Learning Interpretable Relational Structures of Hinge-loss Markov Random Fields

Yue Zhang and **Arti Ramesh**
SUNY Binghamton
{yzhan202, artir}@binghamton.edu

Abstract

Statistical relational models such as Markov logic networks (MLNs) and hinge-loss Markov random fields (HL-MRFs) are specified using templated weighted first-order logic clauses, leading to the creation of complex, yet easy-to-encode models that effectively combine uncertainty and logic. Learning the structure of these models from data reduces the human effort of identifying the right structures. In this work, we present an asynchronous deep reinforcement learning algorithm to automatically learn HL-MRF clause structures. Our algorithm possesses the ability to learn semantically meaningful structures that appeal to human intuition and understanding, while simultaneously being able to learn structures from data, thus learning structures that have both the desirable qualities of interpretability and good prediction performance. The asynchronous nature of our algorithm further provides the ability to learn diverse structures via exploration, while remaining scalable. We demonstrate the ability of the models to learn semantically meaningful structures that also achieve better prediction performance when compared with a greedy search algorithm, a path-based algorithm with L_1 regularization, and manually defined rules on two computational social science applications: i) modeling recovery in alcohol use disorder, and ii) detecting bullying.

1 Introduction

Machine-learning models that possess superior interpretability and ease of specification without compromising modeling power have the potential to positively impact numerous downstream application domains that use them. In this work, we focus on one such recently developed inherently interpretable model that efficiently combines logic and uncertainty, hinge-loss Markov random fields (HL-MRFs) [Bach *et al.*, 2017]. HL-MRFs are specified using probabilistic oft logic (PSL), which uses logical first-order clauses to capture domain knowledge. The weights of these clauses are learned in a supervised manner by grounding them in data instances with ground truth values. The capability of the model to represent and reason about complex relational dependencies in the data using continuous-valued variables, while simultaneously being easy to encode these dependencies using logical clauses/rules makes it a good choice in modeling numerous applications, especially in computational social science.

In this work, we present an approach to learn meaningful structures of HL-MRFs by adapting a deep reinforcement learning (RL) algorithm, asynchronous advantage actor-critic (A3C) [Mnih *et al.*, 2016]. The possibility of tackling huge state spaces asynchronously in A3C together with scalable inference in HL-MRFs paves the way for developing a scalable structure learning algorithm that allows for diverse exploration. Further, guiding this exploration using domain-specific semantic constraints is helpful in learning meaningful structures that can potentially also achieve good prediction performance.

Specifically, we make the following main contributions:
1. We present asynchronous advantage actor-critic for structure learning (A3SL), a deep RL algorithm for learning interpretable structures of HL-MRFs that takes into account semantic constraints on domain-specific insights to guide the discovery of structures while simultaneously maintaining good prediction performance and ability to learn from real-world data instances.
2. We encourage the learning of diverse models through exploration by including a diversity constraint on the actions. This can potentially help the domain expert choose the model with the best semantic meaning from a group of models that have similar prediction performance.
3. We evaluate our structure learning algorithm on two important real-world computational social science applications: i) modeling recovery from alcohol use disorder, and ii) detecting bullying in online interactions. We choose these applications as there is existing work on applying HL-MRFs in these domains and we are able to make a direct comparison of the learned rules from A3SL with the existing manually specified models. We show that the structures learned by our algorithm achieves better prediction performance when compared with structures learned using a greedy structure learning algorithm, structure learning algorithms developed for Markov logic networks (MLNs) (Hypergraph lifting [Kok and Domingos, 2009] and Grafting-light [Zhu *et al.*, 2010]), and manually defined model structures.

4. We demonstrate that our model is able to learn complex clauses that encode network interactions such as the friend network and meaningful latent variables through a complex combination of features and target variables using semantic constraints. We also demonstrate that our learned clauses resemble the manually specified clauses capturing the same semantic meaning, while also learning others via exploration. Ours is the first structure learning algorithm that directly optimizes for interpretability, learning networked dependency structures, and learning meaningful latent variables in the problem formulation itself, thus yielding structures with enhanced interpretability and semantic coherence.

2 Related Work

Perhaps the earliest work in structure learning for undirected models is by McCallum et al. for Markov Random Fields, in which a greedy search is used for either adding or removing candidate features [McCallum, 2003]. Structure learning in MRFs has also been cast as a feature selection problem and solved using L_1 regularization over features [Zhu *et al.*, 2010; Khosravi *et al.*, 2010]. Structure learning has also been studied in SRL models such as Markov Logic Networks (MLNs), where the first approach is to employ a greedy search technique for this problem [Kok and Domingos, 2005]. Bottom-up approaches that generate clauses by using paths to capture structural motifs in the data have also been proposed, where finally the paths are ranked according to a path ranking algorithm [Mihalkova and Mooney, 2007; Kok and Domingos, 2010; 2009]. Most recent algorithms for structure learning in MLNs use functional gradient boosting and online learning to create scalable variants of existing structure learning algorithms [Khot *et al.*, 2011; 2015; Huynh and Mooney, 2011]. Learning the structure of MLNs specific to tasks of interest have also been developed using inductive logic programming [Biba *et al.*, 2008], L_1-regularized learning [Huynh and Mooney, 2008], and iterative local search [Biba *et al.*, 2008]. Embar et al. [Embar *et al.*, 2018] recently develop a greedy path-based structure learning algorithm for PSL.

