A Framework for Incorporating General Domain Knowledge into Latent Dirichlet Allocation using First-Order Logic

David Andrzejewski
Lawrence Livermore National Laboratory
andrzejewski1@llnl.gov

Xiaojin Zhu
University of Wisconsin–Madison
jerryzhu@cs.wisc.edu

Mark Craven
University of Wisconsin–Madison
craven@biostat.wisc.edu

Benjamin Recht
University of Wisconsin–Madison
brecht@cs.wisc.edu

Abstract

Topic models have been used successfully for a variety of problems, often in the form of application-specific extensions of the basic Latent Dirichlet Allocation (LDA) model. Because deriving these new models in order to encode domain knowledge can be difficult and time-consuming, we propose the Fold·all model, which allows the user to specify general domain knowledge in First-Order Logic (FOL). However, combining topic modeling with FOL can result in inference problems beyond the capabilities of existing techniques. We have therefore developed a scalable inference technique using stochastic gradient descent which may also be useful to the Markov Logic Network (MLN) research community. Experiments demonstrate the expressive power of Fold·all, as well as the scalability of our proposed inference method.

1 Introduction

Building upon the success of Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003], a large number of latent-topic-model variants have been proposed for many application domains. Often, these variants are custom-built by incorporating external knowledge specific to the target domain, see e.g., [Wang *et al.*, 2009; Gerrish and Blei, 2010]. However, deriving a "custom" latent topic model, along with an efficient inference scheme, requires machine-learning expertise not common among application domain experts. Furthermore, such effort must be duplicated whenever a new type of domain knowledge is to be used, preventing domain experts from taking full advantage of topic modeling approaches. Previous work has integrated word-level knowledge into topic models for both batch [Andrzejewski *et al.*, 2009; Petterson *et al.*, 2010] and interactive [Hu *et al.*, 2011] settings, but these approaches do not incorporate constraints involving documents, topics, or general side information.

The main contribution of this paper is Fold·all (First-Order Logic latent Dirichlet ALLocation), a framework for incorporating general domain knowledge into LDA. A domain expert only needs to specify her domain knowledge as First-Order Logic (FOL) rules, and Fold·all will automatically incorporate them into LDA inference to produce topics shaped by both the data and the rules. This approach enables domain experts to focus on high-level modeling goals instead of the low-level issues involved in creating a custom topic model. In fact, some previous topic model variants can be expressed within the Fold·all framework.

Internally, Fold·all converts the FOL rules into a Markov Random Field, and combines it with the LDA probabilistic model. As such, it can be viewed as an instance of a Hybrid Markov Logic Network (HMLN) [Wang and Domingos, 2008], which is itself a generalization of a Markov Logic Network (MLN) [Richardson and Domingos, 2006]. However, existing inference schemes developed for HMLNs and MLNs do not scale well for typical topic modeling applications. Another contribution of this paper is a scalable stochastic optimization algorithm for Fold·all, which is potentially useful for general MLN research, too.

2 The Fold·all Framework

We now briefly review the standard LDA model [Blei *et al.*, 2003]. While we describe variables in terms of text (e.g., words and documents), note that both LDA and Fold·all are general and can be applied to non-text data as well. Let $\mathbf{w} = w_1 \dots w_N$ be a text corpus containing N tokens, with $\mathbf{d} = d_1 \dots d_N$ being the document indices of each word token and $\mathbf{z} = z_1 \dots z_N$ being the hidden topic assignments of each token. Each topic $t = 1 \dots T$ is represented by a multinomial ϕ_t over a W-word-type vocabulary. The ϕ's have a Dirichlet prior with parameter β. Likewise, each document $j = 1 \dots D$ is associated with a multinomial θ_j over topics, with another Dirichlet prior with parameter α. The generative model is $P(\mathbf{w}, \mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{d}) \propto$

$$\left(\prod_t^T p(\phi_t|\beta)\right)\left(\prod_j^D p(\theta_j|\alpha)\right)\left(\prod_i^N \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right) \quad (1)$$

where $\phi_{z_i}(w_i)$ is the w_i-th element in vector ϕ_{z_i}, and $\theta_{d_i}(z_i)$ is the z_i-th element in vector θ_{d_i}. One important goal of topic modeling is to estimate the topics ϕ given a corpus $(\mathbf{w}, \mathbf{d})$.

The key to our Fold·all framework is to allow domain knowledge, specified in FOL, to influence the values of the hidden topics $\mathbf{z}$, indirectly influencing ϕ and θ. FOL provides a powerful and flexible way to specify domain knowledge. For example, an analyst working on a congressional debate

corpus where each speech is a document may specify the rule

$$\forall i : \texttt{W}(i, \text{taxes}) \wedge \texttt{Speaker}(d_i, \text{Rep}) \Rightarrow \texttt{Z}(i, 77), \quad (2)$$

which states that for any word token $w_i =$ "taxes" that appears in a speech by a Republican, the corresponding latent topic should be $z_i = 77$. We briefly review some FOL concepts [Domingos and Lowd, 2009] for Fold·all.

