Tractable Learning and Inference with High-Order Representations

Aron Culotta CULOTTA@CS.UMASS.EDU
Andrew McCallum MCCALLUM@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst, MA 01003

Abstract

Representing high-order interactions in data often results in large models with an intractable number of hidden variables. In these models, inference and learning must operate without instantiating the entire set of variables. This paper presents a Metropolis-Hastings sampling approach to address this issue, and proposes new methods to discriminatively estimate the proposal and target distribution of the sampler using a ranking function over configurations. We demonstrate our approach on the task of paper and author deduplication, showing that our method enables complex, advantageous representations of the data while maintaining tractable learning and inference procedures.

1. Introduction

Probabilistic models are commonly factored into a set of local decisions to make exact inference tractable. Often, this factorization sacrifices representational power that could be used to model important non-local phenomena in the data.

For example, language processing models are often restricted to interactions between consecutive words, vision models to adjacent pixels, and bioinformatics models to adjacent molecules. However, these local models ignore global characteristics of the data, such as properties of discourse, cohesion of an entire scene, or gene co-occurrence regularities.

The difficulty in representing these *global* phenomena is that it often results in an exponential increase in the size of the solution search space. In many cases, this is realized by a model in which the number of hidden variables is exponential in the size of the input (for example, a clustering model with a variable for every subset of the input). In these situations, the set of hidden variables cannot even be fully instantiated, much less iterated over for inference.

Recently, a number of relational probabilistic models have been proposed which provide the practitioner great flexibility in specifying the representation and connectivity of the variables. One example, Markov logic networks (MLNs), accepts as input a set of first-order logic formulae, from which a Markov network is instantiated for a given set of observed input constants (Richardson & Domingos, 2006). This instantiation, or *grounding*, creates a random variable for all possible instantiations of each predicate. Thus, if the highest predicate arity is c, and the number of constants is n, then the number of instantiated variables is $O(n^c)$. It is intractable to instantiate such a large set of variables for real-world problems.

However, these high-arity predicates are precisely the sort of predicates that are useful for modeling global characteristics of the data. For example, consider a predicate HaveSameAdvisor($\{a_i \dots a_{i+k}\}$), indicating whether a subset of students $\{a_i \dots a_{i+k}\}$ have the same advisor. This predicate must be instantiated for every subset of student constants, i.e. the power set $\mathcal{P}(\mathcal{A})$. Clearly, this is infeasible for a large set of students. However, constructing these types of *query* predicates enables the construction of powerful *evidence* predicates such as CoauthoredAtLeastTwoPapers($\{a_i \dots a_{i+k}\}$) (indicating whether there are at least two papers that some combination of authors from $\{a_i \dots a_{i+k}\}$ have co-authored) and a *generalized predicate* NumberOfStudents(a_i) (indicating the number of students a researcher advises simultaneously).

It should be noted that this problem is *not* specific to MLNs. Indeed, the problem is inherent in any domain

Presented at the ICML Workshop on Open Problems in Statistical Relational Learning, Pittsburgh, PA, 2006.

in which we would like to model aggregate characteristics of predicted variables. However, the flexible representation language of MLNs brings these issues to the fore.

Because the set of random variables cannot be enumerated, approximations will undoubtedly be required for inference and parameter estimation; however, there are very few existing techniques that are specifically designed to operate without a full grounding of the variable set.

One way to avoid grounding these large networks is to perform *lifted inference* (Poole, 2003; de Salvo Braz et al., 2005). Lifted inference enables predictions to be made about an entire *population* without instantiating a node for each member of the population. The key insight of this approach is to use a lifted version of variable elimination to perform inference on a large set of variables implicitly. However, in many domains it is necessary to make inferences about *specific* input nodes, rather than over the entire population. It is not clear how current versions of lifted inference can address this, although approximate methods may be possible.

Another approach is to perform Markov chain Monte Carlo (MCMC) sampling. MCMC algorithms perform MAP inference by stochastically searching the solution space, guided by an iteratively improved approximation of the joint (or conditional) probability distribution. Many sampling algorithms consider models in which the number of variables is tractable, but calculating the normalization constant is not. For example, recent approaches to memory-efficient MAP in MLNs still require enumerating all ground clauses as part of the initialization procedure (Singla & Domingos, 2006).

