Active Duplicate Detection with Bayesian Nonparametric Models

by

Nicholas Elias Matsakis

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

Author .... [illegible]
Department of
Electrical Engineering and Computer Science
September 17, 2009

Certified by .... [illegible]
Leslie Pack Kaelbling
Professor
Thesis Supervisor

Accepted by .... [illegible]
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Active Duplicate Detection with Bayesian Nonparametric Models

by

Nicholas Elias Matsakis

Submitted to the Department of
Electrical Engineering and Computer Science
on September 17, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

When multiple databases are merged, an essential step is identifying sets of records that refer to the same entity. Called *duplicate detection*, this task is typically tedious to perform manually, and so a variety of automated methods have been developed for partitioning a collection of records into coreference sets. This task is complicated by ambiguous or noisy field values, so systems are typically domain-specific and often fitted to a representative labeled training corpus. Once fitted, such systems can estimate a partition of a similar corpus without human intervention.

While this approach has many applications, it is often infeasible to encode the appropriate domain knowledge *a priori* or to identify suitable training data. To address such cases, this thesis uses an active framework for duplicate detection, wherein the system initially estimates a partition of a test corpus without training, but is then allowed to query a human user about the coreference labeling of a portion of the corpus. The responses to these queries are used to guide the system in producing improved partition estimates and further queries of interest.

This thesis describes a complete implementation of this framework with three technical contributions: a domain-independent Bayesian model expressing the relationship between the unobserved partition and the observed field values of a set of database records; a criterion for picking informative queries based on the mutual information between the response and the unobserved partition; and an algorithm for estimating a minimum-error partition under a Bayesian model through a reduction to the well-studied problem of *correlation clustering*. It also present experimental results demonstrating the effectiveness of this method in a variety of data domains.

Thesis Supervisor: Leslie Pack Kaelbling
Title: Professor

Dedication

This thesis is dedicated to my wife Terri, who has taught me the virtue of patience through her example. Without her tireless support and sacrifice over the winding and frequently bumpy road that has led to this point, this work would never have gotten off the ground, much less been completed.

Acknowledgments

I am greatly indebted to so many people who have provided me unwavering aide and support in this pursuit. In particular to my mother and father, Joanne and Elias Matsakis, and Terri's parents, Carlyn and Harold Iuzzolino, for their encouragement and financial support throughout my graduate career. I am also forever grateful to my advisor and mentor, Leslie, who was willing to take a chance on me and has seen this work through to the very end. It was only with her guidance and support that I was able to turn a disordered collection of half-baked ideas into a doctoral thesis.

I also owe thanks to my committee, Michael Collins, Tomás Lozano-Pérez, and Andrew McCallum; for their ability to tell me when I wasn't making sense, their encouragement when I was, and the thoughtful and considered feedback they provided in preparation for my defense and during the writing of this thesis. To Dan Roy, for his willingness to share his knowledge of Bayesian nonparametrics and to understand what I was trying to do better than myself at times. To David Karger, for sparking my interest in this problem and his support during my early years of graduate school.

To my friends Mike Ross and Victor Luchangco, for their many years of friendship and especially for their willingness to read this document in development and suggest innumerable improvements. Also to my friends Markéta and Joe Foley, for keeping me sane. To all the wonderful students in CSAIL and LIS, especially John Barnett, Jenny Barry, Emma Brunskill, Sam Davies, Sarah Finney, Natalia Hernandez Gardiol, Kaijen Hsiao, Meg Aycinena Lippow, James McLurkin and Luke Zettlemoyer. They have been my friends and teachers, providing many hours of delightful conversation and insightful critique. I only wish I could have been as helpful to them as they have been to me.

Contents

List of Figures

Chapter 1

Introduction

Most of us really aren't horribly unique. There are 6 billion of us. Put 'em all in one room and very few would stand out as individuals.

Herbert Simon

1.1 Motivation and Problem Description

Managers of large data repositories have long been familiar with the problem of duplicate detection. Database software is typically designed under the assumption that there is a single record for each entity of interest. The presence of duplicate, or *coreferent*, records violates this assumption and can result in problems when using the data. For example, duplicates may displace canonical records in search results, result in misleading computed statistics, and use system resources unnecessarily.

In some cases it is possible to avoid creating duplicates in the first place. For example, library catalogers use the Library of Congress and other *authority services* to provide canonical records of works [39]. However, such services are of limited use when a library is cataloging a new work or author, as is frequently the case in academic libraries [82]. In such cases, judicious inference by the cataloger can reduce the incidence of duplicate records.

However, there are a variety of scenarios where the problem cannot be avoided, such as when merging multiple databases, when authority services don't exist, or when identifying traits of entities are inaccessible at record creation time. For example, the use of data-mining algorithms to extract structured data from unstructured text is increasingly common. In this case, the records are created automatically and thus some manner of duplicate detection must be employed.

Even for modest-sized databases, it is typically impractical to manually review each record to find coreferents. For this reason, a variety of automated methods have been developed to address this problem. The earliest systems used hand-tuned *linkage rules* that specified conditions under which a pair of records should be declared coreferent [65]. In these early systems, resources were limited and record access had high latency, so even applying simple pairwise rules was challenging.

Duplicate detection was first formalized in a decision-theoretic framework by Fellegi and Sunter [37], who treated the problem in the statistical hypothesis testing framework, developing a binary classifier that used the agreement of a record pair's field values to identify coreferent pairs or *matches*. This framework modeled the common application of finding matched pairs from two clean *sources*.

problem statement

In general, there may be many sources and many coreferent records within a source, and the problem is that of finding a *partition* of a set of records into a set of *coreference blocks*, or disjoint record sets that refer to the same entity.[1] For methods that describe a class of models for finding partitions, such as supervised methods that fit a model to training data, we refer to a particular instance of that class as a *partitioner*, analogous to a classifier in the classification problem.

A broad variety of approaches have been taken to finding a suitable partitioner for a data environment. We can categorizes these broadly into two groups. Pairwise methods have been developed based on supervised classifiers such as the Naive Bayes model [93], support vector machines [19], and decision trees [88]. Partition-based methods include

[1]Some authors use the terms *partitioning* and *partition* for what we define here as a *partition* and *block* respectively.

generative Bayesian models [67, 14], conditional random fields [59, 66], constraint propagation models [34], and heuristic algorithms [54].

One aspect common to these approaches is that the partitioner must be tailored to the data to which it is applied. In the case of hand-tuned rules, a human expert must determine which rules are appropriate. Even with machine-learning methods, optimal performance typically requires either encoding knowledge into a domain-specific model or choosing appropriate preprocessing steps, domain-specific features and functions, and hand-labeling training examples.

Bilenko and Mooney [20] provide an insightful look into the importance of choosing appropriate training data for duplicate detection algorithms. They compared a supervised model trained on data from the test domain (publications) to one trained on a disparate domain (restaurants), and also to a generic similarity model based on string edit distance. While the appropriately trained model performed best, the generic model outperformed the one trained on inappropriate data.

sources of data variation

One explanation for these results is that there is an ambiguity in explaining the presence of variation in a data corpus. One source, which we call *population variation*, reflects the true variation in the underlying entities and should be used as evidence for coreference decisions. Another source of variation, which we call *observation variation*, is an artifact of the data collection process, and should be disregarded as noise. When faced with a generic collection of data, it cannot be known *a priori* how to account for the observed variation. Domain-specific models and/or training examples serve as constraints that resolve this ambiguity.

An example of this phenomenon is found in the well-studied case of duplicate detection using personal names. In this case, the frequency of particular names and the prevalence of name changes may be considered part of the population variation, while data entry errors, spelling inconsistencies, and the use of nicknames factor into the observation variation. This distinction between signal and noise is not always clear. For example, when an individual with a common name adopts an uncommon nickname as a means of distinction, an algorithm that normalizes known nicknames in preprocessing will eliminate a distinguishing feature. Thus, a variety of rich statistical models have been developed to account for the intricacies of personal names [92, 67, 26, 23, 14].

Another aspect of the problem highlighted by the work of Bilenko and Mooney is that generic similarity models are capable of correctly labeling most of a problem instance, as superficial similarity is often a good indicator of coreference. The problem's difficulty lies along the margin of "somewhat similar" records, which may or may not be duplicates. It is here where the use of domain-specific models and appropriate training examples can significantly improve performance by taking advantage of the characteristics of a particular domain or even the particular data set within that domain.

1.2 Active Duplicate Detection

While it is clear that domain-specific knowledge is required for optimal performance, it is not always feasible to encode such knowledge *a priori*, either through building domain-specific models or through providing labeled training examples, because the necessary work cannot be amortized across a large data corpus. This is particularly true for personal information databases, where users unskilled in database management informally collect information according to their own idiosyncrasies, often without adherence to a particular schema. Such data may also have peculiar statistics; for instance, we expect the distribution of names in a personal contact database to have a very different distribution from that in the population at large.

To address such data environments, this thesis develops a domain-independent active framework for duplicate detection. In this framework, there is no distinction between training and evaluation data: the system is presented with an unlabeled data corpus from an unknown domain, but is allowed to query a human supervisor about the coreference labels of portions of the data. The goal is to find the best coreference labeling for a given amount of supervisor effort.

Ideally, the query responses serve both to label the "hard" portion of the corpus and to simultaneously resolve ambiguities specific to the corpus in order to improve performance on the remainder of the corpus. Although the supervisor may be labeling the difficult portion of the corpus, the system is still tasked with identifying this region and distinguishing between informative and uninformative queries. Note that human labeling does not require an expertise in data modeling or machine learning, but rather the domain knowledge of whether a pair of records is coreferent.

1.3 Summary of Contributions

The method described in this thesis divides active duplicate detection into three stages: *inference*, *partition estimation*, and *query ranking*. In the inference stage, a model of the uncertainty in the coreference partition is generated. This uncertainty model is used both in generating a single best estimate of the coreference partition and also in ranking queries to the supervisor. Labels provided by the supervisor may then be fed back into the inference stage to produce an updated uncertainty model. Figure 1-1 provides a diagram of the system.

This thesis describes an implementation of this framework with three technical contributions: a domain-independent Bayesian model of coreference, a criterion for picking informative queries based on maximizing mutual information, and an algorithm for estimating a minimum-error partition under a Bayesian model based on minimizing expected pairwise error. It also present experimental results demonstrating the effectiveness of this method in a variety of data domains.

1.3.1 Domain-Independent Bayesian Model

Chapter 3 presents a domain-independent generative Bayesian model of the coreference relation of a data corpus: a distribution over observed record values conditioned on the coreference partition of the corpus. Unlike many other Bayesian models for duplicate detection, it uses no schema-specific distributions or structures. Instead, it provides a generic model of record generation based on Bayesian nonparametric distribu-

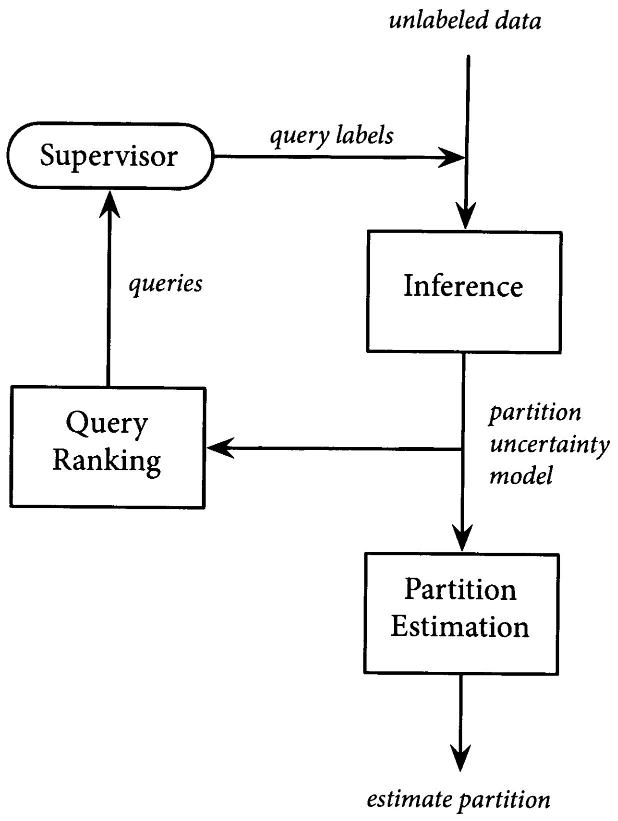


Figure 1-1: Domain-independent active duplicate detection

tions, which allow the model to be used in a variety of different domains without modification.

Two different variations of the model are presented: an *individual value distributions* (IVD) model in which each coreference block has a unique distribution over its field values and a *shared value distributions* (SVD) model in which the field-value generating distributions may be shared between coreference blocks. The distinction between these two models is discussed in detail in section 3.4.

Chpater 4 provides an efficient inference procedure for the model based on Markov Chain Monte Carlo (MCMC) sampling. The sampling procedure generates approximate samples of the posterior distribution over partitions given observed record labels. This sampling procedure can be used completely unsupervised, when no coreference labels are known, and also in the case in which the partition is subject to binary constraints; In the latter case, additional modifications are provided to meet the ergodicity condition of MCMC sampling.

1.3.2 Correlation Clustering for Minimizing Error

While a Bayesian model provides a model of the uncertainty associated with a hypothesis, it is still useful in many applications to produce a single estimate of an unobserved quantity; for instance, producing a single estimate partition from a Bayesian distribution over partitions. The *maximum a posteriori* (MAP) estimator is commonly used in Bayesian modeling.

This thesis proposes an alternative to the MAP estimate for duplicate detection: the estimate partition that minimizes the expected pairwise error under the model. Chapter 4 demonstrates that optimizing this objective under a Bayesian model reduces to *correlation clustering*, an NP-hard problem with a number of known approximation algorithms [33].

The minimum expected error objective function is used both because it provides a more direct measure of the error of an estimate partition and also because it works well in conjunction with the query ranking measure proposed in chapter 5: With this objective function, the same inference computation is used both to provide weights for correlation clustering and also to rank pairwise queries. Both of these depend on an estimate of the marginal probability that a pair of records is coreferent.

Though correlation clustering is NP-hard, the practical running time of a problem instance depends on the amount of ambiguity in the data; thus it may be solved exactly for practical data sets with small coreference blocks and chapter 4 provides a heuristic algorithm for estimating a partition from samples under this objective that runs quickly in practice.

1.3.3 Mutual Information Ranking for Active Queries

While the model may be used for unsupervised inference, it is intended to be used in conjunction with a human supervisor in an active labeling framework. However, in a typical duplicate detection application, an arbitrary pair of records is likely to be a trivial nonmatch and therefore not strong evidence about the true partition. We might expect that such weak evidence would not alter the results of inference and indeed this is one of the empirical findings presented in chapter 6. Therefore, in this framework it is necessary to use the model to pick informative queries.

Chapter 5 develops a formal framework for ranking queries in a Bayesian, partition-based framework. In this formalism, query responses are treated as unobserved variables and queries are ranked according to the *mutual information* between the query response and the partition as a whole. A derivation is given which shows that the query that optimizes mutual information is that with the highest posterior entropy in its label value.

This result leads to an estimate of query effectiveness that can be computed efficiently from MCMC samples. Query responses can then be incorporated into the model as observations, and additional inference conditions on these observations may produce improved partition estimates and further informative queries. This result also provides a theoretical connection between active approaches based on pairwise classifiers and those based on partition search.

1.4 Overview of Experimental Results

Chapter 6 presents experimental results from using this method in six different data domains. These experiments look at four practical considerations: the importance of incorporating negative constraints when merging data from multiple sources; the performance of the proposed query ranking criterion compared to baseline algorithms; the performance characteristics of the individual and shared distributions models; and the effect of periodically reranking queries through additional rounds of inference.

These results demonstrate the effectiveness of the proposed query ranking in improving the performance of the system over its unsupervised performance or with queries ranked using baseline algorithms. They also demonstrate the significance of feedback in the process: periodically reranking queries through additional inference, improves the performance of the system for a given number of queries.

These experiments also reveal limitations of the model, including its unsupervised performance on certain data sets and ability to scale to large data sets, discussed in section ??. These limitations are used as a basis for the suggestions of future research directions provided in chapter 7, including improved value distribution modeling, extending the model to handle relational data, and other issues involved in scaling the method to large data sets.

Chapter 2

Background

Automated duplicate detection has long history in the research literature. The problem was first called *record linkage*, since the intent was to link vital records—such as birth and death certficates and marriage licenses—in order to create a more comprehensive understanding of the data [65, 37]. Researchers have since introduced a seemingly combinatorial number of names for this problem including:[1]

> *instance identification* [91], *semantic integration* [2], *data cleansing* [36], *authority work* [39], *the merge/purge problem*[47], *data integration* [24], *citation matching* [54], *instance matching* [75], *reference matching* [60], *database hardening* [25], *entity-name clustering and matching* [27], *object identification* [88], *fuzzy duplicate elimination* [7], *deduplication* [79], *object consolidation* [61, 58], *identity uncertainty* [67], *object co-identification* [44], *robust reading* [56], *reference reconciliation* [34], *entity resolution* [12], *author disambiguation* [30], and *coreference resolution* [45].

This profusion of names is due in part to interest from different research communities, but also from a desire to highlight aspects of a particular

[1]This list is derived in part from similar ones compiled by William Winkler [94] and Mikhail Bilenko [16].

data environment or method. We use *duplicate detection* for its dissociation with any particular data environment and its alliterative quality.

This chapter provides a high-level overview of various approaches to the problem, categorized according to architecture, model-constructing framework, and data environment. These distinctions emphasize the contributions of this thesis by motivate the scenarios best suited to the domain-independent, active model presented in later chapters.

2.1 Data Environments

The need for duplicate detection typically arises in data collected to express knowledge about a real-world *domain* of entities and their relationships. One example is that of a library database, which contains records for the books and other works curated by the library. It may also contain records for the authors and publishers of the works, and for the library's patrons. In addition to the ubiquitous example of publications [39, 54, 67, 76, 13], the problem has been explored for such domains as vital records [65, 93], corporations [24, 87], films [24, 5], music [44], product listings [17], and biological data [53].

To be effectively processed by a computer, the information of the domain must be expressed in a *data model*, a machine-understandable format for data storage and retrieval. Common data models include relational databases (SQL), object databases [52], document-object models (e.g., XML), and graph databases [55, 29]. Such models are domain-independent: they provide features applicable to a wide variety of domains, such as primitive data types and text encodings.

schema structure

It is frequently desirable to restrict the data to conform to a *schema*, a formal description of the types and ranges of valid field values. Continuing the library example, a schema might dictate that library patron records must contain a field for the the patron's phone number, and restrict the field's value to a valid phone number. Schemas are used to validate the database, facilitate data interchange, and prevent application errors due to invalid data. Such data is often called *structured* data.

Schemas are not essential, however, and some data models [90, 29] allow data conforming to disparate schemas, or to no schemas at all, to intermingle. Such data is often called *semi-structured* [55]. Schemas may also vary in the strictness of their typing. For example, most personal information applications allow users to enter free-form text strings as values and typically do not require a value for every field. Consequently, such data is subject to a greater degree of variation than data that is strictly validated, as is common in business applications.

relational structure

Another aspect in which data environments vary is in the degree of relational structure present in the data. In *flat* databases, each record represents the same type of entity and the field values are primitive types such as text strings or numbers. For example, most personal address databases contain records corresponding to individual people, with textual fields for the person's contact information.

Flat databases hinder the sharing of information between records. In a flat address database, for example, there is no means to express that an address is shared by multiple people, and therefore the street address, city, and zip code values should be the same. Likewise, it may not be possible to model situations in which an individual is associated with multiple addresses. To handle these and many other use cases, many data models admit some form of *relational* values, wherein records are permitted to have fields referring to other records. With relational values, it is possible to explicitly model the many-to-many relation between people and addresses.

Simple relational data environments have a small fixed number of record types, but deeply relational models may allow for an arbitrary number of types with an arbitrary number of relations. This poses a challenge to duplicate detection in these environments because coreference decisions between records must be propagated to all related records.

2.2 Duplicate Detection Architectures

Duplicate detection methods can be broadly categorized into *pairwise methods*, which develop a coreference classifier for pairs of records, and *partition-based methods*, which optimize a scoring function over partitions of the records. However, this distinction is not firm and many methods incorporate aspects of both architectures, for example by using a pairwise classifier as a component in a partition score.

2.2.1 Pairwise Methods

Pairwise methods were first developed to model the application of finding matches between two files or *data sources*. In this situation, each record is associated with a source that is presumed to be free of coreferents, as is often the case when databases must be merged. Thus coreference blocks consist of either a single record or a pair of records, one from each source. These methods may be extended to data without such a constraint through the use of a *linkage algorithm*, which combines the possibly inconsistent pairwise decisions into a final partition.

A common approach is to partition the records in three stages. First, a similarity function is used to score pairs of records along a real-valued scale indicating the level of confidence in a match. Next, a threshold is used to decide which pairs are matched. Finally, a linkage algorithm is applied to generate a final partition. These stages are not always performed sequentially; an iterative approach may be taken where the similarity function, threshold, and linkage algorithm are adjusted until a termination condition is reached [13].

pairwise features

Typical pairwise classifiers require record pairs to be represented in a feature space. A basic representation would be a vector of binary values, one per field, indicating whether the field values match in the pair, but more elaborate features are typically used. These may combine attributes of multiple fields into a real number. Domain-dependent features are one way of incorporating domain knowledge regarding relevant fields. Parametric features with learned parameters may also be used [18].

Once a pair of records (i, j) is expressed as a feature vector x_{ij}, supervised learning techniques can be applied to generate a classifier from labeled examples (x_{ij}, y_{ij}), with y_{ij} indicating whether the pair is a match. Some techniques applied to duplicate detection include decision trees [88], maximum entropy classifiers [27], support vector machines [19], rule induction [53], and voted perceptron [17]. Supervised learning theory may also be applied to problems such as feature selection and classifier overfitting.

hypothesis testing framework

In their classic presentation, Fellegi and Sunter [37] cast the classification problem as a likelihood ratio test of the form,

$$\frac{P(x;\, Y)}{P(x;\, \overline{Y})} \underset{\text{nonmatch}}{\overset{\text{match}}{\gtrless}} T$$

which requires estimating the two likelihood functions as well as a suitable threshold T for the decision boundary. Fellegi and Sunter additionally show that by using two thresholds, between which records were classified for manual review, the amount of manual review for a given error rate could be minimized. Winkler [93] presents an Expectation-Maximization algorithm for learning the likelihood parameters in a naïve Bayes likelihood model.

thresholding and linkage

The threshold value, or margin in the case of a discriminative classifier, must be tuned to the number of coreferent pairs in a test set. If the threshold is set too high or too low, there will be many misclassified pairs. An appropriate setting can be found using a training corpus, provided that the frequency of coreference is representative of an evaluation corpus; otherwise, it may need to be tuned when the classifier is applied to new data.

Since a pairwise classifier operates on pairs of records, it may produce inconsistent (i.e., nontransitive) decisions. Consider the simple example of three records, u, v, and w. A classifier might decide that pairs (u, v) and (v, w) are matches, but that pair (u, w) is a nonmatch, producing a linkage graph as in Figure 2-1. A linkage algorithm is used to resolve this inconsistency. Often, the classifier scores are used directly by the linkage algorithm and thresholding is applied as part of linkage.

The simplest algorithm, called *single linkage*, computes the transitive closure of the match decisions. In the case of the previous example, single linkage would declare u, v, and w coreferent. Single linkage may be viewed as a greedy hierarchical agglomerative clustering algorithm [17],

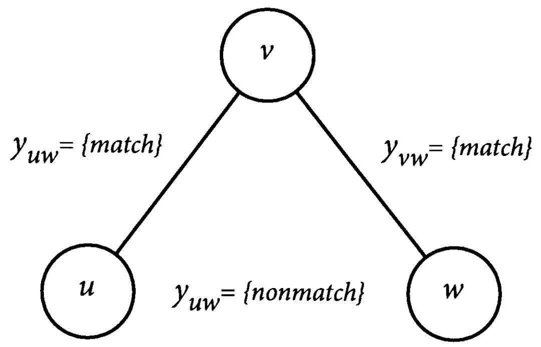


Figure 2-1: A simple linkage graph

in which each record is initialized in its own block and then blocks are greedily merged until some termination condition is met. In single linkage, blocks are merged according to the maximum value the classifier produced for a pair of records that spans the blocks.

One obvious problem with single linkage is that it doesn't take into account the overall connectivity between the blocks: If the classifier considers most of the pairs between two blocks to be strong nonmatches, they may still be joined by a spurious high match. *Average linkage* rectifies this by joining blocks according the the average value of the classifier across all pairs that span the blocks. A still more conservative approach is to use the minimum classifier value between the blocks. Nongreedy approaches have also been used, in which the similarity function and linkage thresholds are iteratively updated [13].

blocking

Applying pairwise classifiers to n records requires $n(n-1)/2$ feature vectors to be computed and classified. This $O(n^2)$ computation may be intractable in large data sets, so many practical systems use *blocking techniques* to identify (possibly overlapping) subsets of the data that are likely to contain coreferent pairs. For publication records, a simple blocking technique might be to group publications by their copyright year field; publications with distinct years are assumed to be nonmatches without the need for classification. In practice, effective blocking requires a linear-time algorithm that clusters potential matches into relatively small subsets.

pairwise relational models

The pairwise architecture presented thus far is primarily suited for flat databases. Consider the problem of duplicate detection in a relational setting, for example merging multiple library databases with publication and author records in a relational structure. An obvious constraint would be to require that declaring publication records coreferent would imply that their related author(s) record(s) would also be coreferent. There is no obvious extension to the pairwise architecture that embodies this constraint.

Despite these difficulties, there are some approaches that combined pairwise decisions with relational data. Dong et al. [34] describe a method for defining a dependency graph on pairwise coreference decisions in a publications domain, allowing coreference decisions to be propagated between record pairs. Bhattacharya and Getoor [13] describe a method for partitioning author records using co-authorship relations as an additional source of disambiguating information.

limitations of pairwise models

Although the pairwise architecture has the advantage of allowing a variety of classification techniques to be used "off the shelf," there are a number of significant limitations to pairwise methods. Milch et al. [62] note that pairwise features cannot capture certain global constraints. For example, it may be likely for an entity to have a small number of distinct values for a field (e.g., email address) but unlikely for there to be dozens. Another problem is the difficulty of applying domain knowledge and constraints in a principled way across blocking, classifier construction, threshold choice, and linkage algorithm. Finally, in some domains, there may arise identical records for distinct entities or disparate records for the same entity; pairwise methods may have trouble training on such outlying examples.