We define logical predicates for each of the standard LDA variables, letting $\texttt{Z}(i, t)$ be *true* if the hidden topic $z_i = t$, and *false* otherwise. Likewise, $\texttt{W}(i, v)$ and $\texttt{D}(i, j)$ are *true* if $w_i = v$ and $d_i = j$, respectively. In addition, Fold·all can incorporate other variables beyond those modeled by standard LDA. In our previous example, a domain expert defines a predicate $\texttt{Speaker}(d_i, \text{Rep})$, which is *true* if the speaker for document d_i is a member of the Republican political party. We use $\mathbf{o}$ to collectively denote these other observed variables and their corresponding logical predicate values.

The domain expert specifies her background knowledge in the form of a *weighted FOL knowledge base* using these predicates: KB $= \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$. The KB is in Conjunctive Normal Form, consisting of L pairs where each rule ψ_l is an FOL clause, and $\lambda_l \geq 0$ is its weight which the domain expert sets to represent the importance of ψ_l. Thus, in general Fold·all treats such rules as soft preferences rather than hard constraints. The knowledge base KB is tied to our probabilistic model via its *groundings*. For each FOL rule ψ_l, let $G(\psi_l)$ be the set of groundings, each mapping the free variables in ψ_l to specific values. For the "taxes" example above, G consists of all N propositional rules where $i = 1 \ldots N$. For each grounding $g \in G(\psi_l)$, we define an indicator function

$$\mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) = \begin{cases} 1, & \text{if } g \text{ is } \textit{true} \text{ under } (\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) \\ 0, & \text{otherwise.} \end{cases}$$

For example, if $w_{100} =$ "taxes", $\texttt{Speaker}(d_{100}, \text{Rep}) =$ *true*, and $z_{100} = 88$, then the grounding $g = (\texttt{W}(100, \text{taxes}) \wedge \texttt{Speaker}(d_{100}, \text{Rep}) \Rightarrow \texttt{Z}(100, 77))$ will have $\mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) = 0$ because of the mismatch in z_{100}.

To combine the KB and LDA, we define a Markov Random Field over latent topic assignments $\mathbf{z}$, topic-word multinomials ϕ, and document-topic multinomials θ, treating words $\mathbf{w}$, documents $\mathbf{d}$, and side information $\mathbf{o}$ as observed. Specifically, in this Markov Random Field the conditional probability $P(\mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{w}, \mathbf{d}, \mathbf{o}, \text{KB})$ is proportional to

$$\exp\left(\sum_{l}^{L} \sum_{g \in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})\right) \times \quad (3)$$
$$\left(\prod_{t}^{T} p(\phi_t|\beta)\right)\left(\prod_{j}^{D} p(\theta_j|\alpha)\right)\left(\prod_{i}^{N} \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right).$$

This Markov Random Field has two parts: the first term acts as a prior from the KB, and the remaining terms are identical to LDA (1). Each satisfied grounding of FOL rule ψ_l contributes $\exp(\lambda_l)$ to the potential function. Note in general, the first term *couples* all the elements of $\mathbf{z}$, although the actual dependencies are determined by the particular form of the KB. The factor graph for the Fold·all Markov Random Field is shown in Figure 1, with a special "mega factor node" corresponding to the first term.

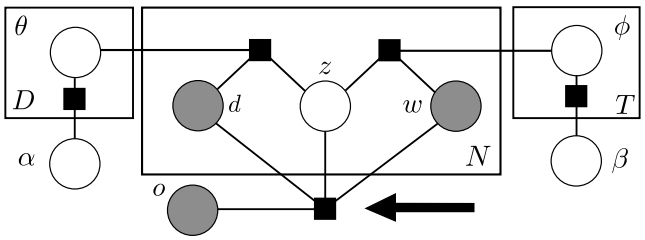

Figure 1: Fold·all factor graph with "mega" logic factor (indicated by arrow) connected to $\mathbf{d}$, $\mathbf{z}$, $\mathbf{w}$, $\mathbf{o}$.

The first term in (3) is equivalent to a Markov Logic Network (MLN) [Richardson and Domingos, 2006]. The remaining terms in (3) involve continuous variables such as θ, ϕ. This combination has been proposed in the MLN community under the name of Hybrid Markov Logic Networks (HMLN) [Wang and Domingos, 2008], but to our knowledge previous HMLN research has not combined logic with LDA.

3 Scalable Inference in Fold·all

Since exact inference is intractable for both LDA and MLN models, it is unsurprising that Fold·all inference is difficult as well. In fact, the combination of logic and topic modeling components presents a unique scalability challenge which cannot be addressed by existing techniques.

We are interested in inferring the most likely ϕ and θ in Fold·all. However, as in standard LDA, the latent topic assignments $\mathbf{z}$ cannot be marginalized out. We instead aim to find the Maximum a Posteriori (MAP) estimate of $(\mathbf{z}, \phi, \theta)$ *jointly*. This can be formulated as maximizing the logarithm of the unnormalized probability (3):

$$\underset{\mathbf{z},\phi,\theta}{\text{argmax}} \quad \sum_{l}^{L} \sum_{g \in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) + \sum_{t}^{T} \log p(\phi_t|\beta)$$
$$+ \sum_{j}^{D} \log p(\theta_j|\alpha) + \sum_{i}^{N} \log \phi_{z_i}(w_i)\theta_{d_i}(z_i). \quad (4)$$

This non-convex problem is particularly challenging due to the fact that the summations over groundings $G(\psi_l)$ are combinatorial: on a corpus with length N, an FOL rule with k universally quantified variables will produce N^k groundings. This explosion resulting from propositionalization is a well-known problem in the MLN community, and has been the subject of considerable research [Singla and Domingos, 2008; Kersting *et al.*, 2009; Riedel, 2008; Huynh and Mooney, 2009]. For instance, one can usually greatly reduce the problem size by considering only *non-trivial* groundings [Shavlik and Natarajan, 2009]. As an example, the rule in (2) is trivially true for all indices i such that $w_i \neq$ "taxes", and these indices can be excluded from computation. Unfortunately, even after this pre-processing, there may be an unacceptably large number of groundings. Furthermore, the inclusion of the LDA terms and the scale

Algorithm 1: Alternating Optimization with Mirror Descent for Fold·all.