The primary differences between our approach and existing work are: i) we guide the model into learning structures that bring out the modeling capabilities of SRL models—modeling relational dependencies and latent variables in the data/domain, and ii) we approach the structure learning problem from the perspective of the domains in which they would be most helpful, allowing us to demonstrate the encoding of domain-specific semantic constraints to learn models that are both meaningful and have good prediction performance, while existing approaches are primarily data-driven.

3 Deep Reinforcement Learning for Structure Learning in HL-MRFs

3.1 Hinge-loss Markov Random Fields

HL-MRFs are a recently developed scalable class of continuous, conditional graphical models [Bach *et al.*, 2017]. HL-MRFs can be specified using probabilistic soft logic (PSL) [Bach *et al.*, 2017], a first-order logic templating language. In PSL, random variables are represented as logical atoms and weighted clauses define dependencies between them of the form: $\lambda : P(a) \wedge Q(a,b) \rightarrow R(b)$, where *P*, *Q*, and *R* are predicates, *a* and *b* are variables, and λ is the weight associated with the clause. The weight of the clause c indicates its importance in the HL-MRF model, which is defined as

$$P(\mathbf{Y}|\mathbf{X}) \propto \exp\Big(-\sum_{c=1}^{M} \lambda_c \phi_c(\mathbf{Y},\mathbf{X})\Big)$$
$$\phi_c(\mathbf{Y},\mathbf{X}) = (\max\{l_c(\mathbf{Y},\mathbf{X}),0\})^{\rho_c} \quad (1)$$

where $P(\mathbf{Y}|\mathbf{X})$ is the probability density function of a subset of logical atoms $\mathbf{Y}$ given observed logical atoms $\mathbf{X}$, $\phi_c(\mathbf{Y},\mathbf{X})$ is a *hinge-loss potential* corresponding to an instantiation of a clause c, and is specified by a linear function l_c and optional exponent $\rho_c \in \{1,2\}$. Since HL-MRFs operate on continuous random variables and encode dependencies using potential functions that are convex, MAP inference in these models is always a convex optimization problem. The combinatorial explosion of state space during inference in SRL models including MLNs [Richardson and Domingos, 2006] makes it computationally challenging for designing a structure learning algorithm that requires repeated inference. In contrast, the possibility of fast exact inference in HL-MRFs opens up opportunities to capitalize recent advancements in deep RL to learn HL-MRF structures.

The logical conjunction of Boolean variables $X \wedge Y$ can be generalized to continuous variables using the hinge function $max\{X+Y-1,0\}$, which is known as the Lukasiewicz t-norm. Disjunction $X \vee Y$ is relaxed to $min\{X+Y,1\}$, and negation $\neg X$ to $1-X$. For example, $A \Rightarrow B = \neg A \vee B$ has a distance to satisfaction which is a hinge function, $max\{A-B,0\}$. Thus, $A \Rightarrow B$ penalizes the probability of a state based on the extent to which $A > B$. Our approach to structure learning focuses on the learning logical constructs that particularly bring out the modeling capabilities in HL-MRFs such as:

1. Relational dependencies and network structures: *friends*$(U,U_1) \wedge$ *usesAlcoholWord*$(U_1,P) \rightarrow \neg$*recovers*(U), which captures if user U's friend U_1 uses alcohol words frequently in their posts, that could potentially negatively influence U's recovery.
2. Latent variables: Incorporating meaningful latent variables can be helpful in learning abstractions that enhance modeling capability and interpretability.

3.2 Problem Definition

The structure learning problem can be modeled as a space search and sequential decision problem using reinforcement learning. Since HL-MRFs are specified using templated first-order logic clauses (also referred to as rules) in continuous space using the probabilistic programming language PSL, our problem translates to learning the structure of these clauses. We know that a first order logic clause has the form *body* $\rightarrow$ *head*, and in PSL *head* only can contain one predicate and *body* has the form $b_1 \wedge b_2 \wedge ... \wedge b_N$, $N < L$, where L denotes the maximum length of a clause. And, the order of predicates b in a clause within the body and clause in the list of clauses do not matter. A clause of the structure $b_1 \wedge b_2 \wedge ... \wedge b_N \rightarrow$ *head* can be represented in different ways as shown below.

