Negative Weights in Hinge-Loss Markov Random Fields

Charles Dickens*[1] **Eriq Augustine***[1] **Connor Pryor**[1] **Lise Getoor**[1]

[1]Computer Science Department , University of California, Santa Cruz , Santa Cruz, California, USA

Abstract

Hinge-loss Markov random fields (HL-MRF) are a class of probabilistic graphical models with density functions that admit tractable MAP inference. When paired with an expressive modeling framework, HL-MRFs are powerful tools for performing structured prediction. One such framework, probabilistic soft logic (PSL), uses weighted first-order logical statements to incorporate domain knowledge and constraints into the HL-MRF structure. Traditionally, PSL restricts weights to be non-negative to ensure MAP inference remains tractable, but this limits the types of relations PSL models can represent. We propose three novel approaches to extending PSL's expressivity to support negative weights. Notably, we propose the use of Gödel logic for defining potentials from negatively weighted rules. This method improves upon prior work on this topic by preserving both the convexity and scale of the MAP inference problem. Moreover, we show where our new methods and two approaches from prior work overlap and where they most differ. All methods are implemented in PSL, and we introduce a tunable synthetic dataset designed to empirically compare the performance of predictions.

1 INTRODUCTION

Hinge-loss Markov random fields (HL-MRF) are a class of probabilistic graphical models (PGM) that are both tractable and expressive [Bach et al., 2017]. HL-MRFs admit log-concave probability density functions with a structure that allows for highly-efficient maximum-a-posteriori (MAP) inference. HL-MRFs are particularly powerful when paired with a statistical relational learning (SRL) framework. SRL frameworks are tools for defining probabilistic models over relational data [Getoor and Taskar, 2007]. Probabilistic soft logic (PSL) is an SRL framework that uses weighted first-order logical statements (*rules*) to encode dependencies between relations and attributes in a domain [Bach et al., 2017]. PSL rules are instantiated with data to create potentials defining the HL-MRF density function. Typically, PSL performs maximum a posteriori (MAP) estimation over this density function in order to make predictions.

To ensure scalable inference, PSL enforces constraints on the HL-MRF density function, and hence the structure of the rules. These constraints limit the expressivity of the framework. In this paper we address the PSL constraint restricting the sign of weights of rules to be non-negative. This constraint is designed to preserve the convexity properties of MAP inference. Other SRL frameworks, such as Markov logic networks (MLN) [Richardson and Domingos, 2006], are not concerned with the convexity properties of the MAP inference problem and can support negative weights [Niu et al., 2011, Noessner et al., 2013]. In MLNs, a negative weight indicates that the negation of the associated rule should be satisfied. Since MLN's work with discrete valued variables, the negation of a rule is defined using Boolean logical semantics. PSL, on the other hand, translates rules to potentials by measuring the distance to satisfaction with Łukasiewicz real-valued logic [Klir and Yuan, 1995]. Instantiating a potential of a negated PSL rule using Łukasiewicz semantics results in concave potentials being added to the MAP objective function, breaking the convexity of inference.

Prior to this work, there have been two approaches proposed for supporting negative weights in PSL [Bach et al., 2017]. The first is to bias all user provided weights by a large enough constant to ensure non-negativity. This approach admits a convex MAP inference problem, however the semantics may not align with user expectations. The instantiated potentials do not measure the distance to satisfaction of the rules as one might expect. Another approach is to replace the non-convex potentials with multiple convex ones that have the same combined loss for critical variable assignments. This approach also results in a convex MAP inference problem, but may have unexpected properties for some variable values. Furthermore, this approach has scalability issues as the resulting PGM can grow exponentially with respect to the size of the rules. A larger HL-MRF model implies both slower MAP inference, as there are a larger number of potentials to optimize over, and more memory consumption.

In this work, we propose three new approaches for supporting negative weights in PSL. The first instantiates potentials using an un-modified form of the rule and allows the weights to be negative in the objective. This method results in a MAP inference problem that is non-convex but admits an objective that can be easily separated into a difference of convex functions. The second method negates the negatively weighted rule and instantiates a potential using Łukasiewicz logic. This method again breaks the convexity of inference, but we show in special-cases that this method can be equivalent to the first. Lastly, we propose a method which similarly negates the negatively weighted rule, but instead instantiates a potential using Gödel logic [Klement et al., 2000]. The conjunction semantics of Gödel logic is defined by the concave function: $T_{\min}(x,y) = \min\{x,y\}$. Using this property we show that potentials instantiated from rules negated with Gödel logic preserve convexity and faithfully measure distance to satisfaction. We analyze and compare the convexity and scalability properties of the two previously proposed and the three novel methods. In addition, we develop a taxonomy of the negative weight methods and show when the methods are equivalent and when they differ. We introduce synthetic dataset generator designed to evaluate the impact of negative weights and measure the effectiveness of each approach.

2 PROBABILISTIC SOFT LOGIC

Probabilistic soft logic (PSL) is a declarative framework for defining a *hinge-loss Markov random field (HL-MRF)*. PSL provides a syntax for describing dependencies, referred to as *rules*, between attributes and relations in a domain, called *atoms*. Rules are expressed as weighted first-order logical statements and act as templates for instantiating potentials that define the HL-MRF. The following is an example of a weighted logical rule:

$$w: \text{LIKES}(\mathtt{U},\mathtt{I}_1) \wedge \text{SIMILAR}(\mathtt{I}_1,\mathtt{I}_2) \rightarrow \text{LIKES}(\mathtt{U},\mathtt{I}_2)\char`^2$$

This rule, defined over the atoms $\text{LIKES}(\mathtt{U},\mathtt{I}_1)$, $\text{SIMILAR}(\mathtt{I}_1,\mathtt{I}_2)$, and $\text{LIKES}(\mathtt{U},\mathtt{I}_2)$, models the idea that a user, $\mathtt{U}$, that likes an item $\mathtt{I}_1$, will also like a similar item $\mathtt{I}_2$. The *weight* of the rule, denoted by the variable w, represents the relative importance of satisfying the rule in the model. The squared term, ^2, modifies the form of the potential functions created by the rule.