Input: $\mathbf{w}, \mathbf{d}, \mathbf{o}, \alpha, \beta, \text{KB}$
for N_{outer} *iterations* **do**
  set ϕ, θ with (5) (6)
  set $\mathbf{z} \setminus \mathbf{z}_{\text{KB}}$ with (7)
  for N_{inner} *iterations* **do**
    sample term f from (9)
    update z_{it}'s in f with (10)
  end
  set $z_i \in \mathbf{z}_{\text{KB}}$ with $\arg\max_t z_{it}$
end
return $(\mathbf{z}, \phi, \theta)$

of our domain prevent us from directly taking advantage of many techniques developed for MLNs. In what follows, we describe a stochastic gradient descent algorithm, Alternating Optimization with Mirror Descent, to find a local maximum of (4). This approach may also be applied to standard MLNs, although we leave that application as future work.

3.1 Alternating Optimization with Mirror Descent

We propose Alternating Optimization with Mirror Descent (Mir) to optimize (4). The complete procedure is presented in Algorithm 1; it proceeds by alternating between optimizing the multinomial parameters (ϕ, θ) while holding $\mathbf{z}$ fixed, and vice versa. The optimal (ϕ, θ) for fixed $\mathbf{z}$ can be easily found in closed-form as the MAP estimate of the Dirichlet posterior:

$$\phi_t(w) \quad \propto \quad n_{tw} + \beta - 1 \tag{5}$$
$$\theta_j(t) \quad \propto \quad n_{jt} + \alpha - 1 \tag{6}$$

where n_{tw} is the number of times word w is assigned to topic t in topic assignments $\mathbf{z}$. Similarly, n_{jt} is the number of times topic t is assigned to a word in document j.

Optimizing $\mathbf{z}$ while holding (ϕ, θ) fixed is more difficult. One can divide $\mathbf{z}$ into an "easy part" and a "difficult part." The easy part consists of all z_i which only appear in trivial groundings, where a trivial grounding is defined as any grounding g such that the corresponding indicator function $\mathbb{1}_g$ is insensitive to the latent topic assignment $\mathbf{z}$. For example, if the knowledge base consists of only one rule $\psi_1 = (\forall i : \mathtt{W}(i, \text{apple}) \Rightarrow \mathtt{Z}(i, 1))$, then the majority of the z_i's (those with $w_i \neq$ apple) appear in groundings which are trivially true. These z_i's only appear in the last term in (4). Consequently, the optimizer is simply

$$z_i = \operatorname*{argmax}_{t=1\ldots T} \phi_t(w_i)\theta_{d_i}(t). \tag{7}$$

The difficult part of $\mathbf{z}$ consists of those z_i appearing in non-trivial groundings, subsequently in the first term of (4). Denote this part $\mathbf{z}_{\text{KB}}$. We use stochastic gradient descent to optimize $\mathbf{z}_{\text{KB}}$. The key idea is to first relax (4) into a continuous optimization problem, and then randomly sample groundings from the knowledge base, such that each sampled grounding provides a stochastic gradient to the relaxed problem.

Table 1: Step-by-step example of the logic polynomial procedure for the formula $g = \mathtt{Z}(i, 1) \vee \neg\mathtt{Z}(j, 2)$ with $T = 3$ (i.e., $t \in \{1, 2, 3\}$).

Original formula g	$\mathtt{Z}(i, 1) \vee \neg\mathtt{Z}(j, 2)$
1: Take complement $\neg g$	$\neg\mathtt{Z}(i, 1) \wedge \mathtt{Z}(j, 2)$
2: Remove negations $(\neg g)_+$	$(\mathtt{Z}(i, 2) \vee \mathtt{Z}(i, 3)) \wedge \mathtt{Z}(j, 2)$
3: Binary $z_{it} \in \{0, 1\}$	$(z_{i2} + z_{i3}) * z_{j2}$
4: Polynomial $\mathbb{1}_g(\mathbf{z})$	$1 - (z_{i2} + z_{i3}) * z_{j2}$
5: Relax discrete z_{it}	$z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$

Here we describe a procedure for converting the logic grounding indicator $\mathbb{1}_g$ into a *continuous* polynomial over relaxed z_{it} variables. Table 1 gives a simple step-by step example of this procedure; individual steps in the following text reference the corresponding steps in this example.

Step 1: Because we assume the knowledge base KB is in Conjunctive Normal Form, each non-trivial grounding g consists of a disjunction of $\mathtt{Z}(i, t)$ atoms (positive or negative), whose logical complement $\neg g$ is therefore a *conjunction* of $\mathtt{Z}(i, t)$ atoms (each negated from the original grounding g).