$$b_1 \wedge b_2 \wedge ... \wedge b_N \rightarrow head = head \vee \neg(b_1 \wedge b_2 \wedge ... \wedge b_N)$$
$$= \neg b_0 \vee \neg b_1 \vee \neg b_2 \vee .. \vee \neg b_N$$

Each clause has a formula $\neg b_0 \vee \neg b_1 \vee ... \vee \neg b_N$, where we replace *head* as b_0 for readability. This can be represented as a sequence $b_0, b_1, ..., b_N$, END, where END signifies the end of the sequence. If current length of sequence equals to L, we stop appending the current sequence. We can see that when we generate a sequence we do not need to specify which predicate is head in a clause beforehand, since inherently the model is undirected. After clause generation, we can randomly choose any predicate $\neg b_i$ as head, so finally the clause becomes $b_0 \wedge ... \wedge b_{i-1} \wedge b_{i+1} \wedge ... \wedge b_N \rightarrow \neg b_i$. In practice, we can always choose the target predicate as head for ease of interpretability. Here, any predicate b_i can refer to the original predicate or its negative counterpart. Not all the predicates have meaningful negative counterparts, for example for network predicates, which capture relationship between two users, such as *replies*(U_1, U_2, I): user U_1 replies to U_2's message I, the negative counterpart $\neg$ *replies*(U_1, U_2, I) is not very meaningful as we are primarily interested in modeling interactions and not lack thereof.

3.3 A3SL: Asynchronous Advantage Actor-critic Structure Learning algorithm for PSL

We present our asynchronous advantage actor-critic structure learning algorithm, A3SL, which adapts a recently developed neural policy gradient algorithm asynchronous advantage actor-critic (A3C) [Mnih *et al.*, 2016] for the structure learning problem. A3C is one of the most general and successful learning agents to date and has been shown to perform well in discrete and continuous action spaces. A3C's ability to asynchronously execute multiple agents in parallel on multiple instances in the environment offers both algorithmic improvements that allow for effective integration of feedforward and recurrent deep neural network architectures in RL algorithms as well as practical benefits in speed, making it an appropriate choice for our structure learning problem. The description of A3SL is given in Algorithm 1 and the details are given in Algorithm 2.

Environment - Actions, Rewards, and Constraints

Here, we define the problem space and reinforcement learning algorithm setup.

Environment ε: Our environment consists of predicates for features (denoted by X), target variables (Y), and latent variables (Z) and data corresponding to X and ground truth data for Y, except latent variables.

State s_t: Each intermediate state s_t at time t comprises of either a partially constructed or a complete set of first-order logic clauses, denoted by C. The asynchronous nature of A3SL helps in combating the combinatorial explosion in the state space, making it an appropriate fit for the problem.

Action a_t: During an action a_t at time t the algorithm adds a new predicate to the current clause or chooses to return the clause by adding an END to the clause. The algorithm can choose from the action space of all the defined predicates X, Y, Z, and their negative counterparts.

Transition: The environment evolves by updating the current clause c with the predicate chosen during a_t.

Reward r_t and Cumulative Reward R_t: r_t is the reward at a given time step t. $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total accumulated reward from time step t with discount factor $\gamma \in (0, 1]$. In our setting, R_t is equal to the value of the objective function J (see below). As in a standard reinforcement learning setting, $V^{\pi}(s) = E[R_t | s_t = s]$ is the expected return for following policy π from state s. The goal of the agent is to maximize the expected return from each state s_t. During training, the agent learns to maximize the cumulative reward, so the model can find the corresponding optimal model structure.

Objective: We define the objective function $J = L(Y, X) +$ *Interpretability Constraint*, where $L(Y, X)$ is the HL-MRF probability density, $L(Y, X) = \log P(Y|X)$, given by Eqn 1. Given a set of target predicates Y, a PSL model *m* consists of model structure C and corresponding weight vector Λ, comprising of individual clauses c and weights $\lambda_c \in R^+$. *Interpretability Constraint* consists of the constraints on the total number of clauses, the maximum possible length of a clause, and the domain-specific semantic constraints. These qualities have been shown to enhance interpretability in Bayesian clause sets [Wang *et al.*, 2015; Angelino *et al.*, 2018; Wang *et al.*, 2017a; Letham *et al.*, 2015]. The combination of semantic constraints and a performance-based utility allows our algorithm to learn structures that are interpretable and data-driven, thus optimizing for both while being able to rectify any domain-specific intuitions that are not true in the data. Hence, the objective function J is defined as,

$$J = \Big(L(Y, X) - \alpha_{\text{len}} * \frac{1}{|C|} \sum_{c \in C} \text{length}(c) - \alpha_{\text{num}} * |C| - \alpha_{\text{sem}} * \sum_{c \in C} (\text{Dist}(c) * \lambda_c) \Big) \quad (2)$$

where α_{len}, α_{num}, and α_{sem} parameters denote the strength of the different constraints and $\text{Dist}(c)$ denotes the deviation of clause c from semantic constraints discussed below. A3SL returns the value of C, Λ that maximizes the objective function.