2.2.2 Partition-based Methods

Partition-based methods mitigate some of these limitations of pairwise methods, though typically at the expense of additional computation. A partition-based method provides a measure over the space of possible partitions of a data corpus. This measure serves to score candidate partitions by their estimated closeness to the true partition. Because the number of possible partitions grows combinatorially with the number of records,[2] nontrivial measures must be combined with an effective optimization technique.

generative Bayesian models

Generative Bayesian models, including the one described in this thesis, are ones means of providing a measure over partitions. In Bayesian models, conditional probability distributions are used to model the uncertainty in unobserved quantities. In Bayesian coreference models, the primary quantity of concern is the coreference partition but other unknown values may be incorporated into the model as well. One advantage of these models is that these unknown values may be inferred from data, for example a "clean" version of a publication title may be inferred from noisy representations.

A generative model provides a *prior* distribution over partitions $\mathrm{P}(B)$ and a *likelihood* distribution over observed record values $\mathrm{P}(V \mid B)$. We may think of the value of the coreference partition as hypothesis, with the prior specifying our belief about the hypothesis in the absent of data and the likelihood distribution specifying the likelihood of any particular observation under any particular hypothesis. Bayesian inference, then, provides a means of updating posterior belief $\mathrm{P}(B \mid V)$ of a hypothesis in light of observations using Bayes Rule:

$$\mathrm{P}(B \mid V) = \frac{\mathrm{P}(V \mid B)\,\mathrm{P}(B)}{\mathrm{P}(V)}$$

For a fixed set of observations V the posterior probability is proportional to the prior multiplied by the likelihood. Thus, the posterior provides a score for each partition given the evidence. However, optimizing this function over the space of all partitions is challenging. Monte Carlo sampling techniques are commonly used. In these techniques, described further in chapter 4, the defined distributions guide a stochastic search through partition space.

relational Bayesian models

Another advantage of Bayesian methods is they allow relational data to be modeled in a principled way, combining evidence from multiple

[2]The number of partitions of n items is commonly known as the nth Bell number [77]; the Bell numbers grow faster than any exponential function.

types of records to make simultaneous coreference decisions. Pasula et al. use the language of relational probability models (RPMs) [69] to model a relational citation domain [67]. Milch et al. develop BLOG, a language for describing domains with identity uncertainty, and provide BLOG models for aircraft observations and citation matching [62]. Carbonetto et al. extend BLOG to a nonparametric setting and also model citation generation [22].

These models are domain-dependent in that they provides explicit descriptions of the dependencies between the entities, relations, and field values specific to the domain of citation coreference. For instance, they each describe a model of citation generation with distinct string distributions for a publication's title versus an author's name. Adapting them to a new domain would require a modeler to chose different distributions appropriate to that domain.

discriminative models

An alternative to generative models are discriminative models, which score partitions without requiring the strict conditional independence assumptions of Bayesian modeling. McCallum and Wellner [58] introduce a Conditional Random Field (CRF) model of noun phrase coreference, in which the distribution $P(B \mid V)$ is modeled directly. One advantage of this model is that non-independent and overlapping features of the data may be included as evidence.

In the CRF model of McCallum and Wellner, pairwise features are used as evidence for the coreference decisions. The training procedure for the model weights is partition-based, however, maximizing the likelihood of the coreference decisions of the training data as a whole. Maximum likelihood inference in this model is equivalent to the graph partitioning problem known as *correlation clustering* [10, 33]. The CRF model was extended to multiple field citation records by Parag and Domingos [66] and Culotta and McCallum [31].

Discriminative partition-based models may also incorporate block-level features. Culotta et al. [30] present a feature representation in which the score of a partition is a weighted sum of block-level features. Efficient training procedures for the features weights based on minimizing errors are presented, and greedy agglomerative clustering is used to apply learned models to test corpuses. Though not strictly required, the features used in this approach were also specific to the evaluation domain of citation coreference.

2.2.3 Other Methods

This review of methods is far from exhaustive; a recent survey is provided by Winkler [94]. The categorization of methods into pairwise and partition-based is intended to be illustrative, but there is not always a clear distinction, particularly when linkage algorithms take the form of a partition search.

There are also models which don't fall into either category. For example, the *Discriminant Descriptions* model of Guha [44] finds a feature function for each entity, called a *key*, that distinguishes records referencing that entity from those that do not. We may call this an *entity-based* or *key-based* model.

2.3 Model-Constructing Frameworks

Most duplicate detection methods use structures that depend on the data environment in which they will be applied. These structures may be chosen by the modeler during a *modeling phase* (e.g., feature functions) or they may be estimated from data during a *learning phase* (e.g., model parameters). A particular choice of parameters defines a partitioner, which can then be applied to new data in the *evaluation phase*. Different methods take different approaches to constructing a partitioner; we consider three such frameworks here, which we call *fixed*, *learned*, and *active*, and discuss their suitability to various data environments.

model construction

Many methods depend on a substantial modeling effort. In generative Bayesian models, the partitioner is tasked with defining the domain variables and specifying their conditional independence relations and probability distributions. In discriminative models, appropriate features must be chosen. Even when structure learning or feature selection algorithms are used, the modeler is still responsible with defining the space of features from which to choose. Effective modeling requires both knowledge about the particular method being used and also domain knowledge about the data to which it is being applied.

However they are constructed, most models incorporate some form of domain or schema-dependent knowledge. As discussed in the previous chapter, domain-specific features can be essential to achieving acceptable performance in practice. Domain-specific knowledge may be as simple as normalizing common abbreviations [63] or it may involve constructing an entire Bayesian description of a relational domain [67, 62]. Still, there remains interest in designing more flexible models that require less modeling up front by using machine-learning techniques to adapt domain-independent features to a particular domain [88, 5, 19]

fixed model

The most basic duplicate detection method is to construct, by hand, linkage rules that determine whether a record pair should be classified as coreferent. Once specified, such rules define a unique partitioner which may be evaluated on a test corpus. In this *fixed model framework*, illustrated in Figure 2-2, a partitioner is produced directly by the modeler. Though no training is required, such models need not be trivial to implement: a fully specified Bayesian model may require advanced techniques to perform effective inference.

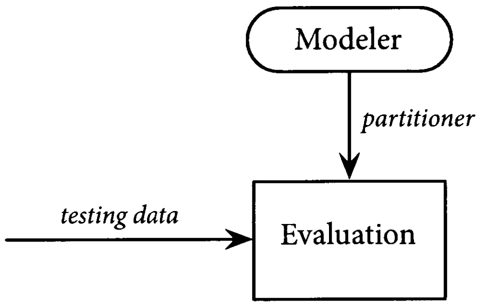


Figure 2-2: Fixed model framework

learned model

Most of the methods discussed previously involve some form of fitting models to data. In the *learned model framework*, depicted in figure 2-3, we may think of the modeler as defining a class of partitioners, with the particular partitioner chosen by a learning algorithm during training. Training may take two forms: supervised algorithms are commonly trained on a hand-partitioned corpus, but unsupervised algorithms may be used either separately or in conjunction to estimate parameters from an unlabeled corpus. For example, Pasula et al. used a census name file to estimate parameters in a generative model for author names [67].

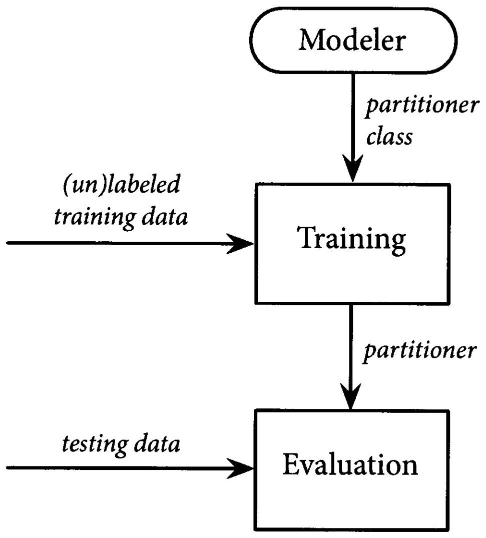


Figure 2-3: Learned model framework

active-learned model

The learned model framework may be extended with an active component. In this *active-learned model framework*, the training algorithm is able to take advantage of unlabeled or partially labeled data to generate a partitioner, but is also augmented by a *querying algorithm* that identifies interesting portions of the training corpus for a human *supervisor* to label. These labels are then fed back into the system to produce further partitioners and queries. The process may terminate when a particular condition is met or when the results are deemed satisfactory. A diagram of this framework is given in figure 2-4.

Both Tejada et al. [87] and Sarawagi and Bhamidipaty [79] have proposed active approaches to partitioner training. These works use similar architectures, in which a pairwise classifier is first trained using a small portion of the records in a corpus, then used along with a query-by-committee [81] approach to identify additional training pairs. Addi-

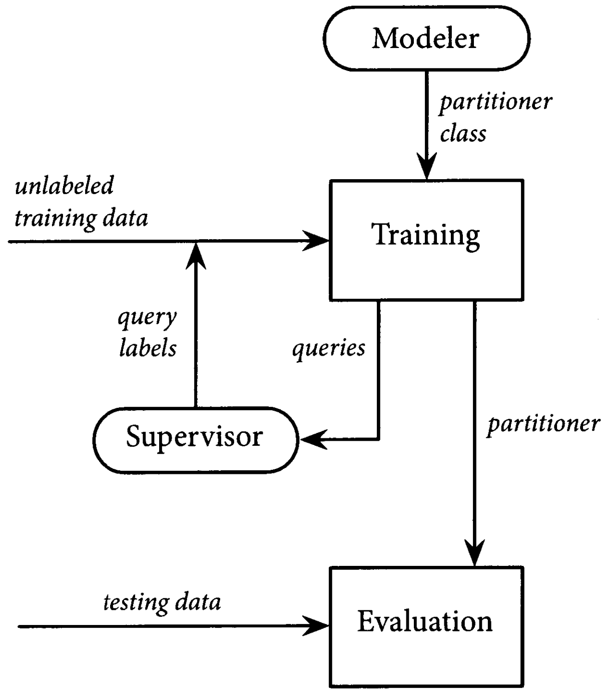


Figure 2-4: Active model framework

tional classifiers and queries are then generated iteratively. Bilenko [16] uses a "static" active approach, where a schema-generic heuristic is used to suggest likely-positive examples for training a pairwise classifier. An active approach has also been applied to clustering documents by Basu et al. [11]; In this model, binary match constraints were incorporated into a *k-means* clustering algorithm.

framework properties

The choice of model construction framework comes with tradeoffs. Fixed models are human-designed and so may be easier to introspect and correct; however they are also unable to take advantage of unanticipated features of the data that may improve performance. Learned models may provide better performance, but they too require judicious design and an appropriate choice of training data. When the work of model design and labeling training data can be amortized across a sufficient amount of evaluation data, a learned model may be the best choice.

The active approach is suited to applications where the work of labeling a training corpus cannot be amortized to a large amount of evaluation data. One compelling application is duplicate detection in personal information databases: these may have characteristic statistics not found in a generic training corpus and may be semi-structured, with an ad hoc schema determined by the user; such data may have novel fields and values that cannot be anticipated during the modeling phase.

Another use case may be data generated by a process with frequently changing statistics. For example, in a web commerce application there may be a large amount of unsupervised data but the characteristics of duplicate records may vary with market conditions. In this case, active queries could be used to keep a partitioner performing well while minimizing human involvement.

Finally, even if the statistics of the data are relatively static, an active approach may be useful in minimizing the amount of human labeling necessary to achieve an acceptable level of performance. Sarawagi and Bhamidipaty have shown that selecting a training corpus using an active learning technique can produce better classifiers than an arbitrarily selected training corpus [79].

2.4 Evaluation Metrics

During the evaluation phase, a partitioner produces an *estimate partition* of a data corpus. For nontrivial data, it is unlikely that a partitioner will estimate the true partition perfectly, so evaluating performance quantitatively requires defining a difference metric between the estimated and true partitions. We discuss the most commonly used metrics here, along with their properties.

pairwise metrics

Like the methods themselves, these metrics can be broadly categorized as either pairwise or partition-based. Pairwise metrics treat item pairs as independent points of comparison. The *error rate* is the percentage of pairs for which the estimate and true partition disagree on the match/nonmatch labeling. The *recall* is the percentage of true matches that are correctly labeled in the estimate, while the *precision* is the percentage of estimated matches that are true matches. When the true partition has no matches, we define the recall to be 100% and when the estimated partition has no matches, we define the precision to be 100%.

Precision and recall are often preferred to error rate since they do not credit a partitioner for accepting the strong null hypothesis that two records are distinct[3]. There is also a natural tradeoff between these metrics: perfect recall is trivially obtained by the *single-block partition*, with all items in a single block, while perfect precision is obtained by the *singleton partition*, with each item in its own block. The harmonic mean of precision p and recall r is called the *f-measure*: $f = \frac{2 \cdot p \cdot r}{p+r}$. The f-measure ranges from zero, when either precision or recall is zero, to 1 when the estimate is the true partition.

While pairwise metrics fit neatly into the framework of statistical hypothesis testing, they disregard the fact that coreference decisions are transitive. Consider a partitioner that leaves one record out of an otherwise correct block. In this case, the pairwise recall will vary with the size of the block; larger blocks will accrue greater recall error than smaller ones. An analogous situation may occur with pairwise precision. This penalizes a partitioner repeatedly for what may be regarded as a single error.

partition-based metrics

For this reason, many researchers prefer partition-based metrics, which directly compare the generated partition to the true partition. One commonly used metric is a block-based variant of f-measure called MUC, developed for scoring the noun-phrase coreference task at the 6th DARPA Message Understanding Conference [89].

The MUC metrics may be understood by considering how the true partition may be constructed through a series of *link operations* that connect coreferent elements. Starting from the singleton partition, a block b from the true partition can be constructed with $|b| - 1$ link operations (a block of size 1 requires no links). This is an upper bound; it can be shown that at most $|b| - 1$ links are requires to join the elements from b starting from any partition, including an estimate partition.

MUC recall for an estimate partition is one minus the MUC recall er-

[3]this is considered either a strength or a weakness of the precision-recall framework, depending on one's point of view

ror, which is the ratio of the number of link operations required to join coreferent elements starting from the estimate partition with the number of such operations required starting from the singleton partition. As in pairwise recall, the single-block partition has zero error and thus perfect recall. *MUC precision* is computed in the same way as MUC recall, but with the roles of the estimate and true partitions reversed. *MUC f-measure* is the harmonic mean of precision and recall, as before. When the true partition has blocks of size no larger than two, then pairwise and MUC precision and recall are identical. A thorough description of the MUC metrics is provided by Bagga and Baldwin [9].

The MUC metrics has the desired property of linear dependence on the size of the blocks, rather than quadratic dependence as in pairwise metrics. However, this property has a flaw, noted by Popescu-Belis and Robba [74], that makes it unsuitable for data sets with large blocks in the true partition: as block size grows, the penalty in precision for spuriously joining two blocks becomes proportionally smaller. For example, merging two block of size 20 is interpreted as 19 correct links per block and 1 incorrect link, for a precision of $1 - 1/39 \approx 0.97$.

Like pairwise precision and recall, the MUC metrics do not credit for correctly placing records in singleton blocks. An alternative partition-based metric that takes singleton blocks into account is the percentage of blocks in the true partition that are also in the estimate partition (i.e., labelled completely correct) [54, 67]. This strict measure is easy to understand and compute, but does not distinguish between false matches and false nonmatches nor does it weight blocks by their size or provide partial credit for large blocks that are "almost correct".

evaluating active approaches

Although machine learning techniques are typically evaluated on an independent evaluation corpus, in active approaches the training and test sets are often the same, which leads to the question of whether to exclude labeled queries from reported results. This is complicated further by evaluation sets with blocks of size 3 or greater, where the labels of query pairs may imply the label of non-query pairs through transitivity.

In the case of data from paired sources, Tejada et al. include results from labeled pairs [87], while Sarawagi & Bhamidipaty exclude labeled pairs [79]. There is some justification for either choice. Including the labeled pairs models the scenario in which active fitting is used for each new evaluation corpus. This is the approach we take in this thesis. Excluding labels models the scenario in which actively selected data are used to train a partitioner that will be applied to many corpora without active fitting.

In practice there is no universally accepted evaluation metric or framework. It has been noted that the variety of error metrics hinder side-by-side comparisons between methods, even when standard data sets are used [20]. While agreeing with this sentiment, we note also that the choice of architecture, model-construction framework, or application may lead to different choices of metric.

2.5 Related Problems

Some practical problems in artificial intelligence are closely related to duplicate detection, so much so that similar methods are often applied. This section serves to briefly review these problems while highlighting their additional constraints and structures that have no direct analog in duplicate detection.

Clustering is the problem of grouping a set of data points into clusters, either disjoint or overlapping, based on some measure of similarity. A common application is to automatically organize documents by their contents into topic groups. From one perspective, duplicate detection may be thought of as a special case of clustering, since matched records will generally be more similar than unmatched records.

However, there is a fundamental distinction between these problems in that clustering is primarily concerned with modeling *similarity* while duplicate detection is concerned with modeling *identity*. In a clustering problem, identical points are usually considered maximally similar, while in duplicate detection they may still represent distinct entities. Unsupervised methods are more commonly used in clustering whereas supervised methods are more common in duplicate detection, perhaps because in clustering the choice of similarity metric space serves the role of labeled coreference examples.

Noun phrase coreference is the problem in natural language processing of identifying coreferent noun phrases in a corpus of text [59, 45]. For example, in a news story the same person may be referred to alternately as “Mr. Obama”, “the president”, or “he”, with the context of the phrase determining the most likely referent. This problem may be viewed as a special case of the duplicate detection problem in which the records consist of a single plain-text field. However this disregards the important role of phrase context, which has no analog in duplicate detection. Without some model of context, it is impossible to effectively resolve pronouns.

Image correspondence and *segmentation* are problems in machine vision. In image correspondence, different views of the same scene are put into correspondence by finding points in the images that correspond to the same point in the scene. In image segmentation, the task is to partition an image into related regions, for example, by partitioning a foreground subject from the background. As in noun phrase coreference, modeling context using methods specific to vision is essential to these problems.

Schema mapping is the data-mining problem of identifying corresponding fields in different schemas. For example, one address database might have a single address field, while another may have separate fields for the street address, city, and zip code. Such correspondences can be inferred from data and, like duplicate detection, this problem is particularly challenging in highly relational, semi-structured data environments.

2.6 Context for the Work

The distinctions presented in this chapter provide the context for this work. The technique presented in the following chapters is intended for estimating a coreference partition in situations where little can be assumed *a priori* about the data, perhaps because it comes from an unknown schema or source, and it is not practical to build an explicit model of the data, either through Bayesian modeling or by choosing domain-specific feature functions and labeling a training corpus.

Using a partition-based model, the technique is able to take advantage of features of the data involving entire blocks of records, rather than just pairs. Also, as a Bayesian model, it provides a formal methodology for incorporating additional constraints as Bayesian evidence, so that blocking, thresholding, and linkage decisions are computed in a unified manner.

Finally, the proposed technique operates in an active learning framework, able both to estimate a partition of unlabeled data, but also to adjust its estimate by requesting labels for the ambiguous portions of the data. By combining these three components, we hope to reduce the human effort required to address duplicate detection in novel data environments.

Chapter 3

Bayesian Nonparametric Model

All models are wrong. Some models are useful.

George Box

This chapter presents a domain-independent Bayesian nonparametric model of the coreference relation of a data corpus. This model consists of two probability distributions: a prior over the coreference partition and a likelihood model for the distribution over observed record values given a particular coreference partition. The following chapter presents an inference procedure for this model based on MCMC sampling, which generates approximate samples from the posterior distribution over coreference partitions conditioned on a data corpus. These samples serve as a model of the partition uncertainty.

We begin here with the motivation for the model and a high-level description of its dependency structure, then continue with an introduction to some Bayesian nonparametric distributions before describing the complete model. Two variations of the model are presented: the *individual value distributions* (IVD) variant, in which each coreference block has its own distribution over field values, and the *shared value distributions* (SVD) variant, in which value distributions may be shared by multiple coreference blocks. The chapter concludes with a qualitative description of the model as well as connections to related models from the literature.

3.1 Motivation

As discussed in section 2.2.2, generative Bayesian models provide an attractive formal framework for duplicate detection: they explicitly model the uncertainty over possible coreference partitions and allow evidence to be combined from multiple records, rather than considering only pairwise features. Bayesian inference also provides a principled means of integrating different kinds of evidence. Unlike in most pairwise methods, for which decisions related to blocking, classification, and linkage are typically unrelated, in Bayesian models both observed values and high-level constraints are combined as evidence.

Another advantage Bayesian models is that they explicitly model the independence assumptions of the data. Although these assumptions are typically violated by nontrivial data, even simplified models (e.g., naïve Bayes) often provide reasonable performance. Moreover, when models are inadequate, the Bayesian framework provides modelers with a principled means to improve performance by relaxing assumptions or choosing more appropriate distributions for the data.

Many prior Bayesian models for duplicate detection have been developed for a specific data domain. In particular, a number of models have been developed for identifying duplicate research citations [67, 62, 22]. This thesis takes a different approach, presenting a domain-independent model of coreference.

Because the schema is unknown, this model is unable to exploit potentially useful domain-specific features or noise models. However, as a nonparametric model, its structures are primarily determined by the data itself. This allows its use on data expressed in a variety of schemas or data with unusual statistics that may not be anticipated by a domain modeler.

The model is also intended for an active framework, in which a human supervisor provides coreference labels to queries made by the system. Provided the model is sufficiently accurate to distinguish regions of true ambiguity from unambiguous labels, the supervisor can provide the labels for the most difficult cases and the system can resolve the remain-

name	addr	city	type
river cafe	1 water st.	brooklyn	american (new)
cava	3rd st.	los angeles	mediterranean
sofi	3rd st.	los angeles	mediterranean

Figure 3-1: A portion of a flat file database

ing ambiguity without human intervention.

3.2 Model Overview

To explore domain-independent modeling, we start with the simple data model of a flat table of textual values, with rows representing records and columns representing fields. The development of richer models, incorporating other data types and relational values, is left to future work. This section presents an abstract overview of the model and its dependence assumptions, with the precise distributions presented in section 3.4.

data model

Figure 3-1 provides a portion of the restaurant corpus from the RIDDLE repository [15]. This corpus has 5 textual fields: name, street address, city, type, and phone number (not shown). Records are assumed to be distinguishable even when their field values are identical or even unspecified. This is often accomplished through a special field holding an identifier for the record itself, but such fields are not explicitly represented in the model.

We model record values as stochastic variables. Each record r_i in the corpus is associated with a random variable V_i representing its field values: V_{i1}, ..., V_{im}. We denote the values for an entire corpus simply as V. Fields with known values, $V_{ij} = v_{ij}$, are treated as observed variables to be conditioned on during posterior inference. Missing values are taken to be missing completely at random and are simply treated as unobserved variables that are implicitly marginalized during inference.

Any particular corpus has a fixed number n of records and fields m forming a table of $n \cdot m$ value variables. These quantities are taken to be determined by the data and not treated as stochastic variables. However, it is shown in section 3.4.2 that the model presented in this chapter may be viewed as a distribution over infinite data sets, with unbounded numbers of records and fields. The unobserved records and fields are implicitly marginalized by the inference procedure as a natural consequence of the nonparametric distributions of the model.

partition-valued random variables

The distribution over record values is defined conditionally on the coreference partition of the records, B. The partition is represented as a vector of integer random variables that are the *block indices* of the items, $B = [B_1, \ldots, B_n]$. In this representation, the partition $\beta = \{\{1\}, \{2, 4\}, \{3\}\}$ would be denoted as $B = [1, 2, 3, 2]$. Thus when a record pair (i, j) is coreferent, their block indices are equal: $B_i = B_j$. If the records are totally-ordered, B can be expressed in a canonical form such that every partition has a unique block index representation.

posterior distribution

The prior distribution $\mathrm{P}(B)$ of the coreference partition is conditioned on a set Θ_B of Bayesian parameters that influence the partition structure. The field-value likelihoods $\mathrm{P}(V \mid B)$ are likewise conditioned on parameters Θ_F which control the variation present in the fields. Specifying priors $p(\Theta_B)$ and $p(\Theta_F)$ for these parameters is sufficient to completely specify the posterior distribution over partitions $\mathrm{P}(B \mid V)$. For a given V, the posterior is proportional to the factored expression,

$$\mathrm{P}(B \mid V) \propto \int_{\Theta_B} \int_{\Theta_F} \mathrm{P}(V \mid B, \Theta_F)\, \mathrm{P}(B \mid \Theta_B)\, p(\Theta_F)\, p(\Theta_B). \tag{3.1}$$

Though computing the posterior exactly is intractable for the model presented in this chapter, this equation is provided as a high-level description of the conditional dependency structure of the model. Figure 3-2 provides another representation of this structure in graphical model notation, in which random variables are represented by nodes, dependencies by directed edges, and repeated variables by square plates.

additional independence assumptions

To facilitate tractable inference, the model makes even stronger conditional independence assumptions than those apparent in Equation 3.1 and Figure 3-2. One such assumption is that the field values of records in distinct blocks are independent under certain conditions. In the IVD

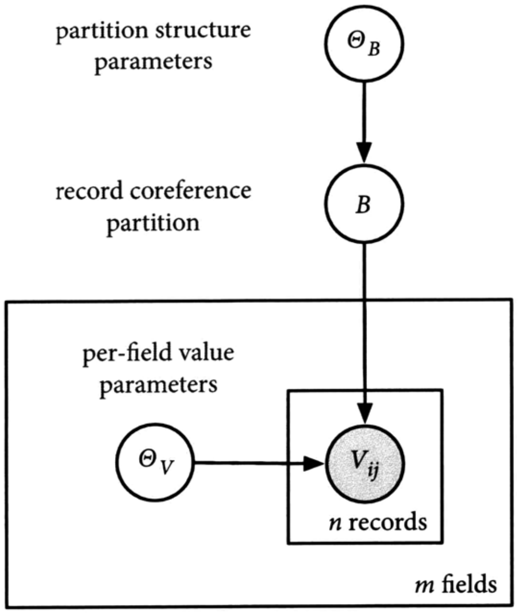

partition structure parameters
Θ_B
record coreference partition
B
per-field value parameters
Θ_V
V_{ij}
n records
m fields

Figure 3-2: Abstract representation of the model dependency structure

variant, the values of records in distinct blocks are conditionally independent given the value parameters $V_i \perp\!\!\!\perp V_j \mid \Theta_F, B_i \neq B_j$. This assumption is relaxed slightly in the SVD variant of the model.

