to enter a high distortion mode, particularly for any attributes with a non-constant distance measure. For example, all of the attributes in `cora` are predicted to be extremely distorted, at a level greater than 90%. The same is true for the two name attributes in `RLdata`, which are modelled with non-constant distance measures. When the model enters a high distortion mode, it has a tendency to over-link, thereby resulting in lower precision.

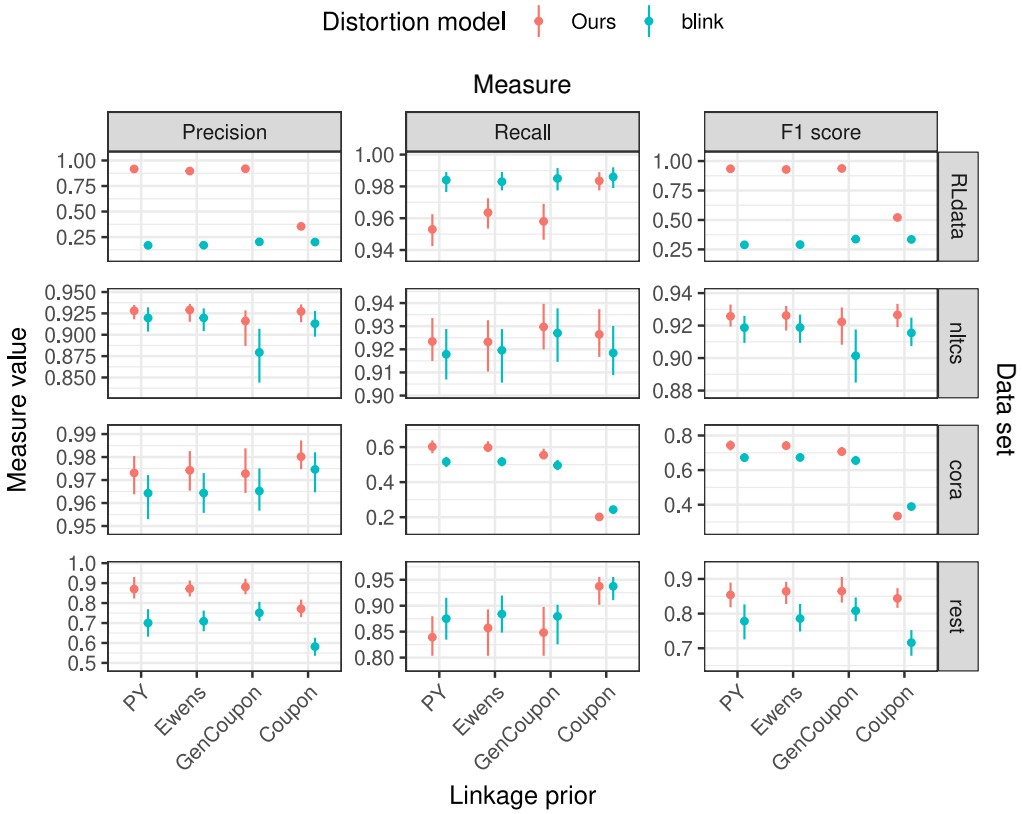


Figure 4.2: Evaluation of ER quality as a function of the linkage structure prior (plotted on the x-axis) and distortion model (indicated by the line colour). Three pairwise evaluation measures are shown (grouped by column) for four data sets (grouped by row). 95% Bayesian credible intervals are shown.

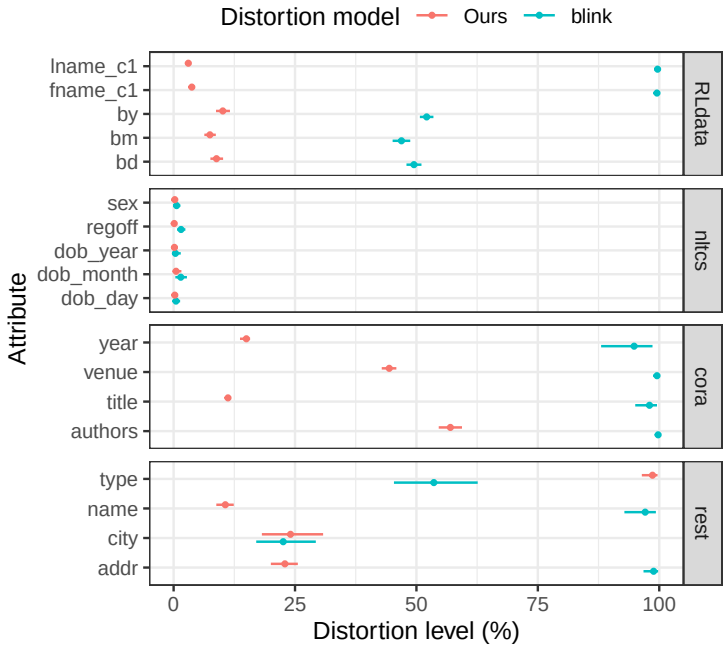


Figure 4.3: Comparison of the posterior attribute-level distortion for two distortion models (a) and (b). The distortion levels are shown for each linkage structure prior (indicated by the line colour), attribute (labelled on the y-axis) and data set (grouped by panel). 95% equi-tailed credible intervals are shown.

Our refined distortion model does not appear to suffer from this problem. It provides estimates of the distortion level consistent with our expectations. Specifically, we expect `RLdata` and `nltcs` to exhibit low levels of distortion as they are relatively clean. On the other hand, we expect `cora` and `rest` to exhibit relatively high levels of distortion as they contain records from different sources, and there is obvious variation in the way attribute values are represented semantically.

Linkage structure prior. We now examine the performance of the linkage structure priors based on the results in Figure 4.2. Assuming our refined distortion model is used, we find that the Coupon linkage structure prior achieves the lowest (or equal-lowest) F1 score among all linkage structure priors for all data sets. The difference in performance is significant for `cora` and `RLdata`, and insignificant for `rest` and `nltcs`. We expect the good performance on `nltcs` is a coincidence, as the Coupon linkage structure prior specifies an expected population size of 3,387 which happens to be very close to the true value of 3,307 (see Table 4.2).

We observe little difference in performance between the three EP parameter regimes considered (PY, Ewens and GenCoupon). This is an interesting result, as each regime is known to exhibit different asymptotic behaviour, as discussed in Section 4.3. In the non-asymptotic (small N) regime, it appears as if all three regimes are expressive enough when vague hyperpriors are used. Figure 4.4 is consistent with this argument—we see vastly different values of the EP parameters are selected for each data set, which is facilitated by the vague hyperpriors.

For another perspective on the quality of fit for the linkage structure priors, we can

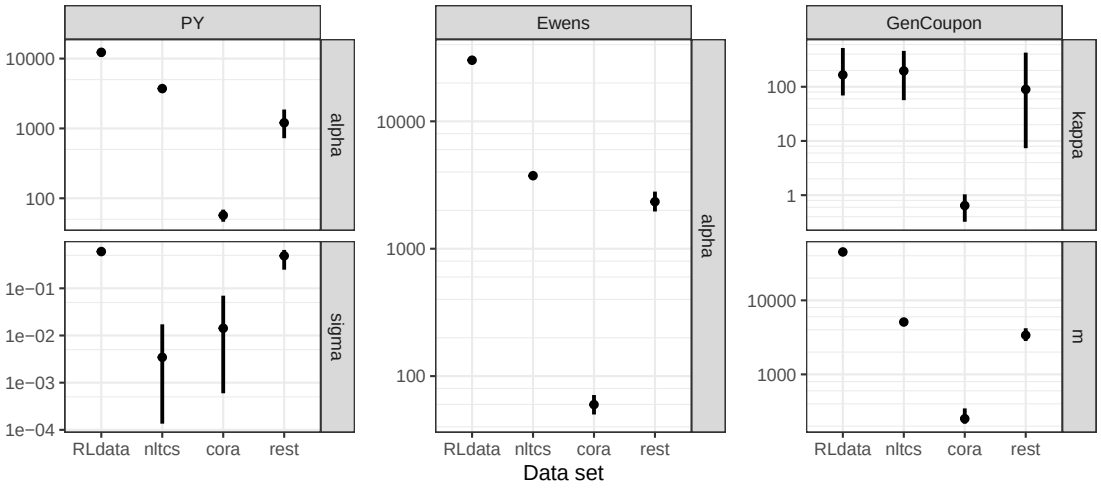


Figure 4.4: Posterior Ewens-Pitman parameters for three parameter regimes (grouped by column) under our refined distortion model. Note that the parameter values are presented on a log-scale. 95% equi-tailed credible intervals are shown.

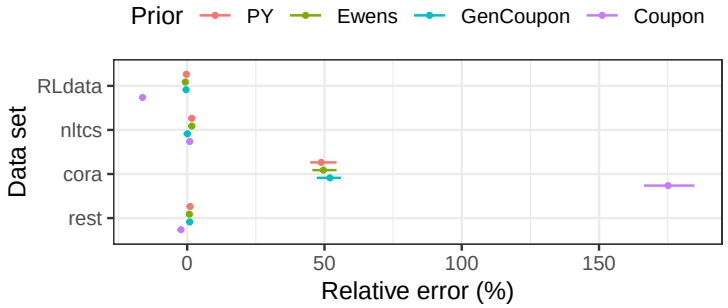


Figure 4.5: Posterior error in the number of entities for all data sets (plotted on the y-axis) and linkage structure priors (indicated by the line colour). 95% equi-tailed credible intervals are shown.

examine the posterior error in the number of entities present in the data, as shown in Figure 4.5. The relative error for `RLdata`, `rest` and `cora` is low for all three of the EP parameter regimes, indicating a good fit. However, there is systematic bias in the estimate for `cora`. We expect the bias can be attributed to the distortion model, rather than the linkage structure prior, as it is not well-suited to data sets with high levels of semantic heterogeneity.

4.6.4 Comparison with baseline models

In this section, we compare our refined model with two unsupervised baselines:

- `blink` by Steorts [Ste15]. Since the supplied implementation does not support custom distance measures and is not optimised for performance, we use our own implementation in the `exchanger` R package. We note that the original `blink` model is *not equivalent* to the model considered in the previous section with the Coupon linkage structure prior and `blink` distortion model. In the previous section,

Table 4.3: Posterior performance of our model against two baselines: `blink` [Ste15] and Sadinle [Sad14]. A point estimate for each performance measure is reported based on the median, along with a 95% equi-tailed credible interval.

		Performance measure		
Data set	Model	Precision	Recall	F1 score
RLdata	Ours	0.917 (0.902,0.932)	0.957 (0.945,0.967)	0.937 (0.927,0.944)
	`blink`	0.336 (0.329,0.346)	0.992 (0.988,0.996)	0.503 (0.494,0.513)
	Sadinle	0.045 (0.045,0.045)	0.915 (0.909,0.918)	0.086 (0.086,0.086)
nltcs	Ours	0.913 (0.888,0.929)	0.930 (0.919,0.940)	0.922 (0.907,0.930)
	`blink`	0.903 (0.889,0.917)	0.917 (0.907,0.926)	0.910 (0.899,0.920)
	Sadinle	0.036 (0.036,0.036)	0.952 (0.948,0.954)	0.070 (0.070,0.070)
cora	Ours	0.974 (0.965,0.985)	0.556 (0.517,0.591)	0.708 (0.675,0.735)
	`blink`	0.981 (0.974,0.988)	0.211 (0.200,0.221)	0.348 (0.332,0.361)
	Sadinle	0.981 (0.981,0.984)	0.584 (0.583,0.585)	0.732 (0.731,0.734)
rest	Ours	0.884 (0.842,0.932)	0.848 (0.804,0.893)	0.868 (0.833,0.903)
	`blink`	0.632 (0.588,0.677)	0.920 (0.893,0.946)	0.747 (0.714,0.781)
	Sadinle	0.752 (0.752,0.752)	0.812 (0.812,0.812)	0.781 (0.781,0.781)

we used a Dirichlet Process prior on the entity attribute distribution, while the original `blink` model treats the entity attribute distribution as a fixed parameter. There are also differences in the hyperparameter settings. In the previous section we used a vague prior on the distortion probabilities (see Section 4.6.2), while an informative prior is recommended for `blink` (see Section 3.7.2). We also follow the original recommendations for setting the distortion distribution, instead of those presented in Section 4.4.4.

- Sadinle by Sadinle [Sad14]. This model can be viewed as an extension of the Fellegi-Sunter model [FS69], which incorporates multiple levels of attribute agreement and ensures transitivity of the linkage structure. Unlike `blink` and our proposed model, it is not generative, and instead operates on pairwise attribute comparison data. It uses a uniform prior on the linkage structure, which is expected to be less flexible than our proposed priors. In contrast to our model and `blink`, Sadinle requires blocking in order to run efficiently. We use generous blocking rules in order to minimize the number of false negatives attribute to blocking. Since an implementation of the model is not publicly available, we use our own implementation in the BDD R package.

Table 4.3 presents pairwise performance measures for each model and data set. Our proposed model (with the GenCoupon linkage structure prior) achieves the highest (or equal-highest) F1 score within the credible intervals for all data sets. While Sadinle achieves competitive F1 scores on `cora` and `rest`, it's performance is otherwise poor due to over-linkage. The `blink` model is also inconsistent, achieving good results on `nltcs` and `rest`, but poor results otherwise.

We have already explained the likely reasons for the poorer performance of `blink`, which can be attributed to lack of flexibility and misspecified assumptions in the distortion model. The Sadinle model also uses a less flexible linkage structure prior, which

may partially explain the poorer performance in some cases. However, we expect the priors on the m- and u-probabilities are a more significant cause for concern. These are set to be uniform as recommended by the author, however this appears to cause difficulty in separating the two mixture components: the matches and non-matches. The author recommends truncating the priors on the $m^\star$ probabilities from below as a possible solution, however it is unclear how the truncation points should be set—even in a subjective way. When experimenting with truncation points, we observed a high degree of sensitivity and bimodal behaviour in the posterior distribution for some data sets, hence we opted to present results without truncation.

4.7 Concluding remarks

Bayesian models provide a natural framework for making predictions under uncertainty, and are therefore an attractive solution for solving entity resolution tasks. However, care must be taken when designing models, in order to minimise error and ensure solutions are robust to misspecified priors. In this chapter, we identified and addressed several potential issues with the `blink` ER model [Ste15]. Our proposed changes focused on improving flexibility of the model, by placing priors on parameters that were previously held fixed. We also considered a broader class of priors on the linkage structure and made corrections to logic in the distortion model. In proposing these changes, we attempted to balance model complexity and expressiveness with computational tractability and efficiency.

We tested the effect of our proposed changes in an empirical study using four ER data sets from different domains. The results showed that our changes were well-motivated. The refined distortion model and flexible linkage structure priors both contributed individually to reductions in error rates, and were found to be most effective in combination. We believe the corrected distortion model was arguably the most important change, as it addressed the tendency of `blink` to enter a pathological state characterised by high distortion. Another notable conclusion was the relative insensitivity of our model to different linkage structure priors from the class of Ewens-Pitman (EP) random partitions. Despite the fact that EP partitions are known to exhibit different asymptotic behaviour depending on the parameter regime, we found minimal differences in the quality of fit for the three parameter regimes tested. We believe this is due to our use of hyperpriors on the EP parameters, which improve flexibility.

There are several interesting directions for future work. Although the linkage structure priors we tested seemed to perform well, we were unable to test the quality of fit for large data sets closer to the asymptotic regime. Future work could examine this regime and explore the use of microclustering priors [Mil+15] as an alternative. Another direction could involve improvements to the distortion model, as we observed reductions in performance for noisy heterogeneous data. For instance, more flexible alternatives could be proposed for the distortion distribution—e.g. which leverage a weighted combination of distance measures. Techniques from natural language processing could potentially be used to incorporate semantic understanding. Lastly, we note that scalability is important in practice. Our proposed approach to inference is limited to tens of thousands of records, however we could improve scalability by adapting ideas from the previous chapter, such as integrated blocking and distributed inference.

Chapter 5

A theoretical framework for label-efficient evaluation

Entity resolution (ER) presents unique challenges for evaluation methodology. In practical settings, there may be significant uncertainty about the accuracy of an ER system when it is applied to previously unseen data. While accuracy can be assessed by evaluating against ground truth labels, the quantity of labelled examples required is often excessive due to severe class imbalance. Moreover, the importance of the unsupervised ER approaches examined in Chapters 3 and 4 is motivated by a general lack of ground truth labels in many application settings. In this chapter, we develop a statistically-grounded framework for evaluating ER systems based on adaptive importance sampling. In contrast to standard passive or ad-hoc approaches, our framework adaptively biases the selection of items to label, while correcting for the bias. This can significantly reduce the amount of labels required to yield a precise estimate of performance. Since adaptivity breaks data independence assumptions, we establish theoretical results which ensure that estimates produced by our framework converge to the population performance measure. These results hold under verifiable conditions on the performance measure and adaptive labelling policy. They also permit us to study the asymptotically-optimal labelling policy, which provably minimises the variance of the estimated performance measure. This policy is used as a basis for designing practical algorithms in Chapter 6.

5.1 Introduction

Evaluation plays a crucial role in any entity resolution (ER) workflow. There is always a risk that ER may fail to yield accurate results, regardless of how straightforward the task may seem or how advanced the methods may be. Inaccurate results may occur when the ER system is poorly configured, when the methods are ill-suited for the data, or when the data is particularly challenging to resolve. If a poorly-performing ER system goes

This chapter incorporates material from the following publication:

N. G. Marchant and B. I. P. Rubinstein. "Needle in a Haystack: Label-Efficient Evaluation under Extreme Class Imbalance". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '21. Virtual Event, Singapore: ACM, 2021. DOI: 10.1145/3447548.3467435. Accepted.

undetected, it may have an adverse impact on downstream applications, including: poor user experience [NRG12], lost productivity [Ver+00] and biased downstream statistical analyses [Har+14]. It is therefore imperative that ER systems are evaluated in a rigorous manner to accurately assess their performance. This should ideally occur prior to deployment, and periodically during a system's lifecycle for dynamic applications.

From a statistical perspective, evaluation can be formulated as a population estimation problem. The goal is to estimate performance measures, such as precision and recall, with respect to the complete data or data generating process (the "population"). For pairwise performance measures, this is done using a sample of records pairs, which are labelled as *matches* (referring to the same entity) or *non-matches* (referring to distinct entities). While unlabelled data is plentiful in ER applications, labels must typically be acquired from humans—e.g. by employing expert annotators or through crowdsourcing.

The cost of labelling is a major challenge for ER, as the amount of labelled data required to achieve precise performance estimates is typically very large [Kas+19]. This is due to severe class imbalance between matches and non-matches, which may be as high as $1 : N$ when performing ER on two data sources with N records each. As a result, standard unbiased evaluation methods may require N samples on average before a single pair of matching records is found. This motivates the study of label-efficient evaluation methods for ER, which reduce the variance of performance estimates by biasing the selection of items to label [Saw+10].

Label-efficient evaluation methods have not been studied in the ER literature, however there has been some work in a general machine learning context. Most existing approaches are based on stratified sampling [BC10; DM11; Gao+19] or importance sampling [SLS10; Sch+16], which are both well-established variance reduction methods [RK16]. However, existing approaches suffer from one or more of the following limitations:

- lack of support for a broad range of performance measures;
- lack of support for estimating multiple performance measures in parallel;
- lack of support for interactive evaluation—where previously acquired labels are used to inform the selection of future instances to label; and
- limited effectiveness at improving label efficiency.

In this chapter, we address these limitations by proposing a framework for label-efficient evaluation based on *adaptive importance sampling (AIS)* [Bug+17]. Our framework encompasses a broader class of performance measures than have previously been considered in the literature, which includes non-linear transformations of vector-valued risk functionals. Moreover, it is the first framework of its kind to incorporate AIS, which is known to be one of the most effective variance reduction methods available [RK16].

Although AIS can effectively reduce labelling requirements, the adaptivity breaks data independence assumptions which are commonly used to obtain theoretical guarantees. Thus, an important goal of this chapter is to establish asymptotic results, including strong consistency and a central limit theorem. Strong consistency ensures that performance estimates converge to the population performance asymptotically, and the central limit theorem is useful for assessing asymptotic efficiency and computing approximate confidence intervals.

An important component of our framework is the *adaptive labelling policy*, also known as the proposal or instrumental distribution in the AIS literature. This policy is responsible for selecting instances to label and is adapted based on the the incoming labels. Throughout this chapter, we leave the adaptive labelling policy unspecified and obtain theoretical results under general conditions. In Section 5.7 we derive the asymptotically-optimal policy, which minimises the asymptotic variance of the performance estimates. This policy cannot be used directly as it depends on the unknown oracle response, however we us it as a guide for developing practical policies in Chapter 6.

While label-efficient evaluation of ER is the primary motivation of this chapter, our proposed framework is more broadly applicable to evaluation in other domains. We therefore present the framework in a generic setting, assuming that the goal is to evaluate any collection of systems that produce some output.

Chapter outline. We discuss related work next in Section 5.2. In Section 5.3, we formulate the evaluation problem and define a class of performance measures called *generalised risks* that correspond to transformations of vector-valued risk functionals. In Section 5.4 we show that conventional methods for estimating generalised risks can be grossly inefficient in some cases. Then, in Section 5.5 we propose a framework for estimating generalised risks based on adaptive importance sampling. We analyse the asymptotic behaviour of estimates from our framework in Section 5.6, and derive the asymptotically-optimal labelling policy in Section 5.7. We discuss practicalities in Section 5.8, and summarise our contributions in Section 5.9.

5.2 Related work

Existing approaches to label-efficient evaluation largely fall into three categories: model-based [WWP13], stratified sampling [BC10; DM11] and importance sampling [SLS10; Sch+16].

The model-based approach proposed by Welinder et al. [WWP13] is designed to estimate precision-recall curves for binary classifiers. It assumes the joint distribution of scores and labels is well-approximated by a two-component mixture model, where the components are standard parametric distributions. While this assumption can improve efficiency, it is not guaranteed to hold in practice, and may yield biased estimates of performance. Severe class imbalance may also pose a problem for this approach, as instances are selected uniformly at random for labelling.

Stratified sampling has been used to estimate scalar performance measures such as precision, accuracy and F1-score [BC10; DM11]. Under this approach, the test instances are partitioned into strata (blocks) such that the within-stratum variance of some variable of interest (the stratification variable) is likely to be small. Instances are then sampled from the strata for labelling, in a possibly biased manner. In [BC10] and [DM11] the *optimal allocation principle* [Coc77] is used to sample items in proportion to the size of the stratum and the within-stratum standard deviation of the stratification variable. However, this principle is merely a heuristic, as it does not explain how to select the stratification variable. Furthermore, the stratification variable may depend on the unknown labels. Druck and McCallum [DM11] leave the choice of stratification variable to the user, while

supporting a variety of performance measures. Bennett and Carvalho [BC10] propose a specialised method for estimating precision.

Sawade et al. [SLS10] consider the problem of estimating generalised F-measures (which includes the F1 score, precision and recall) using importance sampling. Their approach is similar to ours, in that it frames evaluation as a Monte Carlo estimation problem. However their approach is more limited overall: it is non-adaptive, it only supports evaluation of a single scalar performance measure, and the class of performance measures considered is less general.

Novel evaluation methods have also been studied in the information retrieval (IR) community (see survey [Kan16]). Some evaluation tasks in the IR setting can often be cast as standard prediction problems, by treating query-document pairs as features (inputs) and relevance judgements as labels (outputs). Early approaches for evaluating IR systems were not statistically rigorous, and used relevance scores from the system to ignore irrelevant documents [CPC98]. This comes with a risk of producing overly-optimistic performance estimates. Schnabel et al. [Sch+16] and Li and Kanoulas [LK17] deal with this problem by adopting a statistical framework similar to ours, however their methods are specialised to IR systems. While the method of Li and Kanoulas [LK17] is adaptive, theoretical guarantees are not rigorously established. In the IR setting, stratified sampling and cluster sampling have also been used to efficiently evaluate knowledge graphs [Gao+19].

Our proposed framework is based on adaptive importance sampling (AIS). AIS is studied more generally in the context of Monte Carlo integration (see review [Bug+17]). Most AIS methods are inappropriate for our application, as they assume a continuous space without constraints on the proposal (see Remark 5.2). Oh and Berger [OB92] introduce the idea of adapting the proposal over multiple stages using samples from the previous stages. Cappé et al. [Cap+08] devise a general framework using independent mixtures as proposals. The method of Cornuet et al. [Cor+12] continually re-weights all past samples, however it is more computationally demanding and less amenable to analysis since it breaks the martingale property. Delyon and Portier [DP18] analyse parametric AIS in the large sample limit. This improves upon earlier work which assumes the number of stages goes to infinity [DM08] or that the sample allocation at each stage is monotonically increasing [MPS19].

5.3 Problem formulation

Consider the task of evaluating a set of systems $\mathcal{S}$ which solve a prediction problem on a feature space $\mathcal{X} \subseteq \mathbb{R}^m$ and label space $\mathcal{Y} \subseteq \mathbb{R}^l$. Let $f^{(s)}(x)$ denote the output produced by system $s \in \mathcal{S}$ for a given input $x \in \mathcal{X}$—e.g. a predicted label or a distribution over labels. We assume instances encountered by the systems are generated i.i.d. from an unknown joint distribution with probability density $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. Our objective is to obtain accurate and precise estimates of system performance measures (e.g. precision, recall) with respect to $p(x, y)$ at minimal cost.

We consider the common scenario where an *unlabelled* test pool $\mathcal{T} = \{x_1, \ldots, x_M\}$ drawn from $p(x)$ is available upfront. However, we assume labels are *unavailable* initially and can only be obtained by querying a stochastic *oracle* that returns draws from the conditional $p(y|x)$. We assume the response time and cost of oracle queries far outweigh

contributions from other parts of the evaluation process. This is reasonable in practice, since the oracle requires human input—e.g. annotators on a crowdsourcing platform or domain experts. To reduce the cost of evaluation, we seek to *minimise* the number of oracle queries required to estimate a target performance measure to a given precision.

Remark 5.1. *A* stochastic oracle *covers the most general case where the label response is random conditional on the features. This may be due to a set of heterogeneous or noisy annotators (not modelled) or genuine ambiguity in the label. We also consider a* deterministic oracle *whose label response is non-random. This is appropriate when trusting an expensive source of truth—e.g. individual judgements from expert annotators or purchased access to a datum.*

We consider performance measures from a broad family, which corresponds to transformations of vector-valued risk functionals.

Definition 5.1 (Generalised measures). *Let $\ell(x, y; f)$ be a vector-valued loss function that maps instances $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to vectors in $\mathbb{R}^d$ dependent on the system outputs $f = \{f^{(s)}\}$. We suppress explicit dependence on f where it is understood. Assume ℓ is uniformly bounded such that*

$$\sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \|\ell(x, y; f)\|_\infty < \infty. \tag{5.1}$$

for a given set of system outputs f. Denote the corresponding vector-valued risk functional by

$$R = \mathop{\mathbb{E}}_{X,Y\sim p}[\ell(X, Y; f)] = \iint \ell(x, y; f)p(x, y)\,\mathrm{d}x\,\mathrm{d}y. \tag{5.2}$$

For any choice of loss function ℓ and continuous mapping $g : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ differentiable at R, the generalised measure *is defined as $G = g(R)$.*

While this family may seem abstract, it encompasses a wide array of important performance measures:

(i) A scalar measure for a single system. Commonly-used measures for classification and regression are supported, as demonstrated in Tables 5.1 and 5.2.

(ii) A vector of measures for a single system. For instance, one could target precision G_{pre} and recall G_{rec} simultaneously by setting $G = [G_{\text{pre}}, G_{\text{rec}}]^\intercal$.

(iii) A vector of measures for multiple systems. For instance, one could target the accuracy of two systems $G^{(1)}_{\text{acc}}$ and $G^{(2)}_{\text{acc}}$ simultaneously by setting $G = \left[G^{(1)}_{\text{acc}}, G^{(2)}_{\text{acc}}\right]^\intercal$.

(iv) A vector of comparative measures. For instance, one could target the difference in accuracy between two candidate systems (2, 3) and a baseline (1) by setting $G = \left[G^{(2)}_{\text{acc}} - G^{(1)}_{\text{acc}}, G^{(3)}_{\text{acc}} - G^{(1)}_{\text{acc}}\right]^\intercal$.

It is important to note that Definition 5.1 establishes a population generalised measure, which is defined with respect to the unknown population distribution $p(x, y)$. We argue that this is the proper way to define a performance measure, as we'd ideally like to measure the performance on the entire population. However in practice, performance measures are usually defined with respect to a sample. A sample measure can be obtained from our more general population measure by substituting the empirical joint distribution $\frac{1}{M}\sum_{i=1}^{M} \mathbb{I}[x_i = x]\,\mathbb{I}[y_i = y]$ for $p(x, y)$. This is illustrated below for recall.

Table 5.1: Parameterisations of common performance measures for binary classification, assuming the class labels are parameterised as $\mathcal{Y} = \{0, 1\}$. Here $f(x)$ denotes the predicted class label according to the system, and $\hat{p}_1(x)$ is an estimate of $p(y = 1|x)$ from the system.

Measure	$\ell(x, y)^\intercal$	$g(R)$
Accuracy	$\mathbb{I}[y \neq f(x)]$	$1 - R$
Balanced accuracy	$[yf(x), y, f(x)]$	$\frac{R_1+R_2(1-R_2-R_3)}{2R_2(1-R_2)}$
Precision	$[yf(x), f(x)]$	$\frac{R_1}{R_2}$
Recall	$[yf(x), y]$	$\frac{R_1}{R_2}$
F_β score	$\left[yf(x), \frac{\beta^2 y+f(x)}{1+\beta^2}\right]$	$\frac{R_1}{R_2}$
Matthews correlation coefficient	$[yf(x), y, f(x)]$	$\frac{R_1-R_2R_3}{\sqrt{R_2R_3(1-R_2)(1-R_3)}}$
Fowlkes-Mallows index	$[yf(x), y, f(x)]$	$\frac{R_1}{\sqrt{R_2R_3}}$
Brier score	$2(\hat{p}_1(x) - y)^2$	R

Table 5.2: Parameterisations of common performance measures for regression. Here $f(x)$ denotes the predicted/fitted value according to the system.