This paper presents a Metropolis-Hastings (M-H) sampler to perform MAP inference in models in which it is infeasible to instantiate all hidden variables. The M-H sampler relies on two distributions: a *proposal distribution*, from which we sample possible changes to the predicted configuration, and a *target distribution*, which scores the quality of the proposed configuration. Because this approach to inference only requires a method of comparing two configurations, we can avoid instantiating many variables that are irrelevant to this comparison.

While there has been some work on inference in these models, there has been noticeably less work on parameter estimation. Standard maximum likelihood approaches are infeasible here: Computing the data likelihood requires computing a normalization constant that sums over all possible settings of an exponential number of variables.

Most approximate learning techniques again assume the difficulty is in computing the normalizer, but in this paper we address the case where it is intractable not only to compute the normalizer, but also to enumerate each possible prediction in the training set.

Knowing that M-H sampling will be used for inference, it is natural to attempt to learn parameters that improve the efficiency and accuracy of that sampler. This paper presents a way to directly estimate the target and proposal distributions of the M-H sampler. Given examples of two candidate configurations, we propose discriminatively learning a ranking function that will assign higher scores to configurations that are closer to the true configuration. We also discuss ways of sharing parameters between the target and proposal distributions to improve learning efficiency.

We present results with these techniques on two real-world datasets for coreference resolution, in which the task is to cluster paper and author mentions into co-referent sets. Unlike traditional techniques which factor the problem into a set of pairwise variables indicating whether two mentions are co-referent, we represent interactions among *entire sets* of mentions, which enables *first-order* features over candidate clusters. We compare our approach against the traditional approach, and conclude that even with simple search procedures this broader representational power can improve accuracy.

2. Model

Let $X = \{x_1 \dots x_n\}$ be a set of observed variables, and let $Y = \{y_1 \dots y_m\}$ be a set of unobserved variables that we wish to predict. This paper is concerned with cases in which the Y variables cannot be explicitly enumerated, e.g. $m = O(2^n)$.

Although our learning and inference algorithms could be applied to a variety of model structures, this paper focuses on a single (general) class of clustering models: models where the Y variables indicate some compatibility among sets of X variables. We introduce two types of Y variables: those that indicate the compatibility among a cluster of X variables, and those that indicate the compatibility of a *pair* of clusters of X variables.

The model represents the conditional distribution $P(Y|X)$ as follows: Each possible clustering of the data can result in both a different set of instantiated Y variables and in a different assignment to those instan-

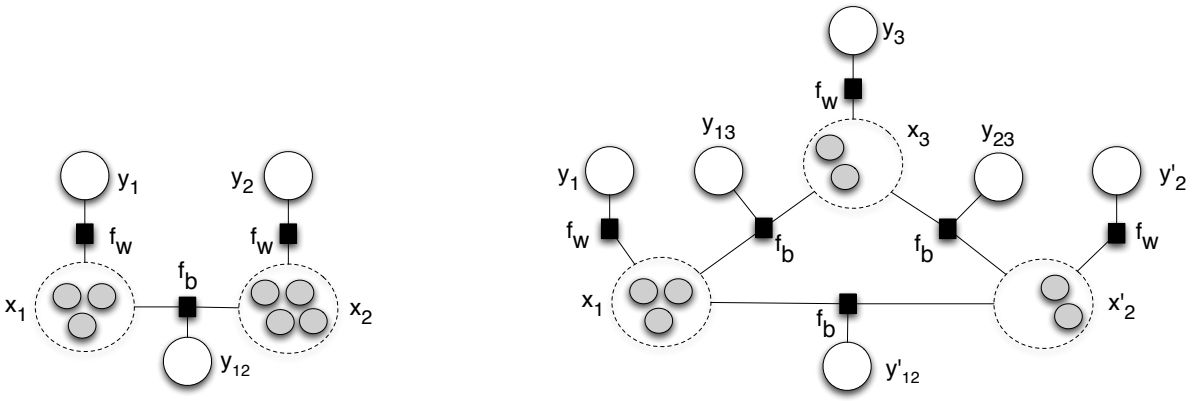

Figure 1. Two factor graphs for two different instantiation of the clustering model. The figure on the left depicts the model for a configuration predicting two clusters; the figure on the right for a model with three clusters. The model contains two types of factors: those computing the *within-cluster* compatibility (f_w), and those computing the *between-cluster* compatibility (f_b). Note that the model on the right introduces five new Y variables ($y_3, y'_2, y'_{12}, y_{23}, y_{13}$).

tiated variables. The value of $P(Y|X)$ is represented with a normalized log-linear model representing how likely configuration Y is. Thus, configuration Y is assigned a score, and the conditional density of Y is obtained by dividing this score by the sum of the scores of all possible configuration.

