# Scalable Gibbs Sampling for Domain Approximated MLNs

By: Kyle Cherry

Advisor: Deepak Venugopal

## Abstract

Up until the mid-2000's, a long sought after goal of AI was to unify the power of probability, or uncertainty, with first-order logic. This goal was met with the introduction of Markov Logic Networks (MLNs). Since its inception scalability of inference has been a key issue in applying MLNs to real world applications. Lifted inference, and more recently, domain approximation (or approximate lifting) have been key developments that aid in scaling up inference in MLNs. Specifically, these techniques create an approximate version of the original MLN by clustering the atoms in the original into "meta-atoms", reducing the size of the original MLN. However, performing inference on these domain-approximated MLNs remains a difficult task. In this project we propose two novel methods based on Gibbs sampling to perform marginal inference over domain-approximated MLNs. Further, several benchmarks will be used to clearly demonstrate the promise of my approaches in terms of scalability, accuracy, and convergence.

## Introduction

Markov Logic Networks [1] provide a compact way of representing uncertain knowledge in the form of weighted first-order sentences. Specifically, MLNs define a template model that encodes an undirected probabilistic graphical model, or Markov network, compactly. MLNs have been shown to be applicable to a vast number of domains such as natural language understanding [3] and computer vision [4]. It is well known that probabilistic graphical models are powerful for inference tasks but suffer from scalability problems as the model grows. Similarly, MLNs suffer from this problem at an even larger scale since they typically encode extremely large graphical models, and as the domain (real world objects that can be instantiated in the model) of the first-order formulas grow, the variables in the underlying Markov network grow exponentially [2]. Thus, In order to utilize MLNs to solve large real-world problems, inference scalability is the key problem that must be solved.

A major advancement in scalable inference for MLNs was the idea of lifted inference [7]. Due to the first-order structure of the MLN, it turns out that there are several symmetrical, or exchangeable, variables in the probability distribution. This essentially means that we can perform inference on a small subset of variables and project the same results over the other variables due to the exchangeability property. However, though sound in theory, many real-world MLNs do not have such symmetrical variables and when presented with evidence, any symmetries present are likely to be broken. More recently, a new approach called approximate lifting, or domain approximation, was introduced where the key idea is to use "approximate symmetries" to lift the inference computations in several practical MLNs [2,8]. Specifically, machine learning techniques such as clustering [2] or low-rank Boolean matrix factorization [8]

were used to the atoms of the original MLN into meta-atoms by using domain approximation such that a single variable in the new distribution represents several variables in the original.

In this project, we present two novel sampling based approaches to perform marginal inference over domain approximated MLNs. These approaches are both based on Gibbs sampling, which is the most widely used sampling based approximate inference algorithms for graphical models. Specifically, in our first approach, we adapt the selection probabilities in the Gibbs sampler based on the evidence given to the MLN. Furthermore, if a meta-atom is associated with a larger amount of observed evidence, it should be sampled less often. In the next approach, we modify the stationary distribution of the Gibbs sampler by adding an informed prior term to the model, which is also based on the evidence characteristics of the meta-atoms. We run several experiments on MLN benchmarks from Alchemy [6] to illustrate and compare the accuracy, scalability, and convergence properties of our methods.

# Background

### First-Order Logic

First-order logic is a formal way of representing knowledge using predicates, logical symbols ($\vee$, $\wedge$, $\neg$, $\Rightarrow$), and quantifiers ($\forall$, $\exists$). A predicate (ex. Friends($x$,$y$)) is a named entity that can take any number of parameters, representing the relationship between objects. In this example there is a "Friends" relationship between "*x*" and "*y*". The variables "*x*" and "*y*" represent any object from the domain in which each variable is defined over. A simple example of a first-order logic sentence might be: $\forall x,y,z$ Friends($x$,$y$) $\wedge$ Friends($x$,$z$) $\Rightarrow$ Friends($y$,$z$). Here, this formula is saying that if $x$ and $y$ are friends and $x$ and $z$ are friends, then $y$ and $z$ are friends. Furthermore, if each of these variables is substituted with a constant from their respective domains, it becomes a ground formula. Assuming $\Delta x,y,z$ = {Bob, Alice, Tom, Jane}, a possible grounding of this formula would be: Friends(Bob,Alice) $\wedge$ Friends(Bob,Jane) $\Rightarrow$ Friends(Alice,Jane). Now, if the formula is instantiated with all possible domain elements, it becomes a grounding of the knowledge base (KB), where here the KB consists of only one sentence.

