Learning with Clusters

Matt Barnes

CMU-RI-TR-19-02

January, 2019

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Artur Dubrawski, *chair*
Geoff Gordon
Kris Kitani
Beka Steorts, *Duke University*

To my family.

Abstract

Clustering, the problem of grouping similar data, has been extensively studied since at least the 1950's. As machine learning becomes more prominent, clustering has evolved from primarily a data analysis tool into an integrated component of complex robotic and machine learning systems, including those involving dimensionality reduction, anomaly detection, network analysis, image segmentation and classifying groups of data.

With this integration into multi-stage systems comes a need to better understand interactions between pipeline components. Changing parameters of the clustering algorithm will impact downstream components and, quite unfortunately, it is usually not possible to simply backpropagate through the entire system. Instead, it is common practice to take the output of the clustering algorithm as ground truth at the next module of the pipeline. We show this false assumption causes subtle and dangerous behavior for even the simplest systems – empirically biasing results by upwards of 25%.

We address this gap by developing scalable estimators and methods to both quantify and compensate the impact of clustering errors on downstream learners. Our work is agnostic to the choice of other components of the machine learning systems, and requires few assumptions on the clustering algorithm. Theoretical and empirical results demonstrate our methods and estimators are superior to the current naive approaches, which do not account for clustering errors.

We also develop several new clustering algorithms and prove theoretical bounds for existing algorithms, to be used as inputs to our error-correction methods. Not surprisingly, we find that learning on clusters of data is both theoretically and empirically easier as the number of clustering errors decreases. Thus, our work is two-fold. We attempt to provide the best clustering possible as well as establish how to effectively learn on inevitably noisy clusters.

Acknowledgments

First, I want to thank Artur, my advisor, for having the patience to take on a mechanical engineering student who was curious about machine learning. I'm forever grateful for your trust and freedom you gave me to explore at will.

To my parents. Without you, this thesis would quite literally not have been possible. Thank you for your unending support.

To my brother Brian, thanks for vetting all my crazy ideas first. Glad we converged.

Thanks to my collaborators, especially Beka Steorts, for charting new research territory together and meeting us in all corners of the world.

I owe tremendous gratitude to my mentors, especially some early figures who had incredible patience with me and witnessed some of my most spectacular (and most formative!) failures. Looking at you, Sean Brennan and Liz Kisenwether.

And last but not least, a big shout out to all my friends both inside and outside CMU who made Pittsburgh such a special place in my heart. Zhe Zhang, Galen Privett, Grant Cole, Benedikt Boecking, Maria De Arteaga, Emma Diehl, Caroline Delson, Jeff Pontell, Dave Kolvek, Eva Gjekmarkaj, Willie Neiswanger, Micol Marchetti-Bowick, Tim Hyde, Ian Rosenberg, Shervin Javdani, Wen Sun, and all the Pharaoh Hounds. I'd do it again anytime.

Funding

This work was supported through an NSF GRFP fellowship, along with additional funding from DARPA (FA8750-14-2-0244, FA8750-17-2-0130 and FA8750-12-2-0324) and AFRL (FA8750-17-2-0212).

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Broadly speaking, clustering is the problem of grouping similar data, for some notion of similarity. Unlike supervised classification or regression problems, where the objective is well defined (e.g. minimizing mean-squared-error or classification error), clustering objectives are often less clear. The vast number of clustering objective functions and algorithms is testament to the number of goals one may have when clustering. Usually, clustering is framed as an unsupervised learning problem, though we also explore cases where semi-supervised information is available.

Clustering plays a role in many applications and machine learning systems, including but not limited to:

Anomaly detection: Clusters of similar data are "normal", whereas outliers are dissimilar to other data and do not clearly belong to any cluster. Removing anomalies is useful in reducing classification or regression error.

Dimensionality reduction: For example, a simple approach would be to cluster high-dimensional data into k clusters, and compute new k-dimensional features based on the distance to each cluster center. The lower-dimensional featurization could improve downstream classification or regression performance, in terms of error or computational speed.

Network analysis: Social networks exhibit well-known community structures. Clustering is often used to discover these groups, which can then be used to better target advertisements, trace the spread of information and infer population statistics.

Medical: Patient records across multiple hospitals and databases are matched and merged together to some a single, holistic view of each patient's medical history for better predicting patient diagnostics and care.

Online shopping: eBay, Google Shopping, and other online shopping websites wish to match all unique products, so that comparing prices of the same product from different sellers is straightforward. Here, we usually wish to know who is the cheapest seller of a particular product.

Counter-human-trafficking: Our original motivation for studying clustering originated from the counter-human-trafficking domain. Here, we have billions of advertisements for escorts scraped from websites such as Craigslist or Backpage.com and wish to cluster ads according to the person they describe. Much like the medical domain, forming a holistic view of each escort's behavior (e.g. posting frequency, travel pattern, acquaintances) to improve prostitution estimates and ultimately better classify potential human trafficking victims (defined as those who are underage or being coerced into such activities).

Image segmentation: Grouping image pixels or LIDAR points according to the object it belongs to is a critical step in self-driving car perception and other computer vision systems, where this is usually a precursor to then localizing, tracking and classifying each object.

With the exception of exploratory data analysis, the output of clustering is rarely useful in its own right, and is instead usually used as a component of a larger machine learning or data mining system. In each of the applications above, we have identified exactly how the clustering output could be used.

1.1 Thesis Problem

The key problem in using clustering as a component of machine learning systems, and the primary focus of this thesis, is that clustering outputs are essentially never perfect and these errors can have unintended consequences for the larger system.

Consider the medical domain problem of first (a) clustering patient records across hospitals and medical databases such that each cluster corresponds to a set of records of an individual patient and then (b) predicting whether each patient (i.e. a single cluster of records) has cancer. For now, let's naïvely assume we are able to infer a perfect clustering in part (a). In part (b), the proper approach to cross-validating a lung cancer classifier is to train on one set of clusters (e.g. patients) and then validate of a different, disjoint set of clusters. Thus, the error measured on the validation set is an unbiased estimate of the classifier error on new, previously unseen patients, or in other words the *out-of-cluster loss.* Training and validating on different sets of patients prevents the learner from overfitting to patient-specific features such as social security number, name and date-of-birth, which are not useful for prediction on new patients. Better predictors generalize across patients, e.g. unexplained weight loss, fatigue, and tumor image features.

The problem here is we are essentially never able to infer a perfect clustering in part (a). Instead, we are only able to find an approximation through some clustering or record linkage algorithm. Until now, the approach has been to *proceed as if the clustering was perfect, even if it is not.* There are subtle, and potentially significant, consequences of this false assumption.

Mistakes in the clustering algorithm are equivalent to samples flipping between the train and test sets. Instead of training and testing of a disjoint set of patients, we are now training and validating on records from some of the same patients – a major faux pas in machine learning. Our observed validation set error will now be optimistically biased by the learner overfitting to the patient-specific features previously mentioned, or worse yet, to less blatant overfitting such as an image classifier learning the shapes of each patient's bone structure to predict whether they have lung cancer. Clearly, this is not useful for new patients. We term this phenomenon *dependency leakage* and show that even at small clustering errors, it can cause significant bias in cross-validation results. This thesis addresses these challenging issues, summarized in the following problem:

Thesis Problem: Clustering algorithms are inevitably imperfect, and existing machine learning systems are unable to account for these errors.

Outside of the medical domain, we are familiar with similar problems in the census and counter-human-trafficking communities. At the US Census Bureau, matching persons across censuses is a challenging, imperfect process and the impact of using a noisy clustering for demographic, socioeconomic, and other statistical analysis is unclear. Similarly, imperfect clustering results are used to estimate death counts in Syria and to both estimate and predict human trafficking in the United States. A major concern in these domains is that dependency leakage can bias a learner against certain sub-populations (i.e. clusters). Later, we empirically demonstrate how dependency leakage causes bias against certain demographics in US Census data. This is increasingly relevant as data science plays a greater role in credit and policy decisions.

1.2 Summary of Thesis Approach

To that end, this thesis will address the issue of clustering errors in machine learning systems by (Part I) bounding and improving clustering performance on complex datasets involving categorical, string and numerical data and (Part II) learning on imperfect clusterings. Part I is a classically studied problem in machine learning, and many of our results extend previous work to new settings or provide better theoretical bounds. Part II, on the other hand, is a previously unaddressed problem within the domain of learning on noisy data, and most of the work here is relatively novel, including many of the mathematical tools.

1.2.1 Technical Formulation

Part I Given a set of n_x samples $X = x_1, \ldots, x_{n_x}$, the goal of clustering is to find a cluster partition function $\hat{c} : \{1, \ldots, n_x\} \to \{1, \ldots, k\}$ which maps each sample to one of k clusters. For example, each sample x_i may correspond to a hospital visit record and each of the k clusters corresponds to a patient. Samples are not necessarily limited to the numerical domain, they may also include categorical (e.g. blood type) and string data (e.g. name, city). As previously mentioned, the exact mathematical objective which measures the quality of $\hat{c}$ is somewhat subjective – it depends on our model and measure of similarity. However, we do assume that a true partition function c does exist, though it is likely unknown. In this thesis, we only consider *hard* partition functions, i.e. each sample belongs to exactly one cluster.

We consider several approaches, including correlation clustering, stochastic block models and Bayesian models. We leave the formulations of each these methods for their respective chapters, as they are too complex to fully describe here, but emphasize that in each case they provide a clustering approximation $\hat{c}$.

Part II In the second part, we wish to utilize the clustering approximation $\hat{c}$ from Part I to perform some useful task, such as predicting whether the hospital patient has heart disease. To illustrate, consider samples generated according to the simple k-mixture model

$$\begin{aligned} \phi_j &\overset{iid}{\sim} H(\gamma) && \text{for } j = 1, \ldots, k \\ c_i &\overset{iid}{\sim} \text{Categorical}(\pi) && \text{for } i = 1, \ldots, n_x \\ x_i, y_i &\overset{iid}{\sim} G(\phi_{c_i}) && \end{aligned} \tag{1.1}$$

where ϕ are latent cluster parameters; c are (potentially latent) cluster assignments; $X = x_1, \ldots, x_{n_x}$ are n_x samples; y are the corresponding labels; H is some distribution over cluster parameters; γ, π, k, n_x are model parameters and π is in the k-dimensional probability simplex. This includes, for example, many mixture models and topic models. Note that without conditioning on the latents ϕ, samples within the same cluster are dependent while samples in different clusters are independent.

Specifically, our goal in this setting is to find a learner $f : \mathcal{X} \to \mathcal{Y}$ which performs well on new clusters, i.e. has small out-of-cluster loss $\mathbb{E}_{x',y'}\ell(y', f(x' \mid X_{1:n_x}, y_{1:n_x}))$, where ℓ is a continuous loss function, $x', y' \sim G(\phi')$ and $\phi' \sim H(\gamma)$.

Recall we are only given a clustering approximation $\hat{c}$ and do not observe c – thus even measuring the out-of-cluster loss remains a difficult proposition, for reasons mentioned previously in Section 1.1.

1.2.2 Thesis Statement

Together, improved clustering performance and learning on imperfect clustering allows clustering to be integrated into more complex machine learning systems, where the clustering output is utilized to perform some task. These two essential objectives are summarized in our thesis statement:

Thesis Statement: Clustering errors cause subtle and adverse behavior in machine learning systems. These errors can be theoretically bounded, characterized, and corrected for using novel estimators.

This thesis, though not the first to tackle the problem of improving clustering performance, is the first to study the problem of how to learn on imperfect clusterings. We argue that since it is unrealistic to expect perfect clustering performance, knowing how to learn on imperfect clusterings is an equally essential component for practical machine learning systems.

1.3 Thesis outline

This thesis is broken in two parts. For chronological reasons, we first address the problem of clustering performance through new theoretical bounds and algorithms in Part I, and then proceed to how to learn on the clustering output in Part II. We emphasize that although Part I may require some clustering background, Part II is accessible to most audiences with some statistical background. Furthermore, we have made every attempt to make each part as general as possible, such that although Part II will benefit from the results of Part I, it is in fact agnostic to the particular clustering algorithm. In fact, for most readers, we recommend reading Part II first as it addresses a completely novel problem, and returning to the chapters of interest in Part I.

In Part I, we study the classical problem of clustering from several perspectives. Each chapter is self-contained, and provides as an output some clustering approximation $\hat{c}$. In Chapter 2, we consider graphical clustering approaches where edges between samples X are assumed to be generated according to a *planted partition* or *stochastic block model*. In these models, edges between samples are generated according to some unknown probability which depends on which clusters the samples belong to. We expect there to be a much higher probability of observing within-cluster edges than between-cluster edges, and can formulate the problem as a maximum-likelihood objective. Our key contribution here is leveraging semi-supervised information in the form of some *labeled edges*, which are more straightforward for humans to label than the daunting task of manually clustering a large dataset. We can then reduce the maximum-likelihood estimation problem into an instance of a known approximation algorithm.

In Chapter 3, we prove error bounds for the popular Swoosh algorithm, which is widely used in record linkage applications such as with hospital patient databases. Swoosh is unique compared to other clustering algorithms in that it allows both the *matching* of records (i.e. 'Do these two records belong in the same cluster?') and the *merging* of records (i.e. 'If the name in one record is Jane D. and the name in another is J. Doe, then their full name must be Jane Doe'). Our bounds are again empirically tight, and allow us to derive lower-bound optimal merge functions for the clustering algorithm.

In Chapter 4 we consider a Bayesian model for how the samples X are generated, which allows us to both infer the clustering approximation $\hat{c}$ and latent cluster parameters (such as ϕ in Eq. (1.1)) using MCMC. Our contribution here are novel bounds, which we show are empirically tight, characterizing exactly how clustering performance degrades with respect to key parameters.

We then proceed to Part II, which addresses the interaction between clustering errors and downstream prediction algorithms. It takes as an input a clustering approximation $\hat{c}$ from Chapter 2, Chapter 3, Chapter 4 or any other clustering algorithm. In Chapter 5, we characterize the behavior of the interaction effects between clustering and prediction algorithms. Specifically, we prove certain adverse properties hold under various reasonable conditions. Fortunately, we are able to leverage these same properties to develop a simple hypothesis test for whether interaction effects are present in a machine learning system. From a practitioner's standpoint, this may be one of the most powerful takeaways of this thesis.

In Chapter 6, we propose the key algorithm to correcting for interactions between clustering and prediction algorithms: the Binomial Block Bootstrap (B3) estimator. For any predictor f, the B3 estimator provides an unbiased and asymptotically consistent estimate of the out-of-cluster loss for f on the true clustering c, given only an approximate clustering $\hat{c}$. In other words, the B3 estimator allows proper cross-validation on noisy clusters, for any learner, a somewhat surprising result. We provide empirical evidence that cross-validating on noisy clusters can bias cross-validation results by upwards of 25%, and that the B3 estimator is able to provide much better estimates even in the finite sample setting.

Finally, we scale the B3 estimator to larger datasets in Section 6.3 by basis function and matrix sketching approximation techniques. These tools make it possible to reduce the computational complexity from $\mathcal{O}(n'^3)$ to $\mathcal{O}(1)$, where n' is the size of the training dataset. In practice, this allows the B3 estimator to scale to previously intractable problem classes and reduces solution times by multiple orders-of-magnitude on smaller datasets.

Part I

New theory and algorithms for clustering

Chapter 2

Stochastic block models and correlation clustering

Graphical approaches to clustering are appealing because they offer a natural way to compare samples, in the form of edge information. However, which graph to use for clustering remains an open question [100]. Previous work has considered edges to be the output of a similarity function[1] (e.g. spectral clustering), a Bernoulli random variable (e.g. stochastic block models), or some more general measure of similarity/dissimilarity (e.g. correlation clustering).

In reality, edge information can take a variety of forms. Edges in social graphs correspond to communication exchanges, mutual interests and types of relationships. In biology, protein-protein interaction networks involve complex underlying mechanisms and conditions under which an event may occur. And in economics, trades constitute numerous goods, prices and transaction types.

We are inspired by the complex interactions happening around us. Our relationships are more complicated than friend/not-friend, and our transactions are about more than the monetary value. The motivation of this chapter is to cluster with the additional information provided by multivariate edge features. This is partly supported by Thomas and Blitzstein's [97] recent results showing that converting to a binary graph makes recovering a partition more difficult. We are also interested in how to choose similarity functions which better capture the relationship between nodes,

[1] A similarity function is a function of two nodes' features, e.g. the RBF kernel

one of the challenges of spectral clustering [100]. Choosing a scalar similarity function (e.g. the RBF kernel) may be overly restrictive and underutilize useful information. This is partly the cause of scale issues in spectral clustering [112]. Our approach allows more complex similarity functions, such as the absolute vector difference.

We believe these results will be particularly useful for image segmentation, community discovery and entity resolution. These are all applications (a) with a large number of clusters and (b) where we have access to some labeled edges. With a large number of clusters, it is unlikely we have training samples from every class, let alone enough samples to train a multi-class supervised classifier. However, the small number of labeled edges will enable us to learn the typical cluster structure.

In this chapter, we extend the planted partition model to general edge features. We also show how to partially recover a maximum likelihood estimator which is $\mathcal{O}(\log(n))$-close to the log likelihood of the true MLE by using an LP-rounding technique. Much of the analysis in planted partition models consider the probability of exactly recovering the partition. Depending on the cluster sizes and number of samples, this is often improbable. Our analysis addresses how good the result will be, regardless if it is exactly correct. Further, our theoretical results provide some insights on how to perform edge feature selection or, likewise, how to choose a similarity function for clustering. Experimental results show interesting clustering capabilities when leveraging edge feature vectors.

2.1 Prior Work

Two areas of research are closely related to our work. Our graphical model is an extension of the stochastic block model from the mathematics and statistics literature. We also use some key results from correlation clustering in our algorithm and analysis.

2.1.1 Stochastic Block Model

The stochastic block model (SBM) was first studied by Holland et al. [39] and Wang and Wong [102] for understanding structure in networks. In its simplest form, every edge in the graph corresponds to a Bernoulli random variable, with probability depending on the two endpoints' clusters. In planted partition models[2] there are two Bernoulli probabilities p and q corresponding to if the endpoints are in the same or different clusters, respectively. These models are actually generalizations of the Erdős-Rényi random graph, where $p = q$. Random graph models have a storied history and include famous studies such as the small-world experiment (popularized as "six-degrees of separation") by Milgram [64] and Zachary's Karate Club network [111]. For a more complete overview, we refer the interested reader to the review by Goldenberg et al. [34].

More recently, planted partition models have gained popularity in the machine learning community for clustering. McSherry [61] and Condon & Karp [22] provided early spectral solutions to exactly recovering the correct partition, with probability depending on a subset of the parameters p, q, the number of samples n, the number of clusters k, and the smallest cluster size. Most results show recovering the partition is more difficult when p and q are close, n is small, k is large, and the smallest cluster size is small. Intuitively, if there are a high proportion of singleton clusters (i.e. "dust"), mistaking at least one of them for noise is likely.

Some of the numerous alternative approaches to recovering the partition include variational EM [3, 24, 72], MCMC [72], and variational Bayes EM [1, 38]. Some of these approaches may also be applicable to the model in this chapter, though we found our approach simple to theoretically analyze.

The work most closely related to ours extends the stochastic block model edge weights to other parametric distributions. Motivated by observations that Bernoulli random variables often do not capture the degree complexity in social networks, Karrer & Newman [41], Mariadassou et al. [59] and Ball et al. [6] each used Poisson distributed edge weights. This may also be a good choice because the Bernoulli degree distribution is asymptotically Poisson [110]. Aicher et al. considered an SBM with edge weights drawn from the exponential family distribution [1]. Like Thomas & Blitzstein [97], he also showed better results than thresholding to binary edges. Lastly, Balakrishnan et al. [5] consider Normally distributed edge weights as a method of analyzing spectral clustering recovery with noise.

Other interesting extensions of the SBM include mixed membership (i.e. soft

[2] There is some inconsistency in the literature regarding the distinction between planted partition and stochastic block models. Occasionally the terms are used interchangeably

clustering) [3], hierarchical clustering [5, 21] and cases where the number of clusters k grows with the number of data points n [19, 76]. Combining our ideas on general edge features with these interesting extensions should be possible.

2.1.2 Correlation Clustering

Correlation clustering was introduced by Bansal et al. [8] in the computer science and machine learning literature. Given a complete graph with ±1 edge weights, the problem is to find a clustering that agrees as much as possible with this graph. There are two 'symmetric' approaches to solving the problem. MinimizeDisagreements aims to minimize the number of mistakes (i.e. $+1$ inter-cluster and -1 intra-cluster edges), while MaximizeAgreements aims to maximize the number of correctly classified edges (i.e. -1 inter-cluster and $+1$ intra-cluster edges). While the solutions are identical at optimality, the algorithms and approximations are different.

The original results by Bansal et al. [8] showed a constant factor approximation for MinimizeDisagreements. The current state-of-the-art for binary edges is a 3-approximation [2], which Pan et al. [70] recently parallelized to cluster one billion samples in 5 seconds. Ailon et al. [2] also showed a linear-time 5-approximation on weighted probability graphs and a 2-approximation on weighted probability graphs obeying the triangle inequality. Demaine et al. [26] showed an $\mathcal{O}(\log(n))$-approximation for arbitrarily weighted graphs using the results of Leighton & Rao [50]. Solving MinimizeDisagreements is equivalent to APX-hard minimum multi-cut [17, 26].

For MaximizeAgreements, the original results by Bansal et al. [8] showed a polynomial-time approximation scheme (PTAS) on binary graphs. State-of-the-art results for non-negative weighted graphs are a 0.7664-approximation by Charikar et al. [17] and similar 0.7666-approximation by Swamy [94]. Both results are based on Goemans and Williamson [33] using multiple random hyperplane projections.

Later, we will use correlation clustering to partially recover the maximum likelihood estimator of our planted partition model. Kollios et al. [44] consider a similar problem of using correlation clustering on probabilistic graphs, although their algorithm does not actually solve for the MLE.

2.2 Problem Statement

Consider observing an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |V|$ vertices. Let $\psi : \{1, \ldots, n\} \rightarrow \{1, \ldots, k\}$ be a partition of the n vertices into k classes. We use the notation $\psi_{ij} = 1$ if nodes i and j belong to the same partition, and $\psi_{ij} = 0$ else. Edges $e_{ij} \in \mathbb{R}^d$ are d-dimensional feature vectors. Note we say that graph G is 'observed,' though the edges E may also be the result of a symmetric similarity function s, where $e_{ij} = s(v_i, v_j)$.

We assume a planted partition model, $e_{ij} \sim P(e|\psi_{ij})$. From now on, we will use the shorthand $P_0(\cdot) = P(\cdot|\psi_{ij} = 0)$ and $P_1(\cdot) = P(\cdot|\psi_{ij} = 1)$. In the conventional planted partition model, P_0 and P_1 are Bernoulli distributions with parameter q and p, respectively. However, in this work we make no assumptions about the probability density functions P_0 and P_1. We will make the key assumption that all stochastic block models make – that the edges are independent and identically distributed, conditioned on ψ. Note if the edges E are generated by a similarity function then it is unlikely the edges are actually independent, but we proceed with this assumption regardless.

In most planted partition models, the goal is to either partially or exactly recover ψ after observing G. We aim to find the most likely partition, and bound our performance in terms of the likelihood. There is a subtle distinction between the two goals. Even if the maximum likelihood estimator is consistent, the non-asymptotic MLE may be different than the true partition ψ.

2.3 Approximating the Maximum Likelihood Estimator

Let $\theta : \{1, \ldots, n\} \rightarrow \{1, \ldots, k\}$ be a partition under consideration. In exact recovery, our goal would be to find a θ such that $\theta = \psi$. However, our goal is to find a partition $\hat{\theta}$ which is close to the likelihood of the maximum likelihood estimator θ_{MLE}. Using the edge independence assumption, the likelihood L is

$$L(\theta) = \prod_{i<j} P(e_{ij}|\theta_{ij}) \mathbb{1}(\theta \in \Theta) \tag{2.1}$$

where Θ is the space of all disjoint partitions.

The trick to finding an approximation $\hat{\theta}$ to the MLE θ_{MLE} is to reduce the problem to a correlation clustering instance. Consider forming a graph $G_O = (V, E_0)$ with binary edges defined by the sign of the log-odds $e_{0;ij} = \text{sign}\left(\log\left(P_1(e_{ij})/P_0(e_{ij})\right)\right)$. Let the cost of mislabeling each edge be the absolute log-odds $C_{ij} = \left|\log\left(P_1(e_{ij})/P_0(e_{ij})\right)\right|$. Then we can rewrite the log-likelihood ℓ as[3]

$$\begin{aligned} \ell(\theta) &= \ell(G_0) - \sum_{\theta_{ij} \neq e_{0;ij}} \left|\log\left(\frac{P_1(e)}{P_0(e)}\right)\right| \\ &= \ell(G_0) - \sum_{\theta_{ij} \neq e_{0;ij}} C_{ij} \end{aligned} \tag{2.2}$$

Maximizing $\ell(\theta)$ is equivalent to minimizing $\sum_{\theta_{ij} \neq e_{0;ij}} C_{ij}$, which is exactly MinimizeDisagreements where edges are labeled according to E_0 and have weighted costs $C \geq 0$. Intuitively, we consider the most likely graph G_0 (which is not a valid partition) and try to find the minimum number of weighted edge flips required to create a valid partition.

Unfortunately, we only have non-negativity bounds on the weights C. Thus we believe the only appropriate MinimizeDisagreements algorithm to solve Eq 2.2 is the LP-rounding technique by Demaine et al. [26].

Theorem 1. *The above estimated clustering $\hat{\theta}$ is $c_1 DIS \log(n)$-close to the log-likelihood of the true maximum likelihood estimator $\hat{\theta}_{MLE}$. This is an $\exp(-DIS(c_1 \log(n) - 1))$-approximation algorithm for the likelihood.*

The constant $c_1 = 2 + 1/\log(n+1)$ is just slightly larger than 2. *DIS* is a measure of disagreement between the graph G_0 and the optimal clustering, to be discussed shortly.

Proof. The results follow directly from Leighton & Rao [50] and Demaine et al. [26]. Let *DIS* be the optimal solution to MinimizeDisagreements on graph G_0 with weighs C. Then the log likelihood of the true MLE θ_{MLE} is

$$\ell(\theta_{MLE}) = \ell(G_0) - DIS \tag{2.3}$$

[3] G_0 is not required to be a valid partition and thus the $\mathbb{1}(G_0 \in \Theta)$ term is not included in $\ell(G_0)$. However, θ is still required to be a valid partition.

Demaine et al. [26] showed an $c_1 \log(n)$-approximation to MinimizeDisagreements on general weighted graphs. Thus the approximated MLE using this algorithm will yield

$$\ell(\hat{\theta}) \geq \ell(G_0) - c_1 \log(n) DIS \tag{2.4}$$

The approximation ratio result follows likewise.

$$\begin{aligned} L(\hat{\theta}) &\geq L(G) \exp(-c_1 \log(n) DIS) \\ L(\theta_{MLE}) &= L(G) \exp(-DIS) \\ \frac{L(\hat{\theta})}{L(\theta_{MLE})} &\geq \exp(-DIS(c_1 \log(n) - 1)) \end{aligned}$$

□

2.3.1 Choosing Edge Features or Similarity Functions

How to choose a similarity function remains a fundamental question in spectral clustering [100]. A “meaningful” similarity function should have high similarity for samples belonging to the same cluster and low similarity for samples in different clusters, but how to judge that remains unclear. In practice, the radial basis function is commonly used and often provides favorable results. More precisely, we want to know which similarity functions make clustering easier and understand why they do.

This same question applies when doing edge feature selection. We want to choose features which are most informative for clustering and ignore the others. We can provide a more scientific answer to these questions by analyzing the *DIS* coefficient.

Theorem 2. *Let n_0 and n_1 be the number of inter and intra-cluster edges in ψ, respectively. Then*

$$\mathbb{E}[DIS] = -n_1 D_{KL}(P_1 || P_0)\Big|_{P_1 \leq P_0} - n_0 D_{KL}(P_0 || P_1)\Big|_{P_0 \leq P_1} \tag{2.5}$$

where we use the notation $D(\cdot||\cdot)\Big|_S$ to denote the divergence evaluated only over the closed set S.

Proof.

$$
\begin{aligned}
\mathbb{E}[DIS] &= (n_1 + n_2)\mathbb{E}_{\psi_{ij} \neq e_{0;i,j}}[C] \\
&= n_1 \mathbb{E}_{\psi_{ij}=1, e_{0;i,j}=0}[C] + n_2 \mathbb{E}_{\psi_{ij}=1, e_{0;i,j}=0}[C] \\
&= n_1 \int_{P_1(e) \leq P_0(e)} P_1(e) \log\left(\frac{P_0(e)}{P_1(e)}\right) de \\
&\quad + n_2 \int_{P_0(e) \leq P_1(e)} P_0(e) \log\left(\frac{P_1(e)}{P_0(e)}\right) de \\
&= -n_1 D_{KL}(P_1 || P_0)\Big|_{P_1 \leq P_0} - n_0 D_{KL}(P_0 || P_1)\Big|_{P_0 \leq P_1}
\end{aligned}
$$

□

Notice these restricted Kullback-Leibler divergences are always negative, and thus $\mathbb{E}[DIS] \geq 0$.

The intuition here is to choose edge features or similarity functions which are unlikely to create edges in the disagreement regions (i.e. edges which contribute to DIS). If P_0 and P_1 are completely divergent, then exactly recovering the partition is trivial because G_0 will be the set of disconnected cliques induced by ψ. Additionally, when mistakes are made, we want the KL divergence to be small (i.e. the mistake is not too 'bad').

Along these lines, choosing higher dimensional edge features and similarity functions (e.g. the absolute vector difference instead of the Euclidean distance) makes clustering easier, by decreasing the disagreement region between P_0 and P_1. This confirms our earlier motivation that useful clustering information may be lost by only considering binary or scalar edge features and functions.

Considering only this approximation ratio when choosing a similarity function or edge features does not quite capture the complete picture. A trivial solution is select for $P_0 = P_1$ (a type of an Erdős-Rényi random graph), which results in an approximation ratio of 1. Since every partition is equally likely in this scenario, finding an approximation to the MLE is trivial. However, exactly recovering ψ is unlikely.

Sparsity

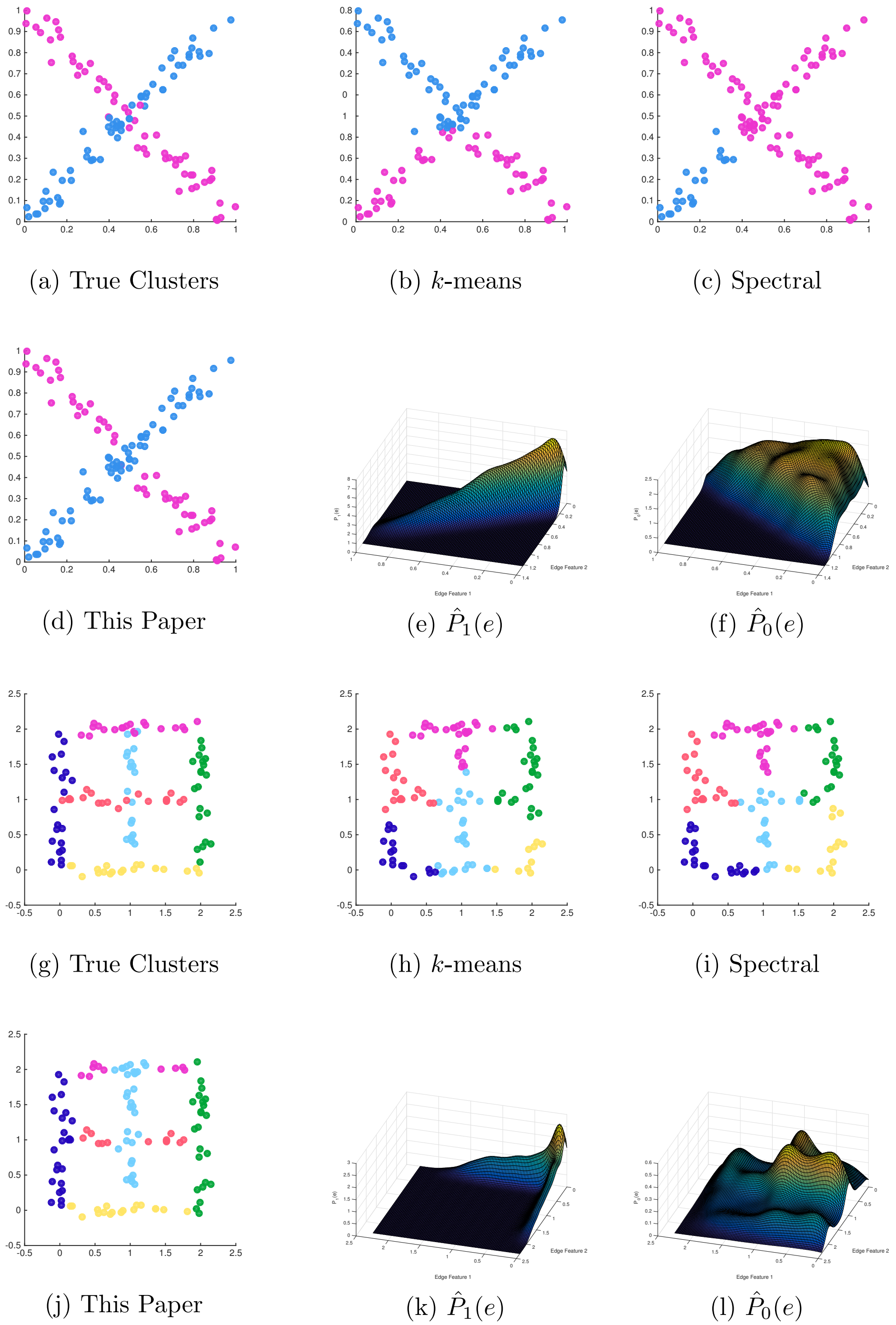
In many situations it is advantageous to induce sparsity into the graph G. Spectral clustering employs this trick to cluster large graphs, by only considering the most similar nodes when performing eigen decompositions. In the proposed approach, sparsity will also reduce the number of variables and constraints in the LP used to maximize Eq 2.2.