We parameterize this scoring function with potential functions that examine both within-cluster attributes and between-cluster attributes.

The functional form of this distribution is as follows:

$$P(Y|X) = \frac{1}{Z_X} \prod_{y_i \in Y} f_w(y_i, \mathbf{x}_i) \prod_{y_i, y_j \in Y} f_b(y_{ij}, \mathbf{x}_{ij}) \quad (1)$$

where Z_X is the input-dependent normalizer, factor f_w parameterizes the *within-cluster* compatibility, and factor f_b parameterizes the *between-cluster* compatibility. We assume a log-linear model of the potential functions, i.e $f(y_i, x_i) = \exp(\sum_k \lambda_k g_k(y_i, x_i))$.

Note that this differs from a more traditional factored representation that contains a Y variable for each pair of X variables, enforcing transitivity among the Y's to ensure a consistent clustering. Many common clustering algorithms (k-Means, greedy agglomerative clustering, etc) can be viewed as instances of these pairwise models. However, by only considering "pairwise" Y variables, these approaches have a limited representation, since they can only consider evidence about pairs of X variables, rather than sets of X variables. By considering sets of X variables, the feature functions in Equation 1 can compute what we term *first-order features*: features that can calculate existential and universal properties of a set of objects.

Figure 1 displays two factor graphs for two different instantiation of Y. The figure on the left displays the set of variables instantiated for a configuration predicting two clusters, and the figure on the right displays the variables instantiated for a configuration of three clusters. From these figures, we can see that the set of all possible Y variables is doubly-exponential: Not only is there a y_i for each subset of the input, but there is also a y_{ij} for each pair of subsets. We can see that this greater representation leads to a model that is infeasible to completely instantiate.

Since we cannot instantiate all of the Y variables, we must modify traditional inference and learning procedures to operate with partial instantiations. To do this, we introduce methods that consider only the *difference* between two instantiations. As we will describe below, this will allow us avoid instantiating many variables that are irrelevant to this comparison.

For example, using Equation 1, we can calculate unnormalized scores for both configurations given in Figure 1; however, these scores need not consider variables that are the same in both configurations.

In the following sections, we will describe ways to discriminatively learn the λ weights from Equation 1 so that better configurations are given higher scores. We can then employ this learned model to perform MAP inference with a sampling algorithm that stochastically searches the configuration space, proposing and accepting moves based on the relative scores of each configuration.

3. MAP Inference

Maximum a posteriori (MAP) inference seeks the solution to the following optimization problem:

$$Y^* = \operatorname*{argmax}_Y P(Y|X)$$

Exact MAP inference in models with small tree-width can typically be computed with a dynamic program (such as the max-product algorithm). When the model contains many loopy dependencies, variational or sampling approximations become necessary. However, these approximations generally assume all variables can be instantiated, which is not true in our case.

Instead, we employ Metropolis-Hastings (M-H) sampling. M-H is an MCMC sampling procedure that estimates a distribution from a sequence of samples of one or more variables from that distribution.

M-H can be viewed as a way to search the configuration space. Given an initial configuration Y_i, the probability of moving to a new configuration Y_i' is proportional to the acceptance probability

$$\alpha(Y_i'|Y_i) = \frac{P(Y_i|X)}{P(Y_i'|X)} \cdot \frac{Q(Y_i'|Y_i)}{Q(Y_i|Y_i')}$$

If $\alpha(Y_i'|Y_i) > 1$, then the new configuration Y_i' is chosen. Otherwise, the original configuration Y_i is retained with probability $\alpha(Y_i'|Y_i)$. A temperature parameter may also be used to reduce the amount of randomness in this decision, similar to simulated annealing approaches. This sampling procedure has been used in a generative model for similar problems in Pasula et al. (2003).