Step 2: In order to standardize each grounding, let $(\cdot)_+$ be an operator which returns a logical formula equivalent to its argument where we replace all negated atoms $\neg\mathtt{Z}(i, t)$ with equivalent disjunctions over positive atoms $\mathtt{Z}(i, 1) \vee \ldots \vee \mathtt{Z}(i, t-1) \vee \mathtt{Z}(i, t+1) \vee \ldots \vee \mathtt{Z}(i, T)$, and eliminate any duplicate atoms. We now have $(\neg g)_+$, which is the logical complement of our original formula g expressed entirely in terms of non-negated literals.

Step 3: We convert this Boolean formula over logical predicates to a polynomial over binary variables. To do this, we replace each $\mathtt{Z}(i, t)$ with a binary indicator variable $z_{it} \in \{0, 1\}$ defined to be equal to 1 if $\mathtt{Z}(i, t)$ is *true* and 0 otherwise. Each conjunction $\wedge$ is then replaced with multiplication $*$, and each disjunction $\vee$ is replaced with addition $+$. In this way, the conjunction of disjunctions $\neg g$ is converted into a product of sums over binary z_{it} variables.

Step 4: We now have a binary polynomial that is equivalent to $\neg g$, the negation of our original formula g. In order to remove this negation, we take the binary complement of this expression (i.e., $1 - x$ where x is the result of Step 3).

We now have a binary polynomial over z_{it} that is exactly equivalent to our original logical formula g. We can formally express this result as

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{i : g_i \neq \emptyset} \left(\sum_{\mathtt{Z}(i,t) \in (\neg g_i)_+} z_{it} \right) \tag{8}$$

where g_i is the *set* of atoms in g which involve index i. For example, if $g = \mathtt{Z}(0, 1) \vee \mathtt{Z}(0, 2) \vee \mathtt{Z}(1, 0)$, then $g_0 = \{\mathtt{Z}(0, 1), \mathtt{Z}(0, 2)\}$. Note the observed variables $\mathbf{w}, \mathbf{d}, \mathbf{o}$ are no longer in (8) because g is a non-trivial grounding where the disjunction of $\mathbf{w}, \mathbf{d}, \mathbf{o}$ atoms is always *false*.

Step 5: With our polynomial representation in hand, we *relax* the binary variables $z_{it} \in \{0, 1\}$ to continuous values $z_{it} \in [0, 1]$, with the constraint $\sum_t z_{it} = 1$ for all i. Under

this relaxation, Equation (8) takes on values in the interval $[0, 1]$, which can be interpreted as the expectation of the original Boolean indicator function under a distribution where each relaxed z_{it} represents the multinomial probability that $\mathtt{Z}(i, t)$ is *true*. We note that this function is non-convex due to bilinearity in z_{it}.

Dropping terms that are constant w.r.t. $\mathbf{z}_{\text{KB}}$ and reintroducing the LDA objective function terms yields the continuous optimization problem

$$\begin{aligned} \underset{\mathbf{z}\in[0,1]^{|\mathbf{z}_{\text{KB}}|}}{\text{argmax}} \quad & \sum_{l}^{L} \sum_{g\in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}) + \sum_{i,t} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) \\ \text{s.t.} \quad & z_{it} \geq 0, \quad \sum_{t}^{T} z_{it} = 1. \end{aligned} \tag{9}$$

This relaxation allows us to use gradient methods on (9). However a potentially huge number of groundings in $\cup_l G(\psi_l)$ may still render the full gradient impractical to compute. Critically, the next step is to use *stochastic gradient descent* for scalability, specifically the Entropic Mirror Descent Algorithm (EMDA) [Beck and Teboulle, 2003], of which the Exponentiated Gradient (EG) [Kivinen and Warmuth, 1997] algorithm is a special case. Unlike approaches [Collins *et al.*, 2008] which randomly sample *training examples* to produce a stochastic approximation to the gradient, we randomly sample *terms* in (9). A term f is either the polynomial $\mathbb{1}_g(\mathbf{z})$ on a particular grounding g, or an LDA term $\sum_t z_{it} \log \phi_t(w_i)\theta_{d_i}(t)$ for some index i. We use a weighted sampling scheme. Let Λ be a length $L + 1$ weight vector, where $\Lambda_l = \lambda_l |G(\psi_l)|$ for $l = 1 \dots L$, and the entry $\Lambda_{L+1} = |\mathbf{z}_{\text{KB}}|$ represents the LDA part. To sample individual terms, we first choose one of the $L + 1$ entries according to weights Λ. If an FOL rule ψ_l is chosen, we then sample a grounding $g \in G(\psi_l)$ uniformly. If the LDA part is chosen, we uniformly sample an index i from $\mathbf{z}_{\text{KB}}$. Once a term f is sampled, we take its gradient ∇f and perform a mirror descent update with step size η:

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}. \tag{10}$$

The process of sampling terms and taking gradient steps is then repeated for a prescribed number of iterations. Finally, we recover a hard $\mathbf{z}_{\text{KB}}$ assignment by rounding each z_i to $\arg\max_t \ z_{it}$. The key advantage of this approach is that it requires only a means to sample groundings g for each rule ψ_l, and can avoid fully grounding the FOL rules. We now consider several alternatives to the Mir approach.