By the previous analysis, we want to choose a similarity function or edge features which achieve the desired sparsity while maintaining a small *DIS* coefficient. In the MinimizeDisagreements problem, sparse edges will have cost $C_{ij} = 0$. This occurs when $P_0(e_{ij}) = P_1(e_{ij})$. Intuitively, the best edges to sparsify are the ones which we do not have strong evidence for whether they should be labeled positive or negative. For these edges, the probabilities P_0 and P_1 will be close. Unlike spectral clustering, which only considers the most similar edges, this sparsification considers the most similar *and* the most dissimilar edges.

2.4 Experiments

We experimentally demonstrate the performance of the proposed model and algorithm on several synthetic and real world datasets. Specifically, we show studying edge features enables learning the *structure* of clusters. When compared to k-means and spectral clustering, the planted partition model with general edge features can correctly cluster some rather interesting examples which are not attainable with scalar similarity functions. And more importantly, it seems to outperform existing methods on real world datasets.

In practice, it is unlikely we have access to P_0 and P_1. By assuming a prior parametric distribution, previous approaches have inferred these distributions while simultaneously learning the clustering. To remain as general as possible, we do not make any prior assumptions on P_0 and P_1. Our focus here is different. In datasets where the number of classes is very large or when new previously unseen classes are introduced, it is unlikely we can perform traditional supervised classification. This is especially true in the entity resolution and record linkage domains, where clusters correspond to millions or billions of entities (e.g. people, businesses, items) and new



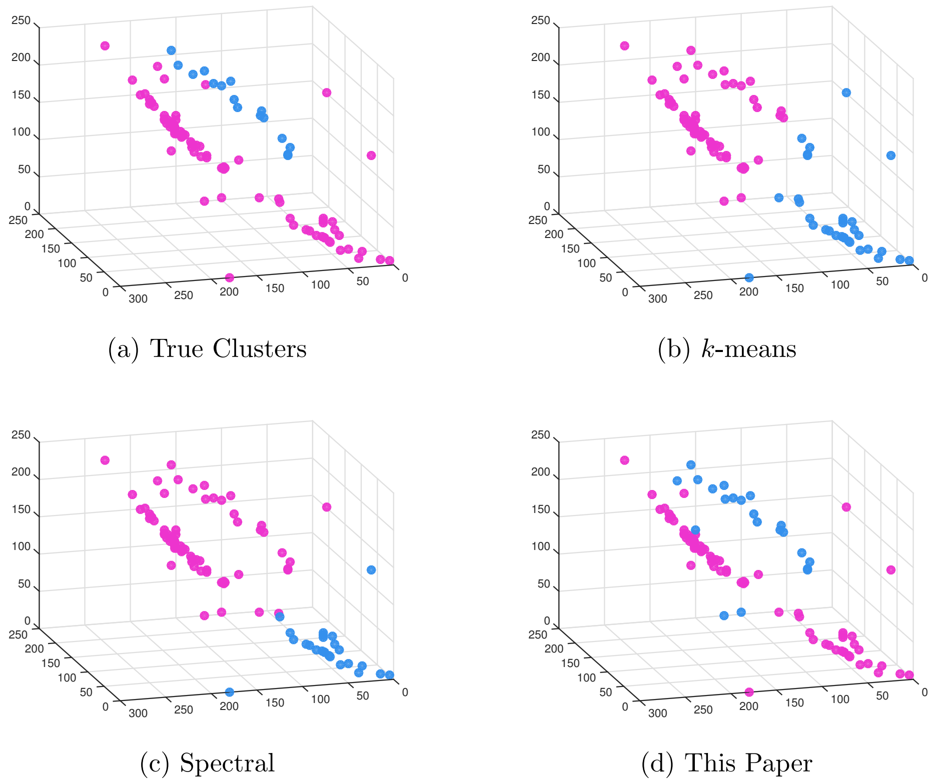
(a) True Clusters (b) k-means (c) Spectral

(d) This Paper (e) $\hat{P}_1(e)$ (f) $\hat{P}_0(e)$

(g) True Clusters (h) k-means (i) Spectral

(j) This Paper (k) $\hat{P}_1(e)$ (l) $\hat{P}_0(e)$

Figure 2.1: Results on synthetic 2D datasets. The true number of clusters k is given as an input to k-means and spectral clustering, while our model naturally learns the correct number of clusters. $\hat{P}_0$ and $\hat{P}_1$ are the learned edge densities.



(a) True Clusters

(b) k-means

(c) Spectral

(d) This Paper

Figure 2.2: Results on the UCI Skin Segmentation dataset. Pink represents skin samples and blue represents non-skin samples. The axes correspond to RGB pixel values. The associated normalized mutual information scores are (b) 0.0042, (c) 0.1016 and (d) 0.6804.

entities are frequently introduced. Statistical networks and image segmentation also exhibit this property.

In these problems, we frequently have access to labeled *pairs*. Manually labeling whether two samples describe the same person is a straightforward task for human adjudicators, compared to manually clustering a large number of samples. This is the information we use to learn the cluster structure. Thus we assume we have access to some labeled pairs in order to learn P_0 and P_1. Standard dimensionality reduction techniques can be employed to perform analysis in a reasonable space. We show this is still more powerful than using conventional scalar similarity functions. For all of our experiments, we use kernel density estimation to estimate P_0 and P_1. To improve the dimensional scalability, we could also perform a single estimation of P_1/P_0 using

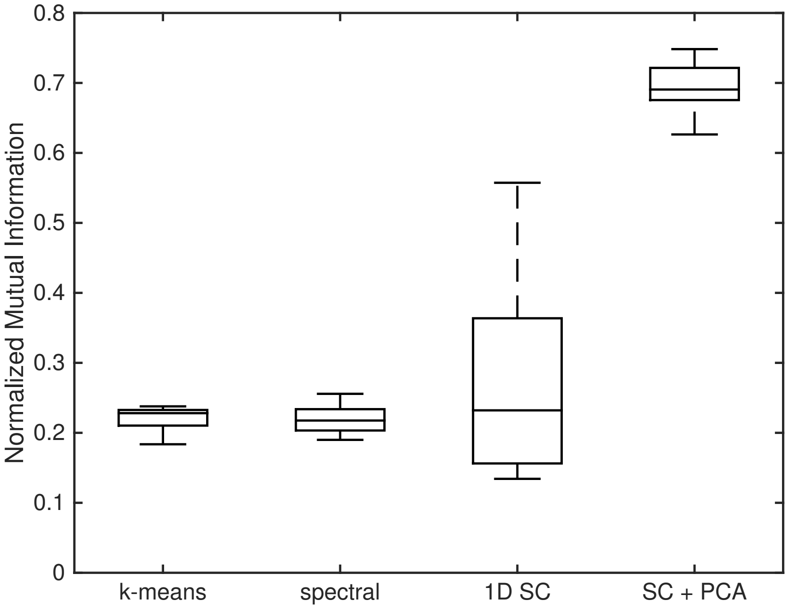


Figure 2.3: Structured clustering with PCA (`SC + PCA`) outperforms competitors on the 11 class, 48 dimensional UCI Sensorless Drive dataset. `1D SC` is the same structured clustering model, except using the simpler Euclidean similarity function ($e_{ij} = ||v_i - v_j||_2$). Boxes correspond to the 25^{th} and 75^{th} percentile of 17 trials. Whiskers are the most extreme values.

direct density ratio estimation [40].

To compare performance we evaluate against k-means and spectral clustering [82]. Per the recommendations of Luxburg [100], we use the Gaussian similarity function, a mutual k-nearest neighbor graph where $k = 20$ and the random walk graph Laplacian. Unless otherwise noted, the edge features used for our method are from the absolute vector difference function $e_{ij} = |v_i - v_j|$ and thus not independent. However, the results indicate that it may be an acceptable assumption.

Lastly, our model consistently and naturally learns the correct number of clusters k. We found it occasionally labeled outlier samples as singleton clusters, though this would have a very small impact on the normalized mutual information score. For k-means and spectral clustering we do provide k as an input. There are certainly methods of estimating k for these competitors (e.g. analyzing the spectral gap), although they are not intrinsic to the methods.

2.4.1 Results on Synthetic Data

We consider the two interesting synthetic examples shown in Figure 2.1. Traditional clustering algorithms such as k-means and spectral clustering are unable to correctly label these examples because the clusters occasionally cross each other. Our method is able to capture the unidirectional cluster structure, and thus correctly label the samples. This is not an occasional event, in fact we have yet to see our method fail on these examples.

For all the synthetic experiments, we estimated P_0 and P_1 using 5,000 labeled pairs and clustered 100 hold-out samples. We use the absolute vector difference as our similarity function, which is able to capture the distance *and* direction, unlike the Gaussian similarity function. There may be other excellent choices for similarity function, this is the only one we have tried so far.

We have achieved comparable results on the classic Gaussian, two moons, concentric circles and swiss roll examples. There was little distinction between our method and spectral clustering on these problems, so they were omitted from this chapter.

2.4.2 Results on Real World Data

The first real world data we consider is the UCI Skin Segmentation dataset[4], shown in Figure 2.2. Samples are RGB values and labeled according to whether they are skin or non-skin image pixels. Again, we estimated P_0 and P_1 using 5,000 labeled pairs and clustered 100 hold-out samples, and use the absolute vector difference similarity function.

Visually, this seems much easier than the previous synthetic examples. However, k-means and spectral clustering are still unsuccessful due to the data scale issue introduced by the oblong cluster nature. Feature whitening did not help the competitors, though we believe some extensions to the standard spectral clustering may be able to handle this type of data [112].

The second realistic example we consider is the UCI Sensorless Drive Diagnosis dataset[5]. Features are derived from current and frequency measurements in defective electric motors, including the statistical mean, standard deviation, skewness and

[4] https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation

[5] https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis

kurtosis of intrinsic mode function subsequences. In total, there are 48 features and 11 classes.

We repeat the same previous procedures, except we additionally perform principal component analysis on the training and hold-out edge features prior to estimating P_0 and P_1 (`SC + PCA`). We also consider one dimensional features using the Euclidean distance similarity function (`1D SC`). The results from 17 trials are shown in Figure 2.3. The strong performance on the PCA reduced edge features leads us to believe that even if the original vertices have high dimensional structure, the distinguishing edge features in clusters have a lower dimensional representation.

2.5 Conclusions

Overall, incorporating multivariate edge features and more powerful similarity functions improves performance in all the experiments we have conducted. And even when the edges are clearly not independently generated, our structured clustering model still outperforms competitors.

The key insight from our approach is that multidimensional edge features can be used to effectively learn structure in clusters. Relationships in real world data are more complex than a simple scalar similarity function, and our methods can benefit from capturing that additional complexity. Then we can use the learned cluster structure to both determine the correct number of clusters and to handle situations where we are given new, previously unseen clusters, by assuming similar structure.

Applications which may especially benefit from structured clustering usually (a) have some labeled edges to learn P_0 and P_1 and (b) have a large number of clusters which make training a supervised classifier impractical. For example, in community detection and entity resolution, we have many examples of communities or entities to learn P_0 and P_1, though we certainly do not have examples of every community and entity to perform classification. Intuitively, we expect communities and entities to exhibit some common behavior, and we can leverage this structure while clustering. In image segmentation we usually have many images with human labeled segments, but the segments (i.e. classes) in new images are likely of a different object. However, it is not unreasonable to assume the *structure* of these new segments is similar to the previously seen segments.

We used the approximation algorithm by Demaine et al. [26] for MinimizeDisagreements. The solution for this sub-problem is not the main focus of this chapter, and unfortunately this particular algorithm requires solving a large linear program which limited the scalability of our experiments. Pan et al. recently clustered 1 billion samples in 5 seconds using a parallelizable, linear time algorithm for MinimizeDisagreements, but only with edge weight restrictions [70].

Other interesting extensions include applying the same method to stochastic block models, which would require estimating a separate P_0 and P_1 for every pair of blocks. In record linkage problems the same technique could be used to cluster vertices with different feature types. For example, clustering *across* multiple social networks is of particular interest for advertising and law enforcement.

We have independently provided similar analysis for MaximizeAgreements by extending the results of Swamy [94] and Charikar et al. [17] to graphs with negative edge weights, though the theoretical and experimental results were not as convincing as MinimizeDisagreements.

Chapter 3

Match-and-merge for record linkage

In addition to having access to a pairwise match function, as studied in Chapter 2, we now consider the additional availability of a merge function $\langle \cdot, \cdot \rangle : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$. In many domains such as census or medical records, we have a priori knowledge about the structure of the data that naturally lends itself to a merge function. For example, consider two records x_1 and x_2 with name features "B Obama" and "Barack O," respectively. Clearly, if these two records match, i.e. $m(x_1, x_2) = 1$, then $\langle x_1, x_2 \rangle =$ "Barack Obama."

In this chapter, we prove the first known theoretical performance bounds for a class of clustering algorithms known match-and-merge record linkage [1] and demonstrate their practical usage on multiple real world datasets. We also propose a record linkage algorithm which is optimal with respect to the lower bounds and show its connection to graph clustering techniques. In the traditionally algorithm driven domain of record linkage, our bounds not only provide a more formal understanding of record linkage, but also enable better performance in practice.

[1] There are conflicting distinctions between the terms clustering and record linkage [89]. Here, we default to the term more common in the respective area of research, which in this chapter is record linkage.

3.1 Prior Work

Fellegi and Sunter's seminal work [29] created a formal statistical foundation for record linkage based on the earlier work by Newcombe [68, 69]. For fixed bounds on certain error rates, they proved an algorithm which minimized the number of records sent for clerical review (i.e. human adjudication). The original Fellegi-Sunter model, which many modern approaches are built upon, only considers linking two 'clean' sets of records. A record set is 'clean' if there are no intra-set links. For example, Zagat's review website is 'clean' because it does not have multiple listings for the same restaurant. Attempting to merge the Zagat's listings with another 'clean' set (i.e. Yelp) would be a Fellegi-Sunter record linkage problem.

Recently there has been an effort to extend the Fellegi-Sunter model to more general settings. Sadinle and Fienberg generalized the model to linking multiple 'clean' sets of records [78]. McCallum and Wellner formalized the concept of transitive closure, but do not provide any theoretical guarantees [60, 84]. In this chapter, we consider the most general record linkage problem: a single set of records with no restrictions on links, informally referred to as a 'dirty' record set. For example, the set of all listings on eBay is a dirty set. Any two listings could potentially describe the same product, and cluster sizes are unrestricted.

Note our approach also applies to the 'clean' setting, because the union of clean sets is a dirty set. The more general 'dirty' setting is interesting because it encompasses a broad set of problems not considered by the original Fellegi-Sunter model, including co-reference resolution in natural language processing, near duplicate detection in search engines and image segmentation in computer vision.

Perhaps the work most closely related to ours is on estimating the error rates associated with the Fellegi-Sunter model. Belin et al. [10] and Winkler [107] used unsupervised methods, which apply when the match and mismatch classes are separable. With small amounts of training data, Larsen [49] and Winkler [106] are able to more accurately estimate the error rates. Our work builds on the latter approaches, by considering performance on the general record linkage problem. In continuation with trends towards learning-based approaches, we do not send any records to clerical review. Besides reducing the human adjudication cost, Newcombe and Smith noted that fully automated record linkage often outperformed combinations of computers and highly trained clerks [85].

One of the best frameworks for the general record linkage setting was presented by Benjelloun et al. [11]. They outlined a theoretically disciplined approach, wherein certain properties of the record linkage match and merge functions guarantee a deterministic output in the optimal number of record comparisons. We exploit the use of some of these properties in the derivation of our performance bounds.

3.2 Problem Statement

We consider the record linkage problem where we are given a large set of unlabeled records $X = (x_1, \dots, x_n) \overset{\text{iid}}{\sim} P(x)$ (e.g. medical records, noun phrases, product descriptions) and need to infer the entity label y for each record $x \in X$. In the semi-supervised setting, we also have access to a small set of pairs of samples which belong to the same cluster $V_T = (x_{n+1}, x_{n+2}), \dots (x_{n+2n_T-1}, x_{n+2n_T}) \overset{\text{iid}}{\sim} P(x_i, x_j | y_i = y_j)$ and another set of pairs which belong to different clusters $V_F = (x_{n+2n_T+1}, x_{n+2n_T+2}), \dots (x_{n+2n_T+2n_F-1}, x_{n+2n_T+2n_F}) \overset{\text{iid}}{\sim} P(x_i, x_j | y_i \neq y_j)$. Collectively, these labeled pairs form a validation set $V = V_T \cup V_F$ with corresponding class balance $C_V = |V_T|/|V|$. Forming such a pairwise validation set is a natural form of semi-supervision. Pairs of samples are randomly sampled from validation set and then sent to human adjudicators to decide whether the two samples describe the same latent entity. This follows the same type of human adjuciation as described in [29]. We note that good pairwise performance of m on V does not imply similar clustering performance on X because the complexity of the clustering problem increases as the number of entities increases and depends on the record linkage algorithm itself.

Our goal is to bound the performance of a record linkage algorithm $\mathcal{A}$ on the unlabeled records X, where $\mathcal{A}(X)$ produces predicted cluster labels $\hat{Y}$. Specifically, we make the following assumption about $\mathcal{A}$.

A1) Record linkage algorithm $\mathcal{A}$ is composed of a binary match function $m(x_i, x_j)$ and merge function $\langle x_i, x_j \rangle$. $\mathcal{A}$ merges any two matching records and terminates only when no matching records remain.

The binary match function $m(x_i, x_j)$ predicts whether two records $x_i, x_j \in X$ describe the same latent entity (i.e. whether $y_i = y_j$) and outputs `True` or `False`.

In learning-based record linkage, the match function is usually trained on some of the records in X'. The match function may be any classifier that satisfies several assumptions to be discussed later.

If the binary match function m determines two records describe the same latent entity (i.e. $m(x_i, x_j) = \texttt{True}$), then the records are merged into a new record $x_{new} = \langle x_i, x_j \rangle$. The merge function $\langle x_i, x_j \rangle$ may include domain specific knowledge on how to best merge specific features (e.g. phone numbers 377-8328 and 412-377-8328 should resolve to the latter, as the former is likely missing an area code) or it may be something more general, such as the set union operator for each feature.

We consider the class of all match functions $m(x_i, x_j)$ and merge functions $\langle x_i, x_j \rangle$ that satisfy the following assumptions:

A2) Idempotence: $\forall x \in \mathcal{X}$, $m(x, x) = \texttt{True}$ and $\langle x, x \rangle = x$. A record always matches itself and merging a record with itself yields the same record.

A3) Commutativity: $\forall x_i, x_j \in \mathcal{X}$, $m(x_i, x_j) = \texttt{True}$ iff $m(x_j, x_i) = \texttt{True}$, and if $m(x_i, x_j) = \texttt{True}$, then $\langle x_i, x_j \rangle = \langle x_j, x_i \rangle$. The match and merge functions are symmetric.

A4) Associativity: $\forall x_i, x_j, x_k \in \mathcal{X}$ such that $\langle x_i, \langle x_j, x_k \rangle \rangle$ and $\langle \langle x_i, x_j \rangle, x_k \rangle$ exist, $\langle x_i, \langle x_j, x_k \rangle \rangle = \langle \langle x_i, x_j \rangle, x_k \rangle$. In other words, the ordering of merges does not matter.

A5) Representativity: If $x_k = \langle x_i, x_j \rangle$ then for any x_l such that $m(x_i, x_l) = \texttt{True}$, we also have $m(x_k, x_l) = \texttt{True}$. An important consequence of representativity is that merging any two records can only monotonically increase their probability of matching with other records. This is sometimes referred to as the 'no negative evidence' clause.

Collectively, Assumptions A2-A5 are referred to by their acronym ICAR. A convenient property of record linkage algorithms satisfying the ICAR assumptions is that when run until no matches remain, the output is deterministic [11]. The first three properties are straightforward and reasonable to assume for most record linkage systems. The crux of determinism falls on the final property, representativity. This assumption may seem strong, but a deterministic record linkage algorithm derived from these properties is popular for database applications [11].

3.3 Lower Bounds of Performance

Our approach to bounding the performance of $\mathcal{A}(X)$ is to use the empirical performance of the match function on validation pairs V. We note that due to the small clustering property [89], $\mathcal{A}$ may perform poorly on X even if its match function performs well on a small validation set. We can bound performance of $\mathcal{A}(X)$ by deriving a particular relationship between the two performances.

We use the validation set of labeled pairs V to empirically estimate the performance of the match function m (we assume m is trained a priori on a disjoint set of training samples). Let V_M denote the pairs of records in V that m predicts are matches, $V_M = \{(x_i, x_j) : (x_i, x_j) \in V, m(x_i, x_j) = \texttt{True}\}$. Then the estimated precision and recall of the match function are $\mathit{Prec}(V_M, V_T) = |V_M \cap V_T|/|V_M|$ and $\mathit{Recall}(V_M, V_T) = |V_M \cap V_T|/|V_T|$, respectively.

Note the validation set is used to estimate performance of the match function m and we want to bound performance of $\mathcal{A}(X)$. The former is a measure of binary classification performance, whereas the latter is a measure of clustering quality. Specifically, we use *pairwise precision* and *pairwise recall*, which are the same as precision and recall in binary classification, except operating on the space of *record pairs*. For more information on record linkage metrics, we refer the interested reader to the review by Menestrina et al. [62].

Lemma 3. *For record linkage algorithms $\mathcal{A}(X) = \hat{Y}$ satisfying the representativity property (Assumption A5), every record pair that directly matches will resolve to the same entity.*

$$X_M \subseteq X_{\hat{Y}}. \tag{3.1}$$

where X_M is the set of record pairs in X that directly match and $X_{\hat{Y}}$ is the set of record pairs that belong to the same predicted cluster. Formally, $X_M = \{(x_i, x_j) : x_i, x_j \in X, i < j, m(x_i, x_j) = \texttt{True}\}$ and $X_{\hat{Y}} = \{(x_i, x_j) : x_i, x_j \in X, i < j, \hat{y}_i = \hat{y}_j\}$.

Intuitively, additional pairs in $X_{\hat{Y}}$ can occur from chains of matches. For example, consider the case where $m(x_i, x_j) = \texttt{True}$, $m(x_j, x_k) = \texttt{True}$, but $m(x_i, x_k) = \texttt{False}$. The record pair $(r_1, r_3) \notin X_M$, but $(r_1, r_3) \in X_{\hat{Y}}$. However, we are unable to make strong claims about the additional matches in $X_{\hat{Y}}$ since chains of records do not occur in the validation set (which only had pairs of unmerged samples).

Proof. Suppose on the contrary there exists a pair of records (x_i, x_j), such that $(x_i, x_j) \in X_M$ but $(x_i, x_j) \notin X_{\hat{Y}}$. In other words, $m(x_i, x_j) = \texttt{True}$ and they are resolved to separate entities $\langle x_i, ...\rangle$ and $\langle x_j, ...\rangle$. Since these clusters were not merged in the record linkage process, $m(\langle x_i, ...\rangle, \langle x_j, ...\rangle) = \texttt{False}$, which contradicts the representativity property. □

Theorem 4. *The pairwise precision of a record linkage result can be lower bounded by:*

$$\mathbb{E}\left[Prec(X_{\hat{Y}}, X_Y)\right] \geq \frac{|X_M|}{|X_{\hat{Y}}|}\mathbb{E}\left[\frac{C_X(1-C_V)Prec(V_M, V_T)}{C_V(1-C_X)+(C_X-C_V)\mathbb{E}Prec(V_M, V_T)}\right] \tag{3.2}$$

where C_X is the true match/non-match class balance of all pairs in X, i.e. $C_X = 2|\{\{(x_i, x_j) : x_i, x_j \in X, y_i = y_j, i < j\}\}|/(n(n-1))$.

Intuitively, the bound is composed of two parts. $|X_M|/|X_{\hat{Y}}|$ is the fraction of record pairs in $\mathcal{A}(X)$ that directly match. We can make strong guarantees about these pairs using the measured performance of the validation set. $Prec(V_M, V_T)$ is the precision of these direct matches, adjusted for the change in class balance.

Proof. From Lemma 1 and applying the definitions of pairwise precision for $X_{\hat{Y}}$ and V_M:

$$\begin{aligned}
\mathbb{E}\left[Prec(X_{\hat{Y}}, X_Y)\right] &= \mathbb{E}\left[\frac{|X_{\hat{Y}} \cap X_Y|}{|X_{\hat{Y}}|}\right], \\
&\geq \mathbb{E}\left[\frac{|X_M \cap X_Y|}{|X_{\hat{Y}}|}\right], \\
&= \frac{|X_M|}{|X_{\hat{Y}}|}\mathbb{E}\left[Prec(X_M, X_Y)\right],
\end{aligned}$$

$$\geq \frac{|X_M|}{|X_{\hat{Y}}|}\mathbb{E}\left[\frac{C_X(1-C_V)Prec(V_M, V_T)}{C_V(1-C_X)+(C_X-C_V)\mathbb{E}Prec(V_M, V_T)}\right],$$

where the last step follows from equating the match function validation set performance to the expected match function test set performance using change in match/nonmatch class balance. □

All the necessary quantities to compute the bound are easy to measure from the predicted clustering. $|X_{\hat{Y}}|$ is the number of pairs in the clustering output. $|X_M|$ is the number of records that directly match, which by Lemma 1 can be efficiently computed as $\sum_{(x_i,x_j)\in X_{\hat{Y}}} m(x_i, x_j)$.

The class balance of the validation set C_V is known, but we must estimate C_X. We refer the reader to state-of-the-art results for class prior estimation [28].

Theorem 5. *The pairwise recall of a record linkage result can be lower bounded by:*

$$\mathbb{E}\left[Recall(X_{\hat{Y}}, X_Y)\right] \geq \mathbb{E}\left[Recall(V_M, V_T)\right]. \tag{3.3}$$

In other words, the recall on the validation set already forms a lower bound for the pairwise recall on the test resolution.

Proof. From the definitions of pairwise recall for X_M and $X_{\hat{Y}}$ and then applying Lemma 1:

$$\begin{aligned}
\mathbb{E}\left[Recall(X_{\hat{Y}}, X_Y)\right] &= \mathbb{E}\left[\frac{|X_{\hat{Y}} \cap X_Y|}{|X_Y|}\right], \\
&\geq \mathbb{E}\left[\frac{|X_M \cap X_Y|}{|X_Y|}\right], \\
&= \mathbb{E}\left[Recall(X_M, X_Y)\right], \\
&= \mathbb{E}\left[Recall(V_M, V_T)\right],
\end{aligned}$$

where the last step does not require class rebalancing because recall is not a function of class balance (unlike precision, recall is function of only the positive pairs). □

A lower bound on pairwise F_1 (the harmonic mean of pairwise precision and recall) can be computed with the two former lower bounds. We will focus more on measuring both pairwise precision and recall as they are more informative than the aggregated F_1 metric.

Theorems 1 and 2 demonstrate interesting differences between pairwise precision and recall. For instance, pairwise precision is more susceptible to changes in dataset size, whereas the estimated lower bound for pairwise recall is consistent across all test sets. If anything, the true recall will improve on larger datasets due to increasing feature space density. Indeed, optimizing a record linkage algorithm for recall on a validation set is statistically well motivated, but the same is not true when considering precision.

These implications can partly be explained by the representativity property. As the size of the test dataset increases, both the feature space density and number of false positives increase. Especially with noisy data, as in the case of the counter-human-trafficking domain, this increases the probability of entities ‘snowballing’ together, a phenomenon we have seen in practice. Multiple options exist to combat this problem, including improving the match function and running at a more conservative threshold.

3.3.1 A Note on Blocking

We do not explicitly consider the use of blocking, which improves scalability of record linkage algorithms [71]. Though blocking will violate Lemma 1, if we redefine T_M and V_M as:

T_M^* Set of record pairs in T_M also in the blocking scheme

V_M^* Set of record pairs in V_M also in the blocking scheme

then Lemma 1, Theorem 1, and Theorem 2 again hold. Notice blocking may increase or decrease pairwise precision, but it can only hurt pairwise recall (because $|V_M^*| \leq |V_M|$).

3.4 An Optimal Algorithm

We wish to design an algorithm $\mathcal{A}^*$ that is optimal in terms of the performance lower bounds. Algorithm $\mathcal{A}^*$ will be a conservative strategy, which provides several advantages over a less theoretically driven approach. Most importantly, we can provide the best possible performance guarantees without knowing labels Y. In practice, this means deploying a record linkage algorithm which will provide an acceptable level of performance with high probability.

We assume we are given a trained match function m, and condition the optimality of $\mathcal{A}^*$ on m. In practice, m should be a match function that performs well on the validation set V. The only restrictions we make on match function m are Assumptions A2 and A3, which are both extremely mild. Even if m does not inherently possess these properties, it is trivial to satisfy Assumption A2 by checking both directions

$m(x_i, x_j) \vee m(x_j, x_i)$ and Assumption A3 by forcing $m(x, x) = \texttt{True}\ \forall x \in \mathcal{X}$. Thus, the guarantees for our algorithm $\mathcal{A}^*$ hold for essentially all match functions.

Our approach to maximizing the performance lower bounds is to make just enough matches to satisfy Assumptions A4 and A5 while avoiding extraneous matches that decrease the performance bounds. We propose a specific merge function and 'wrapper' for the match function m that achieve optimality.

Consider a new type of record $z \in \mathcal{Z}$ that is defined as a subset of the original records X (e.g. $z_1 = \{x_2, x_9, x_{11}\}$). Our new algorithm $\mathcal{A}^*$ operates on the space of records $\mathcal{Z}$.

Theorem 6. *For any match function m, the pairwise precision and recall estimated lower bounds are optimal for the merge function:*

$$\langle z_1, z_2 \rangle = \bigcup_{x_i \in z_1, z_2} x_i \tag{3.4}$$

and the match function 'wrapper' for the new record types z_1 and z_2:

$$m^*(z_1, z_2) = \max_{x_i \in z_1, x_j \in z_2} m(x_i, x_j). \tag{3.5}$$

Proof. For our bounds to hold, we must first show $\mathcal{A}^*$ satisfies Assumptions A2-A5. Then by recognizing that the recall lower bound in Theorem 5 only depends on the match function performance on the validation set and $Recall(V_{M^*}, V_T) = Recall(V_M, V_T)$, we can show the recall lower bound is already optimal. Thus, we are primarily concerned with showing $\mathcal{A}^*$ achieves the optimal precision lower bound.

$\mathcal{A}^*$ satisfies Assumptions A2-A5: By the definition of the set union operator, the merge function satisfies Assumptions A2-A4. We assumed the match function m satisfied Assumptions A2-A3, and A4 does not apply to the match function. Then Assumption A5 holds by the definition of the max function.

To show $\mathcal{A}^*$ maximizes the precision lower bound in Theorem 4, notice that the only parameter of the precision lower bound that depends on $\mathcal{A}^*$ is $1/|X_{\hat{Y}}|$. Consider comparing $\mathcal{A}^*$ against any other algorithm $\tilde{A}(X) = \tilde{Y}$ satisfying Assumptions A2-A5. In other words, could $\mathcal{A}^*$ have made fewer matches? Formally, we want to show $|\tilde{X}_{\tilde{Y}}| \geq |X^*_{\hat{Y}}|$ for all $\tilde{X}$. Assume on the contrary there exist two records z_1 and z_2, such that $m^*(z_1, z_2) = \texttt{False}$ but one pair of their constituent records match, i.e.

$m(x_i, x_j) = \texttt{True}$, for some $x_i \in z_1$, $x_j \in z_2$. By definition, this contradicts the representativity property.

□

The simplicity of this approach is due to only assuming knowledge about direct pairwise matches, learned from the labeled pairs in the validation set. Interestingly, this record linkage system is equivalent to finding all connected components in an undirected graph with adjacency matrix $Adj_{ij} = m(x_i, x_j)$. We stress that though this may optimize the estimated lower bound performances, it does not necessarily guarantee better performance. However, if ground truth is not available for a dataset of comparable size to the deployed system, then this is now a theoretically well motivated approach.

A significant benefit of Theorem 6 is the provided match function need not satisfy the restrictive representativity property (Assumption A5). Further, since Assumptions A2 and A3 are trivial to satisfy (A4 only applies to the merge function), m can essentially be any match function. For example, one could use more complex machine learning based match functions (e.g. kernelized SVM, random forests) and featurizations which may not have intuitive merge operations (e.g. word2vec [63], Brown clustering [15]). Using less restrictive match functions undoubtedly enables better $Prec(V_M, V_T)$ and $Recall(V_M, V_T)$, further improving the lower bounds.

3.5 Experiments

We conducted experiments on multiple datasets with known ground truth to empirically demonstrate the tightness of the estimated lower bounds. We also show the true performance and estimated lower bound curves have similar shapes over parameters of $\mathcal{A}$. Thus, choosing model parameters using the lower bounds is a good approximation to choosing model parameters using the true (unknown) performance.