Measure	$\ell(x, y)^\intercal$	$g(R)$
Mean absolute error	$\lvert y - f(x)\rvert$	R
Mean squared error	$(y - f(x))^2$	R
Coefficient of determination	$[y, y^2, f(x), f(x)^2]$	$\frac{R_4-2R_1R_3+R_1^2}{R_2-R_1^2}$

Example 5.1. *The familiar sample-based definition of recall can be obtained by setting* $\ell(x, y) = [yf(x), y]^\intercal$, $g(R) = R_1/R_2$ *and* $p(x, y) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[x_i = x]\,\mathbb{I}[y_i = y]$. *Then*

$$G_{\text{rec}} = g(R) = \frac{\frac{1}{N}\sum_{i=1}^{N} y_i f(x_i)}{\frac{1}{N}\sum_{i=1}^{N} y_i} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

5.4 Limitations of conventional estimation approaches

Since generalised measures are defined in terms of expectations, it is natural to consider Monte Carlo (MC) estimation methods [Liu04]. In this section, we review two MC approaches for evaluation: (i) passive sampling which is the simplest baseline and (ii) importance sampling which has been adopted as a label-efficient evaluation method by Sawade et al. [SLS10] and Schnabel et al. [Sch+16].

5.4.1 Passive sampling

Passive sampling (or conventional MC) estimates an expectation using an unbiased sample from the underlying distribution. It can be applied to estimate the risk R which appears in the definition of the generalised measure, as illustrated below.

Definition 5.2 (Passive estimation of generalised measures). *Let* $\mathscr{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ *be an* unbiased *labelled sample, obtained by drawing items i.i.d. from the marginal distribution* $p(x)$ *and querying labels from the oracle* $p(y|x)$. *We define the passive estimator for* G *based on* $\mathscr{L}$ *as follows:*

$$\hat{R}^{\text{MC}}_{\mathscr{L}} = \frac{1}{|\mathscr{L}|}\sum_{(x,y)\in\mathscr{L}} \ell(x, y) \quad \textit{and} \quad \hat{G}^{\text{MC}}_{\mathscr{L}} = g(\hat{R}^{\text{MC}}_{\mathscr{L}}). \tag{5.3}$$

In general, $\hat{G}^{\text{MC}}_{\mathscr{L}}$ is a biased estimator since g may be non-linear. However, it is asymptotically unbiased—i.e. $\hat{G}^{\text{MC}}_{\mathscr{L}}$ converges to G with probability one in the limit $|\mathscr{L}| \longrightarrow \infty$. This property is known as *strong consistency* and it follows from the strong law of large numbers [Fel68, pp. 243–245] and continuity of g. There is also a central limit theorem for $\hat{G}^{\text{MC}}_{\mathscr{L}}$, reflecting the rate of convergence: $\mathbb{E}[\|\hat{G}^{MC}_{\mathscr{L}} - G\|] \leq \|\Sigma\|/\sqrt{|\mathscr{L}|}$ asymptotically where Σ is an asymptotic covariance matrix (see Theorem 5.5).

Passive sampling is reasonably label-efficient when the generalised measure G is sensitive to regions of the input space $\mathscr{X}$ with high density as measured by $p(x)$. In this case, a sample from $p(x)$ effectively captures the variance in G. One measure that falls into this category is accuracy, as shown below.

Example 5.2 (Passive estimation of accuracy). *Consider estimating accuracy* G_{acc} *of a classifier. The passive estimator for* G_{acc} *is asymptotically normal with variance* $G_{\text{acc}}(1 - G_{\text{acc}})/|\mathscr{L}|$. *Thus, to estimate* G_{acc} *with precision* w *we require a sample of size* $|\mathscr{L}| \propto G_{\text{acc}}(1 - G_{\text{acc}})/w^2$. *Although this is suboptimal (see Proposition 5.8) it is not impractical.*

On the other hand, passive sampling can become impractical in some situations. An important example arises in the context of imbalanced classification. In the presence of class imbalance, one typically uses precision and recall to assess the performance of a classifier. However, these measures are insensitive to instances from the majority class which cover $\mathscr{X}$ w.h.p. We illustrate this problem for recall next.

Example 5.3 (Passive estimation of recall). *Consider estimating recall G_{rec} of a binary classifier where the positive class has frequency ϵ. The asymptotic variance of the passive estimator for G_{rec} is $G_{\text{rec}}(1 - G_{\text{rec}})/|\mathcal{L}|\epsilon$. Thus the sample size required to estimate G_{rec} with precision w is $|\mathcal{L}| \propto G_{\text{rec}}(1 - G_{\text{rec}})/w^2\epsilon$. This inverse scaling in ϵ is problematic for highly imbalanced problems where $\epsilon \ll 1$. For example, in entity resolution ϵ scales inversely in the number of records.*

5.4.2 Importance sampling

We have seen that passive sampling can be inefficient when G is sensitive to parts of the input space $\mathcal{X}$ with *low density* as measured by $p(x)$. In these circumstances, we can improve efficiency substantially by *biasing* sampling towards parts of the space where G is most sensitive. One of the simplest biased sampling methods is *importance sampling (IS)*, which estimates an expectation using samples drawn from a proposal distribution q that differs from p [RK16].

Definition 5.3 (IS estimation of generalised measures). *Select a proposal $q(x)$ whose support includes the support of $p(x)$. Obtain a labelled sample $\mathcal{L}$ by drawing items i.i.d. from $q(x)$, then querying labels from the oracle $p(y|x)$ as before. To correct for the bias, we replace the passive estimator for the* risk *R with an importance-weighted estimator:*

$$\hat{R}^{\text{IS}}_{\mathcal{L}} = \frac{1}{|\mathcal{L}|} \sum_{(x,y)\in\mathcal{L}} \frac{p(x)}{q(x)} \ell(x, y) \quad \text{and} \quad \hat{G}^{\text{IS}}_{\mathcal{L}} = g(\hat{R}^{\text{IS}}_{\mathcal{L}}). \tag{5.4}$$

The challenge with IS lies in choosing an effective proposal. Later (see Section 5.7) we derive a proposal $q^\star(x)$ that minimises the asymptotic variance of $\hat{G}^{\text{IS}}_{\mathcal{L}}$, in some cases to zero. However, we cannot compute $q^\star(x)$ exactly, since it depends on the unknown oracle distribution $p(y|x)$. While $p(y|x)$ can in principle be estimated using the systems under evaluation, the reliability of the estimate places a hard limit on label efficiency. To address this limitation, we propose an adaptive importance sampling (AIS) framework in the next section, which continually refines the proposal as labels are received from the oracle.

Remark 5.2 (Constraint on the proposal). *In ordinary applications of IS, one would be free to select any proposal that satisfies $q(x, y) > 0$ wherever $\|\ell(x, y)\|p(x, y) \neq 0$. However, in our application we have an additional constraint: we cannot bias sampling from the oracle distribution $p(y|x)$. Thus we consider proposals of the form $q(x, y) = q(x)p(y|x)$.*

5.5 An AIS-based framework for evaluation

Motivated by the limitations of passive sampling and importance sampling (IS), we now outline a framework for estimating generalised measures based on *adaptive importance sampling (AIS)*. Unlike passive sampling and IS, AIS is not restricted to selecting instances to label in an i.i.d. fashion. Instead, instances are selected for labelling in stages and the labelling policy (proposal) is adapted based on labels collected in previous stages. This can yield more precise estimates for a given sample size, particularly if the policy converges rapidly to optimality.

Figure 5.1 and Algorithm 5.1 provide an overview of the proposed framework. Before evaluation begins, the following five components must be in place:

- the set of systems under evaluation $\mathcal{S}$;
- the performance measure G;
- the unlabelled test pool $\mathcal{T}$;
- the adaptive labelling policy; and
- the oracle.

As illustrated in the figure, the performance measure G may depend implicitly on the systems under evaluation. It is also used to inform the labelling policy, e.g. through a variance minimisation approach.

The first stage of the evaluation process begins by sampling N_1 instances to label $\{x_{1,1}, \dots, x_{1,N_1}\}$ i.i.d. from the test pool according to an initial proposal q_0. The initial proposal can be configured based on prior knowledge or information from the systems under evaluation. Labels for the sampled instances $\{y_{1,1}, \dots, y_{1,N_1}\}$ are then queried from the oracle, potentially in parallel. Finally, the labelled samples and their importance weights are used to update the proposal q_1 for the next stage. The same process is followed in all subsequent stages: at the t-th stage N_t instances are sampled for labelling according to the proposal q_{t-1}, and the entire sampling history $\mathcal{L}$ is used to update the proposal for the next stage q_t. At any time during the evaluation process, an estimate of the performance measure G can be obtained using an importance-weighted (bias-corrected) estimator:

$$\hat{R}_{\mathcal{L}}^{\mathrm{AIS}} = \frac{1}{|\mathcal{L}|} \sum_{(x,y,w)\in\mathcal{L}} w\, \ell(x,y)\,, \quad \hat{G}_{\mathcal{L}}^{\mathrm{AIS}} = g\left(\hat{R}_{\mathcal{L}}^{\mathrm{AIS}}\right). \tag{5.5}$$

An important component of the framework is the adaptive labelling policy, which is responsible for generating and updating the sequence of proposals $\{q_{t-1}\}$. In order to improve label efficiency, the sequence of proposals can be designed to minimise the asymptotic variance of the performance estimate $\hat{G}_{\mathcal{L}}^{\mathrm{AIS}}$. In Section 5.7, we derive the proposal that minimises the asymptotic variance assuming the oracle response $p(y|x)$ is known. This proposal serves as an optimal "target" for designing a policy, however it cannot be used directly since $p(y|x)$ is unknown. In Chapter 6 we investigate practical policies based on adaptive estimates of $p(y|x)$. However, for the remainder of this chapter we consider arbitrary policies in order to provide general theoretical results.

Remark 5.3 (Variations of AIS). *There are many variations of AIS which differ in: (i) the way samples are allocated among the stages; (ii) the dependence of the proposal on previous stages; (iii) the types of proposals considered; and (iv) the way samples are weighted within and across stages. Our framework is completely flexible with respect to points (i)–(iii). For point (iv), we use simple importance-weighting because it is amenable to asymptotic analysis using martingale theory [DP18]. A more complex weighting scheme is proposed by Cornuet et al. [Cor+12] which may have better stability, however its asymptotic behaviour is not well understood.*[1]

[1] Marin et al. [MPS19] proved consistency for this weighting scheme in the limit $T \longrightarrow \infty$ where $\{N_t\}$ is a monotonically increasing sequence. To our knowledge, a CLT remains an open problem.

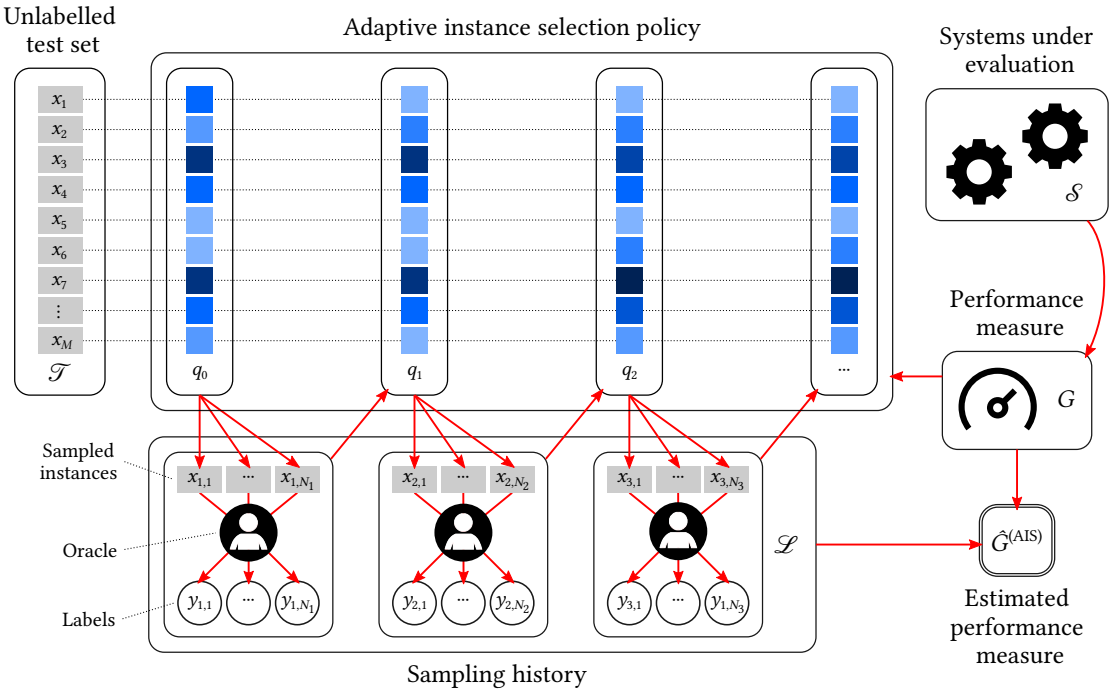


Figure 5.1: Schematic of our proposed evaluation framework

5.6 Asymptotic analysis

Since our proposed framework produces a sequence of dependent samples, we cannot rely on standard asymptotic theory to characterise the properties of the resulting estimates. In this section, we therefore establish two important asymptotic results for the performance estimates produced by Algorithm 5.1: strong consistency and a central limit theorem (CLT).

The analysis in this section does not depend on how samples are allocated among stages, so we switch to single index j rather than the pair of indices (t, n) used in Algorithm 5.1. Concretely, we bijectively map each (t, n) to index $j = n + \sum_{t'=1}^{t-1} N_{t'}$. As a result, j takes on values in $\{1, \dots, N\}$ where $N = \sum_{t=1}^{T} N_t$ is the total number of samples. In addition, we modify the indexing for the sequence of proposals so that $q_{j-1}(x)$ denotes the proposal used to generate sample j. It is important to note that this notation conceals the dependence of $q_{j-1}(x)$ on the previous samples. Thus $q_{j-1}(x)$ should be understood as shorthand for $q_{j-1}(x|\mathcal{F}_{j-1})$ where $\mathcal{F}_j = \sigma((X_1, Y_1), \dots, (X_j, Y_j))$ denotes the filtration.

Following Delyon and Portier [DP18], our analysis relies on the fact that

$$Z_N = N(\hat{R}_N^{\mathrm{AIS}} - R) = \sum_{j=1}^{N} \left\{ \frac{p(X_j)}{q_{j-1}(X_j)} \ell(X_j, Y_j) - R \right\} \tag{5.6}$$

is a martingale with respect to $\mathcal{F}_N$. The consistency of Algorithm 5.1 then follows by a strong law of large numbers for martingales [Fel71] and the continuous mapping theorem.

Theorem 5.4 (Consistency). *Suppose the support of proposal $q_j(x, y) = q_j(x)p(y|x)$ is a*

Algorithm 5.1 AIS for estimating generalised measures

Require: Unlabelled test pool $\mathcal{T}$; generalised measure G (specified by ℓ and g); procedure for initialising and updating the proposal; number of stages T; sample allocations $N_1, \dots, N_T$.
Initialise proposal q_0
Initialise sample history: $\mathcal{L} \longleftarrow \emptyset$
for $t \in \{1, \dots, T\}$ **do**
  for $n \in \{1, \dots, N_t\}$ **do**
    Draw item: $x_{t,n} \sim q_{t-1}$
    Compute weight: $w_{t,n} \longleftarrow {}^{p(x_{t,n})}/_{q_{t-1}(x_{t,n})}$
    Query label: $y_{t,n} \sim \text{Oracle}(x_{t,n})$
    Update history: $\mathcal{L} \longleftarrow \mathcal{L} \cup \{(x_{t,n}, y_{t,n}, w_{t,n})\}$
  end for
  Compute updated proposal q_t using $\mathcal{L}$
end for
$\hat{R}^{\text{AIS}}_{\mathcal{L}} \longleftarrow \frac{1}{|\mathcal{L}|} \sum_{(x,y,w)\in\mathcal{L}} w\, \ell(x, y)$
$\hat{G}^{\text{AIS}}_{\mathcal{L}} \longleftarrow g(\hat{R}^{\text{AIS}}_{\mathcal{L}})$
Return: $\hat{G}^{\text{AIS}}_{\mathcal{L}}$ and history $\mathcal{L}$

superset of $\{x, y \in \mathcal{X} \times \mathcal{Y} : \|\ell(x, y)\| p(x, y) \neq 0\}$ *for all* $j \geq 0$ *and assume*

$$\sup_{j\in\mathbb{N}} \mathbb{E}\left[\left(\frac{p(X_j)}{q_{j-1}(X_j)}\right)^2 \middle| \mathcal{F}_{j-1}\right] < \infty. \tag{5.7}$$

Then $\hat{G}^{\text{AIS}}_N$ *is strongly consistent for* G.

Proof. We first prove that $\hat{R}^{\text{AIS}}_N \xrightarrow{\text{a.s.}} R$ using a strong law of large numbers (SLLN) for martingales [Fel71, p. 243]. Consider the i-th component of the j-th contribution to Z_N as defined in (5.6):

$$\delta_{j,i} = \frac{p(X_j)}{q_{j-1}(X_j)} \ell_i(X_j, Y_j) - R_i.$$

Since (X_j, Y_j) is drawn from $p(y|x)q_{j-1}(x)$ and $q_{j-1}(x) > 0$ wherever $p(x)\|\ell(x, y)\| \neq 0$, it follows that $\mathbb{E}[\delta_{j,i}|\mathcal{F}_{j-1}] = 0$. In addition, we have

$$\begin{aligned}\sum_{j=1}^{\infty} \frac{\mathbb{E}[\delta_{j,i}^2]}{j^2} &= \sum_{j=1}^{\infty} \frac{1}{j^2} \left\{ \mathbb{E}\left[\left(\frac{p(X_j)}{q_{j-1}(X_j)} \ell_i(X_j, Y_j)\right)^2\right] + R_i^2 \right\} \\ &\leq \sum_{j=1}^{\infty} \frac{U^2 C}{j^2} < \infty,\end{aligned}$$

where the inequality follows from (5.1) and (5.7). Thus the conditions of Feller's SLLN are satisfied and we have $\frac{1}{N}\sum_{j=1}^{N} \delta_{i,j} \xrightarrow{\text{a.s.}} 0$, which implies $\hat{R}^{\text{AIS}}_N \xrightarrow{\text{a.s.}} R$.

Now the continuous mapping theorem implies that

$$\hat{R}^{\text{AIS}}_N \xrightarrow{\text{a.s.}} R \quad \Longrightarrow \quad g(\hat{R}^{\text{AIS}}_N) \xrightarrow{\text{a.s.}} g(R),$$

provided R is not in the set of discontinuity points of g. This condition is satisfied by assumption. □

We also obtain a central limit theorem (CLT) for Algorithm 5.1, which is useful for assessing asymptotic efficiency and computing approximate confidence intervals. Our proof invokes a CLT due to Delyon and Portier [DP18] and the multivariate delta method.

Theorem 5.5 (CLT). *Let*

$$V_j = \operatorname{var}\left[\frac{p(X_j)}{q_{j-1}(X_j)}\ell(X_j, Y_j) - R \middle| \mathcal{F}_{j-1}\right], \tag{5.8}$$

and let V_∞ be an a.s. finite random positive semidefinite matrix. Suppose

$$V_j \longrightarrow V_\infty \quad \text{a.s.}, \quad \text{and} \tag{5.9}$$

$$\exists \eta > 0 \,:\, \sup_{j\in\mathbb{N}} \mathbb{E}\left[\left(\frac{p(X_j)}{q_{j-1}(X_j)}\right)^{2+\eta} \middle| \mathcal{F}_{j-1}\right] < \infty \quad \text{a.s.} \tag{5.10}$$

Then $\sqrt{N}(\hat{G}_N^{\mathrm{AIS}} - G)$ converges in distribution to a multivariate normal $\mathcal{N}(0, \Sigma)$ with covariance matrix

$$\Sigma = \mathrm{D}g(R) V_\infty \mathrm{D}g(R)^\intercal \tag{5.11}$$

where $[\mathrm{D}g]_{ij} = \frac{\partial g_i}{\partial R_j}$ is the Jacobian of g.

Proof. The CLT of Delyon and Portier [DP18] implies that $\sqrt{N}(\hat{R}^{\mathrm{AIS}} - R) \Longrightarrow \mathcal{N}(0, V_\infty)$. We note that the second condition of their theorem

$$\exists \eta > 0 \,:\, \sup_{j\in\mathbb{N}} \iint \frac{\|\ell(x,y)p(x,y)\|^{2+\eta}}{q_j(x)^{1+\eta}}\,\mathrm{d}x < \infty \quad \text{a.s.}$$

is satisfied by boundedness of the loss function (5.1) and (5.10). The delta method [Vaa98] then implies that $\sqrt{N}(g(\hat{R}^{\mathrm{AIS}}) - g(R)) \Longrightarrow \mathcal{N}(0, \mathrm{D}g(R) V_\infty \mathrm{D}g(R)^\intercal)$, since g is assumed to be differentiable at R in Definition 5.1. □

In order to simplify subsequent analysis of adaptive labelling policies, we establish conditions under which Theorems 5.4 and 5.5 hold for the special case where $\mathcal{X}$ is a finite pool of test data.

Corollary 5.6. *Suppose the generalised measure G is defined with respect to a finite input space $\mathcal{X}$ (e.g. a test pool).*

(i) If the support of proposal $q_j(x, y) = q_j(x)p(y|x)$ is a superset of $\{x, y \in \mathcal{X} \times \mathcal{Y} : p(x, y)\|\ell(x, y)\| \neq 0\}$ for all $j \geq 0$, then Theorem 5.4 holds.

(ii) If in addition $q_j(x) \xrightarrow{\text{a.s.}} q_\infty(x)$ pointwise in x, then Theorem 5.5 holds.

Proof. For the first statement, we check conditions (5.7) and (5.10) of Theorem 5.4. Let $\mathbb{Q}_j$ be the support of $q_j(x)$ and let $\delta_j = \inf_{x\in\mathbb{Q}_j} q_j(x) > 0$. For $\eta \geq 0$ we have

$$\mathbb{E}\left[\left(\frac{p(X_j)}{q_{j-1}(X_j)}\right)^{2+\eta} \middle| \mathcal{F}_{j-1}\right] = \int \sum_{x\in\mathbb{Q}_{j-1}} \left(\frac{p(x)}{q_{j-1}(x)}\right)^{2+\eta} q_{j-1}(x)p(y|x)\,\mathrm{d}y \leq \left(\frac{1}{\delta_j}\right)^{2+\eta} < \infty.$$

For the second statement, we must additionally check condition (5.9) regarding the convergence of V_j. Using (5.8) we write $V_j = \int \sum_{x \in \mathbb{Q}_j} f_j(x, y)\, \mathrm{d}y$ where the integrand is

$$f_j(x, y) = \left(\frac{p(x)}{q_j(x)} \ell(x, y) - R \right) \left(\frac{p(x)}{q_j(x)} \ell(x, y) - R \right)^\mathsf{T} q_j(x) p(y|x).$$

By the a.s. pointwise convergence of $q_j(x)$ and the continuous mapping theorem, we have $f_j(x, y) \longrightarrow f_\infty(x, y)$ a.s. pointwise in x and y. Now observe that

$$\begin{aligned} \|f_j(x, y)\|_2 &= q_j(x, y) \left\| \frac{p(x)}{q_j(x)} \ell(x, y) - R \right\|_2^2 \\ &\leq q_j(x, y) \left(\frac{p(x, y)^2}{q_j(x, y)^2} \|\ell(x, y)\|_2^2 + \|R\|_2^2 \right) \\ &\leq p(y|x) \left(\frac{1}{\epsilon^2} \|\ell(x, y)\|_2^2 + \|R\|_2^2 \right) = h(x, y) \end{aligned}$$

It is straightforward to show that $\int \sum_{x \in \mathbb{Q}_j} h(x, y)\, \mathrm{d}y < \infty$ using (5.1). Thus we have $V_j \longrightarrow V_\infty$ by the dominated convergence theorem. □

5.7 Asymptotic optimality

The CLT obtained in the previous section can be used to assess the asymptotic label efficiency of our framework under different labelling policies. Concretely, it implies that the number of labelled samples required to estimate the target performance measure G with precision w is proportional to Σ / w^2 asymptotically, where Σ is the asymptotic variance.[2] In general, the asymptotic variance depends on the asymptotic behaviour of the labelling policy. In this section, we derive the asymptotically-optimal policy that *minimises* the asymptotic variance assuming the oracle response $p(y|x)$ is known. We also demonstrate that the asymptotic variance cannot be reduced to zero in general, even when the asymptotically-optimal policy is used.

To avoid degenerate cases, we assume the target performance measure G is such that the Jacobian $\mathrm{D}g(R)$ has full row rank. This ensures the asymptotic covariance matrix Σ is positive (semi)definite, provided V_∞ is positive (semi)definite (see 5.11). If this was not the case and Σ was degenerate, then we would need to use a higher-order expansion than the CLT provides to assess asymptotic efficiency.

We must also decide how to measure asymptotic efficiency when G is a vector-valued target performance measure. In this case, Σ becomes an asymptotic covariance matrix so it cannot be used directly as the objective for a minimisation problem. We opt to use the *total variance* (the trace):

$$\operatorname{tr} \Sigma = \mathop{\mathbb{E}}_{X, Y \sim p} \left[\frac{p(X) \|\mathrm{D}_g(R)\, \ell(X, Y)\|_2^2}{q_\infty(X)} \right] - \|\mathrm{D}_g(R)\, R\|_2^2. \tag{5.12}$$

This is a reasonable choice because the diagonal elements of Σ are directly related to statistical efficiency, while the off-diagonal elements measure correlations between

[2] Here we assume a scalar target performance measure for simplicity.

components of $\hat{G}_N^{\text{AIS}}$ that are beyond our control. Another benefit of this choice, is that it results in a tractable optimisation problem for the asymptotically-optimal policy, as demonstrated below.

Proposition 5.7 (Asymptotically-optimal policy). *Suppose* $\mathrm{D}g(R)$ *has full row rank and* $\mathbb{E}_{X,Y\sim p}\|\mathrm{D}_g(R)\,\ell(X,Y)\|_2^2 > 0$. *If the labelling policy converges to the proposal*

$$q^\star(x) = \frac{v(x)}{\int v(x)\,\mathrm{d}x} \quad \textit{where} \quad v(x) = p(x)\sqrt{\int \|\mathrm{D}_g(R)\,\ell(x,y)\|_2^2\, p(y|x)\,\mathrm{d}y}, \tag{5.13}$$

then it achieves the minimum asymptotic total variance

$$\operatorname{tr}\Sigma[q^\star] = \left(\int v(x)\,\mathrm{d}x\right)^2 - \|\mathrm{D}_g(R)\,R\|_2^2. \tag{5.14}$$

Proof. We want to find the proposal q_∞ that minimises $\operatorname{tr}\Sigma$. Using (5.12), we express this as a functional optimisation problem:

$$\begin{aligned} \min_{q_\infty} \quad & \int \frac{c(x)}{q_\infty(x)}\mathrm{d}x \\ \text{s.t.} \quad & \int q_\infty(x)\,\mathrm{d}x = 1, \end{aligned} \tag{5.15}$$

where $c(x) = p(x)^2 \int \|\mathrm{D}_g(R)\,\ell(x,y)\|_2^2\, p(y|x)\mathrm{d}y$.

Using the method of Lagrange multipliers, Sawade et al. [SLS10] show that the solution to (5.15) is $q^\star(x) \propto \sqrt{c(x)}$. This yields the required result. □

In typical applications of importance sampling, one can theoretically select a proposal that achieves zero variance. However, this is not always possible in our application, since we do not have complete freedom in selecting the proposal (see Remark 5.2). Below we provide sufficient conditions on the target performance measure and oracle, which ensure that the asymptotic total variance can be reduced to zero.

Proposition 5.8. *Suppose the oracle is deterministic (i.e.* $p(y|x)$ *is a point mass for all* x*) and the generalised measure is such that* $\operatorname{sign}(\ell(x,y)\cdot\nabla g_l(R))$ *is constant for all* $(x,y)\in\mathcal{X}\times\mathcal{Y}$ *and* $l\in\{1,\dots,m\}$. *Then the asymptotically-optimal policy given in Proposition 5.7 achieves* $\operatorname{tr}\Sigma = 0$.

Proof. We evaluate the two terms in (5.14) separately. Using the fact that $p(y|x) = \mathbb{I}[y = y(x)]$, the first term becomes

$$\begin{aligned} \left(\int v(x)\,dx\right)^2 &= \left(\int \|\mathrm{D}_g(R)\,\ell(x,y(x))\|_2\, p(x)\,\mathrm{d}x\right)^2 \\ &\le \left(\int \|\mathrm{D}_g(R)\,\ell(x,y(x))\|_1\, p(x)\,\mathrm{d}x\right)^2 \\ &= \left(\sum_{l=1}^m \int \ell(x,y(x))\cdot\nabla g_l(R)\, p(x)\,\mathrm{d}x\right)^2. \end{aligned}$$

The second line follows by application of the inequality $\|x\|_2 \leq \|x\|_1$, and the third line follows by assumption. For the second term we have

$$\begin{aligned}\|\mathrm{D}_g(R)\, R\|_2^2 &= \left\| \int \mathrm{D}_g(R)\, \ell(x, y(x))\, p(x)\, \mathrm{d}x \right\|_2^2 \\ &= \sum_{l=1}^{m} \left(\int \ell(x, y(x)) \cdot \nabla g_l(R)\, p(x)\, \mathrm{d}x \right)^2 \\ &\geq \left(\sum_{l=1}^{m} \int \ell(x, y(x)) \cdot \nabla g_l(R)\, p(x)\, \mathrm{d}x \right)^2,\end{aligned}$$

by application of Jensen's inequality. Subtracting the second term from the first, we have $\operatorname{tr} \Sigma \leq 0$. □

By way of illustration, we apply the above proposition to two common performance measures: accuracy and recall. We assume a deterministic oracle in both cases.

Example 5.4 (Asymptotic variance for accuracy). *From Table 5.1, we have that accuracy can be expressed as a generalised performance measure by setting $\ell(x, y) = \mathbb{I}[y \neq f(x)]$ and $g(R) = 1 - R$. Evaluating the condition in Proposition 5.8, we have*

$$\operatorname{sign}\left(\ell(x, y) \cdot \nabla g(R)\right) = \operatorname{sign}\left(-\mathbb{I}[y \neq f(x)]\right) = -1$$

for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Thus our framework can achieve zero asymptotic total variance when estimating accuracy under a deterministic oracle.