### Markov Logic

Markov Logic is an extension of first-order logic that allows for uncertainty in a first-order sentence. The above example formula has hard constraints that cannot be broken. If the first two predicates are true then the last must be true. However, in Markov Logic these hard constraints are relaxed by the introduction of a weight ($w$) corresponding to the first-order formula. A "soft" formula is a pair $(f, w)$ where $f$ is a first-order formula and $w$ is a corresponding weight that describes how much belief is given to the corresponding formula. A

Markov Logic Network, denoted by $M$, is a set of $(f, w)$ pairs along with a given domain of interest. Grounding the MLN, that is instantiating the formulas with all possible domain objects, yields an undirected graphical model (Markov network). The MLN represents the following probability distribution:

$$P_M(\omega) = \frac{1}{Z(M)} \exp\left(\sum_i w_i N(f_i, \omega)\right)$$

where $\omega$ is a world (truth assignment to every ground atom), $N(f_i, \omega)$ is the number of true groundings of formula $f_i$ in world $\omega$ and $Z(M)$ is the partition function [2].

The most interesting problem in graphical model inference, and what we are concerned with here, is computing the marginal probabilities. For example, consider the transitive MLN with 1 formula: Friends(*x*,*y*) ∧ Friends(*x*,*z*) ⇒ Friends(*y*,*z*) , a weight 0.75, and a domain Δ*x*,*y*,*z* = {Bob, Alice, Tom, Jane}. Then, one might be interested in

$$P(Friends(Bob, Alice) | Friends(Alice, Tom), \neg Friends(Tom, Bob))$$

where P($Friends(Bob, Alice)$) is the marginal probability being conditioned on some evidence.

There are several existing inference algorithms for MLNs and these can be grouped as either propositional or lifted algorithms. Propositional algorithms work on the Markov network obtained by grounding the MLN while lifted algorithms work directly on the first-order sentences by exploiting symmetrical variables in the distribution, grounding the formulas only when necessary [8,9,10,11]. Propositional algorithms, such as Gibbs sampling and Belief Propagation, do not scale because computing the marginal probabilities in these large networks becomes infeasible as the domain size grows. On the other hand, lifted algorithms such as Lifted Belief Propagation [11] and Lifted Gibbs sampling [10] work better on large domain sizes until evidence is introduced. Evidence causes the symmetrical structure the first-order formulas have in the ground MLN to be broken and thus turning it into the nearly completely grounded MLN that the propositional methods would use. In order to utilize MLNs without restrictions to the evidence presented, previous approaches have sought to "approximately lift" propositional algorithms by finding approximate symmetries in the MLN distribution [2,8,9].

**Gibbs Sampling**

Gibbs sampling is a well-known Markov Chain Monte Carlo algorithm for approximate inference in probabilistic graphical models. For MLNs, estimating $P_M(\omega)$ is very difficult since the number of variables in $\omega$ is typically extremely large and computing Z is intractable. Instead, Gibbs sampling constructs a Markov chain whose stationary distribution $P_M(\omega)$ is as follows. In each step, we sample a single variable of the distribution from a conditional distribution

$P(X_i|\ X_{-i})$, where $P(X_i|\ X_{-i})$ is the probability of the randomly chosen variable $X_i$ conditioned on its directly connected neighbors in the MLN, or its Markov blanket. After enough iterations of sampling, typically called its "burnin" time, the Gibbs sampler is said to have mixed away from its randomly assigned initialization converges to $P_M(\omega)$. The mixing time of the Gibbs sampler is the time it takes to reach the stationary distribution. Mixing time depends on the total number of variables in the distribution. Therefore, performing Gibbs sampling on a grounded MLN typically has exponentially higher mixing time than Gibbs sampling on a lifted, or approximately-lifted, MLN.