3.5.1 Datasets

We used one synthetic and three real world record linkage datasets with known ground truth for our experiments, as described in Table 3.1. Not surprisingly, human labeled datasets rarely number beyond several thousand records [62] – a relatively easy record

Table 3.1: Datasets used in the experiments

Dataset	# dim	# records	# matches
Synthetic	100	100000	4500
Restaurant[1]	4	864	112
Abt-Buy[2]	3	2173	1118
Escort (subset)	20	10000	10596

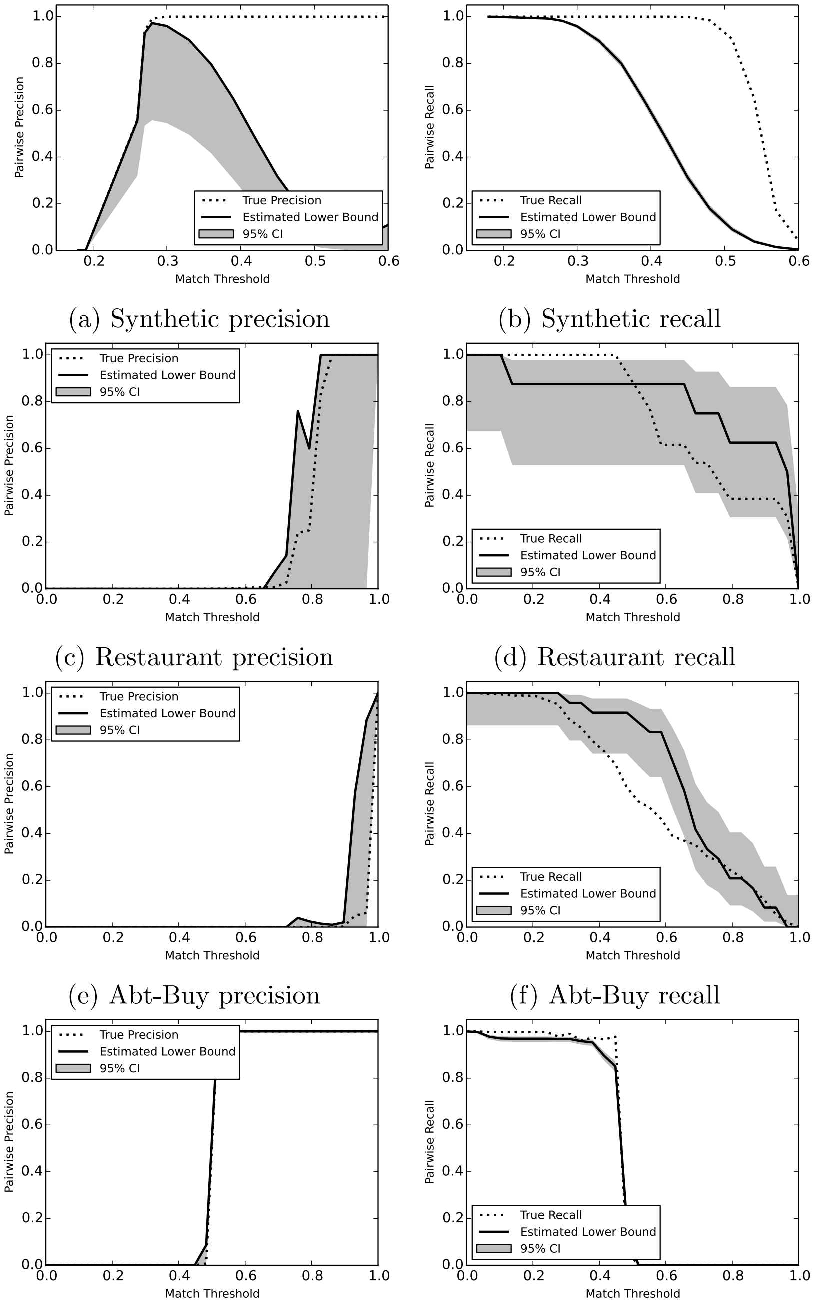
[1] cs.utexas.edu/users/ml/riddle/data/restaurant.tar.gz
[2] http://dbs.uni-leipzig.de/file/Abt-Buy.zip

linkage problem. The lack of large, publicly available general record linkage datasets with ground truth is an unfortunate obstacle to advancing the field. The authors were unsuccessful in obtaining a large, open-source record linkage dataset, so the only large-scale results in this chapter are on synthetic and proprietary datasets.

For the synthetic dataset, we created approximately 10,000 latent entities with 100 unique strings as features. Then we created records from the latent entities by replicating their features, such that the true cluster sizes were drawn i.i.d. from $\mathcal{N}(100, 25)$. In total, this resulted in exactly 100,000 records. At this point, there is no feature overlap between latent entities because records describing the same latent entity are identical. Then we randomly corrupted features according to a Bernoulli probability of 0.3, and replaced these features with the corresponding feature from a common 'corrupt' record. A corruption probability of 0.0 is a trivial problem and a probability of 1.0 is an impossible problem because all the 1 million records would be identical.

The restaurant dataset is one of the earliest record linkage tasks discussed in literature [96], and still used today [45, 101]. It consists of conflicting restaurant information from Zagat and Fodor's, including name, phone number, street address, city and cuisine. Unfortunately, the dataset is also relatively small – numbering only 864 records. We threw away the phone number feature because it made the problem too simple. The Abt-Buy dataset is more recent, larger at 2173 records, and used extensively in current research [46, 101]. It consists of product information from two retailers, including product name, description, and price.

Both the Restaurant and Abt-Buy datasets are in the class of 'clean-clean' record linkage problems, where we know a priori that matches only occur between disjoint



(a) Synthetic precision (b) Synthetic recall

(c) Restaurant precision (d) Restaurant recall

(e) Abt-Buy precision (f) Abt-Buy recall

(g) Escort precision (h) Escort recall

Figure 3.1: Experimental results demonstrate the estimated lower bound is tight to the true performance. Pairwise F_1 is not shown because it is the harmonic mean of the two former metrics, and is thus less informative. See in-line comments regarding exceptions to the 95% confidence interval.

databases, and no matches occur within databases. Note this additional information makes the problem strictly easier [71]. To formulate these datasets in a more general context, we merged them together into a single dataset and ignored the advantageous contextual 'clean-clean' information.

Lastly, we evaluated the bounds on a subset of an escort advertisements dataset scraped from several websites over the past few years [35]. We used natural-language-processing algorithms to extract 20 features, such as name, age, location, and hair color of the person being advertised. For ground truth, we used a subset of the data containing phone number matches as a proxy cluster label.

3.5.2 Algorithms

We used two different record linkage algorithms, depending on the dataset. For the larger synthetic experiment, we used the lower bound optimal algorithm in Theorem 6 with an approximated Jaccard similarity coefficient match function. For two records x_1 and x_2 with sets of string n-gram features f_1 and f_2, respectively, the Jaccard similarity coefficient is defined as $J(f_1, f_2) = |f_1 \cap f_2|/|f_1 \cup f_2|$. Then the match function $m(x_1, x_2) = \mathbb{1}(J(f_1, f_2) \geq t)$, where $\mathbb{1}$ is the indicator function and t is some cut-off threshold. Computing the full $\mathcal{O}(n^2)$ record comparisons is prohibitively expensive with 100,000 records, so we used the locality sensitive hashing technique MinHash, an unbiased approximation to the Jaccard coefficient [14]. Finding all matching record pairs with a Jaccard similar coefficient above the cut-off threshold can be found with high probability without making the full $\mathcal{O}(n^2)$ comparisons by indexing on the hash function outputs. We refer the interested reader to the seminal works on locality sensitive hashing by Broder and Charikar [14, 18].

For the smaller experiments, we used the general record linkage framework R-Swoosh by Benjelloun et al. [11]. For the match function, we trained a binary logistic regression classifier using known matches and nonmatches in the training dataset. Like all pairwise record linkage algorithms, it operates on pairwise features, which we computed from two records' features using either a binary match (e.g. state, hair color), numerical difference (e.g. ages, weights), or Levenshtein string edit distance (e.g. name) of each feature pair. If a record had multiple of a particular feature from a merge operation, we used the closest feature match. For the merge function, we simply used the set union of the respective features.

A significant and subtle caveat of using logistic regression is the need for no negative evidence (i.e. the representativity property). This restricts each logistic weight to either the positive or negative domain, depending if a larger or smaller pairwise feature is indicative of a match, respectively (a convex inequality constraint).

For both record linkage algorithms, the parameter choices are reduced to a single value: the cut-off threshold. The choice of cut-off threshold is a classic trade-off between precision and recall – an ideal setting to examine the results of our bounds.

Lastly, estimating the class balance ratio C_X is outside the scope of this chapter. State-of-the-art results for this task have been achieved using direct density ratio estimation [28]. For this purposes of verifying the bound, we used a gold standard of this ratio.

3.5.3 Results

To examine the tightness of the estimated lower bound, which may be used to optimize a record linkage system, we evaluated the true and lower bound performances across finely spaced intervals of match cut-off thresholds, as shown in Figure 3.1. The tightness of the bounds demonstrate two important qualities. First, they enable using the lower bounds as an approximation of the true performance when choosing model parameters (e.g. cut-off threshold). Though this may not necessarily result in the true (unknown) optimal parameters, it will result in the best estimated lower bound. Second, it enables enforcing a level of acceptable quality for any record linkage results.

The 95% confidence intervals are obtained via the propagation of validation set Wilson scores for $Prec(V_M, V_S)$ and $Recall(V_M, V_S)$ [104]. In the combined 240 trials across the two metrics, four datasets, and 30 match thresholds, this occurred on precisely 10 occasions (4.2% of the trials). This falls well within our statistical confidence bounds, especially because experiments across match threshold are correlated, effectively decreasing the number of independent trials.

The confidence interval widens as the gap between validation set and test set sizes widen. For very small datasets such as Restaurant, we were restricted to using minimal validation samples due to the small number of labels. However, for larger experiments such as Abt-Buy and Escort, we could afford hundreds or thousands of validation samples, significantly reducing uncertainty. This is also theoretically motivated by the shift in class balance in Theorem 1.

The four experiments demonstrate different record linkage behavior. The synthetic experiment has a narrow range of model parameters with perfect precision and recall, where performance degrades dramatically outside this range. The Restaurant experiment has a more gradual tradeoff between precision and recall, though there is a significant uncertainty in the lower bound estimate due to the limited number of validation samples. Precision in Abt-Buy quickly degrades, though recall is much more gradual. Our bounds correctly capture the need to improve the underlying record linkage systems for the Abt-Buy and Escort datasets. Without this lower bound, the poor performance on larger datasets would not be evident from smaller validation experiments.

3.6 Conclusions

In this chapter, we proved the first known performance bounds for a wide class of match-and-merge record linkage algorithms. The bounds are simple yet effective and feasible to compute in practice. We experimentally demonstrated the bounds are tight to the true performance and can be used to optimize parameter choices.

Further, we showed the optimal lower bound strategy for any match function is the connected components problem from graph theory – a relatively conservative clustering approach compared to many record linkage systems. We understand that this does not necessarily guarantee better performance, but it does provide a better lower-bound guarantee. However, when labeled datasets of comparable size to the deployed system are not available, this is now a theoretically well motivated approach.

Our bounds specifically addressed performance of pairwise record linkage algorithms satisfying the ICAR properties in Assumptions A2 - A5 [11]. Pairwise algorithms are intuitive, easy to implement, and thus not surprisingly, popular. However, they are also only a subset of record linkage approaches [13, 32, 84, 101]. Further, we only considered pairwise precision, recall and F_1 metrics due to their popularity, intuitive interpretation and mathematical convenience, though other existing metrics have been shown to produce conflicting rankings [62].

Estimating the lower bounds relies on accurate estimations of several other quantities, including recall and precision on the validation set and class prevalence estimation in the test set. Especially as datasets scale to much larger sizes, our bounds rely on these estimates. Our theory and experiments show we are able to say more regarding performance guarantees as the gap between validation and testing set sizes narrows.

Chapter 4

A Bayesian perspective on record linkage

In this chapter, we consider a specific clustering task known as record linkage (i.e. entity resolution, deduplication). Record linkage usually involves identifying records (i.e. samples) containing numerical, categorical and/or string data from one or multiple databases which describe the same latent entity. The distinction between the terms record linkage and clustering is often ambigious in the literature, so we default the term more common in the respective area of research.

Traditional linkage methods that directly link records to one another become computationally infeasible as the number of records grows [20, 105], and thus, it is increasingly common for researchers to treat linkage as a clustering task, in which latent entities are associated with one or more noisy database records, and the inferential goal is to identify the latent entity underlying each observed database record [86, 87, 88]. Although there are many probabilistic, generative models for clustering — of which several have been used for record linkage — the theoretical properties, such as performance bounds, have such not been critically assessed.

The work of [86, 87, 88] attempted to deconstruct distorted data by latent variable mixture models. The authors achieved this by clustering similar records to a hypothesized latent entity for each observed record, where their *linkage structure* kept track of which latent entity belongs to the same observed records. This is modeled through a latent variable mixture model with a distortion process on the data (sections 4.2.1 and 4.2.3). Thus, the main goal is to be able to take distorted data and uncover the underlying structure in the presence of noise. This is similar to signal processing, where a signal is received in the presence of some noise and often the goal is to understand if the underlying true (latent) signal can be recovered. We develop performance bounds under the framework proposed by [86, 87, 88].

We provide an upper bound on the Kullback-Leibler (KL) divergence between models with different linkage structures and use it to provide a lower bound on the minimum probability of associating a record (i.e. sample) with an incorrect latent entity (i.e. cluster). More precisely, under the categorical model of [87, 88] and string model of [86], we find the minimum probability of getting a latent entity incorrect. Finally, we explore how our bounds perform in practice and describe their user practicality.

4.1 Prior work

Bayesian methods and latent variable modeling have become recently popularized in record linkage models. A major advantage of Bayesian methods is their natural handling of uncertainty quantification for the resulting estimates. The first notion of understanding a distortion process for record linkage is the hit-miss-model, which uses a binary distortion process on the data [23]. Within the Bayesian paradigm, most work has focused on specialized approaches related to linking two files [36, 95]. These contributions, while valuable, do not easily generalize to more than two files or to de-duplication within a single file. For a review of recent development in Bayesian methods, see [53].

The work of [87, 88] recently introduced a Bayesian model that simultaneously handled record linkage and de-duplication for categorical data. Their approach allowed for natural uncertainty quantification during analysis and post-processing. Finally, [77] recently extended the work of [88] to both categorical and string valued data using a coreference matrix or a partitioning approach. In the later paper, it was shown that the coreference matrix is a special case of the linkage structure, thus, we work with the linkage structure. Another advantage of [88] and similar approaches is that their linkage structure is amenable to an efficient MCMC inference algorithm. These models have become practically relevant as they have been shown to perform well on a variety of applications, including official statistics and medical data.

Given the noted distortion process, deriving performance bounds seems natural to recover the underlying structure. For example, much work has been done in information theory for subset selection in graphical model selection, signal de-noising, compressive sensing, and others. In compressed sensing, one question recently addressed in [27], was directly measuring the part of the data from sounds and images that *will not* be thrown away. We make a connection here, as in record linkage we wish to take noisy, distorted data and recover this under the KL divergence. Divergence functions by [47, 81] are useful in many applications including recent statistical applications of clustering, as done in [7] for hard clustering to obtain optimal quantization by minimizing the Bregman divergence (motivated by rate distortion theory).

The rest of this chapter proceeds as follows. Two recent record linkage models are given in Section 4.2; Section 4.2.1 and Section 4.2.3 review these models. Section 4.3 derives the respective performance bounds, and Section 4.4 shows performance of the bounds in practice, discusses our findings and user practicality. Section Section 4.5 discusses future directions along this line of Bayesian clustering and record linkage.

4.2 Bayesian Models

We assume two Bayesian record linkage models, one dealing with categorical data and the other dealing with both categorical and noisy string data, such as names, addresses, etc. The first is that of [87, 88], and the second is that of [86].

4.2.1 Categorical Model

We review common notation to both models.[1] Let $X = (x_1, \ldots, x_n)$ represent the data indexed by i. Each record corresponds to one of N latent entities, indexed by j. Assume $N = n_i$ without loss of generality. Each record or latent entity has values on p fields, indexed by ℓ, and are assumed be categorical and the same across all records and entities [87, 88]. M_ℓ denotes the number of possible categorical values for the ℓth field.

[1] For a toy example of the record linkage process, see the Supplementary Material 4.2.2.

In both models, $x_{i\ell}$ denotes the observed value of the ℓth field for the ith record, and $y_{j\ell}$ denotes the true value of the ℓth field for the jth latent entity. Then Λ_i denotes the latent entity to which the jth record in the ith list corresponds, i.e., $x_{i\ell}$ and $y_{j\ell}$ represent the same entity if and only if $\Lambda_i = j$. Then $\boldsymbol{\Lambda}$ denotes the Λ_i collectively. Distortion is denoted by $z_{i\ell} = I(x_{i\ell} \neq y_{\Lambda_i \ell})$, where $I(\cdot)$ denotes the indicator function. As usual, I represents the indicator function (e.g., $I(x_{i\ell} = m)$ is 1 when the ℓth field in record i has the value m), and let δ_a denote the distribution of a point mass at a (e.g., $\delta_{y_{\Lambda_i \ell}}$). The model of [87, 88] is:

$$\begin{aligned}
x_{i\ell} \mid \Lambda_i, y_{\Lambda_i \ell}, z_{i\ell}, \boldsymbol{\theta}_\ell &\overset{\text{ind}}{\sim} \begin{cases} \delta_{y_{\Lambda_i \ell}} & \text{if } z_{i\ell} = 0 \\ \text{MN}(1, \boldsymbol{\theta}_\ell) & \text{if } z_{i\ell} = 1 \end{cases} \\
z_{i\ell} &\overset{\text{ind}}{\sim} \text{Bernoulli}(\beta_\ell) \\
y_{j\ell} \mid \boldsymbol{\theta}_\ell &\overset{\text{ind}}{\sim} \text{MN}(1, \boldsymbol{\theta}_\ell) \\
\boldsymbol{\theta}_\ell \overset{\text{ind}}{\sim} \text{Dirichlet}(\boldsymbol{\mu}_\ell) \;\; \text{and} \;\; \beta_\ell &\overset{\text{ind}}{\sim} \text{Beta}(a_\ell, b_\ell) \\
\Lambda_i &\overset{\text{ind}}{\sim} \text{Uniform}\,(1, \ldots, N)\,, \qquad (4.1)
\end{aligned}$$

where MN denotes the Multinomial distribution and $a_\ell, b_\ell, \boldsymbol{\mu}_\ell$ are all known. Guidance for the hyper-parameters and a justification of the (discrete) uniform prior are given in [86, 87, 88]. Eq. (4.1) assumes that different records are independent conditional on the deeper variables of the model. Moreover, it assumes the same conditional independence of different fields for the same record.

4.2.2 Example of the Linkage Process

We provide a toy illustration of the general record linkage process in figure 4.1. Consider three databases D_1, D_2, D_3 and the notation already introduced, where here $k = 3$. Suppose the "population" entities have four members, where name and address are stripped for anonymity and they are listed by state, age, and sex, as is often the case with de-identified data.

For instance, assume the true latent entity vector $\boldsymbol{y}$ is *known*:

$$\boldsymbol{y} = \begin{bmatrix} \text{NC, 72, F} \\ \text{SC, 73, F} \\ \text{PA, 91, M} \\ \text{VA, 94, M} \end{bmatrix}.$$

The observed records X are given in three separate databases (k=3), which would combine into a three-dimensional array. We write this here as three two-dimensional arrays for notational simplicity:

$$D_1 = \begin{bmatrix} \text{NC, 72, F} \\ \text{SC, 70, F} \\ \text{PA, 91, M} \end{bmatrix}, D_2 = \begin{bmatrix} \text{SC, 37 , F} \\ \text{VA, 93, M} \\ \text{PA, 92, M} \end{bmatrix},$$

$$D_3 = \begin{bmatrix} \text{NC, 72 , F} \\ \text{NC, 72, F} \\ \text{SC, 72, F} \\ \text{VA, 94, M} \end{bmatrix}.$$

Here, for the sake of keeping the illustration simple, only age is distorted. Comparing X to $\boldsymbol{y}$, the intended linkage and distortions are

$$\Lambda = \begin{bmatrix} 1 & 2 & 3 & \\ 2 & 4 & 3 & \\ 1 & 1 & 2 & 4 \end{bmatrix},$$

$$\boldsymbol{z_1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{z_2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \boldsymbol{z_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In this linkage structure, every entry of Λ with a value of 2 means that some record from X refers to the latent entity with attributes “SC, 73, F.” Here, the age of this entity is distorted in all three databases, as can be seen from $\boldsymbol{z}$. (Note that $\boldsymbol{z}$, like X, is also really a three-dimensional array.) Looking at $\boldsymbol{z}_1$ and $\boldsymbol{z}_3$, we see that there is only a single record in either list that is distorted, and it is only distorted in one field. In list 2, however, every record is distorted, though only in one field.

Figure 4.1 illustrates the interpretation of the linkage structure as a bipartite graph in which each edge links a record to a latent entity. For clarity, figure 4.1 shows that X_{11} and X_{22} are the same entity and shows that X_{13}, X_{21}, and X_{34} correspond to the same entity. The rest are non-matches (or singleton entities).

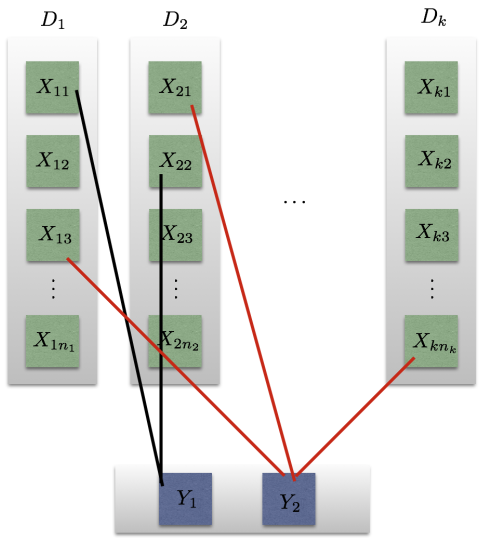


Figure 4.1: A general illustration of the record linkage process. We assume databases $D_1, \ldots D_k$. We assume records X that we cluster to latent entities Y. Records that belong to the same same latent entity are kept track of using the data structure or linkage structure Λ.

4.2.3 Empirical Bayesian Model

The work of [86] assumes fields $1, \ldots, p_s$ are string-valued, while fields $p_s+1, \ldots, p_s+p_c$ are categorical, where $p_s + p_c = p$ is the total number of fields. They assume an empirical Bayesian distribution on the latent parameter. For each $\ell \in \{1, \ldots, p_s + p_c\}$, let S_ℓ denote the set of *all* values for the ℓth field that occur anywhere in the data, i.e., $S_\ell = \{x_{i\ell} : 1 \leq i \leq k, 1 \leq j \leq n_i\}$, and let $\alpha_\ell(w)$ equal the empirical frequency of value w in field ℓ. Let G_ℓ denote the empirical distribution of the data in the ℓth field from all records in all databases combined. So, if a random variable W has distribution G_ℓ, then for every $w \in S_\ell$, $P(W = w) = \alpha_\ell(w)$. Hence, let G_ℓ be the prior for each latent entity $y_{j\ell}$. The distortion process changes such that

$$P(x_{i\ell} = w \mid \Lambda_i, y_{\Lambda_i \ell}, z_{i\ell}) = \frac{\alpha_\ell(w)\, \exp[-c\, d(w, y_{\Lambda_i \ell})]}{\sum_{w \in S_\ell} \alpha_\ell(w)\, \exp[-c\, d(w, y_{\Lambda_i \ell})]},$$

where $c > 0$ is a fixed normalizing constant corresponding to an arbitary distance metric $d(\cdot, \cdot)$. Denote this distribution by $F_\ell(y_{\Lambda_i \ell})$. The model becomes

$$\begin{aligned}
x_{i\ell} \mid \Lambda_i,\, y_{\Lambda_i \ell},\, z_{i\ell} &\overset{\text{ind}}{\sim} \begin{cases} \delta(y_{\Lambda_i \ell}) & \text{if } z_{i\ell} = 0 \\ F_\ell(y_{\Lambda_i \ell}) & \text{if } z_{i\ell} = 1, \ell \le p_s \\ G_\ell & \text{if } z_{i\ell} = 1, \ell > p_s \end{cases} \\
y_{j\ell} &\overset{\text{ind}}{\sim} G_\ell \\
z_{i\ell} \mid \beta_{i\ell} &\overset{\text{ind}}{\sim} \text{Bernoulli}(\beta_{i\ell}) \\
\beta_{i\ell} &\overset{\text{ind}}{\sim} \text{Beta}(a, b) \\
\Lambda_i &\overset{\text{ind}}{\sim} \text{Uniform}\,(1, \ldots, N)\,,
\end{aligned} \tag{4.2}$$

where all distributions are also independent of each other; assume that a, b, N are assumed known. This framework was shown to work well in applications and simulation studies, however, it was quite sensitive to the choice of the hyperparameters. This method beat supervised methods, such as random forests when the amount of training data input into the supervised methods was $< 10\%$.

Figure 4.2 contains a graphical representation of models 4.1-4.2.

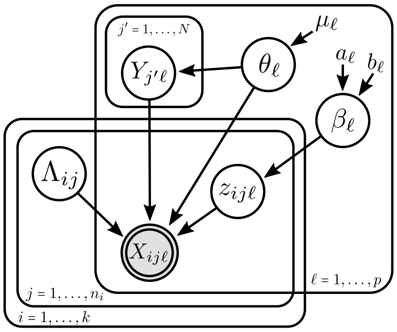


Figure 4.2: Graphical representation of models 4.1-4.2.

4.3 Performance Bounds

Recall the connection to KL divergence in the sense that for any two distributions P and Q, the maximum power for testing P versus Q is $\exp\{-nD_{\text{KL}}(P||Q)\}$. Hence, a low value of D_{KL} means that we need many samples to distinguish P from Q. A natural question is how does changing Y (latent entity) or Λ (linkage structure) change the distribution of X (observed records)? We search for both meaningful upper and lower bounds, since an upper bound will say that P and Q are never more than so far apart, whereas a lower bound says how easy it is to tell P and Q apart. Moreover, we investigate how well can we recover Y (latent entity) and Λ (linkage structure) from X (data).

Assuming the conditions of [86, 87], let $\mathcal{P} = \{f(X \mid Y, \Lambda_i, \boldsymbol{\theta}, \boldsymbol{\beta}) : \forall \Lambda_i \in \{1, \ldots, N\}.\}$ We know that $X_1, X_2, \ldots, X_N$ are all independent given $(Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta})$ under both $P, Q \in \mathcal{P}$. This implies that $D_{X_1,X_2,\ldots,X_N}(P\|Q) = \sum_i D_{X_i}(P\|Q)$. We first provide a theorem under the model of [87], which assumes categorical data and a hierarchical model. In Theorem 7, we find the minimum probability of getting a latent entity wrong. Moreover, we are able to say that with growing distortion of the data, there is no difference between two latent entities and the bound becomes infinite and non-informative in this case. Next, under the model of [86] we provide a general theorem, which assumes both categorical and noisy text data. This theorem provides an upper bound on the KL divergence of arbitrary distributions P and Q.

4.3.1 Kullback-Leibler Divergence under Categorical Data

We use Fano's inequality [73] to bound the probability of misclassification, as a function of the KL divergence between P and Q, as defined in the previous section. Assume that Λ and $\hat{\Lambda}$ are two distinct linkage structures that correspond to the same latent entity ($\boldsymbol{y}$). Let $r+1$ be the cardinality of $\mathcal{P}$, i.e. $r+1 = N$.

Theorem 7. *This result finds an upper bound on the KL divergence and a lower bound for the probability that model 4.1 gets the linkage structure incorrect. Let* $\gamma = \max_{\Lambda_i \neq \Lambda'_{ij}} 2 \sum_{ij\ell} I(y_{\Lambda_i \ell} \neq y_{\Lambda'_{ij}\ell})(1-\beta_\ell) \ln \left\{ \dfrac{1}{\min_m \theta_{\ell m} \beta_\ell} \right\}.$

i) The KL divergence is bounded above by γ. That is, $D_X(P||Q) \leq \gamma \;\; \forall P, Q \in \mathcal{P}$.

ii) The minimum probability of getting a latent entity wrong is $Pr(\Lambda_{ij} \neq \Lambda'_i) \geq 1 - \dfrac{\gamma + \ln 2}{\ln r}, \;\; \forall i, j$

That is, as the latent entities become more distinct, γ increases. On the other hand, as the latent entities become more similar, $\gamma \to 0$.

Remark. *Consider Theorem 7 (i). Suppose* $\beta_\ell \to 1$*. Then* $D_X \geq 0$*. If instead* $\beta_\ell \to 0$*, then* $D_X \geq 1$*. The lower bound is only informative when* $\beta_\ell \to 0$*. We have more information when the latent entities are separated.*

Proof. To show this, we simply apply Pinsker's inequality, where for all $P, Q \in \mathcal{P}$:

$$\begin{aligned} D(P\|Q) &\geq 2||P - Q||_1^2 \implies \\ D(P\|Q) &\geq I(y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell})(1 - \beta_\ell)^2 \implies \\ D_X(P\|Q) &\geq \sum_{ij\ell} I(y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell})(1 - \beta_\ell)^2. \end{aligned} \tag{4.3}$$

□

Proof. We assume the model of [87, 88], which assumes that data is categorical. We assume model 4.1 holds in section 4.2.1. We first prove (i). Consider $f(X \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta})$. Then

$$Pr(x_{i\ell} = m \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}) = 1(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell.$$

It follows from equation 4.4 that

$$D_{x_{i\ell}}(P\|Q) = \sum_{m=1}^{M_\ell} I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell\} \times \log \left[\frac{I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell}{I(y_{\Lambda'_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell} \right].$$

It directly follows that

$$D_X(P\|Q) = \sum_{ij\ell m} \{I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell\} \; \log \left[\frac{I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell}{I(y_{\Lambda'_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m}\beta_\ell} \right] \Bigg\}.$$

If $y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell}$, then

$$\begin{aligned} \|P - Q\|_1 &= \sum_m \left| I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m} \beta_\ell \; - I(y_{\Lambda'_i \ell} = m)(1 - \beta_\ell) - \theta_{\ell m} \beta_\ell \right| \quad (4.4) \\ &= 2(1 - \beta_\ell). \quad (4.5) \end{aligned}$$

Eq. (4.5) holds since $P(m) = Q(m)$ unless $m = y_{\Lambda_i \ell}$ or $m = y_{\Lambda'_i \ell}$. If $y_{\Lambda_i \ell} = y_{\Lambda'_i \ell}$, then $P = Q$ and $\|P - Q\|_1 = 0$. The reverse Pinsker inequality of [12] relates the KL divergence to the L_1 norm in the following way: $D(P\|Q) \leq \|P - Q\|_1 \ln\{(\min Q)^{-1}\}$. Using this, we find that (if $y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell}$), then

$$\begin{aligned} D(P\|Q) &\leq 2(1 - \beta_\ell) \ln \left\{ \frac{1}{\min_m I(y_{\Lambda'_i \ell} = m)(1 - \beta_\ell) + \theta_{\ell m} \beta_\ell} \right\} \\ &\leq 2(1 - \beta_\ell) \ln \left\{ \frac{1}{\min_m \theta_{\ell m} \beta_\ell} \right\}. \end{aligned}$$

Hence,

$$\begin{aligned} \max_{P,Q \in \mathcal{P}} D_X(P\|Q) \leq \max_{\Lambda_i \neq \Lambda'_i} 2 \sum_{ij\ell} I(y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell})(1 - \beta_\ell) \ln \left\{ \frac{1}{\min_m \theta_{\ell m} \beta_\ell} \right\} \\ := \gamma. \end{aligned}$$

This proves (i). We now prove (ii). Using Fano's inequality [73], the minimum probability of getting a latent entity wrong is $Pr(\Lambda_i \neq \Lambda'_i) \geq 1 - \frac{\gamma + \ln 2}{\ln r}$, where $r + 1$ is the cardinality of $\mathcal{P}$, i.e. $r + 1 = N$. As the latent entities become more distinct, γ increases. On the other hand, as the latent entities become more similar, $\gamma \to 0$. □

4.3.2 KL Divergence Bounds for String and Categorical Data

We now consider P and Q under [86] for both categorical and noisy string data. Recall that β_ℓ tunes the amount of distortion as defined in Eq. (4.2). Recall that $d(\cdot, \cdot)$ denotes any arbitrary distance metric between an observed string and a latent string as seen in Eq. (4.2), and $c > 0$ is a fixed normalizing constant corresponding to the distance metric d.

In Theorem 8, for any distinct linkage structures, the minimum probability of getting a latent entity wrong is governed by a lower bound, which is growing at a rate $c \to \infty$ that is determined by the moment generating function of the distances between an observed string in data and a latent string.

Theorem 8. *Assume data X, and distributions $P, Q \in \mathcal{P}$ defined in section 4.3. Assume two distinct linkage structures, denoted by $y_{\Lambda_i \ell}, y_{\Lambda'_i \ell}$.*

i) There is an upper bound on the KL divergence between any $P, Q \in \mathcal{P}$ given by κ, that is $D_X(P||Q) \leq \kappa$.

ii) $Pr(\Lambda_i \neq \Lambda'_i) \geq 1 - \dfrac{\kappa + \ln 2}{\ln r}$, where

$$\kappa = \max_{\Lambda_i \neq \Lambda'_i} \Bigg[2 \sum_{\ell} (1 - \beta_\ell) I(y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell}) + \sum_{\ell m} I(y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell}) \left(1 - e^{-cd(y_{\Lambda_i \ell}, y_{\Lambda'_i \ell})}\right) \\ \times E[e^{-cd(m, y_{\Lambda_i \ell})}] \Bigg] \ln\{(\min Q)^{-1}\}$$

and $r + 1$ is the cardinality of $\mathcal{P}$.