Example 5.5 (Asymptotic variance for recall). *From Table 5.1, we have that recall can be expressed as a generalised performance measure by setting $\ell(x, y) = [yf(x), y]^\intercal$ and $g(R) = R_1/R_2$. The conditions of Proposition 5.8 are not satisfied in this case, since*

$$\operatorname{sign}\left(\ell(x, y) \cdot \nabla g(R)\right) = \operatorname{sign}\left(\frac{y}{R_2}(f(x) - G_{\mathrm{rec}})\right) = \operatorname{sign}\left(f(x) - G_{\mathrm{rec}}\right)$$

which is not constant for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Indeed, when we evaluate the expression for the asymptotic total variance (5.14), we find that $\Sigma = 4G_{\mathrm{rec}}^2(1 - G_{\mathrm{rec}})^2$. Thus, even if we knew the deterministic oracle response in advance, there is a strict positive lower bound on the asymptotic variance that can be achieved by our framework when estimating recall.

5.8 Practicalities

Until this point, we have mainly focused on theoretical properties of our proposed framework. In this section, we discuss several issues that may arise when using the framework for practical evaluation tasks. We continue to assume a generic adaptive labelling policy throughout the discussion. Issues related to the adaptive labelling policy are discussed in Chapter 6.

5.8.1 Batch size

For generality, we permit the user to specify the batch size (or sample allocation) N_t in each stage t of the evaluation process (see Algorithm 5.1). Ideally, we recommend selecting a small batch size, as empirical studies suggest that efficiency improves when the policy is adapted more frequently [DP18]. However, the selection of the batch size must be balanced with practical constraints on the oracle, since a small batch size limits parallelisability of labelling. This is because all labels for a batch must be returned before a new batch of items can be selected for the next round.

5.8.2 Sample reuse

Suppose our framework is used to estimate a generalised measure G_1. If the population distribution $p(x, y)$ has not changed, it may be desirable to reuse the weighted samples $\mathscr{L}$ acquired while estimating G_1 to estimate a *different* generalised measure G_2. This is possible so long as the sequence of proposals $\{q_j\}$ used to estimate G_1 have the required support for G_2 (see Theorem 5.4). More precisely, the support of $q_j(x, y)$ must include $\{x, y \in \mathscr{X} \times \mathscr{Y} : \|\ell(x, y)\|p(x, y) \neq 0\}$ for the loss functions associated with G_1 *and* G_2.

This condition may not be satisfied if the labelling policy is asymptotically-optimal for G_1. For instance, consider the problem of estimating the precision of two binary classifiers labelled "1" and "2" sequentially without gathering more samples. Specifically, we would like to estimate the precision G_2 of binary classifier 2 using weighted samples previously obtained while estimating the precision G_1 of binary classifier 1. Referring to (5.13), we note that the asymptotically-optimal policy for G_1 places *zero weight* on any instances that are predicted negative by classifier 1. Thus if classifier 2 disagrees with any of the negative predictions made by classifier 1, it is not possible to reuse the weighted samples while ensuring that consistency holds. It would therefore be necessary to gather more samples, so that all of the instances which are predicted positive by classifier 2 have a non-zero chance of being included in the sample.

If one anticipates sample reuse, it is possible to avoid this issue by sacrificing asymptotic optimality of the policy. In the example above, the target policy could be made less specialised to G_1 by mixing with the marginal distribution $p(x)$, i.e. $q^\star(x) \longrightarrow (1 - \delta)q^\star(x) + \delta p(x)$ where $\delta \in (0, 1]$ is a hyperparameter that controls the degree of specialisation. The parameter setting $\delta = 0$ recovers the asymptotically-optimal policy, and $\delta \longrightarrow 0$ corresponds to passive sampling.

5.8.3 Approximate confidence regions

The CLT can be used to compute asymptotic confidence regions for the performance estimates produced by our framework.[3] These may be used as a rough guide for assessing statistical precision of the performance estimates after evaluation is complete. In the most general case where G is vector-valued, the approximate $100(1 - \alpha)\%$ confidence region is an ellipsoid formed by vectors $G^\star \in \mathbb{R}^k$ that satisfy:

$$(G^\star - \hat{G})^\intercal \hat{\Sigma}^{-1} (G^\star - \hat{G}) \leq \frac{(N-1)k}{N(N-k)} F_{\alpha,k,N-k},$$

[3] Bootstrap methods can alternatively be used to compute approximate confidence intervals [DE96].

where $\hat{G}$ is the sample mean, $\hat{\Sigma}$ is the sample covariance matrix, and F_{α,d_1,d_2} is the critical value of the F distribution with d_1, d_2 degrees of freedom at significance level α.

We can approximate this region using the AIS estimator for G in (5.5) and the following estimator for Σ:

$$\hat{\Sigma}^{\text{AIS}} = \mathrm{D}_g(\hat{R}^{\text{AIS}})\left(\frac{1}{N}\sum_{j=1}^{N}\frac{p(x_j)^2\ell(x_j,y_j)\ell(x_j,y_j)^\intercal}{q_N(x_j)q_{j-1}(x_j)} - \hat{R}^{\text{AIS}}\hat{R}^{\text{AIS}\intercal}\right)\mathrm{D}_g(\hat{R}^{\text{AIS}})^\intercal.$$

This estimator is derived from the following expression for the asymptotic covariance matrix:

$$\Sigma = \mathrm{D}g(R)\left(\mathop{\mathbb{E}}_{X,Y\sim p}\left[\frac{p(X)\ell(X,Y)\ell(X,Y)^\intercal}{q_\infty(X)}\right] - RR^\intercal\right)\mathrm{D}g(R)^\intercal.$$

Specifically, plug-in AIS estimators are used to approximate the expectation and R, and the most recent proposal $q_N(x)$ is used to approximate the asymptotic policy $q_\infty(x)$.

5.9 Concluding remarks

Evaluation is an important tool for users of ER systems, as it provides assurance that the system is meeting or exceeding performance targets. However, since evaluation relies on a sample of labelled data, sources of statistical bias or noise can impact the reliability of performance estimates. In this chapter, we have argued that these sources of error are a major problem for evaluation of ER systems, as severe class imbalance between matches and non-matches leads to performance estimates with high variance. This means enormous samples of labelled data are required to drive down the variance under standard (passive) labelling strategies.

In order to reduce the amount of labelled data required to achieve precise performance estimates, we proposed an evaluation framework based on adaptive importance sampling (AIS) in this chapter. In contrast to standard labelling strategies based on passive (uniform) sampling, our AIS framework samples items to label in a *biased* fashion, adapting the sampling as labels are received. This can significantly improve the precision of performance estimates if the adaptive labelling policy is carefully designed.

In addition to outlining the framework, our primary focus in this chapter was on obtaining theoretical results, which are important since adaptivity breaks standard theoretical guarantees. We proved strong consistency and a central limit theorem for performance estimates produced by our framework, under verifiable conditions on the target performance measure and adaptive labelling policy. These results ensure that performance estimates converge to the population performance asymptotically, and provide a route for computing approximate confidence intervals. In addition, we used the central limit theorem to derive the asymptotically-optimal labelling policy. In the next chapter, we continue to explore our evaluation framework in a highly practical setting. We design adaptive labelling policies which approximate the asymptotically-optimal policy derived in this chapter, and run simulations on a variety of evaluation tasks to assess expected gains in label-efficiency.

Chapter 6

Adaptive policies for label-efficient evaluation

In this chapter, we design and evaluate adaptive labelling policies for the theoretical framework presented in Chapter 5. All of the policies are designed to approximate the *asymptotically-optimal policy*, which provably minimises the asymptotic variance of the estimated performance measure for the system(s) under evaluation. Our approximations rely on adaptive estimates of the unknown oracle response $p(y|x)$. We devise two stratified estimators for $p(y|x)$ based on Bayesian models: one which assumes the strata are independent and another which assumes the strata are hierarchically dependent. We conduct an empirical study to compare the label-efficiency of our evaluation framework (under three adaptive labelling policies) against non-adaptive baselines. The results of the study suggest that our evaluation framework can achieve significant gains in label efficiency—by several orders of magnitude in some cases—especially for problem domains such as entity resolution which suffer from severe class imbalance.

6.1 Introduction

In the previous chapter, we proposed a theoretical framework for evaluation based on adaptive importance sampling (AIS). A key component of the framework is the adaptive labelling policy, which is responsible for sampling the "most informative" instances to label based on the previously acquired labels. While there is much freedom in choosing a policy that satisfies the minimum requirement of consistency, it is important that the policy is tailored to the data distribution and target performance measure in order to achieve good label-efficiency. In the AIS literature, it is common to tailor the policy using

This chapter incorporates material from the following publications:

- N. G. Marchant and B. I. P. Rubinstein. "In Search of an Entity Resolution OASIS: Optimal Asymptotic Sequential Importance Sampling". In: *Proc. VLDB Endow.* 10.11 (2017), pp. 1322-1333. DOI: 10.14778/3137628.3137642.
- N. G. Marchant and B. I. P. Rubinstein. "Needle in a Haystack: Label-Efficient Evaluation under Extreme Class Imbalance". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '21. Virtual Event, Singapore: ACM, 2021. DOI: 10.1145/3447548.3467435. Accepted.

a *variance minimisation* criterion [Bug+17]. In Proposition 5.7, we derived an expression for the *asymptotically-optimal policy* which minimises the total asymptotic variance of the estimated target performance measure. However, there is a chicken and egg problem, in that the expression for the asymptotically-optimal policy depends on the unknown response $p(y|x)$ of the labelling oracle. Hence, we must devise methods for approximating the asymptotically-optimal policy based on a limited sample of labels acquired from the oracle.

We decompose the problem into two sub-problems. First, we design estimators for the asymptotically-optimal policy, which depend on plug-in estimators for the oracle response $p(y|x)$ and the risk R associated with the target performance measure. These estimators are used to compute a new policy at the end of each round t in Algorithm 5.1 based on the samples collected in round t and all preceding rounds. Designing a plug-in estimator may seem trivial, since we already derived an expression for the asymptotically-optimal policy in Proposition 5.7, however there are subtleties to consider. In particular, we must ensure that the asymptotically-optimal policy satisfies the conditions of Theorem 5.4 when $p(y|x)$ and R are replaced by inexact estimates. We also propose different estimators depending on whether the oracle response is stochastic or deterministic.

Second, we consider the sub-problem of estimating the oracle response $p(y|x)$ adaptively based on the incoming labels. While there are many conceivable approaches for estimating $p(y|x)$, we focus on stratified Bayesian models, which naturally incorporate prior knowledge from the system(s) under evaluation, and are label-efficient due to sharing of statistical strength. Concretely, we partition the test pool into disjoint strata (blocks), such that all instances within a stratum are likely to elicit a similar response from the oracle. We then approximate $p(y|x)$ using a stratum-level estimate, rather than an instance-level estimate. We consider two models based on this idea. The first and simplest model assumes the response in each stratum follows an *independent* Dirichlet-Categorical distribution. It is a generalisation of the additive smoothing approach used by Bennett and Carvalho [BC10] to estimate the stratum-level variance of a binary response. The second model attempts to exploit dependencies between neighbouring strata, by assuming each instance is propagated through a hierarchy of strata, conditional on the label. We consider two variants of the second model for stochastic and deterministic oracles.

As a final step, we combine solutions for each sub-problem to produce three adaptive labelling policies. We show that the policies satisfy the conditions of our theoretical framework, thereby ensuring strong consistency and asymptotic normality of the performance estimates. We also present a comprehensive empirical study of our framework under the three adaptive labelling policies, with comparisons to baseline approaches. The results of the study show that our framework can achieve significant improvements in label-efficiency, particularly when the labels are severely imbalanced. This demonstrates the effectiveness of our framework for evaluating entity resolution systems in an efficient, statistically-sound manner.

Chapter outline. Section 6.2 provides background on stratification methods, which are used in subsequent sections to develop approximations to the asymptotically-optimal policy. Section 6.3 explores several options for estimating the asymptotically-optimal policy, assuming estimates of the oracle response and target risk are available. Concrete

estimators for the oracle response are developed in Section 6.4 using stratified Bayesian models. Section 6.5 combines the results of the previous two sections to produce three adaptive labelling policies. These policies are studied empirically in Section 6.6, along with comparisons to baseline evaluation methods. Section 6.7 concludes with a summary of our contributions and ideas for future work.

6.2 Stratification methods

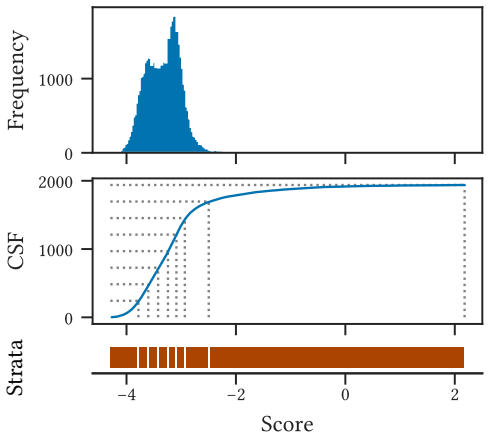
Stratification is the process of partitioning a space or population into disjoint sub-spaces or sub-populations, which are approximately homogeneous. It is a commonly-used principle for reducing sampling error in the experimental design and Monte Carlo simulation literature [Coc77; RK16]. Our use of stratification is somewhat atypical, as we are not using it to estimate a scalar expectation/population parameter. Instead, we are interested in using stratification to obtain a good approximation of the oracle response $p(y|x)$ for all instances x in a test pool $\mathcal{T}$, when labelled data is scarce. The naïve approach to estimating $p(y|x)$ requires $(|\mathcal{Y}| - 1) \times |\mathcal{T}|$ free parameters, however if we stratify the test pool into K strata (sub-populations) such that the response is *similar* within each stratum, we can estimate $p(y|x)$ at the stratum-level reasonably accurately using only $(|\mathcal{Y}| - 1) \times K$ free parameters. While the stratum-level estimate does not necessarily converge to the true response for all instances, it may be sufficiently accurate, especially since labelled data is limited in our application.

In the remainder of this section, we review methods for constructing a stratified test pool $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{T}_k$ in an unsupervised manner—i.e. assuming the oracle response is unknown. Section 6.2.1 describes methods that leverage scores from the systems under evaluation, while Section 6.2.2 describes methods that depend on feature vectors.

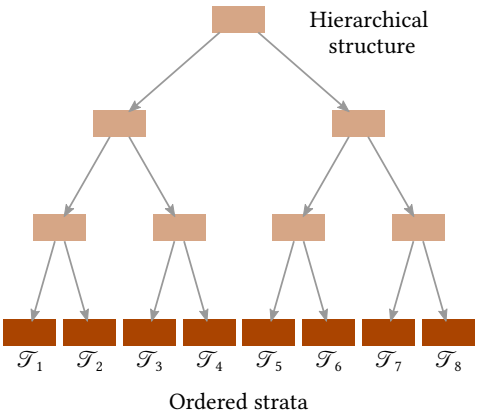
6.2.1 Score-based methods

In most practical scenarios, the system(s) under evaluation are capable of providing scores for instances in the test pool, which are correlated with the oracle response. For example, an entity resolution system might return a real-valued score on the unit interval that quantifies the match confidence. Such scores can be used as a proxy for the true oracle response. We briefly review two methods from the experimental design literature, which construct strata by binning each instance $x \in \mathcal{T}$ according to some variable of interest $v(x) \in \mathbb{R}$, known as the *stratification variable*. Assuming $\mathcal{Y} = \{0, 1\}$ (a binary classification setting), we can achieve a roughly homogeneous oracle response within each stratum by setting $v(x) = p(y = 1|x)$ and approximating $v(x)$ using scores from the system(s) under evaluation.

CSF method. Under the assumption that one would like to estimate the mean of the stratification variable $\bar{v} = \frac{1}{M} \sum_{i=1}^{M} v(x)$, Dalenius and Hodges [DH59] propose a method for selecting bin boundaries which approximately minimise the variance of a stratified estimator for $\bar{v}$. They make several simplifying assumptions: they assume *optimal allocation* is used to sample instances, they ignore the finite population correction, they assume the population distribution of the stratification variable is continuous, and they assume the probability density of the stratification variable within each bin is constant.



(a) Score-based stratification using the CSF method.



(b) Illustration of imputed hierarchy (with branching factor 2) for ordered strata.

Figure 6.1: Stratification for a finite test pool.

After doing so, they obtain a simple rule known as the *cumulative square-root frequency (CSF) method*. In essence, the rule is to choose the bin edges such that the integral of the square-root of the probability density over each bin is approximately equalised.

Figure 6.1a illustrates the CSF method for one of the data sets (`abt-buy`) considered later in our empirical study. The top panel shows a histogram approximation of the distribution of the stratification variable (in this case, the scores). This is then used to compute the cumulative square-root of the frequencies in each bin, as shown in the middle panel. The CSF scale is then divided into equal sub-intervals, which are mapped to bins on the scale of the stratification variable. These bins define an ordered sequence of strata, as depicted in the bottom panel.

Geometric method. [GH04] propose an alternative to the CSF method which is appropriate when the distribution of the stratification variable is highly positively skewed. By assuming that the distribution within each stratum is approximately uniform and requiring that the coefficient of variation is approximately equalised across the strata, they find that the optimal bin edges are equal-width bins in log-space. In other words, the bin edges $v_0 < v_1 < \cdots < v_K$ are the terms of a geometric progression $v_k = v_0(v_K/v_0)^{k/K}$ where v_0 is the minimum of the stratification variable and v_K is the maximum. This method is interesting for entity resolution applications, as we do expect the distribution of the scores to be highly positively skewed (reflecting the abundance of non-matches compared to matches).

Imputing hierarchical structure. Score-based stratification is conventionally used to produce strata without hierarchical structure. However, since the strata are ordered, it is straightforward to "fill in" hierarchical structure as shown in Figure 6.1b. Given some branching factor b and tree depth d, one can select the number of strata K to match the number of leaves b^d of the tree. The ordered strata are then associated with the leaf nodes of the tree in breadth-first order.

6.2.2 Feature-based methods

When scores are not available, unsupervised clustering methods can be used to partition the test pool into strata (clusters) which are expected to be approximately homogeneous. This relies on the fact that neighbouring points in a feature space are likely to elicit a similar response from the oracle. Aggarwal and Reddy [AR14] provide a thorough survey of clustering methods, including hierarchical clustering methods. Of course, one must be aware of the curse of dimensionality which may present an issue for some methods in high-dimensional feature spaces. Most clustering methods are also likely to scale poorly (e.g. quadratically) for large test pools. In this case, a k-d tree [Ben75] may be a more efficient alternative for feature-based stratification.

6.3 Estimators for the asymptotically-optimal policy

In Proposition 5.7, the asymptotically-optimal policy for estimating performance measure G, parameterised by loss function ℓ and mapping g, was determined to be:

$$q^\star(x) = \frac{v(x)}{\int v(x)\,\mathrm{d}x} \quad \text{where} \quad v(x) = p(x)\sqrt{\int \|\mathrm{D}_g(R)\,\ell(x,y)\|_2^2\, p(y|x)\,\mathrm{d}y}, \tag{6.1}$$

where $\mathrm{D}_g(R)$ is the Jacobian of g evaluated at the risk $R = \mathbb{E}[\ell(X,Y)]$. Since this policy depends on the unknown risk R and oracle response $p(y|x)$, we cannot compute it exactly. However, we can compute an estimate using labelled data collected from the oracle. In this section, we propose estimators for $q^\star(x)$ which are functions of estimators for R and $p(y|x)$. We consider deterministic and stochastic oracles separately.

Stochastic oracles. We assume an estimator $\hat{p}(y|x)$ for $p(y|x)$ is available, as well as an estimator $\hat{R}$ for R. The simple plug-in estimator for $q^\star(x)$ is

$$\hat{q}^\star(x) = \frac{\hat{v}(x)}{\int \hat{v}(x)\,\mathrm{d}x} \quad \text{where} \quad \hat{v}(x) = p(x)\sqrt{\int \|\mathrm{D}_g(\hat{R})\,\ell(x,y)\|_2^2\, \hat{p}(y|x)\,\mathrm{d}y}. \tag{6.2}$$

However, the support of this distribution is not guaranteed to satisfy the conditions of our framework (see Theorem 5.4). For example, if the estimated Jacobian $\mathrm{D}_g(\hat{R})$ takes on a particularly unfortunate value, it may cause $\hat{q}^\star(x)$ to vanish at instances x in the required support

$$\{x \in \mathcal{X} \,:\, p(x)p(y|x)\|\ell(x,y)\| \neq 0 \text{ for any } y \in \mathcal{Y}\}. \tag{6.3}$$

We discuss solutions to this problem shortly.

Deterministic oracles. A deterministic oracle is a special case of a stochastic oracle for which the response $p(y|x)$ collapses to a point mass at $y(x)$ for all $x \in \mathcal{X}$. As a result, the expression for $v(x)$ in (6.1) simplifies to $v(x) = p(x)\|\mathrm{D}_g(R)\ell(x, y(x))\|_2$. We assume a posterior distribution $\pi(y|x)$ is available for the unknown oracle response $y(x)$, as well as an estimator $\hat{R}$ for R. In order to capture the uncertainty in $y(x)$, we approximate $q^\star(x)$

using the full posterior rather than a point estimate. Specifically, we approximate $v(x)$ using the expectation with respect to $\pi(y|x)$:

$$\hat{q}^\star(x) = \frac{\hat{v}(x)}{\int \hat{v}(x)\,\mathrm{d}x} \quad \text{where} \quad \hat{v}(x) = p(x) \int \|\mathrm{D}_g(\hat{R})\ell(x, y(x))\|_2 \pi(y|x)\,\mathrm{d}y. \tag{6.4}$$

This estimator, like the plug-in estimator (6.2), may also fail to have the necessary support. In the following two sections, we present corrected estimators which do not suffer from this problem.

6.3.1 Epsilon-greedy estimator

One of the simplest ways of designing an estimator for $q^\star(x)$ with the necessary support, is to use a mixture between the passive proposal $p(x)$ and the plug-in estimator $\hat{q}^\star(x)$. The resulting proposal

$$\hat{q}^\star_\epsilon(x) = (1 - \epsilon)\hat{q}^\star(x) + \epsilon p(x) \tag{6.5}$$

with $0 < \epsilon \leq 1$ inherits the support of $p(x)$, which is a superset of the required support (6.3). Here ϵ can be interpreted as a "greediness" parameter, which controls the trade-off between exploration and exploitation. A small value of ϵ close to 0 encourages exploitation of the current estimate for $q^\star(x)$ (which may be inaccurate), while a large value of ϵ close to 1 encourages exploration (possibly at the expense of short-term label efficiency). This strategy for managing the explore-exploit trade-off is used more generally in online decision making, where it is known as the *epsilon-greedy strategy* [CL06]. We note that a similar approach was suggested in Section 5.8.2 to design a policy that enables sample reuse under different performance measures.

6.3.2 Threshold estimator

The epsilon-greedy estimator is not a particularly refined solution, as it inherits the support of $p(x)$, which may be larger than the required support (6.3). We therefore propose an alternative estimator which applies careful thresholding to correct the support of the plug-in estimators (6.2) and (6.4).

Proposition 6.1. *Let $\hat{R}$ be an estimator for R such that $\|\mathrm{D}_g(\hat{R})\|_2 \leq K < \infty$ and*

- *let $\hat{p}(y|x)$ be an estimator for $p(y|x)$ for a* stochastic oracle *whose support includes the support of $p(y|x)$, or*
- *let $\pi(y|x)$ be a posterior distribution over label $y(x)$ for a* deterministic oracle, *whose support includes $y(x)$.*

Then for any threshold parameter $\epsilon > 0$, the estimator

$$\hat{q}^\star(x) \propto \begin{cases} p(x)\sqrt{\int \max\{\|\mathrm{D}_g(\hat{R})\,\ell(x, y)\|_2^2, \epsilon\mathbb{I}[\|\ell(x, y)\| \neq 0]\}\hat{p}(y|x)\,\mathrm{d}y}, \\ \qquad \textit{for a stochastic oracle,} \\ p(x)\int \max\{\|\mathrm{D}_g(\hat{R})\,\ell(x, y)\|_2, \epsilon\mathbb{I}[\|\ell(x, y)\| \neq 0]\}\pi(y|x)\,\mathrm{d}y, \\ \qquad \textit{for a deterministic oracle,} \end{cases}$$

yields a joint proposal $q(x, y) = \hat{q}^\star(x)p(y|x)$ whose support is a superset of $\{x, y \in \mathcal{X} \times \mathcal{Y} : p(x, y)\|\ell(x, y)\| \neq 0\}$.

Proof. We give a proof for the deterministic case. The proof for the stochastic case follows by a similar argument. Using the previous notation, we write the estimator for the asymptotically-optimal proposal as

$$\hat{q}^\star(x) = \frac{\hat{v}(x)}{\int \hat{v}(x)\,\mathrm{d}x} \quad \text{where}$$
$$\hat{v}(x) = p(x) \int \max\left\{ \|\mathrm{D}_g(\hat{R})\,\ell(x,y)\|_2, \epsilon \mathbb{I}[\|\ell(x,y)\| \neq 0] \right\} \pi(y|x)\,\mathrm{d}y.$$

Observe that

$$\begin{aligned}
\epsilon\, p(x) \int \mathbb{I}[\|\ell(x,y)\| \neq 0]\ \pi(y|x)\,\mathrm{d}y \leq \hat{v}(x) &\leq p(x) \int \{\epsilon + \|\mathrm{D}_g(\hat{R}\}\|_2\, \|\ell(x,y)\|_2) \pi(y|x)\,\mathrm{d}y \\
&\leq p(x) \left(\epsilon + d^2 K \sup_{x,y \in \mathcal{X}\times\mathcal{Y}} \|\ell(x,y)\|_\infty \right) \\
&\leq C p(x)
\end{aligned}$$

where $C < \infty$ is a constant. The upper bound follows from (5.1), the boundedness of ϵ and the boundedness of the Jacobian. Since

$$\int \hat{v}(x)\,\mathrm{d}x \geq \epsilon \iint \mathbb{I}[\|\ell(x,y)\| \neq 0]\, \pi(y|x) p(x)\,\mathrm{d}y\,\mathrm{d}x > 0$$

by assumption and $\hat{v}(x)$ is bounded from above, we conclude that $\hat{q}^\star(x)$ is a valid distribution. The lower bound on $\hat{v}(x)$ implies that the support of $q(x,y) = \hat{q}^\star(x)p(y|x)$ is

$$\{(x,y) \in \mathcal{X}\times\mathcal{Y} \,:\, p(x,y)\pi(y|x)\|\ell(x,y)\| \neq 0\} \supseteq \{(x,y) \in \mathcal{X}\times\mathcal{Y} \,:\, p(x,y)\|\ell(x,y)\| \neq 0\}.$$

The inequality follows from the fact that the support of $\pi(y|x)$ includes the support of $p(y|x)$. Thus $\hat{q}^\star(x)$ has the required support. □

6.3.3 Stratified estimator

The estimators for $q^\star(x)$ we have proposed thus far are highly flexible, in the sense that they allow for a distinct weight to be placed on each instance $x \in \mathcal{X}$. This flexibility is beneficial in principle, as it allows for a better approximation of $q^\star(x)$. However, in practice there is only a small amount of labelled data available to estimate $q^\star(x)$, so a highly flexible estimator may be excessive. In this section, we consider a simpler label-efficient estimator for $q^\star(x)$ based on the stratification principle.

We begin by partitioning the instance space[1] $\mathcal{X}$ into K strata (or blocks) $\mathcal{X} = \bigcup_{k=1}^K \mathcal{X}_k$ indexed by $k \in \{1,\dots,K\}$. This could be done using one of the methods discussed in Section 6.2. Since the instances within the k-th stratum are assumed to be similar, we can approximate $q^\star(x)$ with a stratified estimator:

$$\hat{q}^\star_{\mathrm{st}}(x) = \frac{1}{|\mathcal{X}|} \sum_{k=1}^K \hat{q}_k \mathbb{I}[x \in \mathcal{X}_k] \tag{6.6}$$

[1] Assumed to be compact—e.g. a finite test pool.

which places the same weight $\hat{q}_k^\star$ on all instances $x \in \mathcal{X}_k$. Under this constraint, drawing an instance from $\hat{q}_{\text{st}}^\star(x)$ is equivalent to drawing a stratum according to the probability mass function $\hat{q}_k^\star$, followed by an instance uniformly at random from $\mathcal{X}_k$.

All that remains is to specify an estimator for the distribution $\hat{q}_k^\star$ over the strata. One idea is to select $\hat{q}_k^\star$ such that the KL divergence from the stratified estimator $\hat{q}_{\text{st}}^\star(x)$ to $q^\star(x)$ is minimised. The solution to this minimisation problem is given by

$$\hat{q}_k^\star \propto \int v(x) \mathbb{I}[x \in \mathcal{X}_k] \, \mathrm{d}x.$$

However, the integral over $\mathcal{X}$ is intractable since $v(x)$ as defined in (6.1) is non-linear in x. Hence we take a heuristic approach, approximating $\hat{q}_k^\star$ by $v(x)$ after replacing all quantities that depend on x by their stratum averages:

$$\begin{aligned} p(x) \longrightarrow p_{\text{st}}(k) &= \int p(x) \mathbb{I}[x \in \mathcal{X}_k] \, \mathrm{d}x \\ \ell(x, y) \longrightarrow \ell_{\text{st}}(k, y) &= \int p(x) \ell(x, y) \mathbb{I}[x \in \mathcal{X}_k] \, \mathrm{d}x \\ p(y|x) \longrightarrow p_{\text{st}}(y|k) &= \int p(x) p(y|x) \mathbb{I}[x \in \mathcal{X}_k] \, \mathrm{d}x. \end{aligned}$$

This leads us to propose the following estimator:

$$\hat{q}_k^\star = \frac{\hat{v}_k}{\sum_{k=1}^K \hat{v}_k} \quad \text{with} \quad \hat{v}_k = p_{\text{st}}(k) \sqrt{\int \| \mathrm{D}_g(\hat{R}) \ell_{\text{st}}(k, y) \|_2^2 \hat{p}_{\text{st}}(y|k) \, \mathrm{d}y}, \tag{6.7}$$

where $\hat{R}$ is an estimator for R, and $\hat{p}_{\text{st}}(y|k)$ is an estimator for the stratum-averaged oracle response $p_{\text{st}}(y|k)$. Like the earlier plug-in estimators (6.2) and (6.4), the stratified estimator specified by (6.6) and (6.7) may also fail to have the necessary support to satisfy Theorem 5.4. This issue can be managed by generalising the epsilon-greedy or threshold estimators discussed in the preceding sections.