## Related Work

The authors in [2] propose a method to reduce the size of the ground MLN by clustering the domains of the first-order formulas. This yields a smaller MLN in which inference algorithms such as Gibbs sampling can converge and hopefully yield minimal error when comparing the approximated probability distribution with the true distribution (if known). The methods used in this paper are standard parametric clustering algorithms (k-means, EM, etc) with a distance function that captures how similar evidence structures in the groundings are to one another. Now, with the clustered domains, once obtaining the marginal probability for a cluster (meta-atom), the same marginal probability can be assigned to all elements of the meta-atom. The results from this paper show that domain approximated MLNs are more scalable without trading off too much accuracy in distribution approximation. However, the main drawback of these methods is having to specify the model parameters.

In a currently under review conference paper submission, these domain approximation strategies were extended to overcome the drawbacks of using parametric clustering. In [5] the authors unify the standard K-means algorithm with Bayesian non-parametrics by showing that performing Gibbs sampling in a Dirichlet process mixture model to obtain an optimal number of mixture components is asymptotically equal to solving an objective function very similar to that of K-means. The main contribution is an additional penalty term λ that controls the number of clusters for each domain being approximated by allowing for a bounded error rate. This method, called DP-means, was applied to MLNs along with one of the upcoming sampling strategies in the previously mentioned paper under review.

## Sampling Strategies

As mentioned in the previous section, domain-clustering an MLN gives rise to a smaller MLN where atoms are replaced by meta-atoms. Example 1 illustrates this with an example. Here the original MLN contains 20 atoms but the domain clustered MLN contains 6 meta-atoms. The main contribution of this project is the introduction of new sampling methods that arise when working on MLNs with domain-approximated meta-atoms. Specifically, the problem is

that for the marginal inference task, evidence is specified on the original atoms of the MLN and not on the meta-atoms. The task is to systematically project this evidence on the domain-approximated distribution.

**Formulas:**
$\mathtt{R}(x) \vee \mathtt{S}(x, y), w$
**Domains:**
$\Delta_x = \{A_1, B_1, C_1, D_1\}$
$\Delta_y = \{A_2, B_2, C_2, D_2\}$
(a)

**Formulas:**
$\mathtt{R}_1(\mu_1) \vee \mathtt{S}(\mu_1, \mu_3), w$; $\mathtt{R}_1(\mu_2) \vee \mathtt{S}(\mu_2, \mu_3), w$
$\mathtt{R}_1(\mu_1) \vee \mathtt{S}(\mu_1, \mu_4), w$; $\mathtt{R}_1(\mu_2) \vee \mathtt{S}(\mu_2, \mu_4), w$
**Domains:**
$\zeta(\mu_1) = \{A_1, B_1\}$; $\zeta(\mu_2) = \{C_1, D_1\}$
$\zeta(\mu_3) = \{A_2, B_2\}$; and $\zeta(\mu_4) = \{C_2, D_2\}$
(b)

*Example 1: (a) an example MLN M and (b) MLN M' obtained from M by grounding each logical variable in M by cluster centers $\mu_1, \ldots, \mu_4$ [12]*

Prior to this work, a "hard-evidence" strategy was used to project the evidence from the original atoms to the meta-atoms [2]. Specifically, we set the meta-atom as true evidence if on at least 50% of the atoms represented by that meta-atom, the original evidence had a "true" assignment. Similarly, a meta-atom was deemed as false evidence if on at least 50% of the atoms represented by that meta-atom, the original evidence assigned had a "false" assignment. The main motivation of our work here is to develop and evaluate smarter techniques that outperform the hard-evidence approach in a domain-approximated MLN.