Proof. We first prove (i). Consider

$$\begin{aligned} Pr(x_{i\ell} = m \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}) &= Pr(x_{i\ell} = m \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}, z_{i\ell} = 1) \times Pr(z_{i\ell} = 1 \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}) \\ &\quad + Pr(x_{i\ell} = m \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}, z_{i\ell} = 0) \times Pr(z_{i\ell} = 0 \mid Y, \Lambda, \boldsymbol{\theta}, \boldsymbol{\beta}) \\ &\propto I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \alpha_\ell(x_{i\ell}) \beta_\ell \times \Big[\exp\{-c\, d(x_{i\ell}, y_{\Lambda_i \ell})\} \Big]. \end{aligned} \tag{4.6}$$

Suppose that $y_{\Lambda_i \ell} \neq y_{\Lambda'_i \ell}$. Equation 4.6 implies that

$$D_{x_{i\ell}}(P \| Q) \propto \sum_{m=1}^{M_\ell} I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \alpha_\ell(m) \beta_\ell \times \left[e^{-c\, d(x_{i\ell}, y_{\Lambda_i \ell})} \times \phi \right], \tag{4.7}$$

where

$$\phi = \log \left[\frac{I(y_{\Lambda_i \ell} = m)(1 - \beta_\ell) + \alpha_\ell(m) \beta_\ell \left[e^{-c\, d(m, y_{\Lambda_i \ell})} \right]}{I(y_{\Lambda'_i \ell} = m)(1 - \beta_\ell) + \alpha_\ell(m) \beta_\ell \left[e^{-c\, d(m, Y'_{\Lambda'_i \ell})} \right]} \right].$$

We now consider $\|P-Q\|_1$ and by equation 4.7, we find

$$\begin{aligned}\|P-Q\|_1 = \sum_{m\in M_\ell} \Big| & I(y_{\Lambda_i\ell}=m)(1-\beta_\ell) + \alpha_\ell(m)\beta_\ell \exp\{-c\, d(m, y_{\Lambda_i\ell})\} \\ & -I(y_{\Lambda_i'\ell}=m)(1-\beta_\ell) - \alpha_\ell(m)\beta_\ell \exp\{-c\, d(m, y_{\Lambda_i'\ell})\}\Big| . \end{aligned} \tag{4.8}$$

Then by equation 4.8, it is clear that

$$\begin{aligned}\|P-Q\|_1 &\le \sum_m (1-\beta_\ell)\left|\left[I(y_{\Lambda_i\ell}=m) - I(y_{\Lambda_i'\ell}=m)\right]\right| \\ &\quad + \sum_m \alpha_\ell(m)\beta_\ell \times \left|\exp\{-c\, d(m, y_{\Lambda_i\ell}) - \exp\{-c\, d(m, y_{\Lambda_i'\ell})\}\right| \\ &\le 2(1-\beta_\ell) + \beta_\ell \sum_m \alpha_\ell(m)\left|\exp\{-c\, d(m, y_{\Lambda_i\ell}) \ - \exp\{-c\, d(m, y_{\Lambda_i'\ell})\}\right| .\end{aligned}$$

Now assume that two field attributes are different. That is, suppose there exists an $m \neq m'$. Then we assume that there exists a $\delta > 0$ such that $d(m,m') \geq \delta$. By the reverse triangle inequality, for any m, m', m'',

$$|d(m,m') - d(m,m'')| \le d(m',m'') \implies e^{-c[d(m,m')-d(m,m'')]} \ge e^{-cd(m',m'')}. \tag{4.9}$$

Equation 4.9 in turn implies that

$$\begin{aligned}&\sum_m \left[\left(1-e^{-c[d(m,m')-d(m,m'')]}\right) e^{-cd(m',m'')}\alpha_\ell(m)\right] \\ &\ge \sum_m \left(1-e^{-c[d(m',m'')]}\right) e^{-cd(m',m'')}\alpha_\ell(m).\end{aligned}$$

Then $\sum_m \alpha_\ell(m)\left[e^{-cd(m,m')} - e^{-cd(m,m'')}\right] = \sum_m \alpha_\ell(m) e^{-cd(m,m')}\left(1-e^{-cd(m',m'')}\right) = \left(1-e^{-cd(m',m'')}\right)\sum_m \alpha_\ell(m) e^{-cd(m,m')} = \left(1-e^{-cd(m',m'')}\right) E[e^{-cd(m,m')}]$ where $M \sim \alpha_\ell$.

That is, $\sum_m \alpha_\ell(m) e^{-cd(m,m')}$ is the moment generating function of $d(M,m')$ (evaluated at c), where $M \sim \alpha_\ell$. This implies that $\|P-Q\|_1 \le 2(1-\beta_\ell) + \beta_\ell \sum_m \left(1-e^{-cd(y_{\Lambda_i\ell}, y_{\Lambda_i'\ell})}\right) E[e^{-cd(m,y_{\Lambda_i\ell})}]$. Then by reverse Pinker's inequality [12], we can write

$$\begin{aligned}\max_{P,Q\in\mathcal{P}} D_X(P||Q) \leq \max_{\Lambda_i\neq\Lambda_i'} &\Bigg[2\sum_{ij\ell}(1-\beta_\ell)I(y_{\Lambda_i\ell}\neq y'_{\Lambda_i\ell}) \\ &+\sum_{ij\ell m} I(y_{\Lambda_i\ell}\neq y'_{\Lambda_i\ell})\left(1-e^{-cd(y_{\Lambda_i\ell},y_{\Lambda_i'\ell})}\right) \\ &\times\left[E[e^{-cd(m,y_{\Lambda_i\ell})}]\right]\times\ln\{(\min Q)^{-1}\}\Bigg] \\ =:\ &\kappa,\end{aligned}$$

where $Q = I(y_{\Lambda_i'\ell} = m)(1-\beta_\ell) - \alpha_\ell(m)\beta_\ell\exp\{-c\, d(m, y_{\Lambda_i'\ell})\}$. Thus, (i) is established. Using Fano's inequality, we find that $Pr(\hat{\Lambda}_i \neq \Lambda_i) \geq 1 - \frac{\kappa+\ln 2}{\ln r}$.

We have established that for any $y_{\Lambda_i\ell} \neq y_{\Lambda_i'\ell}$, the minimum probability of getting a latent entity wrong is governed by the constant c. That is, the lower bound grows as c goes to ∞, and its rate of growth is determined by the moment generating function of the distances. We have now established (ii). □

4.4 Simulation Study and Discussion

We consider how the bounds in Sections 4.3.1 and 4.3.2 hold for two simulated experiments. In our experiments Experiment I and Experiment II, synthetic categorical data are generated according to either model 4.1 or 4.2 using the parameters shown in Table 4.1 and 4.2, respectively. In order to consider a realistic set of strings for S, we consider the set of 20 most popular female baby names from 2014, according to the United States Census. Then for the distance d, we consider the generalized Levenshtein edit distance.

We then generate both categorical and string records according to either model 4.1 or 4.2. For each experiment, we vary exactly one of the parameters to demonstrate its impact of the linkage error rate $Pr((\hat{\Lambda}_{ij}, Y) \neq (\Lambda_i, Y))$. We choose the other values such that the performance is neither extremely low nor extremely high. We set the distortion parameter β_ℓ to the same value for each ℓ, i.e. $\beta_\ell = 0.6$ denotes a distortion probability of 0.6 for every field. $\beta_\ell = 0.0$ to 1.0 means we started with $\beta_\ell = 0$ for all ℓ and swept the values until $\beta_\ell = 1$ for all ℓ. Recall p is the number of fields, and

Experiment	N	β_ℓ	$p = p_c$	$\theta_{\ell m}$
Fig. 1(a)	10 to 500	0.6	3	0.1
Fig. 1(b)	100	0 to 1	3	0.1
Fig. 1(c)	100	0.6	1 to 8	0.25
Fig. 1(d)	100	0.8	5	$\frac{1}{46}$ to 1

Table 4.1: Categorical Experiments

Experiment	N	β_ℓ	$p = p_s$	c
Fig. 2(a)	100 to 500	0.6	1	1.0
Fig. 2(b)	100	0.2 to 1	1	1.0
Fig. 2(c)	100	0.6	1 to 10	1.0
Fig. 2(d)	100	0.6	1	0 to 2

Table 4.2: String Experiments

thus the maximum value of ℓ. We also set each $\theta_{\ell m}$ to the same value, i.e. $\theta_{\ell m} = 0.1$ denotes $\theta_{\ell m} = 0.1$ for all ℓ and all m. This further implies each field ℓ takes on exactly $M_\ell = 1/\theta_{\ell m}$ values in order for θ_ℓ to be a valid probability distribution.

We compare the bound in Theorem 7 to two record linkage algorithms [86, 87, 88]. The first is an exact sampler, which samples directly from $Pr(\Lambda_i|x_i, Y, \boldsymbol{z})$. The second is a more realistic Gibbs sampler with empirically motivated priors proposed by [86]. We run the Gibbs sampler for 10,000 iterations on all experiments to ensure proper mixing. There is some difficulty in comparing Λ to $\hat{\Lambda}$, as there are multiple equally correct modes due to arbitrary re-orderings of the latent individuals $\hat{\boldsymbol{Y}}$ and corresponding linkage structure $\hat{\Lambda}$. Even though the Gibbs sampler may infer the correct latent individuals Y and linkage structure, because the ordering is arbitrary, it is unlikely that $\Lambda = \hat{\Lambda}$. To avoid such an issue of label switching, we fix $\hat{\boldsymbol{Y}}$ during the sampling process.

Specifically, we compare the bound to the empirical error rate of the Gibbs sampler proposed by [86]. In order to compute the empirical probability $Pr(\hat{\Lambda_{ij}} \neq \Lambda_i)$, we hold Y fixed during Gibbs sampling to ensure errors in $\hat{\Lambda}$ are not due to arbitrary changes in the ordering of the labels of Y. In addition, we compare the linkage error rate to an exact sampler, which samples directly from $Pr(\Lambda|X, Y, \boldsymbol{z})$.

Results of Experiment I In Figures 4.3 (a)-(d) we vary the number of records N, distortion parameter β, number of fields p and number of values each field takes M_ℓ, respectively. The empirical results demonstrate Theorem 7 captures the dependence between the error rate and the all relevant latent parameters θ, N and β. Specifically, linking records becomes more difficult as N increases, the distortion parameter β increases, the number of fields p decreases or the number of values each field can take M_ℓ decreases. The bound nicely captures the logarithmic increase in error with respect to N in Figure 4.3 (a), which gives hope for linking records in very large databases. Other terms appear to be $\bar{O}(n)$ when not near extreme error values, implying low noise and a larger feature space are essential to performing high quality record linkage.

Results of Experiment II Figures 4.4 (a)-(d) show Theorem 8 is tight to the true performances on string data when varying N, β, number of string fields p_s and c, respectively. As expected, and similarly to the categorical results, linking records becomes more difficult as N increases, the distortion parameter β increases and the parameter c decreases. The effects of parameter variation is less noticeable in the string experiments due to the fact that linking string fields is easier than ones that have been anonymized, i.e., categorical fields.

The Gibbs sampler (blue diamonds) performs almost as well as the exact sampler (grey circles). In fact, due to the conditional entropy version of Fano's inequality and the fact that $H(X|Y) \leq H(X)$, any Gibbs sampler cannot perform better in expectation than an exact sampler. Thus, we believe the gap between the bound (gold squares) and the exact sampler does not necessarily indicate the existence of a better algorithm, but perhaps only some unnecessary slack due to the application of Pinsker's and then reverse Pinsker's inequalities.

4.4.1 Discussion of Results

As illustrated in Theorems 7 and 8 we have derived an upper bound on the KL divergence as well as lower bounds for misclassifying a latent entity. In Theorem 7 (i), we showed that the latent entities become more distinct when γ is increasing. This is in contrast to when γ gets closer to 0, since then the latent entities become more

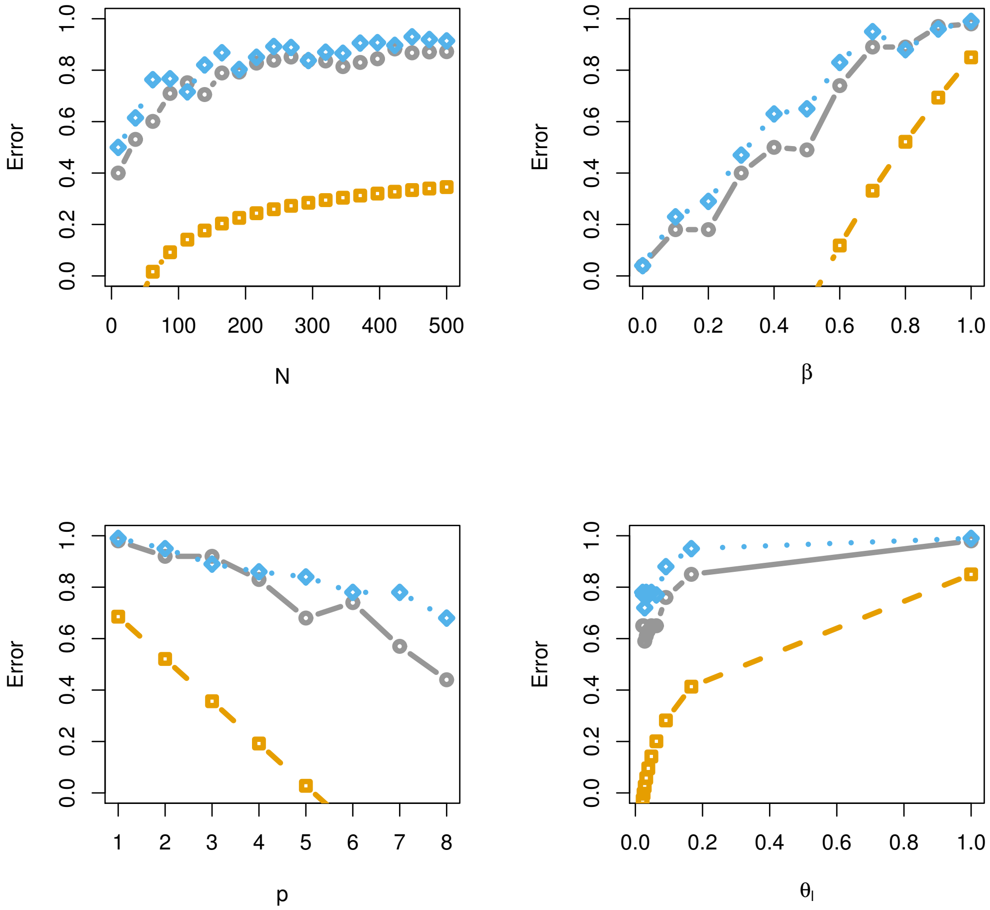


Figure 4.3: Theorem 7 (gold squares) holds on simulated categorical records compared to exact sampling (grey circles) and Gibbs sampler (blue diamonds).

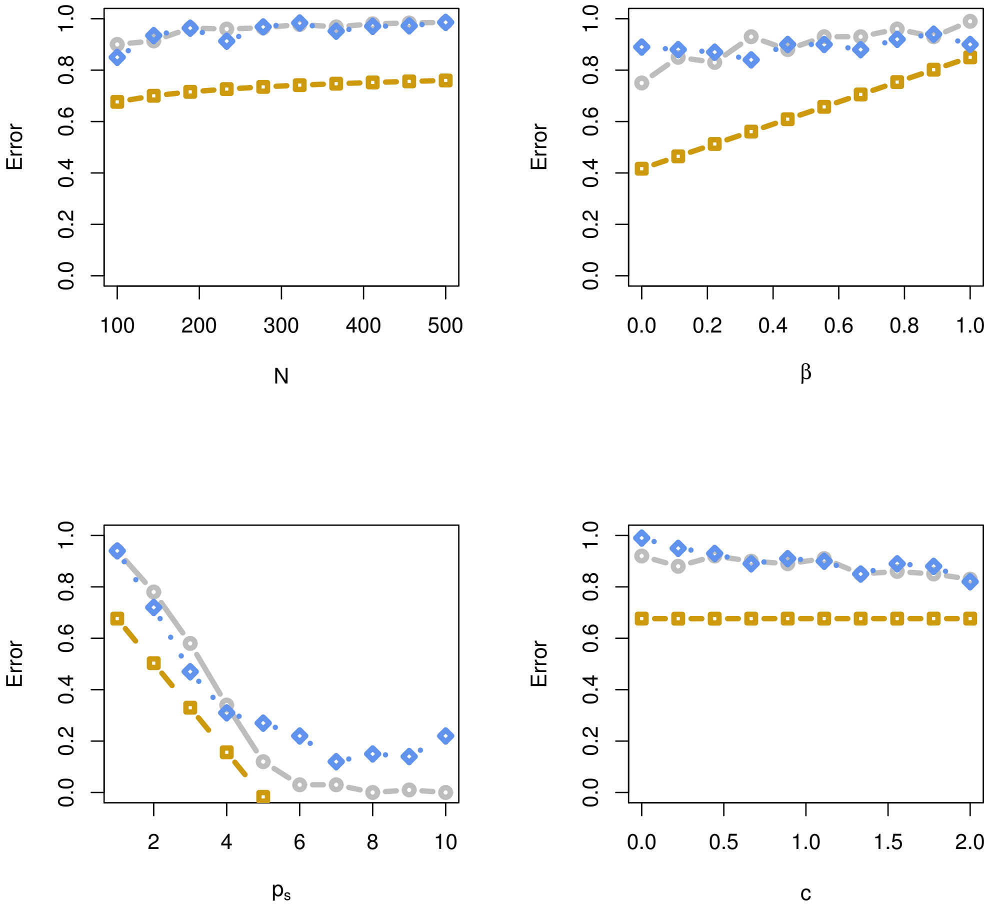


Figure 4.4: Theorem 8 (gold squares) holds on simulated noisy string records compared to exact sampling (grey circles) and Gibbs sampler (blue diamonds).

similar. In Theorem 7 (ii), we showed that as the distortion parameter $\beta_\ell \to 1$, then the upper bound γ is infinite. In practice, as illustrated in [87], the latent entities are difficult to distinguish when the amount of distortion is more than 5% at every field value. Thus, this corresponds to when the bound is too loose. On the other hand, as $\beta_\ell \to 0$, the latent entities become more separated.

We discuss how separated the latent entities are under choices of β_ℓ, θ_ℓ and N, providing guidance to the user in this setting given our simulation results. As practical guidance when the distortion is between 0 to 5% at every feature value, the latents will be more separated and the bound will be be loose. On the other hand, as β_ℓ increases, the bound becomes tighter. The choice of β_ℓ can be made using subjective information about the underlying data and tuned using the hyper-parameters a, b. (See [86, 87] for choosing such values). On the other hand, we can see that for more realistic values of the distortion parameter in Figure 4.3 (a), (b), and (d) , the bound is quite loose when the distortion parameter β_ℓ is large. Thus, a loose bound here is warranted due to the amount of noise or model-misspecification being placed into the model as well as the fact that all of the fields being used are categorical. Such results match the intuition given in [87].

In Theorem 8 (ii), we derived a lower bound where the minimum probability of getting a latent entity wrong is controlled by c, which is determined by the moment generating function of the distances between an observed string and a latent string. This bound has the same type of form as the bound in Theorem 7, however, since we now have string-valued data, we see that the minimum probability of getting a latent entity wrong is dominated by the string-valued variables and specifically, the distances functions used and the constants used. In comparison to [86], this completely matches up with the sensitivity that was seen to the choice of the distance functions as well as the choice of c as this will completely dominate the posterior, and hence, the ability to tell latent entities apart under this posterior.

In practice, the driving force of the tightness of the bound is c, the steepness parameter of the string distribution in equation 2. As c increases, it is less likely for a string-valued record's features to be distorted to values that are far from that of their latent feature values. This is verified in Figure 4.4(c), where linkage error decreases as c increases. The work of [86] gave practical choices for c, which were [0,2]. Similarly, we can speak to the tightness of d, which relies on the distortion parameter β_ℓ not being too small in practice, as verified in Figure 4.4(b). In terms of the bounds found in Theorem 1 and 2, the empirical Gibbs sampler has tight bounds in almost all situations, except when the number of features is large, N is too small, or β_ℓ is too small (and similarly for θ_ℓ). This coincides with exactly what we would expect in practice from the real experiments of [86].

For all applications in both categorical and string data, we expect the bounds to be as loose in practice (corresponding to easier record linkage), when the distortion parameter is small $(0 - 4)$ and when the the number of fields is large $(p \geq 5)$ or the number of values that each field can take, M_ℓ, increases (this will be application specific). Finally, the bounds should be tighter, corresponding to more difficult record linkage, as the total number of records N increases (see Figure 4.3). These parameter values match almost exactly with two real data experiments (corresponding ranges of parameters) as well as a simulation study from [87, 88].

4.5 Conclusions

First, we have derived general performance bounds for record linkage, making connections to KP models and other related Bayesian models. More specifically, we have drawn connections to a wide class of models from Bayesian record linkage. Second, our bound for the categorical Bayesian record linkage model is easily interpretable and matches the intuition of the generative model. Third, our bound for the categorical and noisy string model, takes a similar form to that of the categorical model. We are also able to interpret this bound in a way that aligns with the interpretations [86, 87, 88] as well as show the practicality of our bounds to the aforementioned papers. More specifically, our bounds are empirically loose for categorical data, which is not unexpected since there is little information available to match on. This contrasts the empirical tight bounds for both categorical and noisy string data. As illustrated in our experiments, with just one string variable, our bounds become much tighter, and as the number of strings increases, the bound becomes more tighter when compared to exact and Gibbs sampling.

In addition, there has been early work in Bayesian nonparametrics to push forward record linkage. The work of [66] pointed out that most clustering tasks assume cluster sizes grow linearly with the size of the database. Such examples include infinitely exchangeable clustering models, including finite mixture models, Dirichlet process mixture models, and Pitman–Yor process mixture models, which all make this linear growth assumption. However, in record linkage such an assumption is undesirable since linkage methods require models that yield clusters whose sizes grow sublinearly with the total number of data points. This observation led the authors to define the microclustering property as well as a new model exhibiting such growth. Our work has been able to provide bounds for the aforementioned work since the prior consider is a KP model. In future work it would be helpful to try and draw connections between those proposed in [66] and [86, 87, 88] in order to generalize such bounds and provide tighter bounds using conditional entropy or other sophisticated bounding methods.

The contents of this chapter were presented at the 2017 International Conference on Artificial Intelligence and Statistics (AISTATS) [92].

Part II

Learning on clusters

Chapter 5

Challenge of Dependency Leakage

Machine learning systems increasingly depend on pipelines of multiple algorithms to provide high quality and well structured predictions. This chapter argues interaction effects between machine learning algorithms can cause subtle adverse behaviors that may not be initially apparent. In particular, we focus on the broad class of prediction and clustering problems, where clustering algorithm errors impact the predictor's performance on these clusters. No previous work has been able to characterize the conditions under which these effects occur, and if they do, what properties they have. We precisely answer these questions by providing theoretical properties which hold in various settings, and prove that expected behavior rapidly decays with even minor interaction effects. Fortunately, we are able to leverage these same properties to construct hypothesis tests and scalable estimators necessary for correcting the problem. Empirical results on benchmark datasets validate our characterizations.

5.1 Introduction

With the increasing prevalence of machine learning solutions, there is a growing concern over the interactions between algorithms in complex systems [80]. Leveraging multiple learning algorithms is a common technique to optimize performance and incorporate structured prior knowledge. For example, most autonomous vehicles benefit from using separate models for perception of traffic lights, object detection and tracking, localization, predicting actor behavior and ultimately planning an optimal trajectory. Although attempting to directly map from visual inputs to control outputs is simpler, this approach is known to achieve inferior performance. Breaking the larger problem into a sequence of smaller problems may be advantageous for many reasons, but we argue it can create additional challenges which must be addressed. At the most basic level, the errors or modifications in one component can cause dangerous and unintended behavior in other components or in the overall system.

In this chapter, we address the broad class of interaction effects between clustering and prediction algorithms. In the self-driving vehicle example, this encompasses a significant portion of the autonomy stack, including clustering tasks (e.g. pixel and LIDAR point segmentation) and prediction tasks (e.g. object type, current and future states). We observe this is often also a concern in domains including online shopping, medical systems and census statistics, which are further explored in the experimental section.

To elucidate the potential behavior induced by interaction effects, consider the problem of predicting heart disease from a collection of medical records. Each patient may have several records due to multiple hospital visits but it is unlikely we are able to collect multiple records for every patient. Thus, we must find a learner which generalizes well to new patients not in our training set. The typical approach is to match records belonging to the same individual using some record linkage (i.e. clustering) algorithm. Then the records are split by patient into a training and validation sets, such that all records for a single patient end up in either the training or validation set. This provides an unbiased estimate of the learner's error on new patients, i.e. the out-of-cluster (OOC) loss.

The underlying challenge in this example is that we do not have access to the oracle clustering (i.e. the mapping from medical records to patients), but only a noisy approximation of it from the record linkage algorithm. Even in relatively low-noise domains like medical and census, these algorithms are known to be imperfect [91, 107, 108]. If we instead take the approach of splitting the dataset according to the *approximated* patient clustering, this effectively causes samples to spill across the true training and validation folds. Some samples which should have been grouped with a validation patient may have ended up with a training patient, and vice versa, without our knowledge. In other words, the training and validation sets are no longer conditionally independent, leading to a problem called *dependency leakage* [9]. This allows the learner to overfit to patient-specific features and optimistically biases our OOC loss estimate. For example, if a patient's records are incorrectly clustered and samples are partitioned into both the training and validation sets, the learner is rewarded for predicting whether a patient has heart disease based on their name – which clearly will not generalize to new patients. This overfitting need not be so blatant. The learner may overfit to subtle patterns in a chest x-ray, a form of bias which may be hard to identify even by experienced radiologists.

This interaction between clustering errors and a prediction algorithm is particularly dangerous because our learner may appear to be doing well on the validation set, but does far worse when we deploy it in the real world on new patients. This is compounded by the fact that some application domains (e.g. medical, census) involve extreme consequences, including patient misdiagnosis and misguided public policy decisions. Note that this bias is undetectable during standard cross-validation procedures unless an explicit attempt is made to estimate and correct for it, which is the primary focus of this paper. Saeb et al. note that over half of selected medical studies failed to account for any clustering, allowing records for the same patient to occur in both the training and validation datasets, a significant statistical mistake [79].

The contributions and organization of the remainder of this chapter is as follows. We begin in Section 5.2 by formalizing the problem and notation. In Section 5.3, we present theoretical properties for interaction effects between clustering and prediction algorithms which hold under various conditions. In Section 5.4, we demonstrate how these properties can be used to construct a simple hypothesis test for the presence of bias in cross-validation results. Finally, we conducted empirical studies on Parkinson's, heart disease, 1994 US Census and Dota 2 video game data, and provide results in Section 5.5 which demonstrate the practical behavior of interaction effects closely aligns with our theoretical results.

5.2 Problem Statement

More formally, let $X = x_1, \ldots, x_{n_x}$ be the n_x observed samples, y be the corresponding labels, and $c : \{1, \ldots, n_x\} \to \{1, \ldots, k\}$ be the oracle clustering algorithm which partitions the data into k clusters (e.g. k is the number of patients, n_x is the number of medical records). Our high level goal is to train a prediction algorithm f which generalizes to new clusters, i.e. has low out-of-cluster loss. The the leave-one-cluster-out (LOCO) estimator

$$\widehat{\mathrm{Err}}_{\mathrm{LOCO}} = \frac{1}{|c_1^{-1}|} \sum_{j \in c_1^{-1}} \ell(y_j, f(x_j \mid x_{\bar{c}_1^{-1}}, y_{\bar{c}_1^{-1}})), \tag{5.1}$$

is an unbiased estimator of the OOC loss[1]. Here, $\mathcal{T} = (X_{\bar{c}_1^{-1}}, Y_{\bar{c}_1^{-1}})$ and $\mathcal{V} = (X_{c_1^{-1}}, Y_{c_1^{-1}})$ denote the training and validation sets, where c_i^{-1} and $\bar{c}_i^{-1}$ denote all sample indices belonging and not belonging to cluster i, respectively. Without loss of generality, we have arbitrarily chosen to leave the first cluster out.

The key question here is: how will errors in the clustering algorithm $\hat{c}$ effect our ability to train and validate the predictor f? By examining the LOCO estimator used to train and validate f, we see that errors in $\hat{c}$ result in noisy training and validation sets $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, where some samples have flipped between $\mathcal{T}$ and $\mathcal{V}$. For now, consider the unidirectional leakage scenario where samples move from $\mathcal{V}$ to $\mathcal{T}$ to create $\hat{\mathcal{V}}$ and $\hat{\mathcal{T}}$, such that $\hat{\mathcal{T}} \overset{n}{\sim} M_{P_{\mathcal{T}}, P_{\mathcal{V}}}(1 - p_0, p_0)$, where $M_{a,b}(w_a, w_b)$ denotes the mixture distribution of a and b with weights w_a and w_b and p_0 is the leakage probability (a function of $\hat{c}$'s error). If the clustering is perfect (i.e. $\hat{c} = c$), then $p_0 = 0$. Let e_i be the expected loss at some other $p = {}^{i}\!/_{n}$ fraction of corrupted samples (we use the notational shorthand $e(p)$ to denote e_{pn}). The expected OOC loss is equivalent to e_0 (i.e. zero dependency leakage, $p = 0$), but we only observe the empirical loss at some $p_0 > 0$. Thus, our specific goals are to characterize the behavior of the interaction effects e and to efficiently estimate e_0 in order to train and validate f.

5.3 Theoretical Properties

In this section, we present theoretical results on interaction effects between prediction and clustering algorithms. First, we prove that under mild conditions, the sequence of losses $e = e_0, e_1, \ldots, e_n$ is monotonically decreasing due to dependency leakage. Second, under slightly stronger conditions, the sequence will be convex with respect

[1] An unbiased estimate of training on $k - 1$ clusters. It is slightly biased compared to training on all k clusters.

to p. Intuitively, errors in the clustering algorithm allows the prediction algorithm to ‘peak’ at samples in the validation distribution, which will improve its performance with diminishing returns.

We say a learner f is optimal under its training distribution if

$$f(\cdot|\mathcal{T}) \in \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y\sim P_{\mathcal{T}}} \ell(f(x), y). \tag{5.2}$$

Generally speaking, this tends to be true for large $|\mathcal{T}|$, small model complexity of $\mathcal{F}$ or sufficient regularization in ℓ. This does not imply f is overfit to the training set, but in fact that it generalizes well across $P_{\mathcal{T}}$.

Theorem 9. *The sequence $e_0, e_1, \ldots, e_n$ is monotonically decreasing if f is optimal under its training distribution.*

Prior to introducing the full proof, we begin by introducing a key lemma about the minimization of function mixtures.

Lemma 10. *For functions $a, b : \Theta \to \mathbb{R}$ and $\alpha \in [0, 1]$,*

$$a(\operatorname*{argmin}_{\theta\in\Theta}(\alpha a(\theta) + (1-\alpha)b(\theta)))$$
$$b(\operatorname*{argmin}_{\theta\in\Theta}(\alpha a(\theta) + (1-\alpha)b(\theta)))$$

are monotonically decreasing and increasing, respectively, with respect to α.

Proof. Let $\Delta(\theta) = a(\theta) - b(\theta)$, $1 \geq j > i \geq 0$ and

$$\theta_i \in \operatorname*{argmin}_{\theta\in\Theta} b(\theta) + i\Delta(\theta)$$
$$\theta_j \in \operatorname*{argmin}_{\theta\in\Theta} b(\theta) + j\Delta(\theta)$$

Then a is monotonically decreasing with respect to α if and only if $a(\theta_i) \geq a(\theta_j)$.

Case 1: $\theta_i = \theta_j$. Then $a(\theta_i) = a(\theta_j)$, $b(\theta_i) = b(\theta_j)$ and the statements holds.

Case 2: $\theta_i \neq \theta_j$. Then both the following conditions must be true.

$$b(\theta_j) - b(\theta_i) + i\Delta(\theta_j) - i\Delta(\theta_i) > 0 \tag{5.3}$$
$$b(\theta_j) - b(\theta_i) + j\Delta(\theta_j) - j\Delta(\theta_i) < 0 \tag{5.4}$$

If Eq. (5.3) did not hold, then θ_j would have been optimal at $\alpha = i$, i.e. $\theta_j \in \operatorname{argmin}_{\theta\in\Theta} b(\theta) + i\Delta(\theta)$. Likewise, if Eq. (5.4) did not hold, then $\theta_i \in \operatorname{argmin}_{\theta\in\Theta} b(\theta) + j\Delta(\theta)$.

Together, they imply

$$\begin{aligned} b(\theta_j) - b(\theta_i) + i\Delta(\theta_j) - i\Delta(\theta_i) &> b(\theta_j) - b(\theta_i) + j\Delta(\theta_j) - j\Delta(\theta_i) \\ i\Delta(\theta_j) - i\Delta(\theta_i) &> j\Delta(\theta_j) - j\Delta(\theta_i) \\ (i-j)(\Delta(\theta_j) - \Delta(\theta_i)) &> 0 \\ \Delta(\theta_j) - \Delta(\theta_i) &< 0 \end{aligned}$$

since $i - j < 0$. Plugging this into Eq. (5.3),

$$\begin{aligned} b(\theta_j) - b(\theta_i) + i\Delta(\theta_j) - i\Delta(\theta_i) &> 0 \\ b(\theta_j) - b(\theta_i) &> i(\Delta(\theta_i) - \Delta(\theta_j)) \\ b(\theta_j) - b(\theta_i) &> 0 \qquad (5.5) \end{aligned}$$

which proves the second statement. Finally, plugging Eq. (5.5) into Eq. (5.4) concludes the proof.

$$\begin{aligned} (1-j)(b(\theta_j) - b(\theta_i)) + j(a(\theta_j) - a(\theta_i)) &< 0 \\ a(\theta_j) - a(\theta_i) &< 0 \end{aligned}$$

□

The proof of Theorem 9 follows.

