

# Structure Learning for Relational Logistic Regression: An Ensemble Approach

**Nandini Ramanan**  
Indiana University

**Gautam Kunapuli**  
University of Texas at Dallas

**Tushar Khot**  
Allen Institute for AI

**Bahare Fatemi**  
University of British Columbia

**Seyed Mehran Kazemi**  
University of British Columbia

**David Poole**  
University of British Columbia

**Kristian Kersting**  
TU Darmstadt University

**Sriraam Natarajan**  
University of Texas at Dallas

## Abstract

We consider the problem of learning Relational Logistic Regression (RLR). Unlike standard logistic regression, the features of RLRs are first-order formulae with associated weight vectors instead of scalar weights. We turn the problem of learning RLR to learning these vector-weighted formulae and develop a learning algorithm based on functional-gradient boosting methods for probabilistic logic models. Our empirical evaluation on standard data sets demonstrates the superiority of our approach over other methods for learning RLR.

## Introduction

Statistical Relational AI (Getoor and Taskar 2007; Raedt et al. 2016) combine the representational power of logic with the ability of probability theory specifically, and statistical models in general to model noise and uncertainty. We consider the more recent, well-understood directed model of **Relational Logistic Regression** (RLR) (Kazemi et al. 2014a; 2014b; Fatemi, Kazemi, and Poole 2016). One of the key advantages of RLR is that they scale well with population size unlike other methods such as Markov Logic Networks (Poole et al. 2014). While these models are attractive from the modeling perspective, learning them is computationally intensive as this involves searching over multiple levels of abstraction. Most methods focus on the task of learning the parameters (weights of the logistic function) from data, where the rules (or relational features) are provided by the human expert. Inspired by the success of non-parametric learning methods for SRL models, we develop a learning method for full model learning of RLR models.

More specifically, we develop a gradient-boosting technique for learning RLR models. We derive the gradients for the different weights of RLR and show how the rules of the logistic function are learned simultaneously with their corresponding weights. Unlike the standard adaptations of the functional gradients, RLR requires learning a different set of weights per rule in each gradient step and hence requires learning multiple weights jointly for a single rule. The gradients correspond to a set of vector weighted clauses that are learned in a sequential manner. Each clause can be seen as a **relational feature** for the logistic function. Hence, RLR can

be viewed as a probabilistic combination function in that it can stochastically combine the distributions due to different set of parents. We demonstrate the effectiveness of this combination function on real data sets and compare against several baselines

## Functional Gradient Boosting for RLR

The conditional probability of an example  $y$  given weighted formulae  $\langle \varphi_1, w_{T1}, w_{F1} \rangle, \dots, \langle \varphi_k, w_{Tk}, w_{Fk} \rangle$  corresponding parents  $J_1, \dots, J_k$  is defined as:

$$\begin{aligned} \text{Prob}(y = 1 \mid J_1, \dots, J_k) = & \\ & \sigma(w_0 + w_{T1}\eta_T(\varphi_1\theta, J_1) + w_{F1} \cdot \eta_F(\varphi_1\theta, J_1) \\ & + \dots \\ & + w_{Tk}\eta_T(\varphi_k\theta, J_k) + w_{Fk} \cdot \eta_F(\varphi_k\theta, J_k)) \quad (1) \end{aligned}$$

where  $\sigma(\cdot)$  is the sigmoid function. For example, let  $y = \text{Popularity}(a)$  indicate the popularity of a professor  $a$ . Consider two formulae  $\phi_1 = \text{Publication}(A, P)$  and  $\phi_2 = \text{AdvisedBy}(A, S)$ . The weights of the first formula control the influence of the number of publications on the popularity of the professor where  $J_1 = \text{Publication}(a, P)$ . Similarly the second formula controls the influence of the number of students advised by the professor. For learning a model for RLR, we thus need to learn these clauses  $\phi_i$  and their weights  $w_{Ti}, w_{Fi}$  (the parents are determined by the structure of the clause). Also, we can assume that the bias term  $w_0$  can be part of the weight vectors for all the learned clauses. This allows a greedy approach that incrementally adds new clauses, such as FGB, to automatically update the bias term by learning  $w_0$  for each new clause. Our learning problem can be defined as:

**Given:** A set of grounded facts and the corresponding positive and negative grounded literals

**To Do:** Learn the set of formulae  $\varphi_i$  with their corresponding weight vector  $\mathbf{w}_i = [w_0, w_1, w_2]$ .