An additional assumption is that the values associated with each field are assumed to be conditionally independent within a coreference block. That is, the joint distribution over the values within a block, $V_{\{b\}} = \{V_i : B_i = b\}$, may be factored according to its fields,

$$\mathrm{P}(V_{\{b\}} \mid B, \Theta_F) = \prod_{j=1}^{m} \mathrm{P}(V_{\{b\}j} \mid B, \Theta_F).$$

This completes the abstract description of the model and its dependency structure. The model is described in terms of its component distributions completely in section 3.4.

3.3 Bayesian Nonparametric Processes

In recent years, there has been a great deal of interest Bayesian nonparametric distributions for clustering, segmentation, and natural language processing [86, 48, 50, 83]. Despite their name, nonparametric distributions are not those without parameters, but distributions in which the number of parameters grows with the amount of data.

There are a number of theoretical justifications for using these distributions, such as allowing the number of mixture components in Bayesian mixture models to be data-driven [64] and also the fact that certain nonparametric distributions exhibit the power-law behavior commonly found in natural data sets, such as word frequencies and citation graphs [43, 85].

These qualities also make Bayesian nonparametrics an attractive choice for developing a domain-independent Bayesian model of database generation, both in characterizing the distribution of records into coreference blocks and for modeling the values observed in coreferent records. These properties have been noted by other researchers using nonparametric models for duplicate detection in the domains of publication authors [14] and noun phrase coreference [45].

We now introduce three closely related distributions prevalent in Bayesian nonparametric models: the Ewens distribution, the Chinese Restaurant process, and the Pitman-Yor process.

3.3.1 The Ewens Distribution

The Ewens distribution is a distribution over partitions of the set of integers $\{1, \ldots, n\}$. Described by Ewens [35] as a single-parameter distribution arising in population genetics, it was later generalized to two parameters by Pitman [71]. We briefly present the distribution here without derivation in order to provide a qualitative description. For a thorough treatment and associated literature, see Pitman [71].

The two real-valued parameters of the distribution are commonly called the *strength* (or *concentration*) parameter, $s > 0$, and the *discount* parameter, $0 \leq d < 1$.[1] The distribution over partitions is compactly de-

[1]In mathematics, the strength and discount parameters are typically denoted θ and α, while in machine learning, α or a are sometimes used for the strength parameter and α, b, or d for the discount parameter. We choose s and d here for clarity.

scribed by representing a partition as a set of blocks,

$$\beta = \{b_1, \ldots, b_k\},$$

with $k = |\beta|$ as the number of blocks and $|B_i|$ as the number of items in a block B_i. In the two-parameter variant, $d > 0$, and the Ewens distribution over B is:

$$\mathrm{P}(B = \beta;\, s,\, d,\, n) = \frac{\Gamma(s)}{\Gamma(s+n)} \frac{d^k\,\Gamma(s/d + k)}{\Gamma(s/d)} \prod_{i=1}^{k} \frac{\Gamma(|B_i| - d)}{\Gamma(1-d)},$$

where the Gamma function Γ is the well-known extension of the factorial function. In the one parameter variant, $d = 0$ and the distribution takes the simpler form,

$$\mathrm{P}(B = \beta;\, s,\, n) = \frac{\Gamma(s)\, s^k}{\Gamma(s+n)} \prod_{i=1}^{k} \Gamma(|B_i|).$$

Some observations follow directly from the formulae above. First, the probability of a partition depends only on the number and size of its blocks, so it may be used as a distribution over partitions of any set of n elements. This also implies that the probability of a partition is invariant under a permutation of its elements: a permutation of the elements with the same block sizes yields the same probability.

Second, this distribution has support (nonzero probability) on all possible partitions, but the parameters govern the number and size of the blocks in a typical partition. Informally, when s and d are near zero, the distribution favors partitions with a small number of large blocks. In the limit of $s \to 0$ and $d \to 0$ the probability mass moves to single-block partition, $\mathrm{P}(B = \{\{1, 2, \ldots, n\}\}) \to 1$.

As d increases, the distribution favors more blocks and as s increases, the distribution favors smaller blocks. These tendencies are clearly related, as size of the blocks necessarily decreases as the number of blocks increase. The relationship between these parameters is elucidated in the next section. In the limit of $s \to \infty$ or $d \to 1$, the probability mass moves to the singleton partition, $\mathrm{P}(B = \{\{1\}, \{2\}, \ldots, \{n\}\}) \to 1$.

3.3.2 The Chinese Restaurant Process

The Chinese restaurant process (CRP) is a discrete-time stochastic process over partitions, $B^1, \ldots, B^n$, with the number of elements in $B^i = i$ and B^i being Ewens-distributed with fixed parameters s and d.

Aldous [6] provided its eponymous metaphor, in which a sequence of distinguishable customers are seated at a spacious Chinese restaurant with an infinite number of infinite capacity, indistinguishable tables. At each time step, a customer enters the restaurant and is seated at a table, either empty or occupied. After n steps, the seating arrangement defines a partition β^n, with the customers representing the n elements and the tables representing the k blocks. The seating arrangement at a particular table is irrelevant, as blocks are unordered.

The seating probabilities depend on strength and discount parameters, s and d, as in the Ewens distribution. At time step 1, the first customer is seated at an empty table. At time step $n+1$, a customer enters the restaurant with k occupied tables, $b_1, \ldots, b_k$. Customer $n+1$ is seated at an empty table with probability $(s+k\cdot d)/(n+s)$ or at an occupied table, b_j, with probability $(|b_j|-d)/(n+s)$. Thus the seating follows a *preferential attachment* model, wherein tables with many seated customers are likely to grow even larger.

This construction further elucidates the relationship between the parameters and the asymptotic behavior of the Ewens distribution. As s increases, so does the probability of occupying additional tables. The parameter d discounts the attraction of existing tables in favor occupying new tables. When $d \neq 0$, the expected number of tables follows a power-law, growing as $O(s \cdot n^d)$ [84]. In the case in which $d = 0$, there is no discounting and the expected number of tables grows logarithmically, as $O(s \log n)$ [84, 72].

3.3.3 The Pitman-Yor Process

The Pitman-Yor (PY) process [73] is a two-parameter generalization of the Dirichlet Process [38] and closely related to the CRP. There are a number of good tutorials on these processes available [42, 51, 64]; here we provide a brief introduction to the process, focused on the properties that are relevant to modeling coreference.

A PY process provides a random probability distribution $\mathbf{G}$ over some some discrete or continuous sample space Ω. It takes strength s and discount d parameters as in the CRP, as well as a *base distribution* G_0 which may be any probability distribution over Ω. A number of schemes exist for constructing samples from a PY process including the *stick-breaking model* of Sethuraman [80] and the *Pólya urn model* of Blackwell and MacQueen [21]. Here we limit the discussion to the Chinese Restaurant construction [70].

In this construction, the Chinese Restaurant metaphor is extended by associating each customer i with a random variable X_i over Ω. Customers are seated at tables as before, with probabilities determined by the strength and discount parameters. The value of a particular X_i is determined by the table customer i is seated at: each table has a value drawn independently from G_0 that is shared by all customers seated at the table. Constructed in this way, an infinite series of X_i are distributed according to a random draw $\mathbf{G}$ from a PY process.

A number of properties of the PY process are apparent from this construction. If G_0 is discrete, there is nonzero probability that two tables share a value. If G_0 is continuous, $\mathbf{G}$ is still discrete with probability 1, having support on a countably infinite number of points in Ω. The PY process also exhibits the same preferential attachment statistics as the CRP, with previously sampled values being more likely to be drawn in the future. However, in the limits of $s \to \infty$ or $d \to 1$, the probability that each customer i is seated at a new table goes to 1 and $P(X_i) \to G_0$.

In practical implementations, the CRP construction provides a means of computing the marginal distribution of a set of variables $X_1, \ldots, X_t$ distributed according to a random $\mathbf{G}$ drawn from a PY process with parameters s, d, and G_0. This marginal distribution integrates over all possible values of $\mathbf{G}$:

$$\begin{aligned}&\mathrm{P}(X_1 = x_1, \ldots, X_t = x_t; s, d, \mathrm{G}_0) = \\ &\qquad \int_G \mathrm{P}(X_1 = x_1, \ldots, X_t = x_t \,|\, \mathbf{G} = G)\mathrm{P}(\mathbf{G} = G; s, d, \mathrm{G}_0).\end{aligned}$$

To compute this marginal with the CRPconstruction, we define a partition β_x of the variables according to their values $x_1, \ldots, x_t$ such that all variables in a block b share a value x_b. If G_0 is continuous, there is only one partition β_x since the values at each table in the CRP are distinct with probability 1. In this case, the marginal probability may be computed as

$$\mathrm{P}(X_1 = x_1, \ldots, X_t = x_t; s, d, \mathrm{G}_0) = \mathrm{P}(\beta_x; s, d) \prod_{b \in \beta_x} \mathrm{P}(x_b; \mathrm{G}_0),$$

where $\mathrm{P}(\beta_x)$ is Ewens-distributed. Thus, for a continuous G_0, the CRP construction provides an efficient means of modeling a PY process without modeling **G**; all that is necessary to compute the likelihood of a set of values under a PY prior is the counts of each value. If G_0 is discrete, it is possible for multiple blocks in the CRP construction to share a value. In this case, computing the marginal distribution requires a sum over all possible partitions β_x as well.

In the Dirichlet process, this sum can be done analytically without enumerating the partitions. This is because, if the discount parameter is zero, the probability of a customer being seated at a table with value x depends only on the number of other customers seated at such tables and not the exact seating arrangement. When the discount parameter is nonzero, then the seating arrangement must be summed over explicitly or incorporated as state in an MCMC sampling routine. A thorough analysis is provided by Teh [84].

3.3.4 Exchangeability

Efficiently sampling from a CRP or PY process relies on the fact that random variables distributed according to these processes are *exchangeable.*

That is, the distribution over block memberships B_i in in the CRP and the distribution over variables X_i draw from a PY-distributed **G** are invariant to permutations of their indices i.

Concretely, if the position of two customers in a CRP is exchanged, then the distribution over partitions induced is the same. Thus, the block membership distribution of one element B_i conditioned on the remaining block memberships B_{-i} is the same as if that element were the last one added in a CRP,

$$P(B_i = b \mid B_{-i}, s, d) = \begin{cases} (s + k \cdot d)/(n-1+s) & \text{if } b \text{ is a new block} \\ (|b_j| - d)/(n-1+s) & \text{if } b = b_j \in B_{-i}, \end{cases}$$

where k is the number of blocks manifest in B_{-i} and n is the number of elements in B.

3.4 Model Description

Using the nonparametric Bayesian processes described previously, we describe here the complete generative Bayesian model of coreference, consisting of a prior over coreference partitions and the IVD and SVD likelihood models for values, conditioned on a partition.

3.4.1 Coreference Partition Prior

The two-parameter Ewens distribution is used as a prior over the coreference partition B. Because the frequency of coreference is generally not known *a priori*, the strength and discount parameters $\Theta_B = \{s, d\}$ of the prior are also treated as Bayesian variables, with hyperpriors discussed in section 3.4.4. Thus the prior is actually an superposition of the Ewens distribution over the range of parameter values: $P(B \mid s, d)\, P(s, d)$. Using Metropolis steps for parameter updates, this prior can be efficiently incorporated into a Gibbs sampling routine for posterior inference of B.

preferential attachment

More than its computational properties, though, this choice is motivated by the observation that the coreference structure in real data corpuses often seems to exhibit some form of preferential attachment statistics: As new records are observed, they are more likely to be associated with large coreference blocks versus smaller blocks. An illustration of this phenomenon is provided in figure 3-3, showing the block structure for four of the data sets used in the experiments of chapter 6, representing publications, email addresses, institutions, and authors.

Each of these graphs depicts a symmetric coreference matrix, where elements (i, j) and (j, i) are white if records i and j are coreferent and black otherwise. The records have been ordered according to block so that the matrix is block normal, with the largest blocks in the upper left. Though the scale is different in each figure, it can be seen that the block sizes fall off quickly for each data set, with many small or singleton blocks.

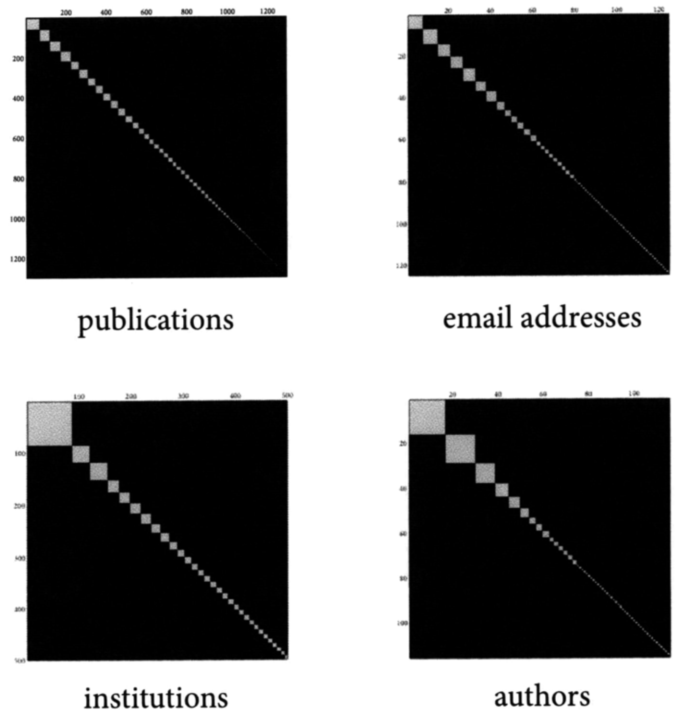


Figure 3-3: Some data sets exhibiting preferential attachment statistics

Some data sets may have a known upper bound on the block sizes, however. As discussed in section 2.2.1, if data is merged from a set of clean sources, then the maximum block size is bounded by the number of sources. For instance, the *restaurant* data set used in chapter 6 is a merger of data from Fodors and Zagats restaurant listings. The maximum block size in this data is two.

The Ewens distribution may still be used as a prior in cases such as this by conditioning on constraints as Bayesian evidence. Section 4.2 discusses incorporating constraints into inference and section 6.4.1 describes experiments demonstrating the performance gains from using constraints appropriate to the data set.

3.4.2 Individual Value Distributions Likelihood Model

We specify here the IVD likelihood model $\mathrm{P}(V \mid B, \Theta_F)$ for record values conditioned on a particular partition $B = \beta$. In this model, the values $V_{\{b\}j}$ for field j of block $b \in \beta$ are drawn from a *value-generating distribution* G_{bj}. This distribution serves as a model of co-generation, accounting for the fact that coreferent records are likely, but not required, to share values for a particular field.

value-generating distributions

The value-generating distributions for each block and field are modeled as independent draws from a Dirichlet Process $\mathrm{P}(\mathbf{G}_{bj} = G_{bj} \mid \theta_j, G_{j0})$ with a shared strength parameter θ_j and base distribution G_{j0}. The field value parameters Θ_F are simply the strength parameters $\{\theta_1, \ldots, \theta_m\}$.

The base distribution for a field is a uniform multinomial distribution over the field's vocabulary, a discrete set of tokens. The field vocabularies are taken from the data, allowing the same model to be used for all fields, irrespective of domain. Because the base distributions are fixed by the data, they are handled implicitly in the remaining presentation.

As determined by the base distribution, the sample space of the value-generating distributions are also multinomial, but with probability concentrated on particular tokens rather than uniform. For field values consisting of a sequence of tokens, the tokens in $V_{\{b\}j}$ are taken to be independent draws from $\mathbf{G}_{bj}$. Thus the order of tokens within a field is insignificant, constituting what is often called a unigram or "bag of words" model.

IVD dependency structure

A graphical representation of the dependency structure of the IVD

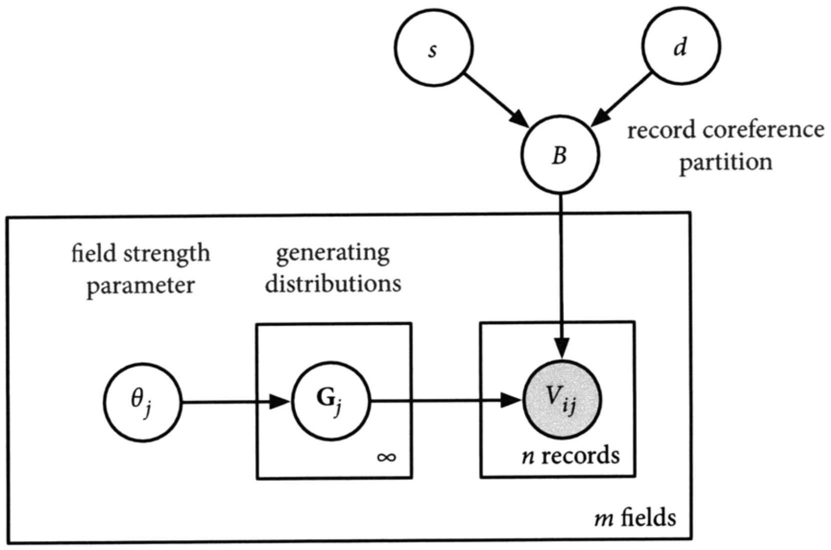


Figure 3-4: Individual value distributions model

model is depicted in figure 3-4. In this diagram, each node represents a random variable, directed edges represent conditional dependencies, and the rectangular plates represent repeated structures. A table of n records with m fields has n by m value variables V_{ij}. Though the number of value-generating distributions for each field is unbounded, any particular record i draws its values from the distribution indexed by its block index B_i. Therefore, the number of generating distributions manifest in a particular data set is bounded by the number of observed records.

modeling observation variation

In this model, the per-field strength parameter θ_j governs the noisiness of field j. If θ_j is low, the value distributions are concentrated and the tokens within a block are likely to be the same. If it is high, then the value-generating distributions for a field approach the uniform distribution and values within a block are no more likely to be similar than values from different blocks. Thus this parameter serves to model observation variation for the field.

As described in section 3.2, values from different fields within a block are conditionally independent under the partition structure and parameters. For records within a block $b \in \beta$, the joint distribution over all field values is

$$\begin{aligned}\mathrm{P}(V_{\{b\}} \mid B, \Theta_F) &= \prod_{j=1}^{m} \int_G \mathrm{P}(V_{\{b\}} \mid B, \mathbf{G}_{bj} = G)\mathrm{P}(\mathbf{G}_{bj} = G \mid \theta_j) \\ &= \prod_{j=1}^{m} \mathrm{P}(V_{\{b\}} \mid B, \theta_j).\end{aligned}$$

marginalizing generating distributions

By exploiting the implicit marginalization provided by the CRP, the likelihood $\mathrm{P}(V_{\{b\}} \mid B, \Theta_F)$ of the block values conditioned on the parameters can be computed directly without reference to value-generating distributions. This distribution is a measure of the coherence of an individual block. Nonparameteric distributions also allow the model to handle missing values for a particular field and even treat unobserved records as variables that are implicitly marginalized from the model. If additional data is added to the observation, then blocks and partitions are

joint likelihood

Because the values of records from distinct blocks are also assumed to be conditionally independent given the field value parameters, the value likelihood for a partition is simply the product of the per-block and per-field factors,

$$\mathrm{P}(V \mid B = \beta, \theta_1, \ldots, \theta_m) = \prod_{b \in \beta} \prod_{j=1}^{m} \mathrm{P}(V_{\{b\}j} \mid \theta_j).$$

This distribution, depicted as a graphical model in figure 3-5, is a measure of the coherence of a partition as a whole. Because fields are conditionally independent, unobserved fields are also implicitly marginalized by the model. Thus the IVD model can be interpreted as placing a probability distribution on database tables that are bounded neither in their number of records or number of fields.

3.4.3 Shared Value Distributions Likelihood Model

In order to be tractable, the IVD model makes a number of strong independence assumptions. In particular, it does not model the significance of token order, inter-field dependencies, or inter-block dependen-

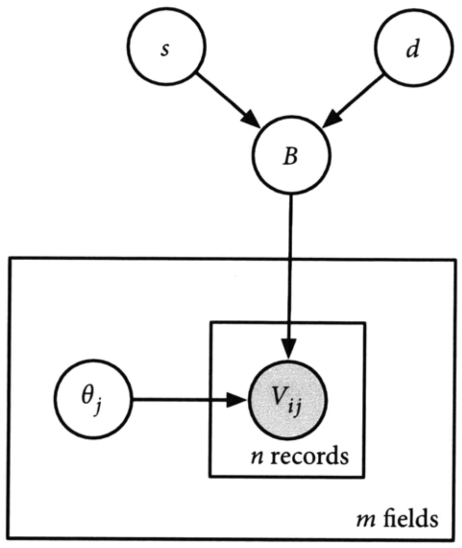


Figure 3-5: IVD model with marginalized Dirichlet Process priors

cies within a field. The SVD model is an extension of the IVD model which addresses inter-block dependencies within a particular field. Richer models of value generation are left future work with some possible extensions presented in chapter 7.

In the IVD model, the value-generating distributions G_{bj} for a particular block are assumed to be independent draws from a Dirichlet Process. While this model captures observation variation in the per-field strength parameters, it does not capture population variation, however. If record values are similar, then the most likely explanation under the IVD model is that the records were co-generated. However, in actual databases it is quite common for records which are *not* coreferent to have similar or identical values for a particular field.

For example, in a academic publications database, many publications may share the same venue and thus have a similar distribution over record values. This example illustrates a different explanation for record similarity, which is that the records refer to distinct entities which share some latent properties, giving rise to similar values. In this case, properties common to many entities, such as a common publication venue or year, should be discounted as evidence of coreference. Likewise rare properties, such as an unusual publication year, should provide additional evidence of coreference.

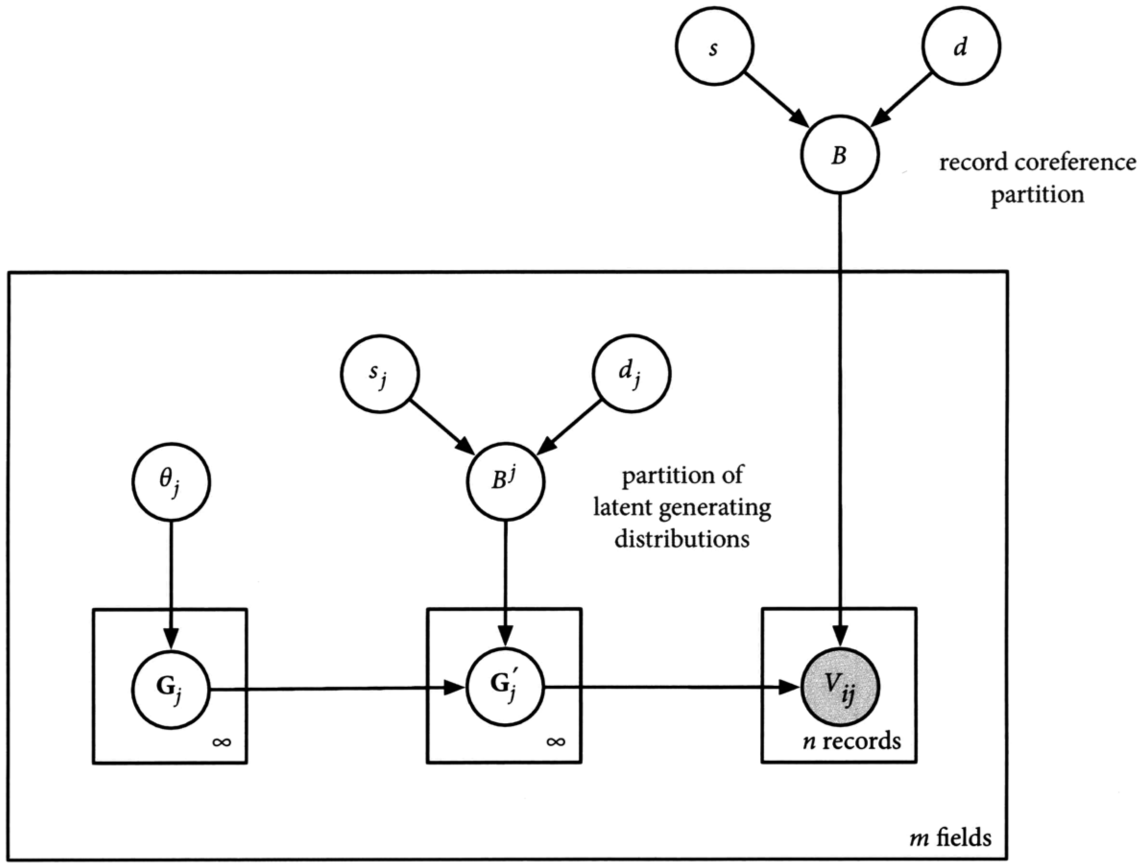


Figure 3-6: Shared value distributions model

The SVD likelihood model captures this population variation by allowing the value-generating distributions to be shared among different blocks. Sharing is accomplished by drawing the value-generating distributions $\mathbf{G}_j'$ for a block from a PY process with a Dirichlet process as a base distribution. Thus, instead of drawing a new generating distribution for each block, sometimes a distribution shared by existing blocks is drawn. A graphical model representation of the SVD model is given in figure 3-6.

In implementation, the PY process of the SVD model adds another partition variable for each field B^j, which partitions the blocks of B according to value-generating distribution. The PY process of B^j has its own parameters s_j and d_j which govern the amount of sharing. There is one PY process per field, and the partitions B^j are not constrained to be the same. For example, a publication record may share its `venue` field generating distribution with one set of records and its `year` field generating distribution with an entirely different set of records. In the limits of the PY parameters, B^j is the singleton partition and the SVD model reduces to the IVD model, with independent value distributions.