Proof. **Direction $\mathcal{V}$ to $\mathcal{T}$** We say f is optimal under its training distribution if

$$f(\cdot|\mathcal{T}) \in \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y\sim P_\mathcal{T}} \ell(f(x), y).$$

Let $f_0, f_1, \ldots, f_n$ be models learned at each level of dependency leakage, such that each model is optimal under its training distribution, i.e.

$$f_i \in \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y\sim M_{P_\mathcal{T},P_\mathcal{V}}(1-\frac{i}{n},\frac{i}{n})} \ell(f(x), y).$$

The sequence $e_0, e_1, \dots, e_n$ is monotonically decreasing when

$$e_i - e_{i+1} \geq 0 \quad \forall i \in \{0, \dots, n-1\}.$$

Starting from the definition of e and using the notational shorthand $\ell_P(f) = \mathbb{E}_{x,y\sim P}\ell(f(x), y)$,

$$\begin{aligned} e_i &= \mathbb{E}_{x,y\sim P_\mathcal{V}}\ell(f_i(x), y) \\ &= \ell_{P_\mathcal{V}}(f_i) \\ &= \ell_{P_\mathcal{V}}(\operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y\sim M_{P_\mathcal{T},P_\mathcal{V}}(1-\frac{i}{n},\frac{i}{n})}\ell(f(x), y)) \\ &= \ell_{P_\mathcal{V}}\left(\operatorname*{argmin}_{f\in\mathcal{F}} \frac{i}{n}\ell_{P_\mathcal{V}}(f) + \left(1 - \frac{i}{n}\right)\ell_{P_\mathcal{T}}(f)\right) \end{aligned} \tag{5.6}$$

By Lemma 10, e is monotonically decreasing with respect to $\frac{i}{n}$, and thus also with respect to i since n is a fixed constant.

Direction $\mathcal{T}$ to $\mathcal{V}$. In this direction, e will further be linear:

$$\begin{aligned} e_0 &= \mathbb{E}_{x,y\sim P_\mathcal{V},\mathcal{T}\overset{n}{\sim}P_\mathcal{T}}\ell(f(x|\mathcal{T}), y) \\ e_n &= \mathbb{E}_{x,y\sim P_\mathcal{T},\mathcal{T}\overset{n}{\sim}P_\mathcal{T}}\ell(f(x|\mathcal{T}), y) \\ e_i &= \mathbb{E}_{x,y\sim M_{P_\mathcal{T},P_\mathcal{V}}(\frac{i}{n},1-\frac{i}{n}),\mathcal{T}\overset{n}{\sim}P_\mathcal{T}}\ell(f(x|\mathcal{T}), y) \\ &= \left(\frac{i}{n}\right)\mathbb{E}_{x,y\sim P_\mathcal{T},\mathcal{T}\overset{n}{\sim}P_\mathcal{T}}\ell(f(x|\mathcal{T}), y) + \left(1 - \frac{i}{n}\right)\mathbb{E}_{x,y\sim P_\mathcal{V},\mathcal{T}\overset{n}{\sim}P_\mathcal{T}}\ell(f(x|\mathcal{T}), y) \\ &= \left(\frac{i}{n}\right)e_n + \left(1 - \frac{i}{n}\right)e_0 \end{aligned}$$

and $e_n \leq e_0$ by the assumption that f is optimal under its training distribution. □

This theorem implies that the interaction will always *optimistically* bias our cross-validation results. This is in fact the most dangerous type of bias, as our heart disease classifier will perform well on the off-line hold-out set, but then perform worse when we deploy it in the real world on new patients or at new hospitals. If f is not optimal among $\mathcal{F}$, it is possible to construct counterexamples such that $e_0, \dots, e_n$ is not monotonically decreasing.

In our second theoretical result, we show that the expected loss is convex with respect to the strength of interaction effect p. Let $\ell_P(f) = \mathbb{E}_{x,y\sim P}\ell(f(x), y)$ be the expected loss of the learner f under distribution P. Then the following theorem holds.

Theorem 11. *The sequence $e_0, e_1, \ldots, e_n$ is convex if f is optimal under its training distribution and $\ell_{P_\mathcal{T}}$ and $\ell_{P_\mathcal{V}}$ are strictly convex and differentiable over f.*

Prior to discussing the proof, we begin by introducing a lemma on the minimization of mixtures of convex functions.

Lemma 12. *For $\alpha \in [0, 1]$, let $a, b : \Theta \to \mathbb{R}$ be strictly convex and differentiable (where $\dot{a}$ denotes $\frac{\partial a}{\partial \theta}$) over*

$$\begin{aligned}\Theta^* &= \{\theta \in \operatorname*{argmin}_{\theta\in\Theta}(\alpha b(\theta) + \alpha\Delta(\theta))\} \quad \forall\alpha \in [0, 1] \\ &= \{\theta \in g(\alpha)\} \quad \forall\alpha \in [0, 1] \subseteq \Theta.\end{aligned}$$

If $\frac{\dot{a}}{\dot{b}}$ is convex, decreasing over Θ^*, then

$$a(\operatorname*{argmin}_{\theta\in\Theta}(\alpha a(\theta) + (1-\alpha)b(\theta)))$$

is convex over α.

Proof. If $\frac{\dot{a}}{\dot{b}}$ is convex, decreasing then $\frac{-\dot{b}}{\dot{\Delta}}$ is also convex decreasing.

$$\frac{\dot{a}}{\dot{b}}\text{convex, decreasing} \Leftrightarrow \frac{-\dot{\Delta}}{\dot{b}}\text{concave, increasing} \tag{5.7}$$

because $\frac{-\dot{\Delta}}{\dot{b}} = \frac{\dot{b}-\dot{a}}{\dot{b}} = 1 - \frac{\dot{a}}{\dot{b}}$.

Further, we know $\frac{-\dot{\Delta}}{\dot{b}} \geq 0$ because $\dot{a} \leq 0$ and $\dot{b} \geq 0$ by Lemma 10. Then $\frac{-\dot{b}}{\dot{\Delta}}$ is convex decreasing by the composition of the convex, decreasing function $\frac{1}{x}$ and the concave increasing $\frac{-\dot{\Delta}}{\dot{b}}$. Note in the case where $\dot{\Delta} = 0$, $g(\alpha)$ is constant and the lemma holds.

At the minimum of $b(\theta) + \alpha\Delta(\theta)$,

$$\begin{aligned} 0 &= \dot{b} + \alpha\dot{\Delta} \\ \alpha &= \frac{-\dot{b}}{\dot{\Delta}} \end{aligned}$$

Thus, $g^{-1}(\theta) = \frac{-\dot{b}}{\dot{\Delta}}$ is convex, decreasing and $g(\alpha)$ is concave, increasing. Finally $a(g(\alpha))$ is convex, decreasing by the composition of a convex, non-increasing and concave function. □

The proof for Theorem 11 follows.

Proof. **Direction $\mathcal{T}$ to $\mathcal{V}$** Holds by Theorem 9, as linearity implies convexity.

Direction $\mathcal{V}$ to $\mathcal{T}$ Starting from the definition of e and using the notational shorthand $\ell_P(f) = \mathbb{E}_{x,y\sim P}\ell(f(x), y)$,

$$\begin{aligned} e_i &= \mathbb{E}_{x,y\sim P_{\mathcal{V}}}\ell(f_i(x), y) \\ &= \ell_{P_{\mathcal{V}}}(f_i) \\ &= \ell_{P_{\mathcal{V}}}(\operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y\sim M_{P_{\mathcal{T}},P_{\mathcal{V}}}(1-\frac{i}{n},\frac{i}{n})}\ell(f(x), y)) \\ &= \ell_{P_{\mathcal{V}}}\left(\operatorname*{argmin}_{f\in\mathcal{F}} \frac{i}{n}\ell_{P_{\mathcal{V}}}(f) + \left(1 - \frac{i}{n}\right)\ell_{P_{\mathcal{T}}}(f)\right) \end{aligned} \tag{5.8}$$

By Lemma 12, e is convex with respect to $\frac{i}{n}$, and thus also with respect to i since n is a fixed constant. □

Strictly convex and differentiable loss functions hold for a wide class of problems, including support vector machines and linear or ridge regression. The convexity of e compounds the monotonic behavior in Theorem 9, as it implies that even a small amount of error in our clustering $\hat{c}$ can cause large amounts of cross-validation bias in f. In Section 5.5, we empirically demonstrate both these properties hold on all examined datasets.

5.4 Hypothesis Testing

A principal question for data scientists is whether an interaction effect exists between their clustering and prediction algorithms. Here, we show how to use the theoretical properties from Section 5.3 to quickly construct a two-sample t-test for dependency leakage, which avoids the complexity of constructing an estimator for the OOC loss $\hat{e}_0$.

Consider the alternative hypothesis $H_a : e_0 > e(p_0)$, where $p_0 > 0$ is the unknown leakage probability and e_0 is the OOC loss with zero leakage (i.e. no interaction effect). By Theorem 9, we can use a one sided test because $e(p_0) \geq e(p_n)$. First, form $n_{\mathcal{T}}$ training folds each of size n' from $\hat{\mathcal{T}}$. Additionally, form $n'_{\mathcal{T}}$ training folds of size n' and $n_{\mathcal{T}} + n'_{\mathcal{T}}$ validation folds of size $n_{\mathcal{V}}$ from $\hat{\mathcal{V}}$.

Train and validate f on the disjoint $n_{\mathcal{T}} + n'_{\mathcal{T}}$ training folds and corresponding validation folds. Let $z = z_1, \ldots, z_{n_{\mathcal{T}}}$ and $z' = z'_1, \ldots, z'_{n'_{\mathcal{T}}}$ be the validation loss of f trained on the folds from $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, respectively. Let $\bar{z}$ and $\bar{z}'$ be the mean of these two sequences. Then

$$\bar{z} - \bar{z}' \sim N(e(p_0) - e(p_n), \sigma^2(\bar{z}) + \sigma^2(\bar{z}'))$$

and the two-sample t-test statistic is

$$T = \frac{\bar{z} - \bar{z}'}{\sqrt{\frac{s_1^2}{n_{\mathcal{T}}} + \frac{s_2^2}{n'_{\mathcal{T}}}}} \tag{5.9}$$

where s_1^2 and s_2^2 are the sample variances of z and z', respectively.

Rejecting the null hypothesis $H'_0 : e(p_0) \leq e(p_n)$ when $T > t_{1-\alpha,v}$ is a level α test, where $t_{1-\alpha,v}$ is the critical value of the t-distribution with v degrees of freedom. Further, by Theorem 9 and Theorem 11, $e(p_0) \neq e(p_n) \Rightarrow e_0 \neq e(p_0)$ so long as $p_0 > 0$. Thus, rejecting the null hypothesis $H_0 : e_0 \neq e(p_0)$ when $T > t_{1-\alpha,v}$ is also a level α test.

There are two takeaways to consider when using this test. The first powerful property is that it does not require actually knowing the clustering error or leakage probability p_0 a priori, only that it is not perfect (a very weak assumption). Second, the Type II error rate of this test largely depends on the convexity of e. If $p_0 < 0.5$

and e is linear, then $e(p_0) - e(p_n) > e_0 - e(p_0)$ and the Type II error rate will actually be *lower* than if we could directly test $e_0 \neq e(p_0)$. Conversely, the Type II error rate becomes larger as e becomes more strongly convex.

5.5 Empirical study

Finally, we conducted an empirical, finite-sample study which validates the theoretical properties in Section 5.3. In all experiments, we used either a linear SVM classifier or linear regression as the predictor f. This is a best-case scenario, as interaction effects depend on the predictor f's ability to overfit to mistakes from the clustering algorithm $\hat{c}$. Thus, as the complexity of the predictor class increases, the interaction effect worsens.

Note that in order to compute the true interaction effects, we are required to use a dataset where the oracle clustering is indeed available. For many of these experiments, we used data collected in very controlled settings to guarantee no clustering error in the ground truth. In more practical scenarios, this information would not be available.

Synthetic Experiment For the synthetic simulation study, we use a partition model with $k = 2$ parts and n sufficiently large such that duplicate resamples are improbable, a subsample of which is depicted in Fig. 6.1. For f, we use a linear regression model and set the loss ℓ as the mean squared error. To simulate the effects of noisy clusters $\hat{c}$, we move samples between the two parts $\mathcal{T}$ and $\mathcal{V}$ with uniform probability between $p = 0$ and $p = 1$.

1994 US Census Experiment In the second experiment, we use data from the 1994 US Census to validate our claim that conventional cross-validation introduce bias against sub-populations due to dependency leakage [52]. Here, we consider the task of predicting a person's income given their demographic, educational and occupational

[2] https://archive.ics.uci.edu/ml/datasets/heart+Disease

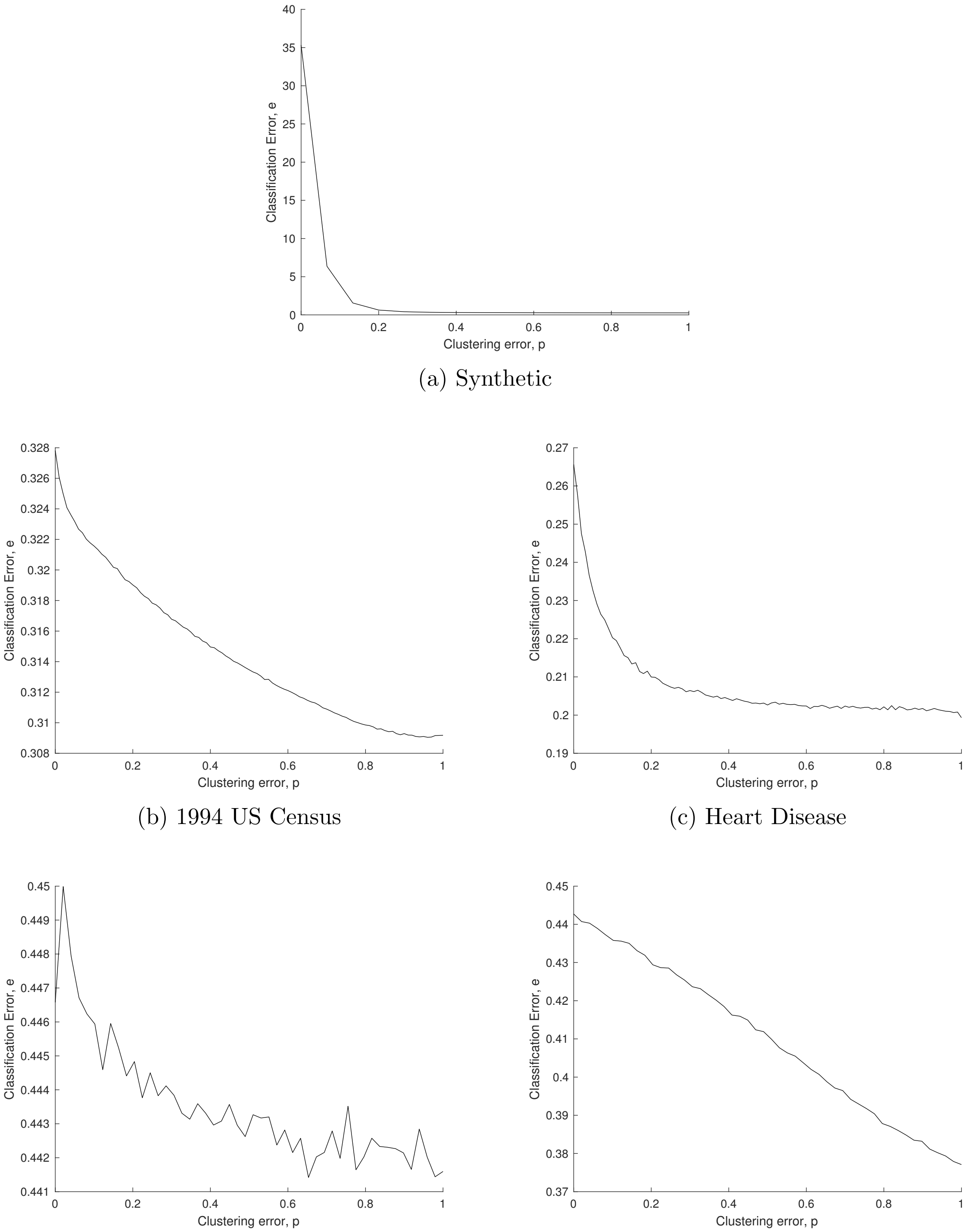
[3] https://archive.ics.uci.edu/ml/datasets/adult

[4] https://archive.ics.uci.edu/ml/datasets/Parkinson+Speech+Dataset+with++Multiple+Types+of+Sound+Recordings

[5] https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results

Table 5.1: Parameters used in all experiments. n is the number of samples in the training set, $|\mathcal{V}|$ is the number of samples in the validation set, and d is the number of features in the dataset.

	Parameter							
Dataset	n	$\|\mathcal{T}\|$	$\|\mathcal{V}\|$	d	Latent cluster	Training clusters	Validation clusters	Features
Synthetic	∞	15	1000	2	-	-	-	-
Heart[2]	100	100	100	12	Location	Cleveland, VA, Switzerland	Hungary	age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, thal
1994 US Census[3]	100	100	100	5	Native country	United States, El Salvador, Germany, Mexico, Philippines, Puerto Rico	India, Canada	age, education_num, hours_per_week, race, occupation
Parkinson[4]	100	100	100	26	Subject	$2, 3, 4, 6, 7, 8, \ldots$	$1, 5, 9, \ldots$	jitter_local, jitter_abs, jitter_rap, jitter_ppq5, jitter_ddp, shimmer_local, shimmer_db, shimmer_apq3, shimmer_apq5, shimmer_apq11, shimmer_dda, ac, nth, htn, median_pitch, mean_pitch, std_dev, min_pitch, max_pitch, pulses, periods, mean_period, std_dev_period, unvoiced, breaks, deg_breaks
Dota 2[5]	100	100	100	114	Type	1, 2	3	hero0, hero1, . . . , hero112



(a) Synthetic

(b) 1994 US Census

(c) Heart Disease

(d) Dota 2

(e) Parkinson's

Figure 5.1: Empirical results show the loss is indeed convex and monotonically decreasing, validating our theoretical results in Section 5.3.

.

information. Our training set consists of samples from certain origin countries and we wish to train a learner which performs well for people of all countries. In other words, we minimize the LOCO generalization loss, where clusters correspond to origin countries. For this experiment, we use 30368 persons from the United States, El Salvador, Germany, Mexico, Philippines and Puerto Rico for training set $\mathcal{T}$ and validate with $\mathcal{V}$ on 221 immigrants from India and Canada. For features, we consider their age, years of education, work hours per week, race, and occupation. We trained an SVM classifier to predict whether their yearly income is greater than US$50k per year (finer resolution income was unavailable due to privacy reasons).

The results demonstrate interaction effects causes the learner to be biased against Indian and Canadian immigrants, due to dependency leakage. In other words, the classifier is rewarded for learning attributes specific to the training countries, even though they do not generalize across all countries.

Heart Disease Experiment In the third experiment, we use heart disease data collected from Cleveland, USA; VA Long Beach, USA; Switzerland and Hungary [52]. The task is to predict whether a patient has heart disease, given their demographic information and vital signs. We need to train a classifier which performs well at new hospitals – given data from only these 4 locations. Thus, clusters correspond to hospital location and we use LOCO to estimate the generalization error. Training clusters correspond to 479 patients in Cleveland, Long Beach and Switzerland, testing clusters correspond to 262 patients in Hungary. All other experimental details are the same as Experiment II. The results are shown in Fig. 6.2c.

Parkinson's Experiment In the first experiment, we attempted to predict whether a patient has Parkinson's disease based on multiple voice recordings featurized according to doctor specifications [52]. Here, each cluster corresponds to an individual, and each cluster contains multiple voice recordings. The OOC error corresponds to the ability to predict Parkinson's on new individuals not in the training set.

Dota 2 Experiment In the final experiment, we attempt to predict the winner of a Dota 2 video game based on the heroes each team selects at the beginning of the game. This is equivalent to learning an undiscounted value function for a binary, sparse reward function in reinforcement learning. Here, clusters correspond to the type of game played, and we wish to learn a predictor f which generalizes across new game types.

5.5.1 Results

Fig. 5.1 demonstrates that interaction effects between the clustering and prediction algorithm cause the cross-validation error e to decay monotonically and convexly, as predicted by Theorem 9 and Theorem 11. This visually demonstrates the expected adverse behavior – if our clustering algorithm makes even a few mistakes, we may think our predictor has a low error rate, but when we deploy it in the real world on new clusters, it will perform far worse. Empirically, the interaction biases cross-validation results by upwards of 25%.

5.6 Conclusions

We argued that interaction effects between clustering and prediction algorithms can cause dangerous and elusive behavior in machine learning systems. We theoretically characterized when and how this interaction behavior is exhibited, and demonstrated these properties hold in practice on all examined datasets. An important practical takeaway from the analysis of the discussed properties is the introduction of a statistical hypothesis test to detect the bias. In the next chapter, we introduce scalable estimators for the magnitude of the dependency leakage bias, a necessary step in correcting for interaction effects.

The interaction between clustering and prediction algorithms is one common instance of an interaction effect. [80] discussed several other issues in complex machine learning systems, including hidden feedback loops and undeclared data dependencies, which may warrant further exploration.

Chapter 6

The Binomial Block Bootstrap estimator

In Chapter 5, we empirically and theoretically demonstrated that clustering errors cause subtle yet significant adverse behavior in downstream prediction algorithms. Ultimately, our goal is to train and validate a prediction algorithm which is robust under these conditions. Thus, this chapter focuses on precisely estimating the interaction effects, allowing us to correct for the cross-validation bias introduced by the dependency leakage. In other words, we wish to estimate the predictor's loss on new clusters (an extension of the out-of-bag error), given a noisy approximation of the true clustering (the result of the clustering algorithm.)

To this end, we present a novel bootstrap technique for learning on blocks of dependent data, which both estimates and corrects for dependency leakage. This enables learning on clusters of dependent data, where we only observe a noisy approximation of the true clustering. The key insight is to increase dependency by further corrupting $\hat{c}$, in order to extrapolate an unbiased and consistent estimator for the true c. Simulation studies in the non-asymptotic case show our method significantly outperforms standard cross-validation techniques.

6.1 Prior work

There are two largely independent research threads related to our problem of dependency leakage. First, and perhaps most relevant, is the problem of learning with dependent data. Second, previous work has considered the widespread problem of learning with noisy labels, although this has primarily been restricted to standard supervised settings where the ground truth class labels are noisy due to human annotation errors. In our setting, dependency between the training and validation sets is caused by noisy clustering labels.

6.1.1 Learning with dependent data

The problem of constructing estimators for dependent data has been studied since Singh [83], who provided the first theoretical confirmation of the naive bootstrap's performance with IID data, and also showed its inadequacy for dependent data. Since then, the bootstrap has been extended to both time-series and cluster data. In time-series data, blocks of data are dependent according to some stochastic process [37, 54]. By varying the size and separation of the blocks, these block bootstrap methods can limit the dependency and thus control the bias and variance of the estimator, while sometimes achieving consistency. We refer the reader to [48] for a more thorough overview of the subject.

In cluster data, within-cluster samples are dependent while inter-cluster samples are typically assumed to be independent. This is the same formulation as Eq. (1.1). Many bootstrap methods have been proposed for variance estimation in the clustering setting, as classical bootstrap estimators will typically be downward biased [16]. Model-based methods assume a parametric model for the within-cluster error correlation. Model-free methods perform post-estimation bias-correction, such as the cluster-robust variance estimator (CRVE) for ordinary least squares [103] and non-linear settings [51]. CRVE suffers from having unbalanced or a small number of clusters, which is addressed in [57]. Field and Welsh [30] provide theoretical asymptotic analysis for several cluster bootstrap techniques, including the randomized cluster bootstrap, two-stage bootstrap [25] and residual bootstrap [4]. Multi-way bootstrap clustering is slightly more general, but still assumes samples belonging to none of the same clusters are independent [65]. Neither these bootstrap techniques nor LOCO cross-validation account for inter-cluster dependency, and will be inadequate for $\hat{c}$ and non-trivial f, ℓ, X and y.

In practice, when the clustering c is latent, researchers choose a coarse clustering $\hat{c}$ to ensure intra-cluster samples are as independent as possible [16]. A coarser clustering decreases bias and increases variance. This approach both lacks guarantees and requires choosing an appropriate clustering coarseness, which is an open problem. The key differentiation of our work is we directly address the issue of inter-cluster dependency due to $\hat{c}$.

6.1.2 Learning with noisy labels

Previous work has considered the related problem of learning with noisy classification labels, unlike our setting where the clustering labels which define the cross-validation split are corrupted. Schlimer et al. proposed one of the first procedures for predicting class labels which are noisy and drift over time. Kearns provided an early theoretical analysis of which model classes can be efficiently learned in the presence of classification noise [42]. More recent work has focused on constructing estimators of the true loss in the presence of class dependent label errors [55, 67]. A challenging and open problem in this domain is how to learn the class dependent noise rate, which often requires an i.i.d. assumption.

A common source of class label noise is human annotation errors. These have been partially mitigated by explicitely modeling the annotator quality and labeling difficulty, and inferring the true latent label from multiple human annotations, instead of taking a simple majority vote [74]. However, the resulting label will inevitably still be imperfect. This is especially true in the computer vision community, which relies on large, manually labeled datasets for training complex learners. A standard approach is to pretrain a model on noisy data and fine tune on cleaner data, although more recent approaches have benefited from jointly learning the noise and true label (e.g. using a graphical model) [99, 109].

The major difference between previous work and our problem formulation is that cluster labeling errors result in samples being placed in the incorrect cross-validation fold, thus making these folds dependent. This is conceptually distinct from standard classification label noise. We do find that similar assumptions simplify the analysis in our setting – namely that the labeling errors are i.i.d.

6.2 The Binomial Block Bootstrap estimator

We introduce the binomial block bootstrap (B3) class of estimators for cross-validating with dependent blocks of data. First, we begin by formalizing notation to simplify analysis of the core problem. We then proceed with the simplest leakage scenario and gradually build complexity until arriving at our final result. In Section 6.2.2 we begin with the case where samples are moved with known probability in a single direction, from the test blocks to the train blocks or vice versa. Then we show how to solve for the unidirectional dependency leakage in Section 6.2.3 and generalize to the bidirectional case in Section 6.2.4.

6.2.1 Problem Setup

Broadly, we address the problem of LOCO cross-validation when a noisy approximation of the true latent partition c. For the remainder of the chapter, we consider some arbitrary fixed i in Eq. (5.1) (i.e. a single fold). In the LOCO estimator with known partitioning c, each fold is created such that $\mathcal{T}$ and $\mathcal{V}$ are sets of training and testing samples, respectively, split by the partition, i.e. $c(i) \neq c(j)\ \forall x_i \in \mathcal{T}, x_j \in \mathcal{V}$. Without conditioning on the latent cluster parameters in Eq. (1.1), samples within the same cluster are dependent while samples in different clusters are independent. Thus, $\mathcal{T}$ and $\mathcal{V}$ are independent.

Using this notation, we think about the core problem as a learner f trained on samples $\mathcal{T}$ drawn IID from distribution $P_{\mathcal{T}}$ and tested on samples $\mathcal{V}$ drawn IID from a related but different distribution $P_{\mathcal{V}}$, a form of transfer learning.

Now, suppose we instead observe noisy datasets $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, where samples have randomly moved between $\mathcal{T}$ and $\mathcal{V}$. This question arises naturally when we only have $\hat{c}$, an approximation of c, likely obtained through clustering. Most importantly, $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ are dependent — which provides additional information to the learner and biases our cross-validation estimator. Our goal then is to answer questions regarding the continuous loss function ℓ evaluated on new clusters, for example

Algorithm 1 B3: Unidirectional leakage with known probability

```
procedure KnownUnidirectional($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, p_0, dir, n', t$)
    $\bar{b} \leftarrow \vec{0}$
    for $p_i$ in $\{p_0, p_0 + \delta, p_0 + 2\delta, \ldots, 1\}$ do        ▷ Choose $\delta > 0$ s.t. $|\{p_i\}| > n'$
        $p' \leftarrow \frac{p_i - p_0}{1 - p_0}$
        for $j \leftarrow 1$ to $t$ do
            if $dir$ is $\mathcal{V}$ to $\mathcal{T}$ then
                $\mathcal{T}'_j \overset{n'}{\sim} M_{\hat{\mathcal{T}},\hat{\mathcal{V}}}(1 - p', p')$        ▷ $M$ is a mixture distribution[1]
                $\mathcal{V}'_j \leftarrow \hat{\mathcal{V}} \setminus \mathcal{T}'_j$
            else
                $\mathcal{V}'_j \overset{n'}{\sim} M_{\hat{\mathcal{T}},\hat{\mathcal{V}}}(p', 1 - p')$
                $\mathcal{T}'_j \leftarrow \hat{\mathcal{T}} \setminus \mathcal{V}'_j$
            end if
            $\hat{b}_i \leftarrow \frac{1}{|\mathcal{V}'_j|} \sum_{(x,y) \in \mathcal{V}'_j} \ell(y, f(x \mid \mathcal{T}'_j))$        ▷ $\ell$ is any continuous loss function
            $\bar{b}_i \leftarrow \bar{b}_i + \frac{\hat{b}_i}{t}$
        end for
    end for
    $A_{ij} \leftarrow \mathbb{P}(\text{Binomial}(n', p_i) = j) \quad \forall p_i \in p, j \in \{0, 1, \ldots, n'\}$
    $\hat{e}, residual \leftarrow A(A^\intercal A)^{-1} A^\intercal \bar{b}$
    return $\hat{e}_0, residual$
end procedure
```

$\mathbb{E}_{\mathcal{T} \sim P_{\mathcal{T}}} \mathbb{E}_{(x,y) \sim P_{\mathcal{V}}} \ell(f(x \mid \mathcal{T}), y)$, given only noisy datasets $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$.

6.2.2 Unidirectional leakage with known probability

First, consider the case where samples move with known uniform probability from either $\mathcal{V}$ to $\mathcal{T}$ or vice versa to create $\hat{\mathcal{V}}$ and $\hat{\mathcal{T}}$. Without loss of generality, we consider the case where samples move from $\mathcal{V}$ to $\mathcal{T}$. In other words, $\hat{\mathcal{V}}$ contains only samples from $P_{\mathcal{V}}$ while $\hat{\mathcal{T}}$ contains samples from both $P_{\mathcal{T}}$ and $P_{\mathcal{V}}$. Let p_0 be the fraction of samples in $\hat{\mathcal{T}}$ from $\mathcal{V}$, i.e. $p_0 = \frac{|\hat{\mathcal{T}} \cap \mathcal{V}|}{|\hat{\mathcal{T}}|}$. The analysis for the other direction is identical.

The unidirectional B3 estimator (presented in Algorithm 1) is based on the observation that the number of corrupted samples in a bootstrap sample $\mathcal{T}'$ from $\hat{\mathcal{T}}$ is binomially distributed according to p_0 and $n' = |\mathcal{T}'|$. The bootstrap sample $\mathcal{T}'$ is

[1] $M_{S_1,S_2}(w_1, w_2)$ is a mixture distribution of sets S_1 and S_2, where the probability of sampling from the sets are $w_1 + w_2 = 1$, respectively. Within set samples are drawn uniformly.

Algorithm 2 B3: Unidirectional leakage with unknown probability

```
procedure UnknownUnidir($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, dir, n', t$)
    $residual^* \leftarrow \infty$
    $n \leftarrow |\hat{\mathcal{T}}|$
    for $\hat{p}_0$ in $\left\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n-1}{n}\right\}$ do
        $\hat{e}_0, residual \leftarrow$ KnownUnidir($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, \hat{p}_0, dir, n', t$)
        if $residual < residual^*$ then
            $\hat{e}_0^* \leftarrow \hat{e}_0$
            $\hat{p}_0^* \leftarrow \hat{p}_0$
            $residual^* \leftarrow residual$
        end if
    end for
    return $\hat{e}_0^*, \hat{p}_0^*$
end procedure
```

formed by resampling with replacement n' times from $\hat{\mathcal{T}}$, which we notate as $\mathcal{T}' \overset{n'}{\sim} \hat{\mathcal{T}}$. Let b_0 be the expected bootstrap loss estimate, $b_0 = \mathbb{E}_{(x,y)\sim\hat{\mathcal{V}}} \mathbb{E}_{\mathcal{T}' \overset{n'}{\sim} \hat{\mathcal{T}}} \ell(f(x|\mathcal{T}'), y)$. We can express b_0 as a binomial weighting of the expected error at all numbers of corrupted samples in $\mathcal{T}'$. Formally,

$$b_0 = \langle a_0, e \rangle \tag{6.1}$$

where a_0 is the probability mass function (pmf) of Binomial(n', p_0), e_i is the expected loss with i corrupted samples in $\mathcal{T}'$ and $\langle \cdot, \cdot \rangle$ denotes the inner product operation. Our goal is to recover e_0, the loss with zero corruption.