3.2 MaxWalkSAT

A simple alternative means of introducing logic into LDA is to perform standard LDA inference and then post-process the latent topic vector $\mathbf{z}$ in order to maximize the weight of satisfied ground logic clauses in the KB (i.e., optimize the MLN objective in (4) only). This can be done using a weighted satisfiability solver such as MaxWalkSAT (MWS) [Selman *et al.*, 1995], a stochastic local search algorithm that selects an unsatisfied grounding and satisfies it by flipping the truth state of a single atom, repeating for N_{inner} iterations. Choosing which atom to flip is done either *randomly* (with probability p) or *greedily* w.r.t. the change Δ_{KB} to the *global* weighted satisfaction objective function. We keep the best (highest satisfied weight) assignment $\mathbf{z}$ found, although the fact that MWS does not take the learned topics into account means that this may actually *decrease* the full objective (4).

3.3 Alternating Optimization with MWS+LDA

A more principled approach is to integrate the logic and LDA objectives. In Algorithm 1, we replace the $\mathbf{z}_{\text{KB}}$ inner loop with MWS+LDA (M+L), a form of MWS modified to incorporate the LDA objective in the greedy selection criterion by selecting an atom according to $\Delta = \Delta_{\text{KB}} + \Delta_{\text{LDA}}$, where Δ_{LDA} is the change to the LDA objective. This aims to maximize the objective (4), balancing the gain from satisfying a logic clause and the gain of a topic assignment given the current ϕ and θ parameters. We initialize using standard LDA, and the inner MWS+LDA loop keeps the best $\mathbf{z}_{\text{KB}}$ found with respect to both satisfied logic weight *and* the LDA objective.

3.4 Collapsed Gibbs Sampling

We also perform collapsed Gibbs sampling (CGS) with respect to the full Fold·all distribution (3). While Gibbs sampling is not aimed at *maximizing* the objective, the hope is that the sampler will explore high probability regions of the $\mathbf{z}$ space. The collapsed Gibbs sampler iteratively resamples z_i at each corpus position i, with the probability of candidate topic assignment $z_i = t$ given by $P(z_i = t|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{d}, \mathbf{o}, \text{KB}, \alpha, \beta) \propto$

$$\left(\frac{n_{d_i t}^{(-i)} + \alpha_t}{\sum_{t'}^{T}(n_{d_i t'}^{(-i)} + \alpha_{t'})}\right)\left(\frac{n_{t w_i}^{(-i)} + \beta_{w_i}}{\sum_{w'}^{W}(n_{t w'}^{(-i)} + \beta_{w'})}\right) \times$$
$$\exp\left(\sum_{l} \sum_{g\in G(\psi_l): g_i \neq \emptyset} \lambda_l \mathbb{1}_g(\mathbf{z}_{-i} \cup \{z_i = t\})\right), \tag{11}$$

where each $-i$ indicates that we exclude the word token at position i. Note that (11) is the product of the standard LDA collapsed Gibbs sampler [Griffiths and Steyvers, 2004] and the MLN Gibbs sampling equation [Richardson and Domingos, 2006]. We keep the sample which maximizes (4). This approach may suffer from poor mixing in the presence of highly weighted logic rules [Poon and Domingos, 2006].

4 Experiments

Our experiments evaluate the generalization of the Fold·all model by measuring whether learned topics reflect both corpus statistics and the user-defined logic rules when applied to unseen documents and associated logic rule groundings. Simultaneously, we evaluate the *scalability* of inference methods by applying Fold·all to datasets and KBs with large numbers of non-trivial groundings. Our experiments demonstrate that: i) Fold·all successfully incorporates logic into topic modeling, and ii) Mir is a scalable and effective inference method for Fold·all that works when other methods fail.

We conduct experiments on several datasets and corresponding KBs using the four Fold·all inference methods developed in Section 3 (Mir, MWS, M+L, and CGS). We also

use two baseline methods which do *not* integrate topic modeling and logic: topic modeling alone (standard LDA inference with a collapsed Gibbs sampler), and logic alone (MAP inference with the Alchemy MLN software package [Kok *et al.*, 2009]). These baselines use existing techniques to model the LDA and MLN components in isolation.

For each KB, all free variables are universally quantified and we set logic rule weights λ to make the scale of the logic contribution comparable to the LDA contribution in the objective function (4). Table 2 shows details such as the λ's used and the number of non-trivial groundings $|\cup_l G(\psi_l)|$. We set (N_{outer}, N_{inner}) to $(10^2, 10^5)$ and run Gibbs samplers for 2,000 samples. The Mir inner loop (Algorithm 1) step size decays as $\eta_m = \sqrt{N_{inner}}/\sqrt{N_{inner}+m}$ for inner iteration m. We initialize all Fold·all algorithms with the final collapsed Gibbs sample from standard LDA, and fix Dirichlet parameters to $\alpha = 50/T$ and $\beta = 0.01$. We now present example datasets and KBs along with qualitative assessments.