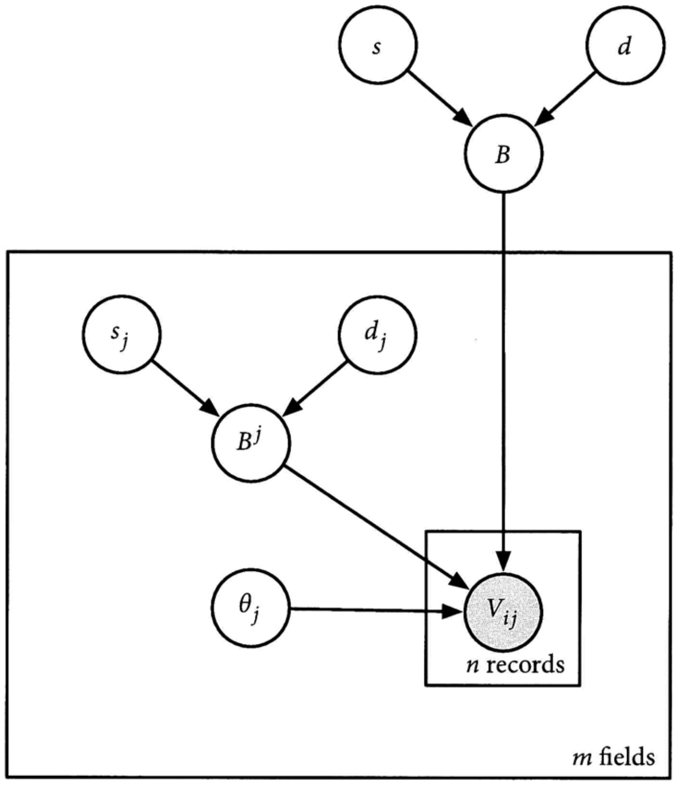


Figure 3-7: SVD model with marginalized Dirichlet Process priors

Implicit marginalization of value-generating distributions is also used in the marginalized SVD model, depicted in figure 3-7. In this case, the marginal distributions of values for field j are computed conditioned on sets of blocks in B^j which share a value-generating distribution.

3.4.4 Hyperpriors

For a data set with m fields, the IVD model has $m + 2$ hyperparameters: the strength s and discount parameter d for the coreference partition and the strength parameters θ_j for the m field Dirichlet processes. The SVD model adds another $2m$ parameters in the strength s_j and discount d_j parameters for the value distribution sharing process.

The per-field parameters θ_j use an exponential prior distribution with parameter 1 for a prior distribution. For the coreference and distribution sharing CRP parameters the same priors are used: the strength parameters s and s_k use the heavy-tailed lognormal distribution with standard deviation of 3; this distribution was found to approximately recover the value of the strength parameter from CRP samples with known strength values across a wide range (0.5–10000). The discount parameters d and d_j use a uniform distribution over $[0, 1]$.

Though all of these parameters are treated as Bayesian variables, previous work with Bayesian nonparametric distributions [84, 46] has often found models to be relatively insensitive to the parameter values, with the data tending to dominate.

3.5 Related Models

The model of record coreference presented in this chapter is closely related to recent nonparametric models of natural language, particularly the hierarchical Pitman-Yor language model of Teh [85] and the adaptor grammar model of Johnson et al. [50]. We briefly describe these models here, providing additional insight into the properties of the PY process and drawing connections between its use in natural language and coreference modeling.

hierarchical PY language model

In the hierarchical Pitman-Yor language model (HPYLM) of Teh, sentences are statistically modeled as n-gram sequences of words. The probability of observing a particular word in a sentence is determined by the preceding $n - 1$ words of context. In this model, the conditional probability distributions for each context are modeled as draws from a PY process. Additionally, the model has a hierarchical structure, where the base distribution over words for context length i is drawn from a PY process for context length $i - 1$, allowing distributions from different contexts to share support.

One motivation for using nonparametric distributions with an n-gram sequence model is to avoid overfitting due to the large number of parameters in a parametric sequence model. For example in a bigram (n=2) sequence model, the number of probabilities to be specified is the square of the vocabulary size, which typically numbers in the thousands of words. Fully specifying such a distribution would require millions of parameters to be estimated from data. However, since a training set will only exhibit a small portion of the possible occurrences, "smoothing" techniques are required to assign reasonable probabilities to unobserved occurrences.

The HPYLM capacity grows with the data: as more data is observed in a particular context the more of the probability mass of the context's distribution is determined by the data. Teh also demonstrates [84] that the hierarchical structure of this model implements a Bayesian interpretation of interpolated Kneser-Ney smoothing, providing a theoretical underpinning for the technique.

adaptor grammars

The adaptor grammar model of Johnson et al. extends probabilistic context-free grammars (PCFG) to a nonparametric setting. In a PCFG, sentences are taken to be generated from a stochastic grammar, which places a probability distribution over the sentences produced by the grammar. This distribution can be used to infer a posterior distribution over the parse for a sentence. In the adaptor grammar model, the probability distribution of each grammar production taken as drawn from a PY process.

One advantage of nonparametric distributions in adaptor grammars is that they provides broad support: Likely productions in a PY sample need not be specified *a priori* but may be any of the points in the sample space, allowing the support of the model to be data-driven. This model is used [50] to decompose words into stems and suffixes in an unsupervised fashion, without explicitly specifying valid stems and suffixes.

Language models have a natural connection to models of coreferent field values. The values of a coreference block may be viewed as a limited language: we might imagine a sequence or grammar model which generates a set of alternate observed titles for a book, perhaps sometimes including a subtitle and sometimes not, or with variations in spelling. The field value-generating distributions described in this chapter serve as a model of this language.

One challenge in taking this approach to coreference modeling, however, is that there are typically very few examples of any particular language in a data set. Often there is only a single example (e.g. the title of a particular book may only appear once in the data). The model presented in this chapter uses a flat unigram model for generating distributions and the SVD only allows sharing entire generating distributions. More sophisticated models could allows for sharing between *parts* of field values, thus incorporating evidence from all the values in drawing inferences about a particular record. Some suggestions for how this might be implemented are provided as future directions in chapter 7.

Chapter 4

Inference and Estimation

To generalize is to be an idiot.

William Blake

Doubt is not a pleasant condition, but certainty is an absurd one.

Voltaire

The previous chapter defines a generative model of the coreference relation of a data corpus. This chapter describes an efficient inference procedure for the model based on Markov Chain Monte Carlo (MCMC) sampling: given a data corpus, this procedure generates approximate samples from the posterior distribution over the corefence partition and other model structures. Section 4.2 extends this procedure to handle inference when portions of the partition are observed; that is when the posterior distribution is conditioned on observations which take the form of match or nonmatch labelings on pairs of records as well as the record values. The chapter concludes with an algorithm for estimating a single partition from samples based on the optimization problem known as *correlation clustering* [49].

4.1 Markov Chain Monte Carlo Sampling

According to sampling theory, the expectation of a function of the random variables $\phi = \mathrm{E}[f(\cdot)]$ may be estimated from posterior samples

$\sigma_1, \ldots, \sigma_k$ through the empirical distribution,

$$\hat{\phi} = \frac{1}{k} \sum_{s=1}^{k} f(\sigma_s).$$

We do not describe the complete theory of MCMC sampling here, but provide instead a brief introduction for the purposes of explicating the computational properties of the model and the modifications necessary for sampling distributions over partitions in the presence of pairwise constraints. A thorough introduction is provided by MacKay [57].

Sampling is used in this work to estimate the pairwise marginal probability $\hat{p}_{ij}$ that a pair of records (i, j) is coreferent,

$$\hat{p}_{ij} = \frac{1}{k} \sum_{s=1}^{k} Y_{ij}(\sigma_s).$$

Where Y_{ij} is the indicator random variable for the event that (i, j) is coreferent in B. Y_{ij} is treated here as a function which returns its value in sample σ_s. These estimates are used to address two problems: estimating an optimal partition from samples using correlation custering, described in section 4.3, and generating an informative query for active duplicate detection, presented in chapter 5.

In MCMC sampling, a probability model P is used to randomly evolve the state of a system in a series of discrete steps. At each step, a sample from a tractable *proposal distribution* is used to generate a candidate next state. The candidate state is stochastically accepted or rejected according to a function of the probability of the current and candidate state under the model. If accepted, the candidate state becomes the current state, otherwise the system remains in the current state.

This process defines a *Markov chain*, a distribution over states $\tilde{P}_t$ at time t defined conditionally on the state at time $t - 1$. In general, the distribution $\tilde{P}_t$ are distinct from P, however under appropriate conditions on the proposal distribution and acceptance function, $\tilde{P}_t \to P$ as $t \to \infty$.

ergodicity

One of the required conditions is that the chain be *ergodic*. A Markov chain is ergodic if it is positive recurrent and aperiodic. A positive recurrent chain is one in which it is possible to move from any state to any other state in a number of steps t with finite expectation $\mathrm{E}[t]$. An aperiodic chain is one in which the number of steps to return to a state is not constrained to be a multiple of some integer $k > 1$. Ensuring ergodicity is important when sampling under positive pairwise match constraints as discussed in section 4.2.

In practice, *approximate* samples from P are generated by evolving a system for a specified number of time steps using pseudorandom number generators. It is left to the practitioner to choose an appropriate proposal distribution and sufficient step count and interactions between the data, model, and proposal distribution may result in a model that converges poorly. Andrieu and Thoms [8] provide a tutorial on adaptive step methods, with discussion of many of the practical issues in MCMC sampling.

4.1.1 Sampling Block Memberships

A sample from the model of chapter 3 is a set of values for each of the unobserved variables in the model: the parameters, the coreference partition, and in the SVD variant, the partitions of value-generating distributions for each field. Because the observed record values are taken to be fixed and the model implicitly marginalizes unobserved values, sampling does not generate missing values for records. Though such samples could be generated in principle, they would offer limited utility given the simplicity of the value-generating distribution model.

The block membership variables of the coreference partition are sampled using *Gibbs sampling*, a form of MCMC sampling where the proposal distribution is the conditional distribution of a set of variables given the state of the remainder of the model. Provided the MCMC conditions are met, this proposal distribution is accepted with probability 1.

sampling IVD blocks

Gibbs steps are used to update the block membership variable B_i of a record i conditioned on the remainder of the model, including the remainder of the coreference partition B_{-i}. Due to the exchangeability properties of nonparametric distributions, described in section 3.3.4, this distribution may be computed in the IVD variant as follows:

$$\begin{aligned}
\mathrm{P}(B_i \mid B_{-i}, s, d, V, \theta_F) &= \frac{\mathrm{P}(B, s, d, V, \theta_F)}{\mathrm{P}(B_{-i}, s, d, V, \theta_F)} \\
&= \frac{\mathrm{P}(B \mid s, d)\,\mathrm{P}(V \mid B, \theta_F)}{\sum_{B_i} \mathrm{P}(B \mid s, d)\,\mathrm{P}(V \mid B, \theta_F)} \\
&= \frac{\mathrm{P}(B_i \mid B_{-i}, s, d)\,\mathrm{P}(V_i \mid B_i, B_{-i}, V_{-i}, \theta_F)}{\sum_{B_i} \mathrm{P}(B_i \mid B_{-i}, s, d)\,\mathrm{P}(V_i \mid B_i, B_{-i}, V_{-i}, \theta_F)},
\end{aligned}$$

where V_i are the observed values of record i and V_{-i} are the remaining observed values. The probability that $B_i = b$, then, is the product of the prior and likelihood of the model normalized by the sum of all possible blocks b, including a fresh block.

Informally, block membership sampling in this model takes the form of a stochastic walk through the space of all partitions. At each step, a record is removed from the partition and its membership is resampled, with model favoring membership in largeer blocks, as determined by the prior, and more self-similar blocks, as determined by the likelihood.

sampling SVD *blocks*

In the SVD variant, the block memberships are sampled as in the IVD variant, with two modifications. For computing the prior and likelihood of $B_i = b$ for existing blocks b, the prior and likelihood are computed as before, except that if b has distributions shared with other blocks, the values of the records in those blocks are also incorporated into the likelihood computation.

However, calculating the likelihood of $B_i = b'$ under a fresh block b' requires marginalizing B^j for each field—In the IVD variant, the values of a record placed in a fresh block are modeled as drawn from a distribution unique to the block. However in the SVD variant, a record placed in a new block may have values drawn from the generating distribution of other blocks.

Thus to compute the likelihood of values V_i under a fresh block requires computing the probability that V_i came from any of the existing distributions defined by B^j, as well as the probability that the values are drawn from a new distribution. However, these are the same likelihoods required to compute the probability of i joining an existing block, so these computations may be cached. The Gibbs step for the SVD variant takes a similar form to the IVD step:

$$\begin{aligned}&\mathrm{P}(B_i \mid B_{-i}, s, d, V, B^F, \theta_F) = \\ &\qquad \frac{\mathrm{P}(B_i \mid B_{-i}, s, d)\, \mathrm{P}(V_i \mid B_i, B_{-i}, V_{-i}, B^F, \theta_F)}{\sum_{B_i} \mathrm{P}(B_i \mid B_{-i}, s, d)\, \mathrm{P}(V_i \mid B_i, B_{-i}, V_{-i}, B^F, \theta_F)},\end{aligned}$$

where B^F is the current state of the generating distribution partitions.

In the case where the SVD sampler places a record in a fresh block b', the membership of b' in the generating distribution partitions B_i are simultaneously resampled. The Gibbs step for this sample is analogous to that for the block membership samples.

4.1.2 Sampling Parameters

In addition to the block membership variables, the model has CRP parameters s and d for the coreference partition and, in the SVD variant, parameters s_j and d_j for each field j. Since the distributions $\mathrm{P}(B \mid s, d)$ and $\mathrm{P}(B^j \mid s_j, d_j)$ can be evaluated analytically according to the Ewens distribution, any of a number of Metropolis-Hastings step methods may be used to resample the parameters in a Pitman-Yor model. Johnson et al. [50] use slice sampling and Teh [84] presents a sampling routine based on auxiliary variables. The results reported in this thesis were obtained using a symmetric Metropolis step with a symmetric Gaussian proposal distribution.

4.2 Conditional Inference and Blocking

There are a number of use cases where it is desirable to incorporate hard constraints in posterior inference. One such case is the application of blocking methods, discussed in section 2.2.1, another is the case in which data comes from known clean sources, discussed in section 2.2.1.

In these cases, the constraints are a set of record pairs taken to be nonmatches, which we call *negative constraints.*

A final use case is the active framework of this thesis, discussed in section 1.2, in which a supervisor provides the true coreference labels of record pairs. In this case, pairs may be either matches or nomatches, so constraints may be either positive or negative. In order to incorporate constraint sets into the model, we consider constraints as Bayesian evidence, to be conditioned on in all steps methods.

Sampling the block memberships in the presence of negative constraints requires only a slight modification to the sampling algorithm: when B_i is resampled, the probability of it being assigned to blocks containing known nonmatches of i is taken to be zero. The probabilities of the remaining blocks are calculated and normalized as before, constituting a valid distribution over blocks. In this case we are simply conditioning on the event that B_i doesn't violate the constraints.

sampling with negative constraints

However, sampling block memberships in the presence of positive constraints requires additional care to maintain ergodicity; if block memberships were sampled individually and sequentially, valid states of the model might never be reached. Consider as an example a case where there are four records, w, x, y, and z, with (w, x) and (y, z) constrained to be coreferent pairs. Now suppose that the system is in a state where w and x share block b and y and z share block c.

If the constraints are incorporated into the probability model, then resampling the block membership of w alone would result in it joining x in b with probability 1; it would be impossible for w to join block c, since that would violate the constraints. Of course, the same is true for x and an analogous situation occurs with records y and z. In this case, it is impossible for the system to reach the state where all the records are in the same block, thus violating the ergodicity constraint of MCMC sampling.

sampling with positive constraints

In the presence of positive constraints, the Gibbs sampling algorithm is modified to resample the block memberships of a set of known matches *simultaneously*. If a set of records are known to be matches, then their membership is normalized by the sum of joint probabilities of their joining existing blocks plus the probability of a new block being created.

4.3 Estimating a Partition from Samples

By the strong law of large numbers, an empirical distribution estimated from samples approaches the generating distribution almost surely in the limit of infinite samples. Thus MCMC samples serve as an estimate of the posterior distribution of the coreference partition under the model. However, in a practical system it is often desirable to generate a single, or *point*, estimate of the partition.

MAP estimate

In Bayesian statistics, the *maximum a posteriori* (MAP) estimator is commonly used to estimate an unknown variable. This estimator is defined as the mode, or point with the greatest posterior probability. In a Monte Carlo setting, a direct way to approximate the MAP estimate would be to find the empirical mode, or partition which occur most often in a set of approximately independent samples from the posterior.

However for high dimensional variables, such as partitions, the number of posterior samples necessary to find a good MAP estimate typically grows exponentially with the number of dimensions. Consider sampling a sequence of independent, but biased, coin flips: the probability of obtaining the mode as a sample, or even two identical samples, decreases exponentially with the length of the sequence. An analogous situation occurs in duplicate detection where there are a large number of independent points of ambiguity in a data set; it is unlikely that *every* ambiguous group will be sampled in its most likely configuration.

Simulated annealing is a technique that can mitigate this problem [57]. In simulated annealing, a "temperature" parameter is added to the distribution that, at large settings, permits transitions that are unlikely under the unmodified distribution, allowing the system to "explore" the state space more freely. At small settings, the parameter penalizes unlikely transitions and favors likely states. By decreasing the temperature gradually according to an annealing schedule, samples with high posterior probability may be obtained.

MAP versus error

While approximate MAP estimates may be suitable for use in an application, they are not directly tied to other measures of performance such as error, precision, or recall. In particular, it can be seen that with a Ewens prior over the partition, a MAP estimate is not guaranteed to minimize pairwise error.

Consider the counterexample of a pathological data set in which no record values are observed; in this case, the posterior distribution $\mathrm{P}(B \mid V)$ is simply the prior $\mathrm{P}(B)$. By the symmetry of the Ewens distribution, there are a large number of modes or MAP partitions, each with the same assortment of block sizes but differing by a permutation of the elements.

Also, in the Ewens distribution the marginal probability of any two items sharing a block is the same. If this marginal probability is less than 0.5, then any matches declared in a MAP partition are expected to be in error: the expected error can be decreased by declaring the pair a nonmatch. In this case, the expected error is minimized by the singleton partition.

This example serves mainly to show that the MAP partition might not be the one which minimizes error. However it also exhibits the intuition that, in the absence of distinguishing information, declaring nonmatches is expected to result in fewer errors than letting the prior dominate.

4.3.1 Minimum Expected Pairwise Error Partition

We consider as alternative estimate the partition that minimizes expected pairwise error (MEPE) according to the empirical posterior distribution. As discussed in section 2.4, the pairwise error $\mathcal{E}(\cdot)$ of a partition estimate $\hat{\beta}$ is the number of pairs (i, j) for which $\hat{\beta}$ and the true partition disagree,

$$\mathcal{E}(\hat{\beta}) = \sum_{(i,j)\in\hat{\beta}} (\hat{y}_{ij} - y_{ij})^2$$

Where $\hat{y}_{ij}$ is the binary label of pair (i, j) in $\hat{\beta}$, y_{ij} is the label of the pair in the true partition, and a unit of error is accrued if $\hat{y}_{ij} \neq y_{ij}$ (error may also be expressed as a percentage of pairs).

Pairwise error cannot be used as an objective function for optimizing $\hat{\beta}$ because it requires access to the true partition, obviating the need for optimization. In a Bayesian model, however, a distribution over partitions $P(B = \beta)$ may be used to calculate the expected error of $\hat{\beta}$ in terms of indicator random variables Y_{ij}, taking the value 1 when (i, j) share a block in β and 0 otherwise. The expected pairwise error is then defined to be

$$\begin{aligned}
E\left[\mathcal{E}(\hat{\beta})\right] &= E\Big[\sum_{(i,j)} (\hat{y}_{ij} - Y_{ij})^2\Big] \\
&= \sum_{(i,j)} E\left[(\hat{y}_{ij} - Y_{ij})^2\right] \\
&= \sum_{(i,j)} E\left[\hat{y}_{ij}^2 - 2\hat{y}_{ij}Y_{ij} + Y_{ij}^2\right] \\
&= \sum_{(i,j)} E\left[\hat{y}_{ij} - 2\hat{y}_{ij}Y_{ij} + Y_{ij}\right].
\end{aligned}$$

Considering the possible values of $\hat{y}_{ij}$ separately, we note that

$$E\left[(\hat{y}_{ij} - 2\hat{y}_{ij}Y_{ij} + Y_{ij}\right] = \begin{cases} P(Y_{ij}) & \text{if } \hat{y}_{ij} \text{ is } 0 \\ 1 - P(Y_{ij}) & \text{if } \hat{y}_{ij} \text{ is } 1. \end{cases}$$

Thus we can break out the sum into matched pairs $(i, j)^+ \in \hat{\beta}$ and the disjoint set of nonmatched pairs $(i, j)^- \in \hat{\beta}$, giving the form

$$E\left[\mathcal{E}(\hat{\beta})\right] = \sum_{(i,j)^+ \in \hat{\beta}} (1 - P(Y_{ij})) + \sum_{(i,j)^- \in \hat{\beta}} P(Y_{ij})$$

The MEPE partition β^* is simply the partition that minimizes this quantity,

$$\beta^* = \arg\min_{\hat{\beta}} \sum_{(i,j)^+ \in \hat{\beta}} (1 - P(Y_{ij})) + \sum_{(i,j)^- \in \hat{\beta}} P(Y_{ij})$$

4.3.2 Minimizing Error with Correlation Clustering

It is straightforward to estimate the marginal probabilities $\hat{p}_{ij} = P(Y_{ij} \mid V)$ from a set of approximate samples from the posterior distribution. Finding β^* from these estimates is an instance of the optimization problem known as *correlation clustering* [10, 33, 49].

Correlation clustering is typically described as a graph optimization problem. In the *general* form [33], a problem instance is complete undirected graph $G = (V, E)$ with positive weights w^+_{ij} and w^-_{ij} on the each edge (i, j). The objective is to partition the vertices, minimizing the sum of weights w^-_{ij} for edges that *respect* the partition (i and j in the same block) with the sum of weights w^+_{ij} for edges that *violate* the partition (i and j in different blocks).

Finding the MEPE partition can be cast as an instance of correlation clustering by letting G have one vertex for each record and setting the weights for a pair of records to be

$$\begin{aligned} w^-_{ij} &= 1 - \hat{p}_{ij} \\ w^+_{ij} &= \hat{p}_{ij} \end{aligned}$$

Correlation clustering is NP-hard, with a number of known approximation algorithms [33, 49]. In practice, the difficulty of optimizing a particular instance depends on the degree of confusion in the instance; even in large instances it may be possible to perform this optimization exactly, provided $\hat{p}_{ij}$ is low for most pairs.

To see why this is the case, consider an instance in which $w^+_{ij} < w^-_{ij}$ for all pairs (i, j). It is straightforward to prove that the singleton partition optimizes the instance. Assume to the contrary that a non-singleton partition β' is optimal. Since β' is not the singleton partition, there exists at least one pair (k, l) that share a block b. By removing k from the b, the weights w^-_{kj} for all $j \in b$ are replaced in the sum with the lower weights w^+_{kj}, forming a new partition β'' with a lower value for the objective function, contradicting the assumption that β' is optimal.

An analogous argument can be made for a set of vertices c in which $w^+_{ij} < w^-_{ij}$ for all outgoing edges $\{(i, j) | i \in c, j \notin c\}$. In the optimal partition, no vertices in c will share a block with vertices not in c, since the objective could lowered by separating such vertices from the block. Thus the partition of c may be optimized independently from the remainder of the graph.

This fact leads to an efficient heuristic algorithm for partitioning graphs where most of the edges have the property $w^{-}_{ij} > w^{+}_{ij}$. In the MEPE optimization problem, this property corresponds to $\hat{p}_{ij} < 0.5$. The first step in this algorithm is to partition the records into *confusion blocks*, blocks of records connected by paths of edges (i, j) for which $\hat{p}_{ij} > 0.5$. This operation is simply computing the transitive closure of the relation defined by such edges. Since the vertices in a confusion block c are guaranteed to only be joined with other vertices in c in the optimal partition, confusion blocks may be optimized separately; thus the computational complexity of the problem grows with the size of the largest confusion block, rather than the instance as a whole.

heuristic partitioning algorithm

If $\hat{p}_{ij} > 0.5$ for all edges (i, j) within the block, then the block is *coherent* and guaranteed to in the optimal partition. Confusion blocks with less than three vertices are trivially coherent and in many practical instances, many larger blocks are coherent as well. Coherent confusion blocks are simply added to the partition returned by the algorithm.

Non-coherent confusion blocks are optimized according to their size. For confusion blocks with less than ten vertices, it is practical to optimize through an exhaustive search: all possible partitions of the block are considered and scored and the blocks of the minimum scoring partition is added to the partition returned by the algorithm. These blocks are also guaranteed to be in the optimal partition.

Large confusion blocks are optimized heuristically. One option for this optimization would be to do a local search of some kind, however the MCMC algorithm already perform such a search in generating samples; every sample already contains within it a likely partition of the confusion block. To exploit this search, the algorithm partitions large confusion blocks by scoring just the partitions generated by the sampler and adding the one with the lowest score to the partition returned by the algorithm.

This algorithm is simple to implement and worked quickly on all the data sets used in the experiments of chapter 6; It does have the flaw that using partitions produced through sampling offers no guarantee about the approximation returned. In a practical implementation, existing approximation algorithms or more sophisticated searches could be used for large confusion blocks to provide such guarantees.

Chapter 5

Query Ranking

To know what we know, and know what we do not know, that is understanding.

Confucius

Errors using inadequate data are much less than those using no data at all.

Charles Babbage

Previous chapters describe a Bayesian model for coreferent record generation and an efficient inference procedure for the model that takes advantage of pairwise coreference labels as observations. This chapter incorporates this model into the active framework described in Section 1.2. Unlike prior active approaches [87, 79, 16], in the framework described here the testing and training corpuses are the same; this approach is motivated by the goal of domain-independent duplicate detection with minimal test-time labeling.