Algorithm 1 A3SL: Asynchronous advantage actor-critic structure learning algorithm for each actor-learner thread

Input: A collection of predicates, $X = \{x_j, j = 1, ..., m\}$, $Y = \{y_j, j = 1, ..., n\}$, $Z = \{z_j, j = 1, ..., k\}$, Ground truth labels Y_g for Y

Let $C = \{c_0, c_1, .., c_M\}$ denote set of first-order logic clauses

Let C_{list} denote list of C obtained with reward > 0.

Output: Optimal C denoted by C^*

Global Variables: Input and output variables, parameter vectors θ and θ_v, counter T, convergence criteria T_{max}, thread convergence criteria t_{max}.

function C^* = *A3SL()*
  Initialize $T = 0$, thread step counter $t = 0$
  RuleEnd = false, ListEnd = false, $i = 0$, $C = \emptyset$, $t_{start} = t$
  repeat
    if ListEnd **then**
      Initialize ListEnd = false, $i = 0$, $C = \emptyset$
    Synchronize thread-specific local copies of parameters $\theta' = \theta$, $\theta'_v = \theta_v$
    Obtain state $s_t = \Psi(C)$, where Ψ is the first-order logic embedding function
    repeat
      PSL-Actor(θ')
    until ListEnd **or** $t - t_{start} == t_{max}$
    PSL-Critic(θ', θ'_v)
  until $T > T_{max}$
  C^* = optimal C from C_{list}
  return C^*

Algorithm 2 A3SL algorithm details for updating action and reward

function *PSL-Actor*(θ')
  Perform action $a_t \in \{X, \neg X, Y, \neg Y, Z, \neg Z, \text{END}\}$, according to policy $\pi(a_t|s_t;\theta')$
  // *Finding end of clause*
  if $a_t ==$ END **then**
    RuleEnd = true
  else
    Append a_t to c_i
    if $|c_i|$==N **then**
      RuleEnd = true
  // *Checking validity of clause*
  if RuleEnd **then**
    i++, RuleEnd = false
    if c_i is invalid **or** i==M **then**
      ListEnd = $true$
  // *Calculating weights and reward*
  if ListEnd **then**
    Initialize weights Λ for C
    Perform weight learning and update Λ
    Obtain reward r_t = *Utility(C,* Λ*)*
    Add C to C_{list}
  else
    Obtain reward r_t = 0
  New state $s_{t+1} = \Psi(C)$
  $t = t + 1$
  $T = T + 1$

function *PSL-Critic*(θ', θ'_v)
  Reset gradients $d\theta = 0$ and $d\theta_v = 0$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \end{cases}$
  for $k \in \{t-1, ..., t_{start}\}$ **do**
    $R = r_k + \gamma R$
    Accumulate gradients wrt θ':
    $d\theta = d\theta + \nabla_{\theta'}(\log \pi(a_i|s_k;\theta')(R - V(s_k;\theta'_v)) + \beta * H(\pi(s_t)))$
    Accumulate gradients wrt θ'_v:
    $d\theta_v = d\theta_v + \partial(R - V(s_k;\theta'_v))^2/\partial\theta'_v$
  Perform asynchronous update of θ using $d\theta$ and θ_v using $d\theta_v$

Constraints on Actions

We incorporate constraints on actions to guide the algorithm to discover clause structures that are grammar compliant, semantically coherent, and diverse.

Grammar Compliance After clause construction, it is checked for compliance with PSL grammar. If not compliant, the list of clauses excluding the current clause is returned. The following constraints encode necessary grammar compliance with PSL.

1. If the algorithm chooses a redundant predicate in the same clause, the action is ignored.
2. After clause construction, if there is no target or latent predicate in the learned clause, then it is discarded.

Domain-specific Semantic Constraints We illustrate the capability of our model to encode domain-specific semantic constraints using one of the computational social science application domains, modeling recovery from alcohol use disorder (AUD). By intuition, presence of alcohol-related signals indicates relapse (denoted by ¬recovery and sobriety-related signals indicates recovery (denoted by recovery). These right reasons can be captured using simple logical clauses such as the ones given below, which can guide the discovery of model structures that capture these reasons [Ross *et al.*, 2017]. We use a distance function, *Dist(c)* to capture if the learned clause structure complies with or deviates from the right reasons identified by the expert: *Dist(c)* = 0, if the clause complies with the right reasons and *Dist(c)* = 1, otherwise.