We start with some notation. Let $X_1, X_2, \ldots, X_N$ be the meta-atoms of the domain-approximated MLN. As mentioned earlier, each meta-atom represents a set of atoms in the original domain. Let $E = e_1, e_2, \ldots, e_N$ be a set of proportions where $e_i$ is the proportion of evidence atoms among all atoms represented by the meta-atom $X_i$.

**Method 1: Adapting the Selection Probabilities**

The selection probability refers to the probability with which a certain variable is chosen to be sampled by the Gibbs sampler. In our first new method, we change the Gibbs sampler into an adaptive Gibbs sampler such that the selection probabilities for sampling come from a distribution directly corresponding to the proportions of evidence atoms encoded by a meta-atom in the MLN. Formally, let $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ represent a distribution over the meta-atoms where $\alpha_i$ is the probability with which meta-atom $X_i$ is selected by the Gibbs sampler during a sampling iteration. For a typical Gibbs sampler, $\alpha$ is a uniform distribution which means each variable is equally likely to be changed during the sampling procedure. Here, we bias $\alpha$ based on the evidence given to the original MLN. Specifically, we set

$$\alpha_i = 1 - e_i,$$

where $X_i$ is the i-th meta-atom and $\alpha_i$ is the selection probability for $X_i$. Intuitively, the above method seeks to assign those meta-atoms that are associated with a greater amount of uncertainty a larger selection probability as compared to those meta-atoms whose truth value is more or less deterministic. Further, we want to sample uncertain meta-atoms more often than meta-atoms whose truth value we are more-or less certain of.

**Method 2: Evidence Based Prior**

The second method that we develop adjusts the stationary distribution of the Gibbs sampler based on the evidence presented to the MLN. Let $P_\mu$ be the probability distribution represented by the domain-approximated MLN. In other words, the stationary distribution of the Gibbs sampler that is run on the domain-approximated MLN is guaranteed to be $P_\mu$. To incorporate the evidence on the original MLN, we now change the stationary distribution to

$$P_{\mu'} = P_\mu * Q,$$

where $Q$ is a prior distribution computed as follows. Let ω be a world in the domain-approximated MLN. Let $\mathbb{I}(X_i, \omega)$ be an indicator function defined as,

$$\mathbb{I}(X_i, \omega) = \begin{cases} 1 \; if \; X_i \; is \; true \; in \; \omega \\ 0 \; otherwise \end{cases}$$

$$Q(\omega) = \prod_i \exp(\mathbb{I}(X_i, \omega) * e_i)$$

Intuitively, the above method assigns a larger prior value to those meta-atoms which represent a higher proportion of evidence atoms as compared to those meta-atoms representing a higher proportion of unknown atoms. This has the desired effect of projecting the evidence information from the original distribution into the stationary distribution of the Gibbs sampler that runs on the domain approximated MLN.

## Evaluation

We evaluate our two sampling approaches in terms of accuracy of approximation and convergence of the Gibbs sampler to the stationary distribution. For data, the following three benchmarks from Alchemy [6] are used: WebKB MLN that models webpage links and page topics, Protein MLN that models interaction between proteins, and ER MLN that is used for entity resolution in natural language processing. The sizes of these benchmarks are detailed in Table 1.

| Dataset | #Clauses | #Atoms | #Parameters |
|---|---|---|---|
| WebKB | 892 million | 20 million | 64 |
| Protein | 408 million | 3.3 million | 211 |
| ER | 1.7 trillion | 5.5 million | 15 |

Table 1: Dataset sizes.

We use non-parametric clustering to preprocess each MLN and derive the domain-approximated MLN. In the charts below, "gibbs" is to the standard Gibbs sampling on a non-domain approximated MLN, "cgibbs" is clustered Gibbs sampling using the "hard evidence" approach used in [2], "acgibbs" is our adaptive Gibbs sampling approach, and "pgibbs" is our approach that uses priors. To measure accuracy, we use the Hellinger distance [13], or Bhattacharyya distance, to capture how far apart the approximated probability distribution is from the "known" true distribution.