A PSL rule is instantiated via a process referred to as *grounding*. During grounding, atom arguments are substituted with distinct entities from a provided dataset to create *ground rules*. Logical ground rules are converted to a disjunctive clause. For instance, one ground rule from the example provided above is:

$$\begin{aligned} w: &\neg\text{LIKES}(Alice,Coffee) \vee \neg\text{SIMILAR}(Coffee,Tea) \\ &\vee \text{LIKES}(Alice,Tea)\char`^2 \end{aligned}$$

Every unique instantiation of atoms, a_i, from the grounding process is associated with a corresponding random variable y_i. Then, PSL uses Łukasiewicz logical semantics to define potential functions over all n random variables $\mathbf{y} = (y_1, \cdots, y_n)$ [Klir and Yuan, 1995]. For a single ground logical rule, let I^- and I^+ be the set of indices corresponding to atoms that are and are not negated, respectively. Łukasiewicz logic defines the degree of truth of the disjunctive clause of the ground rule as

$$\min\left\{\sum_{i\in I^+} y_i + \sum_{i\in I^-}(1-y_i), 1\right\}$$

Then, potentials measure the distance to satisfaction of the ground rule.

$$\begin{aligned} \phi(\mathbf{y}) &= \left(1 - \min\left\{\sum_{i\in I^+} y_i + \sum_{i\in I^-}(1-y_i), 1\right\}\right)^p \\ &= \left(\max\left\{1 - \sum_{i\in I^+} y_i - \sum_{i\in I^-}(1-y_i), 0\right\}\right)^p \end{aligned}$$

Here, p represents the exponential term in the rule. If the ground rule is squared, $p = 2$, then the potential measures the squared distance. Continuing with the same example, let y_1, y_2, and y_3 correspond to the atoms $\text{LIKES}(Alice,Coffee)$, $\text{SIMILAR}(Coffee,Tea)$, and $\text{LIKES}(Alice,Tea)$, respectively. The potential instantiated from the logical ground rule mentioned above is:

$$\begin{aligned} \phi(\mathbf{y}) &= (\max\{1 - y_3 - (1-y_1) - (1-y_2), 0\})^2 \\ &= (\max\{y_1 + y_2 - y_3 - 1, 0\})^2 \end{aligned}$$

If an instantiation of an atom, a_i is observed to take the value x_i, then the corresponding random variable y_i is set to x_i. This partitions the random variables into an unobserved set $\mathbf{y}$ and an observed set $\mathbf{x}$. All potentials can thus be expressed as a function of $\mathbf{y}$ and $\mathbf{x}$.

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

Where $\ell(\mathbf{x},\mathbf{y})$ is a linear function of the random variables. The set of all potentials created from the grounding process are combined to define the HL-MRF:

Definition 1 (Hinge-loss Markov random field)**.** *Let* $\mathbf{y} = (y_1, \cdots, y_n)$ *be a vector of* n *variables and* $\mathbf{x} = (x_1, \cdots, x_{n'})$ *a vector of* n' *variables with joint domain* $[0,1]^{n+n'}$. *Let*

$\Phi = (\phi_1, \cdots, \phi_m)$ be a vector of m continuous potentials of the form

$$\phi_i(\mathbf{y}, \mathbf{x}) = (\max\{\ell_i(\mathbf{y}, \mathbf{x}), 0\})^{p_i} \tag{1}$$

where ℓ_i is a linear function of $\mathbf{y}$ and $\mathbf{x}$ and $p_i \in \{1, 2\}$.

*Given a vector of m weights, $\mathbf{w} = (w_1, \cdots, w_m)$, a **hinge-loss energy function** $f(\cdot)$ is defined as:*

$$f(\mathbf{y}, \mathbf{x}, \mathbf{w}) = \begin{cases} \sum_{i=1}^{m} w_i \phi_i(\mathbf{y}, \mathbf{x}) & (\mathbf{y}, \mathbf{x}) \in [0,1]^{n+n'} \\ \infty & o.w. \end{cases} \tag{2}$$

*A **hinge-loss Markov random field** $\mathscr{P}$ over random variables $\mathbf{y} \in [0,1]^n$ and conditioned on $\mathbf{x} \in [0,1]^{n'}$ is a probability density defined as:*

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{w}, \mathbf{x})} \exp(-f(\mathbf{y}, \mathbf{x}, \mathbf{w})) \tag{3}$$

where $Z(\mathbf{w}, \mathbf{x}) = \int_{\mathbf{y}} \exp(-f(\mathbf{y}, \mathbf{x}, \mathbf{w})) d\mathbf{y}$ is the partition function for the conditional distribution.

