Practical Markov Logic Containing First-Order Quantifiers with Application to Identity Uncertainty

Aron Culotta and Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{culotta, mccallum}@cs.umass.edu

September 8, 2005

Abstract

Markov logic is a highly expressive language recently introduced to specify the connectivity of a Markov network using first-order logic. While Markov logic is capable of constructing arbitrary first-order formulae over the data, the complexity of these formulae is often limited in practice because of the size and connectivity of the resulting network. In this paper, we present approximate inference and training methods that incrementally instantiate portions of the network as needed to enable first-order existential and universal quantifiers in *Markov logic networks*. When applied to the problem of object identification, this approach results in a conditional probabilistic model that can reason about objects, combining the expressivity of recently introduced BLOG models with the predictive power of conditional training. We validate our algorithms on the tasks of citation matching and author disambiguation.

1 Introduction

Markov logic networks (MLNs) combine the probabilistic semantics of graphical models with the expressivity of first-order logic to model relational dependencies Richardson and Domingos (2004). They provide a method to instantiate Markov networks from a set of constants and first-order formulae.

While MLNs have the power to specify Markov networks with complex, finely-tuned dependencies, the difficulty of instantiating these networks grows with the complexity of the formulae. In particular, expressions with first-order quantifiers can lead to networks that are too large to instantiate, making inference intractable. Because of this, existing applications of MLNs have not exploited the full richness of expressions available in first-order logic.

For example, consider the database of researchers described in Richardson and Domingos (2004), where predicates include `Professor(person)`, `Student(person)`, `AdvisedBy(person, person)`, and `Published(author, paper)`. First-order formulae include statements such as "students are not professors" and "each student has at most one advisor." Consider instead statements such as "all the students of an advisor publish papers with similar words in the title" or "this subset of students belong to the same lab." To instantiate an MLN with such predicates requires existential and universal quantifiers, resulting in either a densely connected network, or a network with prohibitively many nodes. (In the latter example, it may be necessary to ground the predicate for each element of the powerset of students.)

In this paper, we present approximate inference and training methods that incrementally instantiate portions of the network as needed to enable such first-order quantifiers in MLNs.

We apply MLNs to the prevalent problem of *object identification* (also known as record linkage, deduplication, identity uncertainty, and coreference resolution). Object identification is the task of determining whether a set of constants (*mentions*) refer to the same object (*entity*). Successful object identification enables vision systems to track objects, database systems to deduplicate redundant records, and text processing systems to resolve disparate mentions of people, organizations, and locations. We present results on citation matching and author disambiguation that validate the use of complex first-order formulae in MLNs.

Although Richardson and Domingos (2004) also apply MLNs to object identification, their approach only considers predicates at the mention level. By creating first-order predicates over *objects*, our system reasons at the entity level, providing both the expressive power of the generative BLOG model in Milch et al. (2005) as well as the predictive power of discriminative models McCallum and Wellner (2003).

2 Related Work

MLNs were designed to subsume various previously proposed statistical relational models. *Probabilistic relational models* Friedman et al. (1999) combine descriptive logic with directed graphical models, but are restricted to acyclic graphs. *Relational Markov networks* Taskar et al. (2002) use SQL queries to specify the structure of undirected graphical models. Since first-order logic subsumes SQL, MLNs can be viewed as more expressive than relational Markov networks, although existing applications of MLNs have not fully utilized this increased expressivity. Other approaches combining logic programming and log-linear models include *stochastic logic programs* Cussens (2003) and MACCENTDehaspe (1997), although MLNs can be shown to represent both of these.

Viewed as a method to avoid grounding predicates, this paper is similar to recent work in *lifted inference* Poole (2003), although that work focuses on directed graphical models.

Most relevant to this work are the recent relational models of identity uncer-

tainty Milch et al. (2005); McCallum and Wellner (2003); 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, aggregate 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 all realizations of objects (their number, attributes, and observed mentions) are sampled from a generative process. While BLOG model provides attractive 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.