At first, this may seem difficult as $b_0 = \langle a_0, e \rangle$ is a very underdetermined system (even assuming we know p_0). To overcome this deficiency, the key insight of our bootstrap technique is to artificially inject additional leakage by further mixing $\mathcal{V}$ into $\hat{\mathcal{T}}$ to create a fully or over defined system. This increases p, alters the binomial pmf a_0, and generates a new linear equality $b_1 = \langle a_1, e \rangle$ where a_1 is the pmf of

Binomial(n', p_1). Repeating this process many times results in the linear system

$$\begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & \cdot & \cdot & n' \end{array} \\ \begin{array}{c} p_0 \\ p_1 \\ \cdot \\ \cdot \\ 1 \end{array} & \begin{pmatrix} \leftarrow \text{Binomial pmf} \rightarrow \\ \cdot \\ \cdot \\ \cdot \\ \leftarrow \text{Binomial pmf} \rightarrow \end{pmatrix} \end{array} \begin{pmatrix} \\ \\ e \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix} \tag{6.2}$$

$$A(p_0) \qquad\qquad e \quad = \quad b$$

For any unique choice of $p = (p_0, p_1, \ldots, p_m) \in [0, 1]^m$, this system will be well-defined (by Lemma 13) and can be readily solved for e_0. A somewhat similar clustering randomization idea is used in [98] for estimating treatment effects, though their formulation is quite different than Eq. (6.2).

Lemma 13. *Let matrix A be defined such that*

$$A_{ij} = \mathbb{P}(\text{Binomial}(n', p_i) = j). \tag{6.3}$$

Then A has full rank for any choice of unique parameters $p = (p_0, p_1, \ldots, p_m) \in [0, 1]^m$.

Proof. Let A_j denote column j of matrix A. The entries of A_j correspond to polynomial $g_j(q) = \binom{n'}{j} q^j (1-q)^{(n'-j)}$ evaluated at points $q = p_0, p_1, \ldots, p_m$. First, we show polynomials $g(q) = \{g_0(q), \ldots, g_{n'}(q)\}$ are linearly independent.

We look for a non-trivial solution $\kappa \in \mathbb{R}^{(n'+1)}$ to $\langle \kappa, g(q) \rangle = 0, \forall q \in [0, 1]$. The binomial coefficient is a constant in each polynomial and can be dropped. Expanding and collecting terms,

$$\begin{aligned} 0 &= \kappa_0 (1-q)^{n'} + \kappa_1 q^1 (1-q)^{n'-1} + \ldots + \kappa_{n'} q^{n'} \\ &= \sum_{i=0}^{n'} q^i \sum_{j=0}^{i} \kappa_j \binom{n'-j}{i-j} (-1)^j \end{aligned}$$

for all $q \in \left[0, \frac{n-1}{n}\right]$ which implies

$$0 = \sum_{j=0}^{i} \kappa_j \binom{n' - j}{i - j} (-1)^j$$

for all $i \in \{0, 1, \ldots, n'\}$ and all $q \in \left[0, \frac{n-1}{n}\right]$. Clearly, $\kappa_0 = 0$, and the remainder of the terms follow by induction to $\kappa = \vec{0}$. Thus the polynomials $\{g_0, \ldots, g_{n'}\}$ are linearly independent.

Next, the polynomials $\{g_0, \ldots, g_{n'}\}$ are unisolvent by the unisolvence theorem, which implies the vectors

$$\begin{bmatrix} g_0(p_0) \\ g_0(p_1) \\ \vdots \\ g_0(p_m) \end{bmatrix}, \begin{bmatrix} g_1(p_0) \\ g_1(p_1) \\ \vdots \\ g_1(p_m) \end{bmatrix}, \ldots, \begin{bmatrix} g_{n'}(p_0) \\ g_{n'}(p_1) \\ \vdots \\ g_{n'}(p_m) \end{bmatrix}, \tag{6.4}$$

are also linearly independent for any unique $p_0, \ldots, p_m$, $m \geq n'$. Thus, matrix A is full rank. □

Roughly speaking, Algorithm 1 is estimating the loss at increasing levels of dependency leakage, and then extrapolating the loss at zero dependency. It is possible to achieve reasonable results in practice because we know the true formulation to be a binomial weighted regression problem and thus know matrix A exactly. Further, the extrapolation does not extend far beyond the known range for practical clusterings $\hat{c}$ with small p_0.

The estimator $\hat{e}_0$ in Algorithm 1 is consistent, unbiased and has variance decreasing linearly with respect to the number of bootstrap samples t.

Theorem 14. *The estimator $\hat{e}_0$ in Algorithm 1 satisfies*

1. *Consistent: $\hat{e}_0 \xrightarrow{p} \mathop{\mathbb{E}}_{\mathcal{T}' \overset{n'}{\sim} P_{\mathcal{T}}} \mathop{\mathbb{E}}_{(x,y) \sim P_{\mathcal{V}}} \ell(y, f(x \mid \mathcal{T}'))$ as $t, |\hat{\mathcal{T}}|, |\hat{\mathcal{V}}| \to \infty$*
2. *Unbiased: $\mathbb{E}[\hat{e}_0] = e_0$ for finite t and infinite $|\hat{\mathcal{T}}|, |\hat{\mathcal{V}}|$.*

3. $\mathrm{Var}(\hat{e}_0) =$

$$\sum_{i=0}^{n'} \left[\frac{\sum\limits_{\substack{0 \leq m_0 < \cdots < m_{n'-1} \leq n' \\ m_0, \ldots, m_{n'-1} \neq i}} p_{m_0} \cdots p_{m_{n'-1}}}{\prod\limits_{0 \leq m \leq n', m \neq i} (p_m - p_i)} \right]^2 \frac{\sigma_{b_i}^2}{t}$$

where $\sigma_{b_i}^2$ is the variance of $\hat{b}_i$ in Algorithm 1, which is a function of f, ℓ and the data.

Proof.

Statement 1 Without loss of generality, we prove the corruption direction from $\mathcal{V}$ to $\mathcal{T}$. The empirical loss of KnownUnidirectional at corruption level p_i is $\bar{b}_i$.

$$\begin{aligned}
\bar{b}_i &= \frac{1}{t|\hat{\mathcal{V}}|} \sum_{\mathcal{T}' \in \{\mathcal{T}'_0, \ldots, \mathcal{T}'_t\}} \sum_{(x,y) \in \hat{\mathcal{V}}} \ell(y, f(x \mid \mathcal{T}')) \\
&= \frac{1}{t|\hat{\mathcal{V}}|} \sum_{(x,y) \in \hat{\mathcal{V}}} \sum_{j=0}^{n'} \sum_{\mathcal{T}' : |\mathcal{T}' \cap \mathcal{V}| = j, \mathcal{T}' \in \{\mathcal{T}'_0, \ldots, \mathcal{T}'_t\}} \ell(y, f(x \mid \mathcal{T}')) \\
&\xrightarrow[t \to \infty]{p} \frac{1}{|\hat{\mathcal{V}}|} \sum_{(x,y) \in \hat{\mathcal{V}}} \sum_{j=0}^{n'} A_{ij} \mathbb{E}_{\mathcal{T}''(j)} \ell(y, f(x \mid \mathcal{T}'')) \\
&\xrightarrow[|\mathcal{V}|, |\mathcal{T}| \to \infty]{p} \mathop{\mathbb{E}}_{(x,y) \sim P_{\mathcal{V}}} \sum_{j=0}^{n'} A_{ij} \mathbb{E}_{\mathcal{T}'''(j)} \ell(y, f(x \mid \mathcal{T}''')) \\
&= \sum_{j=0}^{n'} A_{ij} e_j = b_i
\end{aligned}$$

where A_{ij} is the probability of sampling j samples from $\mathcal{V}$ at corruption p_i as defined in Eq. 6.3 and

$$\begin{aligned}
\mathcal{T}''(j) &= \left\{ \{\mathcal{T}'''' \overset{n'-j}{\sim} \mathcal{T}\} \cup \{\mathcal{V}'' \overset{j}{\sim} \mathcal{V}\} \right\} \\
\mathcal{T}'''(j) &= \left\{ \{\mathcal{T}'''' \overset{n'-j}{\sim} P_{\mathcal{T}}\} \cup \{\mathcal{V}'' \overset{j}{\sim} P_{\mathcal{V}}\} \right\} \\
e_j &= \mathbb{E}_{\mathcal{T}'''(j), (x,y) \sim P_{\mathcal{V}}} \ell(y, f(x \mid \mathcal{T}'))
\end{aligned}$$

The proof can be understood as splitting the t bootstraps into bins $j = 0, \ldots, n'$

each with probability A_{ij}. Then $\hat{e} = A(A^{\intercal}A)^{-1}A^{\intercal}\bar{b} \xrightarrow{p} A(A^{\intercal}A)^{-1}A^{\intercal}b = e$ by the continuous mapping theorem and Lemma 13. We require $|\mathcal{V}|, |\mathcal{T}| \to \infty$ sufficiently faster than $t \to \infty$, such that the probability of resampling the same data sample in Algorithm 1 also goes to 0.

Statement 2 Recall $\hat{e} = A(A^{\intercal}A)^{-1}A^{\intercal}b$ and $\hat{e} = A(A^{\intercal}A)^{-1}A^{\intercal}\bar{b}$. Then

$$\begin{aligned}
\mathbb{E}[\hat{e}] &= \mathbb{E}[A(A^{\intercal}A)^{-1}A^{\intercal}\bar{b}] \\
&= A(A^{\intercal}A)^{-1}A^{\intercal}\mathbb{E}[\bar{b}] \\
&= A(A^{\intercal}A)^{-1}A^{\intercal}b \\
&= e
\end{aligned}$$

for finite t but infinitely large $|\hat{\mathcal{V}}|$ and $|\hat{\mathcal{T}}|$. For finite sample sets, bias may be introduced due to resampling the same data sample more than once.

Statement 3 We consider the case where A is square for analysis simplicity. Let $\sigma_{b_i}^2$ be the variance of $\hat{b}_i$ in line 13 of Algorithm 1. Then

$$\begin{aligned}
\mathrm{Var}(\hat{e}_0) &= \sum_{i=0}^{n'} (A^{-1})_{0i}^2 \mathrm{Var}(\hat{b}_i) \\
&= \sum_{i=0}^{n'} (A^{-1})_{0i}^2 \frac{\sigma_{b_i}^2}{t} \\
&= \frac{1}{t} u^{\intercal} (A^{-1})^{\intercal} \Sigma A^{-1} u
\end{aligned}$$

where Σ is the diagonal matrix with entires $\Sigma_{ii} = \sigma_{b_i}^2$ and u denotes a standard unit vector, such that $u_0 = 1$. Let $z = A^{-1}u$. Now solving for z

$$\begin{aligned}
Az &= u \\
\binom{n'}{j} \sum_{i=0}^{n'} p_i^j (1 - p_i)^{n'-j} e_i &= \begin{cases} 1 & j = 0 \\ 0 & j = 1, \dots, n' \end{cases}
\end{aligned}$$

For $j = n'$, note $\sum_{i=0}^{n'} p_i^{n'} z_i = 0$. Then expanding terms for $j = n' - 1$

$$\begin{aligned} \sum_{i=0}^{n'} p_i^{n'-1}(1 - p_i) z_i = 0 \\ \sum_{i=0}^{n'} p_i^{n'-1} z_i - p_i^{n'} z_i = 0 \\ \sum_{i=0}^{n'} p_i^{n'-1} z_i = 0 \end{aligned}$$

Continuing these expansion and substitution steps for $j = n' - 1, n' - 2, \ldots, 1, 0$ results in

$$\sum_{i=0}^{n'} p_i^k z_i = u_k \quad \text{for } k = 0, \ldots, n'$$

which is a transposed Vandermonde system

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ p_0 & p_1 & p_2 & \cdots & p_{n'} \\ p_0^2 & p_1^2 & p_2^2 & \cdots & p_{n'}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_0^{n'} & p_1^{n'} & p_2^{n'} & \cdots & p_{n'}^{n'} \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{n'} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{aligned} W^\intercal z &= u \\ z &= (W^{-1})^\intercal u \end{aligned}$$

Thus, $\mathrm{Var}(\hat{e}_0) = \frac{1}{t} u^\intercal W^{-1} \Sigma (W^{-1})^\intercal u$ and the inverse of the Vandermonde matrix is known [58]

$$(W^{-1})_{ij} = \begin{cases} (-1)^i \left(\frac{\sum_{0 \le m_0 < \cdots < m_{n'-1} \le n', m_0, \ldots, m_{n'-i} \ne j} p_{m_0} \cdots p_{m_{n'-i}}}{\prod_{0 \le m \le n', m \ne j} (p_m - p_j)} \right) & \text{for } 0 \le i \le n' \\ \frac{1}{\prod_{0 \le m \le n', m \ne j} (p_m - p_j)} & \text{for } k = n' \end{cases}$$

which gives the result. □

Remark. *For classification error ℓ, note $\sigma_{b_j}^2 \le \frac{1}{4}$ by Popoviciu's inequality. Generally speaking, there exists a variance tradeoff when choosing $p_0, \ldots, p_{n'}$ — we can expect* *lower variance as the values are spaced further apart (larger denominator) and when they are closer to p_0 (smaller numerator), which are competing choices.*

Remark. *The quality of the clustering $\hat{c}$ plays an important role in the performance of our estimator. As p_0 increases, the estimator remains unbiased but the variance increases according to Statement 3.*

6.2.3 Unidirectional leakage with unknown probability

We now extend the unidirectional leakage scenario from Section 6.2.2 to the situation where p_0 is unknown a priori. The general strategy is to minimize the residual $||A(\hat{p}_0)e - \bar{b}||$ over $\hat{p}_0$ and show that a unique minimum exists and it is always the true leakage probability p_0. The most basic optimization procedure detailed in Algorithm 2 searches over the discrete set of possible solutions, though one can imagine other optimization procedures. The search space will be, at most, the one dimensional line defined by $\left[0, \frac{n-1}{n}\right]$ where $n = |\hat{\mathcal{T}}|$.

Our optimization routine in Algorithm 2 converges to the true leakage probability p_0 if the following assumption holds

Assumption 1. *b is independent of the columns of $A(\hat{p}_0)$ (except, obviously, at $\hat{p}_0 = p_0$).*

Remark. *This is a weak assumption when choosing $m >> n'$: it is unlikely the loss vector b happens to fall in the column space of A.*

Theorem 15. *If Assumption 1 holds, then the estimators $\hat{p}_0^*$ and $\hat{e}_0^*$ in Algorithm 2 are consistent, i.e. $\hat{p}_0^* \xrightarrow{p} p_0$ and $\hat{e}_0^* \xrightarrow{p} e_0$ as $t, |\mathcal{T}|, |\mathcal{V}| \to \infty$ and for $p_0 < 1$.*

Proof. Without loss of generality, we prove the case where samples move in the direction from $\mathcal{V}$ to $\mathcal{T}$. We begin by proving the convergence of p_0^*. Let $n = |\hat{\mathcal{T}}|$. In Algorithm 2, $p_0^{*(t)} = \text{argmin}_{p_0 \in \left\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n-1}{n}\right\}} g^{(t)}(p_0)$, where the function $g^{(i)}(p_0) = ||A(p_0)(A^\intercal(p_0)A(p_0))^{-1}A^\intercal(p_0)\bar{b}^{(i)} - \bar{b}^{(i)}||_2^2$ if $p_0 \in \left[0, \frac{n-1}{n}\right]$ and else infinity. We use $\bar{b}^{(i)}$ to denote the mean estimator $\bar{b}$ in Algorithm 1 after $t = i$ samples. Let $g(p_0) = ||A(p_0)(A^\intercal(p_0)A(p_0))^{-1}A^\intercal(p_0)b - b||_2^2$ if $p_0 \in \left[0, \frac{n-1}{n}\right]$ and else infinity.

Both g and the sequence of functions $\{g^{(0)}, g^{(1)}, \ldots\}$ are level-bounded, lower semi-continuous and proper. By Lemma 16, $g^{(i)} \xrightarrow{e} g$ where $\xrightarrow{e}$ denotes convergence in epigraph. Thus, *residual* $= \min_{p_0 \in \left[0, \frac{n-1}{n}\right]} g^{(t)}(p_0) \xrightarrow{p} \min_{p_0} g(p_0)$ [75]. We know at least one perfect solution $g(p_0) = 0$ exists, that this solution is unique (by Assumption 1) and that this solution is in $\left\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}\right\}$. Thus, $p_0^* \xrightarrow{p} p_0$ and *residual* $\xrightarrow{p} 0$. □

Lemma 16. *Let*

$$g^{(i)}(p_0) = \begin{cases} ||A(p_0)(A^\intercal(p_0)A(p_0))^{-1}A^\intercal(p_0)\bar{b}^{(i)} - \bar{b}^{(i)}||_2^2 \\ \qquad\qquad\qquad\qquad \textit{if} \quad p_0 \in \left[0, \frac{n-1}{n}\right] \\ \infty \qquad\qquad\qquad\quad\;\; \textit{else} \end{cases}$$

$$g(p_0) = \begin{cases} ||A(p_0)(A^\intercal(p_0)A(p_0))^{-1}A^\intercal(p_0)b - b||_2^2 \\ \qquad\qquad\qquad\qquad \textit{if} \quad p_0 \in \left[0, \frac{n-1}{n}\right] \\ \infty \qquad\qquad\qquad\quad\;\; \textit{else} \end{cases}$$

Then $g^{(i)} \xrightarrow{e} g$, where we use $\xrightarrow{e}$ to denote convergence in epigraph.

Proof. Recall, $g^{(i)} \xrightarrow{e} g$ if and only if at each point e

$$\liminf_i g^{(i)}(e^{(i)}) \geq g(e), \text{ for every } e^{(i)} \to e \tag{6.5a}$$

$$\limsup_i g^{(i)}(e^{(i)}) \leq g(e), \text{ for some } e^{(i)} \to e \tag{6.5b}$$

Let $\mathcal{N}_\infty^\# = \{N \in \mathbb{N} | N \text{ is infinite}\}$ be all infinite sets of natural numbers, which we require for cases of periodicity. To establish Eq. (6.5a), it is sufficient to show that whenever $e^{(i)} \underset{N}{\rightarrow} e$ and $f^{(i)}(e^{(i)}) \underset{N}{\rightarrow} \alpha$, then $f(e) \leq \alpha$. We consider three cases, when $e \in \left(0, \frac{n-1}{n}\right)$, when $e \notin \left[0, \frac{n-1}{n}\right]$ and when $e \in \left\{0, \frac{n-1}{n}\right\}$. The first case is readily established from the proof of Theorem 14, where we showed that $\bar{b}^{(i)} \underset{N}{\rightarrow} b \; \forall N \in \mathcal{N}_\infty^\#$, $A(e^{(i)})(A^\intercal(e^{(i)})A(e^{(i)}))^{-1}A^\intercal(e^{(i)}) \underset{N}{\rightarrow} A(e)(A^\intercal(e)A(e))^{-1}A^\intercal(e)$, and thus $f^{(i)}(e^{(i)}) \underset{N}{\rightarrow} f(e) \; \forall N \in \mathcal{N}_\infty^\#$. In the case where $e \notin \left[0, \frac{n-1}{n}\right]$, $g^{(i)}(e) = \infty$ readily establishes the inequality. In the boundary cases $e \in \left\{0, \frac{n-1}{n}\right\}$, note either $g^{(i)}(e^{(i)}) \underset{N}{\rightarrow} \infty$ or $g^{(i)}(e^{(i)}) \underset{N}{\rightarrow} g(e)$, respectively. To establish Eq. (6.5b), choose the sequence $\{e^{(i)}\} = e \; \forall i \in \mathbb{N}$. □

6.2.4 Bidirectional leakage with unknown probabilities

Lastly, we extend the unidirectional leakage results in Section 6.2.2 and Section 6.2.3 to the full bidirectional setting, where samples move with unknown uniform probability between $\mathcal{T}$ and $\mathcal{V}$. More specifically, let $p_{\mathcal{T},0} = \frac{|\hat{\mathcal{T}} \cap \mathcal{V}|}{|\hat{\mathcal{T}}|}$ and $p_{\mathcal{V},0} = \frac{|\hat{\mathcal{V}} \cap \mathcal{T}|}{|\hat{\mathcal{V}}|}$ be the probabilities a sample in $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ do not belong in that set, respectively. Similar to the unidirectional case, we independently resample with replacement $n'_{\mathcal{T}}$ and $n'_{\mathcal{V}}$ samples from $\mathcal{T}$ and $\mathcal{V}$ to form the bootstrap sample sets $\mathcal{T}'$ and $\mathcal{V}'$, respectively. Thus, the number of corrupted samples in $\mathcal{T}'$ and $\mathcal{V}'$ is drawn according to a joint distribution of two independent binomials. We then formulate a regression problem analogous to Eq. (6.2),

$$\begin{array}{cc}
 & \begin{array}{ccccc} 0 & 1 & \cdot & \cdot & n' \end{array} \\
\begin{array}{c} p_{\mathcal{T},0}, p_{\mathcal{V},0} \\ \cdot \\ \cdot \\ \cdot \\ p_{\mathcal{T},n_{\mathcal{T}}}, p_{\mathcal{V},n_{\mathcal{V}}} \end{array} & \begin{pmatrix} \leftarrow \text{Joint Bin pmf} \rightarrow \\ \cdot \\ \cdot \\ \cdot \\ \leftarrow \text{Joint Bin pmf} \rightarrow \end{pmatrix}
\end{array}
\begin{pmatrix} \\ \\ e \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix}$$

$$A(p_{\mathcal{T},0}, p_{\mathcal{V},0}) \qquad e = b$$

where $n' = (n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)-1$. Note since the joint pmf is defined for $(n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)$ values, we must bootstrap at $(n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)$ levels of leakage.

In the case where the leakage probabilities $p_{\mathcal{T},0}$ and $p_{\mathcal{V},0}$ are unknown, we again minimize the residual. The resulting methods for the bidirectional leakage scenario with known and unknown probabilities are presented in Algorithm 3 and Algorithm 4, respectively.

Here, we show the full rank and consistency results for Algorithm 1 and Algorithm 2 extend to Algorithm 3 and Algorithm 4. The main difference is we consider the *joint* binomial matrix A, which is also full rank and thus the regression problem is well defined.

Lemma 17. *Joint binomial matrix A has full rank for any choice of unique parameters $p_{\mathcal{T}} = (p_{\mathcal{T},0}, p_{\mathcal{T},1}, \ldots, p_{\mathcal{T},m}) \in [0,1]^m$ and $p_{\mathcal{V}} = (p_{\mathcal{V},0}, p_{\mathcal{V},1}, \ldots, p_{\mathcal{V},m'}) \in [0,1]^{m'}$.*

Algorithm 3 B3: Bidirectional leakage with known probabilities

```
procedure KnownBidirectional($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, p_{\mathcal{T},0}, p_{\mathcal{V},0}, n'_{\mathcal{T}}, n'_{\mathcal{V}}, t$)
    $\bar{b} \leftarrow \nvdash^{(n_{\mathcal{T}'}+1)\times(n_{\mathcal{V}'}+1)}$
    for $p_i$ in $\{p_{\mathcal{T},0}, p_{\mathcal{T},0}+\delta_{\mathcal{T}}, p_{\mathcal{T},0}+2\delta_{\mathcal{T}}, \ldots, 1\}$ do    ▷ Choose $\delta_{\mathcal{T}} > 0$ s.t. $|\{p_i\}| > n_{\mathcal{T}}$
        $p'_{\mathcal{T}} \leftarrow \frac{p_i + p_{\mathcal{V},0} - 1}{p_{\mathcal{T},0} + p_{\mathcal{V},0} - 1}$
        for $p_j$ in $\{p_{\mathcal{V},0}, p_{\mathcal{V},0}+\delta_{\mathcal{V}}, p_{\mathcal{V},0}+2\delta_{\mathcal{V}}, \ldots, 1\}$ do    ▷ Choose $\delta_{\mathcal{V}} > 0$ s.t. $|\{p_j\}| > n_{\mathcal{V}}$
            $p'_{\mathcal{V}} \leftarrow \frac{p_j + p_{\mathcal{T},0} - 1}{p_{\mathcal{V},0} + p_{\mathcal{T},0} - 1}$
            for $k \leftarrow 1$ to $t$ do
                $\mathcal{T}'_k \overset{n'_{\mathcal{T}}}{\sim} M_{\hat{\mathcal{T}},\hat{\mathcal{V}}}(p'_{\mathcal{T}}, 1-p'_{\mathcal{T}})$    ▷ $M$ is a mixture distribution[2]
                $\mathcal{V}'_k \overset{n'_{\mathcal{V}}}{\sim} M_{\hat{\mathcal{V}},\hat{\mathcal{T}}}(p'_{\mathcal{V}}, 1-p'_{\mathcal{V}})$
                $\hat{b}_{ij} \leftarrow \frac{1}{|\mathcal{V}'_k|}\sum_{(x,y)\in\mathcal{V}'_k} \ell(y, f(x \mid \mathcal{T}'_k))$    ▷ $\ell$ is any continuous loss function
                $\bar{b}_{ij} \leftarrow \bar{b}_{ij} + \frac{\hat{b}_{ij}}{t}$
            end for
        end for
    end for
    $\bar{b} \leftarrow \text{flatten}(\bar{b})$
    $A_{ijkl} \leftarrow \mathbb{P}(\text{Bin}(n'_{\mathcal{T}}, p_i) = k)\mathbb{P}(\text{Bin}(n'_{\mathcal{V}}, p_j) = l) \quad \forall p_i, p_j, k \in \{0, 1, \ldots, n'_{\mathcal{T}}\}, l \in \{0, 1, \ldots, n'_{\mathcal{V}}\}$
    $A \leftarrow reshape(A \in \mathbb{R}^{|\{p_i\}||\{p_j\}|\times(n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)})$    ▷ Each row of $A$ is a joint binomial pmf
    $\hat{e}, residual \leftarrow A(A^{\intercal}A)^{-1}A^{\intercal}\bar{b}$
    return $\hat{e}_0, residual$
end procedure
```

Proof. Let $A_{k+n'_{\mathcal{T}}l}$ denote column $k+n'_{\mathcal{T}}l$ of matrix A. The entries of $A_{k+n'_{\mathcal{T}}l}$ correspond to polynomial $g_{k+n'_{\mathcal{T}}l}(q_1, q_2) = \binom{n'_{\mathcal{T}}}{k} q_1^k (1-q_1)^{(n'_{\mathcal{T}}-k)} \binom{n'_{\mathcal{V}}}{l} q_2^l (1-q_2)^{(n'_{\mathcal{V}}-l)}$ evaluated at points $q_1 = p_{\mathcal{T},0}, p_{\mathcal{T},1}, \ldots, p_{\mathcal{T},m}$ and $q_2 = p_{\mathcal{V},0}, p_{\mathcal{V},1}, \ldots, p_{\mathcal{V},n}$. First, we show polynomials $g = \{g_0, \ldots, g_{(n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)-1}\}$ are linearly independent.

We look for a non-trivial solution $K \in \mathbb{R}^{(n'_{\mathcal{T}}+1, n'_{\mathcal{V}}+1)}$ to $\langle c, g(q_1, q_2)\rangle = 0, \forall q_1, q_2 \in [0, 1]$ where κ is a flattened version of K. The binomial coefficient is a constant in

Algorithm 4 B3: Bidirectional leakage with unknown probabilities

```
procedure UnknownBidirectional($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, n'_{\mathcal{T}}, n'_{\mathcal{V}}, t$)
    $residual^* \leftarrow \infty$
    $n_{\mathcal{T}} \leftarrow |\hat{\mathcal{T}}|, n_{\mathcal{V}} \leftarrow |\hat{\mathcal{V}}|$
    for $p_{\mathcal{T},0}$ in $\left\{\frac{0}{n_{\mathcal{T}}}, \frac{1}{n_{\mathcal{T}}}, \dots, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right\}$ do
        for $p_{\mathcal{V},0}$ in $\left\{\frac{0}{n_{\mathcal{V}}}, \frac{1}{n_{\mathcal{V}}}, \dots, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right\}$ do
            $e, residual \leftarrow$ KnownBidirectional($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, p_{\mathcal{T},0}, p_{\mathcal{V},0}, n'_{\mathcal{T}}, n'_{\mathcal{V}}, t$)
            if $residual < residual^*$ then
                $e^* \leftarrow x$
                $p^*_{\mathcal{T},0} \leftarrow p_{\mathcal{T},0}$
                $p^*_{\mathcal{V},0} \leftarrow p_{\mathcal{V},0}$
                $residual^* \leftarrow residual$
            end if
        end for
    end for
    return $e^*, p^*_{\mathcal{T},0}, p^*_{\mathcal{V},0}$
end procedure
```

each polynomial and can be dropped. Expanding and collecting terms,

$$\begin{aligned}
0 &= \sum_{i=0}^{n'_{\mathcal{T}}} \sum_{j=0}^{n'_{\mathcal{V}}} K_{ij} q_1^i (1-q_1)^{n'_{\mathcal{T}}-i} q_2^j (1-q_2)^{n'_{\mathcal{V}}-j} \\
&= \sum_{j=0}^{n'_{\mathcal{V}}} q_2^j (1-q_2)^{n'_{\mathcal{V}}-j} \sum_{i=0}^{n'_{\mathcal{T}}} q_1^i \sum_{k=0}^{i} K_{kj} \binom{n'_{\mathcal{T}}-k}{i-k} (-1)^k \\
&= \sum_{i=0}^{n'_{\mathcal{T}}} q_1^i \sum_{k=0}^{i} \binom{n'_{\mathcal{T}}-k}{i-k} (-1)^k \sum_{j=0}^{n'_{\mathcal{V}}} K_{kj} q_2^j (1-q_2)^{n'_{\mathcal{V}}-j} \\
&= \sum_{i=0}^{n'_{\mathcal{T}}} q_1^i \sum_{k=0}^{i} \binom{n'_{\mathcal{T}}-k}{i-k} \sum_{j=0}^{n'_{\mathcal{V}}} q_2^j \sum_{l=0}^{j} K_{kl} \binom{n'_{\mathcal{V}}-l}{j-l} (-1)^{k+l}
\end{aligned}$$

which implies,

$$0 = \sum_{k=0}^{i} \sum_{l=0}^{j} K_{kl} \binom{n'_{\mathcal{T}}-k}{i-k} \binom{n'_{\mathcal{V}}-l}{j-l} (-1)^{k+l}$$

for all $i \in \{0, 1, \dots, n'_{\mathcal{T}}\}$ and $j \in \{0, 1, \dots, n'_{\mathcal{V}}\}$. When $i = 0, j = 0 \Rightarrow K_{00} = 0$ and the remainder of the terms follow by induction to $K = \nvdash^{(n'_{\mathcal{T}}+1)\times(n'_{\mathcal{T}}+1)}$. Thus the polynomials $\{g_0, \dots, g_{(n'_{\mathcal{T}}+1)(n'_{\mathcal{V}}+1)-1}\}$ are linearly independent and A is full rank by the unisolvence theorem. □

Likewise, the consistency results in Theorems 14 and 15 extend to the bidirectional leakage scenario.