Synthetic Cannot-Link (Synth): This small synthetic dataset demonstrates the ability of Fold·all to encode the Cannot-Link preference [Andrzejewski *et al.*, 2009], which states that occurrences of a pair of words should not be assigned to the same topic. We encode Cannot-Link (A, B) as $\mathtt{W}(i, \mathrm{A}) \wedge \mathtt{W}(j, \mathrm{B}) \Rightarrow \neg\mathtt{Z}(i,t) \vee \neg\mathtt{Z}(j,t)$ (the opposite Must-Link can be encoded similarly). Alchemy and Fold·all are able to enforce the KB, while standard LDA often does not.

Comp.* newsgroups (Comp): This dataset consists of online posts made to comp.* news groups from 20 newsgroups. We consider a user wishing to construct two separate topics around the concepts *hardware* and *software*. Our KB encourages the recovery of these topics using {hardware, machine, memory, cpu} and {software, program, version, shareware} as seed words in two rules: $\mathtt{W}(i, \text{hardware}) \vee \ldots \vee \mathtt{W}(i, \text{cpu}) \Rightarrow \mathtt{Z}(i, 0)$ and $\mathtt{W}(i, \text{software}) \vee \ldots \vee \mathtt{W}(i, \text{shareware}) \Rightarrow \mathtt{Z}(i, 1)$. The topics found by Fold·all inference methods align with our intended concepts: Topic 0 tends to consist of hardware-related terms: {drive, disk, ide, bus, install}, while new Topic 1 terms are software-oriented: {code, image, data, analysis}.

Congress (Con): This dataset consists of floor-debate transcripts from the United States House of Representatives [Thomas *et al.*, 2006]. Each speech is labeled with the political party of the speaker: $\mathtt{Speaker}(d, \mathrm{Rep})$ or $\mathtt{Speaker}(d, \mathrm{Dem})$. The predicate $\mathtt{HasWord}(d, w)$ is *true* if word w appears in document d. We consider an analyst wishing to identity interesting political topics using a KB containing a seed word rule putting {chairman, yield, madam} in Topic 0, as well as two rules exploiting political party labels:

$$\mathtt{Speaker}(d, \mathrm{Rep}) \wedge \mathtt{HasWord}(d, \text{taxes}) \wedge \mathtt{D}(i,d) \Rightarrow \mathtt{Z}(i,1) \vee \mathtt{Z}(i,2) \vee \mathtt{Z}(i,3)$$

$$\mathtt{Speaker}(d, \mathrm{Dem}) \wedge \mathtt{HasWord}(d, \text{workers}) \wedge \mathtt{D}(i,d) \Rightarrow \mathtt{Z}(i,4) \vee \mathtt{Z}(i,5) \vee \mathtt{Z}(i,6).$$

The first rule pulls uninteresting procedural words (e.g., "Mr. Chairman, I want to thank the gentlewoman for yielding...") into their own Topic 0. The other rules aim to discover interesting political topics associated with Rep on *taxes* and Dem on *workers*. As intended, Topic 0 pulls in other procedural words ({gentleman, thank, colleague}), improving the quality of the other topics. The special Rep *taxes* topics uncover interesting themes ({budget, billion, deficit, health, education, security, jobs, economy, growth}), as do the Dem *workers* topics ({pension, benefits, security, osha, safety, prices, gas}). This KB demonstrates how Fold·all can exploit side information to influence topic discovery.

Figure 2: Fold·all *movie* / *film* topics.

Polarity (Pol): This dataset consists of positive and negative movie reviews [Pang and Lee, 2004]. We posit an expert analyst who wishes to study the difference between usage of the word "movie" versus the word "film" in these reviews, placing Cannot-Link rule between those two words. The size of the groundings is too large for all logic-based methods except Mir, which is able to discover topics obeying the KB which reveal subtle sentiment differences associated with the two words (e.g., the *movie* topic contains "bad", while the *film* topic contains "great"). Figure 2 shows word clouds[1] for a pair of Mir topics containing "film" or "movie" only.

Human Development Genes (HDG): This dataset consists of PubMed abstracts; the goal is to learn topics for six concepts formulated by an actual biologist interested in human development. The expert provided seed words for each concept, which were translated into FOL rules. For example, the Topic 2 seed words are {hematopoietic, blood, endothelium}. However, using seed rules alone yields concept topics which stray from basic biology, and are polluted by more clinical terms such as {pressure, hypertension} and {leukemia, acute, myeloid}. We therefore seed an additional *disease* Topic 7 and enforce that this topic not co-occur in the same sentence as our original development concept topic. Let $\mathbf{s} = s_1, \ldots, s_N$ be a vector of sentence indices analogous to $\mathbf{d}$, with logical predicate $\mathtt{S}(i, s)$ being *true* if $s_i = s$. Our *exclusion* rule for the concept Topic 2 is

$$\mathtt{S}(i,s) \wedge \mathtt{S}(j,s) \wedge \mathtt{Z}(i,7) \Rightarrow \neg(\mathtt{Z}(j,1) \vee \ldots \vee \mathtt{Z}(j,6)).$$

In order to further encourage the recovery of development-oriented topics, we also define a *development* Topic 8 with seed words {differentiation, ..., develops}, and define an *inclusion* rule enforcing that our concept topics not be used

[1] http://www.wordle.net

Table 2: Fold·all generalization experiments, showing dataset and KB details along with objective function (4) values (with magnitudes in parentheses) averaged over test folds. All bolded values are significantly different from all non-bolded values in the same row at $p < 10^{-6}$ under Tukey's Honestly Significant Difference (HSD) test. Failing runs are indicated with "−".