Object identification can also be seen as an instance of *supervised clustering*. Daumé III and Marcu (2004) present a Bayesian supervised clustering algorithm that uses a Dirichlet process to model the number of clusters. As a generative model, it has similar advantages and disadvantages as Milch et al. (2005).

In this paper, we present 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.

3 Markov logic networks

Let $F = \{F_i\}$ be a set of first order formulae with corresponding real-valued weights $w = \{w_i\}$. Given a set of constants $C = \{c_i\}$, define $n_i(x)$ to be the number of true *groundings* of F_i realized in a setting of the world given by x. A Markov logic network (MLN) Richardson and Domingos (2004) defines a joint probability distribution over possible worlds x. In this paper, we will work with discriminative MLNs Singla and Domingos (2005), for which we have a set of evidence atoms x and a set of query atoms y. Using normalizing constant Z_x, the conditional distribution is given by

$$P(Y = y|X = x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{|F_y|} w_i n_i(x, y)\right) \quad (1)$$

where $F_y \subseteq F$ is the set of clauses for which at least one grounding contains a query atom. Note that now the MLN specifies the structure of a conditional Markov network (also known as a *conditional random field* Lafferty et al. (2001)).

From Equation 1, the formulae F_y specify the structure of the corresponding Markov network as follows: Each grounding of a predicate specified in F_y has a corresponding node in the Markov network; and an edge connects two nodes in the network if and only if their corresponding predicates co-occur in a grounding of a formula F_y. Thus, the complexity of the formulae in F_y will determine the complexity of the resulting Markov network, and thus the complexity of inference. When F_y contains complex first-order quantifiers, the resulting Markov network may contain a prohibitively large number of nodes,

To address this problem, we present an algorithm that incrementally builds the structure of the network while performing inference. This algorithm iteratively grounds predicates during inference as needed so as to model existential and universal quantifiers while maintaining tractability. Below, we describe this method in more detail for a specific class of first-order quantified formulae applied to task of object identification.

3.1 Identity uncertainty

Typically, MLNs make a *unique names* assumption, requiring that different constants refer to different objects. This simplifies the network structure at the risk of weak or fallacious predictions (e.g., $\texttt{Alive}(\texttt{a}) \wedge \texttt{Dead}(\texttt{b})$ is erroneous if a and b are the same object).

Richardson and Domingos (2004) address this concern by creating the predicate $\texttt{Equals}(\texttt{x}, \texttt{y})$ between each pair of constants x, y. While this retains the coherence of the model, the restriction to pairwise predicates can be a drawback if there exist informative features over sets of constants. In particular, by only capturing features of pairs of constants, this solution cannot model the compatibility of object attributes, only of mention attributes (Section 2).

Instead, we desire a conditional model that allows predicates to be defined over a set of coreferent constants.

One approach to this would be to introduce constants that represent objects, and connect them to their mentions by predicates such as $\texttt{IsMentionOf}(\texttt{x}, \texttt{y})$. In addition to computational issues, this approach also somewhat problematically requires choosing the number of objects. (See Richardson and Domingos (2004) for a brief discussion.)

Instead, we propose creating a set of *object predicates* over sets of constants, such that a setting of these predicates implicitly determines the number of objects. Let $\mathbf{d} = \{d_i\}$ be a subset of constants. Then the object predicate $\texttt{AreEqual}(\mathbf{d})$ is true if and only if all $d_i \in \mathbf{d}$ refer to the same object. Since each

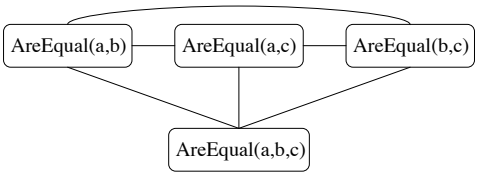

Figure 1: An example of the network instantiated by an MLN with three constants and the object predicate `AreEqual`, instantiated for all possible subsets with size ≥ 2.

subset of constants corresponds to a candidate object, a (consistent) setting of all the `AreEqual` predicates results in a solution to the object identification problem. The number of objects is chosen based on the optimal groundings of each of these object predicates, and therefore does not require a prior over the number of objects. However, a posterior over the number of objects could be modeled discriminatively in an MLN Richardson and Domingos (2004).