In this framework, the algorithm takes as input an unlabeled corpus of records and Bayesian inference is used to generate an initial estimated coreference partition as well as a *query*, a portion of the test corpus to be labeled by a supervisor. These labels, or *query responses*, are then fixed in the model as observed variables and conditional inference is used to produce further estimated partitions and queries.

Active approaches are motivated by the observation that labeling data requires human time and effort, and that computation may be used to identify the portion of the data where that effort may best be applied. One challenge in duplicate detection is that a randomly selected pair of records is typically a trivial nonmatch, and therefore will not provide the inference algorithm with useful information for improving the estimated partition. Thus, to be effective, some method of ranking queries by expected utility is necessary.

mutual information ranking

This chapter presents an information-theoretic framework for ranking active queries based on maximizing the *mutual information* between the query response and the unobserved coreference partition. We consider the scenario in which queries consist of a a pair (i, j) of records to be labeled with a coreference label that indicates whether the pair is a match. The label Y_{ij} is treated as an indicator random variable, and we define the optimal query to be that which maximizes the mutual information $\mathrm{I}(Y_{ij}; B)$ between the query and the partition under the posterior distribution of B.

We demonstrate that the Maximum Mutual Information (MMI) query q is the one whose label has the greatest entropy under the distribution, i.e., the query whose label has marginal probability $\mathrm{P}(Y_q)$ closest to 1/2. This result generalizes to any binary query of the true partition and has the natural interpretation that the maximum information of a binary random variable is 1 bit. It also provides a connection between active learning in a partition-based architecture and active learning using pairwise discriminative methods, as explained in section 5.2.

5.1 Maximizing Query Information

A distribution $\mathrm{P}(B)$ over partitions has as its sample space the set $\mathbb{B}$ of all partitions.[1] Although a natural interpretation of a binary query q is as a predicate on $\mathbb{B}$ (i.e., a function that is either true or false for any particular partition β) an equivalent interpretation of a query is as a stochastic

[1]For clarity we disregard the cardinality n of the set underlying the partition as it is irrelevant to this derivation

event: the subset of the partitions that *respects* q.

For example, consider a query (i, j). The response to this query is true for all partitions in which i and j are in the same block and false otherwise, partitioning $\mathbb{B}$ into two disjoint events: $q = \{\beta \in \mathbb{B} : b_i, b_j \in \beta \text{ and } b_i = b_j\}$ and its complement $\overline{q}$. Figure 5-1 shows an interpretation of a query as an event. Note also that a conjunction of binary queries, $q' = q_1 \wedge \ldots \wedge q_k$ is also a binary query which is the intersection of its component queries.

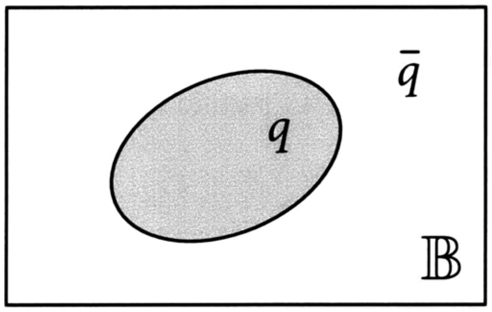


Figure 5-1: A query as an event on the space of partitions

We further define the label Y_q of a query to be an indicator random variable for the event q in $\mathbb{B}$. Thus, a distribution $\mathrm{P}(B)$ over partitions also defines a distribution $\mathrm{P}(Y_q)$ for any binary query q in the query space $\mathbb{Q}$, which is the power set of $\mathbb{B}$.

Interpreting query labels as random variables allows a Bayesian interpretation of the relationship between the label and the partition as a whole. As discrete random variables, B and Y_q have well-defined entropies and conditional entropies according to $\mathrm{P}(B)$. Their mutual information is defined to be $\mathrm{I}(B; Y_q) = \mathrm{H}(B) - \mathrm{H}(B|Y_q)$.

Another quantity of interest is the conditional distribution $\mathrm{P}(B \mid Y_q)$ over partitions conditioned on a query label. Since every sample β is an element of either q or its complement, it is straightforward to show that this conditional distribution can be defined in terms of the marginal probabilities for each case, a fact used in the following derivation:

$$P(B = \beta \mid Y_q) = \frac{P(B = \beta \wedge q)}{P(Y_q)} = \begin{cases} P(B = \beta)/P(Y_q) & \text{if } \beta \in q \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

We now turn to the primary result of this chapter: Given $\mathbf{Q} \subseteq \mathbb{Q}$, we define the optimal query q^* to be the one that maximizes the mutual information between the query label and the partition. We use the definitions of mutual information and conditional entropy to see that the optimal query is also the one with the maximum expected decrease in posterior entropy. In other words, it is the query whose label is expected to most decrease the uncertainty about the partition as a whole:

$$\begin{aligned} q^* &= \operatorname*{arg\,max}_{q \in \mathbf{Q}} \ I(B;\, Y_q) \\ &= \operatorname*{arg\,max}_{q \in \mathbf{Q}} \ H(B) - H(B|Y_q) \\ &= \operatorname*{arg\,max}_{q \in \mathbf{Q}} \ H(B) - \mathop{E}_{y}\left[H(B|Y_q = y)\right] \end{aligned}$$

We express this expectation in terms of q and the base-2 logarithm, lg:

$$\begin{aligned} q^* &= \operatorname*{arg\,max}_{q \in \mathbf{Q}} \ H(B) - [P(q)H(B \mid q) + P(\overline{q})H(B \mid \overline{q})] \\ &= \operatorname*{arg\,max}_{q \in \mathbf{Q}} \ H(B) + P(q) \sum_{\beta \in \mathbb{B}} P(\beta \mid q) \lg (P(\beta \mid q)) \\ &\qquad\qquad + P(\overline{q}) \sum_{\beta \in \mathbb{B}} P(\beta \mid \overline{q}) \lg (P(\beta \mid \overline{q})) \end{aligned}$$

Using $P(B = \beta \mid Y_q)$ as defined in equation 5.1, we limit the bounds of the sums to values with nonzero probability,

$$
\begin{aligned}
q^* &= \arg\max_{q \in \mathbf{Q}} \; \mathrm{H}(B) + \sum_{\beta \in q} \mathrm{P}(\beta)\big[\lg(\mathrm{P}(\beta)) - \lg(\mathrm{P}(q))\big] \\
&\qquad + \sum_{\beta \in \overline{q}} \mathrm{P}(\beta)\big[\lg(\mathrm{P}(\beta)) - \lg(\mathrm{P}(\overline{q}))\big]
\end{aligned}
$$

then combining the sums over $\mathrm{P}(\beta)\lg(\mathrm{P}(\beta))$, we subtract out $\mathrm{H}(B)$ and simplify:

$$
q^* = \arg\max_{q \in \mathbf{Q}} \; -\lg(\mathrm{P}(q)) \sum_{\beta \in q} \mathrm{P}(\beta) - \lg(\mathrm{P}(\overline{q})) \sum_{\beta \in \overline{q}} \mathrm{P}(\beta)
$$

Since the sum of $\mathrm{P}(\beta)$ in an event is just the probability of that event, this simplifies to

$$
\begin{aligned}
q^* &= \arg\max_{q \in \mathbf{Q}} \; -[\mathrm{P}(q)\lg(\mathrm{P}(q)) + \mathrm{P}(\overline{q})\lg(\mathrm{P}(\overline{q}))] \\
q^* &= \arg\max_{q \in \mathbf{Q}} \; \mathrm{H}(Y_q)
\end{aligned}
$$

$\square$

Thus the mutual information between a binary query and the partition does not have an explicit dependence on the joint distribution $\mathrm{P}(B, Y_q)$ but only on the marginal distribution, $\mathrm{P}(Y_q)$. Since Y_q is a binary random variable, its posterior entropy can be efficiently estimated from a small number of samples by estimating the probability of the query $\hat{p}_q$ as the empirical ratio of samples which respect the query to the number of samples taken.

5.2 Connections to Query By Committee

Choosing to label portions of the data with high uncertainty in their classification is common in approaches to active learning in which the data to be labeled are independent points in some input space. However, when the data to be labeled takes on a more complex structure, such as a partition, it is not always clear how the uncertainty of a portion of the structure relates to the uncertainty of the structure as a whole.

The result of the preceding section clarifies the relationship between pairwise coreference labels and the complete coreference partition, providing a connection between active learning in pairwise duplicate detection architectures and the active partition-based approach of this thesis. In fact, ranking queries on a partition by mutual information may be seen as a variant of the *query-by committee* paradigm (QBC) [81] used to train discriminative models in an active setting.

query by committee

The QBC framework was developed in the context of the following learning problem: A learner is tasked with finding a target function t^* from some concept class $\mathcal{C}$ of binary functions over an input space $\mathcal{X}$. The learner is presented with a stream of unlabeled points x_i from $\mathcal{X}$ for which it may request the associated binary label y_i. The goal of the learner is to identify t^* with as few label requests as possible.

In one version of QBC, the previously requested labeled points T are used to restrict $\mathcal{C}$ to just the concept classes which respect the labeling of those points $\mathcal{C}_T$. Then for each new point x_i in the stream, two hypothesis functions are drawn at random from $\mathcal{C}_T$. If these functions disagree on the labeling for x_i, its label y_i is requested.

It is shown by Freund et al. [40] that, under certain assumptions, this QBC algorithm exponentially reduces the number of labeled points required to achieve a certain level of classification accuracy when compared to randomly sampling points from the stream. An intuitive interpretation of this result is that the QBC algorithm requests labels which approximately divide the space of remaining hypotheses in half.

One challenge with QBC algorithms lies in practically sampling from the space of restricted hypotheses $\mathcal{C}_T$. A typical approach in discriminative learning, from which the technique derives its name, is to train a *committee* of binary classifiers using some form of randomness. For example, different subsets of a 'seed' collection of training examples may be used to train a committee of classifiers, which then serves to approximate a statistical distribution over hypotheses.

QBC in pairwise architectures

This approach has been used in active duplicate detection by Tejada et al. [87] and Sarawagi and Bhamidipaty [79]. In these methods, record pairs are represented as independent points x_i in a input space, each with a binary coreference label y_i indicating whether the pair is a match. By treating the pairs as independent training data points and training a committee of classifiers, QBC provides a framework for choosing new points to label based on the uncertainty of the committee.

The connection between pairwise and partition-based labels presented in this chapter further justifies using QBC in pairwise duplicate detection architectures. However in pairwise methods, the relationship between the classifier class and the hypothesis space of partitions isn't well characterized. In general, a particular pairwise classifier is not constrained to generate a consistent partition of the data that respects transitivity[2] nor is it guaranteed that a particular class of classifiers is able to represent every possible partition of the data.

In contrast, partition-based models of coreference make the connection between the hypothesis class and the observed record values explicit, by considering only consistent partitions as hypotheses and allowing information from blocks of records to be used in determining the posterior probability of each hypothesis. Making the hypothesis space explicit also ensures that every possible hypothesis is representable; that is, as long as the partition-based model is able to represent every possible partition, then valid hypotheses are not being eliminated *a priori* due to the construction of the model.

Of course, in neither pairwise nor partition-based models is it possible to consider every possible hypothesis in practice. Since labeled data is scare in active learning settings, a pairwise classifier may be more efficient in practice, in terms of number of pairwise labels, then either a discriminative or generative partition-based model.

[2]unless coupled with a linkage procedure, such as single or average linkage

conditional inference as search

Bayesian models also give a natural interpretation of active queries as a weighted search through the hypothesis space of all partitions. In this interpretation, each binary query divides the hypothesis space into two portions. A Bayesian model provides a measure over these portions that depends on the observed data, namely the posterior probability of each portion.

Thus conditional inference given a set of observed query labels is the same as restricting the hypothesis class to partitions consistent with the observations. It is in this sense that MMI query ranking may be seen as a form of QBC: during each round of querying, the algorithm choses the query which approximately divides the remaining probability mass in half.

This objective of maximizing the expected information gain from queries has also been proposed by Dagan and Eggleston [32], who provide an interpretation of QBC for learning Bayesian classifiers. In this method, called *committee based sampling*, independent data points are presented sequentially and the query label for a point x_i is requested according to a biased random decision. The bias for the decision is determined by the entropy of the label y_i for the point as determined by a committee of Bayesian classifiers sampled from a posterior distribution over classifiers conditioned on previously observed labels.

5.3 Limitations of the Approach

While mutual information provides a theoretically attractive basis for choosing informative queries, there are some limitations that must be considered in its practical application. Perhaps the most significant is that the effectiveness of queries will be limited by the degree to which the Bayesian model truly captures the uncertainty about the coreference partition of real data. Bayesian models are merely models and are not guaranteed to be useful when applied to data that violates their assumptions.

For example, consider a model which places the entirety of its posterior probability mass on a single partition. In this case, the posterior distribution has zero entropy and thus no information is expected to be gained by any query. More generally, if a model has no uncertainty in the label of a particular pair of records, considering them either a match or nonmatch with probability one, then a query about that pair will placed at the bottom of the ranking. If the model was wrong about the labeling, then it will not correct that mistake until all other queries are exhausted.

Another limitation of this approach is that, even if the posterior distribution is correct, the query that maximally reduces posterior entropy is not guaranteed to maximally reduce other objective measures of performance, such as pairwise error. Consider the example in Figure 5.3, in which there are four possible partitions of seven records, each with equal probability of 1/4.

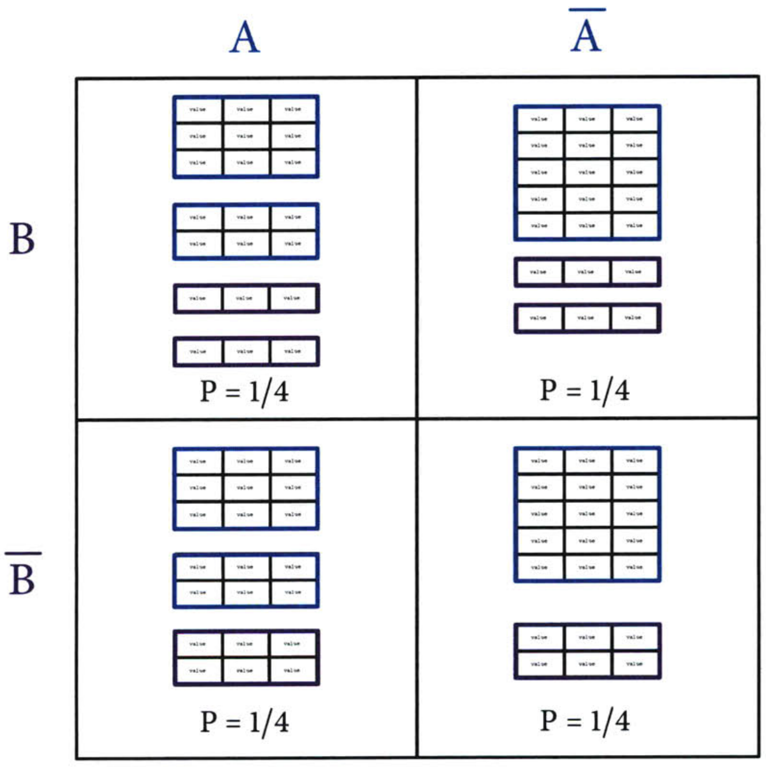


Figure 5-2: An example of independent coreference uncertainty

In this example, there are two independent sources of uncertainty, the events A and B. In the event A, a group of five records is split into two blocks while in its complement Ā, the records are in a single block.

The event B is analogous, but with a group of two records. The marginal probability of queries associated with each of these events, Y_{A} and Y_{B}, is the same, 1/2, and thus each query reduces the entropy by 1 bit. However by any measure of error that depends on the size of the blocks, the label Y_{A} is expected to have a greater reduction in error than Y_{B}.

Thus maximizing an expected reduction in error might serve as an alternate objective for ranking active queries. However, one issue with the straightforward application of this approach is that it requires computing the value of the objective function for each possible outcome of each possible query.

This computation is necessary because the marginal probability of one pairwise label will generally change when conditioned on another; if the probability remained the same, then the labels would be statistically independent and the labels would provide no information about one another. To be feasible, a query ranking criterion requires an efficient way of optimizing the objective, either exactly or approximately, across the space of possible queries.

In practice such a computation may not be intractable. Though computing the expected decrease in error is computationally expensive, it may only need to be done for a relatively small number of queries: in most applications, most pairs of records are not coreferent, so a reasonable Bayesian model should place low probability on such records being nonmatches. In this situation, the expected reduction in error given a query may be bounded using its marginal probability.

For example, if *none* of the posterior samples declares a pair (i, j) coreferent, then the estimated probability of a match $\hat{p}_{ij}$ is zero (or near zero, depending the estimator). In this case, the expected decrease in posterior error given the label Y_{ij} is also near zero. Informally, because the model is nearly certain that a pair is not coreferent, obtaining the label for the pair is expected to leave the estimated partition unchanged.

Thus, an expected error minimization ranking objective would only need to be computed for pairs which the model considered ambiguous, which may be far fewer than the total number of pairs in the data. Of course, if the model fails to identify true matches as possible matches, then these pairs will not be ranked highly and therefore unlikely to be corrected through an active process.

Reduction of error as a loss function for active learning has been proposed by Cohn et al.[28] and also Roy and McCallum [78]. Minimizing error has also been used as objective function in training a partition-based author coreference model by Culotta et al. [30]. Applying this idea to active query ranking in duplicate detection is a promising direction left to future work.

Chapter 6

Experimental Evaluation

In the computer field, the moment of truth is a running program; all else is prophecy.

Herbert Simon

To conjecture and not to test is the mark of a savage.

George Pólya

Previous chapters present a complete method for active duplicate detection: a Bayesian model of the uncertainty of the coreference partition of a data corpus including MCMC sampling procedure, a criterion for ranking queries based on mutual information, and an algorithm for estimating a partition from samples based on minimizing expected pairwise error.

This chapter reports experimental results validating this technique in a variety of data domains. Using the same model for each domain, the experiments presented here look at four practical considerations: the importance of incorporating negative constraints when merging data from multiple sources; the performance of the proposed query ranking criterion compared to baseline algorithms; the performance characteristics of the IVD and SVD model variants; and the effect of periodically reranking queries through additional rounds of inference.

6.1 Implementation Details

The MCMC sampling algorithm described in chapter 4 was implemented the Python programming language using PyMC, a library for MCMC sampling [68]. PyMC provides a framework for running sampling loops, a collection of standard statistical distributions, and extension points for creating custom distributions and Metropolis-Hastings and Gibbs step methods.

These extension points were used to implement custom PyMC objects for the Chinese Restaurant Process, Pitman-Yor Process, and the likelihood models for field values under these processes. Because MCMC sampling is computationally intensive, critical sections of the model were optimized in C. The CRP and Dirichlet Process parameters were sampled using Metropolis step routines provided by PyMC.

preprocessing

Record field values were minimally preprocessed: in each domain, value strings were tokenized using whitespace and punctuation as delimiters and the resulting alphanumeric tokens were case-normalized. No other preprocessing was done, including stemming, address normalization, stop word removal, or removing TeX commands for sources derived from BibTeX.

To quickly compare the likelihoods of arbitrary sets of strings under the Dirichlet Process, each string value is processed into a sorted array of integers indexing the tokens of a string. In this representation, it is possible to efficiently count the token overlaps between two arbitrary sets of strings in a single pass through each array. Sets of strings can also be efficiently merged in a single pass, as in the merge step of merge sort.

MCMC sampling

The model state is initialized with every record in its own coreference block. In the SVD variant, the value-generating distributions are initialized so that no distributions are shared: that is, each block has its own value-generating distribution for each field. The parameters of the model are each set to a value drawn from their respective prior distributions.

During each iteration of sampling, every random variable in the model is updated once. With the partition(s) held fixed, the model parameters are first sampled sequentially using Metropolis steps. Then, in random order, the block membership of each record is sampled conditioned on the state of the remaining records. In the SVD model, when a record is placed into an empty block, its value distribution attachment is resampled at that time, conditioned on the state of the other value distribution attachments.

convergence properties

In practice, this sampling routine was found to converge very quickly, reaching an equilibrium state in as few as three iterations, particular in data sets with small blocks. Once an equilibrium state was reached, however, the system was unlikely to reach a state that was much different. An intuitive explanation for this is that, when resampling an individual record, it is likely return to the block it was previously assigned to. For this reason, the Markov chain was reset to the original (singleton) state after every sample, rather than mixing from that state as is commonly done in MCMC sampling. This has the added benefit of not having to choose a "thinning" parameter for the MCMC chain, as is commonly done.

parameter settings

The only free parameters in the model are in the prior distributions over the CRP and PY process parameters. The same priors were used for both the coreference partition and the partition of the per-field shared value distributions. For the strength parameters, the heavy-tailed log-normal distribution with standard deviation of 3 was used; this distribution was found to approximately recover the value of the strength parameter from CRP samples with known strength values across a wide range (0.5–10000). For the discount parameter, a uninformative uniform distribution over [0, 1] was used. For the per-field Dirichlet Processes strength parameters, an exponential prior with a parameter value of 1 was used.

Informal experimentation with these settings showed the model behavior was bimodal. When parameter values were low (< 10000), the evidence provided the data dominated the model behavior and inference was largely insensitive to the parameter values. When set very high, the prior dominated and the data was essentially ignored. This finding is similar to experiences reported in other work involving nonparametric Bayesian models [84, 46].

6.2 Modeling Active Queries

In order to characterize its performance, the MMI query ranking criterion was compared empirically to three other rankings, called **random**, **alternating**, and **idealized.** In all four cases, interaction with a human supervisor was simulated by allowing the algorithm access to the query labels from a completely labeled coreference partition.

Two baseline rankings were used. The simplest of these is the **random** ranking, where the possible queries are ranked in random order. This models a situation random pairs are picked by the supervisor with no separate ranking stage. This is a very weak baseline, however, because such pairs are very unlikely to be matches in most duplicate detection instances and positive examples of coreference are much more salient to training than negative examples.

A stronger baseline is the **alternating** ranking, in which queries are determined from the estimate partition without regard to the pairwise marginals. In this ranking, the estimate partition is used to alternate between drawing a random pair that is declared a match and a random pair that is declared a nonmatch, thus giving the system a sample of likely matches and nonmatches from which to improve its estimate. This models a situation in which a heuristic algorithm is used to pick matches and nonmatches from which to train, similar to the "static" active approach of Bilenko [16].

As a gold standard, the **idealized** ranking was used, in which the true labels are used to produce queries. After each round of inference, the estimate partition is compared to the true partition and the pairs that are incorrectly labeled are placed in random order at the top of the query queue. This models the situation where a human supervisor picks the query by choosing a pair that the system labeled incorrectly.

periodic reranking

Inference on large data sets is computationally intensive, so for data sets with more than 300 records queries are asked in batches of 10 or 20. Instead of alternating between a round of inference and receiving a single query label, after each round of inference the queries are ranked and the labels for the top 10 or 20 are provided to the system. The experiments described in section 6.4.4 explore the effect of reranking queries frequently or in larger batches.

handling transitivity constraints

In some of the query ranking algorithms described it is possible to rank highly a query whose response is determined from previous responses due to transitivity. Such queries are removed from the ranking and never asked. For example, previous response may have established the coreference of record pairs (x, y) and (x, z) and thus (y, z) is implied by transitivity. The pair (y, z) is then removed from the ranking. Note that the MMI algorithm automatically ranks such queries at the bottom, since inference would assign (y, z) as a match with probability 1, and thus its label provides no information about the partition.

6.3 Experimental Domains

Because the model presented in this thesis is intended to be domain independent, data from a number of different domains were used in evaluation. These data sets, along with their sources and some statistics, are presented in figure 6-1. The *joins* column indicates the number of pairwise join operations required to obtain the correct coreference partition, giving a sense of the prevalence of coreference. Though these data sets vary by domain and the number and type of fields, all use English as the primary language and generally had short (<15 token) field values.

Two of the sets, *vauniv* and *games*, are provided by the SecondString project [1]. Secondstring focuses on approximate string-matching techniques, so these data sets consist of a partition of coreferent strings which are treated here as a single field database table. The *vauniv* data set consists of names and addresses of universities and other institutions in Virginia, while the *games* set consists of names of games.

Two other data sets, *cora* and *learning-publications*, are publication records containing a large number of fields, 12 and 14, respectively. To keep experimental computation manageable, and since the values of most of these fields is empty for most records, only fields which had values for more than 50% of the records were used. This was resulted in a choice of 5 fields each: title, author, year, venue, pages (*cora*) and volume (*learning-*

	Domain	Records	Fields	Joins
cora[1]	publications	1295	5	1183
restaurant[1]	restaurants	864	5	112
karpathos[3]	email addresses	125	3	57
vauniv[2]	institutions	116	1	59
learning-authors[3]	author names	1648	2	948
learning-articles[3]	publications	241	6	40
games[2]	game names	911	1	49

1: RIDDLE Repository 2: SecondString Project 3: personally collected

Figure 6-1: Evaluation data sets

publications).

The *karpathos* data set consisted of email addresses collected from a 10 year archive of the author's eponymous family mailing list. Email addresses were taken to have three fields: a username, a hostname, and a label, the free-form text string included by the sender. An address may generally have multiple labels, which were taken as independent draws from the value-generating distributions of the model. The mailing list is named for the ancestral home of the Matsakis family, the island of Karpathos in Greece. Many family members are named according to the patronymic and matronymic naming traditions of the area and so share both first and last names, making this set particularly challenging.

The *learning-authors* consists of names of authors extracted from the author strings of personal BiBTeX files, with one record being generated per mention. These strings were parsed according to BiBTeX's syntax rules into three fields corresponding roughly to given name(s), surname(s), and a suffix (e.g. "III" or "jr."). Because so few authors had suffixes, the final field was dropped.

6.4 Experimental Results

6.4.1 Effect of Source Constraints

Three of the data sets were created by merging data from clean sources, for which it is assumed that records from the same source are nonmatches. *Restaurant* merges records extracted from Fodor's and Zagat's websites. Since each of these sites was free of duplicates, the maximum block size in *restaurant* is two, with each non-singleton block containing one record from Fodor's listing and one from Zagat's. *Games* also comes from 2 sources, and the *learning-articles* came from the bibtex files of five individuals.[1]

As discussed in section 2.2.1, data from clean sources provides a set of negative constraints that can be used to condition the inferred coreference partition. Though positive constraints are generally considered more useful, negative constraints may still provide useful evidence for inference. In particular, negative examples may highlight areas where superficial similarity should be discounted as evidence towards coreference. They also may improve the model's estimate of the number of latent entities represented by the data.

