# Tractable Marginal Inference for Hinge-Loss Markov Random Fields

**Varun R Embar** [*1] **Sriram Srinivasan** [*1] **Lise Getoor** [1]

## Abstract

Hinge-loss Markov random fields (HL-MRFs) are a class of undirected graphical models that has been successfully applied to model richly structured data. HL-MRFs are defined over a set of continuous random variables in the range [0,1], which makes computing the MAP convex. However, computation of marginal distributions remain intractable. In this paper, we introduce a novel sampling-based algorithm to compute marginal distributions. We define the notion of *association blocks*, which help identify islands of high probability, and propose a novel approach to sample from these regions. We validate our approach by estimating both average precision and various properties of a social network. We show that the proposed approach outperforms MAP estimates in both average precision and the accuracy of the properties by 20% and 40% respectively on the large social network.

## 1. Introduction

Hinge-loss Markov random fields(HL-MRFs) (Bach et al., 2017) are a class of undirected probabilistic graphical models used to model richly structured data. HL-MRFs have been successfully applied to areas such as information extraction (Liu et al., 2016), visual question answering (Aditya et al., 2018), recommender systems (Kouki et al., 2015), knowledge graphs (Pujara et al., 2013) and stance classification (Sridhar et al., 2015; Ebrahimi et al., 2016). Like Markov logic networks(MLNs) (Richardson & Domingos, 2006), HL-MRFs are defined using a set of weighted logical rules. However, unlike MLNs which are defined over a set of boolean random variable, HL-MRFs are defined over continuous random variables in the range $[0, 1]$. HL-MRFs also make use of hinge functions as potentials which enables MAP inference to be cast as a convex optimization problem. As a result, the MAP computation is extremely efficient and can infer the values of 2.5 million random variables in less than a minute(Augustine & Getoor, 2018). However, the computation of marginal distributions in HL-MRFs remain intractable.

Sampling approaches such as importance sampling and Markov Chain Monte Carlo (MCMC) are typically used to compute the marginal distributions empirically (Sudderth et al., 2010; Ihler & McAllester, 2009; Noorshams & Wainwright, 2013). In particular, Gibbs sampling (Gilks et al., 1995), an advanced class of MCMC, has been successfully used to compute marginal distribution for a wide range of probabilistic graphical models (Plummer et al., 2003) including Markov logic networks (Richardson & Domingos, 2006). Gibbs sampling involves iteratively sampling values for each random variable conditioned on all other random variables. However, using Gibbs sampling for HL-MRFs has two main challenges. First, unlike discrete MRFs, where the conditional distributions often follow a multinomial distribution and is easy to sample from, it is non-trivial to generate samples from the conditional distributions of HL-MRFs. The conditional distributions do not correspond to any standard named distributions. Second, Gibbs sampling has poor convergence rates when there are small islands of high probability. Identifying such high probability regions is challenging.

In this paper, we propose a tractable approach to compute marginal distributions for HL-MRFs. Our approach computes the marginal distributions empirically using the samples generated from a Gibbs sampler. To sample from the conditional distributions, we propose a *Metropolis* step within the Gibbs sampler (also called *Metropolis-within-Gibbs* (R. Gilks et al., 1995)) that replaces explicit sampling from the conditionals with a single step of the metropolis algorithm. To identify island of high probability, we define the notion of *association blocks*, and propose a technique that infers them for the logical rules used to define HL-MRFs. We propose a blocked Gibbs sampler that jointly samples random variables in an association block that ensures faster convergence. We perform experiments to show that our approach correctly identifies island of high probability. We further estimate several properties of a social network and show that the proposed approach outperforms MAP estimates by upto 40%.

[*]Equal contribution [1]UC Santa Cruz. Correspondence to: Varun R Embar <vembar@ucsc.edu>, Sriram Srinivasan <ssriniv9@ucsc.edu>.

*3rd Tractable Probabilistic Modeling Workshop, $36^{th}$ International Conference on Machine Learning*, Long Beach, California, 2019. 

## 2. Background

HL-MRFs are conditional distributions defined over a set of random variables in range $[0, 1]$. Given a set of unobserved random variables $Y = \{y_1, y_2, \cdots, y_n\}$, and a set of observed variables or evidence $X = \{x_1, x_2, \cdots, x_m\}$ a HL-MRF is defined as follows:

$$P(Y|X) = \frac{1}{Z(X,Y)} exp^{-E(Y,X)} \tag{1}$$

where $E(Y, X)$ is the energy function and $Z(X, Y)$ is the normalization constant given by $\int_Y exp^{-E(Y,X)}$.

The energy function $E(Y, X)$ is given by

$$E(Y, X) = \sum_{r=1}^{N} w_r \phi_r(X, Y) \tag{2}$$

The potential functions $\phi_r$ are defined using a probabilistic programming language called Probabilistic soft logic(PSL), and $N$ is the total number of potentials generated by PSL. PSL uses weighted first-order logic rules that is instantiated using the data to define hinge-loss potential functions $\phi_r$.