The Q distribution is called the *proposal distribution* because sampling from Q proposes changes to the current configuration. The likelihood ratio of the Q distributions weights the likelihood ratio of the *target distribution* P. Note that we do not have to compute normalization constants for this sampler since they cancel in the likelihood ratios.

In the clustering model, Q considers a set of possible changes to the current configuration. A typical implementation of Q would be to sample a pair of clusters, then sample a perturbation of these clusters that may result in a merging, splitting, or reshuffling of the clusters.

With ergodicity constraints on Q, it can be shown that the Markov chain induced by M-H will converge to the proper stationary distribution for P. Given P, Q, and an initial configuration, we can perform MAP inference by sampling from this Markov chain until convergence or until a fixed number of samples is met.

In the next section, we discuss how to estimate the parameters of P and Q to maximize a discriminative criterion appropriate for the sampler.

4. Parameter Estimation

Given a set of labeled data for which the true setting of Y is known, we wish to estimate the model parameters λ that will maximize performance on unseen data. A standard maximum likelihood approach chooses λ to maximize the log-likelihood of the labeled data, typically found by gradient ascent on the derivative of the log of Equation 1 with respect to λ. Unfortunately, this gradient involves computing marginals over Y variables, requiring the normalizer Z_X, which is infeasible to compute in our model. Furthermore, this method requires enumerating all the Y variables in the labeled data, which is also intractable.

One solution is to estimate the marginals by sampling from $P(Y|X)$ using similar sampling methods discussed in Section 3. However, since these marginals must be computed at each iteration of the learning procedure, this will require a prohibitive number of samples.

Instead, we choose a training criterion that is mindful of how MAP inference is performed. Since the M-H sampler only requires likelihood *ratios*, the full marginal distribution is not required to perform inference. Instead, we optimize λ to maximize the accuracy of the sampler; that is, to encourage the sampler to accept good configurations over poor ones.

We frame this as the following ranking task: Given two configurations, rank the "better" configuration higher than the other.

Let $Z_{p>q}$ be a binary random variable indicating if configuration Y_p receives a rank higher than configuration Y_q. We model the conditional probability of configuration $Z_{p>q}$ as the score given by the log-linear model of Equation 1 to Y_p normalized by the sum of the scores for Y_p and Y_q. That is, $P(Z_{p>q}|X) =$

$$\frac{\prod_{y_i \in Y_p} f_w(y_i, \mathbf{x}_i) \prod_{y_i, y_j \in Y_p} f_b(y_{ij}, \mathbf{x}_{ij})}{\sum_{k \in p,q} \prod_{y_i \in Y_k} f_w(y_i, \mathbf{x}_i) \prod_{y_i, y_j \in Y_k} f_b(y_{ij}, \mathbf{x}_{ij})} \quad (2)$$

This formulation avoids computing the normalizer Z_X, and also enables a training criterion well-suited to the inference procedure.

Given a set of labeled data for which the true optimal configuration is known, we can generate pairs of con-

figurations Y_p and Y_q, and can label the true value of $Z_{p>q}$. The true value of $Z_{p>q}$ can be calculated using any metric of interest. For example, in a clustering task, we can label $Z_{p>q} = 1$ if the pairwise F1 performance of clustering Y_p is greater than that for Y_q. Domain-specific metrics may be used without requiring additional modifications to the training procedure.

Given this set of labeled data, we can perform maximum likelihood learning using the gradient of Equation 2, which is possible now since the normalization over all possible configurations is no longer necessary.

4.1. Training Example Generation

The above approach to learning requires a set of configuration pairs, each labeled with the correct $Z_{p>q}$. For a given labeled data set, the number of possible training examples is quadratic in the size of the configuration space. Certainly, generating all such examples is infeasible. Therefore, we must decide which pairs to sample.

A simple approach is to sample uniformly at random. However, given the sparsity of many tasks, this may result in a large number of "easy" examples, for which there is a large difference in quality between configurations in a pair. Instead, we desire to focus our sampling on the difficult examples the sampler will likely consider at inference time.

With this in mind, we propose the following algorithm to sample M labeled training examples: While the number of labeled training examples is less than M, draw a sample from the Metropolis-Hastings inference engine and accept or reject it using current model parameters λ. Create a labeled example from the configuration pair consisting of the current configuration and the proposed configuration. Re-estimate λ from this augmented labeled data pool.