1) alcohol signal $\Rightarrow$ ¬recovers; 2) sober signal $\Rightarrow$ recovers
3) alcohol signal $\wedge$ ¬recovers $\Rightarrow$ ¬sober signal
4) sober signal $\wedge$ recovers $\Rightarrow$ ¬alcohol signal

For example, consider the clause: *usesAlcoholWord(U,AW)* $\Rightarrow$ $\neg$*recovers*(U), the distance to the right reason is 0, because it satisfies right reason 1 above. For the rule: *usesAcoholWord(U,AW)* $\wedge$ *containsSoberWord(U,I,SW)* $\Rightarrow$ ¬*recovers(U)*, the distance to the right reason is 0, because it satisfies the right reason 4. But for rule: *containsSoberWord(U,I,SW)* $\Rightarrow$ ¬*recovers(U)*, the distance to the right reason is 1. From the perspective of the objective function, we want the weights of the clauses that do not conform to the right reasons to have much smaller values, or even 0, which means the clause list excludes the clause corresponding to the wrong reason.

Diversity Encouragement across Models We may not always have access to this extra domain knowledge or the "right reasons". In this case, we can learn multiple diverse policies [Wang *et al.*, 2017b; Haarnoja *et al.*, 2017; Hong *et al.*, 2018; Liu *et al.*, 2017] so an appropriate policy can be selected from them. To enable this, during training, we encourage diverse exploration by adding an entropy prior in the policy gradient objective function as follows,

$$\arg\max_{\pi} E[\log \pi(a_t|s_t;\theta)(R_t - b_t(s_t)) + \beta * H(\pi(s_t))]$$
$$H(\pi(s_t)) = -\sum_{a_t} \pi(a_t|s_t)\log(\pi(a_t|s_t))$$

where the baseline $b_t(s_t)$ is approximated by the value function $V^{\pi}(s_t)$ and $H(.)$ is the entropy. The diverse exploration across models results in diverse HL-MRF models after training which are all capable of good prediction performance.

Implementation Details During training, we relax the clause length and the number of clauses constraints in the objective by setting maximal clause length L and maximal clause number M. We encode the set of clauses corresponding to each state s_t in vector space using one-hot encoding [Wang and Cohen, 2016]. We incorporate the semantic constraints during weight learning in HL-MRFs by including them in the maximum-likelihood weight update as shown in Equation 3. The second term corresponds to the semantic distance indicating deviation from the semantic constraints.

$$\frac{\partial J}{\partial \Lambda_c} = \frac{\partial \log P(Y|X)}{\partial \Lambda_c} - \alpha_{\text{sem}} * \text{Dist}(c)$$
$$= E_{\Lambda}[\Phi_c(Y,X)] - \Phi_c(Y,X) - \alpha_{\text{sem}} * \text{Dist}(c) \quad (3)$$

With the above relaxations, we use area under the ROC curve as the *Utility* function to calculate the reward. Hence, R_t = AUC-ROC, and r_t=0, when t is not terminal, and r_t = AUC-ROC, when t is terminal. In our experiments, the value and

the policy network both use a 3-layer feed-forward neural network architecture with *tanh* as the activation function. We set t_{max} = 6 and T_{max} = $100,000$. We use distributed implementation using multi-threading that takes advantage of the asynchronous and parallel nature of the algorithm.

4 Experiments

We conduct experiments to answer the following questions:
1. How well does our learned model structures perform in real-world prediction problems? We evaluate the efficacy of our learned structures in two computational social science modeling scenarios: i) predicting recovery from/relapse into alcohol use disorder (AUD) using a combination of behavioral, social, and linguistic signals, and ii) detecting bullying using linguistic signals. We choose these application areas as there is existing work that models these domains using HL-MRF models using manually defined rules that can serve as a comparison both quantitatively and qualitatively with our learned model structures [Zhang *et al.*, 2018; Tomkins *et al.*, 2018]. We compare the structure learned by A3SL with greedy structure learning algorithm (Greedy-SL), a combination of a path-based structure learning algorithm, hypergraph lifting [Kok and Domingos, 2009], with a feature selection based structure learning algorithm [Zhu *et al.*, 2010] (Hypergraph Lifting with L_1 regularization) and model designed by a human expert (human-expert), and logistic regression. Our Greedy-SL algorithm learns the clause structure from a list of predicates, whereas existing greedy structure learning for HL-MRFs [Embar *et al.*, 2018] and inductive logic programming [Zeng *et al.*, 2014] require a pre-defined set of clauses.
2. How well do our rules imbibe the desirable characteristics present in models designed by human experts? We compare the structure of the learned models using A3SL with model structures defined by human experts, both quantitatively and qualitatively, to demonstrate that our algorithm learns semantically meaningful structures that also are capable of achieving good prediction performance, thus combining the benefits of semantic coherence in human-defined clauses and performance-oriented data-driven clause structures.