As an illustration, consider the following PSL rule which encodes that people who live together are friends:

$$\lambda : \text{LIVETOGETHER}(\mathtt{e}_i, \mathtt{e}_j) \implies \text{FRIENDS}(\mathtt{e}_i, \mathtt{e}_j)$$

Predicates such as LIVETOGETHER, whose truth values are observed, are called *closed predicates*. Predicates such as FRIENDS, whose truth value needs to be inferred, are called *open predicates*.

Consider a database that includes information about a large collection of people. For any two people `Alice`, `Bob` in the database, the rule is instantiated or grounded to generate the following ground rule:

$$\lambda : \text{LIVETOGETHER}(\mathtt{Alice}, \mathtt{Bob}) \implies \text{FRIENDS}(\mathtt{Alice}, \mathtt{Bob})$$

For the above ground rule, PSL generates a potential function by computing the distance to satisfaction using *Lukasiewicz* norm and co-norm. The potential function is given by:

$$\phi(x, y) = max\{x - y, 0\}^p$$

where $x$ is the observed random variable associated with $\text{LIVETOGETHER}(\mathtt{Alice}, \mathtt{Bob})$ and $y$ is the unobserved random variable associated with $\text{FRIENDS}(\mathtt{Alice}, \mathtt{Bob})$ and $p \in \{1, 2\}$. The set of potential functions generated from all the ground rules along with the rule weights is used to compute the energy function as given in (2).

## 3. Gibbs Sampling for HL-MRFs

Gibbs sampling is a type of sampling approach based on Markov chains (Neal, 1993), and generates samples from the joint distribution by iteratively sampling from the conditional distribution of each random variable keeping the remaining random variables fixed. It is a preferred choice for many multivariate distributions as it does not require tuning of parameters (Casella & George, 1992). The samples generated from the stationary distribution of the Gibbs sampling scheme is guaranteed to be from the joint distribution (Gilks et al., 1995).

A Gibbs sampler assumes that it is easy to generate samples from the conditional distribution. The conditional distribution for a random variable $y_i$ conditioned on all other variables $X$, $Y_{-i}$ in an HL-MRF is given by:

$$p(y_i|X, Y_{-i}) \propto exp\{-\sum_{r=1}^{N_i} w_r \phi_r(y_i, X, Y_{-i})\} \tag{3}$$

where $N_i$ is the number of groundings that variable $y_i$ participates in. The above distribution neither correspond to any standard named distribution nor has a form amenable to techniques such as inversion sampling. Therefore, it is non-trivial to sample from the above conditional.

We overcome this challenge by using a Metropolis step to sample from the conditional instead of direct sampling. For each random variable $y_i$, we first sample a new value $y_i'$ from a proposal distribution $g(y_i)$ and compute the acceptance ratio $\alpha$ given by:

$$\alpha = \frac{exp\{-\sum_{r=1}^{N_i} w_r \phi_r(y_i', X, Y_{-i})\}}{exp\{-\sum_{r=1}^{N_i} w_r \phi_r(y_i, X, Y_{-i})\}} \tag{4}$$

We then accept the new value $y_i'$ with a probability proportional to the acceptance ratio $\alpha$. Using the right proposal distribution $g$ is critical to ensure that the stationary distribution corresponds to the correct distribution. Since the random variables represents soft truth values, we use the uniform distribution in the range $[0, 1]$ as the proposal distribution.

The Markov chain of the Gibbs sampler requires several iterations before converging to the stationary distribution and is known as *burn-in* time. Therefore, it is essential to ignore these initial iterations (in our experiments we discard the first 1000 samples). The burn-in time is sensitive to the initial point. The time taken to converge from an arbitrary initial state can be large. Since we can efficiently estimate the MAP state for HL-MRFs, we initialize the sampler with the MAP values. This ensures that the chain starts from a region of high density and can converge quickly.

The algorithm for generating samples is shown in Algorithm 1. We initialize the random variables to the MAP state. We