Maximum a posteriori (MAP) inference is the task of finding the assignment of the unobserved random variables, $\mathbf{y}$, given the observations $\mathbf{x}$ that achieves the mode or peak of the conditional HL-MRF distribution function $P(\mathbf{y}|\mathbf{x})$ with potentials $\phi = (\phi_1, \cdots, \phi_m)$ and weights $\mathbf{w} = (w_1, \cdots, w_m)$.

$$\arg\max_{\mathbf{y} \in \mathbb{R}^n} P(\mathbf{y}|\mathbf{x}) \quad = \quad \arg\min_{\mathbf{y} \in [0,1]^n} \sum_{i=1}^{m} w_i \phi_i(\mathbf{y}, \mathbf{x}) \tag{4}$$

Each individual potential $\phi_i(\mathbf{y}, \mathbf{x})$ takes the form of a hinge-loss (Equation 1), and is hence convex. Traditionally, PSL enforces an additional constraint that the weights are all strictly non-negative. With this, the MAP inference objective becomes a finite positive weighted sum of convex functions and is therefore convex. This allows for the application highly scalable convex solvers for finding global optimal solutions.

3 NEGATIVE WEIGHTS

In this work we remove the non-negativity constraint on the weights, $\mathbf{w}$, that PSL uses to define a HL-MRF. A negative weight has a direct impact on the structure of the HL-MRF conditional distribution and hence the MAP inference problem. We introduce five different approaches for interpreting negative weights in PSL. At a high level, the approaches are classifiable as either methods that consider weights independently from grounding or ones that modify the potential instantiation process by using the sign of the weight as an indicator to negate the rule. We refer to the former class of approaches as *weight based* and the latter as *negation based*.

To illustrate each interpretation, we use the following simple PSL model designed to predict unobserved instances of the atom Q(A):

$$w_1 : !\mathrm{Q}(\mathtt{A})\text{\^{}}2$$
$$w_2 : \mathrm{P}(\mathtt{A}) \rightarrow \mathrm{Q}(\mathtt{A})\text{\^{}}2$$

The first rule is a squared negative prior on Q(A) that encourages predictions near 0. Then the second rule is a squared implication that implies values for P(A) can be used as predictive signal for Q(A). A dataset with just a single entity $A = \{a\}$ is used to instantiate the PSL model. The grounding process will therefore create the following ground rules:

$$w_1 : !\mathrm{Q}(a)\text{\^{}}2 \tag{5}$$
$$w_2 : \neg\mathrm{P}(a) \vee \mathrm{Q}(a)\text{\^{}}2 \tag{6}$$

Throughout this discussion, let P(a) be observed with the truth value $x = 0.75$ and let y be the random variable corresponding to the atom Q(a). Finally, we assume that $w_2 < 0$ and $w_1 >= 0$.

3.1 WEIGHT-BASED APPROACHES

For weight-based approaches, the grounding process described in Section 2 is unchanged in the presence of a negative weight. Thus the potential functions for the two rules are:

$$\begin{aligned} \phi_1(y, x) &= [1 - (1 - y)]^2 = y^2 \\ \phi_2(y, x) &= (max\{1 - y - (1 - x), 0\})^2 \\ &= (max\{0.25 - y, 0\})^2 \end{aligned}$$

Therefore, the objective function, $f(\mathbf{y}, \mathbf{x})$, for this model is:

$$\begin{aligned} f(\mathbf{y}, \mathbf{x}) &= w_1 \phi_1(y, x) + w_2 \phi_2(y, x) \\ &= w_1 y^2 + w_2 (max\{0.25 - y, 0\})^2 \end{aligned} \tag{7}$$

3.1.1 Remove Non-Negativity Constraint on Weights

The first approach is to simply remove the non-negativity constraint on the weights in Definition 1. When weights are not constrained to be non-negative the HL-MRF MAP inference objective is no longer necessarily a positive sum of convex functions and is therefore not guaranteed to be convex. This behavior is demonstrated in the plot of the objective function for the example PSL model in Figure 1.

Though the objective is non-convex, it is expressable as a difference of convex (DC) functions [Hartman, 1959]. It is generally challenging to find a decomposition of an objective as a DC function. However, in this case the decomposition of the objective into a sum of a convex and concave functions comes naturally. Let Φ^+ and Φ^- index the positive and negative weighted hinge loss potentials,

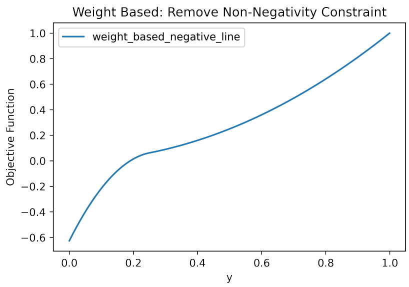

Figure 1: Example non-convex MAP inference objective function for a negative weight PSL model. The negative weight is interpreted using the approach described in Section 3.1.1. The weights for the model in Equation 6 are set to $w_1 = 1$ and $w_2 = -10$.

respectively. Then the HL-MRF MAP inference objective can be expressed as:

$$\arg\min_{\mathbf{y}|(\mathbf{y},\mathbf{x})\in\Omega} \sum_{i\in\Phi^+} w_i\phi_i(\mathbf{y},\mathbf{x}) - \sum_{i\in\Phi^-} (-w_i)\phi_i(\mathbf{y},\mathbf{x}) \tag{8}$$

The terms $\sum_{i\in\Phi^+} w_i\phi_i(\mathbf{y},\mathbf{x})$ and $\sum_{i\in\Phi^-}(-w_i)\phi_i(\mathbf{y},\mathbf{x})$ are both convex. MAP inference can thus be solved using the convex-concave procedure (CCCP) introduced by Yuille and Rangarajan (2003) and extended by Lipp and Boyd (2016). This approach would result in efficient optimization with convergence guarantees for finding local optimal solutions.