**Accuracy Results**

It is important to note that it is intractable to compute the true marginal distribution for any of the datasets in Table 1. In the plots for these datasets, the error is assuming that the standard Gibbs algorithm output (working on the ground Markov network) is the true distribution. In order to ensure that the ground Gibbs sampling outputs are reasonably accurate, we subsample the dataset and use just 10% of the full dataset while evaluating accuracy. Further, we come up with a synthetic MLN in which we know the exact marginal distribution. We next describe our results in both of these cases.

Figures 1-2 show error measurements for our synthetic MLN with a single formula for which the known true distribution is computable. These figures show that the two proposed methods for this project greatly impacted the speed at which the true error approached 0, with the better of the two varying as the model size grew. Figure 3 shows accuracy results on the WebKB benchmark. The two proposed methods in this project both outperform the hard-evidence method used in [2]. Figure 4 shows accuracy results on the Protein benchmark. This graph indicates that these methods do not always work better than the hard-evidence method.

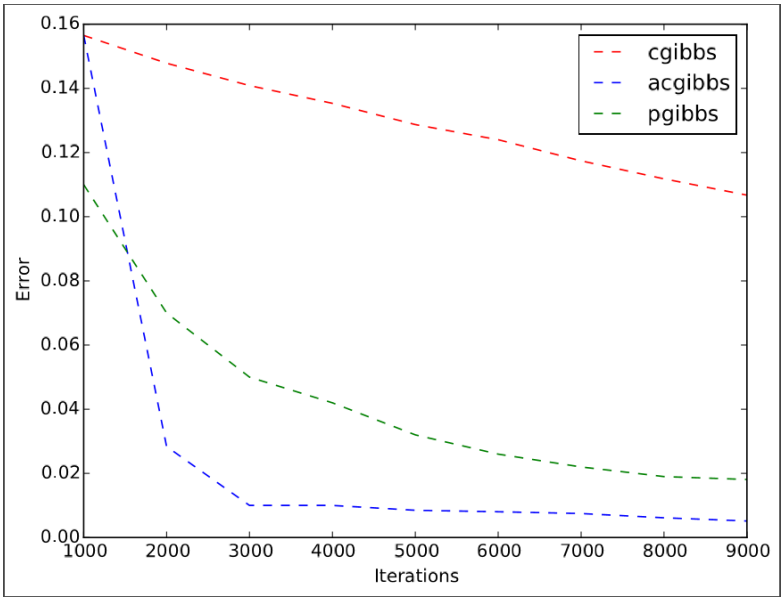


*Figure 1: R(X) => S(X);  20k atoms; 2000 meta-atoms.*

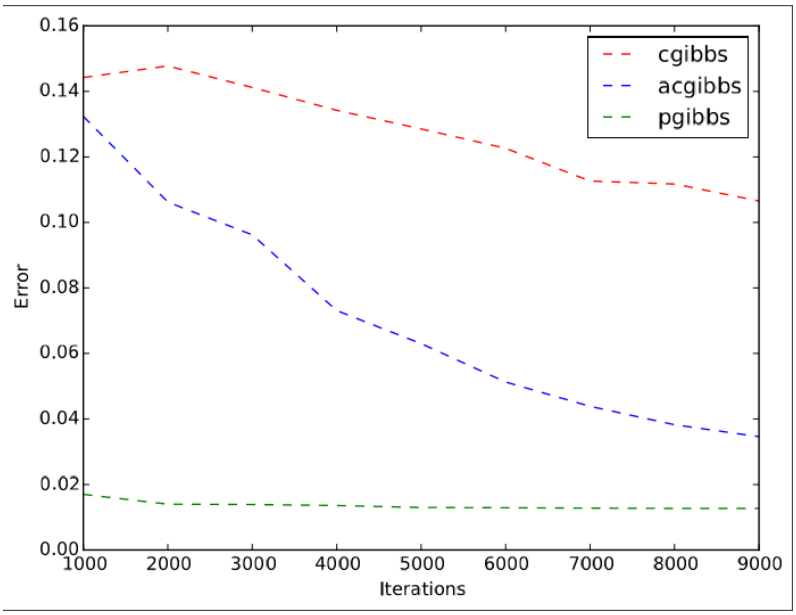


*Figure 2: R(X) => S(X); 20k atoms; 1000 meta-atoms.*

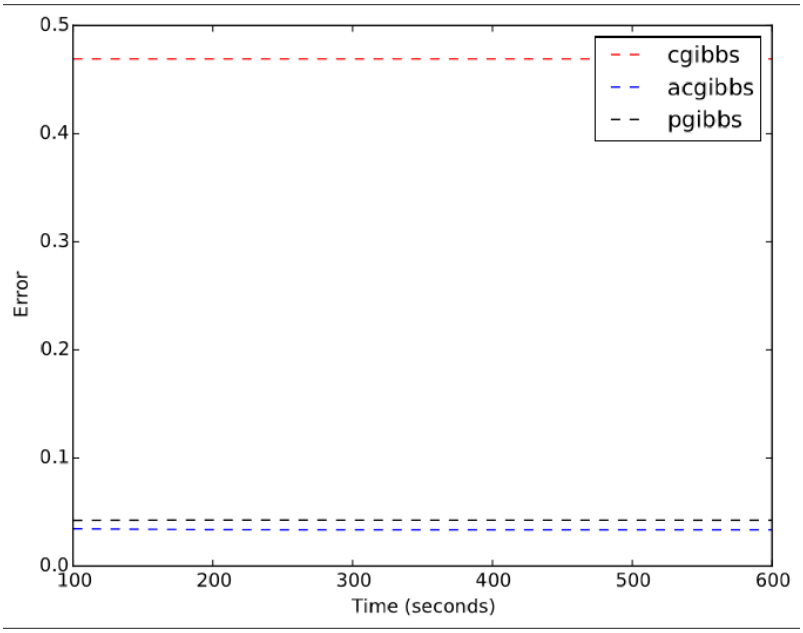


*Figure 3: WebKB benchmark with 10% evidence*

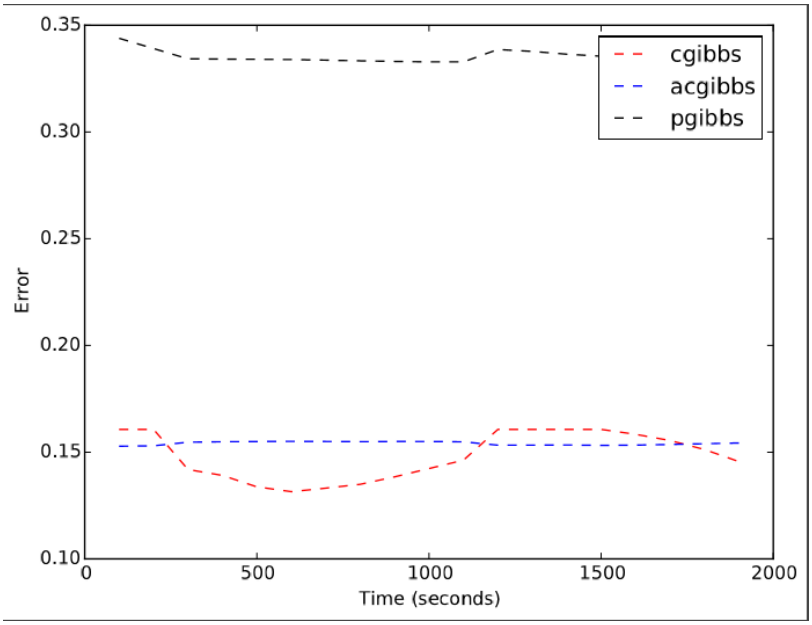


*Figure 4: Protein benchmark with 10% evidence*

**Convergence Results**

To measure convergence of the Markov chain generated by the Gibbs sampler, the Gelman-Rubin (G-R) statistic [14] is used. The G-R statistic is a metric that uses within chain and across chain variances to show how well a MCMC algorithm is mixing away from its initialization to a stationary distribution. As the G-R statistic decreases, this indicates that over time the