**Algorithm 1** Metropolis-in-Gibbs sampler for PSL

**Input:** Set of $N$ ground rules, # of iterations T, burn-in period $b$
**Output:** Set of samples $\mathcal{S}$
# Initialize $Y^{(0)}$ to MAP state
$Y^{(0)} \leftarrow argmax_Y p(Y|X)$
# Sample values for each $y_i$
**for** $t$ from 1 to $T$ **do**
  **for** $y_i^{(t)} \in Y^{(t)}$ **do**
    $y_i' \sim U[0,1]$
    $\alpha = \frac{exp\{-\sum_{r=1}^{N_i} w_r \phi_r(y_i', X, Y_{1:i-1}^{(t+1)}, Y_{i:n}^{(t)})\}}{exp\{-\sum_{r=1}^{N_i} w_r \phi_r(y_i, X, Y_{1:i-1}^{(t+1)}, Y_{i:n}^{(t)})\}}$
    $u \sim U[0,1]$
    **if** $u < \alpha$ **then**
      $y_i^{(t+1)} = y_i'$
    **else**
      $y_i^{(t+1)} = y_i^{(t)}$
    **end if**
  **end for**
  # Consider samples after burn-in period $b$
  **if** $t > b$ **then**
    $\mathcal{S} = \mathcal{S} \cup Y^{(t)}$
  **end if**
**end for**
Return $\mathcal{S}$

then iteratively sample new values for each random variable after burn-in period $b$. We empirical estimate the marginals using these samples.

## 4. Association blocks

Another challenge is the slow convergence of the Gibbs sampler when the stationary distribution has low dispersion. To illustrate this in the case of a HL-MRF generated by a PSL program, consider the following rule that encodes symmetry between two random variables:

$$\lambda : \textsc{Friends}(\mathtt{e}_i, \mathtt{e}_j) \implies \textsc{Friends}(\mathtt{e}_j, \mathtt{e}_i)$$

Consider two people `Alice` and `Bob`, present in the data resulting in the following two ground rules:

$$\lambda : \textsc{Friends}(\text{Alice, Bob}) \implies \textsc{Friends}(\text{Bob, Alice})$$
$$\lambda : \textsc{Friends}(\text{Bob, Alice}) \implies \textsc{Friends}(\text{Alice, Bob})$$

Fig. 1 shows the normalized negative energy function for the two random variables corresponding to the ground atoms FRIENDS(Alice, Bob) and FRIENDS(Bob, Alice). We observe that as the rule weight $\lambda$ increases from 1 to 100, the region of high probability gets concentrated along the $x = y$ line. Once the Markov chain reaches a state where FRIENDS(Alice, Bob) = FRIENDS(Bob, Alice), it cannot transition to a another state unless both FRIENDS(Alice, Bob) and FRIENDS(Bob, Alice) update their values together. We refer these sets of variables that jointly induce regions of high probability as *association blocks*. It is important to identify association blocks and jointly sample them from a suitable proposal distribution for faster convergence of Markov chains.

In the following subsection, we first discuss our approach to identify association blocks from the set of weighted logical rules. We restrict ourselves to rules with at most two open predicates. The class of HL-MRFs generated by such rules correspond to pairwise-MRFs. We then propose a novel sampling approach for variables in the association cluster(proposal distribution $g$). Our approach generates samples from regions of high probability, resulting in faster convergence rates for the sampler. We refer to this approach as **ABGibbs**.

### 4.1. Identifying association blocks

As shown in the previous section, high weighted PSL rules result is distributions with low dispersion. The majority of the probability mass is confined to regions where the sum or difference between two random variables lie in a small interval. We first propose an approach to identify such random variable pairs and extend these pairwise associations to identify association blocks.

**Theorem 1.** *A PSL ground rule with two open predicates result is a potential function that can be minimized by one of the following four conditions:*

$$y_i - y_j \leq c$$
$$y_i - y_j \geq c$$
$$y_i + y_j \leq c$$
$$y_i + y_j \geq c$$

*where $y_i$ and $y_j$ are random variables corresponding to open predicates and $c \in \mathbb{R}$.*

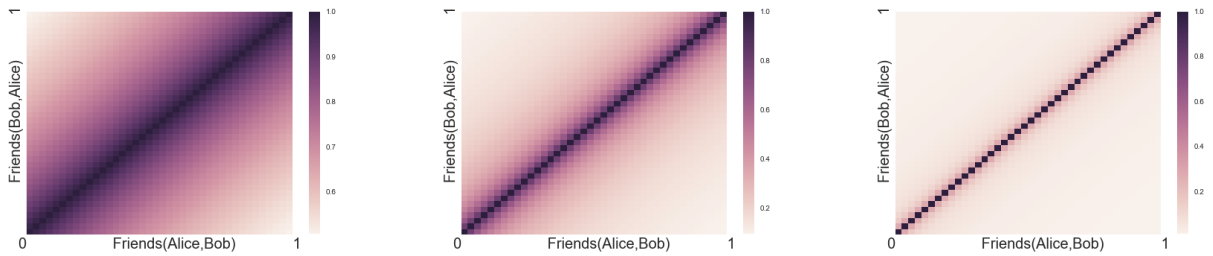
*Proof.* Any logical PSL rule can be written in a disjunctive normal form (DNF) and has one the following forms:

$\mathtt{r}_s(\mathtt{e}_i, \cdots) \vee \mathtt{r}_t(\mathtt{e}_j, \cdots) \vee_u \mathtt{r}_u(\mathtt{e}_k, \cdots) \vee_v \neg \mathtt{r}_v(\mathtt{e}_l, \cdots)$
$\neg \mathtt{r}_s(\mathtt{e}_i, \cdots) \vee \mathtt{r}_t(\mathtt{e}_j, \cdots) \vee_u \mathtt{r}_u(\mathtt{e}_k, \cdots) \vee_v \neg \mathtt{r}_v(\mathtt{e}_l, \cdots)$
$\mathtt{r}_s(\mathtt{e}_i, \cdots) \vee \neg \mathtt{r}_t(\mathtt{e}_j, \cdots) \vee_u \mathtt{r}_u(\mathtt{e}_k, \cdots) \vee_v \neg \mathtt{r}_v(\mathtt{e}_l, \cdots)$
$\neg \mathtt{r}_s(\mathtt{e}_i, \cdots) \vee \neg \mathtt{r}_t(\mathtt{e}_j, \cdots) \vee_u \mathtt{r}_u(\mathtt{e}_k, \cdots) \vee_v \neg \mathtt{r}_v(\mathtt{e}_l, \cdots)$

where $\mathtt{r}_s$ and $\mathtt{r}_t$ are open predicates, $\mathtt{r}_u$ and $\mathtt{r}_v$ refers to the set of non-negated and negated closed predicates.

The potential functions $\phi$ for each these rules are as follows:

$$max\{1 - y_i - y_j - c_1, 0\}^p$$
$$max\{y_i - y_j - c_2, 0\}^p$$
$$max\{y_i + y_j - 1 - c_3, 0\}^p$$
$$max\{y_j - y_i - c_4, 0\}^p$$




(a) Weight = 1 (b) Weight = 10 (c) Weight = 100

Figure 1: Negative energy function normalized between zero and one for two random variables grounded by the symmetric rule. As the weight of rule increases, region of high probability gets concentrated along the $x = y$ line.

where $y_i$ and $y_j$ are random variables corresponding to $\mathtt{r}_s$ and $\mathtt{r}_t$. $c_1, c_2, c_3, c_4 \in \mathbb{R}$ and their values depend on the closed predicates $\mathtt{r}_u$ and $\mathtt{r}_v$.

Each of these potential functions attain a minimum value 0 when the following conditions are satisfied for $c \in \mathbb{R}$:

$$\begin{aligned} y_i + y_j &\geq c \\ y_i - y_j &\leq c \\ y_i - y_j &\geq c \\ y_i + y_j &\leq c \end{aligned}$$

PSL also supports arithmetic rules and has the following form:

$$\begin{aligned} \mathtt{r}_s(\mathtt{e}_i, \cdots) + \mathtt{r}_t(\mathtt{e}_j, \cdots) \pm_u \mathtt{r}_u(\mathtt{e}_k, \cdots) = c \\ \mathtt{r}_s(\mathtt{e}_i, \cdots) - \mathtt{r}_t(\mathtt{e}_j, \cdots) \pm_u \mathtt{r}_u(\mathtt{e}_k, \cdots) = c \end{aligned}$$

These rules result in potentials that attain a minimum value of 0 when following conditions are satisfied:

$$\begin{aligned} c_1 \leq y_i - y_j \leq c_2 \\ c_3 \leq y_i + y_j \leq c_4 \end{aligned}$$

□

**Theorem 2.** *A PSL rule with a single open predicate results in a potential function that can be minimized by the condition $y_i \leq c$, where $y_i$ is the random variables corresponding to the open predicate and $c \in \mathbb{R}$.*

*Proof.* The proof is similar to the above proof. □

Presence of two high weighted ground rules such that they are both minimized only when the sum or difference between $y_i, y_j$ lies in a small region results in distributions with small dispersion. For example, consider the following two rules:

$$\begin{aligned} y_i - y_j &\geq c_1 \\ y_i - y_j &\leq c_2 \end{aligned}$$

The rules together ensure that most of the probability mass is concentrated in region where the difference between $y_i$ and $y_j$ is in the range $[c_1, c_2]$ (Positive association). Similarity the following two rules concentrate the probability mass to region where the sum of $y_i$ and $y_j$ to the range $[c_1, c_2]$ (Negative association):

$$\begin{aligned} y_i + y_j &\geq c_1 \\ y_i + y_j &\leq c_2 \end{aligned}$$

Our approach to identify association blocks is shown in Algorithm 2. We first identify pairwise association from ground rules with weights greater that a threshold $\lambda_t$. We also keep track of the region where these potential functions are minimized. We then identify pairwise associations when the region of minimum potential functions is below a threshold $\theta$. Finally, we merge these pairs into blocks such that all associated pairs lie in the same block.