This formulation allows us to create evidence predicates over objects. For example, `NumberFirstNames`($\mathbf{d}$) returns the number of different first names used to refer to the object with mentions $\mathbf{d}$. In this way, we can model aggregate features of an object, capturing the compatibility among its attributes.

Naively implemented, such an approach would require enumerating all subsets of constants, ultimately resulting in an unwieldy network. In this paper, we provide algorithms to perform approximate inference and parameter estimation by incrementally instantiating these predicates as needed.

3.2 Object predicates

Object predicates are n-ary predicates that take as arguments an arbitrary number of constants. These constants represent a candidate object over which we will compute features.

Let $I_C = \{1 \dots N\}$ be the set of indices into the set of constants C, with powerset $\mathcal{P}(I_C)$. For a given C, there are therefore $|\mathcal{P}(I_C)|$ possible groundings of the `AreEqual` query predicates. We can similarly define evidence predicates over subsets of constants.

The form of the MLN defined by these predicates is the same given by Equation 1. However, by using n-ary predicates, we have increased the complexity of calculating $n_i(x, y)$, since we must search over all subsets of $|C|$ to calculate the number of times each predicate holds for setting y.

An equivalent way to state the problem is that using n-ary predicates results in a Markov network with one node for each grounding of the predicate. Since in the general case there is one grounding for each subset of C, the size of the corresponding Markov network will be exponential in $|C|$. See Figure 1 for an example instantiation of an MLN with three constants (a, b, c) and one `AreEqual` predicate.

Richardson and Domingos (2004) suggest approximating $n_i(x, y)$ using sampling. Because it is prohibitive to even generate the network with these quantified formulae, standard sampling methods seem unsuitable. Instead, we employ a deterministic approach which incrementally creates groundings for n-ary predicates, iteratively growing the number of nodes in the Markov network as needed. This is an attractive alternative to traditional sampling, since it instantiates structures based on how likely they are to contain true predicates.

3.3 Inference

Inference seeks the solution to $y^* = \text{argmax}_y P(Y = y|X = x)$ where y^* is the setting of all the query predicates F_y (e.g. `AreEqual`) with the maximal conditional density.

Whereas in many highly connected models the computational bottleneck is in computing the normalizer Z_x; here, we cannot even instantiate the graph, much less calculate Z_x. For this reason, we cannot employ traditional approximate inference techniques such as loopy belief propagation or Gibbs sampling.

Instead, we employ an incremental inference technique which iteratively fixes predicate values to their MAP estimate given the setting of the other predicates. Based on this setting, the set of F_y predicates is expanded. This can be thought of as a breadth-first search through the instantiations of predicates F_y, and is also related to the *Metropolis–Hastings* method used in Pasula et al. (2003).

At step t of the inference algorithm, let $F^t \subseteq F_y$ be the set of object predicates representing a partial solution to the object identification task for constants C, and let the setting of F^t be specified by y^t.

At step t, we can calculate an unnormalized score for the current setting y^t given the evidence predicates specified by x as follows:

$$S(y^t, x) = \exp\left(\sum_{i=0}^{|F^t|} w_i n_i(x, y^t)\right)$$

Define setting $y^t_{\mathbf{d}}$ to differ from y^t only by the fact that $F(\mathbf{d}) = 0$ in y^t and $F(\mathbf{d}) = 1$ in $y^t_{\mathbf{d}}$; that is, $y^t_{\mathbf{d}}$ merges the constants specified by $\mathbf{d}$. At step t, the algorithm exhaustively solves $\mathbf{d}^* = \text{argmax}_{\mathbf{d}} S(y^t_{\mathbf{d}}, x)$ and sets the corresponding grounded predicate $F(\mathbf{d}^*)$ to 1.