approximated distribution is approaching a stationary distribution. To compute the G-R statistic for these experiments there were three independent Markov chains used. Each run would randomly sample a set of 1000 atoms in the MLN and output their current marginal probabilities every 100 iterations. The G-R statistic was computed for each variable and shown in the figures is the mean G-R statistic. Figures 5-7 show how well each benchmark from Table 1 converged when presented with 10% evidence. Figure 5 shows that the ER benchmark converged only with the pgibbs method. Figures 6 and 7 show that both new methods converge and pgibbs again outperforms the rest.

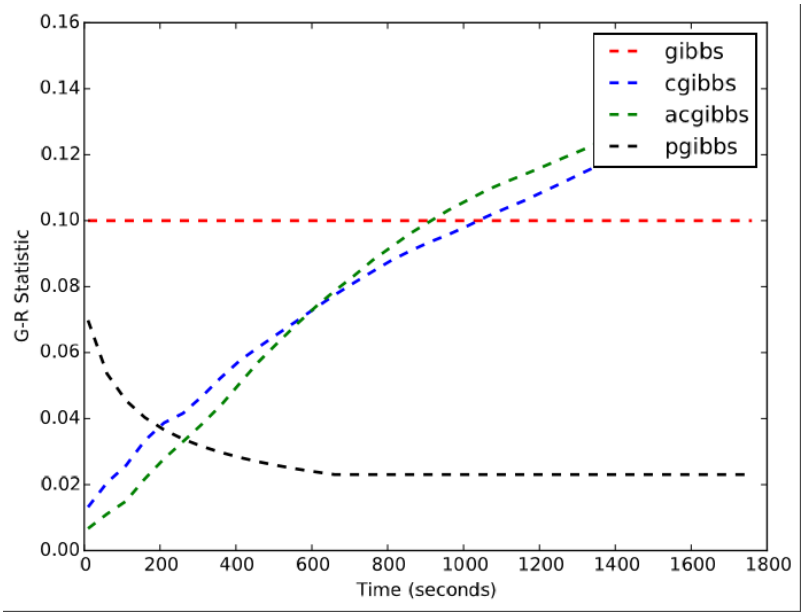


*Figure 5: ER benchmark with 10% evidence.*

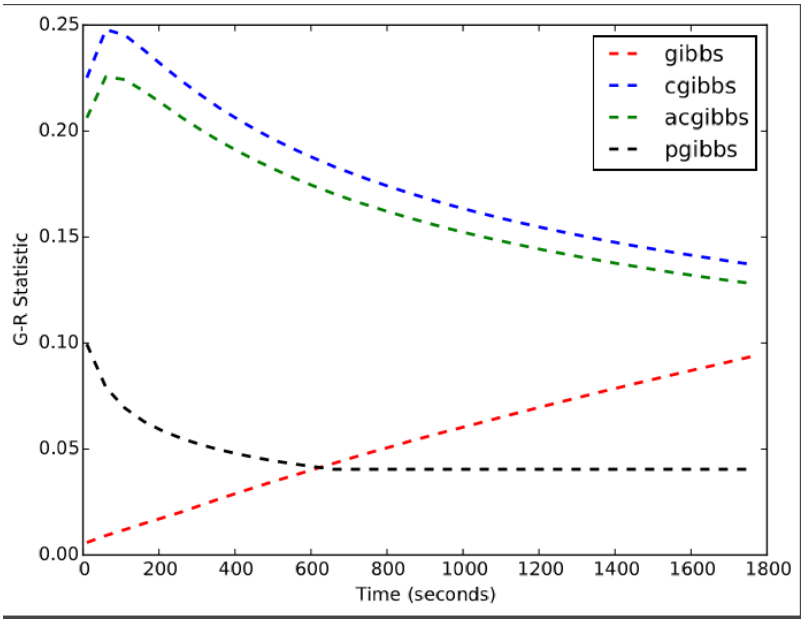

*Figure 6: Protein benchmark with 10% evidence.*

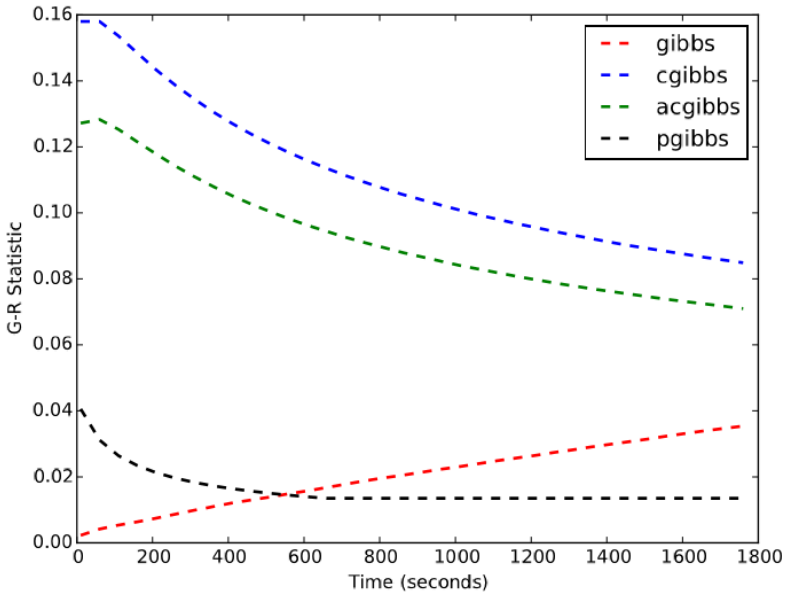


*Figure 7: WebKB benchmark with 10% evidence.*

## Future Work

There is much more work could be done in continuation to this project. There are more sophisticated clustering techniques that could be applied for domain approximation. These techniques might show better promise for accuracy and convergence. Also, more theory can be developed to better explain how certain structures of MLNs behave with these two methods.

## References


1. Domingos, P., Richardson, M. Markov Logic Networks. Machine Learning, 62, 107-136, 2006
2. Venugopal, D., Gogate, V., Evidence-based clustering for scalable inference in Markov logic. In ECML PKDD, 2014.