### 4.2. Sampling approach for association blocks (Metropolis step)

Presence of high weighted rules result in most of the probability mass being accumulated in regions where all rules are satisfied. If this region is small, independently sampling each random variable in a block leads to high reject rate as a large number of samples are outside this region. We propose a novel sampling approach (as the proposal distribution for Metropolis step) for the association blocks that ensures most of the samples lie in the region of high probability mass. This is essential to ensure fast convergence. The proposed sampling approach is given in Algorithm 3. We first randomly chose a variable $y_i$ in the association block, and sample a value from $U[0, 1]$. We then update the bounds for all variables, that contain $y_i$ as a part of pairwise association, based on the sampled value for $y_i$. We randomly chose a variable $y_j$, which is bounded, and sample a value from the bounded range with probability $\beta$ and sample a value in the range $U[0, 1]$ with probability $1 - \beta$. We again update

**Algorithm 2** Identifying blocks of associated random variables

**Input:** Set of $N$ ground rules $G$, weight threshold $\lambda_t$, range threshold $\theta$
**Output:** Blocks of associated random variables $\mathcal{C}$
**Initialize:** Hashmaps $R^+$ and $R^-$ that hold additive and subtraction bounds
**for** $r \in 1\ to\ N$ **do**
  # For rules with high weights
  **if** $\lambda_r > \lambda_t$ **then**
    # Update the bounds
    **if** $r$ is of the form $a - b \leq c$ **then**
      $R^-(a,b).max = \min\{R^-(a,b).max, c\}$
    **else if** $r$ is of the form $a - b \geq c$ **then**
      $R^-(a,b).min = \max\{R^-(a,b).min, c\}$
    **else if** $r$ is of the form $a + b \leq c$ **then**
      $R^+(a,b).max = \min\{R^+(a,b).max, c\}$
    **else if** $r$ is of the form $a + b \geq c$ **then**
      $R^+(a,b).min = \max\{R^+(a,b).min, c\}$
    **end if**
  **end if**
**end for**
# Identify clusters from pairwise associations
**for** $(a,b) \in R^+ \bigcup R^-$ **do**
  **if** $R^+(a,b).max - R^+(a,b).min \leq \theta$ or $R^-(a,b).max - R^-(a,b).min \leq \theta$ **then**
    Merge blocks containing a,b and update $\mathcal{C}$
  **end if**
**end for**
Return set of blocks $\mathcal{C}$

the bounds for all variables that contain $y_j$. This process is performed iteratively for all the variables in block.

## 5. Experimental Evaluation

In this section, we validate our approach by performing empirical evaluation on a synthetic social network where the nodes represent people and edges represent friendship links. We consider the problem of node classification where the task is to infer the political affiliation of each person. We infer the political affiliations using a HL-MRF generated by a PSL model. We evaluate the performance of the inferred affiliations using average precision and the Precision-Recall curve (PR-curve). We further use the inferred affiliations to estimate two properties of the network. We compare five different strategies: MAP estimate ($\mathbf{MAP}$) that corresponds to set of values with the highest probability density, mean of the samples generated by the $\mathbf{Gibbs}$ sampler and the $\mathbf{ABGibbs}$ sampler ($\mathbf{Gibbs_{Mean}}$ and $\mathbf{ABGibbs_{Mean}}$) and expected value under the samples generated by the two samplers ($\mathbf{Gibbs_{Exp}}$ and $\mathbf{ABGibbs_{Exp}}$).

**Algorithm 3** Sampling scheme for variables in a block

**Input:** A block of random variables $\mathbf{c}$, $R^+, R^-$
**Output:** Sample $\mathbf{s}$ for variables in $\mathbf{c}$
$\mathbf{s} = \emptyset$
Pick a variable $y_i$ from $\mathbf{c}$ at random
$y_i \sim U[0,1]$
$\mathbf{s}.add(a)$
**while** $y_j \in \mathbf{c} \setminus \mathbf{s}$ and associated to some variable in $\mathbf{s}$ **do**
  Update range $[u,v]$ for $y_j$ based on $A$, $R^+$, and $R^-$
  $b \sim [0,1]$
  **if** $b \leq \beta$ **then**
    $y_j \sim U[u,v]$
  **else**
    $y_j \sim U[0,1]$
  **end if**
  $\mathbf{s}.add(y_j)$
**end while**
Return $\mathbf{s}$

### 5.1. Experimental setup

**Data:** We generate a social network graph using the NETRATE tool (Gomez-Rodriguez et al., 2011). We then generate cascades using the independent diffusion model and label nodes in the cascade as belonging to party $A$ with probability $0.7$. We stop this process when 50% of the nodes are labeled party $A$. We label remaining nodes in the graph as belonging to party $B$. This ensures that adjacent nodes are more likely to belong to the same party. We sub sampled the nodes in the graphs to generate three versions of the graph: small(S), medium(M) and large(L). The statistics of the graphs are given in table Table 2.