4.1 Case 1: Modeling Recovery in AUD

To model recovery from AUD, we use the dataset in Zhang *et al.* [2018]. In this dataset, there are 302 users attending Alcoholics Anonymous (AA) labeled with recovery and relapse. For each of the AA-attending user, the dataset contains the most recent $3,200$ tweets for each of these users, containing a total of $274,595$ AA user tweets. For 302 AA users, there are $76,183$ friends in dataset. For each friend, the dataset contains $3,200$ tweets, containing a total of $14,921,997$ tweets. We follow Zhang *et al.* [2018] to extract the same features to enable a direct comparison.

Table 1 shows the 5-fold cross-validation results on predicting recovery and relapse of AA-attending users. We observe that A3SL achieves a statistically significant prediction performance with a rejection threshold of $p = 0.01$ when compared with the manually specified model on the same dataset [Zhang *et al.*, 2018], Greedy-SL, and a logistic regression baseline. Examining the learned structures qualitatively, we observe that A3SL learns a variety of different clauses, capturing the semantic domain-specific constraints of associating alcohol-related signals to relapse and sobriety-related signals to recovery, and also learning data-driven variations from explorations resulting in better prediction performance.

Tables 2 and 3 present a comparison of some representative clauses learned by A3SL with manually constructed ones. U refers to the AA-attending user that we want to predict recovery/relapse, U_1 refers to friend of AA-user U, I refers to user post, I_1 refers to friend post, and AW and SW refer to specific alcohol/sober word from a vocabulary of alcohol/sober words. We group the learned rules into two categories: i) local rules: rules that capture dependencies among features on the users to predict their recovery/relapse, and ii) network rules: rules that capture users' relationships with their friends to reason about the users' recovery/relapse. We find that A3SL learns semantically meaningful clauses containing multiple features of the same signal (set B) and both alcohol and sober signals (set C), thus appealing to domain experts as well as capturing uncertainty better. Similarly, A3SL learns a variety of network rules that combine linguistic features from users (*usesSoberWord(U, SW)*) with linguistic features from friends (*friendContainsSoberWord*(U_1, I_1, SW_1)) and uses the network structure (*friends(U,* U_1*)*, *friendRetweets*(U_1, *U,* I_1*)*) to reason about recovery. A3SL's ability to learn these network dependencies makes it a powerful structure learning algorithm that brings out the modeling power of relational models.

4.2 Case 2: Detecting Bullying Messages

To learn a model for detecting bullying messages, we use the Formspring dataset. The dataset has a total of $13,159$ messages. We follow existing work on the dataset that uses HL-MRFs to model bullying and construct similar feature predicates [Tomkins *et al.*, 2018; Zhang and Ramesh, 2018] and consider a similar latent variable *intensity(I)* as in [Tomkins *et al.*, 2018], indicating the extent of bullying in a post I. Note that ours is the first structure learning algorithm that is capable of learning meaningful latent variable dependencies and other existing work including our Greedy-SL algorithm is not capable of learning latent variables. Table 4 gives the comparison of performance of A3SL with Greedy-SL, manually defined rules inspired from Tomkins *et al.* [2018] and logistic regression baseline. We observe that A3SL not only performs the best (statistically significant, $p = 0.05$), but also learns meaningful latent variable associations (*intensity* with negative sentiment, *negative* and *anonymity*) as evident in Table 5. *BW* refers to the presence of a bullying word in the text. We can see that A3SL learns a variety of clauses: local rules, priors, containing different combinations of feature, latent variable, and target, abstracting the latent variable using a complex combination of features and target variables.

4.3 Grounding Coverage and Correlation of Latent Variables

When manually defining the structure of models such as HL-MRFs, it is often challenging to define network rules that correspond to intuition and are also true in the data. For example, the manually-defined network rule in Table 3 captures that

Model	AUC-PR Pos.	AUC-PR Neg.	AUC-ROC
Logistic Regression	0.5628	0.8820	0.7594
Greedy-SL	0.6294	0.8991	0.8105
Hypergraph Lifting (Kok et al. [2009]) with L_1 regularization (Zhu et al. [2010])	0.5313	0.8885	0.8301
Human-Expert [Zhang *et al.*, 2018]	0.7129	0.9180	0.8583
A3SL [our approach]	**0.7342**	**0.9325**	**0.8755**

Table 1: Area under precision-recall curve and ROC values for recovery and relapse prediction for Logistic Regression, Greedy-SL, Human Expert, and A3SL

Comparison of Local Rules learned from A3SL with human rules
A. Human Expert: Contains either alcohol or sober signal
1. soberTopic(U) $\rightarrow$ *recovers(U)*; *2. alcoholTopic(U)* $\rightarrow \neg$*recovers(U)*
B. A3SL: Containing either alcohol or sober signal
1. usesSoberWord(U, SW) $\wedge$ *containsSoberWord(U, I, SW)* $\rightarrow$ *recovers(U)*
2. usesAlcoholWord(U, AW) $\rightarrow \neg$recovers(U)
C. A3SL: Contains both alcohol and sober signals, capturing more uncertainty
1. alcoholTopic(U) $\wedge$ *usesSoberWord(U)* $\rightarrow$ *recovers(U)*
2. soberTopic(U) $\wedge$ *usesAlcoholWord(U, AW)* $\rightarrow \neg$ *recovers(U)*