3.1.2 Biased Weights

Another weight-based approach, initially suggested by Bach et al. (2017), is to add a sufficiently large constant to make all weights positive. More formally, a non-negative parameter ε is introduced and all weights are biased by $\delta = \min(w_1, \cdots, w_m, 0) - \varepsilon$. For instance, the resulting objective of the running example using this method is

$$f(\mathbf{y},\mathbf{x}) = (w_1 - \delta)y^2 + (w_2 - \delta)(max\{0.25 - y, 0\})^2 \tag{9}$$

Biasing all weights in this way guarantees MAP inference is convex but results in an objective function with potentially different optimal solutions than that of the approach described in Section 3.1.1. To illustrate this behavior, consider the weight-based objective of the running example Equation 7. The derivative is:

$$\frac{df(\mathbf{y},\mathbf{x})}{dy} = \begin{cases} 2w_1y & y >= 0.25 \\ 2w_1y + 2w_2(0.25 - y) & y < 0.25 \end{cases}$$

Choosing weights $w_1 = -5$ and $w_2 = -10$, by evaluating the critical points of this objective we find that it is minimized at $y = 1/6$. However, after adding a sufficiently large constant $\delta = \min(-5, -10, 0) - \varepsilon = -(10 + \varepsilon)$ to the weights, the objective is minimized at $y = 0.0$. This result demonstrates that the solution set of the MAP inference problem is generally not invariant to translations of the rule weights.

3.2 NEGATION-BASED APPROACHES

Negation-based approaches modify the potential instantiation process, i.e., grounding. These approaches interpret negative weights as an indication to negate the corresponding rule. For instance, in our running exampled, the approaches described in this section first negate the rule with the negative weight w_2:

$$\mathrm{P}(a) \wedge \neg \mathrm{Q}(a) \tag{10}$$

Then, a potential is instantiated from the negated rule with an associated weight $|w_2|$. The differences between theses approaches comes from how the instantiated potentials measure the distance to satisfaction of this negated rule.

3.2.1 Łukasiewicz Negation

The first negation-based approach to negative weights directly applies Łukasiewicz logical semantics to define potentials. The degree of truth of the example negated rule, (10), using Łukaseiwicz logic is:

$$\max\{(x + (1 - y) - 1, 0\} \tag{11}$$

The distance to satisfaction of this rule, i.e., the potential that would be instantiated, is thus

$$\phi_2(y,x) = (1 - \max\{x + (1 - y) - 1, 0\})^2 \tag{12}$$

$$= \min\{1 + y - x, 1\}^2 \tag{13}$$

The surface plot of this potential as a function of x and y is shown in Figure 2. Notice that this potential is neither a convex nor a concave function of x and y. Adding this potential to the MAP objective breaks the convexity of inference. If, on the other hand, the potential was not squared, then it would be concave and the objective could be decomposed into a sum of concave and convex functions, i.e., a DC function. In this case the CCCP could be applied in the same way as described in Section 3.1.1. In fact, the relation between this approach and the approach discussed in Section 3.1.1 goes deeper. MAP inference for PSL models with Łukasiewicz negated implications and MAP inference with negative weighted potentials is equivalent when the rules are non-squared. That is to say, for non-squared rules, MAP inference, when one allows weights to be negative, is equivalent to allowing logical rules of the form: $w : A \wedge \neg B$, where A and B are arbitrary conjunctive and disjunctive clauses.

Theorem 1. *Let $A = (A_1 \wedge A_2 \wedge \cdots \wedge A_{n_A})$ be a conjunction of n_A (possibly negated) atoms and $B = (B_1 \vee B_2 \vee \cdots \vee B_{n_B})$ be a disjunction of n_B (possibly negated) atoms. Let $w > 0$. The MAP inference problem for a PSL model with the non-squared rule $w : A \wedge \neg B$ is equivalent to the MAP inference problem for a PSL model with the non-squared rule $-w : A \implies B$.*

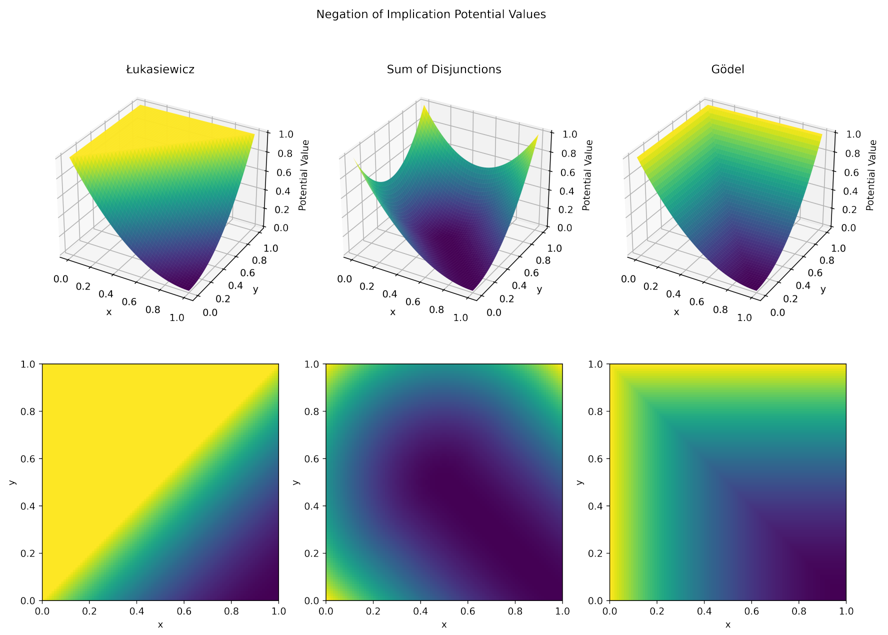

Figure 2: Surface plots and heat maps of the potential values instantiated by the example negated rule (10) using the three negation-based approaches.

