Cutting Plane MAP Inference for Markov Logic

Keywords: MAP inference, Markov Logic, Cutting Planes

Sebastian Riedel RIEDEL@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, USA

Abstract

In this work we present *Cutting Plane Inference* (CPI) for MAP inference in Markov Logic. CPI incrementally solves partial Ground Markov Networks, adding formulae only if they are violated in the current solution. We show dramatic improvements in terms of efficiency, and discuss scenarios where CPI is likely to be fast.

1. Introduction

In this work[1] we present and analyse a MAP inference algorithm for Markov Logic (ML), namely *Cutting Plane Inference* (CPI) (Riedel, 2008), that exploits the *redundancy* inherent in many SRL models. Often several ground features in the Ground Markov Network encourage the same properties of a solution. For example, in pairwise Entity Resolution tasks we may encounter simple *local* formula (say, based on the string distance of two names) that implicitly encourage transitive solutions. Formulae that explicitly enforce transitivity are hence often unnecessary.

2. Markov Logic

Markov Logic (Richardson & Domingos, 2006) is a Statistical Relational Learning language based on First Order Logic and Markov Networks. It can be seen as a formalism that extends First Order Logic to allow formulae that can be violated with some penalty. From an alternative point of view, it is an expressive template language that uses First Order Logic formulae to instantiate Markov Networks of repetitive structure.

In Markov Logic we call a set of weighted formulae a *Markov Logic Network* (MLN). Formally speaking, an MLN L is a set of pairs (ϕ, w) where ϕ is a first order formula and w a positive real weight.[2] L assigns the probability

$$p(\mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{(\phi,w)\in L} w \sum_{\mathbf{c}\in C^\phi} f^\phi_{\mathbf{c}}(\mathbf{y}) \right) \qquad (1)$$

to the set of ground atoms (aka *possible world*) $\mathbf{y}$. Here C^ϕ is the set of all possible bindings of the free variables in ϕ with the constants of our domain. $f^\phi_{\mathbf{c}}$ is a feature function that returns 1 if in the possible world $\mathbf{y}$ the *ground formula* we get by replacing the free variables in ϕ by the constants in $\mathbf{c}$ is true and 0 otherwise. Z is a normalisation constant. Note that the above distribution corresponds to the *Ground Markov Network* where nodes represent ground atoms and factors represent ground formulae.

3. Cutting Plane Inference

MAP inference in Markov Logic amounts to finding a set of hidden ground atoms $\hat{\mathbf{y}}$ with maximum *a posteriori* probability given a set of observed ground atoms $\mathbf{x}$ and a Markov Logic Network L. This amounts to finding the $\mathbf{y}$ with maximal score

$$s(\mathbf{y}, \mathbf{x}) = \sum_{(\phi,w)\in L} \sum_{\mathbf{c}\in C^\phi} w \cdot f^\phi_{\mathbf{c}}(\mathbf{y}, \mathbf{x}). \qquad (2)$$

Before we can present Cutting Plane Inference we need to introduce two concepts. First, we define the set $\text{Separate}(\phi, \mathbf{y}, \mathbf{x})$ for a formula ϕ, hidden ground atoms $\mathbf{y}$ and observed ground atoms $\mathbf{x}$, to be set of variable bindings $\mathbf{c}$ so that $\phi[\mathbf{c}]$ is false in the world $\mathbf{y} \cup \mathbf{x}$.

Second, to compactly represent the partial networks generated during CPI we introduce the notion of a *partial grounding* $\mathbf{G} = (G_\phi)_{(\phi,w)\in L}$ that maps each

[1] Note that this paper contains a summary of (Riedel, 2008), slightly updated results and a novel analysis of CPI.

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

[2] We can convert formulae with negative weights to negated formulae with positive weights.

first order formula ϕ of an MLN L to a set of variable assignments G_ϕ. A partial grounding $\mathbf{G}$ induces a *partial score*

$$s_{\mathbf{G}}(\mathbf{y}, \mathbf{x}) = \sum_{(\phi, w) \in L} w \sum_{\mathbf{c} \in G_\phi} f_{\mathbf{c}}^{\phi}(\mathbf{y}, \mathbf{x}). \quad (3)$$

Algorithm 1 Cutting Plane Inference for ML (Riedel, 2008).