Using the conditional sampling extensions described in section 4.2, negative constraints are incorporated into inference as if they were labels provided by the supervisor. Figure 6-2 provides a comparison of unsupervised performance on sourced data with the performance on the same data using negative constraints. The MUC f-measure, described in section 2.4, is reported for the MEPE estimated partition under the posterior distribution.

Imposing negative constraints improves the MUC f-measure significantly through substantially increased precision. This is to be expected, since given negative labels can prevent the algorithm from declaring superficially similar pairs coreferent. This is particularly evident in the *games* dataset, which is challenging for this model because many dis-

[1]There are a negligible number of same-source matches in *games* and *learning-articles*; these were not constrained to be nonmatches in experiments.

		MUC f-measure	MUC precision	MUC recall
games	no constraints	0.276	0.165	**0.861**
	constraints	**0.670**	**0.934**	0.522
learning-articles	no constraints	0.519	0.350	**1.000**
	constraints	**0.755**	**0.616**	0.975
restaurant (IVD variant)	no constraints	0.413	0.262	**0.975**
	constraints	**0.796**	**0.713**	0.900
restaurant (SVD variant)	no constraints	0.420	0.267	0.982
	constraints	**0.913**	**0.865**	0.966

Figure 6-2: The effect of applying negative constraints on unsupervised performance of the IVD variant (unless indicated); bolded differences are significant at the 95% confidence level

tinct games names differ only by one word (e.g. `Dogz, Your Computer Petz` vs. `Catz, Your Computer Petz`) or contain non-distinguishing terms such as `by Milton-Bradley`.

Recall is negatively impacted, however. One conjecture for the decrease is that the negative constraints push the nonparametric parameters towards creating more blocks. However, as discussed in section 6.1 the model is largely insensitive to these parameters. A better explanation is that the presence of negative constraints increases the total number of blocks, providing more opportunities for a true match to be separated during sampling. As the pairwise coreference probability for a pair decreases, eventually the MEPE partitioning algorithm prefers to separate them, decreasing recall.

6.4.2 Comparison to Baseline Algorithms

The effectiveness of the MMI querying algorithm presented in chapter 5, it was compared to the **random**, **alternating**, and **idealized** algorithms on each of the data sets described previously using the IVD variant of the model. The results of these experiments are presented in figures 6-3 through 6-8.

Each of these figures shows the performance of the system on a given data set as a function of the number of pairwise labels provided; since all algorithms begin with a round of unsupervised inference, any differences in performance at the leftmost point on the graph are entirely attributable to the randomization of the MCMC sampling algorithm.

Performance was measured using three standard metrics: pairwise f-measure, MUC f-measure, and percentage of blocks correct, described fully in section 2.4. MUC f-measure is not shown for *games* and *restaurant*, since it is the same as pairwise f-measure for these sets. These metrics were averaged across five experimental runs for each algorithm and data set. Error bars depicting the 95% confidence intervals for these averages are also provided.

The number of labels provided with each run varies with the size of the data set, but is roughly 1/4 the number of records. As mentioned in section 6.2, queries were asked in batches for large data sets: the batch size was 20 for *cora* and *learning-authors* and 10 for *games* and *restaurant*.

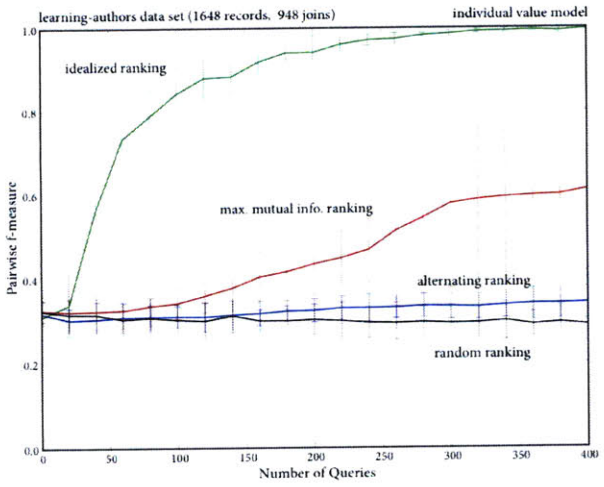
Some general observations are immediately apparent from the graphs. As one might expect, the **random** ranking does not improve performance even after receiving many labels. In fact, the performance of the algorithm typically stays within the confidence interval of its unsupervised performance. This is because the vast majority of record pairs are nonmatches with little or no overlap in their field values. Obtaining the label of such an "obvious" nonmatch has little effect on the posterior distribution and therefore little effect on the MEPE partition. The performance of the **alternating** ranking is more variable. In some data sets, particularly those with relatively large blocks, it performs scarcely better than **random**. For most others, it performs slightly better, but still not as good as the MMI ranking.

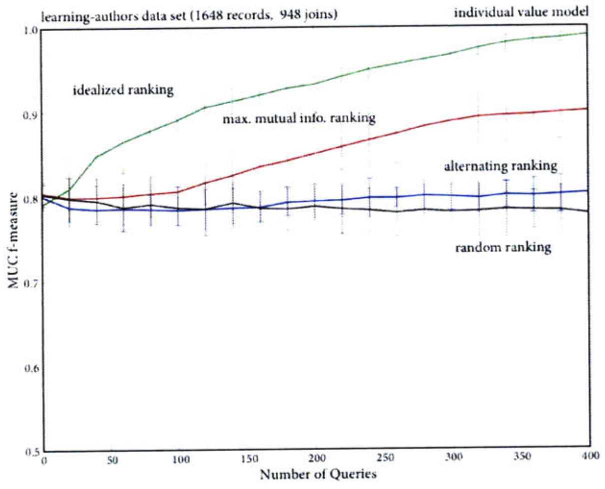
Also to be expected is the fact that the **idealized** ranking generally outperforms the others in all measures by a significant margin, because it has access to the true partition. To the extent that the model doesn't reflect the process which generated the data, there will be "difficult" pairs, pairs for which the model assigns the wrong coreference label with high probability. The MMI ranking will consider such pairs uninformative, and the baseline rankings are unlikely to uncover them because they are relatively rare. The **idealized** ranking, however, will rank such pairs at the top so they will be identified and corrected in relatively few queries. In *vauniv* and *karpathos*, the system was able to achieve perfect performance in short order under the **idealized** ranking.

In most cases, the MMI query ranking performance performed significantly better than the baselines and with a tighter confidence interval as well. However, the performance of the ranking on *cora* and *learning-authors* was equivocal. On both of these data sets, unsupervised pairwise f-measure was poor due to very poor recall; precision on both data sets was above 0.96.

The explanation for this is that both of these data sets have very large blocks relative to the other data sets: the largest block in *cora* is 64 records and the largest block in *learning-authors* is 86. The model tends to break up such blocks into many different smaller blocks during sampling. Since there are so many way to break up a large block, no individual pair is likely to be together in more than half of the samples, causing the MEPE partitioning algorithm to separate all the records into singleton blocks, resulting in very poor pairwise recall.

Though the MMI querying algorithm still performs better than the baselines in these cases, it is not as close to the **idealized** algorithm as in other data sets. One possible explanation for this was that these large data sets had batched updates. Another is simply the fact that the MMI querying algorithm relies on the model being a good fit to the data; if it is not, then there are no guarantees that what is ambiguous according to the model is truly ambiguous.





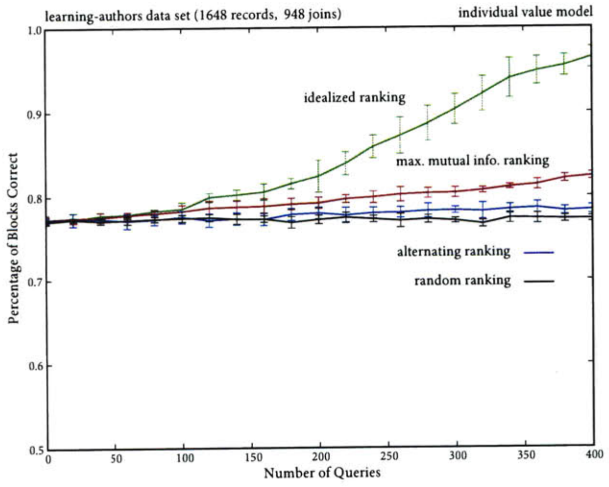
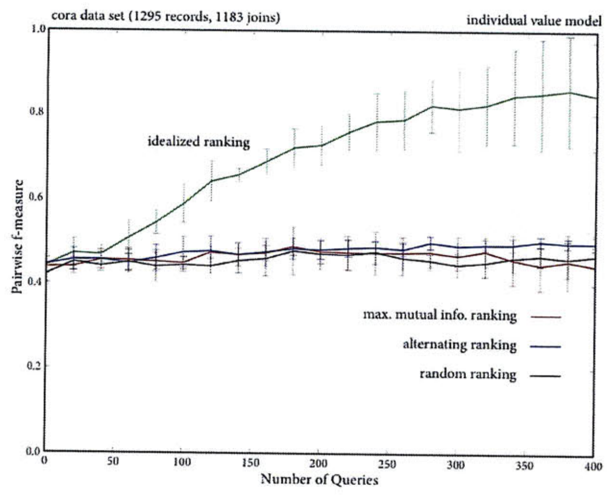
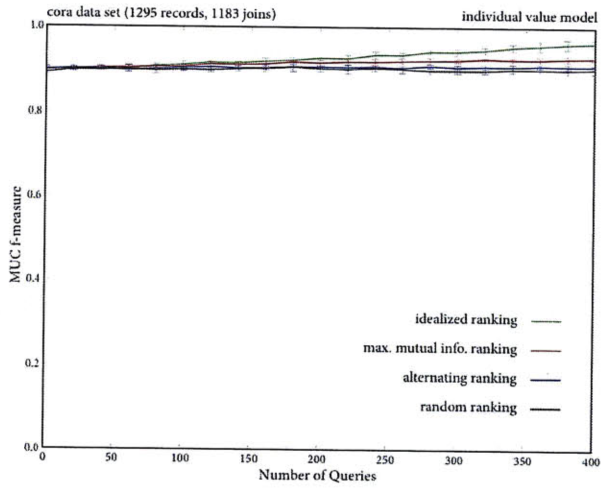


Figure 6-3: Comparison to baseline query ranking algorithms for *learning-authors* (5 runs averaged)





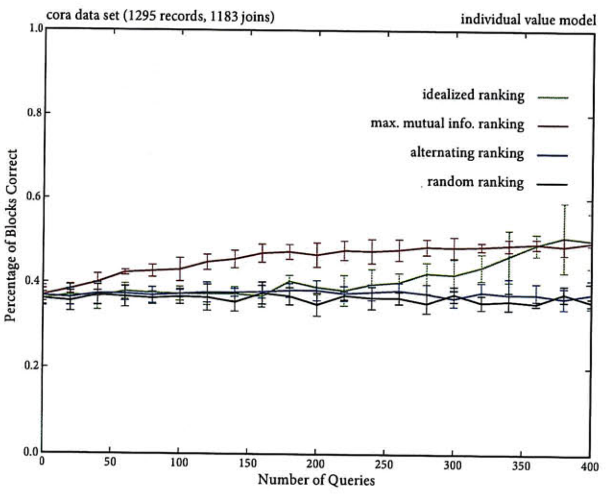
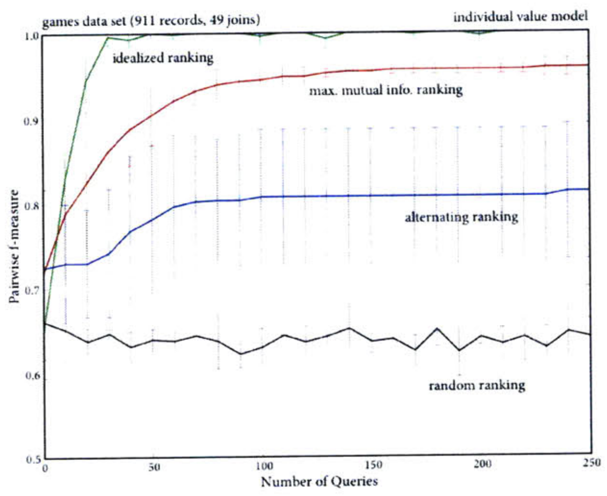


Figure 6-4: Comparison to baseline query ranking algorithms for *cora* (5 runs averaged)



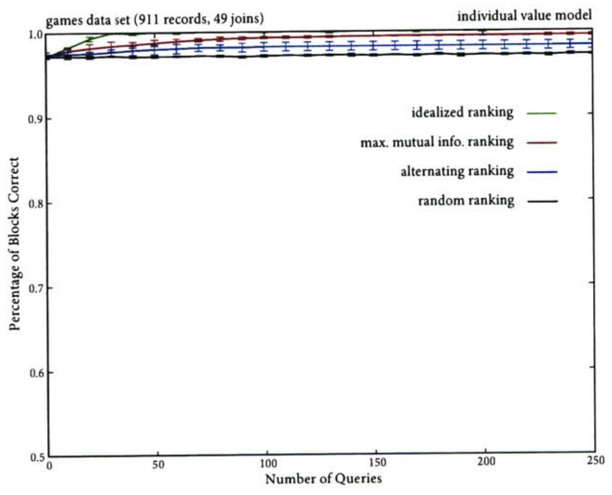
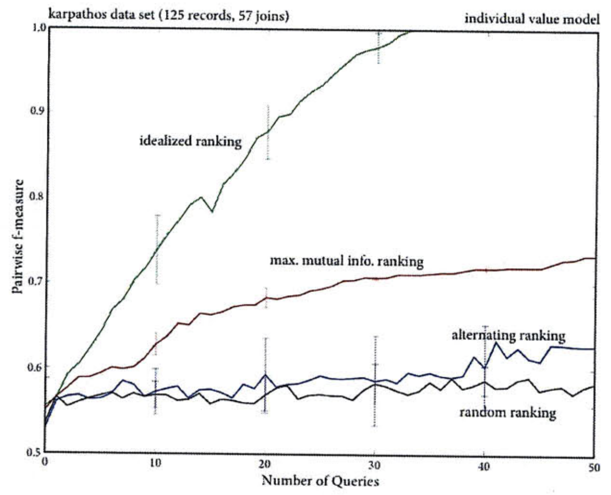
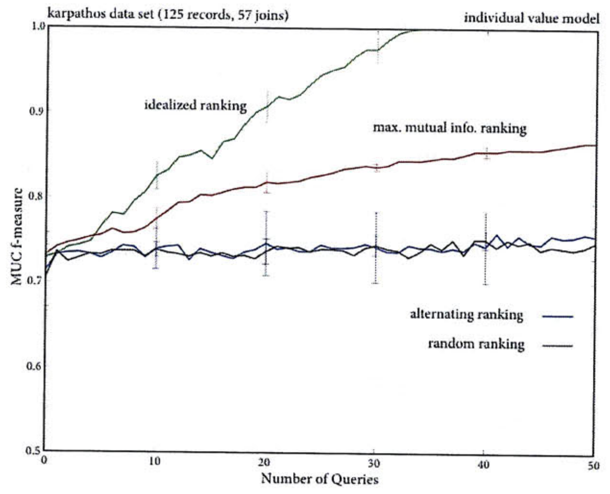


Figure 6-5: Comparison to baseline query ranking algorithms for *games* (5 runs averaged)





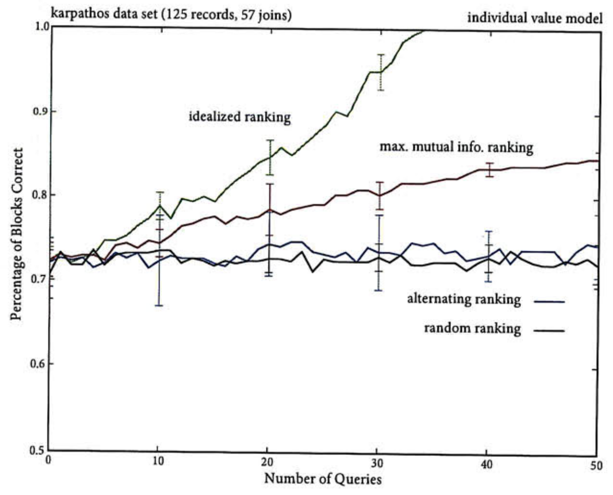
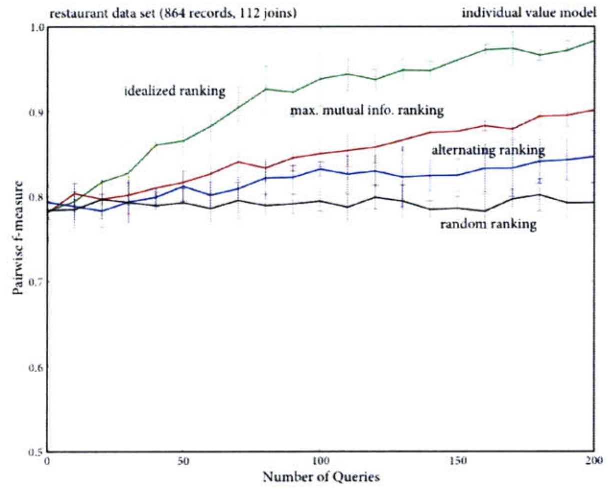


Figure 6-6: Comparison to baseline query ranking algorithms for *karpathos* (5 runs averaged)



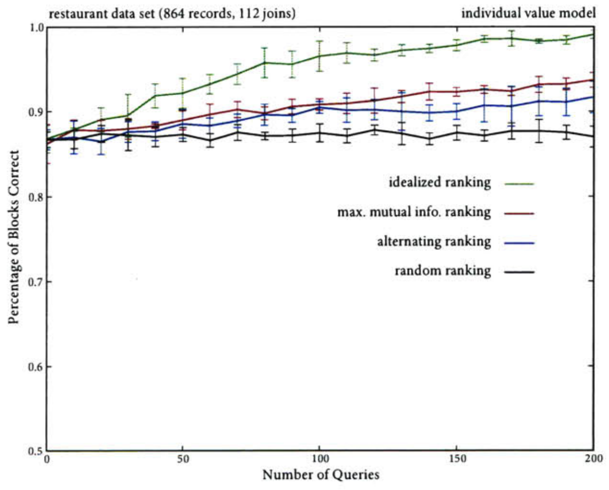
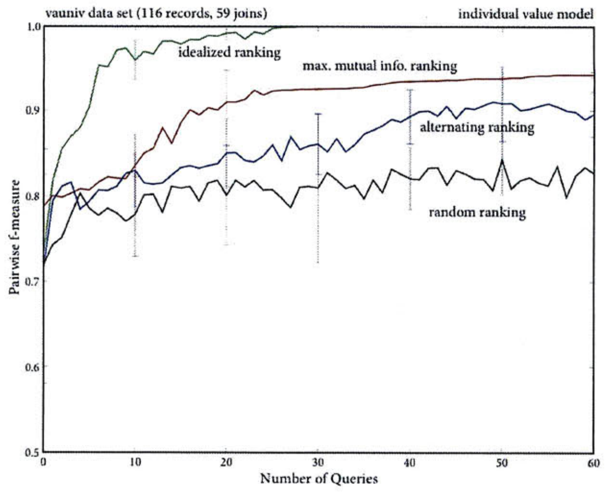
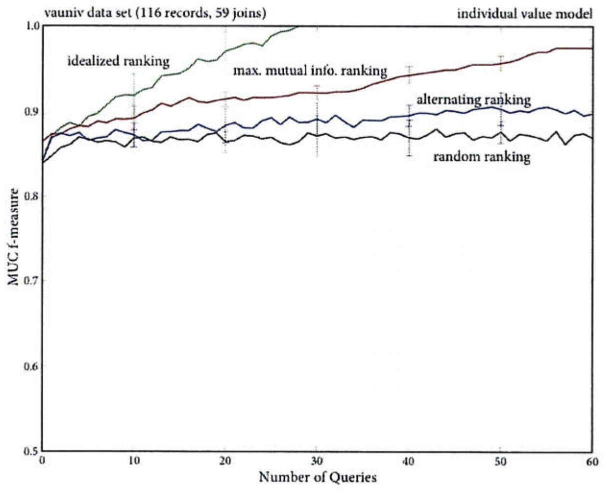


Figure 6-7: Comparison to baseline query ranking algorithms for *restaurant* (5 runs averaged)





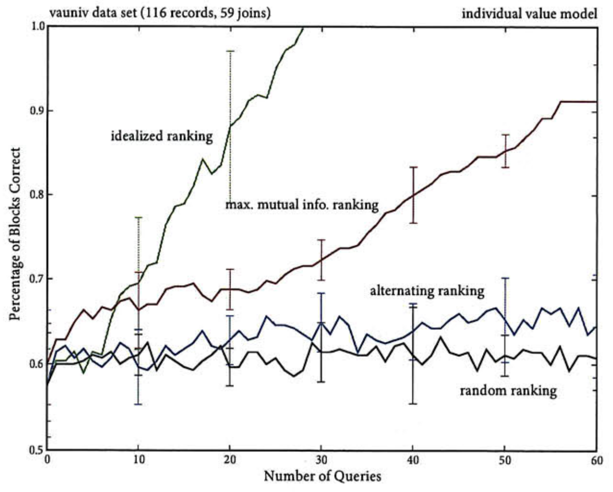


Figure 6-8: Comparison to baseline query ranking algorithms for *vauniv* (5 runs averaged)

6.4.3 Effect of Sharing Value Distributions

The next round of experiments compares the performance of the IVD variant of the model with the SVD variant. These experiments were run in five batches as in the experiments of section 6.4.6, with the same number of queries and batching policy. Only the **mmi** algorithm was used. The results of these experiments are presented in figures 6-9 through 6-14.

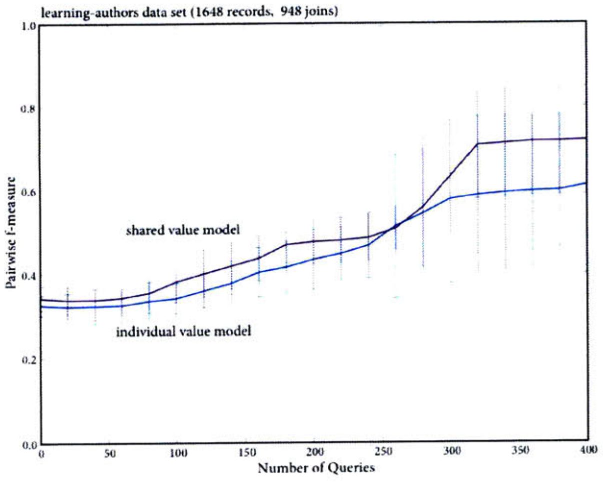
On *restaurant*, SVD variant performed significantly better than the IVD variant, with f-measure increasing from around 0.78 to 0.92 in the unsupervised case. However on the remaining data sets results were mixed, with the SVD variant either performing the same or only improving performance after many labels have been provided.

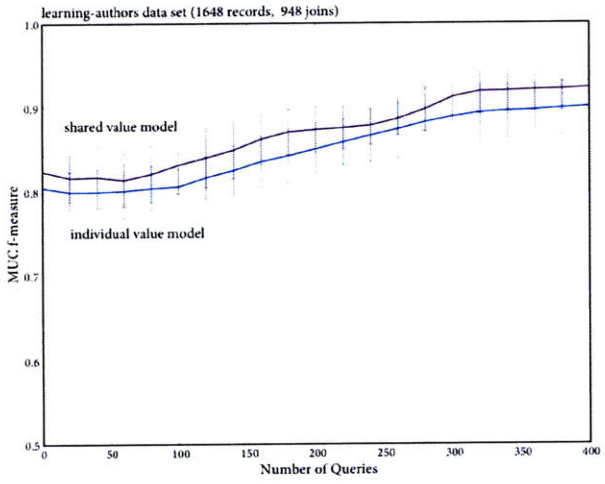
One explanation for the improvement on *restaurant* is that many non-matched records share values for their `city` and `type` fields. Though similar sharing occurs in other data sets, it is not as prevalent as in *restaurant*. Another possible explanation is that this data set has a high correlation between the field values. In particular, the `address`, `city`, and `phone` fields are highly correlated. Two restaurants on the same street in the same city seem very likely to be co-generated to the model, which assumes that fields are generated independently. By allowing non-coreferent values to share generating distributions, the SVD model allows for the possibility of similar records without cogeneration.

Another case that bears discussion is that of the single-field data sets *games* and *vauniv*. In both of these data sets, unsupervised f-measure was zero for the SVD variant. This is because, when only one field value is present, there is no way for the model to distinguish between population variation and observation variation: it might be that similar values are caused by coreference or it might be that they are caused by value distribution sharing.

The samples maintain both hypotheses, but the result is that most true pairs have a relatively low marginal probability of being coreferent, so the MEPE partitioning algorithm conservatively chooses the singleton partition resulting in zero recall. Once there are sufficient labels, however, the SVD variant matches or exceeded the performance of the IVD variant on these single-field data sets.

This leads to the conclusion that, in order to effectively model the distinction between population and observation variation, some type of disambiguating information is necessary. In an active framework where labeled training examples are scarce, a simpler model may improve results.