Proof. We need to show both directions for equivalence.

($\Longrightarrow$) Suppose we have a PSL model with the rule $w : A \wedge \neg B$ such that $w > 0$. This rule is translated into a weighted potential of the form:

$$w\phi(\mathbf{y}) = w\phi_{A \wedge \neg B}(\mathbf{y}) \tag{14}$$

$$= w(1 - \max\left\{\sum_{i=1}^{n_A} y_{A_i} + \sum_{i=1}^{n_B}(1 - y_{B_i}) - (n_A + n_B - 1), 0\right\}) \tag{15}$$

$$=^{*} w - w\max\left\{\sum_{i=1}^{n_A} y_{A_i} - \sum_{i=1}^{n_B} y_{B_i} - (n_A - 1), 0\right\} \tag{16}$$

$$=^{**} w - w\max\left\{1 - \sum_{i=1}^{n_A}(1 - y_{A_i}) - \sum_{i=1}^{n_B} y_{B_i}, 0\right\} \tag{17}$$

* The constant term in the summation corresponding to B cancels with the n_b term.
** Move n_A into summation corresponding to A.

Denote the potential instantiated by $A \rightarrow B$ by $\phi_{A \rightarrow B}(\mathbf{y})$. Note that (17) is equivalent to $w - w\phi_{A \rightarrow B}(\mathbf{y})$. Let t_- and t_+ be the index sets for the potentials instantiated by the rules $w : A \wedge \neg B$ and $w : A \rightarrow B$, respectively. Then MAP inference becomes:

$$\min_{\mathbf{y} \in [0,1]} \sum_{i \in t_+} w_i \phi_i(\mathbf{y}) + \sum_{i \in t_-} w_i \phi_{A \wedge \neg B, i}(\mathbf{y}) \tag{18}$$

$$= \min_{\mathbf{y} \in [0,1]} \sum_{i \in t_+} w_i \phi_i(\mathbf{y}) + \sum_{i \in t_-} w_i - w_i \phi_{A \rightarrow B, i}(\mathbf{y}) \tag{19}$$

$$= \min_{\mathbf{y} \in [0,1]} \sum_{i \in t_+} w_i \phi_i(\mathbf{y}) - \sum_{i \in t_-} w_i \phi_{A \rightarrow B, i}(\mathbf{y}) \tag{20}$$

($\Longleftarrow$) The other direction follows similarly. □

3.2.2 Conjunction to Sum of Disjunctions

Bach et al. (2017) propose a second method for supporting conjunctive rules of the form $w : A \wedge B$^2 while preserving convexity. As the negative form of an implication has this structure, shown by the example in (10), this technique can be applied to support negative weights. They first express the rule as a CNF with the following structure:

$$A \wedge B = (A \vee B) \wedge (\neg A \vee B) \wedge (A \vee \neg B) \tag{21}$$

Each disjunction is then used to define a unique potential using Łukasiewicz semantics. For instance the potentials that are instantiated for the example negatively weighted rule, (10), are:

$$\phi_{2,1}(y,x) = (1 - \min\{x + (1 - y), 1\})^2$$
$$\phi_{2,2}(y,x) = (1 - \min\{(1 - x) + (1 - y), 1\})^2$$
$$\phi_{2,3}(y,x) = (1 - \min\{x + y, 1\})^2 \tag{22}$$

Each instantiated potential is assigned a weight equal to the positive value of the original rule and added to the HL-MRF energy function. The additive loss of the potentials instantiated from each of the disjunctive terms is the same as the single conjunction for discrete variable value assignments. This property can be seen in Table 1. However, Figure 2 shows that the potential values corresponding to Łukasiewicz conjunction and the three disjunctions are different for non-discrete variable value assignments. The number of additional potentials instantiated by this procedure grows exponentially with the number of atoms involved in the rule. Thus, for complex rules involving many atoms, this method can run into scalability issues.

A	B	$\phi(A \wedge B)$	$\phi(A \vee B)$	$\phi(\neg A \vee B)$	$\phi(A \vee \neg B)$
1	1	0	0	0	0
1	0	1	0	1	0
0	1	1	0	0	1
0	0	1	1	0	0

Table 1: A truth table showing the value of the potential function of a conjunction and the three disjunctions that can represent it.

3.2.3 Gödel Negation

Another way to preserve convexity but still support conjunctive rules is to leverage the weak conjunction semantics of Łukasiewicz logic, also known as the Gödel logic [Klement et al., 2000]. The degree of truth of the conjunctive rule $A \wedge B$ using Gödel semantics is:

$$\min\{A, B\} \tag{23}$$

This is a concave function of A and B, and hence the distance to satisfaction of the clause is guaranteed to be convex. For instance, the potential that would be instantiated for the example negative weighted rule in Equation 6 is:

$$\phi_2(y,x) = \max\{1-x, y\}^2 \tag{24}$$

The plot of the potential in Equation 24 for the example negative weighted rule relative to the potential instantiated using Łukasiewicz strong conjunction semantics and the sum of disjunctions method is shown in Figure 2. Notice that for x fixed as 0 or 1, Lukasiewicz and Gödel potentials are the same. Then as the value for x gets closer to 0.5 the potential functions deviate significantly. In general, values of potentials instantiated using Łukasiewicz semantics are equivalent to those instantiated using Gödel semantics for discrete variable value assignments. Moreover, no additional potentials need to be instantiated to achieve this behavior, as was the case in Section 3.2.2.