Theorem 18. *For $p_{\mathcal{T},0}, p_{\mathcal{V},0} < 1$ in Algorithm 3, e_0 converges to the expected error on uncorrupted distributions $\mathcal{T}$ and $\mathcal{V}$, $\hat{e}_0 \rightarrow \mathbb{E}_{\mathcal{T}' \overset{n'_{\mathcal{T}}}{\sim} P_{\mathcal{T}}} \mathbb{E}_{(x,y)\sim P_{\mathcal{V}}} \ell(y, f(x \mid \mathcal{T}'))$ as $t, |\mathcal{T}|, |\mathcal{V}| \rightarrow \infty$.*

Proof. The empirical loss at corruption levels $(p_{\mathcal{T},i}, p_{\mathcal{V},j})$ is $\bar{b}_{ij}$ in Algorithm 3.

$$\begin{aligned}
\bar{b}_{ij} &= \frac{1}{tn'_{\mathcal{V}}} \sum_{\mathcal{T}' \in \{\mathcal{T}'_0,\dots,\mathcal{T}'_t\}} \sum_{\mathcal{V}' \in \{\mathcal{V}'_0,\dots,\mathcal{V}'_t\}} \sum_{(x,y)\in\mathcal{V}'} \ell(y, f(x \mid \mathcal{T}')) \\
&= \frac{1}{tn'_{\mathcal{V}}} \sum_{l=0}^{n'_{\mathcal{V}}} \sum_{\mathcal{V}' : |\mathcal{V}'\cap\mathcal{T}|=l, \mathcal{V}' \in \{\mathcal{V}'_0,\dots,\mathcal{V}'_t\}} \sum_{(x,y)\in\mathcal{V}'} \\
&\qquad \sum_{k=0}^{n'_{\mathcal{T}}} \sum_{\mathcal{T}' : |\mathcal{T}'\cap\mathcal{V}|=k, \mathcal{T}' \in \{\mathcal{T}'_0,\dots,\mathcal{T}'_t\}} \ell(y, f(x \mid \mathcal{T}')) \\
&\underset{t\to\infty}{\xrightarrow{p}} \sum_{l=0}^{n'_{\mathcal{V}}} \sum_{k=0}^{n'_{\mathcal{T}}} A_{ijkl} \mathbb{E}_{\mathcal{T}''(k),(x,y)\in\mathcal{V}''(l)} \ell(y, f(x \mid \mathcal{T}'')) \\
&\underset{|\mathcal{V}|,|\mathcal{T}|\to\infty}{\xrightarrow{p}} \sum_{l=0}^{n'_{\mathcal{V}}} \sum_{k=0}^{n'_{\mathcal{T}}} A_{ijkl} \mathbb{E}_{\mathcal{T}'''(k),(x,y)\in\mathcal{V}'''(l)} \ell(y, f(x \mid \mathcal{T}''')) \\
&= \sum_{l=0}^{n'_{\mathcal{V}}} \sum_{k=0}^{n'_{\mathcal{T}}} A_{ijkl} x_{kl} \qquad (6.6)
\end{aligned}$$

where A_{ijkl} is the probability of k corrupted samples in $\mathcal{T}'$ and l corrupted samples in $\mathcal{V}'$ at corruption levels $p_{\mathcal{T},i}$ and $p_{\mathcal{V},j}$, i.e.

$$A_{ijkl} = \mathbb{P}(\text{Binomial}(n'_{\mathcal{T}}, p_{\mathcal{T},i}) = k)\mathbb{P}(\text{Binomial}(n'_{\mathcal{V}}, p_{\mathcal{V},j}) = l)$$

and

$$\begin{aligned}
\mathcal{T}''(k) &= \left\{ \{\mathcal{T}'''' \overset{n'_{\mathcal{T}}-k}{\sim} \mathcal{T}\} \cup \{\mathcal{V}'''' \overset{k}{\sim} \mathcal{V}\} \right\} \\
\mathcal{V}''(l) &= \left\{ \{\mathcal{T}'''' \overset{l}{\sim} \mathcal{T}\} \cup \{\mathcal{V}'''' \overset{n'_{\mathcal{V}}-l}{\sim} \mathcal{V}\} \right\} \\
\mathcal{T}'''(k) &= \left\{ \{\mathcal{T}'''' \overset{n'_{\mathcal{T}}-k}{\sim} P_{\mathcal{T}}\} \cup \{\mathcal{V}'''' \overset{k}{\sim} P_{\mathcal{V}}\} \right\} \\
\mathcal{V}'''(l) &= \left\{ \{\mathcal{T}'''' \overset{l}{\sim} P_{\mathcal{T}}\} \cup \{\mathcal{V}'''' \overset{n'_{\mathcal{V}}-l}{\sim} P_{\mathcal{V}}\} \right\} \\
x_{kl} &= \mathbb{E}_{\mathcal{T}'''(k),(x,y)\in\mathcal{V}'''(l)} \ell(y, f(x \mid \mathcal{T}'''))
\end{aligned}$$

We flatten x, b and appropriately reshape the tensor A into a matrix such that Eq. 6.6 is always satisfied in the linear system $Ax = b$. Then, $\hat{e} = A(A^\intercal A)^{-1} A^\intercal \bar{b} \xrightarrow{p} A(A^\intercal A)^{-1} A^\intercal b = x$ by the continuous mapping theorem and Lemma 13. □

Theorem 19. *For $p_{\mathcal{T},0}, p_{\mathcal{V},0} < 1$ in Algorithm 4, $p^*_{\mathcal{T},0} \xrightarrow{p} p_{\mathcal{T},0}$, $p^*_{\mathcal{V},0} \xrightarrow{p} p_{\mathcal{V},0}$ and $\hat{e}^* \xrightarrow{p} e^*$ as $t, |\mathcal{T}|, |\mathcal{V}| \to \infty$.*

Proof. We begin by proving the convergence of $p^*_{\mathcal{T},0}$ and $p^*_{\mathcal{V},0}$. Let $n_{\mathcal{T}} = |\hat{\mathcal{T}}|$ and $n_{\mathcal{V}} = |\hat{\mathcal{V}}|$. In Algorithm 4, $(p^*_{\mathcal{T},0}, p^*_{\mathcal{V},0}) = \operatorname{argmin}_{p_{\mathcal{T},0} \in \left\{\frac{0}{n_{\mathcal{T}}}, \frac{1}{n_{\mathcal{T}}}, \dots, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right\}, p_{\mathcal{V},0} \in \left\{\frac{0}{n_{\mathcal{V}}}, \frac{1}{n_{\mathcal{V}}}, \dots, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right\}} g^{(t)}(p_{\mathcal{T},0}, p_{\mathcal{V},0})$, where the function $g^{(i)}$ is defined as

$$g^{(i)}(p_{\mathcal{T},0}, p_{\mathcal{V},0}) = \begin{cases} ||A(p_{\mathcal{T},0}, p_{\mathcal{V},0})(A^\intercal(p_{\mathcal{T},0}, p_{\mathcal{V},0}) A(p_{\mathcal{T},0}, p_{\mathcal{V},0}))^{-1} A^\intercal(p_{\mathcal{T},0}, p_{\mathcal{V},0}) \bar{b}^{(i)} - \bar{b}^{(i)}||_2^2 \\ \qquad\qquad\qquad\qquad \text{if } p_{\mathcal{T},0} \in \left[0, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right], p_{\mathcal{V},0} \in \left[0, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right] \\ \infty \qquad\qquad\qquad\qquad \text{else} \end{cases}$$

$$g(p_{\mathcal{T},0}, p_{\mathcal{V},0}) = \begin{cases} ||A(p_{\mathcal{T},0}, p_{\mathcal{V},0})(A^\intercal(p_{\mathcal{T},0}, p_{\mathcal{V},0}) A(p_{\mathcal{T},0}, p_{\mathcal{V},0}))^{-1} A^\intercal(p_{\mathcal{T},0}, p_{\mathcal{V},0}) b - b||_2^2 \\ \qquad\qquad\qquad\qquad \text{if } p_{\mathcal{T},0} \in \left[0, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right], p_{\mathcal{V},0} \in \left[0, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right] \\ \infty \qquad\qquad\qquad\qquad \text{else} \end{cases}$$

Both g and the sequence of functions $\{g^{(0)}, g^{(1)}, \dots\}$ are level-bounded, lower semi-continuous and proper. By Lemma 20, $g^{(i)} \xrightarrow{e} g$ where $\xrightarrow{e}$ denotes convergence in epigraph. Thus, *residual* $= \min_{p_{\mathcal{T},0} \in \left[0, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right], p_{\mathcal{V},0} \in \left[0, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right]} g^{(t)}(p_{\mathcal{T},0}, p_{\mathcal{V},0}) \xrightarrow{p} \min_{p_{\mathcal{T},0}, p_{\mathcal{V},0}} g(p_{\mathcal{T},0}, p_{\mathcal{V},0})$ [75]. We know at least one perfect solution $g(p_{\mathcal{T},0}, p_{\mathcal{V},0}) = 0$

exists, that this solution is unique (by Assumption 2) and that this solution is in $\left\{0, \frac{1}{n_{\mathcal{T}}}, \ldots, \frac{n_{\mathcal{T}}-1}{n_{\mathcal{T}}}\right\} \mathcal{X} \left\{0, \frac{1}{n_{\mathcal{V}}}, \ldots, \frac{n_{\mathcal{V}}-1}{n_{\mathcal{V}}}\right\}$. Thus, $p^*_{\mathcal{T},0} \xrightarrow{p} p_{\mathcal{T},0}$, $p^*_{\mathcal{V},0} \xrightarrow{p} p_{\mathcal{V},0}$ and *residual* $\xrightarrow{p} 0$. □

Assumption 2. *b is independent of the columns of $A(p_{\hat{\mathcal{T}},0}, p_{\hat{\mathcal{V}},0})$ (except, obviously, at $p_{\hat{\mathcal{T}},0}, p_{\hat{\mathcal{V}},0} = p_{\mathcal{T},0}, p_{\mathcal{V},0}$). This is a very weak assumption when choosing $m >> n_{\mathcal{T}} n_{\mathcal{V}}$. It is unlikely the loss vector b happens to fall in the column space of A.*

Lemma 20. *$g^{(i)} \xrightarrow{e} g$, where we use $\xrightarrow{e}$ to denote convergence in epigraph.*

Proof. Let $x = (p_{\mathcal{T},0}, p_{\mathcal{V},0})$. Then the proof follows exactly from the proof of Lemma 16. □

6.2.5 Connection to Bézier curves and Bernstein polynomials

The B3 estimator in Eq. (6.2) has close ties to the Bernstein basis and Bézier curves. Notice that each column of A corresponds to a Bernstein basis function evaluated at at $p_0, \ldots, 1$. Thus, the B3 estimator is equivalent to solving for the Bernstein coefficients or Bézier control points e, where the system is constructed through the B3's bootstrapping process.

To clearly define the connection, recall that a Bernstein basis of degree n is defined as

$$b_{j,n}(x) = \binom{n}{j} x^j (1-x)^{n-j} \quad j = 0, \ldots, n \tag{6.7}$$

and that this forms a basis for polynomials at most degree n. Then the Bernstein polynomial is defined as

$$B_n(x) = \sum_{j=0}^{n} \beta_j b_{j,n}(x) \tag{6.8}$$

where B_j are the Bernstein coefficients. The B3 estimator $b_i = \sum_{j=0}^{n} e_j A_{ij}$ is equivalent to solving for the Bernstein coefficients $e_j = \beta_j$, where the Berstein basis is $A_{ij} = b_{j,n}(p_i)$.

Bézier curves are closely related to Bernstein polynomials, using slightly different notation

$$B(t) = \sum_{j=0}^{n} \binom{n}{j} t^j (1-t)^{n-j} \mathbf{P}_j \tag{6.9}$$

$$= \sum_{j=0}^{n} b_{j,n}(t) \mathbf{P}_j \tag{6.10}$$

where $\mathbf{P}_j$ are the Bézier control points. Once again, A_{ij} from the B3 estimator is equivalent to the Bernstein basis function $b_{j,n}(p_i)$, and we solve for the Bézier control points $\mathbf{P}_0, \ldots, \mathbf{P}_n$

6.3 Scalability techniques

The B3 estimator is limited by its need to solve a linear system of n variables, where n is the size of the bootstrap training set [9]. Solving the linear system has $\mathcal{O}(n^3)$ cost, and forming the loss estimate b has $\mathcal{O}(n)$ computational cost. If the prediction algorithm f has an expensive training procedure (e.g. deep neural networks), the latter term may outweigh the former due to a large fixed constant. In this section, we present two approaches for dramatically improving the computational efficiency of these estimators.

6.3.1 Basis Function Approximation

Perhaps the most straightforward approach to scaling these estimators is through function approximation, which also conveniently provides a natural form of regularization. We parameterize e by a set of s basis functions $\psi_1, \ldots, \psi_s$, such that

$$e_i = \xi_1 \psi_1(i) + \xi_2 \psi_2(i) + \ldots + \xi_s \psi_s(i) \tag{6.11}$$

where $\xi_1, \ldots, \xi_s = \xi \in \mathbb{R}^s$ are the s parameters. Then $e = \Psi \xi$ where $\Psi \in \mathbb{R}^{(n+1)\times s}$ is the matrix of basis values.

Instead of solving the linear system $Ae = b$, where $A \in \mathbb{R}^{m\times(n+1)}$ and we choose $m \geq n$, we can now solve

$$A' \Psi \xi = b' \tag{6.12}$$

where $A' \in \mathbb{R}^{m' \times (s+1)}$ and we choose $m' \geq s$. Note the size of this system no longer depends on the number of samples n. Instead, it depends on the number of parameters in our approximation of e, which will be a fixed constant. This new linear system is well behaved, depending on the choice of basis function ψ.

Theorem 21. *Let $\psi_0, \dots, \psi_s$ be a set of s unisolvent, bounded and continuous functions over $[0,1]$ and let*

$$e_i = \xi_0 \psi_0 \left(\frac{i}{n} \right) + \ldots + \xi_s \psi_s \left(\frac{i}{n} \right).$$

Then $A\Psi$ is invertible as $n \to \infty$.

Proof.

$$(A\Psi)_{ij} = \underset{k_n \sim \text{Binomial}(n, p_i)}{\mathbb{E}} \psi_j \left(\frac{k_n}{n} \right)$$

By the weak law of large numbers, $\frac{k_n}{n} \xrightarrow{p} p_i$. Further, $\psi_j \left(\frac{k_n}{n} \right) \xrightarrow{p} \psi_j(p_i)$ by the continuous mapping theorem. Finally, $\mathbb{E}\psi_j \left(\frac{k_n}{n} \right) \rightsquigarrow \mathbb{E}\psi_j(p_i) = \psi_j(p_i)$ by the Portmanteau lemma. The matrix formed by $\psi_j(p_i)$ is invertible by the Unisolvence theorem when $p_0, \dots, p_s$ are unique. □

6.3.2 Matrix Sketching

Second, we propose a new matrix sketching technique which reduces the number of columns in the structured matrix A. Unlike typical matrix sketching techniques, which reduce the number of rows, we are able to reduce the number of columns and thus the dimensionality of the solution e by leveraging the structure in A and properties of e from Theorem 9. After reducing the number of columns, one could further apply standard matrix sketching techniques to also reduce the number of rows. Our algorithm guarantees recovering e_0 within a linear factor of the true value.

Consider the setting where $m \leq n$ and the system $Ae = b$ is underdetermined. This is especially relevant for large datasets, where it is computationally infeasible to sample at $m > n$ levels of leakage or perhaps even solve for n unknowns. Let $S \in \mathbb{R}^{m \times (k+1)}$, $m > k$ be our sketching matrix, where S is formed such that the first column of S equals the first column of A, i.e. $S_0 = A_0$. Partition the remaining n columns of A into k sets, for example using k-medoids or simply grouping adjacent columns together (since by the definition of A, these will be close together). Let $r : \{0, \dots, n\} \to \{0, \dots, k\}$ be the resulting partition, where $r(0) = 0$ is the singleton partition of the first column. Finally, form the remaining columns of S from the medoids of the $k+1$ sets. Each column in A is within an ϵ-ball of at least one column in S, i.e.

$$\epsilon = \max_{i \in \{0,\dots,n\}} \|A_i - S_{r(i)}\|$$

Theorem 22. *Let e' be the solution to the sketched system $Se' = b$ and s be the first row of s^{-1}. The error between the true and sketched solution is bounded by*

$$|e'_0 - e_0| \leq \epsilon n \|s'\| e_0. \tag{6.13}$$

Proof. Let e' be the solution to the sketched system $Se' = b$. Then

$$\begin{aligned} Se' &= Ae = b \\ e' &= S^{-1}Ae \\ e'_0 &= (S^{-1}A)_{00}e_0 + \sum_{i=1}^{n}(S^{-1}A)_{0i}e_i \\ e'_0 - e_0 &= \sum_{i=1}^{n}(S^{-1}A)_{0i}e_i. \end{aligned}$$

Let s' be the first row of S^{-1}. By the Cauchy-Schwarz inequality, for all $i \geq 1$

$$\begin{aligned} |(S^{-1}A)_{0i}| &= |s' \cdot A_i| \\ &= |s' \cdot A_i - s' \cdot S_{r(i)}| \\ &\leq \|s'\|\|A_i - S_{r(i)})\| \\ &= \epsilon\|s'\|. \end{aligned}$$

Finally, by Theorem 9

$$\begin{aligned}|e_o' - e_0| &\leq \sum_{i=1}^{n} |(S^{-1}A)_{0i}| e_i \\ &\leq \sum_{i=1}^{n} \epsilon \|s'\| e_i \\ &\leq \epsilon n \|s'\| e_0.\end{aligned}$$

□

6.4 Simulation study

Thus far, we have appealed to asymptotic theory and bias-variance analysis. This is not uncommon for bootstrap and cross-validation analysis, and like others, we now turn to empirical arguments. In this section, we present simulation study results which demonstrate our core method in Algorithm 1 significantly outperforms conventional methods. For all experiments, we consider the more difficult direction where samples move from $\mathcal{V}$ to $\mathcal{T}$.

For comparison, we consider two benchmark estimators for the OOC loss – IID and LOCO. IID is the typical cross-validation split, where samples are uniformly randomly split into training and validation sets, which does not account for the latent clustering. LOCO is the leave-one-cluster-out estimator described in Eq. (5.1) using an approximated clustering $\hat{c}$ with an error of $p_0 = 0.1$.

Our estimators are unbiased and consistent, but they may have large variance (see Theorem 14). When practically implementing these estimators, it is beneficial to add a small amount of regularization to achieve a better bias–variance tradeoff. Although we know from Lemma 13 and Lemma 17 that matrix A is full rank, it may be ill-conditioned. Adding regularization helps to improve the condition number of matrix A. Evidence suggests this is a tradeoff worth making. Specifically, in the linear system objective function within Algorithm 1 and Algorithm 4 we instead solve some variation of

$$\begin{aligned} \underset{\hat{e}}{\text{minimize}} \quad & ||A\hat{e} - \bar{b}||_2^2 + \lambda R(\hat{e}) \\ \text{subject to} \quad & \hat{e}_{j-1} \geq \hat{e}_j \geq 0 \quad \text{for all } j = 1, \dots, n'. \end{aligned} \tag{6.14}$$

where λ is a regularization constant and R is some regularization function. We choose the trend filter regularizer $R(\hat{e}) = ||D\hat{e}||_2^2$ to ensure $\hat{e}$ is smooth [43]. For a second-order filter, which regularizes the second derivative of $\hat{e}$, D is the difference matrix

$$D = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{bmatrix}$$

where unshown entries are zero. Matrices D for higher order trend filters follow similarly. From Theorem 9, the estimator error to degrade both monotonically (the constraint) and we expect it to also degrade somewhat smoothly (the regularizer). Methods with various order trend filters and constraints are denoted as second, third or fourth-order trend filter (T2, T3, T4) with or without a monotonic constraint (+mono).

Experimental details are the same as Section 5.5, with the following additional parameter choices for the estimators. In the sketching approximation, we formed k nearly equally sized groups of adjacent columns from A when forming the sketched matrix S. Even after sketching, we found it beneficial to add some regularization comparable to T4+mono, referred to as λ_s (the regularization used in T4+mono is referred to as λ_{T4}). We found that other approaches, including using k-medoids to group the columns of A, did not provide any benefits and were more complicated. In all experiments we set $k = 7$.

In the basis function approximation, we found that using simple, low-order polynomials was sufficient. Higher order polynomials tended to be unstable. After observing b, we chose to use either a 2nd or 7th order polynomial, depending on the curvature of b.

The complete set of experimental parameters are shown in Table 6.1. We made an

Table 6.1: Parameters used in all experiments. n is the number of samples in the training set, $|\mathcal{V}|$ is the number of samples in the validation set, t is the number of resamples in Algorithm 1, λ's are the regularization strengths in the T4+mono and sketching method, m is the number of corruption levels (i.e. the number of rows in A), k is the number of sketching groups and d is the number of features in the dataset.

	Dataset				
Parameter	Synthetic	Heart[2]	1994 US Census[3]	Parkinson[4]	Dota 2[5]
n	∞	100	100	100	100
$\lvert\mathcal{T}\rvert$	15	100	100	100	1000
$\lvert\mathcal{V}\rvert$	1000	100	100	100	100
t	1000	1000	10000	1000	1000
λ_{T4}	0.1	10	10	1000	1000
λ_s	0.01	0.1	0.1	0.1	0.1
s	7	7	7	2	2
m	30	200	200	20	20
k	10	20	20	20	20
d	2	12	5	26	114

effort to limit fitting to a specific dataset, and kept most parameters the same across all experiments. In the Dota 2 experiments, the availability of sufficient training data allowed us to increase $|\mathcal{T}|$ to 1000. Further, after completing the Heart and 1994 US Census experiments, we reduced the number of rows m in A by an order of magnitude to speed up experimentation, and correspondingly increased the regularization λ.

6.4.1 Results

First, we demonstrate the proposed methods with various forms of regularization are significantly less biased than the baseline IID and LOCO methods. Our main results, presented in Fig. 6.2, are generated over 10 independent trials, where the whiskers correspond to most extreme values over those trials (i.e. no outliers removed).

LOCO outperforms traditional IID cross-validation, which suggests blocking on

[2] https://archive.ics.uci.edu/ml/datasets/heart+Disease

[3] https://archive.ics.uci.edu/ml/datasets/adult

[4] https://archive.ics.uci.edu/ml/datasets/Parkinson+Speech+Dataset+with++Multiple+Types+of+Sound+Recordings

[5] https://archive.ics.uci.edu/ml/datasets/Dota2+Games+Results

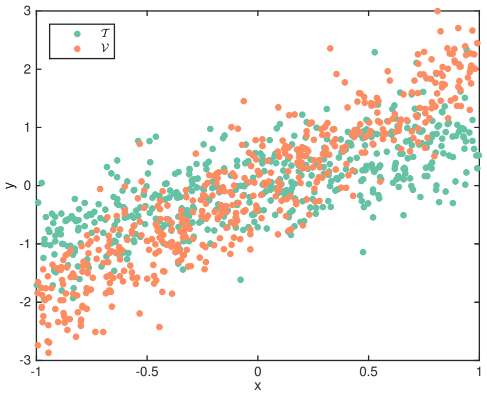


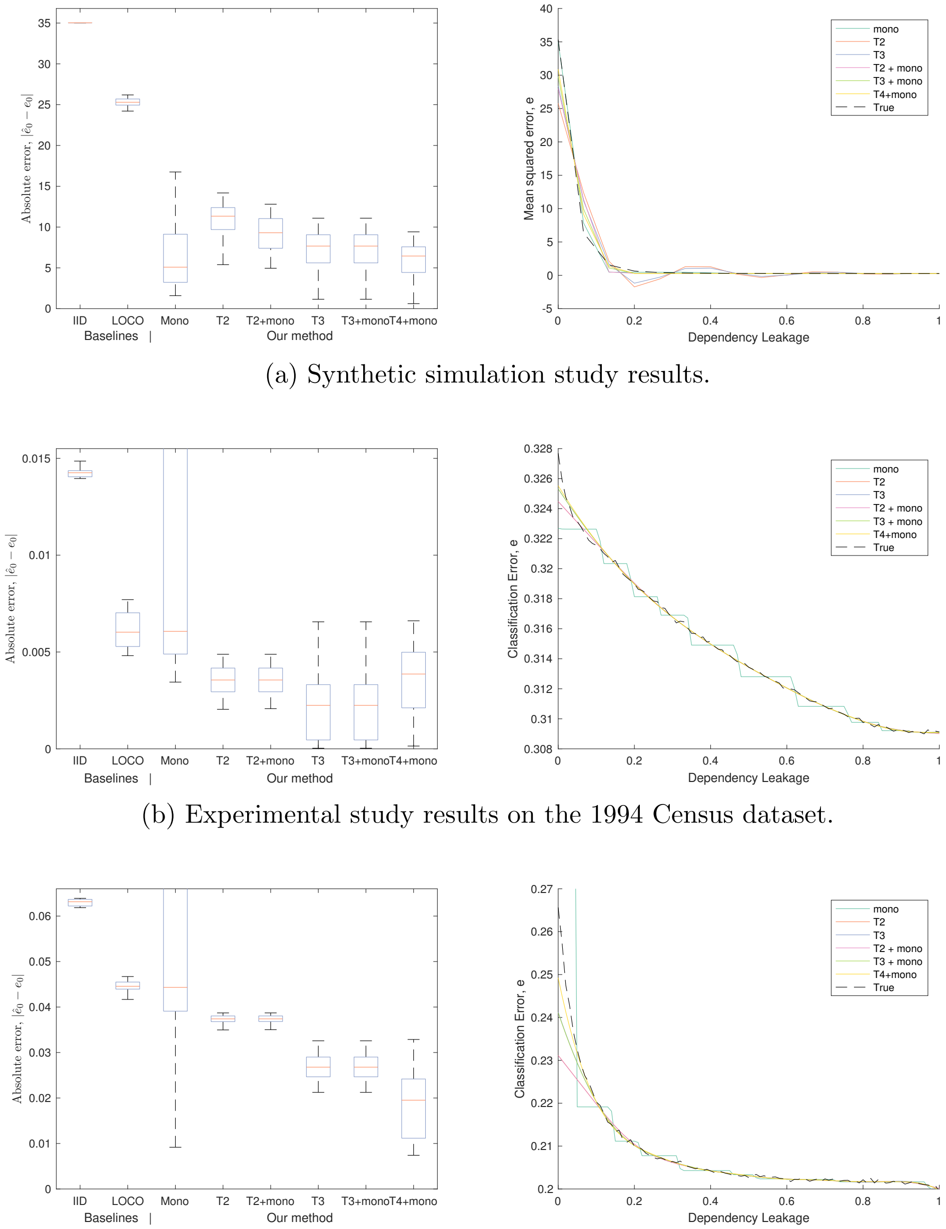
Figure 6.1: Subsample of data used in the synthetic Experiment I.

the corrupted clusters $\hat{c}$ partially limits the effects of dependency leakage. However, even at $p_0 = 0.1$, LOCO is still unacceptably biased. Our methods, with various forms of regularization, all significantly outperform both existing estimators. Fig. 6.2a also suggests a bias-variance trade-off among all the tested methods. IID cross-validation has high bias and low variance, whereas our methods have low bias and higher variance. Ultimately, this tradeoff allows our methods to achieve lower MSE by choosing an appropriate form and strength of regularization.

An interesting consequence of our method is that in addition to recovering the independent partition performance e_0, we also recover the performance $e_1, e_2, \ldots$ at all levels of dependency leakage, as depicted in Fig. 6.2a. The true loss e (dashed black line) decays monotonically and smoothly, which justifies our regularization choices.

6.4.2 Computational Scalability

The proposed approximation techniques, and especially the basis function approximation technique, are faster than existing OOC estimators and are tractable on larger problem classes. To compare performance across a large range of dataset sizes, we generated increasingly large synthetic training sets and compared solution times in Section 6.4.1. All methods used only 10 corruption levels (i.e. the number of rows in A), the bare minimum required to find a reasonable solution. We observed that increasing the number of rows in A exponentially increased solution times. Thus, these

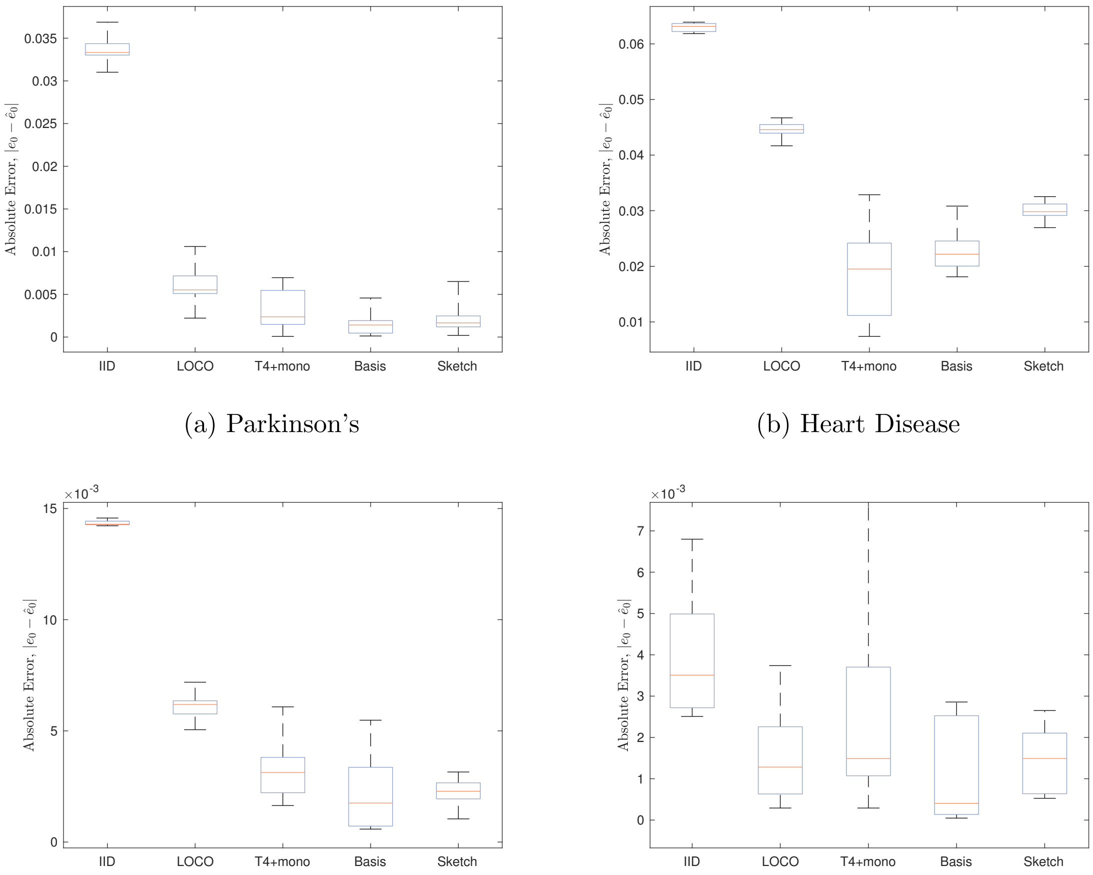


(a) Synthetic simulation study results.

(b) Experimental study results on the 1994 Census dataset.

(c) Experimental study results on the heart disease dataset.

Figure 6.2: **Left** Estimating the generalization loss e_0. Our class of B3 estimators, with various forms of regularization (monotonic; second, third or fourth-order trend filter) outperform existing estimators. Baseline cross-validation methods are biased against the sub-populations we studied, and our class of B3 estimators help correct this bias. **Right** The B3 estimator recovers the full loss vector e. Empirically, the true loss decays monotonically and smoothly in practice, justifying our regularization choices.



(a) Parkinson's

(b) Heart Disease

(c) 1994 US Census

(d) Dota 2

Figure 6.3: Estimating the OOC loss e_0. Our function approximation and novel matrix sketching techniques perform comparably to existing methods at significantly reduced computational cost.

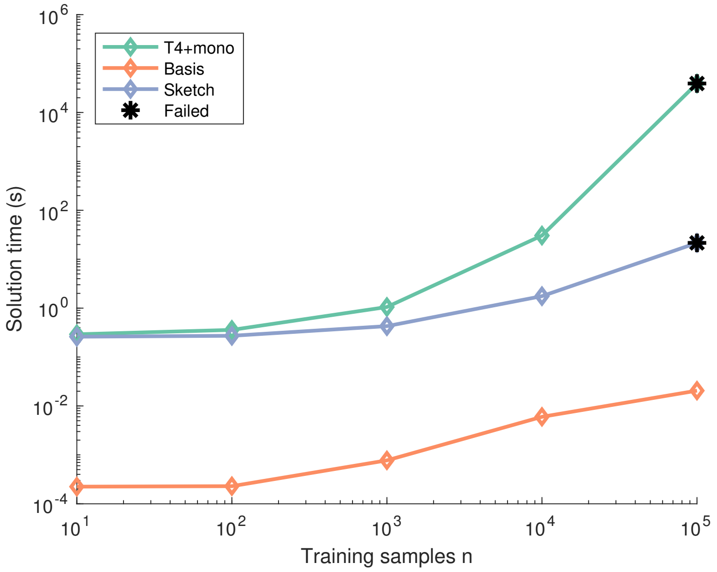


Figure 6.4: Computational scalability results on synthetically generated datasets. Our methods (Sketch, and in particular, Basis) are significantly faster than existing methods (T4+mono). "Failed" indicates the SDPT3 solver failed to find an accurate solution.

results are likely the largest datasets appropriate for existing methods. In particular, notice that the solver failed to find accurate solutions on the largest problem class for all methods except for with the basis approximation technique.

Timing results on real world datasets (described in the following sections) are reported in Table 6.2. Similarly, we find the basis approximation technique is the fastest by several orders of magnitude.

Constrained linear programs (e.g. T4+mono, sketching) were solved using SDPT3's infeasible path-following algorithm, for unconstrained linear systems we took advantage of fast QR solvers (a major reason the basis method is so efficient). All optimizations were performed using an Xeon Gold 6152 CPU @ 2.10GHz and 754 GB RAM. We found that T4+mono, and to a lesser extent, the sketching approximation, required the majority of this memory for the largest problem classes.

Table 6.2: Computational timing results demonstrate our methods, and in particular the basis function approximation technique, are significantly faster than the previous state-of-the-art B3 estimator with fourth order trend filter and monotonicity constraint (T4+mono). Results shown in seconds.

	Method		
Dataset	T4+mono	Sketching	Basis
1994 US Census	0.5662	0.4059	**7.822e-5**
Heart	0.5847	0.4105	**6.582e-5**
Parkinson's	0.6194	0.4338	**2.043e-5**
Dota 2	1.0965	0.4678	**1.946e-5**

6.4.3 Statistical Scalability

The need for regularization, either in the form of a trend filter (Eq. (6.14)) or basis function approximation (Eq. (6.12)), is obvious upon investigating the condition number of the regression coefficient matrix (either A or $A\Psi$). Fig. 6.5 shows that the condition number of matrix A (B3 estimator) degrades as the number of training samples n increases. On the other hand, the basis function approximation $A\Psi$ dramatically improves the condition number, which in fact decreases with respect to n.

A visual representation of the two matrices A and $A\Psi$ in Section 6.4.3 and Section 6.4.3, respectively, show that A is an off-diagonal band matrix, where the top is shifted according to p_0. The matrix $A\Psi$ is relatively constant with respect to n.

6.5 Extensions

This work poses several additional questions, some of which we briefly address now. For example, we have extended these methods from estimating the expected loss e to estimating an expected loss histogram E in Eq. (6.2). To do so, one can simply store the empirical bootstrap histogram $\bar{B}$ in lieu of the empirical bootstrap mean $\bar{b}$. The downside is estimating the additional information in E increases the variance by a linear factor according to the number of histogram bins.