	Fold·all				Baselines		Dataset+KB details			
	Mir	M+L	CGS	MWS	LDA	Alchemy	D	T	λ	$\lvert \cup_l G(\psi_l) \rvert$
Synth ($\times 10^1$)	**9.86**	**11.13**	**8.33**	**11.13**	−2.18	−1.73	100	3	1.5×10^{-1}	1.2×10^5
Comp ($\times 10^5$)	**2.40**	**2.45**	**2.40**	**2.40**	1.19	−	5000	20	1×10^3	6.3×10^3
Con ($\times 10^5$)	**2.51**	**2.56**	**2.51**	**2.51**	1.09	−	2740	25	1×10^2	2.9×10^3
Pol ($\times 10^5$)	**5.67**	−	−	−	**5.67**	−	2000	20	2×10^0	9.6×10^8
HDG ($\times 10^6$)	**10.66**	−	−	−	3.59	−	24073	50	1×10^{-5}	2.3×10^8

within a sentence *unless* Topic 8 also occurs (similar to to the exclusion rule). This KB results in more "on-concept" topics, including new terms {epo, peripheral, erythroid} for Topic 2. While a full discussion is infeasible due to space constraints, blind relevance judgments by our biological collaborator confirm the effectiveness of Fold·all in discovering new terms related to the target concepts [Andrzejewski, 2010].

4.1 Generalization

We assess the quality of the learned topics ϕ by examining their generalization to unseen documents via cross validation. At training time, we perform inference with one of the six methods on the training documents and KB (if applicable) to estimate the topic-word multinomials ϕ. At test time, we hold ϕ fixed and perform LDA-style inference over $\mathbf{z}$ on the testing documents. Note the logic KB is *not* used during the test phase, allowing us to see whether the KB "generalizes" to the test corpus via the learned topics ϕ.

We measure such generalization by evaluating the joint logic and LDA objective (4) on the test documents. The results are presented in Table 2, where each cell contains the test set value of (4) averaged across folds. We collectively refer to Mir, M+L, CGS, and MWS as Fold·all inference methods because they consider both LDA and logic components of Fold·all, and the results show that they are indeed better at optimizing the joint logic and LDA objective (4) than topic modeling alone (LDA) or logic alone (Alchemy), and therefore better at integrating FOL into topic modeling.

We also directly examine the number of satisfied groundings on held-aside test documents for both Mir and LDA. Across all KBs and folds, the topics learned by Mir result in the satisfaction of as many, or more, test set groundings than the topics learned by standard LDA. For example, on the first test fold of Synth, inference with the standard LDA topics satisfies 1,040 the 1,600 non-trivial Cannot-Link groundings, while the topics learned using the KB (all Fold·all methods plus Alchemy) result in the satisfaction of all 1,600 groundings *even though the KB is not used for test set inference*. Similar results across all experiments demonstrate that the learned topics transfer KB influence to the new documents.

4.2 Scalability

We say that an experimental run fails (indicated by a "−" in Table 2) if it does not complete within 24 hours on a standard workstation with a 2.33 GHz processor and 16 GB of memory. For example, even though Pol is a relatively small corpus with a straightforward KB, the two free variables i and j cause the number of non-trivial groundings to grow $O(N^2)$ with corpus length N. This causes the failure of Fold·all inference schemes which work directly with rule groundings (MWS, M+L, and CGS). By sampling groundings, Mir is able to learn topics across the range of datasets and KBs, even in the presence of exponentially many groundings. Mir inference completes in roughly 5 minutes on the full HDG corpus, consuming less than 5 GB memory.

5 Discussion

The Fold·all model can also reformulate some prior LDA extensions, although we stress that rule weights must be user-supplied, not learned. Furthermore, inference for a logic-based encoding may not be more efficient than a custom inference procedure tailored to the specific model.

Concept-Topic Model [Chemudugunta *et al.*, 2008] ties special *concept* topics to specific concepts by constraining these special topics to only emit words from carefully chosen subsets of the vocabulary. The rule $\mathtt{Z}(i,t) \Rightarrow \mathtt{W}(i,w_{c1}) \vee \mathtt{W}(i,w_{c2}) \vee \ldots \vee \mathtt{W}(i,w_{cK})$ enforces that concept topic t only emits the words $w_{c1}, w_{c2}, \ldots, w_{cK}$.

Hidden Markov Topic Model [Gruber *et al.*, 2007] enforces that the same topic be used for an entire sentence, allowing topic transitions only between sentences. Using our previously introduced sentence predicate $\mathtt{S}(i,s)$, we can express the intra-sentence topic consistency constraint as $\mathtt{S}(i,s) \wedge \mathtt{S}(j,s) \wedge \mathtt{Z}(i,t) \Rightarrow \mathtt{Z}(j,t)$. The probabilities of inter-sentence topic transitions can be set by carefully choosing weights for rules of the form $\mathtt{S}(i,s) \wedge \neg\mathtt{S}(i+1,s) \wedge \mathtt{Z}(i,t) \Rightarrow \mathtt{Z}(i+1,t')$ for all transition pairs (t,t').