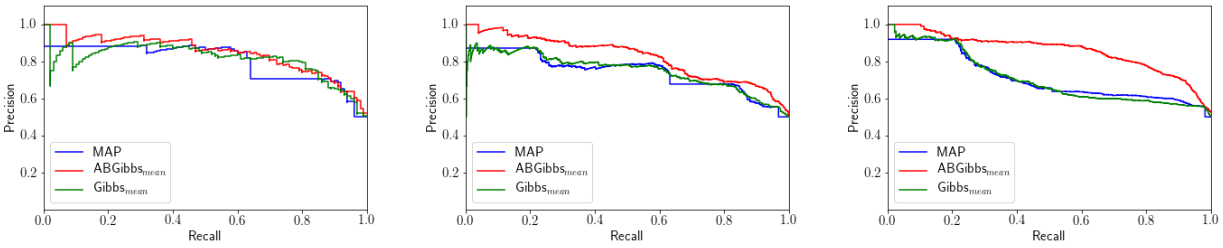
To help infer the party affiliations we generated two signals: STRONG and WEAK. We first randomly sampled 50% of the nodes in the graph. For each sampled node $\mathtt{e}_i$, we sample values for the ground atoms $\textsc{Strong}(\mathtt{e}_i, Party\ A)$ and $\textsc{Strong}(\mathtt{e}_i, Party\ B)$, from a Bernoulli distribution with parameters $0.9$ for the true label, and with parameter $0.1$ for the other party. We similarly generated a weak signal, WEAK, by sampling values from a Bernoulli with parameters $0.65$ and $0.35$.

**PSL model:** The PSL model for inferring political affiliations given a graph is shown in below:

$$10 : \textsc{Strong}(\mathtt{e}_i, \mathtt{e}_j) \implies \textsc{Party}(\mathtt{e}_i, \mathtt{e}_j)$$
$$5 : \textsc{Weak}(\mathtt{e}_i, \mathtt{e}_j) \implies \textsc{Party}(\mathtt{e}_i, \mathtt{e}_j)$$
$$5 : \textsc{Party}(\mathtt{e}_i, \mathtt{e}_k) \wedge \textsc{Friends}(\mathtt{e}_i, \mathtt{e}_j) \implies \textsc{Party}(\mathtt{e}_j, \mathtt{e}_k)$$
$$1000 : \textsc{Party}(\mathtt{e}_i, +\mathtt{e}_k) = 1$$
$$1 : \textsc{Party}(\mathtt{e}_i, \mathtt{e}_j) = 0.5$$

| Methods | Small | | | Medium | | | Large | | |
|---|---|---|---|---|---|---|---|---|---|
| | Avg Prec | P1 | P2 | Avg Prec | P1 | P2 | Avg Prec | P1 | P2 |
| **MAP** | 0.801 | 7 | 3 | 0.744 | 35 | 19 | 0.710 | 130 | 121 |
| **Gibbs$_{\text{Mean}}$** | 0.813 | 22 | 9 | 0.750 | 69 | 31 | 0.707 | 1001 | 339 |
| **Gibbs$_{\text{Exp}}$** | 0.817 | 19 | 8 | 0.751 | 70 | 29 | 0.702 | 1004 | 354 |
| **ABGibbs$_{\text{Mean}}$** | **0.847** | 18 | 8 | **0.823** | 98 | 38 | **0.859** | 280 | 105 |
| **ABGibbs$_{\text{Exp}}$** | 0.782 | **27** | **12** | 0.750 | **113** | **47** | 0.777 | **484** | **172** |
| **Ground Truth** | N/A | 42 | 12 | N/A | 258 | 87 | N/A | 595 | 187 |

Table 1: **Metrics:** Performance of different approaches across graphs of varying sizes. We observe that **ABGibbs$_{\text{Mean}}$** has higher average precision and **ABGibbs$_{\text{Exp}}$** estimates the properties accurately.




(a) Small graph (b) Medium graph (c) Large graph

Figure 2: **PR Curve:** Precision-Recall (PR) curve for **MAP**, **Gibbs$_{\text{Mean}}$** and **ABGibbs$_{\text{Mean}}$** on graphs of varying sizes. We observe that **ABGibbs$_{\text{Mean}}$** outperforms other methods.

| Size | Nodes | Edges |
|---|---|---|
| Small | 100 | 109 |
| Medium | 500 | 832 |
| Large | 1000 | 2026 |

Table 2: Data stats: The number of nodes and edges for three graphs.

The open predicate PARTY encodes the party affiliations and FRIENDS encodes the edges in the graph. The first two rules use the strong and the weak signals to infer party affiliations. The third rule propagates the affiliations over the friendship links in the graph. The fourth rule ensures that the party affiliations for a person sums to one. Finally, the last rule assigns a prior value for 0.5 for all affiliations.