6.4 Model-based estimators for the oracle response

In the previous section, we introduced estimators for the asymptotically-optimal policy $q^\star(x)$ which depended on unspecified estimators for the oracle response $p(y|x)$ and risk R. In this section, we propose model-based estimators for $p(y|x)$, assuming x takes on values in a test pool $\mathcal{T} = \{x_1, \dots, x_M\}$ and y takes on values in a discrete finite label space $\mathcal{Y} = \{1, \dots, C\}$. Since our primary objective is to conduct evaluation under a limited label budget, we cannot expect to produce refined estimates of $p(y|x)$ at the level of individual instances $x \in \mathcal{T}$. We therefore incorporate stratification in our models, to improve label efficiency.

Concretely, we assume the test pool $\mathcal{T}$ can be partitioned into K disjoint strata $\mathcal{T} = \bigcup_{k=1}^K \mathcal{T}_k$ indexed by $k \in \{1, \dots, K\}$, such that all instances within a stratum elicit a similar oracle response. In other words, we assume $p(y|x)$ is well-approximated by a stratum-level oracle response $p_{\text{st}}(y|k)$ for all $x \in \mathcal{T}_k$. This assumption does not need to be satisfied strictly, since a rough approximation of $p(y|x)$ may yield a "good enough"

approximation of $q^\star(x)$ to significantly improve label efficiency of evaluation. We consider two model variants: (i) in Section 6.4.1 we assume the oracle response in each stratum is independent and (ii) in Section 6.4.2 we assume a hierarchical dependence structure. Both models assume a stochastic oracle response (the most general case), however in Section 6.4.3 we develop a specialised model assuming a deterministic oracle response.

Remark 6.1. *The models must be updated at the end of each round. Produce estimate*

6.4.1 Independent strata: stochastic oracle

In this section, we assume the oracle response is independent across the strata. Specifically, we associate an independently-generated pmf $\psi_k = [\psi_{k1}, \dots, \psi_{kC}]$ over the label space $\mathcal{Y} = \{1, \dots, C\}$ with each stratum k:

$$\psi_k | \alpha_k \overset{\text{ind.}}{\sim} \text{Dirichlet}(\alpha_k), \qquad k \in 1, \dots, K,$$

where $\alpha_k = [\alpha_{k1}, \dots, \alpha_{kC}] \in \mathbb{R}_+^C$ are concentration hyperparameters. Then conditional on ψ_k, the oracle response for an instance x_j drawn uniformly at random from stratum k, is assumed to be generated as follows:

$$y_j | \psi_k, x_j \in \mathcal{T}_k \overset{\text{ind.}}{\sim} \text{Categorical}(\psi_k), \qquad j \in 1, \dots, J.$$

In other words, the oracle response data from each stratum follows a Dirichlet-Categorical model.

The posterior predictive distribution for a new response y_j to query instance $x_j \in \mathcal{T}_k$ conditional on the previously observed data $\mathcal{L}$ is

$$\begin{aligned} p(y_j = y | x_j \in \mathcal{T}_k, \mathcal{L}, \alpha_k) &= \int_\psi p(y_j = y | \psi_k, x_j \in \mathcal{T}_k) p(\psi_k | \alpha_k, \mathcal{L}) = \frac{\tilde{\alpha}_{ky}}{\sum_{y' \in \mathcal{Y}} \tilde{\alpha}_{ky'}}, \\ \text{where} \quad \tilde{\alpha}_{ky} &= \alpha_{ky} + \sum_{(x', y', w') \in \mathcal{L}} \mathbb{I}\left[y' = y\right] \mathbb{I}\left[x' \in \mathcal{T}_k\right]. \end{aligned} \tag{6.8}$$

This expression can be used as an estimator for $p_{\text{st}}(y|k)$. However, it is important to note that (6.8) assumes instances are drawn *uniformly at random* from the strata. Hence, this estimator must be paired with a stratified labelling policy (e.g. the one proposed in Section 6.3.3), which ensures instances are drawn uniformly at random from the strata. If one would like to use a more general policy instead, which biases sampling within the strata (e.g. the policies introduced in Sections 6.3.1 and 6.3.2), it is necessary to apply a bias correction to (6.8):

$$\tilde{\alpha}_{ky} = \alpha_{ky} + \sum_{(x', y', w') \in \mathcal{L}} w' \mathbb{I}\left[y' = y\right] \mathbb{I}\left[x' \in \mathcal{T}_k\right]. \tag{6.9}$$

This correction ensures that $\frac{\tilde{\alpha}_{ky}}{N} \xrightarrow{\text{a.s.}} \mathbb{E}[\mathbb{I}[Y = y] \mathbb{I}[X \in \mathcal{T}_k]]$.

6.4.2 Hierarchical strata: stochastic oracle

We now generalise the independent stratified model to allow for strata with underlying hierarchical structure. This is likely to yield more label-efficient estimates of the oracle response, since a hierarchical model permits sharing of statistical strength across

neighbouring strata. As for the independent stratified model, we assume the test pool is split into K strata $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{T}_k$, such that the oracle response $p(y|x)$ is approximately constant within each stratum. However, we now additionally assume that the strata have an underlying hierarchical structure, and that neighbouring strata in the hierarchy elicit a similar oracle response. We represent the hierarchical structure using a tree T whose leaf nodes correspond to the strata. Unsupervised methods for learning T are discussed in Section 6.2.

Since the strata are no longer independent, we model the oracle response globally using a pmf $\theta = [\theta_1, \dots, \theta_C]$ over the label space $\mathcal{Y}$ with a Dirichlet prior:

$$\theta|\alpha \sim \text{Dirichlet}(\alpha),$$

where $\alpha = [\alpha_1, \dots, \alpha_C] \in \mathbb{R}_+^C$ are concentration hyperparameters. The label y_j for each instance j is then assumed to be generated i.i.d. according to θ:

$$y_j|\theta \overset{\text{iid.}}{\sim} \text{Categorical}(\theta), \qquad j \in 1, \dots, J.$$

Conditional on the hierarchical structure T and label y_j, we assume instance j is assigned to a stratum $k_j \in \{1, \dots, K\}$ according to a distribution ψ_y with a Dirichlet-tree prior [Den96; Min99]:

$$\begin{aligned} \psi_y|\beta_y, T &\overset{\text{ind.}}{\sim} \text{DirichletTree}\big(\beta_y; T\big), && y \in \mathcal{Y}, \\ k_j|y_j, \psi_{y_j} &\overset{\text{ind.}}{\sim} \text{Categorical}\big(\psi_{y_j}\big), && j \in 1, \dots, J. \end{aligned}$$

The Dirichlet-tree distribution is a generalisation of the Dirichlet distribution, which allows for more flexible covariance structures over the categories (strata in this case). Prior information about the covariance structure is encoded in the tree T, whose leaf nodes map to the categories, and in a collection of Dirichlet concentration parameters β_y for each internal node of T.

To estimate the stratum-level oracle response $p_{\text{st}}(y|k)$, we use the posterior predictive distribution

$$p(y_j|k_j, \mathcal{L}) = \int_{\psi} \int_{\theta} p(y_j|k_j, \psi, \theta) p(\psi, \theta|\mathcal{L}),$$

which represents our uncertainty about the oracle response y_j for a query instance x_j from stratum k_j conditional on the previously observed samples $\mathcal{L}$. If the observed samples $\mathcal{L}$ were collected through unbiased sampling (as assumed in the model), we would compute the posterior predictive distribution as follows:

$$\begin{aligned} p(y_j|k_j, \mathcal{L}) &\propto \int_{\theta} p(y_j|\theta) p(\theta|\mathcal{L}) \int_{\psi} p(k_j|y_j, \psi) p(\psi|\mathcal{L}) \\ &\propto \int_{\theta} \theta_{y_j} p(\theta|\mathcal{L}) \int_{\psi} \psi_{y_j, k_j} p(\psi|\mathcal{L}) \\ &\propto \tilde{\alpha}_{y_j} \times \prod_{v \in \text{in}(T)} \prod_{c \in \text{children}(v)} \left(\frac{\tilde{\beta}_{y_j c}}{\sum_{c' \in \text{children}(v)} \tilde{\beta}_{y_j c'}} \right)^{\delta_c(k_j)} \end{aligned} \tag{6.10}$$

where

$$\begin{aligned} \tilde{\alpha}_y &= \alpha_y + \sum_{(x', y', w') \in \mathcal{L}} \mathbb{I}\big[y' = y\big], \\ \tilde{\beta}_{yc} &= \beta_{yc} + \sum_{(x', y', w') \in \mathcal{L}} \mathbb{I}\big[y' = y\big]\, \delta_c(k_{x'}), \end{aligned} \tag{6.11}$$

$\text{in}(T)$ denotes the inner nodes of T, $\text{children}(v)$ denotes the children of node v, k_x denotes the stratum assignment of instance x and

$$\delta_v(k) := \begin{cases} 1, & \text{if node } v \text{ is traversed to reach leaf node } k, \\ 0, & \text{otherwise.} \end{cases} \tag{6.12}$$

However, since the observed samples $\mathcal{L}$ are biased in our application, we must apply a bias-correction to (6.11):

$$\begin{aligned} \tilde{\alpha}_y &= \alpha_y + \sum_{(x',y',w') \in \mathcal{L}} w' \mathbb{I}\left[y' = y\right], \\ \tilde{\beta}_{yc} &= \beta_{yc} + \sum_{(x',y',w') \in \mathcal{L}} w' \mathbb{I}\left[y' = y\right] \delta_c(k_{x'}). \end{aligned}$$

This guarantees that $\frac{\tilde{\alpha}_y}{N} \xrightarrow{\text{a.s.}} \mathbb{E}[\mathbb{I}[Y = y]]$ and $\frac{\tilde{\beta}_{yc}}{N} \xrightarrow{\text{a.s.}} \mathbb{E}[\mathbb{I}[Y = y]\, \delta_c(k_X)]$.

6.4.3 Hierarchical strata: deterministic oracle

In this section, we adapt the hierarchical stratified model introduced in the previous section to incorporate a deterministic constraint on the oracle response. Concretely, we now make the assumption that $p(y|x)$ is a point mass at $y(x)$ for all $x \in \mathcal{T}$. Under this assumption, we make two modifications to the hierarchical stratified model:

(i) we replace index $j \in \{1, \ldots\}$ over the samples with index $i \in \{1, \ldots, M\}$ where $M = |\mathcal{T}|$, since there is now a finite set of instance-label pairs; and

(ii) we introduce an observation process for the unknown deterministic labels.

After applying the first modification, the model Section 6.4.2 becomes:

$$\begin{aligned} \theta|\alpha &\sim \text{Dirichlet}(\alpha), && \\ y_i|\theta &\overset{\text{iid.}}{\sim} \text{Categorical}(\theta), && i \in 1, \ldots, M, \\ \psi_y|\beta_y, T &\overset{\text{ind.}}{\sim} \text{DirichletTree}\left(\beta_y, T\right), && y \in \mathcal{Y}, \\ k_i|y_i, \psi_{y_i} &\overset{\text{ind.}}{\sim} \text{Categorical}\left(\psi_{y_i}\right), && i \in 1, \ldots, M. \end{aligned}$$

This part of the generative process is essentially unchanged: we still assume each label y_i is generated according to a global distribution θ with a Dirichlet prior, and that the corresponding instance i is assigned to a stratum k_i, according to a distribution ψ_{y_i} with a Dirichlet-tree prior. However, we now additionally consider the observation process for the deterministic labels $\mathbf{y} = (y_1, \ldots, y_M)$.

Let $\mathbf{o}_t = (o_{t,1}, \ldots, o_{t,M})$ be observation indicators for the labels $\mathbf{y}$ at the end of stage t of our evaluation framework (see Algorithm 5.1). We initialise $\mathbf{o}_0 = 0$ and define $\mathbf{o}_t$ in the obvious way: $o_{t,i}$ is 1 if item i has been selected by the end of stage t and 0 otherwise. From Algorithm 5.1, the n-th item selected in stage t depends on the labels of the previously observed items $\mathbf{y}_{(\mathbf{o}_{t-1})}$ and the stratum assignments:

$$i_{t,n}|\mathbf{o}_{t-1}, \mathbf{y}_{(\mathbf{o}_{t-1})}, \mathbf{k} \sim q_{t-1}(\mathbf{y}_{(\mathbf{o}_t)}, \mathbf{k}).$$

Our goal is to infer the unobserved labels (oracle response) at each stage t of the evaluation process. We assume the stratum assignments $\mathbf{k} = (k_1, \ldots, k_M)$ are fully observed. Since the observation indicators are independent of the unobserved labels conditional on the observed labels, our model satisfies ignorability [Jae05]. This means we can ignore the observation process when conducting inference.

To estimate the oracle response, we treat the labels $\mathbf{y}$ as partially observed and apply the expectation-maximisation (EM) algorithm. Omitting the dependence on stage t, we let $\mathbf{y}_{(\mathbf{o})}$ denote the observed labels and $\mathbf{y}_{(\neg\mathbf{o})}$ denote the unobserved labels. The EM algorithm returns a distribution over the unobserved labels $\mathbf{y}_{(\neg\mathbf{o})}$ and maximum a posteriori (MAP) estimates of the model parameters $\phi = (\theta, \psi)$. At each iteration τ of the EM algorithm, the following two steps are applied:

- **E-step**. Compute the function

$$Q(\phi|\phi^{(\tau)}) = \mathbb{E}_{\mathbf{y}_{(\neg\mathbf{o})}|\mathbf{y}_\mathbf{o},\mathbf{k},\phi^{(\tau)}}\left(\log p(\phi|\mathbf{y},\mathbf{k})\right), \tag{6.13}$$

  which is the expected log posterior with respect to the current distribution over the unobserved labels $\mathbf{y}_{(\neg\mathbf{o})}$, conditional on the observed labels $\mathbf{y}_{(\mathbf{o})}$ and the current parameter estimates $\phi^{(\tau)}$.

- **M-step**. Update the parameter estimates by maximising Q:

$$\phi^{(\tau+1)} \in \arg\max_{\phi} Q(\phi|\phi^{(\tau)}). \tag{6.14}$$

In order to implement the E- and M-steps, we must evaluate the Q function for our model. Since the Dirichlet prior on θ and Dirichlet-tree priors on ϕ_y are conjugate to the categorical distribution, the posterior $p(\phi|\mathbf{y},\mathbf{k})$ is straightforward to compute. We have

$$\theta|\mathbf{y},\alpha \sim \text{Dirichlet}(\tilde{\alpha}),$$
$$\psi_y|\mathbf{y},\mathbf{k},\beta_y,T \sim \text{DirichletTree}(\tilde{\beta}_y,T),$$

where $\tilde{\alpha}_y = \alpha_y + \sum_{i=1}^{M} \mathbb{I}[y_i = y]$, $\tilde{\beta}_{yv} = \beta_{yv} + \sum_{i=1}^{M} \mathbb{I}[y_i = y]\,\delta_v(k_i)$ and $\delta_v(k)$ is defined in (6.12). The posterior density for θ is

$$p(\theta|\mathbf{y},\alpha) \propto \prod_{y=1}^{C} \theta_y^{\tilde{\alpha}_y - 1}.$$

Minka [Min99] gives the density for ψ_y as:

$$p(\psi_y|\mathbf{y},\mathbf{k},\beta_y,T) \propto \prod_{k\in\text{lv}(T)} \psi_{yk}^{\tilde{\beta}_{yk}-1} \prod_{v\in\text{in}(T)} \left(\sum_{k\in\text{lv}(T)} \sum_{v'\in\text{children}(v)} \delta_{v'}(k)\psi_{yk}\right)^{\gamma_{yv}},$$

where $\text{lv}(T)$ denotes the set of leaf nodes in T, $\text{in}(T)$ denotes the set of inner nodes in T, $\text{lv}(v)$ denotes the leaf nodes reachable from node v, and $\tilde{\gamma}_{yv} = \tilde{\beta}_{yv} - \sum_{c\in\text{children}(v)} \tilde{\beta}_{yc}$.

Substituting the posterior densities in (6.13), we have

$$Q(\phi|\phi^{(t)}) = \sum_{y\in\mathcal{Y}} \sum_{k\in\text{lv(T)}} \left(\tilde{\beta}_{yk}^{(\tau)} - 1\right) \log\psi_{ky} + \sum_{y\in\mathcal{Y}} \sum_{v\in\text{in}(T)} \tilde{\gamma}_{yv}^{(\tau)} \log\left(\sum_{k\in\text{lv}(T)} \sum_{v'\in\text{children}(v)} \delta_{v'}(k)\psi_{yk}\right)$$
$$+ \sum_{y\in\mathcal{Y}} \left(\tilde{\alpha}_y^{(\tau)} - 1\right) \log\theta_y + \text{const.}$$

where we define $\tilde{\beta}_{yk}^{(\tau)} = \mathbb{E}_{\mathbf{y}_{(\neg \mathbf{o})}|\mathbf{y}_\mathbf{o},\mathbf{k},\phi^{(\tau)}}[\tilde{\beta}_{yk}]$ and similarly for $\tilde{\alpha}_y^{(\tau)}$ and $\tilde{\gamma}_{yv}^{(\tau)}$. When maximising $Q(\phi|\phi^{(t)})$ with respect to ϕ, we must obey the constraints:

- $\theta_y > 0$ for all $y \in \mathcal{Y}$,
- $\sum_{y\in\mathcal{Y}} \theta_y = 1$,
- $\psi_{yk} > 0$ for all $y \in \mathcal{Y}$ and leaf nodes $k \in \text{lv}(T)$, and
- $\sum_{k\in\text{lv}(T)} \psi_{yk} = 1$.

If we additionally assume that the hyperparameters satisfy $\beta_{yc} \geq 1$ and $\alpha_y \geq 1$ for all y, c, then the optimisation problem is convex. We can maximise θ and $\{\psi_y\}$ separately since they are independent. For θ_y we have the mode of a Dirichlet random variable:

$$\theta_y^{(\tau+1)} = \frac{\tilde{\alpha}_y^{(\tau)} - 1}{\sum_{y'}\{\tilde{\alpha}_{y'}^{(\tau)} - 1\}}$$

and for ψ_y we have (see [Min99]):

$$\psi_{yk}^{(\tau+1)} = \prod_{v\in\text{in}(T)} \prod_{c\in\text{children}(v)} \left(b_{yc}^{(\tau+1)}\right)^{\delta_c(k)} \tag{6.15}$$

$$\text{where} \quad b_{yc}^{(\tau+1)} = \frac{\tilde{\beta}_{yc}^{(\tau)} - \sum_{k\in\text{lv}(T)} \delta_c(k)}{\sum_{c'\in\text{siblings}(c)\cup\{c\}}\{\tilde{\beta}_{yc'}^{(\tau)} - \sum_{k\in\text{lv}(T)} \delta_{c'}(k)\}}. \tag{6.16}$$

The parameters $\{b_{yc} : c \in \text{children}(v)\}$ may be interpreted as *branching probabilities* for node $v \in \text{in}(T)$.

In summary, the EM algorithm reduces to the following two steps:

- **E-step**. Compute the expected value for each unobserved label using $\phi^{(\tau)}$:

$$\mathbb{E}\left[\mathbb{I}\left[y_j = y\right]\middle| k_j, \phi^{(\tau)}\right] = \frac{\psi_{yk_j}^{(\tau)}\theta_y^{(\tau)}}{\sum_{y'\in\mathcal{Y}} \psi_{y'k_j}^{(\tau)}\theta_{y'}^{(\tau)}}. \tag{6.17}$$

  Then make a backward pass through the tree, computing $\tilde{\beta}_{yv}^{(\tau)}$ at each internal node $v \in \text{in}(T)$.

- **M-step**. Make a forward-pass through the tree, updating the branch probabilities $b_{yc}^{(\tau+1)}$ using (6.16). Compute $\psi_y^{(\tau+1)}$ at the same time using (6.15).

We can interpret (6.17) as providing a posterior estimate for the unknown oracle response: $\pi(y|x) \propto \psi_{yk_x}^{(\tau)}\theta_y^{(\tau)}$ where k_x denotes the assigned stratum for instance x. If the response $y(x)$ for instance x has been observed in a previous stage of the evaluation process, then $\pi(y|x)$ collapses to a point mass at $y(x)$.

Table 6.1: Key differences between the adaptive labelling policies.

Name	Supported oracle type	Oracle estimator	Proposal estimator
`IStoch`	Stochastic (incl. deterministic)	Independent stratified §6.4.1	Stratified §6.3.3 and epsilon-greedy §6.3.1
`HStoch`	Stochastic (incl. deterministic)	Hierarchical stratified §6.4.2	Threshold §6.3.2
`HDet`	Deterministic	Hierarchical stratified (deterministic) §6.4.3	Threshold §6.3.2

6.5 Adaptive labelling policies

In this section, we combine estimators for the asymptotically-optimal labelling policy (from Section 6.3) and estimators for the oracle response (from Section 6.4) to design practical adaptive labelling policies. We propose three policies for use in different scenarios, which are summarised in Table 6.1. All three policies satisfy the conditions of our evaluation framework, as stated in the proposition below. We provide proofs of the proposition for each policy in the corresponding subsections.

Proposition 6.2. *Let $\hat{R}_t$ be an estimate of the risk R at stage t of the evaluation process. Suppose the estimated Jacobian $\mathrm{D}_g(\hat{R}_t)$ is bounded for all $t \geq 0$.*[2] *Then the three adaptive labelling policies presented in this section satisfy Theorem 5.4 (consistency of the performance estimate) and Theorem 5.5 (central limit theorem for the performance estimate). Furthermore, the* `HDet` *policy is asymptotically-optimal.*

6.5.1 `IStoch`: a stratified policy for stochastic oracles

This policy is intended to be simple to implement and computationally inexpensive. It is appropriate for stochastic oracles. At each stage t of the evaluation process, the proposal $q_t(x)$ is updated using the stratified estimator for the asymptotically-optimal policy presented in Section 6.3.3. To ensure $q_t(x)$ has the necessary support, an epsilon-greedy correction is applied, as discussed in Section 6.3.1. The resulting proposal $q_t(x)$ depends on:

- a partition of the test pool into K strata $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{T}_k$,
- greediness parameter $0 < \epsilon_t \leq 1$,
- an estimate of the stratum-level oracle response $\hat{p}_{\mathrm{st};t}(y|k)$, and
- an estimate of the risk $\hat{R}_t$.

The stratification design and greediness parameter are left unspecified, to allow some flexibility. The stratum-level oracle response is estimated using the independent stratified model presented in Section 6.4.1, and the risk is estimated using the AIS estimator defined in (5.5).

[2] This condition can be relaxed by clipping $\mathrm{D}_g(\hat{R}_t)$, however then the asymptotic optimality result does not necessarily follow.

Since this policy is fully stratified, it is only necessary to compute quantities at the stratum-level, rather than at the level of individual instances. This reduces the complexity of the policy updates, however label-efficiency may suffer since the resulting policy may be further from optimality.

Proof of Proposition 6.2. The proof is an application of Corollary 5.6. For the first part of the corollary, we must check the support of $q_t(x)$. It is straightforward to show that $q_t(x)$ has support on the entire test pool $\mathcal{T}$ for all t, since the mixture $q_t(x) = (1-\epsilon_t)\hat{q}^\star_t(x)+\epsilon_t p(x)$ inherits the support of $p(x)$ (a discrete uniform distribution over $\mathcal{T}$). Thus consistency of the AIS estimators follows.

For the second part of the corollary, we are required to prove that the sequence of proposals $\{q_t(x)\}$ converges a.s. pointwise in x to a constant proposal. This is equivalent to proving that the stratum weights

$$\hat{q}_{k;t} = \frac{\hat{v}_{k;t}}{\sum_{k=1}^{K} \hat{v}_{k;t}} \quad \text{where} \quad \hat{v}_{k;t} = p_{\text{st}}(k) \sqrt{\sum_{y \in \mathcal{Y}} \|\mathrm{D}_g(\hat{R}_t)\ell_{\text{st}}(k,y)\|_2^2 \hat{p}_{\text{st};t}(y|k)},$$

converge a.s. pointwise in k (see 6.7). By the strong law of large numbers, the posterior parameters $\tilde{\alpha}_{ky}/N$ (as defined in 6.8) converge a.s. to $\mathbb{E}[\mathbb{I}[Y = y]\,\mathbb{I}[X \in \mathcal{T}_k]]$. Thus, by application of the continuous mapping theorem we have that $\hat{p}_{\text{st};t}(y|k)$ converges a.s. to $p_{\text{st}}(y|k) = \mathbb{E}[\mathbb{I}[Y = y]\,|X \in \mathcal{T}_k]$. Also, by the first part of the corollary $\hat{R}_t \xrightarrow{\text{a.s.}} R$. Successive application of the continuous mapping theorem therefore implies:

$$\hat{q}_{k;t} \xrightarrow{\text{a.s.}} \frac{v_k}{\sum_{k=1}^{K} v_k} \quad \text{where} \quad v_k = p_{\text{st}}(k) \sqrt{\sum_{y \in \mathcal{Y}} \|\mathrm{D}_g(R)\ell_{\text{st}}(k,y)\|_2^2 p_{\text{st}}(y|k)}$$

which is independent of t. □

6.5.2 `HStoch`: a policy for stochastic oracles

This policy is a more sophisticated alternative to `IStoch`, which is also appropriate for stochastic oracles. At each stage t of the evaluation process, the proposal $q_t(x)$ is updated using the threshold estimator for the asymptotically-optimal policy presented in Section 6.3.2. This yields a more fine-grained proposal than `IStoch`, as it allows for a distinct weight on each instance $x \in \mathcal{T}$. The resulting proposal q_t depends on:

- threshold parameter $\epsilon_t > 0$,
- an estimate of the oracle response $\hat{p}_t(y|x)$, and
- an estimate of the risk $\hat{R}_t$.

The threshold parameter is set to $\epsilon_t = \epsilon_0/(t+1)$, where $\epsilon_0 > 0$ is a user-specified parameter. This diminishes the effect of thresholding in later stages of evaluation, allowing for a better approximation of the asymptotically-optimal policy. The oracle response is estimated using the hierarchical stratified estimator presented in Section 6.4.2. This estimator depends on a user-specified hierarchical stratification of the test pool. It is expected to be more label-efficient than the independent stratified estimator used in `IStoch`, as it

exploits similarities between neighbouring strata. Finally, the risk is estimated using the AIS estimator defined in (5.5).

Since this policy is more refined than `IStoch`, we expect it to yield a better approximation of the asymptotically-optimal policy. However, it comes at the cost of increased computational complexity. For instance, when updating the estimated oracle response $\hat{p}_t(y|x)$ after observing a new label from the oracle, it is necessary to perform a forward and backward pass through the entire hierarchy of strata. Moreover, since the proposal $q_t(x)$ is not stratified, sampling instances from $q_t(x)$ scales linearly in the size of the test pool, rather than linearly in the number of strata.

Proof of Proposition 6.2. As for `IStoch`, the result follows by application of Corollary 5.6. For the first part of the corollary, we check the support of $q_t(x)$. Proposition 6.1 provides conditions under which $q_t(x)$ (the threshold estimator) has the necessary support. These conditions are satisfied, since the estimated Jacobian $\mathrm{D}_g(\hat{R}_t)$ is assumed to be finite for all t. Furthermore, the estimator for the oracle response $\hat{p}_t(y|x)$ has support on the entirety of $\mathcal{Y}$ for all t. This is due to the priors on θ and ψ which ensure $\hat{p}_t(y|x) > 0$ for all $y \in \mathcal{Y}$ (see 6.10). Thus consistency of the AIS estimators follows.

For the second part of the corollary, we must verify that the sequence of proposals $\{q_t(x)\}$ converge a.s. to a constant proposal pointwise in x. At the end of Section 6.4.2, we claim that the posterior parameters $\tilde{\alpha}_y/N$ and $\tilde{\beta}_{yc}/N$ converge a.s. to constants. This follows from the first part of the corollary. When combined with the continuous mapping theorem, this result implies that $\hat{p}_t(y|x) \xrightarrow{\text{a.s.}} u(y|x)$, where $u(y|x)$ is a conditional pmf over $\mathcal{Y}$ that is independent of t. We also have that $\hat{R}_t \xrightarrow{\text{a.s.}} R$ by the first part of the corollary. Successive application of the continuous mapping theorem therefore implies that

$$q_t(x) \xrightarrow{\text{a.s.}} \frac{v(x)}{\sum_{x \in \mathcal{X}} v(x)} \quad \text{where} \quad v(x) = p(x) \sqrt{\sum_{y \in \mathcal{Y}} \|\mathrm{D}_g(R)\, \ell(x, y)\|_2^2 u(y|x)},$$

which is independent of t. □

6.5.3 `HDet`: a policy for deterministic oracles

This policy is a variant of `HStoch` optimised for deterministic oracles. It differs from `HStoch`, in that it uses deterministic variants of the estimators for the asymptotically-optimal policy and the oracle response. The resulting proposal q_t depends on:

- the threshold parameter $\epsilon_t > 0$,
- a posterior for the deterministic oracle response $\pi_t(y|x)$, and
- an estimate of the risk $\hat{R}_t$.

There are some minor differences in how these parameters are configured compared to `HStoch`. The threshold parameter is set to $\epsilon_t = \epsilon_0(1 - \frac{1}{M}\sum_{i=1}^{M} o_{t,i})$, where $\epsilon_0 > 0$ is a user-specified parameter and $o_{t,i}$ is an indicator variable that records whether the oracle response for instance i has been observed at stage t. Once all responses are observed $\epsilon_t = 0$ and no thresholding is applied. The posterior oracle response $\pi_t(y|x)$ is estimated using

the deterministic variant of the hierarchical stratified model, presented in Section 6.4.3. This is also used to estimate the risk:

$$\hat{R}_t = \frac{1}{M} \sum_{i=1}^{M} \sum_{y \in \mathcal{Y}} \pi_t(y|x_i) \ell(x_i, y).$$

We note that HDet is likely to be more computationally intensive than HStoch, as estimating $\pi_t(y|x)$ using the EM algorithm may require many iterations to converge.

