# Adaptive Incremental Learning for Statistical Relational Models Using Gradient-Based Boosting

Yulong Gu and Paolo Missier

School of Computing Science, Newcastle University
Newcastle upon Tyne, United Kingdom
{y.gu11,paolo.missier}@newcastle.ac.uk

**Abstract.** We consider the problem of incrementally learning models from relational data. Most existing learning methods for statistical relational models use batch learning, which becomes computationally expensive and eventually infeasible for large datasets. The majority of the previous work in relational incremental learning assumes the model's structure is given and only the model's parameters needed to be learned. In this paper, we propose algorithms that can incrementally learn the model's parameters and structure simultaneously. These algorithms are based on the successful formalisation of the relational functional gradient boosting system (RFGB), and extend the classical propositional ensemble methods to relational learning for handling evolving data streams.

**Keywords:** Gradient-Based Boosting, Incremental Learning, Hoeffding Bound, Ensemble Methods, Statistical Relational Learning, Relational Regression Tree, Concept Drift

## 1 Introduction

Statistical Relational Learning (SRL) combines statistical methods with relational or logical models to address the challenge of applying statistical learning and inference approaches to problems which involve rich collections of objects linked together in a complex, stochastic and relational world. While these models are highly compact and expressive, the problem of learning them is computationally intensive. The learning includes two components as with standard graphical models, the structure that encodes the dependency between attributes and the parameters that quantify the uncertainty. Even in batch learning, structure learning is significantly difficult. Most existing incremental learning methods for SRL models assume the structure is given and only learn the model's parameters, which makes these learning methods less applicable to real-world problems.

A recent algorithm called Relational Functional Gradient Boosting (RFGB) [1], based on Friedman's functional gradient boosting [2], addressed the problem by learning structure and parameters simultaneously. This was achieved by learning a set of relational regression trees (RRT) for modelling the distribution of each variable given all the other variables. The key insight is to view the problem of learning relational probabilistic functions as a sequence of relational regression problems.

As the RFGB system has reduced the complex learning problem to a relational regression problem, the potential to extend this system with relational-enabled propositional solutions can be exploited. There have already been extensions that enable the RFGB system to handle imbalanced data with Soft Margin [3] and incomplete data with Structural EM [4]. In this paper, inspired by HTILDE-RT [5] and DWM [6], we have further developed the RFGB system for incremental learning by upgrading the RRT used in the RFGB system with the use of Hoeffding bound and the concept adaptation strategy that is successfully employed in the CVFDT [7], and incorporating the upgraded RRT with ensemble methods through a Hoeffding-based stability metric to cope with concept drift.

In this work, Hoeffding Relational Regression Tree (HRRT), Relational Incremental Boosting (RIB) and Relational Boosted Forest (RBF) are introduced for adaptive incremental learning in SRL setting. This paper is organized as follows: in Section 2, the background including RFGB and CVFDT are reviewed; in Section 3, the rule stability, HRRT, RIB, and RBF are explained; in Section 4, conclusions and future work are provided.

## 2 Background

### 2.1 Relational Functional Gradient Boosting

Friedman [2] proposed a boosting approach where functional gradients are computed for each example over the objective function. The RFGB [1] considers the objective function as a sigmoid function (Equation 2). These gradients correspond to the difference between the true label and predicted probability for an example and are then used to generate a regression dataset. In each iteration, a RRT $\Delta_i$ is learned to fit to these gradients and added into the potential function $\psi_m$:

$$\psi_m = \psi_0 + \Delta_1 + ... + \Delta_{m-1} \tag{1}$$

This approach learns a boosted probabilistic model that corresponds to the local conditional probability distribution in some SRL models.

$$P(y_i|Pa(y_i);\psi) = \frac{e^{\psi(y_i;\boldsymbol{x_i})}}{\sum\limits_{y' \in Y} e^{\psi(y';\boldsymbol{x_i})}} \tag{2}$$

We will upgrade the RRT learner used by the RFGB system to enable incremental learning by incorporating CVFDT.