**Parameters:** In our experiments, for identifying associated blocks, we set $\lambda_t = 100$ and $\theta = 0.1$. For sampling variables in the associated block, we set $\beta = 0.001$. For **ABGibbs** and **Gibbs**, we generate 10000 samples and discard the first 1000 samples as burn-in time $b$.

### 5.2. Inferred party affiliations

In this subsection, we evaluate the inferred party affiliations using average precision and PR curve.

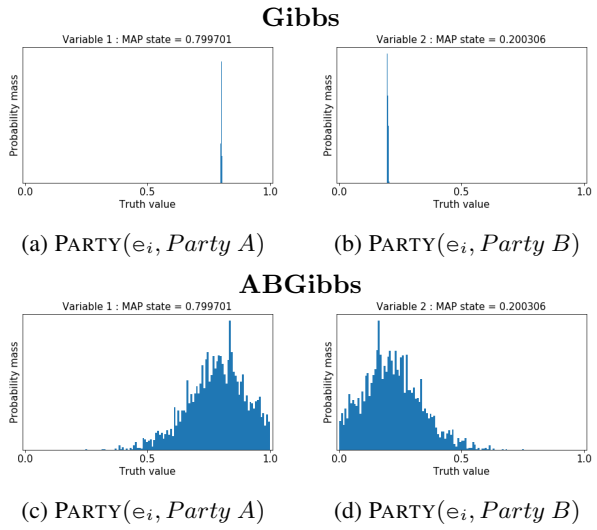
**Precision:** Table 1 shows the average precision for different approaches. We observe that **ABGibbs$_{\text{Mean}}$** has the highest average precision and the performance of **ABGibbs$_{\text{Exp}}$** is relatively poor. Many random variables do not have strong signal to infer their correct affiliations and hence marginal distributions have high dispersion. As a result, many samples have low precision. We also observe that the performance of **Gibbs$_{\text{Mean}}$** and **Gibbs$_{\text{Exp}}$** are similar to the performance of the **MAP**. This is because **Gibbs** starts at the MAP state and does not mix well due to the association between random variables. We also show the Precision-Recall(PR) curve for **MAP**, **Gibbs$_{\text{Mean}}$** and **ABGibbs$_{\text{Mean}}$** in Fig. 2. We observe that **ABGibbs$_{\text{Mean}}$** outperforms other approaches on all three networks.

The run times for various approaches are given in Table 3. Since **MAP** solves an optimization problem, it has the least run time. We also observe that **ABGibbs** takes slightly longer than **Gibbs** for each iteration. However, **ABGibbs** converges with fewer iterations when compared to **Gibbs**.

### 5.3. Estimating network properties

In this section, we evaluate two network properties, one associated with the nodes in the network and the other associated with the edges in the network.

**Property 1 (P1):** We estimate the number of pairs of people who are friends but are affiliated to different parties. Table 1




(a) PARTY($e_i$, *Party A*) (b) PARTY($e_i$, *Party B*)

(c) PARTY($e_i$, *Party A*) (d) PARTY($e_i$, *Party B*)

Figure 3: Inferred distributions of party affiliation for a person in the network by **Gibbs** and **ABGibbs**. **Gibbs** is unable to transition for the MAP state as the two variables belong to the same dependency cluster. However, **ABGibbs** is not contrasted to the initial state.

shows the estimated values by different approaches for all three networks. We observe that the **MAP** underestimates the value by a large margin. $\mathbf{ABGibbs_{Exp}}$ provides the best estimate of property value. This is because of the point-estimate nature of **MAP** that does not take the entire distribution into account.

**Property 2 (P2):** We estimate the number of people who have at least one friend belonging to each party. Table 1 shows the estimated values by different approaches for all three networks. We observe that the results obtained for this property is similar to that of P1. $\mathbf{ABGibbs_{Exp}}$ performs the better than all other methods.

The main reason **Gibbs** performs similar to **MAP** is due to the inability of some random variables to transition states. This is caused by the association between variables in the blocks. As **Gibbs** samples variables individually, the probability of acceptance of a sample in the Metropolis step drops severely for variables in an association block. Fig. 3 shows the inferred distributions of party affiliation for a person in the network. Here we observe that the distribution obtained using **Gibbs** has low dispersion. **ABGibbs**, on the other hand, recovers the true distribution. This is because, each person in the network generates two random variables, one for each party. We have a rule with weight 1000 which states that the sum of random variable generated by one person has to be equal to one. This couples the two random variables tightly and forms a association block. Therefore, conditioned on the first variable (variable 1) the possible values for the second variable (variable 2)

| Methods | Small | Medium | Large |
|---|---|---|---|
| **MAP** | 120 | 832 | 1242 |
| **Gibbs** | 132 | 1760 | 2897 |
| **ABGibbs** | 161 | 1697 | 3797 |

Table 3: Runtime for each approach in seconds. **ABGibbs** takes slightly longer than **Gibbs** for each iteration. However, as shown in Fig. 3 **ABGibbs** converges with fewer iterations when compared to **Gibbs**.

| Initialization | Random | MAP |
|---|---|---|
| **Gibbs** | 0.519 | 0.707 |
| **ABGibbs** | 0.859 | 0.859 |

Table 4: Avg precision for samplers with random and MAP initialization. We observe that **Gibbs** performs poorly for random initialization.