Proof of Proposition 6.2. As for the other proofs, we obtain the required result by applying Corollary 5.6. For the first part of the corollary, we check the support of $q_t(x)$. Proposition 6.1 provides conditions under which $q_t(x)$ (the threshold estimator) has the necessary support. We note that the estimated Jacobian remains finite for all t by assumption. Furthermore, the posterior oracle response $\pi_t(y|x)$ has the true label $y(x)$ in its support for all t, since the priors on θ and ψ ensure $\theta_y > 0$ and $\psi_{ky} > 0$ for all $k \in \{1, \dots, K\}$ and $y \in \mathcal{Y}$ (see 6.17). Once the label for instance x is observed, the posterior degenerates to a point mass at the true value $y(x)$. Thus consistency of the AIS estimators follows.

For the second part of the corollary, we examine the convergence of the sequence of proposals $\{q_t(x)\}$. Since

(i) the sequence of proposals ensure all instances x satisfying $p(x)\|\ell(x, y)\| \neq 0$ are contained in the support until they are observed;

(ii) the posterior $\pi_t(y|x)$ degenerates to a point mass on $y(x)$ once the label for instance x is observed; and

(iii) $\epsilon_t = 0$ once all labels are observed;

we have $\hat{R}_t = R$ and $q_t(x) = q^\star(x)$ for sufficiently large t. Thus the second part of the corollary holds. In addition, we have demonstrated that HDet is *asymptotically-optimal*, since it converges to $q^\star(x)$. □

6.6 Empirical study

We are now ready to assess the effectiveness of our label-efficient evaluation framework using the adaptive labelling policies outlined in the previous section. Our objectives are:

(i) to compare the label efficiency of our AIS-based framework (under the IStoch, HStoch and HDet policies) with baseline approaches;

(ii) to compare the gains in label efficiency for different target performance measures (accuracy, F1-score and precision-recall curves); and

(iii) to assess the sensitivity of our approach with respect to variations in the stratification design and prior specification.

Table 6.2: Summary of the data sets used to simulate evaluation tasks. The imbalance ratio is the ratio of majority class instances to minority class instances.

Name	Domain	Size[3]	Imb. ratio
abt-buy	Entity resolution	1,180,452	1075
amzn-goog	Entity resolution	4,397,038	3381
dblp-acm	Entity resolution	5,998,880	2697
restaurant	Entity resolution	745,632	3328
safedriver	Risk analysis	595,212	26.44
creditcard	Fraud detection	284,807	577.9
tweets100k	Sentiment analysis	100,000	1

We focus primarily on evaluation of entity resolution (ER) systems, where class imbalance is extreme and label-efficient methods are expected to be most beneficial. However, we also consider evaluation tasks from other domains—both to highlight the generality of our framework, and to assess the magnitude of efficiency gains in circumstances where the class imbalance is less severe.

Section 6.6.1 describes how realistic evaluation tasks were prepared using publicly-available data. Details about the experimental setup—including baseline approaches and parameter settings—are provided in Section 6.6.2. Results are presented and discussed in Section 6.6.3.

6.6.1 Preparation of evaluation tasks

We prepare a variety of evaluation tasks using publicly-available labelled data sets. For each evaluation task, we set aside a random subset of the data to serve as an unlabelled test pool. The remaining data is used to train a classifier, which serves as the system under evaluation. The ground truth labels included with each data set (one label for each instance) are used to simulate a deterministic oracle. Further details about the data sets and setup are provided below.

Data sets. We experiment with four entity resolution (ER) data sets and three data sets from other domains, as summarized in Table 6.2. `abt-buy`, `amzn-googl`, `dblp-acm` [KTR10] and `restaurant` [Bil] are benchmark data sets for ER. Each data set contains records from two sources, and the goal is to predict whether a pair of records refer to the same entity or not. The entities in `abt-buy` and `amzn-googl` are products sold on e-commerce websites, the entities in `dblp-acm` are computer science publications, and the entities in `restaurant` are restaurants. The remaining three data sets in Table 6.2 are from outside the ER domain. `safedriver` contains anonymised records from a car insurance company, and the task is to predict drivers who are likely to make a claim [Por17]. `creditcard` relates to fraud detection for online credit card transactions [Poz+15]. `tweets100k` contains a selection of tweets from Twitter, and the goal is to predict whether the sentiment of a tweet is positive or negative [Moz+14].

[3]For the data sets from the entity resolution domain the "size" refers to the total number of record pairs, not the total number of records.

Table 6.3: Summary of unlabelled test pools and the true performance of the classifier for each pool, assuming all labels are known.

				Unknown true performance			
Name	Size	Imb. ratio	Classifier	Precision	Recall	F1-score	Accuracy
abt-buy	53,753	1075	SVM	0.917	0.440	0.595	0.999
amzn-goog	676,267	3381	SVM	0.597	0.185	0.282	1.000
dblp-acm	53,946	2697	SVM	1.000	0.900	0.947	1.000
restaurant	149,747	3328	SVM	0.909	0.889	0.899	1.000
safedriver	178,564	26.56	XGB	0.055	0.565	0.100	0.629
creditcard	85,443	580.2	LR	0.883	0.619	0.728	0.999
tweets100k	20,000	0.990	SVM	0.762	0.778	0.770	0.767

Test pool and system. For each data set, we reserve a random fraction of the complete data to create an unlabelled test pool $\mathcal{T}$. We vary the fraction of data reserved for the test pool from approximately 1 per cent to 30 per cent. The size and class imbalance ratio of each test pool are provided in Table 6.3. The data not in $\mathcal{T}$ is used to train a supervised binary classifier, based on a linear support vector machine (SVM), ℓ_2-regularised logistic regression (LR) or gradient-boosted trees (XGB) [CG16]. Since we are interested in simulating a variety of evaluation tasks, we do not always aim for the best performing classifier—instead we aim for a range of performances from poor to excellent as indicated in Table 6.3. While we use supervised methods here for simplicity, semi-supervised or unsupervised methods might be used in practice if labelled data is scarce. We note that the learning paradigm used to prepare the classifier/system has no bearing on the evaluation methodology.

6.6.2 Setup

Evaluation methods. We consider three variants of our AIS-based framework corresponding to the three adaptive labelling policies presented in Section 6.5:

- `AIS-IStoch`: our framework with the `IStoch` adaptive labelling policy for stochastic/deterministic oracles (see §6.5.1).
- `AIS-HStoch`: our framework with the `HStoch` adaptive labelling policy for stochastic/deterministic oracles (see §6.5.2).
- `AIS-HDet`: our framework with the `HDet` adaptive labelling policy for deterministic oracles (see §6.5.3).

We fix the batch size (N_t in Algorithm 5.1) to 50 samples for all of the AIS variants. While smaller batch sizes are thought to be more efficient [DP18], a batch size of 50 is a reasonable compromise, as it allows for parallelisation of labelling. In addition to the AIS variants, we consider the following baseline approaches:

- `Passive`: passive (uniform) sampling as specified in Section 5.4.1.
- `Stratified`: an online variant of stratified sampling with proportional allocation, as used in [DM11]. The test pool is partitioned into K disjoint strata $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{T}_k$,

and items are sampled for labelling one-at-a-time without replacement in proportion to the size of the allocated stratum. The target performance measure G is estimated using the following stratified estimator:

$$\hat{G}^{\text{st}}_{\mathscr{L}} = g(\hat{R}^{\text{st}}_{\mathscr{L}}) \quad \text{with} \quad \hat{R}^{\text{st}}_{\mathscr{L}} = \sum_{k=1}^{K} \frac{|\mathcal{T}_k|}{|\mathcal{T}|} \frac{\sum_{(x,y)\in\mathscr{L}} \ell(x,y)\mathbb{I}[x \in \mathcal{T}_k]}{\sum_{(x,y)\in\mathscr{L}} \mathbb{I}[x \in \mathcal{T}_k]}, \tag{6.18}$$

where $\mathscr{L}$ denotes the set of labelled samples.

- `IS`: static importance sampling as specified in Section 5.4.2. We set the static proposal using the threshold estimator for the asymptotically-optimal proposal as presented in Section 6.3.2 with $\epsilon = 10^{-9}$. The oracle response $p(y|x)$ is estimated using the normalized classifier scores (see paragraph below).

Repetitions and label budget. Since the evaluation process is randomised, we repeat each evaluation task 1000 times for each method. We report the mean behaviour for all quantities of interest (e.g. squared error of the estimated performance measure) with 95 per cent bootstrap confidence intervals. For some experiments, we report quantities of interest assuming a fixed label budget (e.g. 1000 labels), while for other experiments we consider a range of label budgets. We define the consumed label budget to be the number of oracle queries for *distinct* items in the test pool. In other words, if the label for an item in the test pool is queried more than once, we only count the *first* query, as subsequent queries to a deterministic oracle will always return the same label.

Target performance measures. Since most of the data sets are severely imbalanced, we focus primarily on the F1-score, which is insensitive to the number of correct predictions for majority class instances (commonly known as true negatives). Out of interest, we also estimate accuracy to see whether the biased sampling methods offer improved label efficiency compared to passive sampling (see asymptotic analysis in Example 5.2). Finally, we illustrate the ability of our framework to handle vector measures by estimating precision-recall curves with respect to a grid of score thresholds $\tau_1 < \tau_2 < \cdots < \tau_K$. Let $s(1|x)$ denote the real-valued classifier score for item x. The larger the score, the higher the confidence that the true label $y = 1$ (positive).[4] We define a vector loss function which measures whether instance (x, y) is a predicted positive for each score threshold (the first L entries), a true positive for each threshold (the next L entries) and/or a positive (the last entry):

$$\ell(x,y) = [\mathbb{I}[s(1|x) \geq \tau_1], \ldots, \mathbb{I}[s(1|x) \geq \tau_L], y\mathbb{I}[s(1|x) \geq \tau_1], \ldots, y\mathbb{I}[s(1|x) \geq \tau_L], y]^\intercal .$$

To compute the precision-recall curve, we define the following mapping function:

$$G = g(R) = \left[\frac{R_{L+1}}{R_1}, \ldots, \frac{R_{2LK}}{R_L}, \frac{R_{L+1}}{R_{2L+1}}, \ldots, \frac{R_{2L}}{R_{2L+1}}\right]^\intercal . \tag{6.19}$$

The first L entries of G contain the precision at each threshold in ascending order, and the last L entries contain the recall at each threshold in ascending order.

[4] We assume positive labels are encoded as '1' and a negative labels are encoded as '0'.

Normalised classifier scores. The IS and AIS-based evaluation methods require estimates of the oracle response $p(y|x)$ in order to target the asymptotically-optimal policy. We obtain rough estimates of $p(y|x)$ by leveraging information from the classifiers under evaluation. Probabilistic classifiers (e.g. logistic regression) naturally provide estimates of $p(y|x)$, so no further processing is required. Most non-probabilistic classifiers (e.g. SVMs) output real-valued scores which are correlated with $p(y|x)$. To obtain a rough estimate of $p(y|x)$, we normalise the real-valued score $s(y|x)$ assigned to label $y \in \mathcal{Y}$ for item $x \in \mathcal{T}$ using the softmax function.

Hyperparameter settings. Recall that the three AIS variants incorporate online models of the oracle response, which depend on several hyperparameters. All of the models assume that the test pool is partitioned into K strata $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{T}_k$. We apply the cumulative square-root frequency (CSF) method to determine the stratum allocations, using the normalised classifier score for the minority class as the stratification variable. Unless otherwise specified, we set $K = 256$. For the hierarchical models (`AIS-HDet` and `AIS-HStoch`) we fill in the hierarchical tree structure T for a given depth d and branching factor b as detailed in Section 6.2.1. We fix $b = 2$ and consider two depths: a shallow model with $d = 1$ (`AIS-HDet-1`) and a deep model with $d = 8$ (`AIS-HDet-8`).

Finally, we set the Dirichlet concentration parameters for the independent (§6.4.1) and hierarchical (§6.4.2 and §6.4.3) models using the normalised classifier scores. Let $\tilde{s}(y|k)$ denote the mean normalised classifier score for label y in stratum k and let $\xi > 0$ be a smoothing parameter which is set to 1 unless otherwise specified. For the independent model, we set $\alpha_y = \xi + \tilde{s}(y|k)$. For the hierarchical model, we set

$$\alpha_y = \xi + \sum_{k=1}^{K} \tilde{s}(y|k)$$

$$\beta_{yv} = \text{depth}(v)^2 + \sum_{k=1}^{K} \delta_v(k) \left(\xi + \tilde{s}(y|k)\right)$$

where v denotes a non-root node of the tree T, $\text{depth}(v)$ denotes the depth of node v in T, and $\delta_v(k)$ is defined in (6.12).

6.6.3 Results

Estimating F1-score. Figure 6.2 presents convergence plots for the estimated F1-score for each evaluation method and data set as a function of the label budget. The biased sampling methods (`AIS-HDet-8`, `AIS-HDet-1`, `AIS-IStoch` and `IS`) converge significantly more rapidly than `Passive` and `Stratified` for all of the datasets except `tweets100k`.

Our AIS-based methods generally outperform `IS` (excluding `creditcard`), however `IS` is often competitive, particularly when the classifier is probabilistic (`creditcard`) and highly accurate (`dblp-acm`). There is no clear winner among the AIS variants, however we expect `AIS-HDet` to be more reliable in general as it produces a closer approximation to the asymptotically-optimal proposal. We expect `AIS-HDet-8` (based on a deeper hierarchical model) to perform best when $p(y|x)$ varies smoothly between neighbouring strata. Finally, we note that there is no significant difference in label efficiency between the evaluation methods for `tweets100k`. This is expected, since it is the only data set with well-balanced classes.

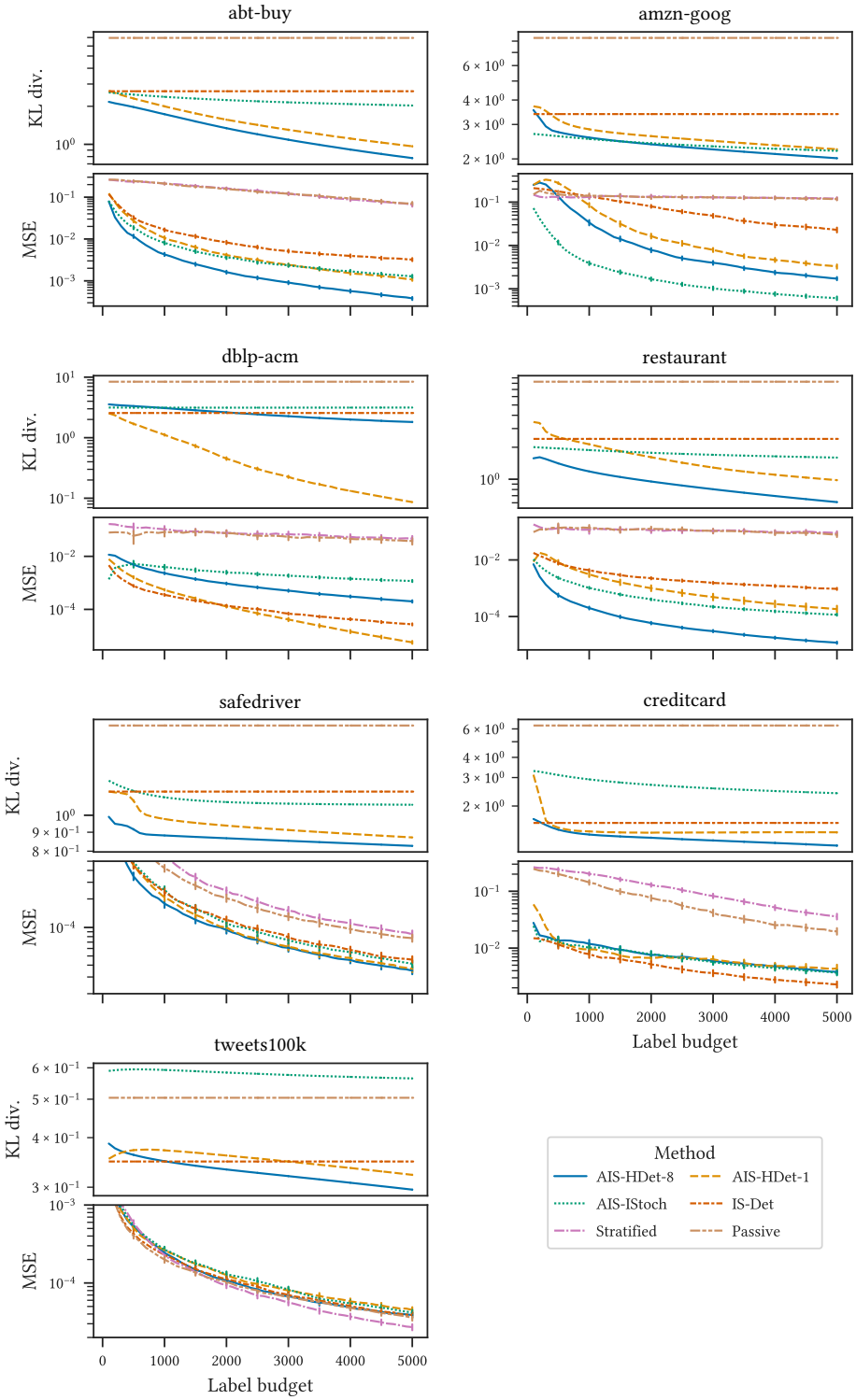


Figure 6.2: Convergence plots for estimating F1-score for several data sets and evaluation methods. The upper panel of each sub-figure plots the mean KL divergence from the proposal to the asymptotically-optimal one. The lower panel plots the mean-squared error of the estimated F1-score. The mean and 95% confidence intervals are computed over 1000 repeats.

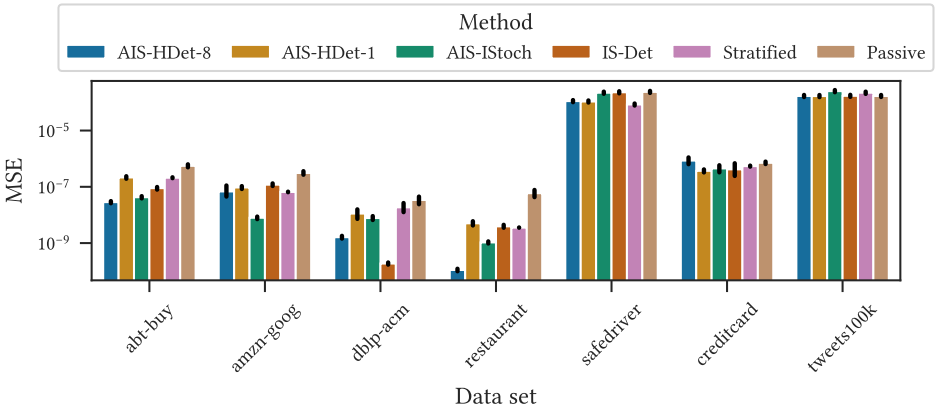


Figure 6.3: Mean-squared error of the estimated accuracy (lower is better) for several data sets and evaluation methods, assuming a label budget of 1000. 95% bootstrap confidence intervals are shown in black.

Estimating accuracy. Figure 6.3 presents the mean-squared error (MSE) of the estimated accuracy for all evaluation methods and data sets, assuming a fixed label budget of 1000. We find that gains in label efficiency are less pronounced when compared to the results for F1-score. However, there is still a marked improvement in the MSE for the biased sampling methods—by an order of magnitude or more—for the four data sets with severe class imbalance. Again, we find that efficiency gains are less pronounced (or non-existent) for data sets with less severe class imbalance. This agrees with the asymptotic analysis of efficiency in Example 5.2.

Estimating precision-recall curves. We estimate precision-recall curves on a uniform grid of score thresholds $\tau_1 < \cdots < \tau_L$, as defined in (6.19). We set τ_1 to be the minimum classifier score in the test pool, τ_L to be the maximum score in the test pool and $L = 2^{10}$. We used the same grid to stratify the test pool into $K = 256$ strata, by associating each stratum with four neighbouring bins on the grid. This was used in place of the CSF stratification method.

Figure 6.4 presents the MSE of the estimated precision-recall curve for three of the test pools in Table 6.3 assuming a label budget of 5000. We find that the biased sampling methods (`AIS-HDet-8`, `AIS-HDet-1` and `IS`) offer a significant improvement in the MSE compared to `Passive` and `Stratified`—by 1–2 orders of magnitude. The difference in the MSE between the AIS-based methods and `IS` is less pronounced here than when estimating F1-score.

To illustrate the vast improvement of `AIS-HDet-8` and `IS` over `Passive`, we present samples of the estimated precision-recall curves for each method in Figure 6.5. The curves relate to `abt-buy` and are estimated using a label budget of 5000. `AIS-HDet-8` and `IS` produce estimates that are quite reliable for selecting an operating threshold—the curves appear to be unbiased and vary minimally about the true red curve. The same cannot be said for estimates produced by `Passive`, which are not practically useful due to high variance.

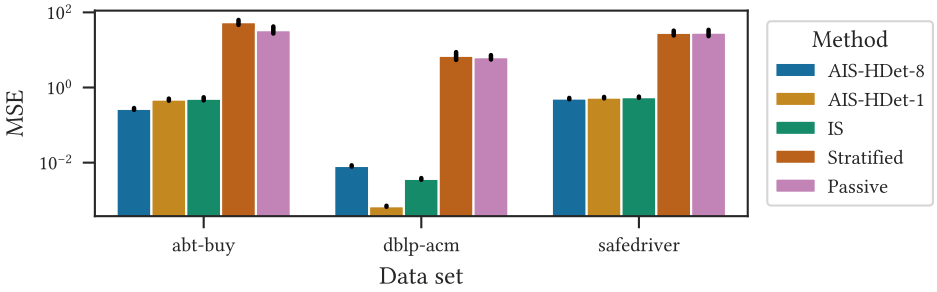


Figure 6.4: Mean-squared error of the estimated precision-recall curve (lower is better) for several data sets and evaluation methods, assuming a label budget of 5000. 95% bootstrap confidence intervals are shown in black.

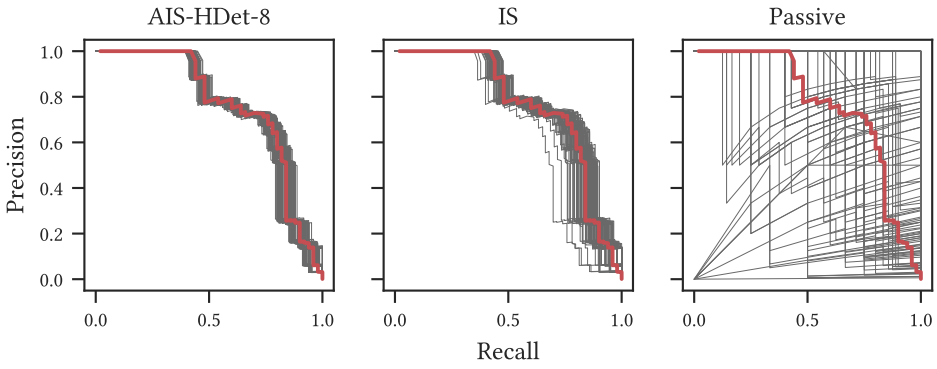


Figure 6.5: Estimated precision-recall curves for `abt-buy` produced by three evaluation methods. 100 sample curves are shown for each method after 5000 labels are consumed. The thick red curve is the true precision-recall curve (assuming all labels are known).

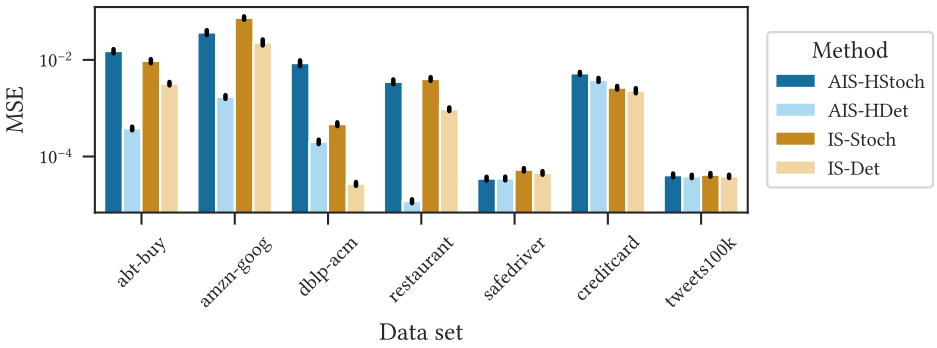


Figure 6.6: Comparison of the mean-squared error achieved by evaluation methods designed for stochastic (`Stoch`) and deterministic (`Det`) oracles (lower is better). Results are shown for estimating F1-score under a deterministic oracle with a label budget of 5000. 95% confidence intervals are shown in black.

Stochastic versus deterministic policies. We are interested to see whether there is a significant loss in label efficiency if a policy designed for a *stochastic oracle* is used to evaluate a system under a *deterministic oracle*. Recall that a deterministic oracle is a special case of a stochastic oracle, where the response $p(y|x)$ collapses to a point mass. We expect a policy designed for a stochastic oracle will not fare as well, as it does not account for the constraint on $p(y|x)$. Indeed, Figure 6.6 shows that AIS and IS methods designed for a stochastic oracle (`AIS-HStoch` and `IS-Stoch`) are less efficient than methods designed for a deterministic oracle (`AIS-HDet` and `IS-Det`) when the oracle is deterministic. The difference is significant for the severely imbalanced data sets—a 1–2 order of magnitude reduction in the MSE. This highlights the importance of tailoring the policy for a deterministic oracle.

Effect of stratification. Figure 6.7 demonstrates the effect of the stratification granularity K on the MSE when estimating F1-score. Three granularities are compared ($K = 16, 64, 256$) for two of the AIS variants (`AIS-HDet` and `AIS-IStoch`) assuming a fixed label budget of 5000. In most cases, there is only a small difference in MSE across the different values of K. The largest difference is approximately one order of magnitude for `dblp-acm` and `restaurant`. We generally observe that `AIS-IStoch` performs best for smaller values of K. This is to be expected for a limited label budget, since the number of model parameters increases linearly in K and there is no sharing of statistical strength across the strata for the independent model.

Effect of smoothing strength. Recall that we add a positive smoothing constant ξ when setting the Dirichlet concentration hyperparameters for the AIS variants. This ensures that the concentration parameters are positive and it reduces the influence of the classifier scores, which may be overly concentrated. Figure 6.8 demonstrates the effect of varying ξ for `AIS-IStoch` assuming a fixed label budget of 5000. The difference in the MSE is most pronounced for the imbalanced data sets. We find that stronger smoothing ($\xi = 1$) is beneficial when the classifier scores are unreliable, as is the case for `restaurant`. However, when the classifier scores are reliable, the efficiency can be

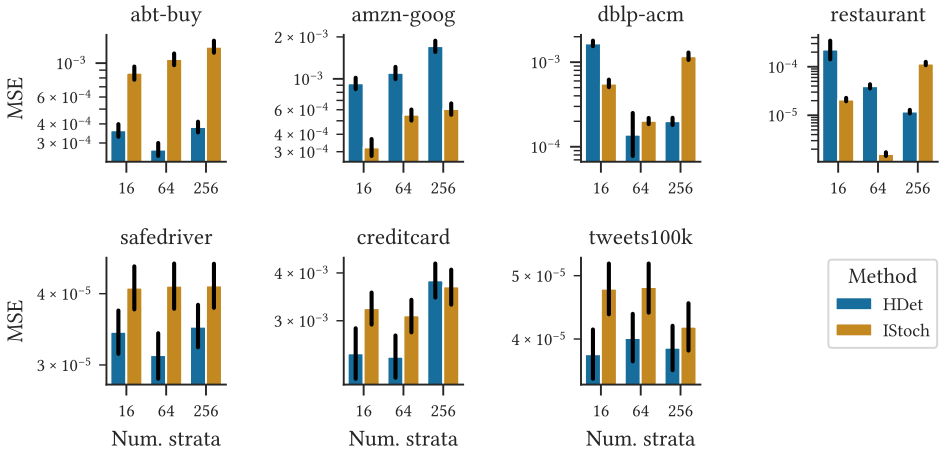


Figure 6.7: Assessing the effect of the number of strata (K) on the mean-squared error of the estimated F1-score (lower is better). Results are shown for two methods (`HDet` and `IStoch`) and several data sets assuming a label budget of 5000. `HDet` is configured with a branching factor of $b = 2$. 95% confidence intervals are shown in black.

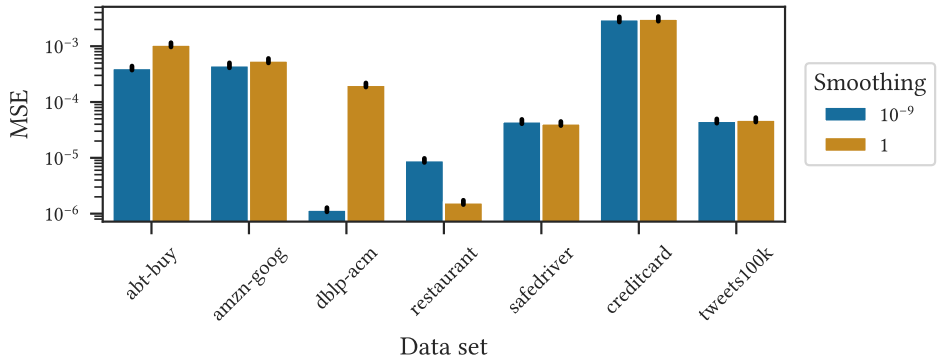


Figure 6.8: Assessing the effect of the smoothing constant on the mean-squared error of the estimated F1-score (lower is better). Results are shown for `AIS-IStoch` for a label budget of 5000. 95% confidence intervals are shown in black.

improved by reducing the degree of smoothing ($\xi = 10^{-9}$), which is especially obvious for `dblp-acm`.

6.7 Concluding remarks

In this chapter, we have demonstrated the effectiveness of our adaptive importance sampling framework for evaluating entity resolution (ER) systems under limited label budgets. Building on the framework and analysis presented in Chapter 5, we designed adaptive labelling policies based on the principle of variance minimisation. The policies select new items to label based on previously labelled items, in order to approximately minimise the asymptotic variance of the estimated performance measure. This can result in label budgets savings compared to a passive policy, as fewer labels are needed to estimate the performance to a given precision.

In designing adaptive labelling policies, we adopt a modular approach. Our approach allows the practitioner to specify an online model for the response $p(y|x)$ from the labelling oracle, and uses estimates of $p(y|x)$ from the model to approximate the asymptotically-optimal labelling policy. We presented several methods for approximating the asymptotically-optimal labelling policy, including variations tailored for deterministic and stochastic oracles. In addition, we proposed example models for the oracle response $p(y|x)$ which leverage stratification for efficiency. Our example models are Bayesian, and can therefore naturally leverage prior information from the systems under evaluation, while adapting to labels received from the oracle.