Based on this solution, the set of groundings of F^t is expanded to F^{t+1} as follows: Let $\hat{F}^t(\cdot)$ be the set of grounded predicates that have been set to 1 thus far. Then $\hat{F}^t(\cdot)$ induces a clustering of $|C|$ — e.g., if $F(a, b) = 1 \wedge F(c, d) = 1$, then the predicted clusters are (a, b) and (c, d). Based on this clustering, consider merging all possible pairs of clusters. Each of the merges under consideration has a corresponding predicate, e.g. $F(a, b, c, d)$. Then, F^{t+1} contains the union of these new predicates and those in F^t.

The algorithm terminates when there is no $F(\mathbf{d})$ that can be set to 1 and improve the score function, i.e. $\max_{\mathbf{d}} S(y^t_{\mathbf{d}}, x) \leq S(y^t, x)$.

In this way, the final setting of F_y is a local maximum of the score function. As in other search algorithms, we can employ look-ahead to reduce the

greediness of the search (i.e., consider multiple merges simultaneously). This approximation is similar to recent work on *correlational clustering* Bansal et al. (2004).

3.3.1 Pruning

The space required for the above algorithm scales $\Omega(|x|^2)$, since in the initialization step we must ground a predicate for each pair of constants. We can use the *canopy method* of McCallum et al. (2000), which thresholds a "cheap" similarity metric to prune necessary comparisons. This pruning can be done at subsequent stages of inference to restrict which predicates variables will be introduced.

Additionally, we must ensure that predicate settings at time t do not contradict settings at $t-1$ (e.g. if $F^t(a,b,c)=1$, then $F^{t+1}(a,b)=1$). The inference algorithm fixes such predicates to maintain consistency and prunes them from the search space.

3.4 Parameter estimation

Given a dataset $\mathcal{D}$ of mentions annotated with their referent objects, we would like to estimate the value of w that maximizes the likelihood of $\mathcal{D}$. That is $w^* = \text{argmax}_w \, P_w(y|x)$.

When the data are few, we can explicitly instantiate all $F(\mathbf{d})$ predicates, setting their corresponding nodes to the values implied by $\mathcal{D}$. The likelihood is given by Equation 1, where the normalizer is $Z_{\mathbf{x}} = \sum_{y'} \exp\left(\sum_{i=1}^{|F'_y|} w_i n_i(x,y')\right)$.

Although this sum over y' to calculate $Z_{\mathbf{x}}$ is exponential in $|\mathbf{y}|$, many inconsistent settings can be pruned as discussed in Section 3.3.1.

However, in general instantiating the entire set of predicates denoted by y and calculating $Z_{\mathbf{x}}$ is infeasible. Existing methods for MLN parameter estimation include pseudo-likelihood and voted perceptron Richardson and Domingos (2004); Singla and Domingos (2005). We instead follow the recent success in *piecewise training* for complex undirected graphical models Sutton and McCallum (2005) by making the following two approximations. First, we avoid calculating the global normalizer $Z_{\mathbf{x}}$ by calculating local normalizers, which only sum over the two values for each predicate *grounded in the training data*. We therefore maximize the sum of *local* probabilities for each query predicate given the evidence predicates. Gradient descent is performed on the resulting convex likelihood function using L-BFGS, a second-order approximation method Liu and Nocedal (1989).

Secondly, since all predicates cannot be instantiated, we sample a subset $F_S \in F$ and maximize the likelihood of this subset. The sampling is not strictly uniform, but is instead obtained by collecting the predicates created while performing object identification using a weak method (e.g. string comparisons). This both ensures that predicates of many different arities will be sampled, and also generates the type of predicates likely to be seen during inference.