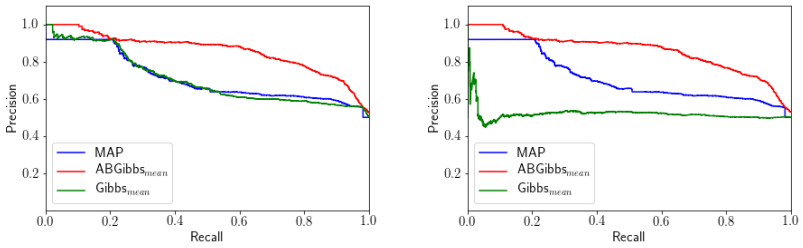
shrinks to a single point. This makes it almost impossible for the Metropolis step to accept any other value for the variable 2. As a result, **Gibbs** does not move and has poor mixing. **ABGibbs** does not suffer from this issue, as it jointly samples all variables in the association block using a proposal distribution described in Algorithm 3.

### 5.4. Impact of initialization

To analyze the impact of initialization, we run both the samplers starting from the MAP and a random state. The average precision for both the sampling methods is given in Table 4. We observe that while **ABGibbs** converges to the stationary distribution for both the initializations, **Gibbs** performs poorly for random initialization. This again due to the slow mixing time which results in the chain not moving far from the initial state. The PR curves for the **MAP**, $\mathbf{ABGibbs_{Mean}}$ and $\mathbf{ABGibbs_{Mean}}$ with MAP and random initialization is given in Fig. 4.

## 6. Conclusion

In this paper, we have presented a novel sampling approach to compute the marginal distributions of HL-MRFs. We also define the notion of association blocks, identify islands of high probability, and propose a sampling approach that samples from the regions of high probability. We estimate various properties of a social network using the marginals and show that the estimates are more accurate than other methods. While the initial results are promising, there are many further directions for research including computing confidence intervals, experimenting our approach to larger realworld datasets, and generalizing our approach to cases where the association blocks have no region where all rules are satisfied.




(a) MAP initialization (b) Random initialization

Figure 4: **PR Curve:** Precision-Recall (PR) curve for $\mathbf{MAP}$, $\mathbf{ABGibbs_{Mean}}$ and $\mathbf{ABGibbs_{Mean}}$ with MAP and random initialization on the large graph. We observe that $\mathbf{Gibbs_{Mean}}$ performs poorly when initialized with a random state.

## 7. Acknowledgments

This work was partially supported by the National Science Foundation grants CCF-1740850 and IIS-1703331 and by AFRL and the Defense Advanced Research Projects Agency.

## References

Aditya, S., Yang, Y., and Baral, C. Explicit reasoning over end-to-end neural architectures for visual question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Augustine, E. and Getoor, L. A comparison of bottom-up approaches to grounding for templated markov random fields. In *SysML*, 2018.

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. Hinge-Loss Markov Random Fields and Probabilistic Soft Logic. *Journal of Machine Learning Research*, 18: 109:1–109:67, 2017.

Casella, G. and George, E. I. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Ebrahimi, J., Dou, D., and Lowd, D. Weakly supervised tweet stance classification by relational bootstrapping. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1012–1017, 2016.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 561–568. Omnipress, 2011.

Ihler, A. and McAllester, D. Particle belief propagation. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 2009.

Kouki, P., Fakhraei, S., Foulds, J., Eirinaki, M., and Getoor, L. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 99–106. ACM, 2015.

Liu, S., Liu, K., He, S., and Zhao, J. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Neal, R. M. Probabilistic inference using markov chain monte carlo methods, 1993.

Noorshams, N. and Wainwright, M. J. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. *J. Mach. Learn. Res.*, 14: 2799–2835, 2013.

Plummer, M. et al. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, 2003.

Pujara, J., Miao, H., Getoor, L., and Cohen, W. Knowledge graph identification. In *International Semantic Web Conference*, 2013.

R. Gilks, W., Best, N., and K. C. Tan, K. Adaptive rejection metropolis sampling within gibbs sampling. *Applied Statistics*, 44:455–472, 1995.

Richardson, M. and Domingos, P. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

Sridhar, D., Foulds, J., Huang, B., Getoor, L., and Walker, M. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of*

*the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

Sudderth, E. B., Ihler, A. T., Isard, M., Freeman, W. T., and Willsky, A. S. Nonparametric belief propagation. *Commun. ACM*, 53:95–103, 2010.