We conducted a thorough empirical study of our framework under three adaptive labelling policies, and compared to baselines from the literature, including importance sampling [SLS10; Sch+16] and stratified sampling [DM11]. The results support our contention that biased sampling and adaptivity, as encompassed in our framework, are effective techniques for improving label-efficiency. We observed substantial improvements over passive and stratified sampling, with reductions in mean-squared error by more than two orders of magnitude in some cases, for a fixed label budget. Our framework also significantly outperformed importance sampling on three of the four ER evaluation tasks and was competitive on the remaining tasks. Among the baseline methods tested, we found that importance sampling was the only method capable of achieving significant efficiency gains.

One limitation of our work, is that we were unable to compare the efficiency of adaptive labelling policies in the non-asymptotic regime. This appears to be a challenging problem—to date, there are few non-asymptotic results for adaptive importance sampling in the literature (a recent exception is [AM19]). From an empirical perspective, we did not observe consistent differences/trends in efficiency among the three adaptive labelling policies tested. However, our results were conclusive on the importance of selecting a policy tailored for the type of oracle (deterministic versus stochastic).

In our empirical study, we simulated human annotators using publicly-available data with ground truth labels. Future work could assess how our framework performs in deployment, as unforeseen issues may arise. For instance, practitioners and annotators may not be used to performing labelling adaptively in batches. Another interesting direction is the integration of our framework with truth-finding algorithms used in crowdsourcing [Zhe+17]. Unlike our framework, truth-finding algorithms generally assume labels are deterministic conditional on the features/input, and attempt to infer the labels based on responses from multiple noisy annotators. By integrating our framework with truth-finding, it may be possible to further improve efficiency, by controlling for the reliability of individual annotators.

Chapter 7

Conclusions and future directions

This thesis investigates statistical approaches for performing and evaluating entity resolution (ER)—a key task for data cleaning and data integration. While statistical approaches have been successfully applied to ER since the 1960s [FS69], there are several practical challenges which have yet to be fully resolved. The first is the need for vast quantities of human-labelled data, which may be used for model fitting, hyperparameter tuning and evaluation. Although the need for labelled data is not unique to ER, the quantities required are extraordinary due to severe class imbalance between coreferent and non-coreferent records. The second challenge is the inability of many ER methods to properly handle uncertainty—both prior uncertainty about the source data and posterior uncertainty about the ER results. Related to this is a third challenge—the fact that ER methods which *are* capable of handling uncertainty are often limited by the computational cost of inference. The fourth and final problem addressed in this thesis is a lack of statistical rigour when evaluating ER, which is largely due to the class imbalance noted above.

This thesis makes several contributions to the ER literature, focused on the challenges outlined above. Motivated by the need for label-efficient methods and proper quantification of uncertainty, Chapters 3 and 4 explore unsupervised Bayesian models for ER, with the aim of improving scalability, robustness and accuracy. Chapters 5 and 6 address challenges for evaluation of ER, by formulating evaluation as a statistical estimation problem that can be efficiently solved using adaptive importance sampling (AIS) methods. The proposed AIS framework reduces the impact of severe class imbalance, while improving statistical precision. We discuss these contributions in greater detail in Section 7.1, and list directions for future research in Section 7.2.

7.1 Summary of contributions

7.1.1 Bayesian models for entity resolution

Bayesian models provide a natural framework for solving entity resolution tasks under uncertainty. They are particularly useful when access to labelled data is limited, as one can rely on prior knowledge and assumptions encoded in the model. While Bayesian models have been applied effectively to small ER tasks [Ste15; SHF16], there has been limited work on scalable inference to date [MSM19]. In Chapter 3, we address this limitation for the `blink` ER model [Ste15], by contributing a scalable and distributed

extension called `d-blink`. One of our key insights in this work is that blocking [Ste+14] can be incorporated in the data generation process, by artificially partitioning the latent space in which the entities reside. After introducing the partition, we assume each record is assigned to one of the blocks according to a random process which preserves the marginal posterior distribution. In doing so, we obtain the benefits of blocking—reducing comparisons between unlikely matches—without compromising posterior correctness asymptotically. The introduction of blocks also enables distributed/parallel inference at the block-level.

Another contribution of this work is the investigation of partially-collapsed Gibbs (PCG) sampling [DP08] as an alternative to regular Gibbs sampling for approximate inference. Although PCG sampling is known to improve statistical efficiency, we find that gains in statistical efficiency may be counteracted by reductions in computational efficiency. We show that it is possible to vastly improve the overall efficiency of inference for a particular variant of PCG sampling, using fast algorithms for parameter updates. This includes a sub-quadratic algorithm for updating the entity assignments based on indexing, and a fast algorithm for updating the entity attributes based on perturbation sampling and the Vose-Alias algorithm [Vos91].

Chapter 3 closes with a thorough empirical evaluation, demonstrating the gains in efficiency that are achievable with `d-blink`. In addition to examining efficiency, we also assess the sensitivity and quality of fit of the ER model on moderate-to-large data sets, which was not previously possible due to limited scalability. We show that `d-blink` can be applied to perform realistic ER tasks, by conducting a case study on U.S. Census data and administrative data from the U.S. Social Security Administration. We also consider synthetic data from the Australian Bureau of Statistics, which is intended to mimic a realistic ER task for census and survey data.

Following our empirical observations in Chapter 3, we propose several modelling refinements in Chapter 4, aimed at reducing the sensitivity of `blink` and improving the goodness of fit. We identify `blink`'s informative prior on the linkage structure (coreference relation) as a potential issue, and argue that the family of Ewens-Pitman (EP) random partitions [Pit06] are a suitable alternative. In fact, we claim that the EP random partitions span the space of possible priors on the linkage structure under the assumption of exchangeability and Kolmogorov consistency. To further improve the flexibility of the EP random partitions, we place hyperpriors on the EP parameters for three distinct asymptotic regimes. To our knowledge, these priors have not been studied comprehensively in the context of ER.

Another key contribution of this work is a modified distortion model. We claim that the hit-miss model of Copas and Hilton [CH90] used in `blink`, is not suitably generalised for atomic distortion distributions. In particular, it allows for a record attribute to be in a distorted state while simultaneously being in perfect agreement with the corresponding entity attribute. We correct this logical inconsistency by explicitly excluding the entity attribute from the support of the distortion distribution. We also modify the distortion probability for a record attribute, by introducing a conditional dependence on the entity attribute value. Finally, we improve the flexibility of the entity and distortion models by introducing priors on parameters that were set empirically in `blink`.

We conclude Chapter 4 with a thorough empirical study, assessing the impact of the EP priors and corrected distortion model. The results indicate that both changes have a

beneficial impact on goodness of fit. We also draw a somewhat surprising conclusion: that there is little difference in performance between the three EP parameter regimes, despite the fact that they are known to exhibit distinct asymptotic behaviour. This finding contributes to existing discussion in the literature about appropriate priors for the linkage structure in ER models [BS14; Zan+16]. Lastly, we conduct experiments which demonstrate the effectiveness of our model in comparison with `blink` and a Bayesian ER model proposed by Sadinle [Sad14].

7.1.2 Label-efficient evaluation of entity resolution

When automated ER systems are applied to unseen data, there may be significant uncertainty about the quality of results. Evaluation can help to quell this uncertainty, by providing estimates of system performance with respect to a representative sample of labelled data. However, obtaining statistically accurate and precise estimates of system performance remains a challenge due to severe imbalance between coreferent and non-coreferent records [Kas+19]. As a result, large quantities of unbiased labelled samples are required to obtain precise performance estimates, which may be infeasible due to costs associated with human labelling. We believe this problem may cause some practitioners to avoid evaluation [MAS14; CBW17], or perform evaluation in a statistically unsound manner [Fu+12; Rah+14]. In Chapter 5, we address these issues by proposing a label-efficient online evaluation framework based on adaptive importance sampling (AIS). While the framework is motivated by ER applications, it is presented as a general framework for evaluating predictive systems—i.e. any system that makes a prediction based on a given input.

The framework improves upon existing online supervised evaluation methods in the literature [BC10; SLS10; DM11; Sch+16]. In particular, it supports a much broader family of performance measures which can be expressed as transformations of vector-valued risk functionals. This enables evaluation of multiple systems and/or performance measures in parallel. Moreover, the framework is adaptive—i.e. the labelling policy can be refined based on incoming labels to optimise statistical efficiency. This is in contrast to the majority of existing online approaches, which are non-adaptive [SLS10; DM11; Sch+16].

Another contribution of this work is asymptotic analysis, which provides statistical guarantees under verifiable conditions. In particular, we show that the estimates produced by our framework are strongly consistent, which means that they converge to the population performance asymptotically. We also obtain a central limit theorem (CLT), which is useful for assessing asymptotic efficiency and computing approximate confidence regions. We find that there exists a finite lower bound on the asymptotic variance for some classes of performance measures, which is an uncommon occurrence in importance sampling applications. The asymptotic variance is also used to derive the asymptotically-optimal labelling policy, which minimises the asymptotic variance.

In Chapter 6, we build on the theoretical results of Chapter 5 to instantiate the evaluation framework. We propose several estimators for the asymptotically-optimal labelling policy, which depend on estimates of the response from the oracle (the entity that responds to label queries). While there are no restrictions on the method used to estimate the oracle response, we suggest two Bayesian models as examples. Both Bayesian models rely on stratification to reduce complexity, with one model assuming

independent strata, and the other assuming hierarchically dependent strata. We provide comprehensive empirical studies, which compare our AIS framework to static importance sampling, stratified sampling and passive (uniform) sampling across a range of data sets and performance measures. The results are in line with expectations—showing that biased sampling is highly effective for evaluation of ER, where class imbalance is severe. In an imbalanced setting, our AIS-based framework is most beneficial when prior information from the system(s) under evaluation is unreliable. To the best of our knowledge, our empirical study is the first to assess online evaluation methods for ER applications.

7.2 Future research directions

While this thesis makes several contributions to ER methodology, there are a number of directions for future research, which we summarise below.

7.2.1 Scaling Bayesian entity resolution to billions of records

In Chapter 3, we proposed methods for improving the scalability of a Bayesian ER model called `blink`. While our methods resulted in a significant improvement, allowing us to scale to approximately 1 million records, we did not experiment with larger ER tasks commonly encountered in industry and government. For example, ER tasks performed by national statistical agencies often involve very large administrative data sources and national census data [Ras+12]. In the U.S., each individual source may contain hundreds of millions of records, assuming good coverage of the population. It is unclear whether our methods can cope with data at this scale—i.e. approaching one billion records.

At very large scales, variational methods may be a promising alternative to Markov chain Monte Carlo (MCMC) for approximate inference. These methods cast inference as an optimisation problem—that of finding a variational approximation that diverges minimally from the true posterior distribution [BKM17]. However, finding a suitable variational approximation can be challenging, particularly for complex models. Future work could explore the viability of variational methods for ER models, building on prior work for Dirichlet Process mixture models [BJ06] and scalable stochastic variational inference [Hof+13]. A further advantage of variational methods, is their ability to handle streaming data, which is likely to be challenging for MCMC [WPB11].

7.2.2 Modelling improvements for Bayesian entity resolution

In Chapter 4, we focused on modelling improvements for Bayesian ER. However, there are several interesting directions that warrant further exploration.

Modelling the distortion process. The Bayesian ER models we considered in Chapters 3 and 4, are tailored for relatively clean structured data with atomic attributes. While they can be applied to dirtier semi-structured data with non-atomic attributes, we observed higher error rates empirically. The cause of the error can likely be attributed to the distortion model, which is not designed to capture semantic heterogeneity. Future work could explore more sophisticated distortion models, which move beyond string similarity/distance measures. For example, generative models for edit-based corruptions could be

used in place of standard edit distance/similarity measures [RY98; BM03]. Techniques from the natural language processing community could also be employed to better capture the semantic meaning of attribute values [Lu+08].

Constraints on the linkage structure. In Chapters 3 and 4, we considered ER of multiple data sources assuming an entity may appear multiple times in each source. While this covers the most general case, in practice it may be reasonable to assume that a data source is deduplicated—i.e. that each entity appears only once (or not at all) in the source. High quality data sources often satisfy this constraint, at least approximately, as the data custodians are incentivised to minimise duplication [GRC11]. Future work could explore the incorporation of these constraints in Bayesian generative models. There is some existing work in this direction [SHF16], however the "no-duplicate" constraints are applied in a post-hoc manner, and do not arise from assumptions encoded in a generative model. Such constraints also break exchangeability of the observed records, and would therefore break the urn-based inference scheme used in Chapter 4.

Microclustering priors. A recent interesting direction in Bayesian modelling for ER, is the idea of *microclustering priors* [Mil+15] (see Section 2.4.3 for a survey). These prior are well-motivated for ER, as they assume *sublinear* growth in the number of records linked to each entity, whereas standard clustering priors assume *linear* growth. Although several microclustering priors have been proposed [Zan+16; KJ16; BCT17], they have not been thoroughly tested in the context of ER. Future work could explore whether these priors yield an improvement over the infinitely exchangeable priors considered in Chapter 4. It would also be interesting to combine microclustering priors with constraints on the linkage structure, as mentioned above.

7.2.3 End-to-end propagation of uncertainty

A central focus of this thesis has been on propagation and quantification of uncertainty throughout the ER process. The Bayesian ER method we developed in Chapter 3 allows for propagation of uncertainty between the blocking, comparison, and clustering steps, which are traditionally performed sequentially (see Figure 2.2). In addition, it theoretically allows for propagation of uncertainty between ER and data fusion (see Figure 2.1), as the latent entity attributes in the model can be regarded as the "fused" values. Although, propagation of uncertainty between ER and data fusion is theoretically possible, we did not experiment with the idea in Chapter 3. Future work could explore this idea, although there may be issues to overcome with label-switching—i.e. the fact that the entities do not have consistent identifiers across iterations of the Markov chain [JHS05].

A more ambitious avenue for future work could explore end-to-end propagation of uncertainty throughout the data integration process (see Figure 2.1). This could be achieved by incorporating schema alignment in the Bayesian model introduced in Chapter 3. Theoretically, this would allow schema alignment to be informed by later steps in the data integration process—e.g. the schema mapping could be revised based on associations between records that are thought to represent the same entity.

7.2.4 Non-asymptotic theory for adaptive importance sampling

In Chapters 5 and 6 we proposed a framework for evaluation based on adaptive importance sampling (AIS). Our primary objective was to improve the precision of performance estimates for a fixed label budget, thereby giving practitioners greater confidence in evaluation results. While we were able to analyse the asymptotic precision (efficiency) asymptotically, non-asymptotic results would be more useful in practice, and could inform the design of adaptive labelling policies (see Chapter 6). Currently, there are limited theoretical results in the literature for AIS in the non-asymptotic regime. Future work could obtain non-asymptotic bounds on the mean-squared error of our approach, which may be feasible when the input space is assumed to be finite. Recent work by Akyildiz and Míguez [AM19] may provide useful proof techniques.

7.2.5 Label-efficient evaluation and crowdsourcing

In our formulation of the evaluation problem, we assumed that labels are provided by a stochastic oracle which represents the conditional distribution of the unknown label $p(y|x)$. While this is reasonable from a formal perspective, in practice the responses from the oracle may deviate from the unknown true data generation process encoded in $p(y|x)$. Deviations are especially likely when the oracle is implemented via a crowdsourcing platform. This is because crowdsourcing workers are generally not domain experts, and error rates are known to vary among workers [BPB16]. In the crowdsourcing literature, this issue is managed by asking multiple workers to complete the same task, so that errors can be minimised using consensus or truth-finding algorithms [Zhe+17]. It would be interesting to explore whether these ideas can be integrated in our evaluation framework to better account for unreliable crowdsourcing workers.

Bibliography

[ACN08] N. Ailon, M. Charikar, and A. Newman. "Aggregating Inconsistent Information: Ranking and Clustering". In: *J. ACM* 55.5 (2008). DOI: 10.1145/1411509.1411513.

[AG09] A. Acquisti and R. Gross. "Predicting Social Security numbers from public data". In: *Proceedings of the National Academy of Sciences* 106.27 (2009), pp. 10975–10980. DOI: 10.1073/pnas.0904891106.

[AGK10] A. Arasu, M. Götz, and R. Kaushik. "On Active Learning of Record Matching Packages". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. SIGMOD'10. Indianapolis, Indiana, USA: Association for Computing Machinery, 2010, pp. 783–794. DOI: 10.1145/1807167.1807252.

[Ald85] D. J. Aldous. "Exchangeability and related topics". In: *École d'Été de Probabilités de Saint-Flour XIII — 1983*. Ed. by P. L. Hennequin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 1–198.

[AM19] Ö. D. Akyildiz and J. Míguez. *Convergence rates for optimised adaptive importance samplers*. 2019. arXiv: 1903.12044 [stat.CO].

[Amb+02] J. L. Ambite et al. "Data Integration and Access". In: *Advances in Digital Government: Technology, Human Factors, and Policy*. Ed. by W. J. McIver and A. K. Elmagarmid. Boston, MA: Springer US, 2002, pp. 85–106. DOI: 10.1007/0-306-47374-7_5.

[AR14] C. C. Aggarwal and C. K. Reddy, eds. 1st. Chapman & Hall/CRC, 2014, p. 652. DOI: 10.1201/9781315373515.

[ARS08] O. Alonso, D. E. Rose, and B. Stewart. "Crowdsourcing for Relevance Evaluation". In: *SIGIR Forum* 42.2 (2008), pp. 9–15. DOI: 10.1145/1480506.1480508.

[ASW14] S. Ahn, B. Shahbaba, and M. Welling. "Distributed Stochastic Gradient MCMC". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, pp. II–1044–II–1052.

[Ban] Banca d'Italia. *Bank of Italy – Survey on Household Income and Wealth*. URL: http://www.bancaditalia.it/pubblicazioni/indagine-famiglie/index.html (visited on 03/09/2018).

[Bar15] M. Barnes. *A Practioner's Guide to Evaluating Entity Resolution Results*. 2015. arXiv: 1509.04238 [cs.DB].

[BBC04] N. Bansal, A. Blum, and S. Chawla. "Correlation Clustering". In: *Machine Learning* 56.1 (2004), pp. 89–113. DOI: 10.1023/B:MACH.0000033116.57574.95.

[BBS05] M. Bilenko, S. Basu, and M. Sahami. "Adaptive Product Normalization: Using Online Learning for Record Linkage in Comparison Shopping". In: *Proceedings of the Fifth IEEE International Conference on Data Mining*. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 58–65. DOI: 10.1109/ICDM.2005.18.

[BC10] P. N. Bennett and V. R. Carvalho. "Online Stratified Sampling: Evaluating Classifiers at Web-scale". In: *CIKM*. 2010, pp. 1581–1584. DOI: 10.1145/1871437.1871677.

[BCT17] G. D. Benedetto, F. Caron, and Y. W. Teh. "Non-exchangeable random partition models for microclustering". In: (2017). arXiv: 1711.07287 [stat.ME].

[BDL09] A. Beygelzimer, S. Dasgupta, and J. Langford. "Importance Weighted Active Learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML'09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 49–56. DOI: 10.1145/1553374.1553381.

[Bec11] H. Becker. "Identification and Characterization of Events in Social Media". PhD thesis. Columbia University, 2011.

[Bel+12] K. Bellare et al. "Active Sampling for Entity Matching". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: ACM, 2012, pp. 1131–1139. DOI: 10.1145/2339530.2339707.

[Ben+09] O. Benjelloun et al. "Swoosh: a generic approach to entity resolution". In: *The VLDB Journal* 18.1 (2009), pp. 255–276. DOI: 10.1007/s00778-008-0098-x.

[Ben75] J. L. Bentley. "Multidimensional Binary Search Trees Used for Associative Searching". In: *Commun. ACM* 18.9 (1975), pp. 509–517. DOI: 10.1145/361002.361007.

[BG06] I. Bhattacharya and L. Getoor. "A Latent Dirichlet Model for Unsupervised Entity Resolution". In: *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM. 2006, pp. 47–58. DOI: 10.1137/1.9781611972764.5.

[BG07] I. Bhattacharya and L. Getoor. "Collective entity resolution in relational data". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), p. 5.

[BH08] P. A. Bernstein and L. M. Haas. "Information Integration in the Enterprise". In: *Commun. ACM* 51.9 (2008), pp. 72–79. DOI: 10.1145/1378727.1378745.

[BH10] W. Buntine and M. Hutter. "A Bayesian View of the Poisson-Dirichlet Process". In: (2010). arXiv: 1007.0296 [math.ST].

[Bil] M. Bilenko. *Duplicate Detection, Record Linkage, and Identity Uncertainty: Datasets*. URL: http://www.cs.utexas.edu/users/ml/riddle/data.html (visited on 12/05/2016).

[BIR00] J. O. Berger, D. R. Insua, and F. Ruggeri. "Bayesian Robustness". In: *Robust Bayesian Analysis*. Ed. by D. R. Insua and F. Ruggeri. New York, NY: Springer New York, 2000, pp. 1–32. DOI: 10.1007/978-1-4612-1306-2_1.

[BJ06] D. M. Blei and M. I. Jordan. "Variational inference for Dirichlet process mixtures". In: *Bayesian Anal.* 1.1 (2006), pp. 121–143. DOI: 10.1214/06-BA104.

[BKM06] M. Bilenko, B. Kamath, and R. J. Mooney. "Adaptive Blocking: Learning to Scale Up Record Linkage". In: *Proceedings of the Sixth International Conference on Data Mining*. ICDM'06. USA: IEEE Computer Society, 2006, pp. 87–96. DOI: 10.1109/ICDM.2006.13.

[BKM17] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (2017). DOI: 10.1080/01621459.2017.1285773.

[BM02] M. Bilenko and R. J. Mooney. *Learning to Combine Trained Distance Metrics for Duplicate Detection in Databases*. Tech. rep. Department of Computer Sciences, University of Texas at Austin, 2002.

[BM03] M. Bilenko and R. J. Mooney. "Adaptive Duplicate Detection Using Learnable String Similarity Measures". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. New York, NY, USA: ACM, 2003, pp. 39–48. DOI: 10.1145/956750.956759.

[BMR11] P. A. Bernstein, J. Madhavan, and E. Rahm. "Generic Schema Matching, Ten Years Later". In: *Proc. VLDB Endow.* 4.11 (2011), pp. 695–701. DOI: 10.14778/3402707.3402710.

[BN09] J. Bleiholder and F. Naumann. "Data Fusion". In: *ACM Comput. Surv.* 41.1 (2009), 1:1–1:41. DOI: 10.1145/1456650.1456651.

[BPB16] A. Burmania, S. Parthasarathy, and C. Busso. "Increasing the Reliability of Crowdsourcing Evaluations Using Online Quality Assessment". In: *IEEE Transactions on Affective Computing* 7.4 (2016), pp. 374–388. DOI: 10.1109/TAFFC.2015.2493525.

[BR95] T. R. Belin and D. B. Rubin. "A Method for Calibrating False-Match Rates in Record Linkage". In: *Journal of the American Statistical Association* 90.430 (1995), pp. 694–707. DOI: 10.2307/2291082.

[BS14] T. Broderick and R. C. Steorts. *Variational Bayes for Merging Noisy Databases*. 2014. arXiv: 1410.4792 [stat.ME].

[Bug+17] M. F. Bugallo et al. "Adaptive Importance Sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 60–79. DOI: 10.1109/MSP.2017.2699226.

[Cal09] C. Callison-Burch. "Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. EMNLP'09. Singapore: Association for Computational Linguistics, 2009, pp. 286–295.

[Cap+08] O. Cappé et al. “Adaptive importance sampling in general mixture classes”. In: *Statistics and Computing* 18.4 (2008), pp. 447–459. DOI: 10.1007/s11222-008-9059-x.

[CBW17] J. Chipperfield, J. J. Brown, and N. Watson. “The Australian Census Longitudinal Dataset: using record linkage to create a longitudinal sample from a series of cross-sections”. In: *Australian & New Zealand Journal of Statistics* 59.1 (2017), pp. 1–16. DOI: 10.1111/anzs.12177.

[CCH04] P. Christen, T. Churches, and M. Hegland. “Febrl – A Parallel Open Source Data Linkage System”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by H. Dai, R. Srikant, and C. Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 638–647.

[CF13] J. Chang and J. W. Fisher III. “Parallel Sampling of DP Mixture Models Using Sub-clusters Splits”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Vol. 1. NIPS’13. NY, USA: Curran Associates Inc., 2013, pp. 620–628.

[CG07] P. Christen and K. Goiser. “Quality and Complexity Measures for Data Linkage and Deduplication”. In: *Quality Measures in Data Mining*. Ed. by F. J. Guillet and H. J. Hamilton. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 127–151. DOI: 10.1007/978-3-540-44918-8_6.

[CG16] T. Chen and C. Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[CG95] S. Chib and E. Greenberg. “Understanding the Metropolis-Hastings Algorithm”. In: *The American Statistician* 49.4 (1995), pp. 327–335. DOI: 10.1080/00031305.1995.10476177.

[CGM05] S. Chaudhuri, V. Ganti, and R. Motwani. “Robust Identification of Fuzzy Duplicates”. In: *Proceedings of the 21st International Conference on Data Engineering*. ICDE ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 865–876. DOI: 10.1109/ICDE.2005.125.

[CH90] J. B. Copas and F. J. Hilton. “Record Linkage: Statistical Models for Matching Computer Records”. In: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 153.3 (1990), pp. 287–320. DOI: 10.2307/2982975.

[Cha+09] J. Chang et al. “Reading Tea Leaves: How Humans Interpret Topic Models”. In: *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. NIPS’09. Vancouver, British Columbia, Canada: Curran Associates Inc., 2009, pp. 288–296.

[CHK09] M. J. Cafarella, A. Halevy, and N. Khoussainova. “Data Integration for the Relational Web”. In: *Proc. VLDB Endow.* 2.1 (2009), pp. 1090–1101. DOI: 10.14778/1687627.1687750.

[Chr12a] P. Christen. “A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication”. In: *IEEE Transactions on Knowledge and Data Engineering* 24.9 (2012), pp. 1537–1555. DOI: 10.1109/TKDE.2011.127.

[Chr12b] P. Christen. "A survey of indexing techniques for scalable record linkage and deduplication". In: *IEEE Transactions on Knowledge and Data Engineering* 24.9 (2012), pp. 1537–1555.

[Chr12c] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Berlin Heidelberg: Springer-Verlag, 2012.

[Chr14] P. Christen. *Preparation of a real temporal voter data set for record linkage and duplicate detection research*. Tech. rep. Australian National University, 2014.

[Chu+16] X. Chu et al. "Data Cleaning: Overview and Emerging Challenges". In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD'16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 2201–2206. DOI: 10.1145/2882903.2912574.

[CKM00] W. W. Cohen, H. Kautz, and D. McAllester. "Hardening Soft Information Sources". In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD'00. Boston, Massachusetts, USA: Association for Computing Machinery, 2000, pp. 255–259. DOI: 10.1145/347090.347141.

[CL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[CM05] A. Culotta and A. McCallum. "Joint Deduplication of Multiple Record Types in Relational Data". In: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. CIKM'05. Bremen, Germany: Association for Computing Machinery, 2005, pp. 257–258. DOI: 10.1145/1099554.1099615.

[CML14] M. Chen, S. Mao, and Y. Liu. "Big Data: A Survey". In: *Mobile Networks and Applications* 19 (2 2014), pp. 171–209. DOI: 10.1007/s11036-013-0489-0.

[Coc77] W. G. Cochran. *Sampling Techniques*. en. 3rd. New York: Wiley, 1977.

[Cor+12] J.-M. Cornuet et al. "Adaptive Multiple Importance Sampling". In: *Scandinavian Journal of Statistics* 39.4 (2012), pp. 798–812. DOI: 10.1111/j.1467-9469.2011.00756.x.

[CPC98] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. "Efficient Construction of Large Test Collections". In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'98. Melbourne, Australia: Association for Computing Machinery, 1998, pp. 282–289. DOI: 10.1145/290941.291009.

[CR02] W. W. Cohen and J. Richman. "Learning to Match and Cluster Large High-Dimensional Data Sets for Data Integration". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD'02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 475–480. DOI: 10.1145/775047.775116.

[CRF03] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. "A Comparison of String Distance Metrics for Name-Matching Tasks". In: *Proceedings of the 2003 International Conference on Information Integration on the Web*. IIWEB'03. Acapulco, Mexico: AAAI Press, 2003, pp. 73–78.

[Cro16] D. F. Crouse. "On implementing 2D rectangular assignment algorithms". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696. DOI: 10.1109/TAES.2016.140952.

[CRT17] C. Culnane, B. I. P. Rubinstein, and V. Teague. *Health Data in an Open World*. 2017. arXiv: 1712.05627 [cs.CY].

[Cul+07] A. Culotta et al. "Canonicalization of Database Records Using Adaptive Similarity Measures". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. San Jose, California, USA: ACM, 2007, pp. 201–209. DOI: 10.1145/1281192.1281217.

[Dal86] T. Dalenius. "Finding a needle in a haystack or identifying anonymous census records". In: *Journal of official statistics* 2.3 (1986), pp. 329–336.

[Das+12] A. Das Sarma et al. "An Automatic Blocking Mechanism for Large-Scale Deduplication Tasks". In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM'12. Maui, Hawaii, USA: Association for Computing Machinery, 2012, pp. 1055–1064. DOI: 10.1145/2396761.2398403.

[de +10] J. de Freitas et al. "Active Learning Genetic programming for record deduplication". In: *IEEE Congress on Evolutionary Computation*. 2010, pp. 1–8. DOI: 10.1109/CEC.2010.5586104.

[De +15] P. De Blasi et al. "Are Gibbs-Type Priors the Most Natural Generalization of the Dirichlet Process?" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.2 (2015), pp. 212–229. DOI: 10.1109/TPAMI.2013.217.

[DE96] T. J. DiCiccio and B. Efron. "Bootstrap confidence intervals". In: *Statist. Sci.* 11.3 (1996), pp. 189–228. DOI: 10.1214/ss/1032280214.

[Den96] S. Y. Dennis. "A Bayesian analysis of tree-structured statistical decision problems". In: *Journal of Statistical Planning and Inference* 53.3 (1996), pp. 323–344. DOI: 10.1016/0378-3758(95)00112-3.