Table 2: Local rules from AUD recovery prediction HL-MRF Model designed by human expert and learned by A3SL

Comparison of Network Rules learned from A3SL with human expert rules
D. Human Expert: Sample Network Rule
1. replies(U, U_1*, I)* $\wedge$ *friendRetweets(*U_1*, U,* I_1*)* $\wedge$ *friendContainsAlcoholWord(*U_1*,* I_1*, AW)* $\rightarrow \neg$*recovers(U)*
E. A3SL: Similar to Human Expert
1. friendReplies(U_1*,U,*I_1*)* $\wedge$ *friendContainsSoberWord(*U_1*,* I_1*, SW)* $\rightarrow$ *recovers(U)*
2. friends(U, U_1*)* $\wedge$ *friendContainsAlcoholWord(*U_1*,* I_1*, AW)* $\rightarrow \neg$*recovers(U)*
F. A3SL: Combines linguistic features from AA user's posts and friends' posts
1. friends(U, U_1*)* $\wedge$ *usesSoberWord(U, SW)* $\wedge$ *friendContainsSoberWord(*U_1*,* I_1*, SW)* $\rightarrow$ *recovers(U)*
2. friends(U, U_1*)* $\wedge \neg$*termFrequency(U)* $\wedge$ *usesAlcoholWord(U, AW)* $\wedge$ *friendContainsAlcoholWord(*U_1*,* I_1*, AW)* $\rightarrow \neg$*recovers(U)*

Table 3: Comparison of network rules from HL-MRF Model designed by human expert and learned by A3SL

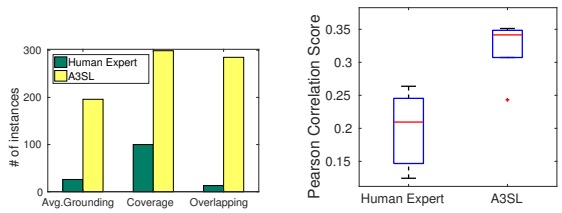
if the AA-user's friend uses alcohol words in a post and the AA-user retweets it, then that indicates that the user does not recover. From Figure 1(a), we observe that the average number of instances across all network rules in the human expert model is less than 30, indicating that the manually identified network rules may not be true in the data. On the other hand, we observe that the average groundings for the network rules learned by A3SL is approximately 200. Similarly, we observe that the network rules learned by A3SL have better data coverage, covering all the AA-user instances, while the manually defined network rules only cover a third of the instances. The human expert clauses have less overlap as humans intuitively tend to associate certain signals with recovery and others with relapse. But, A3SL clauses have superior overlap, allowing for representing the inherent uncertainty present in real-world data. Similarly, examining the correlation of the learned values of latent variable *intensity* with the target variable *bullying* in Figure 1(b), we find that it is consistently better correlated across the cross-validation folds, which indicates A3SL's capability to learn meaningful dependencies between features, latent and target variables.

Model	AUC-PR Pos.	AUC-PR Neg.	AUC-ROC
Logistic Regression	0.2696	0.8608	0.5173
Greedy-SL	0.3798	0.9140	0.6997
Hypergraph Lifting (Kok et al. [2009]) with L_1 regularization (Zhu et al. [2010])	0.2479	0.9116	0.6500
Human-Expert [Tomkins *et al.*, 2018]	0.3269	0.9130	0.6753
A3SL [our approach]	**0.3832**	**0.9202**	**0.7023**

Table 4: Area under precision-recall curve and ROC values for bullying and non-bullying prediction for Logistic Regression, Greedy-SL, Human Expert, and A3SL

Latent Variable Rules learned by A3SL
A. Feature and Latent Variable Rules:
1. hatredTopic(I) $\rightarrow$ *intensity(I)*
2. usesBullyingWord(I, BW) $\wedge \neg$*posSentiment(I)* $\rightarrow$ *intensity(I)*
B. Feature, Latent Variable, and Target Rules:
1. intensity(I) $\rightarrow$ *bullying(I)*
2. $\neg$*anonymity(I)* $\wedge \neg$*bullyings(I)* $\rightarrow \neg$*intensity(I)*
3. negative(I) $\wedge$ *intensity(I)* $\rightarrow$ *bullying(I)*
C. Priors:
1. $\neg$*bullying(I)*
D. Local Rules:
1. negative(I) $\wedge$ *hatredTopic(I)* $\rightarrow$ *bullying(I)*

Table 5: Latent variable associations to features and target learned by A3SL



(a) Comparison in the number of groundings in network rules between human expert and A3SL

(b) Pearson correlation between latent variable values and bullying in manually designed rules and A3SL

Figure 1: Grounding coverage of network clauses in AA recovery prediction (left) and correlation of latent variables with target variables in bullying detection (right)

5 Conclusion

In this work, we presented a deep RL structure learning algorithm for HL-MRFs, A3SL, that asynchronously uses multiple actor-learners in parallel to explore different parts of the state space, thus possessing the capability to learn diverse models, while remaining scalable. Our algorithm has the capability to encode guidance from domain experts to learn models that are semantically meaningful without compromising on performance. Our experiments on two important computational social science applications demonstrate that A3SL can learn model structures that capture the salient modeling requirements of the domains.