In this fashion, performing inference on the training set iteratively updates the model parameters, and as this sampling proceeds, the sampled labeled examples become increasingly similar to the sort of configurations likely to be encountered in unseen data.

4.2. Tying Parameters in P and Q

While we have described the form of P, we have not described the form of the proposal distribution Q. In general, this can be a nearly arbitrary distribution, with the restriction that it results in an *ergodic* Markov chain; that is, given enough samples, the initial configuration does not affect the space explored by the sampler.

However, if Q is uniform and the solution space is highly-peaked, then the sampler will require a large number of iterations before it reaches a high-probability configuration. Thus, Q is often a cheaper, stochastic approximation to P that attempts to propose moves that are accepted more often than not. By adding stochasticity to Q, we can ensure that enough exploration takes place to converge to the correct distribution.

Given this intuition, we propose reusing the λ parameters from P to parameterize Q. Viewing Q as an approximation to P, Q may only require a subset of λ. The specific form of Q and which λ's to reuse are largely domain-dependent decisions; but for the clustering model, a reasonable procedure for sampling Q is to select two clusters, then perform stochastic local search to propose a perturbation to the two clusters. This local search can use many of the same parameters as P for evaluating local decisions. The less stochasticity in this local search, the closer the proposal distribution resembles P. Thus, there is a tradeoff among Q's computational cost, acceptance rate, and stochasticity. Discovering the best trade-off is often an empirical question.

5. Experiments

We perform experiments on two identity uncertainty tasks: *citation matching* and *author disambiguation*. *Citation matching* is the task of determining whether two research paper citation strings refer to the same paper. We use the Citeseer corpus (Lawrence et al., 1999), containing approximately 1500 citations, 900 of which are unique. The citations are manually labeled with cluster identifiers, and the strings are segmented into fields such as author, title, etc. The citation data is split into four disjoint categories by topic, and the results presented are obtained by training on three categories and testing on the fourth.

With our proposed clustering model, we create a number of *first-order features* such as AllTitlesMatch, AllAuthorsMatch, AllJournalsMatch, etc., as well as their existential counterparts, ThereExistsTitleMatch, etc. We also include *count* features, which indicate the number of these matches in a set of mentions.

Additionally, we add edit distance features, which calculate approximate matches[1] between title fields, etc., for each pair of citations in a set of citations. First-order features are used for these as well, such as "there

[1]We use the Secondstring package, found at http://secondstring.sourceforge.net

	Objects	Pairs
constraint	**82.3**	76.7
reinforce	**93.4**	78.7
face	**88.9**	83.2
reason	81.0	**84.9**

Table 1. Pairwise F1 performance for the citation matching task, where OBJECTS is our proposed model that takes advantage of first-order features of the data, and PAIRS is a model restricted to only consider pairwise features. OBJECTS outperforms PAIRS on three of the four testing sets.

	Objects	Pairs
miller d	41.9	**61.7**
li w	**43.2**	36.2
smith b	**65.4**	25.4

Table 2. Pairwise F1 performance on the author disambiguation task. OBJECTS outperforms PAIRS on two of the three testing sets.

exists a pair of citations in this cluster which have titles that are less than 30% similar" and "the minimum edit distance between titles in a cluster is greater than 50%."

We present experiments using a low temperature version of the sampler. That is, with very high probability, the proposal distribution proposes the move with the highest score, and this move is accepted. Additionally, these experiments do not use the f_b factors, relying instead on the within cluster factors f_w. We show that even with this simplified version of the sampler, the higher representational power can result in better performance.

We evaluate using pairwise precision, recall, and F1, which measure the system's ability to predict whether each pair of constants refer to the same object or not. Table 1 shows the advantage of our proposed model (OBJECTS) over a model that only considers pairwise factors between mentions (PAIRS). Note that PAIRS is a strong baseline that performs collective inference of citation matching decisions, but is restricted to use compute features over pairs of citations. Thus, the performance difference is due to the ability to model first-order features of the data.

Author disambiguation is the task of deciding whether two strings refer to the same author. To increase the task complexity, we collect citations from the Web containing different authors with matching last names and first initials. Thus, simply performing a string match on the author's name would not be sufficient in many cases. We searched for three common last name / first initial combinations (MILLER, D; LI, W; SMITH, B). From this set, we collected 400 citations referring to 56 unique authors. For these experiments, we train on two subsets and test on the third.