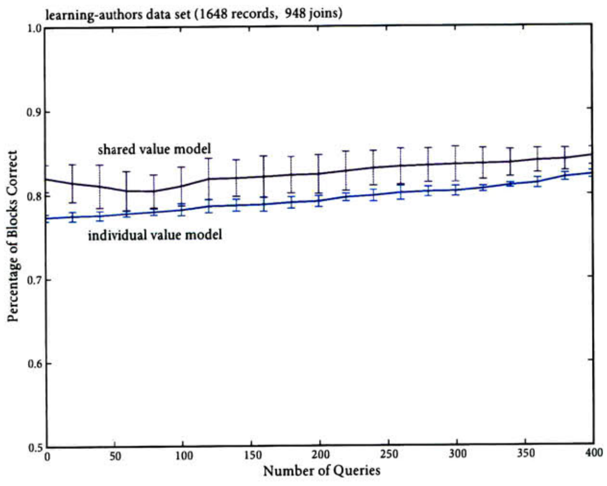
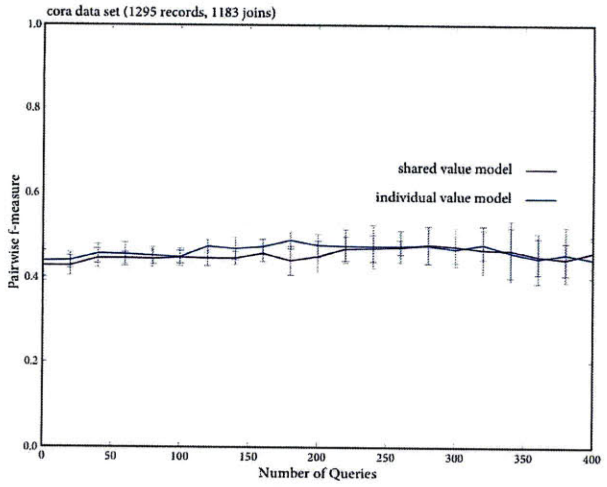
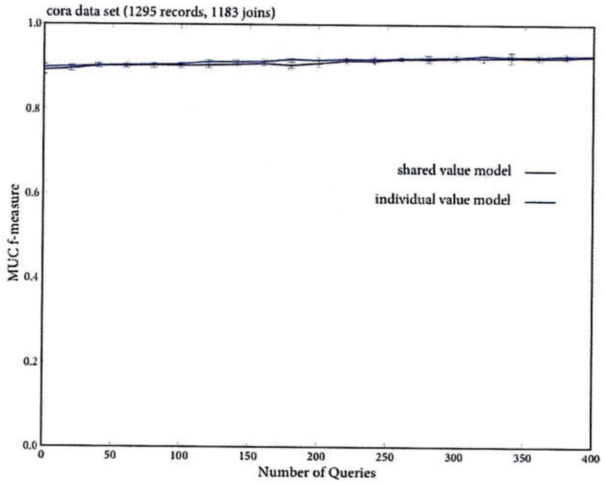


Figure 6-9: comparison between individual and shared value variants on *learning-authors* using MI query ranking (5 runs averaged)





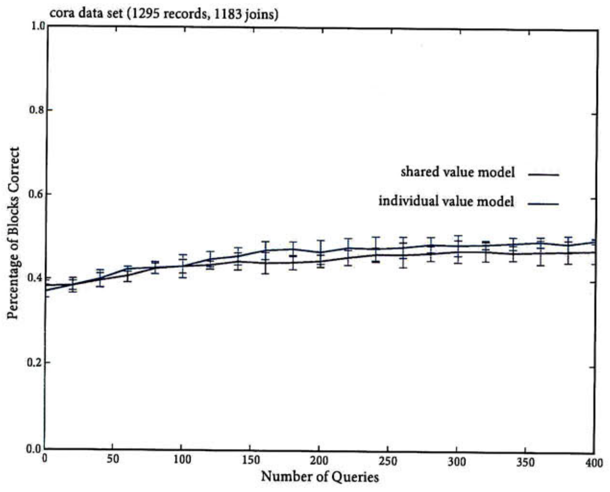
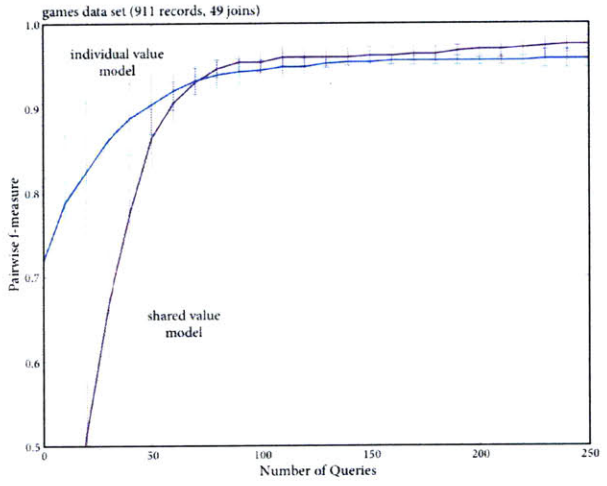


Figure 6-10: comparison between individual and shared value variants on *cora* using MI query ranking (5 runs averaged)



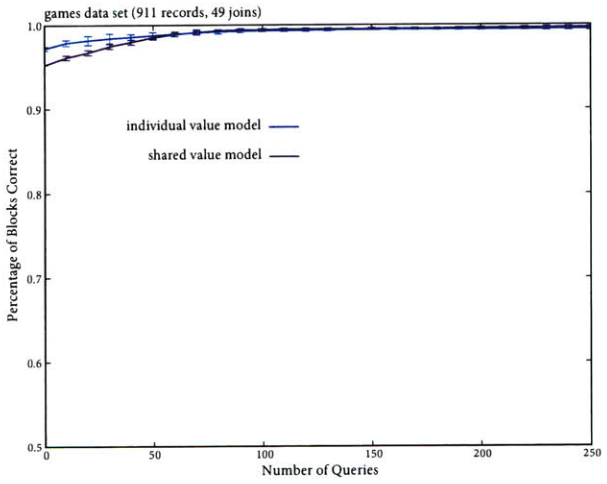
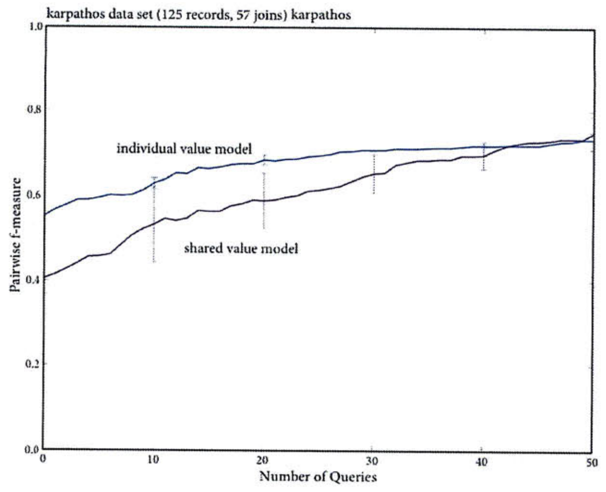
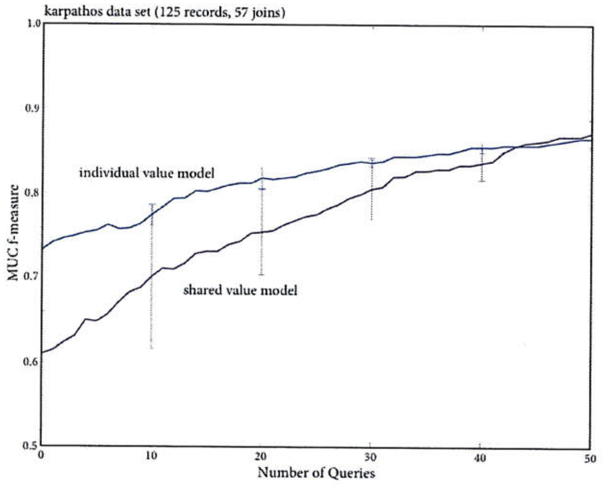


Figure 6-11: comparison between individual and shared value variants on *games* using MI query ranking (5 runs averaged)





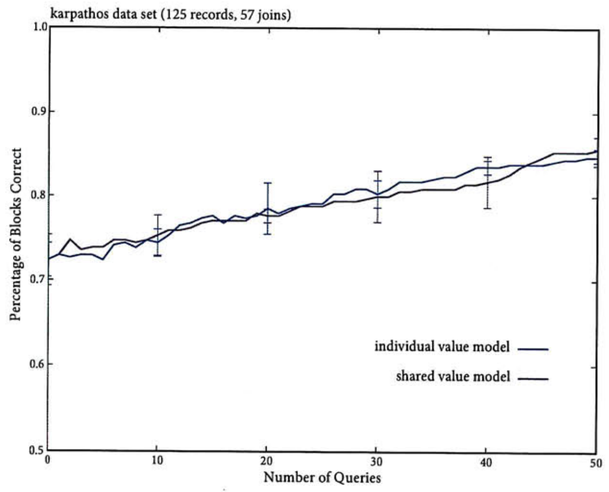
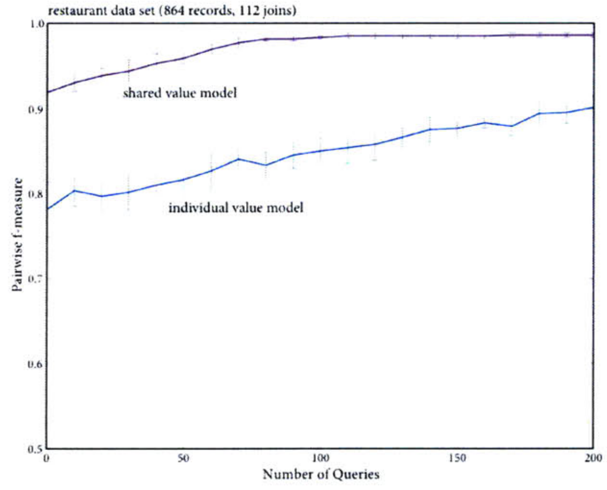


Figure 6-12: comparison between individual and shared value variants on *karpathos* using MI query ranking (5 runs averaged)



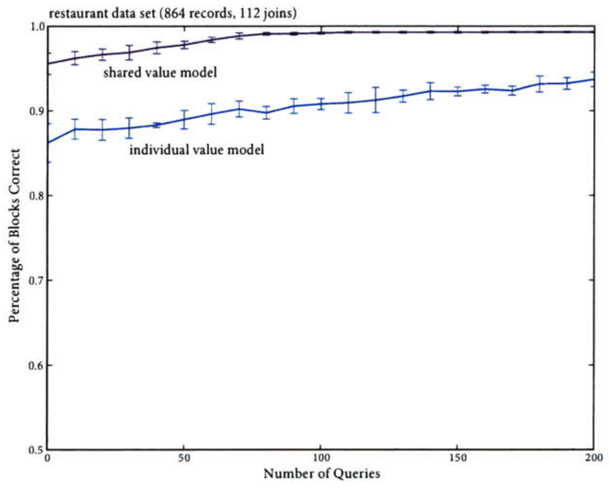


Figure 6-13: comparison between individual and shared value variants on *restaurant* using MI query ranking (5 runs averaged)

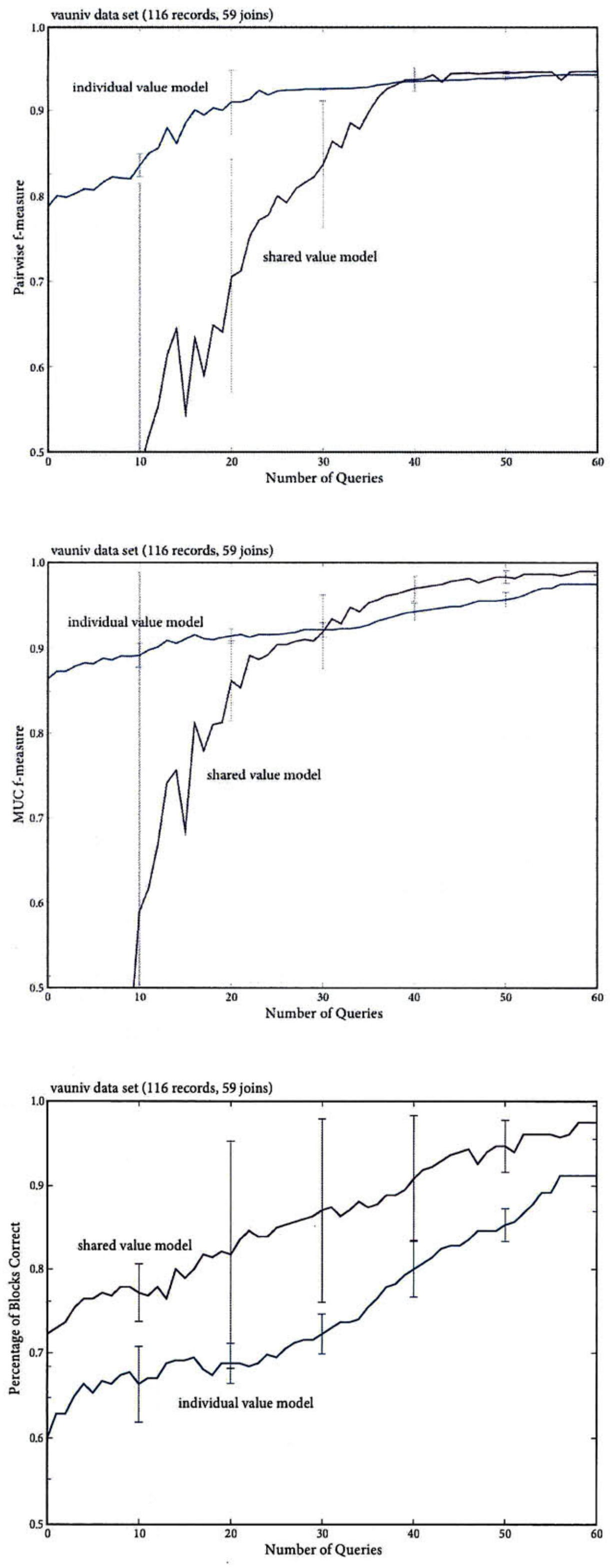


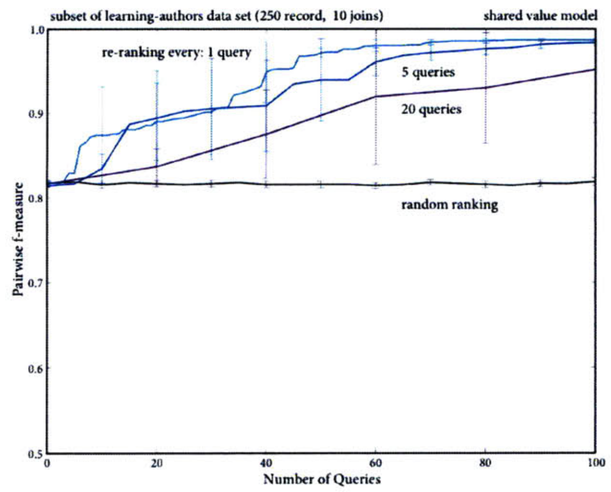
Figure 6-14: comparison between individual and shared value variants on *vauniv* using MI query ranking (5 runs averaged)

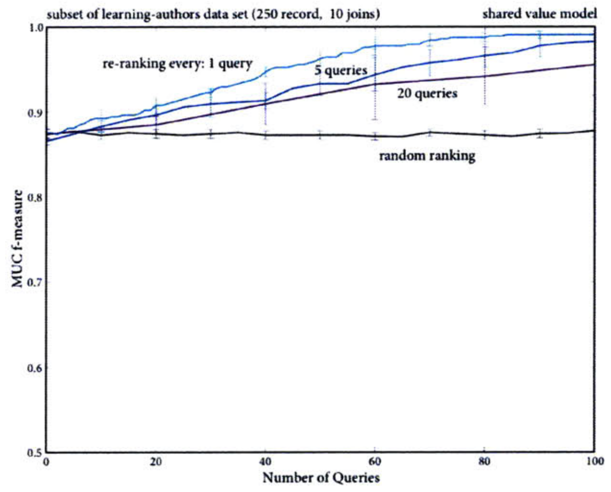
6.4.4 Effect of Query Reranking Frequency

A final question of interest is the impact of reranking queries through the additional inference. It seems desirable for the system to include all available information when deciding what query to ask, however the inference stage is the most computationally intensive in the system; Just as in the experiments of the preceding sections, in practice it may be necessary for the system to request many labels in a batch. Also, it may be the case that simply eliminating queries whose label is already determined, as described in section 6.2, is sufficient to achieve good performance with minimal inference.

Figures 6-15 through 6-19 show the results of a series of experiments examining this issue by comparing runs of 100 rounds of the MMI algorithm with query reranking every 1, 5, and 20 queries. The **random** algorithm, which ignores the results of inference, is also provided as a baseline. The IVD variant was used for data sets with a single field, and the SVD variant was used otherwise. A randomly chosen subset of *cora* and *learning-authors* was used, since running 100 rounds of inference on the full data sets would have been prohibitive.

These results clearly demonstrate that frequent query reranking improves performance above and beyond transitivity constraints. In some cases, such as in *vauniv* a given level of performance could be obtained in half the number of labels if queries were reranked after every query instead of after every 20. More frequent reranking may also have significantly improved the relative performance of the MMI ranking on *cora* and *learning-authors* in the experiments of section 6.4.6.





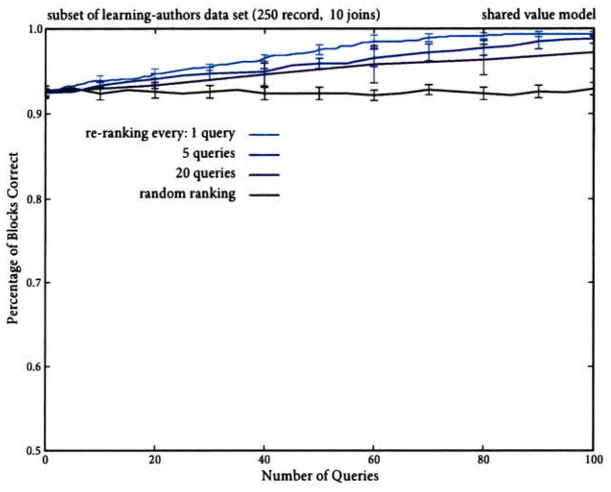
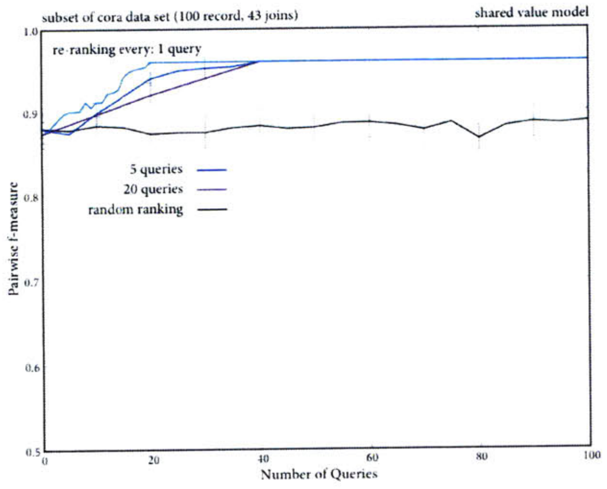
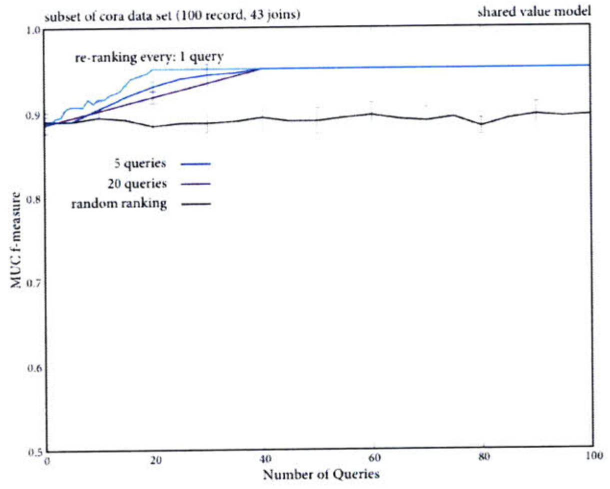


Figure 6-15: effect of inference frequency on *learning-authors* using MI query ranking (5 runs averaged)





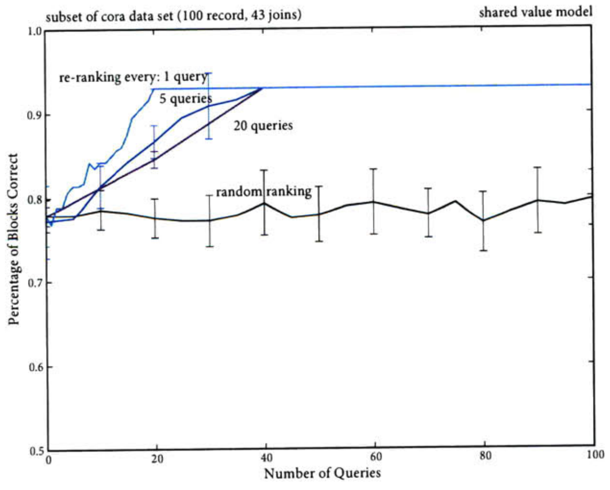
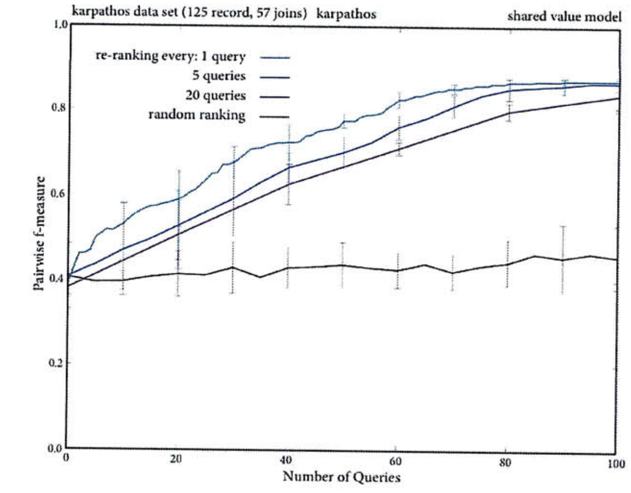
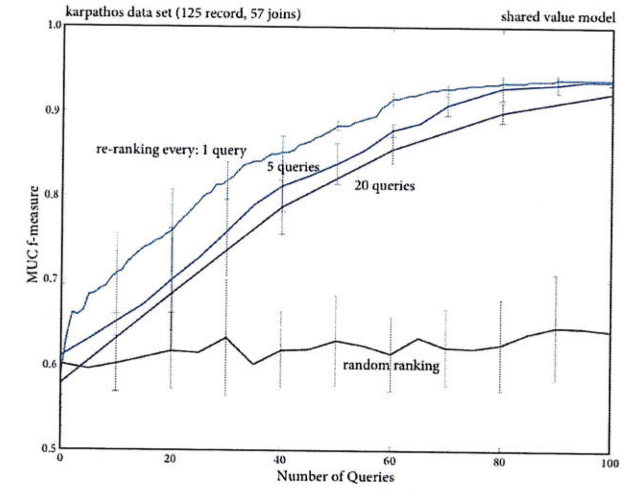


Figure 6-16: effect of inference frequency on *cora* using MI query ranking (5 runs averaged)





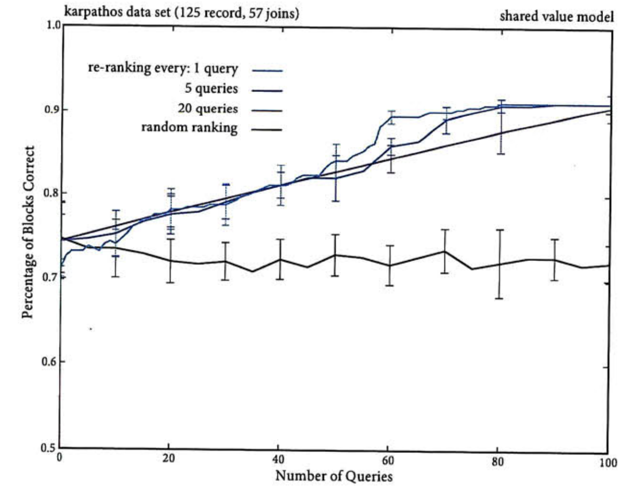
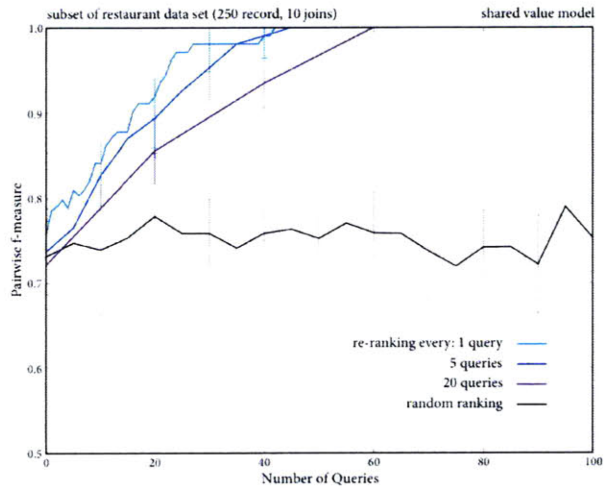


Figure 6-17: effect of inference frequency on *karpathos* using MI query ranking (5 runs averaged)



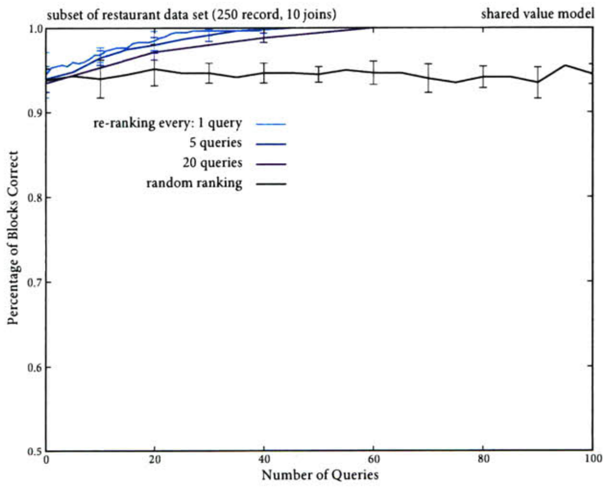
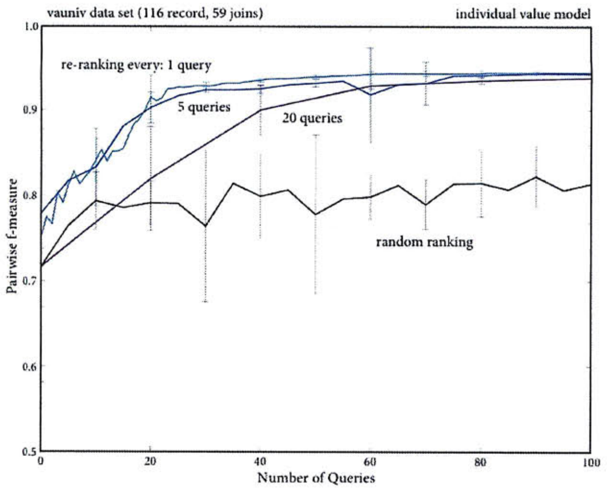
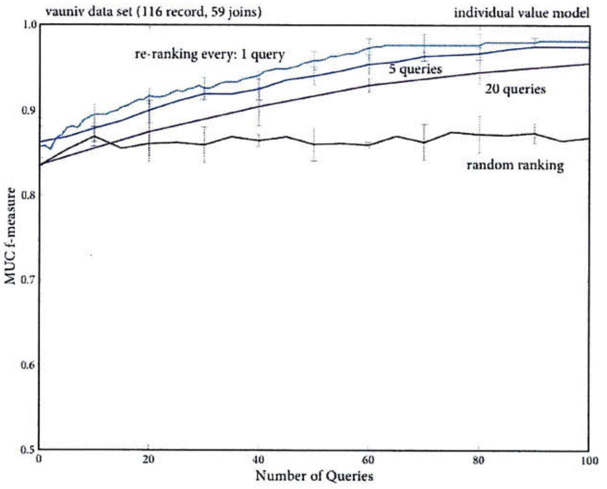


Figure 6-18: effect of inference frequency on *restaurant* using MI query ranking (5 runs averaged)





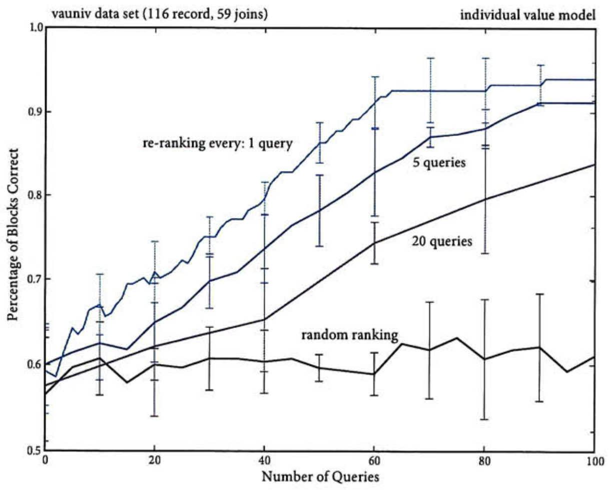


Figure 6-19: effect of inference frequency on *vauniv* using MI query ranking (5 runs averaged)

6.4.5 Examples of MMI Queries

Some example queries chosen by the MMI algorithm are provided in this section along with their responses. The first batch comes from a single run on *vauniv* from the experiments described in section 6.4.4, with reranking every query. One of the aspects that makes this data set challenging for duplicate detection is that both true matches and nonmatches often differ by only a single word, for example the name of a city. Also, many institutions share non-distinguishing words such as "university". Though incorporating such domain-specific knowledge into the model could undoubtably improve performance, the system is still choosing interesting examples which might even be challenging to a human supervisor who was unfamiliar with the institutions in question.