### 2.2 CVFDT

The propositional incremental dceision tree learner VFDT [8] uses Hoeffding bounds to guarantee that its output is asymptotically nearly identical to that of a batch learner. CVFDT [7] introduces a concept adaptation strategy to VFDT for handling concept drift. CVFDT keeps the incrementally learned tree consistent with a sliding window of examples. In this paper, we call the contents of a sliding window the window data. The concept adaptation strategy is implemented as follows: The statistics of the window data is encoded into the sufficient statistics for nodes in the tree. These sufficient statistics are then used to periodically check if the Hoeffding bound holds at each node , and if not, the algorithm will create an alternate sub-tree for the node to compete with the failed sub-tree in terms of prediction accuracy. In the case where the alternate sub-tree beats the failed one, the failed sub-tree and the old conflicting rules encoded in it will be discarded and replaced by the alternate sub-tree. The disadvantage of eliminating old conflicting rules entirely is that the model will only support current window data and usually result in larger prediction variance.

In the following section, we will introduce our algorithms using ensemble methods to handle concept drift by allowing the coexistence of conflicting rules in relational setting.

## 3 Incremental Learning Algorithms

### 3.1 Rule Stability and Hoeffding-Based Stability Metric

In this paper, we qualify a tree as stable with the following considerations. The tree is highly consistent with the window data, and the rules encoded in the tree are real rules that can, with high confidence, interpret the distribution associated with the data generator. We will boost a tree when it is stable so that the objective function (equation 2) is best optimized for the current window data and the newly found stable rules can be solidified and transformed into established rules. On the other hand, a stable tree is highly resistant to concept drift, as the newly found stable rules will require many counter-examples to be invalidated. Inspired by DWM [6] in which the ensemble methods are proven efficient by extensive experiments for adapting drifting concepts, we propose to employ ensemble methods to tackle the concept drift problem in the relational setting.

In the presence of concept drift, we want to learn how stable the rules that the tree has learned so far are, as an indicator of the stability of a tree. We introduce a metric that measures the stability of a tree as follows.

**Definition 1.** Define the *Rule Stability* of a model as the size of the smallest change in sample $D$ that may cause rule $r'$ to become superior to $r$. This is as shown in equation 3, where $D'$ is $D$ after change.

$$Learner: \ (Diff(D, D') = n, r) \rightarrow r' \tag{3}$$

According to CVFDT [7], to guarantee that the rule learned from a data sample is the real rule $r$ from the corresponding population with desired confidence $1-\delta$, the condition $\varDelta\bar{G}_{X_a,X_b} > \epsilon$ must be met, where $\varDelta\bar{G}_{X_a,X_b} = \bar{G}(X_a) - \bar{G}(X_b)$, $\bar{G}(X_i)$ is the average of results from splitting function $G(X_i)$, $X_a$ and $X_b$ is the working and second best test respectively at a node, and $\epsilon$ is a boundary calculated based on the size of sample, and choice of splitting criterion and $\delta$. As the streaming process continues, a conflicting rule $r'$ of rule $r$ might be introduced. To adapt $r'$, the condition $\varDelta\bar{G}_{X_a,X_b} < \epsilon$ must be met to trigger the concept adaptation strategy. Therefore, with confidence $1-\delta$, the size of the smallest change that may cause $r'$ to become superior to $r$ in the CVFDT context is $Tolerance = \varDelta\bar{G}_{X_a,X_b} - \epsilon$. According to Definition 1, the $Tolerence$ measures the rule stability of an inner node. The $TreeTol$ which is the sum of the $Tolerance$ of every inner node in a tree denotes the stability of the tree.