We generate first-order features similar to those used for citation matching. Additionally, we include features indicating the overlap of tokens from the titles and indicating whether there exists a pair of authors in this cluster that have different middle names. This last feature exemplifies the sort of reasoning enabled by first-order features: For example, consider a pairwise feature that indicates whether two authors have the same middle name. Very often, middle name information is unavailable, so the name "Miller, A." may have high similarity to both "Miller, A. B." and "Miller, A. C.". However, it is unlikely that the same person has two different middle names, and our model learns a weight for this feature. Table 2 demonstrates the advantage of this method.

Overall, OBJECTS achieves $F1$ scores superior to PAIRS on 5 of the 7 datasets. These results indicate the potential advantages of using more complex representations of the data.

6. Related Work

Methods to avoid instantiating all the variables in a Markov network have received limited attention in the machine learning literature. In addition to the work discussed in the introduction, Singla and Domingos (2006) present one approach to address the space complexity of MLNs. They propose LazySAT, a variant of a popular weighted SAT solver that takes advantage of the sparsity of true predicates, common in relational domains. Our approach differs from LazySAT in three principal ways. First, the predicates we enable are often not sparse. Consider again the $\mathit{HaveSameAdvisor}(a_i \ldots a_{i+k})$ predicate. For an advisor with 10 students, there exists a ground atom for every subset of those 10 students that will be true.

Second, LazySAT requires iterating over all possible ground clauses as part of its initialization procedure. While these clauses are not necessarily stored in memory simultaneously, simply iterating these clauses is infeasible for our domain.

Third, we propose a discriminative training criterion that directly optimizes the search procedure used at inference time, rather than the pseudo-likelihood training advocated by Richardson and Domingos (2006).

There has been a growing interest in non-local modeling for natural language processing tasks. This has in-

cluded using Gibbs sampling in information extraction (Finkel et al., 2005), approximate inference in loopy sequence models (Bunescu & Mooney, 2004; Sutton & McCallum, 2004), integer linear programming to enforce global constraints in semantic role labeling (Roth & Yih, 2004), and search-based prediction to perform joint inference of multiple tasks (Daumé III & Marcu, 2005).

Additionally, many probabilistic models of object identification have been proposed in the past 40 years in databases and natural language processing. With the introduction of statistical relational learning, more sophisticated models of identity uncertainty have been developed that consider the dependencies between related consolidation decisions.

The most relevant identity uncertainty models are the relational models in Milch et al. (2005), McCallum and Wellner (2003), and Parag and Domingos (2004). McCallum and Wellner (2003) present experiments using a conditional random field that factorizes into a product of pairwise decisions about mention pairs (Model 3). These pairwise decisions are made collectively using relational inference; however, as pointed out in Milch et al. (2004), there are shortcomings to this model that stem from the fact that it does not capture features of *objects*, only of mention pairs. For example, first-order features such as "a researcher is unlikely to publish in more than 2 different fields" or "a person is unlikely to be referred to by three different names" cannot be captured by solely examining pairs of mentions. Additionally, decomposing an object into a set of mention pairs results in "double-counting" of attributes, which can skew reasoning about a single object (Milch et al., 2004). Similar problems apply to the model in Parag and Domingos (2004).

Milch et al. (2005) address these issues by constructing a generative probabilistic model over possible worlds called BLOG, where realizations of objects are typically sampled from a generative process. While BLOG provides useful semantics for reasoning about unknown objects, the transition to generatively trained models sacrifices some of the attractive properties of the discriminative model in McCallum and Wellner (2003) and Parag and Domingos (2004), such as the ability to easily incorporate many overlapping features of the observed mentions. In contrast, generative models are constrained either to assume the independence of these features or to explicitly model their interactions.

Identity uncertainty can also be seen as an instance of *supervised clustering*. Daumé III and Marcu (2004) and Carbonetto et al. (2005) present similar Bayesian supervised clustering algorithms that use a Dirichlet process to model the number of clusters. As a generative model, it has similar advantages and disadvantages as Milch et al. (2005), with the added capability of integrating out the uncertainty in the true number of objects.