	Objects			Pairs		
	pr	**re**	**f1**	**pr**	**re**	**f1**
constraint	85.8	79.1	**82.3**	63.0	98.0	76.7
reinforce	97.0	90.0	**93.4**	65.6	98.2	78.7
face	93.4	84.8	**88.9**	74.2	94.7	83.2
reason	97.4	69.3	81.0	76.4	95.5	**84.9**

Table 1: Precision, recall, and F1 performance for Citeseer data.

	Objects			Pairs		
	pr	**re**	**f1**	**pr**	**re**	**f1**
miller d	73.9	29.3	41.9	44.6	1.0	**61.7**
li w	39.4	47.9	**43.2**	22.1	1.0	36.2
smith b	61.2	70.1	**65.4**	14.5	1.0	25.4

Table 2: Precision, recall, and F1 performance on Author data.

4 Experiments

We perform experiments on two object identification 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 were manually labeled with cluster identifiers, and the strings were segmented into fields such as author, title, etc.

Using first-order logic, we create a number of object predicates such as `AllTitlesMatch`, `AllAuthorsMatch`, `AllJournalsMatch`, etc., as well as their existential counterparts, `ThereExistsATitleMatch`, etc. We also include *count* templates, which indicate the number of these matches in a cluster.

Additionally, we add edit distance templates, which calculate approximate matches[1] between title fields, etc., for each citation in a cluster. Aggregate features are used for these, such as "there 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%." Table compares the performance of our model (`Objects`) with a model that only considers pairwise predicates of the same features (`Pairs`).

Author disambiguation is the task of deciding whether two strings refer to the same author. To increase the task complexity, we collected 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 collected 400 citations referring to 56 unique authors.

We generated object predicates similar to those used for citation matching.

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

Additionally, we included 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 object predicates: For example, consider a pairwise predicate 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 potential of this method.

The results show that `Objects` obtains consistent improvement in precision, while `Pairs` generally has higher recall. Overall, `Objects` achieves superior $F1$ scores on 5 of the 7 datasets. These results suggest the potential advantages of using complex first-order quantifiers in MLNs.

5 Conclusions and Future Work

We have demonstrated an algorithm that enables practical inference in MLNs containing first-order existential and universal quantifiers, and have demonstrated the advantages of this approach on two real-world datasets. Future work will investigate efficient ways to improve the approximations made during inference, as well as to discriminatively model the true number of objects.

6 Acknowledgments

Thanks to Pallika Kanani for helpful discussions. 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

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learining*, 56:89–113, 2004.

J. Cussens. Individuals, relations and structures in probabilistic models. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 126–133, Acapulco, Mexico, 2003.

Hal Daumé III and Daniel Marcu. Supervised clustering with the dirichlet process. In *NIPS'04 Learning With Structured Outputs Workshop*, Whistler, Canada, 2004.

L. Dehaspe. Maximum entropy modeling with clausal constraints. In *Proceedings of the Seventh International Workshop on Inductive Logic Programming*, pages 109–125, Prague, Czech Republic, 1997.

Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

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

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(3, (Ser. B)):503–528, 1989.

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

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 (KDD-2000)*, Boston, MA, 2000.

Brian Milch, Bhaskara Marthi, and Stuart Russell. BLOG: Relational modeling with unknown objects. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields*, 2004.

Brian Milch, Bhaskara Marthi, and Stuart Russell. BLOG: Probabilistic models with unknown objects. In *IJCAI*, 2005.

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

Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

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

M. Richardson and P. Domingos. Markov logic networks. Technical report, University of Washington, Seattle, WA, 2004.

Parag Singla and Pedro Domingos. Discriminative training of markov logic networks. In *Proceedings of the Twentieth National Conference of Artificial Intelligence*, Pittsburgh, PA, 2005.

Charles Sutton and Andrew McCallum. Piecewise training of undirected models. In *Submitted to 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

Ben Taskar, Abbeel Pieter, and Daphne Koller. Discriminative probabilistic models for relational data. In *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference (UAI-2002)*, pages 485–492, San Francisco, CA, 2002. Morgan Kaufmann Publishers.