References

[Angelino *et al.*, 2018] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *JMLR*, pages 1–78, 2018.

[Bach *et al.*, 2017] Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss Markov random fields and probabilistic soft logic. *JMLR*, 18(109):1–67, 2017.

[Biba *et al.*, 2008] Marenglen Biba, Stefano Ferilli, and Floriana Esposito. Discriminative structure learning of Markov logic networks. In *ILP*, 2008.

[Embar *et al.*, 2018] Varun Embar, Dhanya Sridhar, Golnoosh Farnadi, and Lise Getoor. Scalable structure learning for probabilistic soft logic. *IJCAI StarAI Workshop*, 2018.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *ICML*, 2017.

[Hong *et al.*, 2018] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. In *NIPS*, 2018.

[Huynh and Mooney, 2008] Tuyen N. Huynh and Raymond J. Mooney. Discriminative structure and parameter learning for Markov logic networks. In *ICML*, 2008.

[Huynh and Mooney, 2011] Tuyen N. Huynh and Raymond J. Mooney. Online structure learning for Markov logic networks. In *ECML-PKDD*, 2011.

[Khosravi *et al.*, 2010] Hassan Khosravi, Oliver Schulte, Tong Man, Xiaoyuan Xu, and Bahareh Bina. Structure learning for Markov logic networks with many descriptive attributes. In *AAAI*, 2010.

[Khot *et al.*, 2011] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude Shavlik. Learning Markov logic networks via functional gradient boosting. In *ICDM*, 2011.

[Khot *et al.*, 2015] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude Shavlik. Gradient-based boosting for statistical relational learning: the Markov logic network and missing data cases. *Machine Learning*, pages 75–100, 2015.

[Kok and Domingos, 2005] Stanley Kok and Pedro Domingos. Learning the structure of Markov logic networks. In *ICML*, 2005.

[Kok and Domingos, 2009] Stanley Kok and Pedro Domingos. Learning Markov logic network structure via hypergraph lifting. In *ICML*, 2009.

[Kok and Domingos, 2010] Stanley Kok and Pedro Domingos. Learning Markov logic networks using structural motifs. In *ICML*, 2010.

[Letham *et al.*, 2015] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, pages 1350–1371, 2015.

[Liu *et al.*, 2017] Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. *arXiv*, 2017.

[McCallum, 2003] Andrew McCallum. Efficiently inducing features of conditional random fields. In *UAI*, 2003.

[Mihalkova and Mooney, 2007] Lilyana Mihalkova and Raymond J. Mooney. Bottom-up learning of Markov logic network structure. In *ICML*, 2007.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, pages 107–136, 2006.

[Ross *et al.*, 2017] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*, 2017.

[Tomkins *et al.*, 2018] Sabina Tomkins, Lise Getoor, Yunfei Chen, and Yi Zhang. A socio-linguistic model for cyberbullying detection. In *ASONAM*, 2018.

[Wang and Cohen, 2016] William Yang Wang and William W Cohen. Learning first-order logic embeddings via matrix factorization. In *IJCAI*, 2016.

[Wang *et al.*, 2015] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. Or's of and's for interpretable classification, with application to context-aware recommender systems. *arXiv*, 2015.

[Wang *et al.*, 2017a] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A Bayesian framework for learning rule sets for interpretable classification. *JMLR*, pages 2357–2393, 2017.

[Wang *et al.*, 2017b] Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. Robust imitation of diverse behaviors. In *NIPS*, 2017.

[Zeng *et al.*, 2014] Qiang Zeng, Jignesh M Patel, and David Page. Quickfoil: Scalable inductive logic programming. *VLDB Endowment*, 8(3):197–208, 2014.

[Zhang and Ramesh, 2018] Yue Zhang and Arti Ramesh. Fine-grained analysis of cyberbullying using weakly-supervised topic models. In *DSAA*, 2018.

[Zhang *et al.*, 2018] Yue Zhang, Arti Ramesh, Jennifer Golbeck, Dhanya Sridhar, and Lise Getoor. A structured approach to understanding recovery and relapse in AA. In *WWW*, 2018.

[Zhu *et al.*, 2010] Jun Zhu, Ni Lao, and Eric P Xing. Grafting-light: fast, incremental feature selection and structure learning of Markov random fields. In *SIGKDD*, 2010.