This paper has presented a discriminatively trained, conditional model of identity uncertainty that incorporates the attractive properties of McCallum and Wellner (2003) and Milch et al. (2005), resulting in a discriminative model to reason about objects.

Finally, the ranking function estimation in Section 4 is similar to work on discriminative re-ranking for parsing developed by Collins (2000), who optimizes a loss function based on the number of ranking errors made on the training set.

7. Conclusions and Future Work

We have presented learning and inference procedures for models in which instantiating the entire set of random variables is impractical. By building a Metropolis-Hastings sampler over the configuration space, we can perform efficient MAP inference.

We have also proposed a novel method of estimating the parameters of a Metropolis-Hastings sampler, by inducing a ranking function over pairs of configurations. This formulation makes learning in the model tractable, and optimizes a criterion suited to the sampling inference procedure.

By combining these techniques, we can enable models that capture global phenomena of the data. We demonstrate our approach using first-order existential and universal features on two real-world identity uncertainty datasets.

Future work will extend our approach to perform experiments on a wider variety of tasks. The model we have described here can be applied to many tasks other than identity uncertainty. For example, if the potential functions measure how likely it is that a set of fields belong to the same record, then we can learn a database model that can be used to extract entire records from text, as in Wick et al. (2006). Additionally, by treating partial parse trees as clusters in our model, we can construct a parsing model that incorporates global features over candidate trees.

We also plan to empirically evaluate different loss functions for the ranking function, such as a regression model, or a model that considers the long-term effects of a local configuration change, as in recent work on search-based prediction (Daumé III et al., 2006).

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by U.S. Government contract #NBCH040171 through a subcontract with BBNT Solutions LLC, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s)' and do not necessarily reflect those of the sponsor.

References

Bunescu, R., & Mooney, R. J. (2004). Collective information extraction with relational markov networks. *ACL*.

Carbonetto, P., Kisynski, J., de Freitas, N., & Poole, D. (2005). Nonparametric bayesian logic. *UAI*.

Collins, M. (2000). Discriminative reranking for natural language parsing. *Proc. 17th International Conf. on Machine Learning* (pp. 175–182). Morgan Kaufmann, San Francisco, CA.

Daumé III, H., Langford, J., & Marcu, D. (2006). Search-based structured prediction. Technical Note.

Daumé III, H., & Marcu, D. (2004). Supervised clustering with the dirichlet process. *NIPS'04 Learning With Structured Outputs Workshop*. Whistler, Canada.

Daumé III, H., & Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. *International Conference on Machine Learning (ICML)*. Bonn, Germany.

de Salvo Braz, R., Amir, E., & Roth, D. (2005). Lifted first-order probabilistic inference. *IJCAI* (pp. 1319–1325).

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. *ACL* (pp. 363–370).

Lawrence, S., Giles, C. L., & Bollaker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer*, *32*, 67–71.

McCallum, A., & Wellner, B. (2003). Toward conditional models of identity uncertainty with application to proper noun coreference. *IJCAI Workshop on Information Integration on the Web*.

Milch, B., Marthi, B., & Russell, S. (2004). Blog: Relational modeling with unknown objects. *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields*.

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., & Kolobov, A. (2005). BLOG: Probabilistic models with unknown objects. *IJCAI*.

Parag, & Domingos, P. (2004). Multi-relational record linkage. *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining* (pp. 31–48). Seattle, WA.

Pasula, H., Marthi, B., Milch, B., Russell, S., & Shpitser, I. (2003). Identity uncertainty and citation matching. *Advances in Neural Information Processing Systems 15*. MIT Press.

Poole, D. (2003). First-order probabilistic inference. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (pp. 985–991). Acapulco, Mexico: Morgan Kaufman.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Roth, D., & Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. *The 8th Conference on Compuational Natural Language Learning*.

Singla, P., & Domingos, P. (2006). Memory-efficient inference in relational domains. *Proceedings of the Twenty-First National Conference on Artificial Intelligence*. Boston, MA.

Sutton, C., & McCallum, A. (2004). *Collective segmentation and labeling of distant entities in information extraction* (Technical Report TR # 04-49). University of Massachusetts.

Wick, M., Culotta, A., & McCallum, A. (2006). Learning field compatibilities to extract database records from unstructured text. *EMNLP*.