4 EMPIRICAL EVALUATION

In this section, we evaluate the predictive performance and structural properties of PSL models instantiated using the different approaches to supporting negative weights discussed in Section 3. We answer the following questions for each method of interpreting the negative weighted rule: Q1) How does the method effect the predictive performance of the PSL model? and Q2) How does the method scale in terms of both the size of the instantiated model and the rate of convergence of PSL inference?

We evaluate all negative weight approaches on a collective prediction task. The following PSL model is used for all experiments and is designed to predict the unobserved ground $\textsc{Target}(\texttt{X},\texttt{Y})$ atoms.

$$\begin{aligned}
-1.0 :& \textsc{Predictor1}(\texttt{X},\texttt{Y}) \rightarrow \textsc{Target}(\texttt{X},\texttt{Y})\text{\textasciicircum}2 \qquad (25)\\
0.1 :& \textsc{Target}(\texttt{X}_1,\texttt{Y}) \wedge \textsc{Similar}(\texttt{X}_1,\texttt{X}_2)\\
&\qquad \rightarrow \textsc{Target}(\texttt{X}_2,\texttt{Y})\text{\textasciicircum}2 \qquad (26)\\
1.0 :& \textsc{Predictor2}(\texttt{X},\texttt{Y}) = \textsc{Target}(\texttt{X},\texttt{Y})\text{\textasciicircum}2 \qquad (27)\\
0.01 :& \neg\textsc{Target}(\texttt{X},\texttt{Y})\text{\textasciicircum}2 \qquad (28)
\end{aligned}$$

The first rule in the model, 25, is the only negative weight rule. The atom $\textsc{Predictor1}(\texttt{X},\texttt{Y})$ represents an abstract atom for predicting the target atom. Different approaches to interpreting this rule result in a different ground HL-MRF and hence a different MAP inference objective. The remaining rules are common modelling patterns. Each method we have described grounds the same potentials for these rules. The second rule in the model, 26, is a rule for propagating predictions or observed values of the $\textsc{Target}(\texttt{X},\texttt{Y})$ atoms for similar entities $\texttt{X}_1$ and $\texttt{X}_2$. This rule can be read as "if $\texttt{X}_1$ and $\texttt{X}_2$ are similar then the value of $\textsc{Target}(\texttt{X}_1,\texttt{Y})$ should be close to that of $\textsc{Target}(\texttt{X}_2,\texttt{Y})$". The third rule, 27, is referred to as a local predictor rule. This rule uses an external model, the local predictor, as a signal for predicting the $\textsc{Target}(\texttt{X},\texttt{Y})$ atoms. Specifically, this rules says that the local predictor value for $\texttt{X},\texttt{Y}$ should be close to PSL's prediction for $\textsc{Target}(\texttt{X},\texttt{Y})$. The final rule in the model, 28, acts as a negative prior. That is to say, this rule will result in a small loss for any non-zero prediction made for $\textsc{Target}(\texttt{X},\texttt{Y})$. This rule can also be thought of regularizer and is commonly used to get more stable predictions from PSL since it ensures a strongly convex MAP inference objective.

4.1 SYNTHETIC DATASET GENERATION

We generated three synthetic datasets to compare the properties and performance of HL-MRF models instantiated using the approaches discussed in Section 3. The three datasets are designed to represent a different distribution of the target atoms $\textsc{Target}(\texttt{X},\texttt{Y})$. The first dataset is generated such that the target atoms have discrete, $\{0,1\}$, truth values. The second and third datasets are generated such that the target atoms have real, $[0,1]$, truth values with different distributions.

The generation process for all the datasets follows the same general pattern. There are a total of 100 possible unique values that the $\mathtt{X}$ argument can be assigned and 1000 for the $\mathtt{Y}$ argument. First, the values for $\mathtt{X}$ are randomly clustered into 10 groups. Then, an ideal value for $\textsc{Target}(\cdot,\mathtt{Y})$ is generated for every possible value of $\mathtt{Y}$ for every group of $\mathtt{X}$ arguments. Values for $\textsc{Target}(\mathtt{X},\mathtt{Y})$ are generated by adding noise to the ideal group values corresponding to the group assignment of $\mathtt{X}$. The $\textsc{Target}(\mathtt{X},\mathtt{Y})$ data is then split into an observed and test set. The cosine similarity of the observed $\textsc{Target}(\mathtt{X},\mathtt{Y})$ values across the $\mathtt{Y}$ arguments for each $\mathtt{X}$ is used to define the observed $\textsc{Similar}(\mathtt{X}_1,\mathtt{X}_2)$ atoms. The observed local predictor atoms $\textsc{Predictor2}(\mathtt{X},\mathtt{Y})$ are generated by adding Gaussian, $\mathscr{N}(0,0.3)$, noise to the test $\textsc{Target}(\mathtt{X},\mathtt{Y})$ values. Finally, the $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})$ atom that is involved in the negative weighted rule is generated for all $(\mathtt{X},\mathtt{Y})$ arguments by first drawing a uniform $(0,1)$ random value. Then, if the random value is less than $(1-\textsc{Target}(\mathtt{X},\mathtt{Y}))$, the corresponding $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})$ atom is set to a uniform $(0,(1-\textsc{Target}(\mathtt{X},\mathtt{Y})))$ random variable. Otherwise $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})$ is set to $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})+\alpha$ where α is an exponential, $Exp(\beta=0.05)$, random variable. In this way, $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})$ can be used as a possibly noisy lower bound on $\textsc{Predictor1}(\mathtt{X},\mathtt{Y})$.