Require: MLN L, base solver BS, observation $\mathbf{x}$, initial grounding $\mathbf{G}^0$, $maxIterations$

1: $i \leftarrow 0$
2: $\mathbf{y}' \leftarrow \mathbf{0}$
3: **repeat**
4: $\quad i \leftarrow i + 1$
5: $\quad \mathbf{y} \leftarrow$ solve $\left(\mathbf{G}^{i-1}, \mathbf{x}\right)$ using base solver BS.
6: $\quad$ **if** $s(\mathbf{y}, \mathbf{x}) > s(\mathbf{y}', \mathbf{x})$ **then**
7: $\quad\quad \mathbf{y}' \leftarrow \mathbf{y}$
8: $\quad$ **end if**
9: $\quad$ **for** each $(\phi, w) \in L$ **do**
10: $\quad\quad G_\phi^i \leftarrow G_\phi^{i-1} \cup$ Separate $(\phi, \mathbf{y}, \mathbf{x})$
11: $\quad$ **end for**
12: **until** $\mathbf{G}^i = \mathbf{G}^{i-1}$ or $i > maxIterations$
13: **return** $\mathbf{y}'$

Algorithm 1 shows CPI in pseudo-code. It needs a propositional *base solver* as input—this could be any MAP solver for Markov Networks. In a nutshell, it starts by solving the initial grounding using the base solver, then adding violated formulae, solving the next grounding, and so forth. Notice that we return the best solution generated during CPI; only for exact base solvers this is guaranteed to be solution of the final solution.

The loop in steps 9 and 10 finds the ground formulae Separate $(\phi, \mathbf{y}, \mathbf{x})$ which are violated in the current solution $\mathbf{y} \cup \mathbf{x}$, and adds them to the current partial grounding. In cutting plane algorithms this step is usually called *separation*. In CPI separation amounts to the *evaluation problem* and can be efficiently solved for a large class of formulae. We tackle it by using a database representation of the atoms, and execute database queries on this database. In practise we observed that the cost of query evaluation was marginal when compared to the cost of numeric optimisation.

We refer to our algorithm as Cutting Plane Inference due to its resemblance to cutting plane algorithms for Operations Research. These algorithms iteratively add violated linear constraints (cuts) to an optimisation problem. Our approach is therefore also close to recent work that uses cutting plane algorithms for MAP inference in the Marginal Polytope (e.g., by Sontag and Jaakkola (2007)). However, by exploiting first order information CPI scales up to problems with millions of nodes and edges; here the separation routines of Sontag et al. would fail. Our first-order approach also avoids the construction of the ground network—a process that is very expensive and cannot be avoided in propositional cutting plane algorithms. A further advantage of CPI is the fact that we can plug-in any Markov Network MAP algorithm as base solver.

CPI is also related to Lazy Inference (Poon et al., 2008) in the sense that both instantiate edges only when needed by a base solver. However, while Lazy Inference provides edges whenever the base solver calls for them, CPI is less generous: it requires the base solver to run to completion on the current network before new edges can be requested. This can have advantages when the base solver requires edges based on some suboptimal moves (e.g., random moves in MaxWalkSAT) that could be compensated after further moves on the same problem (e.g., after some further random moves that find a better configuration). Note that Lazy Inference has a practical disadvantage: it requires the developer to change the implementation of the algorithm to be made lazy. This is very difficult for complex software such as ILP solvers.

4. Empirical Results

The first task we apply CPI to is Semantic Role Labelling (Carreras & Marquez, 2005), the task of identifying and classifying the semantic arguments of predicates in a sentence. The MLN we use for this problem is inspired by previous work on Semantic Role Labelling. The size in terms non-local factors per problem instance is roughly 10^5. This size is a result of a formula that forbids overlapping spans to be semantic arguments.

Table 1 shows the effect of using CPI with two types of base solver: Integer Linear Programming and MaxWalkSAT (MWS) (with 1000 flips and 10 restarts). We notice a dramatic reduction of ground formulae that comes along with runtime improvements of two orders of magnitude for Integer Linear Programming, and slightly less for MWS. Also notice that in case of the approximate MWS we in fact improve (score) accuracy when using CPI.

Next we look at an MLN for citation matching (Singla & Domingos, 2005). Here we are to find citation pairs that refer to the same publication. In this case the actual Ground Markov Networks we consider have about a million factors (and hundreds of thousands of nodes). The main reason for networks being this large is a transitivity clause that requires A to match with C if A matches with B and B with C.