[DH59] T. Dalenius and J. L. Hodges. "Minimum Variance Stratification". In: *Journal of the American Statistical Association* 54.285 (1959), pp. 88–101. DOI: 10.1080/01621459.1959.10501501.

[DHI12] A. Doan, A. Halevy, and Z. Ives. *Principles of Data Integration*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[DHM05] X. Dong, A. Halevy, and J. Madhavan. "Reference Reconciliation in Complex Information Spaces". In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD'05. Baltimore, Maryland: Association for Computing Machinery, 2005, pp. 85–96. DOI: 10.1145/1066157.1066168.

[DM05] H. Daumé III and D. Marcu. "A Bayesian Model for Supervised Clustering with the Dirichlet Process Prior". In: *J. Mach. Learn. Res.* 6 (2005), pp. 1551–1577.

[DM08] R. Douc and E. Moulines. "Limit theorems for weighted samples with applications to sequential Monte Carlo methods". In: *The Annals of Statistics* 36.5 (2008), pp. 2344–2376. DOI: 10.1214/07-AOS514.

[DM11] G. Druck and A. McCallum. "Toward Interactive Training and Evaluation". In: *CIKM*. New York, NY, USA, 2011, pp. 947–956. DOI: 10.1145/2063576.2063712.

[DN09] X. L. Dong and F. Naumann. "Data Fusion: Resolving Data Conflicts for Integration". In: *Proc. VLDB Endow.* 2.2 (2009), pp. 1654–1655. DOI: 10.14778/1687553.1687620.

[DP08] D. A. van Dyk and T. Park. "Partially Collapsed Gibbs Samplers". In: *Journal of the American Statistical Association* 103.482 (2008), pp. 790–796. DOI: 10.1198/016214508000000409.

[DP18] B. Delyon and F. Portier. "Asymptotic Optimality of Adaptive Importance Sampling". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 3138–3148.

[DS15] X. L. Dong and D. Srivastava. "Big Data Integration". In: *Synthesis Lectures on Data Management* 7.1 (2015), pp. 1–198.

[Dun46] H. L. Dunn. "Record Linkage". In: *American Journal of Public Health and the Nations Health* 36.12 (1946), pp. 1412–1416. DOI: 10.2105/AJPH.36.12.1412.

[Dur19] R. Durrett. *Probability: Theory and Examples*. 5th ed. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. DOI: 10.1017/9781108591034.

[DWW99] P. Damlen, J. Wakefield, and S. Walker. "Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.2 (1999), pp. 331–344. DOI: 10.1111/1467-9868.00179.

[Ebr+18] M. Ebraheem et al. "Distributed Representations of Tuples for Entity Resolution". In: *Proc. VLDB Endow.* 11.11 (2018), pp. 1454–1467. DOI: 10.14778/3236187.3236198.

[EC08] M. Elsner and E. Charniak. "You Talking to Me? A Corpus and Algorithm for Conversation Disentanglement". In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, 2008, pp. 834–842. URL: https://www.aclweb.org/anthology/P08-1095.

[EF11] D. Eddelbuettel and R. François. "Rcpp: Seamless R and C++ Integration". In: *Journal of Statistical Software* 40.8 (2011), pp. 1–18. DOI: 10.18637/jss.v040.i08.

[EFI17] T. Enamorado, B. Fifield, and K. Imai. *Using a probabilistic model to assist merging of large-scale administrative records*. Tech. rep. Department of Politics, 2017.

[EFI19] T. Enamorado, B. Fifield, and K. Imai. "Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records". In: *American Political Science Review* 113.2 (2019), pp. 353–371. DOI: 10.1017/S0003055418000783.

[Eft+17] V. Efthymiou et al. "Parallel Meta-Blocking for Scaling Entity Resolution over Big Heterogeneous Data". In: *Inf. Syst.* 65.C (2017), pp. 137–157. DOI: 10.1016/j.is.2016.12.001.

[EVE02] M. G. Elfeky, V. S. Verykios, and A. K. Elmagarmid. "TAILOR: a record linkage toolbox". In: *Proceedings 18th International Conference on Data Engineering*. San Jose, CA, USA: IEEE, 2002, pp. 17–28. DOI: 10.1109/ICDE.2002.994694.

[FBF77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. "An Algorithm for Finding Best Matches in Logarithmic Expected Time". In: *ACM Trans. Math. Softw.* 3.3 (1977), pp. 209–226. DOI: 10.1145/355744.355745.

[Fel68] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*. 3rd. Wiley Series in Probability and Statistics. Wiley, 1968.

[Fel71] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 2*. 2nd. Wiley Series in Probability and Statistics. Wiley, 1971.

[Fle+17] J. M. Flegal et al. *mcmcse: Monte Carlo Standard Errors for MCMC*. R package version 1.3-2. Riverside, CA, Denver, CO, Coventry, UK, and Minneapolis, MN, 2017.

[FLS15] J. P. Ferry, D. Lo, and T. Seaquist. "A Bayesian Idealization of Entity Resolution". In: *2015 18th International Conference on Information Fusion*. IEEE. 2015, pp. 150–157.

[For+01] M. Fortini et al. "On Bayesian Record Linkage". In: *Research in Official Statistics* 4.1 (2001), pp. 185–198. URL: https://op.europa.eu/en/publication-detail/-/publication/608fb041-c2ff-4457-9487-647a1e02b863.

[FS69] I. P. Fellegi and A. B. Sunter. "A Theory for Record Linkage". In: *Journal of the American Statistical Association* 64.328 (1969), pp. 1183–1210. DOI: 10.1080/01621459.1969.10501049.

[FSS16] D. Firmani, B. Saha, and D. Srivastava. "Online Entity Resolution Using an Oracle". In: *Proc. VLDB Endow.* 9.5 (2016), pp. 384–395. DOI: 10.14778/2876473.2876474.

[Fu+12] Z. Fu et al. "Multiple Instance Learning for Group Record Linkage". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by P.-N. Tan et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 171–182.

[GA17] A. J. Gates and Y.-Y. Ahn. "The Impact of Random Models on Clustering Similarity". In: *J. Mach. Learn. Res.* 18.1 (2017), pp. 3049–3076.

[Gal+18] S. Galhotra et al. "Robust Entity Resolution Using Random Graphs". In: *Proceedings of the 2018 International Conference on Management of Data*. SIGMOD'18. Houston, TX, USA: Association for Computing Machinery, 2018, pp. 3–18. DOI: 10.1145/3183713.3183755.

[Gao+19] J. Gao et al. "Efficient Knowledge Graph Accuracy Evaluation". In: *Proc. VLDB Endow.* 12.11 (2019), pp. 1679–1691. DOI: 10.14778/3342263.3342642.

[GAZ13] R. Gutman, C. C. Afendulis, and A. M. Zaslavsky. "A Bayesian Procedure for File Linking to Analyze End-of-Life Medical Costs". In: *Journal of the American Statistical Association* 108.501 (2013), pp. 34–47. DOI: 10.1080/01621459.2012.726889.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[Ge+15] H. Ge et al. "Distributed Inference for Dirichlet Process Mixture Models". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 2276–2284.

[GH04] P. Gunning and J. M. Horgan. "A New Algorithm for the Construction of Stratum Boundaries in Skewed Populations". In: *Survey Methodology* 30.2 (2004), pp. 159–166.

[GL04] S. Gomatam and M. D. Larsen. "Record Linkage and Counterterrorism". In: *CHANCE* 17.1 (2004), pp. 25–29. DOI: 10.1080/09332480.2004.10554883.

[GM12] L. Getoor and A. Machanavajjhala. "Entity Resolution: Theory, Practice & Open Challenges". In: *Proc. VLDB Endow.* 5.12 (2012), pp. 2018–2019. DOI: 10.14778/2367502.2367564.

[GP06] A. Gnedin and J. Pitman. "Exchangeable Gibbs partitions and Stirling triangles". In: *Journal of Mathematical Sciences* 138.3 (2006), pp. 5674–5685. DOI: 10.1007/s10958-006-0335-z.

[GRC11] J. Gemmell, B. I. P. Rubinstein, and A. K. Chandra. *Improving Entity Resolution with Global Constraints*. 2011. arXiv: 1108.6016 [cs.DB].

[GS08] C. Goble and R. Stevens. "State of the nation in data integration for bioinformatics". In: *Journal of Biomedical Informatics* 41.5 (2008). Semantic Mashup of Biomedical Data, pp. 687–693. DOI: 10.1016/j.jbi.2008.01.008.

[GSR96] M. Ganesh, J. Srivastava, and T. Richardson. "Mining Entity-Identification Rules for Database Integration". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 291–294.

[GW92] W. R. Gilks and P. Wild. "Adaptive Rejection Sampling for Gibbs Sampling". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41.2 (1992), pp. 337–348. DOI: https://doi.org/10.2307/2347565.

[HA85] L. Hubert and P. Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (1985), pp. 193–218. DOI: 10.1007/BF01908075.

[Har+14] K. Harron et al. “Evaluating bias due to data linkage error in electronic healthcare records”. In: *BMC Medical Research Methodology* 14.1 (2014), pp. 36–46. DOI: 10.1186/1471-2288-14-36.

[HEG06] J. Huang, S. Ertekin, and C. L. Giles. “Efficient Name Disambiguation for Large-Scale Databases”. In: *Knowledge Discovery in Databases: PKDD 2006*. Ed. by J. Fürnkranz, T. Scheffer, and M. Spiliopoulou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 536–544.

[HK07] A. Haghighi and D. Klein. “Unsupervised coreference resolution in a nonparametric Bayesian model”. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 2007, pp. 848–855.

[Hof+13] M. D. Hoffman et al. “Stochastic Variational Inference”. In: *J. Mach. Learn. Res.* 14.1 (2013), pp. 1303–1347.

[HRO06] A. Halevy, A. Rajaraman, and J. Ordille. “Data Integration: The Teenage Years”. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*. VLDB’06. Seoul, Korea: VLDB Endowment, 2006, pp. 9–16.

[HS95] M. A. Hernández and S. J. Stolfo. “The Merge/Purge Problem for Large Databases”. In: *SIGMOD Rec.* 24.2 (1995), pp. 127–138. DOI: 10.1145/568271.223807.

[HS98] M. A. Hernández and S. J. Stolfo. “Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem”. In: *Data Mining and Knowledge Discovery* 2.1 (1998), pp. 9–37. DOI: 10.1023/A:1009761603038.

[IB12] R. Isele and C. Bizer. “Learning Expressive Linkage Rules Using Genetic Programming”. In: *Proc. VLDB Endow.* 5.11 (2012), pp. 1638–1649. DOI: 10.14778/2350229.2350276.

[Jae05] M. Jaeger. “Ignorability in Statistical and Probabilistic Inference”. In: *J. Artif. Int. Res.* 24.1 (2005), pp. 889–917.

[Jar89] M. A. Jaro. “Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida”. In: *Journal of the American Statistical Association* 84.406 (1989), pp. 414–420. DOI: 10.1080/01621459.1989.10478785.

[JHS05] A. Jasra, C. C. Holmes, and D. A. Stephens. “Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling”. In: *Statistical Science* 20.1 (2005), pp. 50–67. DOI: 10.1214/088342305000000016.

[JN04] S. Jain and R. M. Neal. “A Split-Merge Markov chain Monte Carlo Procedure for the Dirichlet Process Mixture Model”. In: *Journal of Computational and Graphical Statistics* 13.1 (2004), pp. 158–182. DOI: 10.1198/1061860043001.

[JRB11] D. P. Jutte, L. L. Roos, and M. D. Brownell. “Administrative Record Linkage as a Tool for Public Health Research”. In: *Annual Review of Public Health* 32.1 (2011). PMID: 21219160, pp. 91–108. DOI: 10.1146/annurev-publhealth-031210-100700.

[Kan16] E. Kanoulas. "A Short Survey on Online and Offline Methods for Search Quality Evaluation". In: *Information Retrieval: 9th Russian Summer School, RuSSIR 2015, Saint Petersburg, Russia, August 24-28, 2015, Revised Selected Papers*. Ed. by P. Braslavski et al. Cham: Springer International Publishing, 2016, pp. 38–87. DOI: 10.1007/978-3-319-41718-9_3.

[Kas+19] J. Kasai et al. "Low-resource Deep Entity Resolution with Transfer and Active Learning". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 5851–5861. DOI: 10.18653/v1/P19-1586.

[KBS18] A. Kaplan, B. Betancourt, and R. C. Steorts. *Posterior Prototyping: Bridging the Gap between Bayesian Record Linkage and Regression*. 2018. arXiv: 1810.01538 [stat.ME].

[Kel84] R. P. Kelley. "Blocking Considerations for Record Linkage Under Conditions of Uncertainty". In: *Proceedings of the Social Statistics Section*. American Statistical Association, 1984, pp. 602–605.

[Kin78a] J. F. C. Kingman. "Random Partitions in Population Genetics". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 361.1704 (1978), pp. 1–20. URL: http://www.jstor.org/stable/79629.

[Kin78b] J. F. C. Kingman. "The Representation of Partition Structures". In: *Journal of the London Mathematical Society* s2-18.2 (1978), pp. 374–380. DOI: 10.1112/jlms/s2-18.2.374.

[Kit14] R. Kitchin. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. Sage, 2014.

[KJ16] A. Klami and A. Jitta. "Probabilistic Size-constrained Microclustering". In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'16. Jersey City, New Jersey, USA: AUAI Press, 2016, pp. 329–338.

[Kon+16] P. Konda et al. "Magellan: Toward building entity matching management systems". In: *Proceedings of the VLDB Endowment* 9.12 (2016), pp. 1197–1208.

[KTR10] H. Köpcke, A. Thor, and E. Rahm. "Evaluation of Entity Resolution Approaches on Real-world Match Problems". In: *PVLDB* 3.1 (2010), pp. 484–493. DOI: 10.14778/1920841.1920904.

[Lap00] P. A. Laplante. *Dictionary of Computer Science Engineering and Technology*. USA: CRC Press, Inc., 2000.

[Lar05] M. D. Larsen. "Advances in Record Linkage Theory: Hierarchical Bayesian Record Linkage Theory". In: *Proceedings of the Survey Research Methods Section*. American Statistical Association, 2005, pp. 3277–3284.

[Lar12] M. D. Larsen. *An experiment with hierarchical Bayesian record linkage*. 2012. arXiv: 1212.5203 [math.ST].

[Lea11] M. Lease. "On Quality Control and Machine Learning in Crowdsourcing". In: *Proceedings of the 11th AAAI Conference on Human Computation*. AAAIWS'11-11. AAAI Press, 2011, pp. 97–102.

[Liu04] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. New York: Springer-Verlag, 2004.

[LK17] D. Li and E. Kanoulas. "Active Sampling for Large-scale Information Retrieval Evaluation". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: ACM, 2017, pp. 49–58. DOI: 10.1145/3132847.3133015.

[Lov+13] D. Lovell et al. "ClusterCluster: Parallel Markov Chain Monte Carlo for Dirichlet Process Mixtures". In: (2013). arXiv: 1304.2302 [stat.ML].

[LR01] M. D. Larsen and D. B. Rubin. "Iterative Automated Record Linkage Using Mixture Models". In: *Journal of the American Statistical Association* 96.453 (2001), pp. 32–41. DOI: 10.1198/016214501750332956.

[LR02] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 2002.

[Lu+08] W. Lu et al. "A Generative Model for Parsing Natural Language to Meaning Representations". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP'08. Honolulu, Hawaii: Association for Computational Linguistics, 2008, pp. 783–792.

[Man10] K. G. Manton. *National Long-Term Care Survey: 1982, 1984, 1989, 1994, 1999 and 2004*. Ann Arbor, MI, 2010. DOI: 10.3886/ICPSR09681.v5.

[MAS14] P. Malhotra, P. Agarwal, and G. Shroff. "Graph-Parallel Entity Resolution using LSH & IMM". In: *Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference*. Ed. by K. S. Candan et al. Athens, Greece, 2014.

[ME96] A. E. Monge and C. P. Elkan. "The field matching problem: Algorithms and applications". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Ed. by E. Simoudis, J. Han, and U. Fayyad. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 267–270.

[Mil+15] J. W. Miller et al. "Microclustering: When the Cluster Sizes Grow Sublinearly with the Size of the Data Set". In: *NIPS. Bayesian Nonparametrics: The Next Generation Workshop*. 2015. URL: https://drive.google.com/file/d/0B1-Vx4o3v8xfZ0pDNW5hWDBNMTA/view.

[Min99] T. Minka. *The Dirichlet-tree Distribution*. Tech. rep. Justsystem Pittsburgh Research Center, 1999. URL: https://www.microsoft.com/en-us/research/publication/dirichlet-tree-distribution/.

[MK06] M. Michelson and C. A. Knoblock. "Learning Blocking Schemes for Record Linkage". In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*. AAAI'06. Boston, Massachusetts: AAAI Press, 2006, pp. 440–445.

[MM17] B. S. McVeigh and J. S. Murray. *Practical Bayesian Inference for Record Linkage*. 2017. arXiv: 1710.10558 [stat.ME].

[MNU00] A. McCallum, K. Nigam, and L. H. Ungar. "Efficient Clustering of High-dimensional Data Sets with Application to Reference Matching". In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '00. Boston, Massachusetts, USA: ACM, 2000, pp. 169–178. DOI: 10.1145/347090.347123.

[Moz+14] B. Mozafari et al. "Scaling up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning". In: *Proc. VLDB Endow.* 8.2 (2014), pp. 125–136. DOI: 10.14778/2735471.2735474.

[MPS19] J.-M. Marin, P. Pudlo, and M. Sedki. "Consistency of adaptive importance sampling and recycling schemes". In: *Bernoulli* 25.3 (2019), pp. 1977–1998. DOI: 10.3150/18-BEJ1042.

[MS17] A. Mazumdar and B. Saha. "A Theoretical Analysis of First Heuristics of Crowdsourced Entity Resolution". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 970–976.

[MSM19] B. S. McVeigh, B. T. Spahn, and J. S. Murray. *Scaling Bayesian Probabilistic Record Linkage with Post-Hoc Blocking: An Application to the California Great Registers*. 2019. arXiv: 1905.05337 [stat.ME].

[Mud+18] S. Mudgal et al. "Deep Learning for Entity Matching: A Design Space Exploration". In: *Proceedings of the 2018 International Conference on Management of Data*. SIGMOD '18. New York, NY, USA: ACM, 2018, pp. 19–34. DOI: 10.1145/3183713.3196926.

[MW04] A. McCallum and B. Wellner. "Conditional Models of Identity Uncertainty with Application to Noun Coreference". In: *Proceedings of the 17th International Conference on Neural Information Processing Systems*. NIPS'04. Vancouver, British Columbia, Canada: MIT Press, 2004, pp. 905–912.

[MWG10] D. Menestrina, S. E. Whang, and H. Garcia-Molina. "Evaluating Entity Resolution Results". In: *Proc. VLDB Endow.* 3.1–2 (2010), pp. 208–219. DOI: 10.14778/1920841.1920871.

[NBB11] D. Newman, E. V. Bonilla, and W. Buntine. "Improving Topic Coherence with Regularized Topic Models". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, pp. 496–504.

[NC02] V. Ng and C. Cardie. "Improving Machine Learning Approaches to Coreference Resolution". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 104–111. DOI: 10.3115/1073083.1073102.

[Nea00] R. M. Neal. "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2 (2000), pp. 249–265. DOI: 10.1080/10618600.2000.10474879.

[New+09] D. Newman et al. “Distributed algorithms for topic models”. In: *Journal of Machine Learning Research* 10.Aug (2009). Ed. by A. McCallum, pp. 1801–1828.

[New+59] H. B. Newcombe et al. “Automatic Linkage of Vital Records: Computers can be used to extract "follow-up" statistics of families from files of routine records”. In: *Science* 130.3381 (1959), pp. 954–959. DOI: 10.1126/science.130.3381.954.

[New88] H. B. Newcombe. *Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business*. USA: Oxford University Press, Inc., 1988.

[NH10] F. Naumann and M. Herschel. “An Introduction to Duplicate Detection”. In: *Synthesis Lectures on Data Management* 2.1 (2010), pp. 1–87. DOI: 10.2200/S00262ED1V01Y201003DTM003.

[Ni+20] Y. Ni et al. “Scalable Bayesian Nonparametric Clustering and Classification”. In: *Journal of Computational and Graphical Statistics* 29.1 (2020), pp. 53–65. DOI: 10.1080/10618600.2019.1624366.

[NRG12] S. N. Negahban, B. I. Rubinstein, and J. G. Gemmell. “Scaling Multiple-Source Entity Resolution Using Statistically Efficient Transfer Learning”. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM’12. Maui, Hawaii, USA: Association for Computing Machinery, 2012, pp. 2224–2228. DOI: 10.1145/2396761.2398606.

[NS08] A. Narayanan and V. Shmatikov. “Robust De-anonymization of Large Sparse Datasets”. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. Oakland, CA, USA: IEEE, 2008, pp. 111–125. DOI: 10.1109/SP.2008.33.

[OB92] M.-S. Oh and J. O. Berger. “Adaptive importance sampling in monte carlo integration”. In: *Journal of Statistical Computation and Simulation* 41.3-4 (1992), pp. 143–168. DOI: 10.1080/00949659208810398.

[OEC15] OECD. *Data-Driven Innovation*. Paris: OECD Publishing, 2015, p. 456. DOI: 10.1787/9789264229358-en.

[Pap+14] G. Papadakis et al. “Meta-Blocking: Taking Entity Resolution to the Next Level”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.8 (2014), pp. 1946–1960. DOI: 10.1109/TKDE.2013.54.

[Pap+16] G. Papadakis et al. “Comparative Analysis of Approximate Blocking Techniques for Entity Resolution”. In: *Proc. VLDB Endow.* 9.9 (2016), pp. 684–695. DOI: 10.14778/2947618.2947624.

[Pap+20] G. Papadakis et al. “Blocking and Filtering Techniques for Entity Resolution: A Survey”. In: *ACM Comput. Surv.* 53.2 (2020). DOI: 10.1145/3377455.

[Pas+02] H. Pasula et al. “Identity Uncertainty and Citation Matching”. In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS’02. Cambridge, MA, USA: MIT Press, 2002, pp. 1425–1432.

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pit06] J. Pitman. "Exchangeable random partitions". In: *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII – 2002*. Ed. by J. Picard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 37–53. DOI: 10.1007/3-540-34266-4_3.

[Pit95] J. Pitman. "Exchangeable and partially exchangeable random partitions". In: *Probability Theory and Related Fields* 102.2 (1995), pp. 145–158. DOI: 10.1007/BF01213386.

[Por17] Porto Seguro. *Porto Seguro's Safe Driver Prediction*. 2017. URL: https://www.kaggle.com/c/porto-seguro-safe-driver-prediction (visited on 12/05/2019).

[Poz+15] A. D. Pozzolo et al. "Calibrating Probability with Undersampling for Unbalanced Classification". In: *2015 IEEE Symposium Series on Computational Intelligence*. 2015, pp. 159–166. DOI: 10.1109/SSCI.2015.33.

[PR03] J. Pearl and S. Russell. "Bayesian Networks". In: *Handbook of Brain Theory and Neural Networks*. Ed. by M. A. Arbib. Cambridge, MA, USA: MIT Press, 2003, pp. 157–160.

[PY97] J. Pitman and M. Yor. "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator". In: *The Annals of Probability* 25.2 (1997), pp. 855–900. DOI: 10.1214/aop/1024404422.

[QPS17] K. Qian, L. Popa, and P. Sen. "Active Learning for Large-Scale Entity Resolution". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: ACM, 2017, pp. 1379–1388. DOI: 10.1145/3132847.3132949.

[Rah+14] H. Rahmani et al. "Contextual Entity Resolution Approach for Genealogical Data". In: *Proceedings of the 16th LWA Workshops: KDML, IR and FGWM*. Ed. by T. Seidl, M. Hassani, and C. Beecks. 2014.

[Ran71] W. M. Rand. "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336 (1971), pp. 846–850. DOI: 10.1080/01621459.1971.10482356.

[Ras+12] S. Rastogi et al. *2010 Census Match Study*. Tech. rep. Center for Administrative Records Research and Applications, United States Census Bureau, 2012. URL: https://www.census.gov/content/dam/Census/library/publications/2012/dec/2010_cpex_247.pdf.

[RC04] P. Ravikumar and W. W. Cohen. "A Hierarchical Graphical Model for Record Linkage". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI'04. Banff, Canada: AUAI Press, 2004, pp. 454–461.

[RH07] A. Rosenberg and J. Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 410–420. URL: https://www.aclweb.org/anthology/D07-1043.

[RK16] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Ltd, 2016. DOI: 10.1002/9781118631980.

[RY98] E. S. Ristad and P. N. Yianilos. "Learning string-edit distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.5 (1998), pp. 522–532. DOI: 10.1109/34.682181.

[Sad14] M. Sadinle. "Detecting duplicates in a homicide registry using a Bayesian partitioning approach". In: *Ann. Appl. Stat.* 8.4 (2014), pp. 2404–2434. DOI: 10.1214/14-AOAS779.

[Sad17] M. Sadinle. "Bayesian Estimation of Bipartite Matchings for Record Linkage". In: *Journal of the American Statistical Association* 112.518 (2017), pp. 600–612. DOI: 10.1080/01621459.2016.1148612.

[Sad18] M. Sadinle. "Bayesian propagation of record linkage uncertainty into population size estimation of human rights violations". In: *Ann. Appl. Stat.* 12.2 (2018), pp. 1013–1038. DOI: 10.1214/18-AOAS1178.

[Saw+10] C. Sawade et al. "Active Risk Estimation". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 951–958.

[SB02] S. Sarawagi and A. Bhamidipaty. "Interactive Deduplication Using Active Learning". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD'02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 269–278. DOI: 10.1145/775047.775087.

[SB10] M. Sariyar and A. Borg. "The RecordLinkage Package: Detecting Errors in Data". In: *The R Journal* 2.2 (2010), pp. 61–67.

[SBN17] R. C. Steorts, M. Barnes, and W. Neiswanger. "Performance Bounds for Graphical Record Linkage". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by A. Singh and J. Zhu. Vol. 54. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2017, pp. 298–306.

[SBP11] M. Sariyar, A. Borg, and K. Pommerening. "Controlling false match rates in record linkage using extreme value theory". In: *Journal of Biomedical Informatics* 44.4 (2011), pp. 648–654. DOI: 10.1016/j.jbi.2011.02.008.

[Sch+16] T. Schnabel et al. "Unbiased Comparative Evaluation of Ranking Functions". In: *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. ICTIR '16. Newark, Delaware, USA: ACM, 2016, pp. 109–118. DOI: 10.1145/2970398.2970410.

[SD06] P. Singla and P. Domingos. "Entity Resolution with Markov Logic". In: *Sixth International Conference on Data Mining (ICDM'06)*. 2006, pp. 572–582. DOI: 10.1109/ICDM.2006.65.

[Set09] B. Settles. *Active Learning Literature Survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2009. URL: http://digital.library.wisc.edu/1793/60660.

[SF13] M. Sadinle and S. E. Fienberg. "A Generalized Fellegi-Sunter Framework for Multiple Record Linkage With Application to Homicide Record Systems". In: *Journal of the American Statistical Association* 108.502 (2013), pp. 385–397. DOI: 10.1080/01621459.2012.757231.

[SG19] E. Saralioglu and O. Gungor. "Use of crowdsourcing in evaluating post-classification accuracy". In: *European Journal of Remote Sensing* 52.S1 (2019), pp. 137–147. DOI: 10.1080/22797254.2018.1564887.

[SHF16] R. C. Steorts, R. Hall, and S. E. Fienberg. "A Bayesian Approach to Graphical Record Linkage and Deduplication". In: *Journal of the American Statistical Association* 111.516 (2016), pp. 1660–1672. DOI: 10.1080/01621459.2015.1105807.

[SLS10] C. Sawade, N. Landwehr, and T. Scheffer. "Active Estimation of F-Measures". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 2083–2091. URL: http://papers.nips.cc/paper/3999-active-estimation-of-f-measures.pdf.

[Smi+81] J. M. Smith et al. "Multibase: Integrating Heterogeneous Distributed Database Systems". In: *Proceedings of the May 4-7, 1981, National Computer Conference*. AFIPS'81. Chicago, Illinois: Association for Computing Machinery, 1981, pp. 487–499. DOI: 10.1145/1500412.1500483.

[SN10] A. Smola and S. Narayanamurthy. "An Architecture for Parallel Topic Models". In: *Proc. VLDB Endow.* 3.1-2 (2010), pp. 703–710. DOI: 10.14778/1920841.1920931.

[SNL01] W. M. Soon, H. T. Ng, and D. C. Y. Lim. "A Machine Learning Approach to Coreference Resolution of Noun Phrases". In: *Computational linguistics* 27.4 (2001), pp. 521–544. DOI: 10.1162/089120101753342653.

[SS14] B. Saha and D. Srivastava. "Data quality: The other face of Big Data". In: *2014 IEEE 30th International Conference on Data Engineering*. 2014, pp. 1294–1297.

[Ste+14] R. C. Steorts et al. "A Comparison of Blocking Methods for Record Linkage". In: *Privacy in Statistical Databases*. Ed. by J. Domingo-Ferrer. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 253–268. DOI: 10.1007/978-3-319-11257-2_20.

[Ste15] R. C. Steorts. "Entity Resolution with Empirically Motivated Priors". In: *Bayesian Analysis* 10.4 (2015), pp. 849–875. DOI: 10.1214/15-BA965SI.

[STL18] R. C. Steorts, A. Tancredi, and B. Liseo. "Generalized Bayesian Record Linkage and Regression with Exact Error Propagation". In: *Privacy in Statistical Databases*. Ed. by J. Domingo-Ferrer and F. Montes. Cham: Springer International Publishing, 2018, pp. 297–313.

[Sun+14] C. Sun et al. “Chimera: Large-Scale Classification Using Machine Learning, Rules, and Crowdsourcing”. In: *Proc. VLDB Endow.* 7.13 (2014), pp. 1529–1540. DOI: 10.14778/2733004.2733024.

[SWD20] H. Song, Y. Wang, and D. B. Dunson. *Distributed Bayesian clustering using finite mixture of mixtures*. 2020. arXiv: 2003.13936 [stat.CO].

[SWY75] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing”. In: *Commun. ACM* 18.11 (1975), pp. 613–620. DOI: 10.1145/361219.361220.