3. Poon, H., Domingos, P.: Joint Unsupervised Coreference Resolution with Markov Logic. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 649–658. ACL (2008)
4. Tran, S.D., Davis, L.S.: Event modeling and recognition using Markov logic networks. In: 10th European Conference on Computer Vision. pp. 610–623 (2008)
5. Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. ICML, 2012.
6. S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, and P. Domingos. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science

and Engineering, University of Washington, Seattle, WA, 2006. http://alchemy.cs.washington.edu.

7. David Poole, First-order probabilistic inference, Proc, IJCAI-03, Acapulco, August 2003, pp. 985-991.
8. Guy van den Broeck and Adnan Darwiche. On the complexity and approximation of binary evidence in lifted inference. In: Advances in Neural Information Processing Systems 26. pages 2868–2876, 2013.
9. Parag Singla, Aniruddh Nath, and Pedro Domingos. Approximate Lifting Techniques for Belief Propagation. Pages 2497–2504, 2014.
10. D. Venugopal and V. Gogate. On lifting the Gibbs sampling algorithm. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS), pages 1664–1672, 2012.
11. P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. pages 1094–1099, Chicago, IL, 2008. AAAI Press.
12. Venugopal, D., & Gogate, V. Scaling-up Importance Sampling for Markov Logic Networks. NIPS 2014.
13. Nikulin, M.S. (2001), "Hellinger distance", in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4
14. Gelman, A., & Rubin, D. B.. (1992). Inference from Iterative Simulation Using Multiple Sequences. Statistical Science, 7(4), 457–472.