To simplify the learning problem, we introduce **vector-weighted clauses** (formulae), denoted as  $\{\mathbf{w} : \text{Clause}\}$ , that are a generalization of traditional weighted clauses with single weights. More specifically, our weighted clauses employ **three dimensions**, that is  $\mathbf{w} = [w_0, w_1, w_2]^T$ , where  $w_0$  is a bias/intercept,  $w_1$  is the weight over the satisfiable groundings of the current clause (analogous to  $w_{Ti}$ ) and  $w_2$  is the weight of the unsatisfiable groundings of the current clause

---

**Algorithm 1** Boosted RLR learning method

---

```
1: function B-RLR( $Y, X, p$ )  $\triangleright$  Initially  $F_0 := \gamma$ 
2:    $F_m := F_{m-1}$   $\triangleright$  Compute gradients,  $\Delta_i$  for  $y_i \in Y_p$ 
3:    $S_p := \text{COMPUTEGRAD}(Y_p, X, F_m)$ 
4:    $\hat{\psi}_m := \text{FITREG}(S_p, X, Y_p)$ ,  $F_m := F_m + \hat{\psi}_m$ 
5:   return  $F_m$   $\triangleright$  Repeat 3-5 for  $M$  iteration
6: end function
```

---

---

**Algorithm 2** Vector-weighted regression clause learning

---

```
1: function FITREG( $S, D, y$ )  $\triangleright$  Initially  $S = \{(y_i, \Delta_i)\}$  body
   :=  $\emptyset$ ;  $\mathbf{w} := [0, 0, 0]^T$ 
2:    $L := \text{POSLIT}(p(x), \text{body})$   $\triangleright$  Generate potential literals
3:   clause := ‘ $y := \text{body} \wedge \ell$ .’
4:    $t_i, f_i = \text{CALGROUND}(y_i, D, \text{clause})$   $\triangleright$  Calculate
   groundings per example
5:    $C_i := \text{COUNTMAT}(\{t_i, f_i\})$ ,  $\mathbf{w}(\ell) := (C^T C + \lambda I)^{-1} C^T \Delta_i$ ,  $\text{score}(\ell) := \text{SCOREFIT}(\mathbf{w}(\ell), \Delta)$   $\triangleright$  Repeat 3-5
   for  $L$  literals
6:    $\hat{\ell} := \arg \min_{\ell} \text{score}(\ell)$ ,  $\mathbf{w} := \mathbf{w}(\hat{\ell})$ ,  $\text{body} := \text{body} \wedge \hat{\ell}$ 
7:   return  $\mathbf{w}$ :  $y := \text{body}$ .
8: end function
```

---

(analogous to  $w_{F_i}$ ). For ease of exposition, we restrict our discussion to the highlevel algorithm while our extended paper (Ramanan and et al. 2018) presents detailed derivation of the component-wise functional-gradients.

We initialize the regression function with uniform prior  $\gamma$  i.e.  $F_0(y_i) = \gamma$ . Given the input training examples  $Y$  which correspond to the grounded instances of the target predicate  $y$  and the set of facts, i.e., the grounded set of all other predicates (denoted as  $X$ ) in the domain, we learn the set of vector-weighted clauses that influence the target. Current predicate is  $p$ . In the  $m^{\text{th}}$  iteration, we compute the gradients of the examples using the current model  $F_m$  and the parents of  $y$  as per this model (line 7). Given these regression examples  $S_p$ , we learn a vector-weighted clause using FITREGRESSION. This function uses all other facts  $X$  to learn the structure and parameters of clause. We then add this regression function,  $\hat{\psi}_m$  approximating the functional gradients to the current model,  $F_m$ . We repeat this over  $M$  iterations.