ΔLDA [Andrzejewski *et al.*, 2007], Discriminative LDA [Lacoste-Julien *et al.*, 2008], Labeled LDA [Ramage *et al.*, 2009] all allow the specification of special "restricted" topics which can be used only in specially labeled documents. The goal of this constraint is to encourage these special topics to capture interesting patterns associated with the document labels. If topic t should only be used in documents with label ℓ, we can encode this type of constraint as $\mathtt{Z}(i,t) \wedge \mathtt{D}(i,d) \Rightarrow \mathtt{HasLabel}(d,\ell)$.

We have introduced Fold·all and given some simple examples of how it can incorporate domain knowledge into topic modeling. We have also provided a scalable inference solu-

tion Mir. Experimental results confirm that the influence of the user-defined KB generalizes to unseen documents, and that Mir enables inference when other approaches fail.

Future work could allow the formulation of relational inference problems in Fold·all by defining additional query atoms other than $\texttt{Z}(i,t)$, as in general MLNs. For example, we could define unobserved predicate $\texttt{Citation}(d,d')$ to be *true* if document d cites document d'. Another direction is to investigate the utility of Mir for general MLN MAP inference.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, with additional support from NSF IIS-0953219, AFOSR FA9550-09-1-0313, and NIH/NLM R01 LM07050. We would like to thank Ron Stewart for his participation in the HDG experiments.

References

[Andrzejewski *et al.*, 2007] D. Andrzejewski, A. Mulhern, B. Liblit, and X. Zhu. Statistical debugging using latent topic models. In *ECML*, pages 6–17. Springer-Verlag, 2007.

[Andrzejewski *et al.*, 2009] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *ICML*, pages 25–32. Omnipress, 2009.

[Andrzejewski, 2010] D. Andrzejewski. *Incorporating Domain Knowledge in Latent Topic Models*. PhD thesis, University of Wisconsin–Madison, 2010.

[Beck and Teboulle, 2003] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167 – 175, 2003.

[Blei *et al.*, 2003] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[Chemudugunta *et al.*, 2008] C. Chemudugunta, A. Holloway, P. Smyth, and M. Steyvers. Modeling documents by combining semantic concepts with unsupervised statistical learning. In *ISWC*, pages 229–244. Springer, 2008.

[Collins *et al.*, 2008] M. Collins, A. Globerson, T. Koo, X. Carreras, and P.L. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *JMLR*, 9:1775–1822, 2008.

[Domingos and Lowd, 2009] P. Domingos and D. Lowd. Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155, 2009.

[Gerrish and Blei, 2010] S. Gerrish and D. Blei. A language-based approach to measuring scholarly impact. In *ICML*, pages 375–382. Omnipress, 2010.

[Griffiths and Steyvers, 2004] T.L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(Suppl 1):5228–5235, 2004.

[Gruber *et al.*, 2007] A. Gruber, M. Rosen-Zvi, and Y. Weiss. Hidden topic Markov models. In *AISTATS*, pages 163–170. Omnipress, 2007.

[Hu *et al.*, 2011] Y. Hu, J. Boyd-Graber, and B. Satinoff. Interactive topic modeling. In *ACL*. ACL, 2011.

[Huynh and Mooney, 2009] T.N. Huynh and R.J. Mooney. Max-margin weight learning for Markov logic networks. In *ECML-PKDD*, pages 564–579. Springer, 2009.

[Kersting *et al.*, 2009] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *UAI*, pages 277–284. AUAI Press, 2009.

[Kivinen and Warmuth, 1997] J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[Kok *et al.*, 2009] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, D. Lowd, J. Wang, and P. Domingos. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2009.

[Lacoste-Julien *et al.*, 2008] S. Lacoste-Julien, F. Sha, and M. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pages 897–904. MIT Press, 2008.

[Pang and Lee, 2004] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, pages 271–278. ACL, 2004.

[Petterson *et al.*, 2010] J. Petterson, A. Smola, T. Caetano, W. Buntine, and S. Narayanamurthy. Word features for latent Dirichlet allocation. In *NIPS*, pages 1921–1929. MIT Press, 2010.

[Poon and Domingos, 2006] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI*. AAAI Press, 2006.

[Ramage *et al.*, 2009] D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256. ACL, 2009.

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

[Riedel, 2008] S. Riedel. Improving the accuracy and efficiency of MAP inference for Markov logic. In *UAI*, pages 468–475. AUAI Press, 2008.

[Selman *et al.*, 1995] B. Selman, H. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, pages 521–532. AMS, 1995.

[Shavlik and Natarajan, 2009] J. Shavlik and S. Natarajan. Speeding up inference in Markov logic networks by preprocessing to reduce the size of the resulting grounded network. In *IJCAI*, pages 1951–1956. Morgan Kaufmann, 2009.

[Singla and Domingos, 2008] P. Singla and P. Domingos. Lifted first-order belief propagation. In *AAAI*, pages 1094–1099. AAAI Press, 2008.

[Thomas *et al.*, 2006] M. Thomas, B. Pang, and L. Lee. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *EMNLP*, pages 327–335. ACL, 2006.

[Wang and Domingos, 2008] J. Wang and P. Domingos. Hybrid Markov logic networks. In *AAAI*, pages 1106–1111. AAAI Press, 2008.

[Wang *et al.*, 2009] C. Wang, D. Blei, and F. Li. Simultaneous image classification and annotation. In *CVPR*, pages 1903–1910. IEEE, 2009.