System	Calls	Time	Size	Score	F1
MWS	1	5.7	$1.3\ 10^5$	0.762	0.74
ILP	1	4.6	$1.3\ 10^5$	0.834	0.79
CPI-MWS	2.24	0.11	$8.6\ 10^0$	0.829	0.79
CPI-ILP	2.25	0.065	$8.6\ 10^0$	0.834	0.79

Table 1. Semantic Role Labelling results. Calls to the propositional optimiser, avg. solving time for each instance (in seconds), number of ground formulae, linear score (equation 2) , F1 measure. Averaged over the first 100 examples in WSJ test set.

System	Calls	Time	Size	Score	F1
MWS	1	4.3	$2.0\ 10^6$	-890	0.21
ILP	1	60	$2.0\ 10^6$	N/A	N/A
CPI-MWS	20	2.7	$7.3\ 10^4$	(364)	0.27
CPI-ILP	6.7	1.73	$1.9\ 10^4$	3030	0.72

Table 2. Citation Matching results. Columns as in table 1, but times measured in minutes. Bracketed score for CPI-MWS shows number of hard constraint violations.

Table 2 tells a similar story to table 1, at least for the case of ILP. When applied to the full ground network, our ILP solver failed to even return a solution, due to the large memory requirements of the ILP version of the Ground Markov Network. By contrast, with CPI ILP becomes feasible, leading to exact results for networks with millions of edges and nodes. Interestingly, the time we need to find the exact solutions using CPI-ILP is less than 50% of the time of MWS in the full propositional network. However, note that MWS alone performs poorly in this setting. This led CPI-MWS to run astray, and so we terminated CPI after 20 iteration. This resulted in poor accuracy (and many violated transitivity formulae).

5. Analysis

It can be easily shown that CPI inherits the accuracy of its base solver (and is thus exact if its base solver is), and that it converges in a finite number of steps for finite domains (Riedel, 2008). However, so far the general class of MLNs that can be efficiently solved by CPI has not been investigated. In this section we try to sketch a first characterisation of this class. In particular, we ask the question when the final Markov Network of CPI will be definitely small.

We claim the following (and refer to (Riedel, 2009) for a proof and more thorough analysis): assuming an exact base solver and each solution during CPI to be unique, then a ground formulae $\phi[\mathbf{c}]$ can only be instantiated during CPI if there exists a set of ground formulae P that logically entails $\neg\phi[\mathbf{c}]$. Moreover, it can also be shown that even if $\neg\phi[\mathbf{c}]$ can be entailed, it will definitely not be instantiated if the weights of the formulae in P are relatively low compared to weights of formulae that contradict P. Hence final networks will be small if ground formulae cannot refute each other, or cannot refute each other with confidence.

In our citation matching example the above statement means that if there is not much positive local evidence for matched citations, CPI will only instantiate a small subset of transitivity clauses. Why is this so? Without local evidence for matches we cannot prove the antecedents of transitivity clauses. This in turn means that we cannot refute these clauses, and hence CPI will not instantiate them.

6. Conclusion

We have presented CPI, a MAP Inference algorithm for Markov Logic that can dramatically improve efficiency. CPI exploits the redundancy of a model by using complex factors only when they contradict preferences of the simple ones. The algorithm alternates between calling a Markov Network MAP solver of choice, and performing first order query processing with respect to the returned solution.

We show dramatic improvements in efficiency for Semantic Role Labelling and Entity Resolution. We also briefly discuss a scenario where CPI is guaranteed to lead to small partial problems (and hence is likely to be fast). In future work we seek to extend the idea of CPI to marginal inference, combine it with other lifted inference techniques (such as Lifted Belief Propagation) and empirically compare CPI to Lazy Inference.

References

Carreras, X., & Marquez, L. (2005). Introduction to the conll-2005 shared task: Semantic role labeling. *CoNLL' 05.*

Poon, H., Domingos, P., & Summer, M. (2008). A general method for reducing the complexity of relational inference and its application to mcmc. *AAAI '08.*

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

Riedel, S. (2008). Improving the accuracy and efficiency of MAP inference for markov logic. *UAI '08.*

Riedel, S. (2009). *Efficient prediction of relational structure and its application to natural language processing.* Doctoral dissertation, University of Edinburgh.

Singla, P., & Domingos, P. (2005). Discriminative training of markov logic networks. *AAAI '05.*

Sontag, D., & Jaakkola, T. (2007). New outer bounds on the marginal polytope. *NIPS '07.*