The difference between the three datasets occurs when the true $\textsc{Target}(\mathtt{X},\mathtt{Y})$ atoms are generated. For the *Discrete targets* dataset, the ideal $\textsc{Target}(\cdot,\mathtt{Y})$ values for each $\mathtt{X}$ group are generated as independent Bernoulli, $Bern(p=0.4)$, random variables. Then noise is added to the ideal values to generate $\textsc{Target}(\mathtt{X},\mathtt{Y})$ values for each $(\mathtt{X},\mathtt{Y})$. The ideal target value of the group $\mathtt{X}$ belongs to is flipped with probability $p=0.3$. For the *Uniform Real Targets* dataset, $\textsc{Target}(\cdot,\mathtt{Y})$ values for each $\mathtt{X}$ group are generated as independent uniform, $\mathbb{U}(0,1)$, random variables. Similarly, for the *Centered Real Targets* dataset $\textsc{Target}(\cdot,\mathtt{Y})$ values for each $\mathtt{X}$ group are generated as independent Gaussian, $\mathscr{N}(\mu=0.4,\sigma=0.1)$, random variables. Then for both the Uniform Real Targets and Centered Real Targets datasets, $\textsc{Target}(\mathtt{X},\mathtt{Y})$ values are generated by perturbing the ideal value of the group $\mathtt{X}$ belongs to with $\mathscr{N}(\mu=0,\sigma=0.1)$ noise.

4.2 RESULTS

We first answer question Q1 by running each proposed negative weight approach on the three synthetic datasets and compare the RMSE of their predictions. Ten independent folds of each variation of the synthetic dataset is generated. Then, using the model introduced in this section, an HL-MRF is instantiated using one of the five methods discussed in this paper and MAP inference is performed. MAP inference optimization is solved via ADAM stochastic gradient descent (SGD) [Kingma and Lei Ba, 2015]. The stepsize hyperparameter is set to $\alpha=0.1$ and the exponential decay rate parameters are set to the suggested defaults of $\beta_1=0.9$ and $\beta_2=0.999$. ADAM SGD is determined to converge when the difference in the objective value over a single epoch drops below a tolerance of 10^{-8}. The results of this experiment are shown in Figure 3.

Across all variations of the dataset the Gödel negation method is always among the top three performing approaches. This consistency is not seen for any of the other methods. This result is particularly encouraging as Gödel negation does not sacrifice the convexity properties of PSL MAP inference as does the methods of Łukasiewicz negation and negative weights. On the other end of spectrum, biased weights is consistently the worst performing method. This behavior can be explained by the fact that this method does not capture the relation between the instantiated $\textsc{Predictor1}(\mathtt{X},\ \mathtt{Y})$ and $\textsc{Target}(\mathtt{X},\ \mathtt{Y})$ atoms. The synthetic dataset is designed so that $\textsc{Predictor1}(\mathtt{X},\ \mathtt{Y})$ is a reliable lower bound for $1-\textsc{Target}(\mathtt{X},\ \mathtt{Y})$. However the instantiated potentials for the rule 25 using the biased weights method encourages solutions where the values for $\textsc{Target}(\mathtt{X},\ \mathtt{Y})$ are greater than or equal to $\textsc{Predictor1}(\mathtt{X},\ \mathtt{Y})$ atoms.

We address question Q2 by examining the size of the ground models and the convergence properties of ADAM SGD on the MAP inference problem. Table 2 shows the mean and standard deviation of the number of epochs over the set of instantiated potentials that is required to converge to a local optimal solution of the MAP inference problem for each of the five negative weight methods. This table shows that the method of biased weights consistently requires a larger number of epochs to converge to a MAP state. This behavior can be explained by the fact that the method can create an ill-conditioned problem where there is a larger variation in the magnitude of the weights associated with each potential. Another interesting observation from the table is that the Łukasiewicz negation method can have a high variation in the number of epochs required to reach a MAP state. The Łukasiewicz negation method instantiates an objective that is neither convex nor concave and a local optimal solution can be difficult to find. The table also shows that the Gödel negation method, which preserves convexity of the MAP inference problem, is consistently converging in a relatively low number of iterations.

Regarding the scale of the ground models, i.e., the number of instantiated potentials, the sum of disjunctions method is the only method that instantiates extra potentials. For reference, for the first fold of the synthetic dataset the sum of disjunctions method instantiated $79,744$ more potentials than the other four methods. This is a roughly 2% increase in model size, and this difference can increase exponentially with the number of atoms involved in the negative

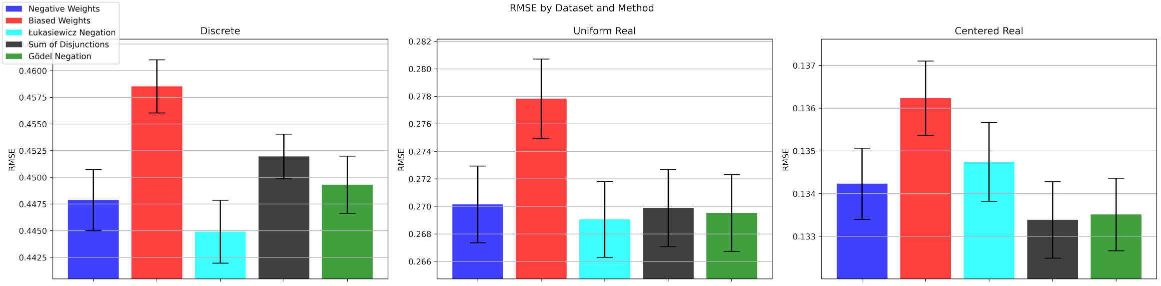

Figure 3: Bar plot of the mean RMSE of all five methods for supporting the negative weights in the experiment model for ten folds of each of the three variations of the synthetic datatset. One standard deviation is shown with error bars.