[Teh06] Y. W. Teh. *A Bayesian interpretation of interpolated Kneser-Ney*. Tech. rep. National University of Singapore, 2006, p. 19. URL: http://dl.comp.nus.edu.sg/bitstream/handle/1900.100/1911/TRA2-06.pdf.

[TKM01] S. Tejada, C. A. Knoblock, and S. Minton. “Learning Object Identification Rules for Information Integration”. In: *Inf. Syst.* 26.8 (2001), pp. 607–633. DOI: 10.1016/S0306-4379(01)00042-4.

[TL11] A. Tancredi and B. Liseo. “A hierarchical Bayesian approach to record linkage and population size problems”. In: *The Annals of Applied Statistics* 5.2B (2011), pp. 1553–1585. DOI: 10.1214/10-AOAS447.

[TL15] A. Tancredi and B. Liseo. “Regression analysis with linked data: problems and possible solutions”. In: *Statistica* 75.1 (2015), pp. 19–35.

[TS10] L. Torrey and J. Shavlik. “Transfer Learning”. In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Ed. by E. S. Olivas et al. IGI Global, 2010, pp. 242–264. DOI: 10.4018/978-1-60566-766-9.ch011.

[TSL20] A. Tancredi, R. Steorts, and B. Liseo. “A Unified Framework for De-Duplication and Population Size Estimation”. In: *Bayesian Analysis* (2020). Advance publication. DOI: 10.1214/19-BA1146.

[TVP16] D. Turek, P. d. Valpine, and C. J. Paciorek. “Efficient Markov chain Monte Carlo sampling for hierarchical hidden Markov models”. In: *Environmental and Ecological Statistics* 23.4 (2016), pp. 549–564. DOI: 10.1007/s10651-016-0353-z.

[Uni] United States Census Bureau. *2010 Census Participation Rates*. URL: https://www.census.gov/data/datasets/2010/dec/2010-participation-rates.html (visited on 05/25/2020).

[Vaa98] A. W. v. d. Vaart. “Delta Method”. In: *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998, pp. 25–34. DOI: 10.1017/CBO9780511802256.004.

[Vau17] J. W. Vaughan. “Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research”. In: *J. Mach. Learn. Res.* 18.1 (2017), pp. 7026–7071.

[VBD14] N. Vesdapunt, K. Bellare, and N. Dalvi. “Crowdsourcing Algorithms for Entity Resolution”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1071–1082. DOI: 10.14778/2732977.2732982.

[VEB10] N. X. Vinh, J. Epps, and J. Bailey. "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance". In: *Journal of Machine Learning Research* 11.Oct (2010), pp. 2837–2854.

[Ver+00] V. S. Verykios et al. "On the Accuracy and Completeness of the Record Matching Process". In: *Proceedings of the 2000 Conference on Information Quality*. Ed. by B. D. Klein and D. F. Rossin. MIT, 2000, pp. 54–69.

[VFJ19] D. Vats, J. M. Flegal, and G. L. Jones. "Multivariate output analysis for Markov chain Monte Carlo". In: *Biometrika* 106.2 (2019), pp. 321–337. DOI: 10.1093/biomet/asz002.

[VG15] V. Verroios and H. Garcia-Molina. "Entity Resolution with crowd errors". In: *2015 IEEE 31st International Conference on Data Engineering*. 2015, pp. 219–230. DOI: 10.1109/ICDE.2015.7113286.

[VGP17] V. Verroios, H. Garcia-Molina, and Y. Papakonstantinou. "Waldo: An Adaptive Human Interface for Crowd Entity Resolution". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. SIGMOD'17. Chicago, Illinois, USA: Association for Computing Machinery, 2017, pp. 1133–1148. DOI: 10.1145/3035918.3035931.

[Vos91] M. D. Vose. "A linear algorithm for generating random numbers with a given distribution". In: *IEEE Transactions on Software Engineering* 17.9 (1991), pp. 972–975. DOI: 10.1109/32.92917.

[Wal+10] H. Wallach et al. "An Alternative Prior Process for Nonparametric Bayesian Clustering". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Teh and M. Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 892–899.

[Wan+12] J. Wang et al. "CrowdER: Crowdsourcing Entity Resolution". In: *Proc. VLDB Endow.* 5.11 (2012), pp. 1483–1494. DOI: 10.14778/2350229.2350263.

[Wan+13] J. Wang et al. "Leveraging Transitive Relations for Crowdsourced Joins". In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. SIGMOD'13. New York, New York, USA: Association for Computing Machinery, 2013, pp. 229–240. DOI: 10.1145/2463676.2465280.

[WDX13] S. Williamson, A. Dubey, and E. Xing. "Parallel Markov Chain Monte Carlo for Nonparametric Mixture Models". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, 2013, pp. 98–106.

[Wel+04] B. Wellner et al. "An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI'04. Banff, Canada: AUAI Press, 2004, pp. 593–601.

[Wha+09] S. E. Whang et al. “Entity Resolution with Iterative Blocking”. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. SIGMOD’09. Providence, Rhode Island, USA: Association for Computing Machinery, 2009, pp. 219–232. DOI: 10.1145/1559845.1559870.

[Wic+08] M. L. Wick et al. “A Unified Approach for Schema Matching, Coreference and Canonicalization”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’08. Las Vegas, Nevada, USA: ACM, 2008, pp. 722–730. DOI: 10.1145/1401890.1401977.

[Wil11] D. R. Wilson. “Beyond probabilistic record linkage: Using neural networks and complex features to improve genealogical record linkage”. In: *The 2011 International Joint Conference on Neural Networks*. 2011, pp. 9–14. DOI: 10.1109/IJCNN.2011.6033192.

[Win00] W. E. Winkler. *Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage*. Tech. rep. No. RR2000/05. U.S. Bureau of the Census, 2000. URL: https://www.census.gov/srd/papers/pdf/rr2000-05.pdf.

[Win02] W. E. Winkler. *Methods for Record Linkage and Bayesian Networks*. Tech. rep. Statistics #2002-05. U.S. Bureau of the Census, 2002.

[Win05] W. E. Winkler. *Approximate String Comparator Search Strategies for Very Large Administrative Lists*. Tech. rep. Statistics #2005-02. Statistical Research Division, U.S. Census Bureau, 2005.

[Win06] W. E. Winkler. *Overview of Record Linkageand Current Research Directions*. Tech. rep. Statistics #2006-2. Statistical Research Division, U.S. Census Bureau, 2006.

[Win14] W. E. Winkler. “Matching and record linkage”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6.5 (2014), pp. 313–325. DOI: 10.1002/wics.1317.

[Win89] W. E. Winkler. “Methods for adjusting for lack of independence in an application of the Fellegi-Sunter model of record linkage ”. In: *Survey Methodology* 15.1 (1989), pp. 101–117. URL: https://www150.statcan.gc.ca/n1/en/catalogue/12-001-X198900114574.

[Win90] W. E. Winkler. “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage”. In: *Proceedings of the Section on Survey Research Methods*. American Statistical Association, 1990, pp. 354–359.

[WMG13] S. E. Whang, D. Marmaros, and H. Garcia-Molina. “Pay-As-You-Go Entity Resolution”. In: *IEEE Trans. on Knowl. and Data Eng.* 25.5 (2013), pp. 1111–1124. DOI: 10.1109/TKDE.2012.43.

[WPB11] C. Wang, J. Paisley, and D. Blei. "Online Variational Inference for the Hierarchical Dirichlet Process". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Gordon, D. Dunson, and M. Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 752–760. URL: http://proceedings.mlr.press/v15/wang11a.html.

[WWP13] P. Welinder, M. Welling, and P. Perona. "A Lazy Man's Approach to Benchmarking: Semisupervised Classifier Evaluation and Recalibration". In: *CVPR*. 2013, pp. 3262–3269. DOI: 10.1109/CVPR.2013.419.

[Xu+13] W. Xu et al. "A case study on entity resolution for distant processing of big humanities data". In: *2013 IEEE International Conference on Big Data*. Silicon Valley, California, USA, 2013, pp. 113–120. DOI: 10.1109/BigData.2013.6691678.

[Yan+07] S. Yan et al. "Adaptive Sorted Neighborhood Methods for Efficient Record Linkage". In: *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL'07. Vancouver, BC, Canada: Association for Computing Machinery, 2007, pp. 185–194. DOI: 10.1145/1255175.1255213.

[YB07] L. Yujian and L. Bo. "A Normalized Levenshtein Distance Metric". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1091–1095. DOI: 10.1109/TPAMI.2007.1078.

[Zah+16] M. Zaharia et al. "Apache Spark: A Unified Engine for Big Data Processing". In: *Commun. ACM* 59.11 (2016), pp. 56–65. DOI: 10.1145/2934664.

[Zan+16] G. Zanella et al. "Flexible Models for Microclustering with Application to Entity Resolution". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. NY, USA: Curran Associates Inc., 2016, pp. 1425–1433.

[Zan20] G. Zanella. "Informed Proposals for Local MCMC in Discrete Spaces". In: *Journal of the American Statistical Association* 115.530 (2020), pp. 852–865. DOI: 10.1080/01621459.2019.1585255.

[Zha+12] B. Zhao et al. "A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration". In: *Proc. VLDB Endow.* 5.6 (2012), pp. 550–561. DOI: 10.14778/2168651.2168656.

[Zhe+17] Y. Zheng et al. "Truth Inference in Crowdsourcing: Is the Problem Solved?" In: *Proc. VLDB Endow.* 10.5 (2017), pp. 541–552. DOI: 10.14778/3055540.3055547.

[ZRG15] D. Zhang, B. I. P. Rubinstein, and J. Gemmell. "Principled Graph Matching Algorithms for Integrating Multiple Data Sources". In: *IEEE Transactions on Knowledge and Data Engineering* 27.10 (2015), pp. 2784–2796. DOI: 10.1109/TKDE.2015.2426714.

[Zua+19] D. A. Zuanetti et al. "Bayesian nonparametric clustering for large data sets". In: *Statistics and Computing* 29.2 (2019), pp. 203–215. DOI: 10.1007/s11222-018-9803-9.

Appendix A

Gibbs updates for `d-blink`

In this appendix, we list the posterior conditional distributions used to implement the Gibbs updates for `d-blink`. These are derived by referring to the posterior distribution in (3.4).

A.1 Update for the distortion probabilities

The distortion probability θ_{sa} for attribute a in source s is updated by sampling from the following conditional distribution:

$$\theta_{ta}|\mathbf{Z}, \mathbf{\Lambda}, \mathbf{\Gamma}, \mathbf{Y}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S} \sim \text{Beta}\left(\tilde{z}_{sa} + \beta_{sa}^{(0)}, N_s - \tilde{z}_{sa} + \beta_{sa}^{(1)}\right)$$

where $\tilde{z}_{sa} := \sum_{i:\, s_i = s} z_{ia}$ is the number of distorted record values for attribute a in source s and $N_s = |\{i : s_i = s\}|$ is the number of records from source s.

A.2 Update for the distortion indicators

The distortion indicator z_{ia} for attribute a of record i is updated by sampling from the following conditional distribution:

$$z_{ia}|\mathbf{\Lambda}, \mathbf{\Gamma}, \mathbf{Y}, \mathbf{\Theta}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S} \sim (1 - o_{ia})\,\text{Bernoulli}(\theta_{ta}) + o_{ia}\,\text{Bernoulli}(\zeta_a(\theta_{s_i a}, x_{ia}, y_{\lambda_i a}))$$

where

$$\zeta_a(\theta, x, y) = \begin{cases} 1, & \text{if } x \neq y, \\ \frac{\theta\psi_a(x|y)}{\theta\psi_a(x|y) - \theta + 1}, & \text{otherwise.} \end{cases}$$

A.3 Update for the linkage structure

The linked entity λ_i for record i is updated by sampling from the following conditional distribution:

$$p(\lambda_i|\mathbf{\Gamma}, \mathbf{Y}, \mathbf{\Theta}, \mathbf{Z}, \mathbf{X}^{(o)}, \mathbf{O}, \mathbf{S}) \propto \mathbb{I}\left[\lambda_i \in \mathscr{E}_{\gamma_i}(\mathbf{Y})\right] \prod_{\substack{a \\ o_{tra}=1}} \left\{(1 - z_{ia})\mathbb{I}[x_{ia} = y_{\lambda_i a}] + z_{ia}\psi_a(x_{ia}|y_{\lambda_i a})\right\}.$$

Appendix B

Gibbs updates for the refined ER model

In this appendix, we derive updates for the partially-collapsed Gibbs sampler used to perform approximate inference for the ER model introduced in Chapter 4. Some of the updates are non-trivial due to non-conjugacy of the model.

B.1 Update for the distortion probabilities

The update for the distortion probability θ_{sa} for source s and attribute a is complicated by the presence of the distortion propensity variables ω_{ia}, which break the conjugacy of the beta prior. To deal with this, we introduce auxiliary variables

$$q_{ia}|\omega_{ia} \sim \text{Bernoulli}(\omega_{ia}) \qquad \forall i, a$$

and modify the conditional distribution for the distortion indicators as follows

$$z_{ia}|\theta_{s_i a}, q_{ia} \sim \text{Bernoulli}(\theta_{s_i a} q_{ia}) \qquad \forall i, a.$$

It is straightforward to show that one recovers the original model (4.4) when the auxiliary variables are marginalised out.

After introducing the auxiliary variables, we alternate between updating $\mathbf{Q} = \{q_{ia}\}$ and $\boldsymbol{\Theta} = \{\theta_{sa}\}$ while holding all other variables fixed. The updates for the other model parameters are unaffected by the introduction of the auxiliary variables. In particular, the update for z_{ia} is deterministic conditional on $y_{\lambda_i a}$ and x_{ia}.

We include the updates below, leaving the derivation as an exercise for the reader:

$$q_{ia}|\omega_{ia}, \theta_{s_i a}, z_{ia} \sim \text{Bernoulli}\left(\frac{\omega_{ia}\theta_{s_i a}^{z_{ia}}(1-\theta_{s_i a})^{1-z_{ia}}}{\omega_{ia}\theta_{s_i a}^{z_{ia}}(1-\theta_{s_i a})^{1-z_{ia}} + 1 - \omega_{ia}}\right) \qquad \forall i, a,$$

$$\theta_{sa}|\mathbf{Q}, \mathbf{Z}, \mathbf{S} \sim \text{Beta}\left(\beta_{sa}^{(0)} + \sum_{i\,:\,s_i = s} z_{ia},\ \beta_{sa}^{(1)} + \sum_{i\,:\,s_i = s} q_{ia}(1 - z_{ia})\right) \qquad \forall s, a.$$

B.2 Update for the entity attributes

When updating the entity attribute y_{ea}, we collapse the base distribution H_{ea} and distortion indicators $\mathbf{Z}$.

Using the result from Section 4.5.3 we have

$$\begin{aligned} P(y_{ea}|\mathbf{Z},\boldsymbol{\Omega},\boldsymbol{\Theta},\mathbf{S},G_a) &\propto P(y_{ea}|G_a)\prod_{i:\lambda_i=e} P(x_{ia}|\theta_{s_ia},\omega_{ia},y_{ea}) \\ &\propto G_a(y_{ea})\frac{n_{ea}^{\neg}(y_{ea})\mathrm{B}(\rho_a;n_{ea}^{\neg}(y_{ea}))}{\prod_{v\in\mathcal{D}_{ea}\setminus\{y_{ea}\}} n_{ea}(v)\mathrm{B}(\rho_a\psi_a(v|y_{ea});n_{ea}(v))} \\ &\quad\times\prod_{\substack{i:\lambda_i=e\\x_{ia}=y_{ea}}}(1-\theta_{s_ia}\omega_{ia})\prod_{\substack{i:\lambda_i=e\\x_{ia}\neq y_{ea}}}(\theta_{s_ia}\omega_{ia}), \end{aligned}$$

where $\mathcal{D}_{ea}=\bigcup_{i:\lambda_i=e}\{x_{ia}\}$, $n_{ea}(v)=\sum_{i:\lambda_i=e}\mathbb{I}[x_{ia}=v]$, $n_{ea}^{\neg}(v)=\sum_{i:\lambda_i=e}\mathbb{I}[x_{ia}\neq v]$ and B is the beta function.

We can then expand the beta functions to yield a more useful expression for implementing the update:

$$\begin{aligned} P(y_{ea}|\mathbf{Z},\boldsymbol{\Omega},\boldsymbol{\Theta},\mathbf{S},G_a) &\propto G_a(y_{ea})\frac{n_{ea}^{\neg}(y_{ea})!\Gamma(\rho_a)}{\Gamma(n_{ea}^{\neg}(y_{ea})+\rho_a)}\prod_{v\in\mathcal{D}_{ea}\setminus\{y_{ea}\}}\frac{\Gamma(n_{ea}(v)+\rho_a\psi_a(v|y_{ea}))}{n_{ea}(v)!\,\Gamma(\rho_a\psi_a(v|y_{ea}))} \\ &\quad\times\prod_{\substack{i:\lambda_i=e\\x_{ia}=y_{ea}}}(1-\theta_{s_ia}\omega_{ia})\prod_{\substack{i:\lambda_i=e\\x_{ia}\neq y_{ea}}}(\theta_{s_ia}\omega_{ia}) \\ &\propto G_a(y_{ea})\frac{\prod_{v\in\mathcal{D}_{ea}\setminus\{y_{ea}\}}\prod_{i=1}^{n_{ea}(v)}\left\{\frac{i-1}{i}+\rho_a\psi_a(v|y_{ea})\right\}}{\prod_{i=1}^{n_{ea}^{\neg}(y_{ea})}\left\{\frac{i-1}{i}+\rho_a\right\}} \\ &\quad\times\prod_{\substack{i:\lambda_i=e\\x_{ia}=y_{ea}}}(1-\theta_{s_ia}\omega_{ia})\prod_{\substack{i:\lambda_i=e\\x_{ia}\neq y_{ea}}}(\theta_{s_ia}\omega_{ia}). \end{aligned}$$

The last two lines follow from expanding the gamma functions and cancelling factors in the numerator/denominator. We note that the above distribution may only have support on a subset of the full domain $\mathcal{D}_a$ when distance thresholds are applied, as discussed in Section 4.5.4. In particular, one can show that the support is a subset of

$$\bigcap_{i:\lambda_i=e}\{y\in\mathcal{D}_a\,:\,\mathrm{dist}_a(y,x_{ia})\leq d_a^{(\mathrm{cut})}\}.$$

B.3 Update for the linkage structure

When updating the linkage structure, we use an urn-based scheme as described by Neal [Nea00]. In doing so, we only need to keep track of entities in the population that are linked to records—any isolated entities not linked to records are ignored. This is important, as the population may be infinite in size for some Ewens-Pitman parameter regimes (when $\sigma\geq 0$).

To update the linked entity λ_i for record i, we remove the current link and allow the record to join one of the remaining instantiated entities (with at least one other record) or spawn a "new" entity. The conditional distribution has the following form:

$$P(\lambda_i=e|\mathbf{Z},\mathbf{X},\mathbf{Y},\boldsymbol{\Lambda}_{-i})\propto\begin{cases} C\frac{|e|-\sigma}{\alpha+N-1}F(i,\mathbf{X},\mathbf{Z},\mathbf{y}_e,\boldsymbol{\Lambda}_{-i}), \\ \qquad\text{if } e \text{ is instantiated and } |e|>0, \\ C\frac{\alpha+\sigma E}{\alpha+N-1}\sum_{\mathbf{y}\in\bigotimes_a\mathcal{D}_a}F(i,\mathbf{X},\mathbf{Z},\mathbf{y},\boldsymbol{\Lambda}_{-i}), \\ \qquad\text{if } e \text{ is "new"}, \end{cases}$$

where C is a normalisation constant; $\mathbf{\Lambda}_{-i} = (\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_N)$ are the linked entities for all records excluding i; $|e| = \sum_{i' \neq i} \mathbb{I}[\lambda_{i'} = e]$ denotes the number of records (excluding i) linked to entity e; $E = \sum_{e' \neq e} \mathbb{I}[|e| > 0]$ is the number of instantiated entities with at least one linked record; and the likelihood factor is given by

$$F(i, \mathbf{X}, \mathbf{Z}, \mathbf{y}_e, \mathbf{\Lambda}_{-i}) = \prod_a \int \left\{ P(H_{ea}|y_{ea}) P(x_{ia}|z_{ia}, y_{ea}, H_{ea}) \prod_{i' \neq i : \lambda'_i = e} P(x_{i'a}|z_{i'a}, y_{ea}, H_{ea}) \right\} dH_{ea}. \tag{B.1}$$

Note that we are conditioning on the distortion indicators $\mathbf{Z} = \{z_{ia}\}$ for computational reasons, however we are collapsing the distortion distributions $\mathbf{H} = \{H_{ea}\}$. After substituting the conditional distributions in (B.1), the likelihood factor becomes

$$\begin{aligned} F(i, \mathbf{X}, \mathbf{Z}, \mathbf{y}_e, \mathbf{\Lambda}_{-i}) &= \prod_a \int \left\{ \frac{1}{\mathrm{B}(\psi_a(y_{ea}))} \prod_{v \in \mathbb{D}_a \setminus \{y_{ea}\}} (H_{ea}(v))^{\psi(v|y_{ea})-1} \right. \\ &\qquad \left. \prod_{i' : \lambda_{i'} = e} \{(1 - z_{i'a}) \mathbb{I}[x_{i'a} = y_{ea}] + z_{i'a} H_{ea}(x_{i'a})\} \right\} dH_{ea} \\ &= \prod_a \left\{ \prod_{i' : \lambda_{i'} = e} (1 - z_{i'a}) \mathbb{I}[x_{i'a} = y_{ea}] \right. \\ &\qquad \left. \frac{1}{\mathrm{B}(\psi_a(y_{ea}))} \prod_{v \in \mathbb{D}_a \setminus \{y_{ea}\}} \int (H_{ea}(v))^{m_{ea}(v) + \psi(v|y_{ea}) - 1} \, dH_{ea} \right\} \end{aligned}$$

where $m_{ea}(v) = \sum_{i' : \lambda'_i = e} z_{i'a} \mathbb{I}[x_{i'a} = v]$.

We then integrate out H_{ea} using the known result for a Dirichlet-Multinomial likelihood to yield

$$\begin{aligned} F(\mathbf{X}, \mathbf{Z}, \mathbf{y}_e, \mathbf{\Lambda}_{-i}) &\propto \prod_a \left\{ \prod_{i' : \lambda_{i'} = e} (1 - z_{i'a}) \mathbb{I}[x_{i'a} = y_{ea}] \right. \\ &\qquad \left. \times \frac{m_{ea}! \, \Gamma(\rho_a)}{\Gamma(m_{ea} + \rho_a)} \prod_{v \in \mathbb{D}_a \setminus \{y_{ea}\}} \frac{\Gamma(m_{ea}(v) + \rho_a \psi(v|y_{ea}))}{m_{ea}(v)! \, \Gamma(\rho_a \psi(v|y_{ea}))} \right\} \\ &= \prod_a \left\{ \prod_{i' : \lambda_{i'} = e} (1 - z_{i'a}) \mathbb{I}[x_{i'a} = y_{ea}] \right. \\ &\qquad \left. \times \frac{\prod_{v \in \mathbb{D}_a \setminus \{y_{ea}\}} \prod_{j=1}^{m_{ea}(v)} \left(\frac{j-1}{j} + \rho_a \psi(v|y_{ea}) \right)}{\prod_{j=1}^{m_{ea}} \left(\frac{j-1}{j} + \rho_a \right)} \right\} \end{aligned}$$

where $m_{ea} = \sum_{i' : \lambda'_i = e} z_{i'a}$.

B.4 Update for the Ewens-Pitman parameters

Since the priors on the Ewens-Pitman parameters α and σ are non-conjugate, we cannot perform a direct Gibbs update. Here we describe tractable updates which require the

introduction of auxiliary variables. The updates (and priors) differ depending on the range of σ. Teh [Teh06] proposed a scheme for beta/gamma priors when $0 \leq \sigma < 1$ and $\alpha > 0$, which is summarised in in Section B.4.1. In Section B.4.2 we propose a similar scheme for gamma/shifted negative binomial priors when $\sigma < 0$.

B.4.1 Case $0 \leq \sigma < 1$ and $\alpha > 0$

Teh [Teh06] proposed an auxiliary variable scheme for the regime $0 \leq \sigma < 1$ and $\alpha > 0$ such that the priors

$$\begin{aligned} \sigma &\sim \text{Beta}(\zeta^{(0)}, \zeta^{(1)}), \\ \alpha &\sim \text{Gamma}(\chi^{(0)}, \chi^{(1)}) \end{aligned}$$

are conjugate. We provide a summary of the scheme here, but refer the reader to [Teh06] for further details. The scheme introduces the following sets of auxiliary variables conditional on the two parameters α and σ:

$$\begin{aligned} w|N, \alpha &\sim \text{Beta}(\alpha + 1, N - 1), \\ u_k|\sigma, \alpha, E &\sim \text{Bernoulli}\left(\frac{\alpha}{\alpha + \sigma k}\right), \quad k \in \{1, \dots, E-1\} \\ v_{ej}|\sigma, \mathbf{\Lambda} &\sim \text{Bernoulli}\left(\frac{j-1}{j-\sigma}\right), \quad \forall e, j \in \{1, \dots, N_e\}. \end{aligned} \tag{B.2}$$

Here $N_e = |\{i : \lambda_i = e\}|$ denotes the number of records linked to entity e and $E = \sum_e \mathbb{I}[N_e > 1]$ denotes the number of entities linked to at least one record.

It follows that the posterior distributions of α and σ conditional on the auxiliary variables and other model parameter are given by:

$$\begin{aligned} \sigma|\{u_k\}, \{v_{ej}\}, \mathbf{\Lambda} &\sim \text{Beta}\left(\zeta^{(0)} + \sum_{k=1}^{E-1}(1-u_k), \zeta^{(1)} + \sum_{e:N_e>1}\sum_{j=1}^{N_e-1}(1-v_{ej})\right), \\ \alpha|\{u_k\}, w, \mathbf{\Lambda} &\sim \text{Gamma}\left(\chi^{(0)} + \sum_{k=1}^{E-1} u_k, \chi^{(1)} - \log w\right). \end{aligned} \tag{B.3}$$

Thus to update α and σ, one would first draw auxiliary variables w, $\{u_k\}$ and $\{v_{ej}\}$ conditional on the linkage structure $\mathbf{\Lambda}$ and the old values of α and σ using (B.2). Then, conditional on the auxiliary variables and the linkage structure, one would draw new values for α and σ using (B.3).

B.4.2 Case $\sigma < 0$ and $\alpha = m\kappa$ for $m \in \mathbb{N}$

We perform a change of variables to $\kappa = -\sigma$ and $m \in \mathbb{N}$ such that $\alpha = m\kappa$. The likelihood factor associated with the partition of N records into E entities is as follows [Pit06]:

$$P(\text{clust config}) = \frac{(m)_{E\downarrow}}{(m\kappa)_{N\uparrow}} \prod_{e=1}^{E} (\kappa)_{N_e\uparrow} = \frac{\kappa^{E-1}(m-1)_{E-1\downarrow}}{(m\kappa-1)_{N-1\uparrow}} \prod_{e=1}^{E} (\kappa-1)_{N_e-1\uparrow}, \tag{B.4}$$

where N_e is the number of records linked to the e-th entity, $(x)_{n\uparrow} = \prod_{i=0}^{n-1}(x+i)$ is the rising factorial, and $(x)_{n\downarrow} = \prod_{i=0}^{n-1}(x-i)$ is the falling factorial. We begin by expressing the denominator in this equation as

$$\begin{aligned}\frac{1}{(m\kappa-1)_{N-1\uparrow}} = \frac{\Gamma(m\kappa+1)}{\Gamma(m\kappa+N)} &= \frac{\mathrm{B}(m\kappa+1, N-1)}{\Gamma(N-1)} \\ &= \frac{1}{\Gamma(N-1)}\int_0^1 w^{m\kappa}(1-w)^{N-2}\,\mathrm{d}w,\end{aligned}$$

which allows us to introduce the following auxiliary variable:

$$w|m,\kappa,N \sim \mathrm{Beta}(m\kappa+1, N-1). \tag{B.5}$$

Expressing the latter factors in (B.4) as

$$(\kappa-1)_{N_e-1\uparrow} = \prod_{j=1}^{N_e-1}(\kappa+j) = \prod_{j=1}^{N_e-1}\sum_{v_{ej}\in\{0,1\}}\kappa^{v_{ej}}j^{1-v_{ej}}$$

permits us to introduce the following additional auxiliary variables:

$$v_{ej}|\kappa \sim \mathrm{Bernoulli}\left(\frac{\kappa}{\kappa+j}\right), \qquad \forall e, j \in \{1,\dots,N_e-1\}. \tag{B.6}$$

With this representation, we can place conjugate priors on κ and m, namely:

$$\kappa \sim \mathrm{Gamma}(\chi^{(0)},\chi^{(1)}) \text{ and } m \sim \mathrm{NegativeBinomial}(r,\nu)+1. \tag{B.7}$$

The distribution on m is a shifted negative binomial with support on the positive integers. The parameterisation we adopt for the negative binomial is in terms of the number of failures $x \in \{0,1,2,\dots\}$ in a sequence of trials before a given number of successes $r > 0$ occur. Each trial is an i.i.d. draw from a Bernoulli distribution with success probability ν. The density of x is given by

$$P(x|r,\nu,a) = \frac{(x+r-1)!}{(r-1)!x!}\nu^r(1-\nu)^x.$$

Finally, we combine the priors in (B.7) with the likelihood factors to obtain the following posterior distributions for the m and κ, conditional on the other model parameters:

$$\begin{aligned} m|w,\kappa,\mathbf{\Lambda} &\sim \mathrm{NegBinomial}(r+E-1, 1-(1-p)w^\kappa)+E, \\ \kappa|\{v_{ej}\},w,m,\mathbf{\Lambda} &\sim \mathrm{Gamma}\left(\chi^{(0)}+E-1+\sum_{e=1}^{E}\sum_{j=1}^{N_e}v_{ej},\ \chi^{(1)}-m\log w\right).\end{aligned} \tag{B.8}$$

Thus to update κ and m, one would first draw auxiliary variables w and $\{v_{ej}\}$ conditional on the linkage structure $\mathbf{\Lambda}$ and the old values of α and σ using (B.5) and (B.6). Then, conditional on the auxiliary variables and the linkage structure, one would draw new values for κ and m using (B.8).