To improve the numerical solution in Algorithm 1 in the direction where samples

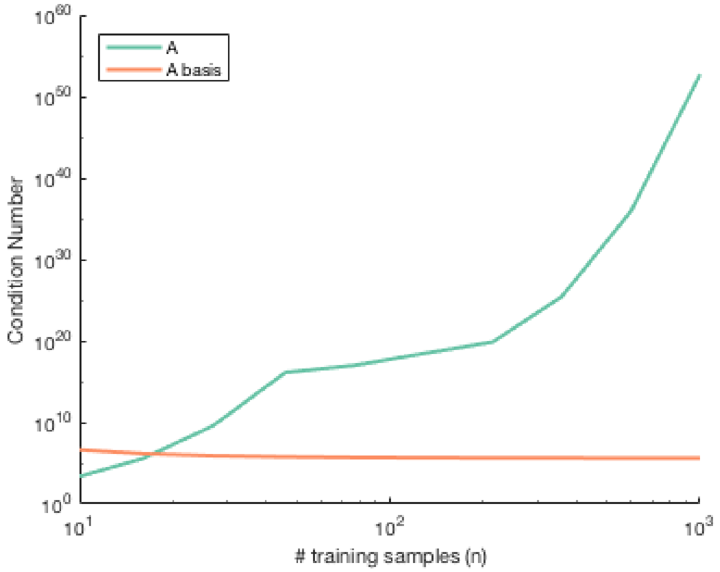
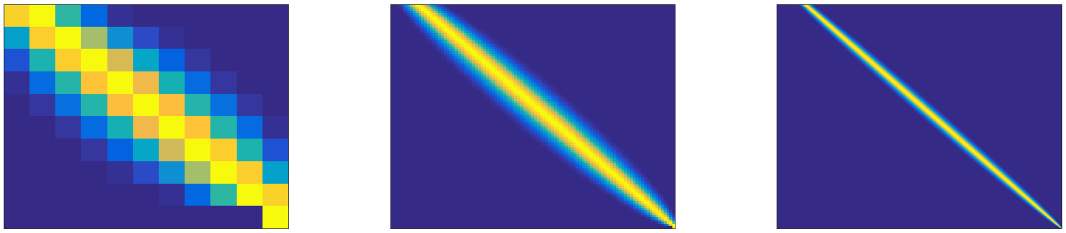


Figure 6.5: Condition number of matrix A (B3 estimator) degrades as the number of training samples n increases. The basis function approximation $A\Psi$ dramatically improves the condition number, which in fact improves with respect to n.

move from $\mathcal{T}$ to $\mathcal{V}$, note that e will be a linear vector, i.e. $e_{i+1} - e_i = \beta \ \forall i \in \{0, \ldots, n' - 1\}$. This is because the training set $\mathcal{T}'$ has zero corruption, the expected number of corrupted samples in $\hat{\mathcal{V}}$ varies linearly with p_i for fixed δ, and the empirical loss is a mean loss of the samples in $\hat{\mathcal{V}}$. Enforcing this constraint on $\hat{e}$ would improve the solution quality for the direction where samples move from $\mathcal{T}$ to $\mathcal{V}$. We always considered the more difficult $\mathcal{V}$ to $\mathcal{T}$ leakage direction, where we have no prior knowledge of e.

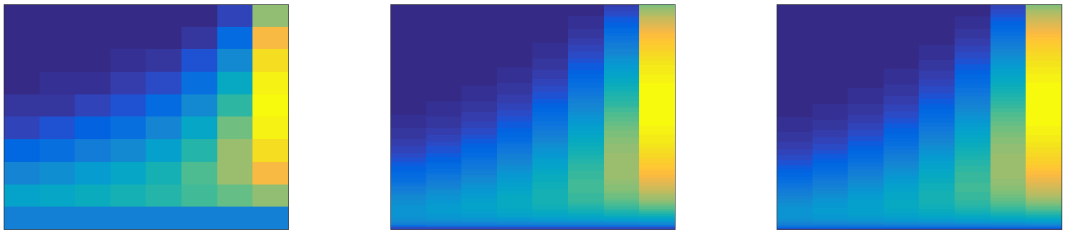
The question of unbalanced clusters for CRVE was addressed in [57]. In our cross-validation method, small $\mathrm{Var}(|\hat{\mathcal{T}}|)$ and $\mathrm{Var}(|\hat{\mathcal{V}}|)$ across the cross-validation folds improves convergence. With unbalanced clusters, instead of leaving one cluster out, we could leave multiple clusters out such that $|\hat{\mathcal{T}}|$ and $|\hat{\mathcal{V}}|$ have lower variance even with high variance cluster sizes. CRVE also suffers from having a small number of clusters k [16]. Our estimator will be nearly unbiased but have high variance with a small number of clusters, due to the same properties as LOCO (see Eq. (5.1)).

Though we have shown asymptotic convergence of our methods, there are several open questions. Notably, we use a naive discrete optimization routine in Algorithm 2



(a) $n = 10$ (b) $n = 100$ (c) $n = 1000$

Figure 6.6: Visualization of matrix A. B3 estimator matrix A resembles an off-diagonal band matrix, where the top is shifted according to p_0



(a) $n = 10$ (b) $n = 100$ (c) $n = 1000$

Figure 6.7: Visualization of matrix $A\Psi$. Applying the basis function matrix Ψ changes the structure of regression coefficient matrix.

and Algorithm 4 to solve for $p_{\mathcal{T},0}$ and $p_{\mathcal{V},0}$. The functions $g(i)(p_0)$ are non-convex, but they are smooth with finite support and faster convergence may be possible.

6.6 Conclusions

In this chapter, we addressed the issue of evaluating a learner on blocks of dependent data. Unlike existing bootstrap methods, which assume a perfect clustering, we allow for imperfect clusterings $\hat{c}$ such that inter-cluster samples may be dependent. Real world applications ranging from medical diagnostics to computer vision fall into this class of problems. Empirical evidence on synthetic data, the 1994 US Census and heart disease data shows dependency leakage biases cross-validation results and thus affects model selection. We presented the B3 class of estimators, which significantly outperform existing cross-validation methods in this setting. The key insight of our bootstrapping methods is that by injecting additional dependency, we can extrapolate an unbiased and asymptotically consistent estimator of the performance on independent clusters.

The contents of this chapter were presented at the 2017 Conference on Uncertainty in Artificial Intelligence (UAI) [9].

Chapter 7

Conclusions

The two primary questions this thesis attempted to answer were:

1. When do clustering algorithms perform poorly?
2. How can we incorporate imperfect clustering results into machine learning systems?

To that end, we derived lower and upper error bounds to answer the first question, across varied methodologies and data distributions. Perhaps more excitingly, we were able to provide conclusive properties and solutions to the second question, which had previously been entirely unaddressed. Together, we found that addressing the first question (Part I) eased the difficulty of solving the second (Part II).

Understanding clustering performance is a long standing challenge in the machine learning community, because unlike standard supervised learning problems, clustering is a difficult problem to clearly articulate a "correct" objective function and equally difficult to evaluate. In Part I, three different approaches to clustering were theoretically analyzed. In Chapter 2, clustering was formulated as a maximum likelihood estimator over a stochastic block model, and shown to be reducible to a correlation clustering algorithm with error upper bounds. For small problems with access to a small set of labeled pairs, this approach is computational feasible and outperforms baselines in empirical analysis. In Chapter 4, we took a Bayesian perspective on clustering, allowing us to construct tight error lower bounds for both categorical and string data. This approach is particular useful for record linkage problems, where we understand the data generation process and require uncertainty estimates. Finally, we provided error upper bounds for the class of match-and-merge algorithms such as Swoosh [11], demonstrating that the most conservative merge function successfully minimizes the worst-case. This attempted to address a dangerous aspect of the no-negative-evidence clause (required to guarantee determinism) – match mistakes compound and quickly cause massive clusters. Thus, this approach remains best suited for low-noise settings with few records.

In Part II of this thesis, we formalized the problem of clustering errors which propagate through downstream machine learning components and cross-validation estimators. This is a serious concern in medical, census, shopping and the counter-human-trafficking domains, where clustering is used as a preprocessing step to merge samples or records corresponding to the same person or product. Our empirical results illustrate these errors can have dangerous, unforeseen impacts on the overall system performance. Theorems 9 and 11 tell us that the largest interaction effect is always caused by the first clustering error, and that additional clustering errors only continue to degrade performance. These effects are particularly dangerous because they are undetectable by standard cross-validation techniques, and not realizable until deploying the system into an online environment with potentially serious consequences.

To alleviate these concerns, we introduced the Binomial Block Bootstrap (B3) estimator in Chapter 6, which estimates the cross-validation bias caused by clustering errors. In practical medical diagnostics (including heart disease and Parkinson's disease), US Census and Dota2 game data, the B3 estimator consistently provides better generalization error estimates than standard cross-validation techniques with clustering errors. Scalability techniques using basis function approximation and matrix sketching techniques enable deploying our estimator to training datasets with millions of samples.

7.1 Practical suggestions

Thus far, we've provided an array of tools for clustering and learning on clusters of data. Based on the analysis of these tools and our anecdotal experiences, we suggest a series of actionable steps when confronted with a new dataset and initial clustering. Depending on the problem context (e.g. dataset size, labeling cost and budget), these suggestions may be more or less feasible, but in rough order of priority we recommend:

1. Conducting a simple hypothesis test for whether dependency leakage is an issue in your problem, as described in Section 5.4. This requires training and validating your prediction algorithm a handful of times, and computing the two sample t-test statistic.
2. Using feature selection and any relevant domain knowledge to remove features which do not generalize to new clusters (e.g. a person's name or phone number).
3. Investing in computing a more accurate clustering, e.g. using the methods described in Part I. In particular, consider collecting labeled pairs of samples for use in a supervised clustering method. It is a common misconception that clustering is strictly an unsupervised method.
4. Collecting additional data in the form of more clusters, which may be able to prevent the prediction algorithm from overfitting to cluster-specific features.
5. Using simpler or more strongly regularized models which are not able to overfit to cluster-specific features.
6. Tending to undersegment your data. This effectively reduces the value of p_0, although it may introduce additional forms of bias since the train/validation split is no longer random among the true latent clusters.
7. Using the basis function approximation estimator and visualize the solution curve e to qualitatively measure overfitting, choose an appropriate basis (e.g. polynomial order) and regularizer.

Some domains, including robotics and lab based scientific studies, are condusive to conducting new experiments which may guarantee a disjoint set of clusters. For example, image instance segmentation datasets collected at different times and locations will certainly contain new object instances.

7.2 Future Directions

There are several interesting directions which this work opens for exploration. In Part I, we introduced well-founded clustering methods with strong empirical performance. Improving the computational scalability would significantly lower the barrier to more widespread adoption of these methods. Bayesian inference, (Chapter 4) including MCMC and variational inference, struggle with the massive space of linkage structures. Recent work on distributed Bayesian inference for record linkage problems has leveraged assumptions on the linkage structure (e.g. a blocking scheme) to allow distribution across multiple machines and reduce the computational complexity [90]. Likewise, the stochastic block model in Chapter 2 used an LP-rounding technique to solve MinimizeDisagreements, which requires a large number of constraints due to the number of pairwise edges. General purpose approximate LP solvers may partially address this problem, although we believe it is possible to leverage additional structure in the graph in a similar fashion to the scalable Bayesian record linkage approaches.

In Part II, we assumed that clustering errors were uniformly random – as many clustering algorithms do not provide a measure of uncertainty. However, if a clustering algorithm is able to provide an estimate of its assignment uncertainty (e.g. Bayesian methods), then this uncertainty may be properly incorporated into the B3 estimator. We believe this is possible to do using a form of importance sampling. Second, although we were able to improve the computational scalability in Section 6.3, some of these approximations invalidate many of our theoretical guarantees.

Finally, we demonstrated the ability to estimate cross-validation bias due to clustering errors. In Section 7.1, we alluded to adjusting the dataset via feature selection to reduce this bias. We pose the question, “Is it possible to learn which samples were incorrectly clustered?” This would complete a feedback loop between the clustering algorithm and an error correction mechanism. Some recent work has attempted to learn a mapping from related but different distributions to a common feature space which has low divergence for each distribution, yet maintains predictive performance [31, 56, 93]. It may be possible to extend these results to our setting, where samples have flipped between each distribution due to clustering errors. This would be equivalent to learning dataset transformations with zero resulting dependency leakage.

In conclusion, we theoretically and empirically demonstrated that the current practice of incorporating clustering algorithms into systems without explicit regard for how their errors propagate through downstream pipeline components can quickly cause dangerous and initially undetectable consequences. This is a growing concern as the complexity of machine learning systems and the problems they address increases. Here, we made contributions towards both understanding when clustering algorithms perform poorly, and perhaps more novelly, characterizing and correcting for the interaction effects between clustering errors and the larger system.

Appendix A

Open source B3 implementation

An open source MATLAB implementation of the methods described in Chapter 5 and Chapter 6 is provided at `https://github.com/mbarnes1/B3`. All processed datasets and links to the original UCI downloads are provided in the repository. Details for reproducing the results in this thesis are provided in the corresponding `README.md` file, and included below.

This repository includes an implementation of the B3 estimator with various approximation techniques, example datasets and reproducible results. Below is a walk-through on the Dota2, Parkinson, Census Income, Synthetic, and Heart Disease datasets.

A.0.1 Finite sample estimates

In practice, the most computational expensive step is often computing a finite sample approximation to vector b. This requires repeatedly repeatedly resampling the training dataset, training and validating the learner, and injecting additional leakage from the validation set. Script `sample_real.m` performs this resampling procedure and saves the resulting estimates.

All required datasets are included in the `data/` folder. We also include example results in `bootstraps/`.

A.0.2 Solving the linear system

Once the resampling procedure is complete, we turn to computing the solution vector e (of which, the first entry e_0 is our desired out-of-cluster loss). The key choices here are using

- An appropriate form of regularization (monotonic constraint, and/or a trend filter)
- An approximation technique (basis function or sketching).

Note the basis function approximation serves as a natural form of regularization, and does not require additional regularizers. The script `example_boxplot.m` reproduces the results in Fig. 6.3a, demonstrating the B3 estimator outperforms baseline methods using various forms of approximation techniques. Other results can be reproduced by using the appropriate bootstrap results in `bootstraps/`.

Bibliography

[1] Christopher Aicher, Abigail Z. Jacobs, and Aaron Clauset. Adapting the stochastic block model to edge-weighted networks. *arXiv preprint*, 2013. URL http://arxiv.org/abs/1305.5782. 2.1.1

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information. *Journal of the ACM*, 55(5):1–27, 2008. ISSN 00045411. doi: 10.1145/1411509.1411513. 2.1.2

[3] E Airoldi, D Blei, S Fienberg, and E Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008. 2.1.1

[4] Michael K Andersson and Sune Karlsson. Bootstrapping error component models. *Computational Statistics*, 16(2):221–231, 2001. ISSN 0943-4062. 6.1.1

[5] Sivaraman Balakrishnan, Min Xu, Akshay Krishnamurthy, and Aarti Singh. Noise thresholds for spectral clustering. *Advances in Neural Information Processing Systemsn*, pages 1–9, 2011. URL http://papers.nips.cc/paper/4342-noise-thresholds-for-spectral-clustering. 2.1.1

[6] Brian Ball, Brian Karrer, and M. E J Newman. Efficient and principled method for detecting communities in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 84(3):1–14, 2011. ISSN 15393755. doi: 10.1103/PhysRevE.84.036103. 2.1.1

[7] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749, 2005. 4.1

[8] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation Clustering. *Machine Learning*, 56(1-3):89–113, 2004. ISSN 08856125. doi: 10.1023/B:MACH.0000033116.57574.95. 2.1.2

[9] Matt Barnes and Artur Dubrawski. The Binomial Block Bootstrap Estimator for Evaluating Loss on Dependent Clusters. *UAI*, 2017. 5.1, 6.3, 6.6

[10] Thomas R. Belin and Donald B. Rubin. A Method for Calibrating False-Match Rates in Record Linkage. *Journal of the American Statistical Associ-*

ation, 90(430):694–707, 1995. ISSN 0162-1459. doi: 10.1080/01621459.1995.10476563. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476563. 3.1

[11] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal – The International Journal on Very Large Data Bases*, 18(1):255–276, 2009. ISSN 1066-8888. 3.1, 3.2, 3.5.2, 3.6, 7

[12] Daniel Berend, Peter Harremoës, and Aryeh Kontorovich. Minimum KL-divergence on complements of l_1 balls. *IEEE Transactions on Information Theory*, 60(6):3172–3177, 2014. 4.3.1, 4.3.2

[13] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5, 2007. ISSN 1556-4681. 3.6

[14] Andrei Z. Broder. On the resemblance and containment of documents. *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 1–9, 1997. ISSN 0818681322. doi: 10.1109/SEQUEN.1997.666900. 3.5.2

[15] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992. ISSN 0891-2017. 3.4

[16] R. C. Cameron and Douglas L. Miller. A Practitioner's Guide to Cluster-Robust Inference. *Journal of Human Resources*, 50(2):317–372, 2015. ISSN 0022-166X. doi: 10.3368/jhr.50.2.317. URL http://jhr.uwpress.org/content/50/2/317.refs. 6.1.1, 6.5

[17] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with Qualitative Information. In *Foundations of Computer Science*, 2003. 2.1.2, 2.5

[18] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002. ISBN 1581134959. 3.5.2

[19] D. S. Choi, P. J. Wolfe, and E. M. Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2):273–284, 2012. ISSN 00063444. doi: 10.1093/biomet/asr053. 2.1.1

[20] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012. 4

[21] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Structural Inference of Hierarchies in Networks. *Statistical Network Analysis: Models, Issues, and New Directions.*, 2007. ISSN 03029743. doi: 10.1007/978-3-540-73133-7{_}1.

URL http://arxiv.org/abs/physics/0610051. 2.1.1

[22] Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001. ISSN 10429832. doi: 10.1002/1098-2418(200103)18:2⟨116::AID-RSA1001⟩ 3.0.CO;2-2. 2.1.1

[23] J. Copas and F.J. Hilton. Record linkage: Statistical models for matching computer records. *Journal of the Royal Statistical Society, Series A*, 153(3): 287–320, 1990. 4.1

[24] J J Daudin, F Picard, and S Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, 2008. ISSN 09603174. doi: 10.1007/s11222-007-9046-7. URL http://link.springer.com/10.1007/s11222-007-9046-7$\delimiter"026E30F$nfile:///Files/E4/E46CEF6F-D709-4C02-9B5B-27C39E000CA8.pdf$\delimiter"026E30F$npapers3://publication/doi/10.1007/s11222-007-9046-7. 2.1.1

[25] Anthony Christopher Davison and David Victor Hinkley. *Bootstrap methods and their application*, volume 1. Cambridge University Press, 1997. ISBN 0521574714. 6.1.1

[26] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361:172–187, 2006. ISSN 03043975. doi: 10.1016/j.tcs.2006.05.008. 2.1.2, 2.3, 2.3, 2.3, 2.5

[27] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006. 4.1

[28] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014. ISSN 0893-6080. 3.3, 3.5.2

[29] Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969. ISSN 0162-1459. doi: 10.1080/01621459.1969.10501049. 3.1, 3.2

[30] C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 69(3):369–390, 2007. ISSN 13697412. doi: 10.1111/j.1467-9868.2007.00593.x. 6.1.1

[31] Yaroslav Ganin and Victor Lempitsky. Proceedings of the 32nd International Conference on Machine Learning. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. ISBN 9781510810587. doi: 10.1016/0022-2364(84)90100-8. 7.2

[32] Lise Getoor and Ashwin Machanavajjhala. Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 5(12):2018–2019, 2012. ISSN 2150-8097. 3.6

[33] M X Goemans and D P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. ISSN 00045411. doi: 10.1145/227683.227684. URL papers2://publication/uuid/0892EDC7-FD65-421F-AE12-FB02C133ABEA. 2.1.2

[34] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A Survey of Statistical Network Models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010. ISSN 1935-8237. doi: 10.1561/2200000005. URL http://dx.doi.org.libproxy1.nus.edu.sg/10.1561/2200000005$\delimiter"026E30F$nhttp://arxiv.org/pdf/0912.5410v1.pdf. 2.1.1

[35] Larry Greenemeier. Human Traffickers Caught on Hidden Internet. *Scientific American*, 2015. URL http://www.scientificamerican.com/article/human-traffickers-caught-on-hidden-internet/. 3.5.1

[36] R. Gutman, C. Afendulis, and A. Zaslavsky. A Bayesian procedure for file linking to analyze end- of-life medical costs. *Journal of the American Statistical Association*, 108(501):34–47, 2013. 4.1

[37] Peter Hall. Resampling a coverage pattern. *Stochastic Processes and their Applications*, 20(2):231–246, 1985. ISSN 03044149. doi: 10.1016/0304-4149(85)90212-1. 6.1.1

[38] Jake M. Hofman and Chris H. Wiggins. Bayesian approach to network modularity. *Physical Review Letters*, 100(25):1–4, 2008. ISSN 00319007. doi: 10.1103/PhysRevLett.100.258701. 2.1.1

[39] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983. ISSN 03788733. doi: 10.1016/0378-8733(83)90021-7. 2.1.1

[40] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. Efficient Direct Density Ratio Estimation for Non-stationarity Adaptation and Outlier Detection. *Advances in Neural Information Processing Systems*, 2009. 2.4

[41] Brian Karrer and Mark EJ Newman. Stochastic block models and community structure in networks. *Physical Review*, 2011. 2.1.1

[42] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998. ISSN 0004-5411. 6.1.2

[43] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. L1 Trend Filtering. *Society for Industrial and Applied Mathematics (SIAM)*

Review, 51(2):339–360, 2009. URL http://www.optimization-online.org/DB{_}FILE/2007/09/1791.pdf. 6.4

[44] George Kollios, Michalis Potamias, and Evimaria Terzi. Clustering large probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):325–336, 2013. ISSN 10414347. doi: 10.1109/TKDE.2011.243. 2.1.2

[45] Hanna Köpcke and Erhard Rahm. Training selection for tuning entity matching. In *QDB/MUD*, pages 3–12, 2008. 3.5.1

[46] Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, 69(2):197–210, 2010. ISSN 0169-023X. 3.5.1

[47] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. 4.1

[48] Soumendra Nath Lahiri. *Resampling methods for dependent data*. Springer Science & Business Media, 2003. ISBN 147573803X. 6.1.1

[49] Michael D Larsen and Donald B Rubin. Iterative Automated Record Linkage Using Mixture Models. *Journal of the American Statistical Association*, 96 (453):32–41, 2001. ISSN 0162-1459. doi: 10.1198/016214501750332956. URL http://www.tandfonline.com/doi/abs/10.1198/016214501750332956. 3.1

[50] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46 (6):787–832, 1999. ISSN 00045411. doi: 10.1145/331524.331526. 2.1.2, 2.3

[51] Kung-Yee Liang and Scott L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986. 6.1.1

[52] Moshe Lichman. UCI Machine Learning Repository, 2013. URL http://archive.ics.uci.edu/ml. 5.5, 5.5, 5.5

[53] B. Liseo and A. Tancredi. Some advances on Bayesian record linkage and inference for linked data. *Technical Report*, 2013. 4.1

[54] Regina Y. Liu and Kesar Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the Limits of Bootstrap*, pages 225–248. Wiley-Interscience, 1992. 6.1.1

[55] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016. ISSN 0162-8828. 6.1.2

[56] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. 7.2

[57] James G. MacKinnon and Matthew D. Webb. Wild Bootstrap Inference for

Wildly Different Cluster Sizes. *Journal of Applied Econometrics*, 32:233–254, 2017. ISSN 10991255. doi: 10.1002/jae.2508. 6.1.1, 6.5

[58] Nathaniel Macon and Abraham Spitzbart. Inverses of Vandermonde matrices. *The American Mathematical Monthly*, 65(2):95–100, 1958. 6.2.2

[59] Mahendra Mariadassou, Stephane Robin, and Corinne Vacher. Uncovering latent structure in valued graphs: A variational approach. *Annals of Applied Statistics*, 4(2):715–742, 2010. ISSN 19326157. doi: 10.1214/10-AOAS361. 2.1.1

[60] Andrew McCallum and Ben Wellner. Conditional Models of Identity Uncertainty with Application to Noun Coreference. *Advances in Neural Information Processing Systems 17*, pages 905–912, 2005. 3.1

[61] Frank McSherry. Spectral Partitioning of Random Graphs. *Proceedings 2001 IEEE International Conference on Cluster Computing*, pages 529–537, 2001. ISSN 02725428. doi: 10.1109/SFCS.2001.959929. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=959929`. 2.1.1

[62] David Menestrina, Steven Euijong Whang, and Hector Garcia-Molina. Evaluating entity resolution results. *Proceedings of the VLDB Endowment*, 3(1-2): 208–219, 2010. ISSN 2150-8097. 3.3, 3.5.1, 3.6

[63] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICRL)*, 2013. URL `http://arxiv.org/abs/1301.3781`. 3.4

[64] Stanley Milgram. The Small-World Problem. *Society*, 39(2):61–66, 1967. ISSN 01472011. doi: 10.1007/BF02717530. 2.1.1

[65] Douglas Miller, A. Cameron, and Jonah Gelbach. Robust Inference with Multiway Clustering. *Journal of Business & Economic Statistics*, 29(2), 2011. ISSN 2282-4189. 6.1.1

[66] Jeffrey Miller, Brenda Betancourt, Abbas Zaidi, Hanna Wallach, and Rebecca Steorts. The Microclustering Problem: When the Cluster Sizes Don't Grow with the Number of Data Points. *NIPS Bayesian Nonparametrics: The Next Generation Workshop Series*, 2015. URL `http://arxiv.org/abs/1512.00792`. 4.5

[67] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013. 6.1.2

[68] H B Newcombe, J M Kennedy, S J Axford, and a P James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959. ISSN 0036-8075. doi: http://dx.doi.org/10.1126\%252Fscience.130.3381.954. 3.1

[69] Howard B. Newcombe and James M. Kennedy. Record linkage: making maximum use of the discriminating power of identifying information. *Communications of the Association for Computing Machinery*, 5(11):563–566, 1962. ISSN 00071447. 3.1

[70] Xinghao Pan, Dimitris Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Parallel Correlation Clustering on Big Graphs. *Advances in Neural Information Processing Systems*, 2015. URL http://arxiv.org/abs/1507.05086. 2.1.2, 2.5

[71] Georgios Papadakis. *Blocking Techniques for efficient Entity Resolution over large, highly heterogeneous Information Spaces*. PhD thesis, Leibniz Universität Hannover, 2013. 3.3.1, 3.5.1

[72] Yongjin Park, Cristopher Moore, and Joel S. Bader. Dynamic networks from hierarchical Bayesian graph clustering. *PLoS ONE*, 5(1), 2010. ISSN 19326203. doi: 10.1371/journal.pone.0008118. 2.1.1

[73] Vyacheslav Valer'evich Prelov and Edward C. van der Meulen. Mutual information, variation, and fano's inequality. *Problems of Information Transmission*, 44(3):185–197, 2008. 4.3.1, 4.3.1

[74] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Charles Florin, Luca Bogoni, and Linda Moy. Learning From Crowds. *Journal of Machine Learning Research*, 11: 1297–1322, 2010. 6.1.2

[75] R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, 2009. ISBN 9783540627722. doi: 10.1021/jp7118845. URL http://books.google.com/books?id=w-NdOE5fD8AC. 6.2.3, 6.2.4

[76] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39(4):1878–1915, 2011. ISSN 00905364. doi: 10.1214/11-AOS887. 2.1.1

[77] Mauricio Sadinle. Detecting duplicates in a homicide registry using a Bayesian partitioning approach. *The Annals of Applied Statistics*, 8(4):2404–2434, 2014. 4.1

[78] Mauricio Sadinle and Stephen E. Fienberg. A Generalized FellegiSunter Framework for Multiple Record Linkage With Application to Homicide Record Systems. *Journal of the American Statistical Association*, 108(502):385–397, 2013. ISSN 0162-1459. doi: 10.1080/01621459.2012.757231. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.2012.757231. 3.1

[79] Sohrab Saeb, Luca Lonini, Arun Jayaraman, David C Mohr, and Konrad P Kording. Voodoo Machine Learning for Clinical Predictions. *bioRxiv*, 2016. doi: 10.1101/059774. 5.1

[80] D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. Machine Learning : The High-Interest Credit Card of Technical Debt. *NIPS 2014 Workshop on Software Engineering for Machine Learning (SE4ML)*, pages 1–9, 2014. ISSN 13613723. doi: 10.1007/s13398-014-0173-7.2. 5.1, 5.6

[81] Claude E Shannon. A note on the concept of entropy. *Bell System Tech. J*, 27: 379–423, 1948. 4.1

[82] J Shi and J Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 888–905, 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL http://www.computer.org/portal/web/csdl/doi?doc=abs/proceedings/cvpr/1997/7822/00/78220731abs.htm$\delimiter"026E30F$npapers3://publication/uuid/268FC197-AF47-4C7C-887F-BEDB94A81320. 2.4

[83] Kesar Singh. On the Asymptotic Accuracy of Efron's Bootstrap. *The Annals of Statistics*, 9(6):11877–1195, 1981. ISSN 00905364. doi: 10.1214/aos/1176348654. URL http://projecteuclid.org/euclid.aos/1176345976. 6.1.1

[84] Parag Singla and Pedro Domingos. Entity resolution with markov logic. In *Sixth IEEE International Conference on Data Mining*, pages 572–582. IEEE, 2006. ISBN 1550-4786. 3.1, 3.6

[85] Martha E Smith and H B Newcombe. Methods for computer linkage of hospital admission-separation records into cumulative health histories. *Methods of information in medicine*, 14(3):118–125, 1975. ISSN 0026-1270. 3.1

[86] R. C. Steorts. Entity resolution with empirically motivated priors. *Bayesian Analysis*, 10(4):849–875, 2015. 4, 4.2, 4.2.1, 4.2.3, 4.3, 4.3.2, 4.4, 4.4.1, 4.5

[87] R. C. Steorts, R. Hall, and S. E. Fienberg. SMERED: A Bayesian approach to graphical record linkage and de-duplication. *Journal of Machine Learning Research*, 33:922–930, 2014. 4, 4.1, 4.2, 4.2.1, 4.2.1, 4.3, 4.3.1, 4.4, 4.4.1, 4.5

[88] R. C. Steorts, R. Hall, and S. E. Fienberg. A Bayesian approach to graphical record linkage and de-duplication. *Journal of the American Statistical Society*, In press. 4, 4.1, 4.2, 4.2.1, 4.2.1, 4.3.1, 4.4, 4.4.1, 4.5

[89] Rebecca C. Steorts. Foundations of Data Science - The small clustering problem: When the cluster sizes dont grow with the data, 2015. URL http://research.microsoft.com/apps/video/default.aspx?id=249219. 1, 3.3

[90] Rebecca C. Steorts. Personal communication, 2018. 7.2

[91] Rebecca C. Steorts, Rob Hall, and Stephen E. Fienberg. SMERED: A Bayesian Approach to Graphical Record Linkage and De-duplication. *Proceedings of the 17th International Con- ference on Artificial Intelligence and Statistics*

(AISTATS), 33:1–39, 2014. URL http://arxiv.org/abs/1403.0211. 5.1

[92] Rebecca C. Steorts, Matt Barnes, and Willie Neiswanger. Performance Bounds for Pairwise Record Linkage. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017. URL http://arxiv.org/abs/1509.03302. 4.5

[93] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016. 7.2

[94] Chaitanya Swamy. Correlation Clustering: maximizing agreements via semidefinite programming. *SODA*, pages 526–527, 2004. URL http://dblp.uni-trier.de/db/conf/soda/soda2004.html{#}Swamy04. 2.1.2, 2.5

[95] A. Tancredi and B. Liseo. A hierarchical Bayesian approach to record linkage and population size problems. *Annals of Applied Statistics*, 5(2B):1553–1585, 2011. 4.1

[96] Sheila Tejada, Craig A Knoblock, and Steven Minton. Learning object identification rules for information integration. *Information Systems*, 26(8):607–633, 2001. ISSN 0306-4379. 3.5.1

[97] Andrew C. Thomas and Joseph K. Blitzstein. Valued Ties Tell Fewer Lies: Why Not To Dichotomize Network Edges With Thresholds. *arXiv preprint*, 2011. URL http://arxiv.org/abs/1101.0788. 2, 2.1.1

[98] Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph Cluster Randomization: Network Exposure to Multiple Universes. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 329–337, 2013. doi: 10.1145/2487575.2487695. URL http://dl.acm.org/citation.cfm?id=2487695. 6.2.2

[99] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J Belongie. Learning From Noisy Large-Scale Datasets With Minimal Supervision. In *CVPR*, pages 6575–6583, 2017. 6.1.2

[100] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. ISSN 09603174. doi: 10.1007/s11222-007-9033-z. URL http://www.springerlink.com/index/JQ1G17785N783661.pdf. 2, 2.3.1, 2.4

[101] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11): 1483–1494, 2012. ISSN 2150-8097. 3.5.1, 3.6

[102] Yuchung J Wang and George Y Wong. Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987. ISSN 01621459. doi: 10.2307/2289119. URL http://www.jstor.org/stable/

2289119. 2.1.1

[103] Halbert White. *Asymptotic theory for econometricians*. Academic Press, 1984. ISBN 1483294420. 6.1.1

[104] Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. ISSN 0162-1459. 3.5.3

[105] W. E. Winkler. Overview of record linkage and current research directions. Technical report, U.S. Bureau of the Census Statistical Research Division, 2006. 4

[106] William E Winkler. Methods for record linkage and bayesian networks. Technical report, 2002. 3.1

[107] William E. Winkler. Overview of record linkage and current research directions. Technical report, U.S. Census Bureau, 2006. 3.1, 5.1

[108] William E. Winkler and Yves Thibaudeau. An application of the Fellegi-Sunter model of record linkage to the 1990 US decennial census. Technical report, U.S. Census Bureau, 1990. URL https://www.census.gov/srd/papers/pdf/rr91-9.pdf. 5.1

[109] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015. 6.1.2

[110] Yan Xiaoran, Cosma Rohilla Shalizi, Jacob E Jensen, Cristopher Moore, Lenka Zdeboro, Pan Zhang, and Yaojia Zhu. Model Selection for Degree-corrected Block Models. *Journal of Statistical Mechanics: Theory and Experiment*, 5, 2014. ISSN 1742-5468. doi: 10.1088/1742-5468/2014/05/P05007. 2.1.1

[111] Wayne W Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, 1977. ISSN 00917710. doi: 10.2307/3629752. URL http://www.maths.tcd.ie/{~}mnl/store/Zachary1977a.pdf. 2.1.1

[112] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems 17*, 2:1601–1608, 2004. doi: 10.1.1.84.7940. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.84.7940. 2, 2.4.2