Method	Discrete Epochs	Uniform Real Epochs	Centered Real Epochs
Negative Weights	33.7 (5.48)	22.3 (3.83)	16.4 (3.38)
Biased Weights	37.6 (6.13)	44.1 (5.20)	32.6 (5.06)
Łukasiewicz Negation	44.4 (5.08)	21.4 (4.14)	22.8 (8.88)
Sum of Disjunctions	24.9 (6.28)	26.0 (5.68)	19.5 (3.41)
Gödel Negation	24.6 (3.86)	28.0 (4.99)	20.6 (4.70)

Table 2: The mean and standard deviation of the number of epochs required to reach convergence of ADAM SGD optimization for each method and dataset.

weighted rule. This has implications on both the time it takes to ground the model and the time to run inference. This is because each epoch of ADAM SGD optimization for the sum of disjunctions method has to optimize over more potentials.

5 RELATED WORK

Increasing the expressivity of SRL formalisms is an active direction of research. Probabilistic logic programs (PLP) define a distribution over a set of logical clauses [De Raedt et al., 2007, Vennekens et al., 2009, Sato and Kameya, 1997]. Meert and Vennekens (2014) extend CP-Logic, a causal PLP, by defining semantics for negations in the head of rules. The negations capture inhibition effects in the model, i.e., the inclusion of rules which can decrease the probability of an event. Based on similar motivations, an interpretation of negative probability weighted rules in PLPs was introduced by Buchman and Poole (2017b). The authors show that the semantics they introduce for negative probabilities capture the same relations as PLPs with negations and more. Moreover, they show that PLPs, even with negative weight semantics, are incapable of representing some relational distributions. Buchman and Poole (2017a) develop this line of research by showing show that PLPs with complex valued parameters are fully expressable.

Markov logic networks (MLN) use weighted logical rules to define probability distributions over binary $\{0, 1\}$ valued variables [Richardson and Domingos, 2006]. In MLNs, Boolean logic is used to define potentials and a negative weight is interpreted as a negation of the rule. Kuželka (2020) shows that the expressivity of the MLN framework is limited, i.e., only certain families of distributions could possibly be represented by MLNs. The authors show that by allowing weights to take complex values, MLNs are fully expressive for binary valued random variables.

6 CONCLUSIONS AND FUTURE WORK

Adding semantics to the PSL framework for supporting negative weights increases the expressivity of the models. With negative weights, more complex relations between atoms in the ground HL-MRF can be captured. In this paper, we discussed five unique of ways of interpreting negative weights, three of which are novel for PSL. Each method has different implications on the convexity and scale of the MAP inference problem. Most notably, we proposed Gödel conjunctive semantics, also referred to as weak Łukasiewicz conjunctive semantics. This method uses a principled and well-studied definition of real-valued logic to instantiate HL-MRF potentials that preserves both the convexity and scale of the HL-MRF MAP inference problem. Additionally, we showed multiple connections between the five methods by highlighting cases where the approaches are equivalent and where they differ. All methods were implemented into the PSL framework and empirically tested on three variations of a synthetic dataset. The Gödel negation method consistently provides quality predictions on the synthetic dataset while maintaining the tractability of HL-MRF MAP inference.

Directions for future work include further exploration of alternate real-valued logical semantics that can be used for instantiating HL-MRF potentials. Another future direction is the integration of negative weight semantics into the weight learning process. Similarly, as negative weights increases the expressivity of PSL, there are certainly implications on the task of rule discovery in PSL, i.e., structure learning.

7 ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation grants CCF-1740850, CCF-2023495, IIS-1703331, and IFDS DMS-2023495; AFRL; and the Defense Advanced Research Projects Agency.

References

Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research*, pages 1–67, 2017.

David Buchman and David Poole. Why rules are complex: Real-valued probabilistic logic programs are not fully expressive. In *Uncertainty in Artificial Intelligence (UAI)*, 2017a.

David Buchman and David Poole. Negative probabilities in probabilistic logic programs. *International Journal of Approximate Reasoning (IJAR)*, 83:43–59, 2017b.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, 2007.

Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. The MIT Press, 2007.

Thomas Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9 (3):707–713, 1959.

Diederik Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Erich Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*. Springer, 2000.

George Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Application*. Prentice Hall, 1995.

Ondrej Kuželka. Complex markov logic networks: Expressivity and liftability. In *Uncertainty in Artificial Intelligence (UAI)*, 2020.

Thomas Lipp and Stephen Boyd. Variations and extensions of the convex-concave procedure. *Optimization and Engineering*, 17:263–287, 2016.

Wannes Meert and Joost Vennekens. Inhibited effects in cp-logic. In *European Workshop on Probabilistic Graphical Models*, 2014.

Feng Niu, Christopher Ré, AnHai Doan, and Jude Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *VLDB*, 4:373–384, 2011.

Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *International Workshop on Statistical Relational AI (StarAI)*, 2013.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning (ML)*, 62:107–136, 2006.

Taisuke Sato and Yoshitaka Kameya. PRISM: A symbolic-statistical modeling language. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. CP-Logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245–308, 2009.

Alan L. Yuille and Anand Rangarajan. The convex-concave procedure. *Neural Computation*, 15(4):915–936, 2003.