Top ten MMI queries on *vauniv* using IVD variant

```
Question 1: True
· United States Army, Engineer Topographic Laboratories, Fort Belvoir
· United States Army, Directorate of Evaluation and Standardization,
Fort Belvoir


Question 2: False
· Science Applications International Corporation, McLean
· Science Applications International Corporation, Hampton


Question 3: False
· Science Applications Laboratory for Atmospheric and Space Sciences,
McLean
· Institute for Computer Applications in Science and Engineering,
Hampton


Question 4: False
· Lockheed Engineering and Science, Hampton
· Institute for Computer Applications in Science and Engineering,
Hampton


Question 5: True
· Hampden-Sydney College
· Hampden-Sydney College, Hampden-Sydney


Question 6: True
· Old Dominion University, Norfolk
· Mechanical Engineering & Mechanics Department, Old Dominion
University, Norfolk
```

```
Question 7: False
· Analytical Mechanics Associates, Inc., Hampton
· W. J. Schafer Associates, Inc., Hampton, Virginia
```

```
Question 8: False
· Systems and Applied Sciences Corporation, Hampton
· ST Systems Corporation, Hampton
```

```
Question 9: False
· Hollins College, Roanoke
· Roanoke College, Salem
```

```
Question 10: True
· West Virginia University, Morgantown
· West Virginia University, Morgantown, W.
```

The next batch of example queries comes from using the MMI algorithm on the subset of 250 records from *restaurant* used in section 6.4.4. The first five are from the IVD variant and the second five from the SVD variant. Though the SVD variant performed significantly better on this data set, the questions do not appear qualitatively different. These example queries make clear how *restaurant* violates the conditional independence assumptions of field value generation. In particular, many restaurants in the data set are on the same street, and thus share a city and also an area code. In the example queries, the algorithm appears to be attempting to resolve such superficial similarities.

Top five MMI queries on *restaurant* subset using IVD variant

```
Question 1: False
· georgia grille; 2290 peachtree rd. peachtree square shopping center;
atlanta; american; 404/352-3517
· original pancake house (at); 4330 peachtree rd.; atlanta; american;
404-237-4116
```

```
Question 2: False
· binion's coffee shop; 128 fremont st.; las vegas; coffee shops/diners;
702/382-1600
· stefano's; 129 fremont st.; las vegas; italian; 702-385-7111
```

```
Question 3: False
· art's delicatessen; 12224 ventura blvd.; studio city; american;
818/762-1221
· pinot bistro; 12969 ventura blvd.; studio city; french bistro;
818-990-0500
```

```
Question 4: False
· nate 'n' al's; 414 n. beverly dr.; los angeles; american; 310/274-0101
· ruth's chris steak house (los angeles); 224 s. beverly dr.; beverly
hills; steakhouses; 310-859-8744
```

```
Question 5: False
· camille's; 1186 n. highland ave.; atlanta; italian; 404/872-7203
· eats; 600 ponce de leon ave.; atlanta; italian; 404-888-9149
```

Top five MMI queries on *restaurant* subset using SVD variant

```
Question 1: False
· art's delicatessen; 12224 ventura blvd.; studio city; american;
818/762-1221
· sushi nozawa; 11288 ventura blvd.; studio city; japanese; 818-508-7017
```

```
Question 2: False
· citrus; 6703 melrose ave.; los angeles; californian; 213-857-0034
· tommy tang's; 7313 melrose ave.; los angeles; asian; 213/937-5733
```

```
Question 3: False
· art's delicatessen; 12224 ventura blvd.; studio city; american;
818/762-1221
· pinot bistro; 12969 ventura blvd.; studio city; french bistro;
818-990-0500
```

```
Question 4: False
· vertigo; 600 montgomery st.; san francisco; mediterranean;
415/433-7250
· la mediterranee; 288 noe st.; san francisco; mediterranean;
415-431-7210
```

```
Question 5: False
· hayes street grill; 320 hayes st.; san francisco; seafood;
415/863-5545
· vicolo pizzeria; 201 ivy st.; san francisco; pizza; 415-863-2382
```

sec

6.4.6 Comparison to Related Work

The *restaurant* and *cora* data sets are also used by Bilenko and Mooney [18] for duplicate detection using a edit distance measure of field values learned from training examples. They present results from four algorithms based on string edit distance models, including two that are learned from training examples. Though largely domain independent, domain-specific stemming and stopword removal were used as preprocessing steps.

Figures 6-20 and 6-21 provide pairwise precision-recall curves for all four algorithms on the *cora* and *restaurant* data sets. These figures were generated as averages of 20 runs of 2-fold cross validation, training on half the data set and testing on the other half. Figure 6-22 reports unsupervised point results for this thesis.

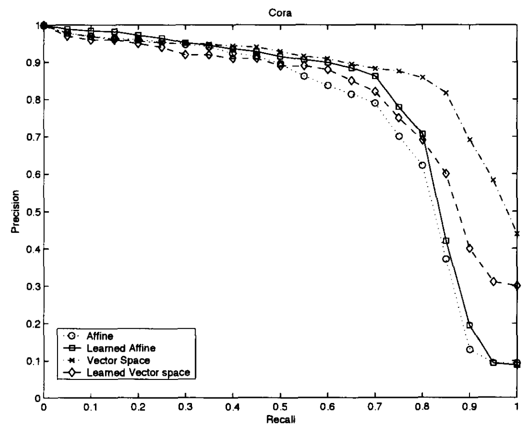


Figure 6-20: Precision-recall curve for *cora* from Bilenko & Mooney

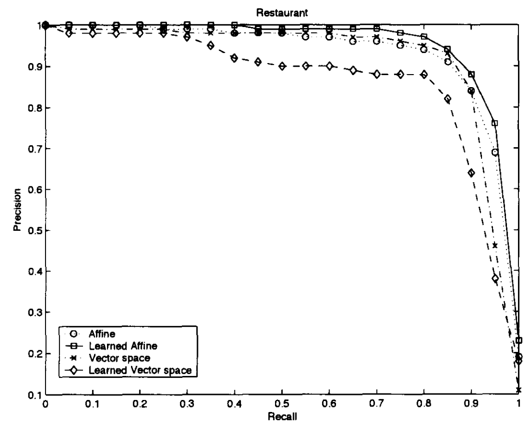


Figure 6-21: Precision-recall curve for *restaurant* from Bilenko & Mooney

Bilenko and Mooney also report the maximum f-measure for their algorithms, ranging from 0.826–0.922 for *restaurant* and from 0.793–0.867 for *cora*. Picking a threshold for a discriminative classifier that achieves

		Pairwise f-measure	Pairwise precision	Pairwise recall
restaurant	IVD variant	0.781	0.684	0.911
	SVD variant	0.919	0.882	0.961
cora	IVD variant	0.438	0.985	0.282
	SVD variant	0.426	0.986	0.273

Figure 6-22: Unsupervised pairwise performance for IVD and SVD variants

the maximum f-measure is nontrivial. One commonly used technique is to hold out a portion of the testing data as a validation set and chose the threshold by maximizing the performance on that set. However, if the training or validation sets have match/nonmatch statistics that differ significantly from the testing set, then precision or recall will suffer in testing.

The unsupervised pairwise f-measure of the SVD variant on *restaurant* is within the range of maximum f-measures reported by Bilenko and Mooney for their system. However their results did not include the telephone number field, which is considered to make the problem easier. Without further experimentation, it cannot be said whether this field made much of a difference; the unigram model of field generation coupled with the fact that many telephone numbers share area codes and exchanges could mean that the field had little impact on the partition found by the model.

The unsupervised pairwise f-measure of the system on *cora* was considerably lower. This is due to the problem of poor recall of the nonparametric model on data sets with large blocks, described in section . It is interesting to note, however, that the precision-recall tradeoff chosen by the model does lie approximately on the curves in figure 6-20.

Chapter 7

Conclusions and Future Directions

A man with one watch knows what time it is. A man with two is never sure.

anonymous

This thesis has described an active system for duplicate detection with three technical contributions: a domain-independent Bayesian model of coreference, a criterion for picking informative queries based on maximizing mutual information, and an algorithm for estimating a minimum-error partition under a Bayesian model based on minimizing expected pairwise error. Experimental results have been provided, both demonstrating the effectiveness of this method in a variety of domains, but also revealing areas for improvement in future research.

7.1 Improved Value Modeling

The model for duplicate detection presenting in this thesis has a number of theoretical appeals: It is domain-independent, and so does not require additional modeling effort to be applied to a new domain; it can be used in an unsupervised or actively supervised setting; it allowed for arbitrary pairwise constraints to be incorporated, including blocking constraints;

and it models the statistics of preferential attachment commonly seen in coreference problems.

However, there are also many areas for improvements. One straightforward improvement would be to use a more sophisticated model of string generation for the value generating distributions. While unigram models are important to document retrieval, they are generally better suited to use in modeling long passages of text, such as news article, than to modeling the shorter strings generally found in database records.

Instead of a unigram model, a bigram model could be used, which would allow the system the capacity to distinguish between the order of words as well as their presence. It is important, though, that any model used for modeling field values maintain sufficient ambiguity in order to be useful for identifying queries. A bigram model was initially attempted in developing this work and, though its unsupervised performance was generally better than the unigram model, it was too certain of its inference and wouldn't improve performance with additional queries.

Another area for improvement is in the model for sharing support between value-generating distributions. The SVD model captures the intuition that distinct entities may share properties and therefore values, however it doesn't capture the fact that some words and phrases *within* a field are less distinguishing than others. For example, the phrase "The proceeding of" is shared by a many conferences while the phrase "neural information processing systems" is fairly distinguishing.

One option for this type of modeling is to use a hierarchical structure for the base distribution of the value generating distributions, similar to the hierarchical Pitman-Yor language model of Teh [85]. This might allow the model to find words that are common to many fields and discount them as evidence toward coreference.

There are also a number of deeper issues in value modeling which were outside the scope of this thesis but are very important to the problem of duplicate detection. One issue is modeling field values which have a temporal component, such that the field values that are common for a particular entity vary across time.

Another modeling challenge occurs when values from one field may be expressed in another. For example, in early digital music databases there often was not separate field for the composer of a classical work, and some users chose to put the composer name in the `title` field, while others used the `artist` field. This same problem is also prevalent in database records created by machine extraction of structured data from free-form text.

Finally, even when using an effective duplicate detection method, there remains the problem of how to merge the duplicates into a "clean" database. Specifically, when multiple different values for a field are present in a coreference block, which should be chosen for the clean data set? In a generative model, the latent structures can often be used to generate a clean version of a particular field, for example the maximum likelihood value for the latent variable. The unigram model for value-generating distributions used in this thesis is too simple for such an application, but it's possible that a different choice of value-generating distribution could be used effectively.

7.2 Relational Value Modeling

Another area for future work is in domain-independent Bayesian models for relational data. A number of relational Bayesian models have been described for duplicate detection in the domain publications [67, 62, 22]. These models provide for a domain with a fixed schema, where all the entity types and permissible relations are chosen *a priori* by the data modeler.

The Bayesian model presented in this thesis is a distribution over distributions of coreferent records. Thus there is a natural extension of this model to allow for relational fields: use the model itself as a value-generating distribution. That is, when generating a relational field value r_2 for a record r_1, draw r_2 from the IVD or SVD variant. This would capture the fact that relational values can refer to records or entities that have been observed before or to entirely new entities.

While defining such a model may be straightforward, there is a challenge in using sampling for inference. In the model presented in this thesis, there was only a single record table, so it was possible to perform Gibbs sampling by "removing" an arbitrary record from the table and resampling its block membership. In a relational model there are multiple tables, connected by relations. If the relations are cyclic, such that a record is related to itself through other relational fields, then the distribution over its block membership may be defined recursively. Dealing with such distributions, even if they could be guaranteed to be well-defined, adds considerable complexity to inference.

7.3 Termination Criteria

A final aspect to active approaches not discussed previously is the problem of choosing a *termination criterion*, a method for determining when further queries are expected to be of limited utility. As is evident in the results reported in chapter 6, performance tends to level off after a certain number of queries.

This is to be expected, since eventually the portions of the data deemed most ambiguous by the model have been resolved. At this point there may still be errors in the estimated partition, however the model will not represent these errors as ambiguous, and so many queries may have to be asked before the erroneous portions are chanced upon.

The MEPE and MMI criteria used in this thesis are a natural starting point for investigating termination conditions. For instance, a threshold for expected entropy reduction could be set so that querying would stop at the point where the expected increase in information from the next query was less than the threshold. The expected error of the partition as a whole could also be used for stopping.

In practice, however, additional cost structures may be associated with querying. For example, a typical user's willingness to answer queries might be expected to decrease as the number of queries increases, especially since later queries are unlikely to improve performance considerably. Analyzing the problem of termination when additional queries have a nonlinear cost function associated with them might lead to insights towards better ranking functions for active duplicate detection.

Bibliography

[1] Secondstring project.

[2] *SIGMOD Record*, volume 20. ACM, December 1991.

[3] *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, August 2002.

[4] *Advances in Neural Information Processing Systems*, number 18, Vancouver, British Columbia, Canada, December 2005.

[5] Exploiting relationships for domain-independent data cleaning. 2005.

[6] D.J. Aldous. Exchangeability and related topics. In P.L. Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XII*, volume 1117 of *Lect. Notes Math.* Springer, 1985.

[7] Rohit Ananthakrishna, Surajit Chaudhuri, and Venkatesh Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 586–597, Hong Kong, aug 2002. Morgan Kaufmann.

[8] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistical Computing*, 18:343–373, 2008.

[9] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, 1998.

[10] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *The 43rd Annual Symposium on Foundations of Computer Science*, pages 238–247, 2002.

[11] Sugato Basu, Anindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pages 333–344, Lake Buena Vista, FL, April 2004.

[12] Omar Benjelloun, Hector Garcia-Molina, Hideki Kawai, Tait Eliott Larson, David Menestrina, Qi Su, Sutthipong Thavisomboon, and Jennifer Widom. Generic entity resolution in the serf project. *IEEE Data Eng. Bull.*, 29(2):13–20, 2006.

[13] Indrajit Bhattacharya and Lise Getoor. Iterative record linkage for cleaning and integration. In Gerhard Weikum, Arnd Christian König, and Stefan Deßloch, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Paris, France, June 2004. ACM.

[14] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava, editors, *Proceedings of the Sixth SIAM International Conference on Data Mining*, Bethesda, MD, April 2006.

[15] Mikhail Bilenko. RIDDLE: Repository of information on duplicate detection, record linkage, and identity uncertainty.

[16] Mikhail Bilenko. *Learnable Similarity Functions and Their Application to Record Linkage and Clustering*. PhD thesis, Department of Computer Sciences, University of Texas at Austin, August 2006.

[17] Mikhail Bilenko, Sugato Basu, and Mehran Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *In Proceedings of ICDM*, pages 58–65. IEEE, 2005.

[18] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, August 2003.

[19] Mikhail Bilenko and Raymond J. Mooney. Employing trainable string similarity metrics for information integration. In *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*, pages 67–72, Acapulco, Mexico, August 2003.

[20] Mikhail Bilenko and Raymond J. Mooney. On evaluation and training-set construction for duplicate detection. In *Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, pages 7–12, 2003.

[21] D. Blackwell and J. B. MacQueen. Ferguson distributions via pólya urn schemes. *Annals of Statistics*, 1:353–355, 1973.

[22] Peter Carbonetto, Jacek Kisynski, Nando de Freitas, and David Poole. Nonparametric bayesian logic. In *UAI*, pages 85–93, Edinburgh, Scotland, July 2005. AUAI Press.

[23] Peter Christen. A comparison of personal name matching: Techniques and practical issues. In *proceedings of the Workshop on Mining Complex Data*, Hong Kong, December 2006.

[24] William W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. In Laura M. Haas and Ashutosh Tiwary, editors, *SIGMOD Conference*, pages 201–212. ACM Press, 1998.

[25] William W. Cohen, Henry Kautz, and David McAllester. Hardening soft information sources. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 255–259, Boston, MA, August 2000.

[26] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, pages 73–78, 2003.

[27] William W. Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [3].

[28] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *J. Artif. Intell. Res. (JAIR)*, 4:129–145, 1996.

[29] The World Wide Web Consortium. Resource Description Framework. `http://www.w3.org/RDF/`, November 2004.

[30] Aron Culotta, Pallika Kanani, Robert Hall, Michael Wick, and Andrew McCallum. Author disambiguation using error-driven machine learning with a ranking loss function. In *Sixth International*

Workshop on Information Integration on the Web (IIWeb-07), Vancouver, Canada, 2007.

[31] Aron Culotta and Andrew McCallum. Joint deduplication of multiple record types in relational data. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, pages 257–258, Bremen, Germany, oct 2005. ACM.

[32] Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, page 150, 1995.

[33] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, September 2006.

[34] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In Fatma Özcan, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland, June 2005. ACM.

[35] Warren Ewens. The sampling theory of selectively neutral alleles. *Theoretical Population Biology*, 3:87–112, 1972.

[36] U. Fayyad, G. Piatetski Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3), 1996.

[37] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[38] T.S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, pages 209–230, 1973.

[39] James C. French, Allison L. Powell, Eric Schulman, and John L. Pfaltz. Automating the construction of authority files in digital libraries: A case study. In *European Conference on Digital Libraries*, pages 55–71, 1997.

[40] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Information, prediction, and query by committee. In S. Hanson et al., editor, *Advances in Neural Information Processing Systems*, volume 5. Morgan Kaufmann, 1993.

[41] Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, August 2003. ACM.

[42] Zoubin Ghahramani. Non-parametric bayesian methods. Tutorial Presentation at the UAI Conference.

[43] Sharon Goldwater, Tom Griffiths, and Mark Johnson. Interpolating between types and tokens by estimating power-law generators. In *NIPS* [4].

[44] R. Guha. Object co-identification on the semantic web. New York, NY, May 2004. ACM Press.

[45] Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007. The Association for Computer Linguistics.

[46] Robert Hall, Charles Sutton, and Andrew McCallum. Unsupervised deduplication using cross-field dependencies. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, August 2008. ACM.

[47] Mauricio A. Hernandez and Salvatore J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.

[48] Hal Daumé III and Daniel Marcu. A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, 6:1577, 2005.

[49] Nicole Immorlica and Anthony Wirth. *Correlation Clustering*, chapter 13. Chapman & Hall/CRC, 2008.

[50] Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. MIT Press, 2007.

[51] Michael I. Jordan. Dirichlet processes, chinese restaurant processes, and all that. Tutorial presentation at the NIPS Conference.

[52] Won Kim. *Introduction to Object-Oriented Databases*. MIT Press, 1990.

[53] Judice L.Y. Koh, Mong Li Lee, Asif M. Khan, Paul T.J. Tan, and Vladimir Brusic. Duplicate detection in biological data using association rule mining. In *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, pages 31–37, 2004.

[54] Steve Lawrence, Kurt Bollacker, and C. Lee Giles. Autonomous citation matching. In Oren Etzioni, editor, *Proceedings of the Third International Conference on Autonomous Agents*, New York, 1999. ACM Press.

[55] Mark Levene. On the information content of semi-structured databases. *Acta Cybern.*, 13(3):257–276, 1998.

[56] Xin Li, Paul Morie, and Dan Roth. Robust reading: Identification and tracing of ambigious names. 2004.

[57] David MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[58] Andrew McCallum and Ben Wellner. Object consolidation by graph partitioning with a conditionally-trained distance metric. In Getoor et al. [41].

[59] Andrew McCallum and Ben Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Advances in Neural Information Processing Systems* [4].

[60] Andrew K. McCallum, Kamal Nigam, and Lyle Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 169–178, Boston, MA, August 2000.

[61] Martin Michalowski, Snehal Thakkar, and Craig A. Knoblock. Exploiting secondary sources for automatic object consolidation. In Getoor et al. [41].

[62] Brian Milch, Bhaskara Marthi, and Stuart Russell. Blog: Relational modeling with unknown objects. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections*, pages 67–73, 2004.

[63] Alvaro E. Monge and Charles P. Elkan. The field matching problem: Algorithms and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[64] Daniel J. Navarro, Thomas L. Griffiths, Mark Seyvers, and Michael D. Lee. Modeling individual differences using dirichlet processes. *Journal of Mathematical Psychology*, 50:101–122, 2005.

[65] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage for vital records. *Science*, 130:954–959, 1959.

[66] Parag and Pedro Domingos. Multi-relational record linkage. In *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining*, pages 31–48, 2004.

[67] Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart J. Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1401–1408, Vancouver, British Columbia, 2003. MIT Press.

[68] Anand Patil, David Huard, and Christopher Fonnesbeck. *PyMC User Guide*. March 2009.

[69] Avi Pfeffer. *Probabilistic Reasoning for Complex Systems*. PhD thesis, Stanford, 2000.

[70] J. Pitman. Some developments of the Blackwell-MacQueen urn scheme. In T. S. Ferguson et al., editor, *Statistics, Probability and Game Theory: Papers in Honor of David Blackwell*, pages 245–267, Hayward, CA, 1996. Institute of Mathematical Studies.

[71] Jim Pitman. The two-parameter generalization of ewens' random partition structure. Technical Report 345, Department of Statistics, University of California, Berkeley, Berkeley, CA, March 1992.

[72] Jim Pitman. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, University of California, Berkeley, CA, August 2002.

[73] Jim Pitman and Marc Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, April 1997.

[74] Andrei Popescu-Belis and Isabelle Robba. Three new methods for evaluating reference resolution. In *Proceedings of LREC'98 Workshop on Linguistic Coreference*, pages 43–48, Granada, Spain, 1998.

[75] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and currrent approaches. *IEEE Bulletin on Data Engineering*, 23(4), 2000.

[76] Pradeep Ravikumar and William W. Cohen. A hierarchical graphical model for record linkage. In *In UAI*, 2004.

[77] Gian-Carlo Rota. The number of partitions of a set. *American Mathematical Monthly*, 71(5):498–504, 1964.

[78] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sample estimation of error reduction.

[79] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [3].

[80] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[81] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Computational Learning Theory*, pages 287–294, 1992.

[82] Stephen Skuce. Rare books cataloging librarian, MIT libraries. personal interview.

[83] Erik B. Sudderth and Michael Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*. MIT Press, 2008.

[84] Yee Whye Teh. A bayesian interpretation of interpolated kneserney. Technical Report TRA2/06, NUS School of Computing, Singapore, 2006.

[85] Yee Whye Teh. A hierarchical bayesian language model based on pitman-yor processes. In *ACL*, Sydney, Australia, July 2006. The Association for Computer Linguistics.

[86] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[87] Sheila Tejada, Craig A. Knoblock, and Steven Minton. Learning object identification rules for information integration. *Information Systems*, 26:2001, 2001.

[88] Sheila Tejada, Craig A. Knoblock, and Steven Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–359, Edmonton, Alberta, Canada, 2002. ACM.

[89] Marc Vilain and et. al. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45–52, November 1995.

[90] W3C. Extensible markup language.

[91] Y. R. Wang and S. E. Madnick. The inter-database instance identification problem in integrating autonomous systems. In *Proceedings of the Sixth International Conference on Data Engineering*, February 1989.

[92] W. Winkler. Matching and record linkage, 1995.

[93] William E. Winkler. Using the em algorithm for weight computation in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 667–671, 1988.

[94] William E. Winkler. Overview of record linkage and current research directions. Technical Report Statistics #2006-2, U.S. Census Bureau, 2006.