## Experiments and Results

We compare B-RLR approach to: (1) AGG-LR, which is standard logistic regression (LR) using the relational features, (2) ILP-RLR uses PROGOL (Muggleton 1995) for rule learning, then weight learning, and (3) MLN-B, boosted MLN structure learning method. We evaluate our approach on 1 synthetic data set; Smokes-Cancer-Friends and 4 standard relational data sets (Raedt et al. 2016); UW-CS, IMDB, WebKB and MovieLens. Due to high imbalance in the data sets, we use Precision-Recall and ROC curves to evaluate the models. In experiments, for MLN-B we set maximum number of clauses to 3, beam-width to 10 and maximum clause length to 4. From Figure 1, we see that B-RLR method is on par or better than most methods across all data sets. While MLN-B appears to be more mixed at first glance in ROC space, B-RLR is generally better in PR space. In WebKB, where we learn about unary predicate, B-RLR performs sig-

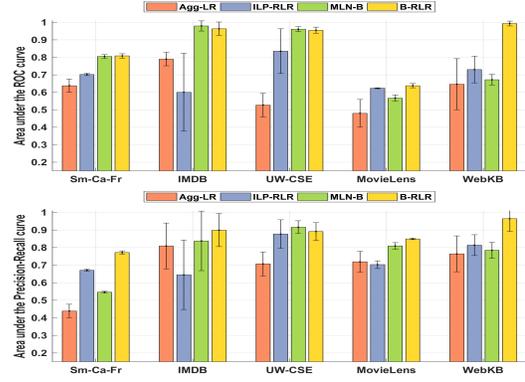


Figure 1: Comparing area under the ROC curve (Top) and PR curve (Bottom) for B-RLR approach to other baselines

nificantly better, depicting that RLR models are **natural aggregators** over the associated features. B-RLR significantly outperforms AGG-LR and ILP-RLR in all data sets.  $\lambda$  reflects high class imbalance in real-world domains and is usually within the set  $\{10^2, 10^{2.5}, 10^3, 10^{3.5}\}$ . We used paired t-test with p-values=0.05 for determining the statistical significance. B-RLR has tighter error bounds indicating lower variance and better generalization as shown in Figure 1.

## Conclusions

We considered the problem of learning relational logistic regression (RLR) models using functional-gradient boosting. Currently we learn a model for a single target. We can extend it to learn a joint model across multiple predicates in a manner akin to learning a relational dependency network. We hypothesize that RLR can be an effective function approximator for relational Markov decision processes.

## References

- Fatemi, B.; Kazemi, S. M.; and Poole, D. 2016. A learning algorithm for relational logistic regression: Preliminary results. *arXiv preprint arXiv:1606.08531*.
- Getoor, L., and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. MIT Press.
- Kazemi, S.; Buchman, D.; Kersting, K.; Natarajan, S.; and Poole, D. 2014a. Relational logistic regression. In *KR*.
- Kazemi, S.; Buchman, D.; Kersting, K.; Natarajan, S.; and Poole, D. 2014b. Relational logistic regression: The directed analog of markov logic networks. In *AAAI Workshop*.
- Muggleton, S. 1995. Inverse entailment and progol. *New generation computing* 13(3-4):245–286.
- Poole, D.; Buchman, D.; Kazemi, S.; Kersting, K.; and Natarajan, S. 2014. Population size extrapolation in relational probabilistic modelling. In *SUM*, 292–305. Springer.
- Raedt, L.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. *Statistical relational artificial intelligence: Logic, probability, and computation*. Morgan Claypool.
- Ramanan, N., and et al. 2018. Structure learning for relational logistic regression: An ensemble approach. In *Arxiv*.