### 3.2 Hoeffding Relational Regression Tree

Our HRRT algorithm upgrades the RRT learner used in RFGB by incorporating methods from CVFDT [7]. It has a test search space including conjunctions of recursive and aggregated predicates by using refinement operator with $\theta$-subsumption and aggregate condition [1]. Most of the algorithm is similar to CVFDT except that HRRT is using learning from interpertation setting, for detailed explanation please refer to CVFDT and HTILDE-RT [5]. We implement the $TreeTol$ in *StabilityCheck* function with a threshold defined by user to periodically check whether the tree qualifies as stable in the sense that the tree is stable when $TreeTol$ is greater than the threshold. HRRT and *StabilityCheck* are the building blocks for the following algorithms.

### 3.3 Relational Incremental Boosting

As shown in Algorithm 1, the initial tree $\psi$ is incrementally learned by HRRT and will be boosted when it has satisfied the *StabilityCheck* (line 9). The empty tree $\eta$ will then be trained to fit on the functional gradients of the boosted tree $\psi$ (lines 5-7) for example in the window data. In the presence of concept drift, the error introduced by the old conflicting rules in $\psi$ will be corrected by $\eta$. When $\eta$ has also met the *StabilityCheck* and the execution signal $S$ is true (line 6), the algorithm adds $\eta$ to $\psi$, and then boosts their sum in that $\eta$ itself is just a functional gradient tree (FGT) of $\psi$, only the composite of them can represent the newly found stable rules.

The execution signal $S$ is set by the *EvalCentre* (line 4) that handles two crucial issues with RIB and the following RBF. One issue is that the complexity of the resulting model grows with the increasing size of learned data. Inspired by DWM [6], we introduce an evaluation centre that periodically evaluates the contribution to error of each FGT in the resulting model, and discards the FGTs that perform poorly over time. In such a way, the model's complexity decreases at the expense of its integrity. Another issue is that if the concept never drifts, new boosted trees will be added to the model constantly but provide trivial

**Algorithm 1** Relational Incremental Boosting

```
procedure RIB($DataStream, p$)
    Initialize empty tree $\psi$ and $\eta$
    for each $d$ in $DataStream$ do
        After every $p$ examples do $\{\psi, S\} \leftarrow EvalCentre(\psi, d)$
        if $\psi.boosted$ then $\eta \leftarrow HRRT(\eta, GradExpGen(\psi, d))$
            if $StabilityCheck(\eta)$ and $S$ then $\psi \leftarrow Boosting(\eta + \psi)$ and reset $\eta$
            end if
        else $\psi \leftarrow HRRT(\psi, d)$
            if $StabilityCheck(\psi)$ then $\psi \leftarrow Boosting(\psi)$, $\psi.boosted \leftarrow True$
            end if
        end if
    end for
    return $(\psi + \eta)$
end procedure
```

improvement in performance. In this case, the *EvalCentre* evaluates the global performance of the model and stops executing boosting by setting the signal $S$ to false when the performance reaches a pre-defined threshold, indicating that strong consistency of the model to the window data is achieved.

### 3.4 Relational Boosted Forest

In this RBF explanation, the model makes predictions based on the weighted average of regression values of trees in the forest with normalized weights. Other prediction models can be applied depending on the scenario. As shown in Algorithm 2, the forest and weights are initialized along with an empty tree $\psi$ and its associated weight $w$ set to 1 (line 2). When $\psi$ has passed the *StabilityCheck* and the execution signal $S$ is true, the boosted $\psi$ and its weight $w$ will be added to the forest and weights respectively (lines 7-8). When a boosted tree makes a mistake in a predictive attempt, the *EvalCentre* in RBF will decrease its weight. The forest is in such a way recursively populated. Each boosted tree contains established rules that co-exist in the forest, and the weights are dynamically tuned to adapt the window data. In response to the complexity problem, the poorly performing trees with a weight less than a pre-defined threshold will be removed from the forest. The signal $S$ is set in the same way as RIB to handle the no drifting concept issue.

## 4 Conclusions and Future Work

In this paper, we have introduced three adaptive incremental learning algorithms: the HRRT, RIB and RBF. All these algorithms can incrementally and adaptively learn the parameters and structure simultaneously for SRL models such as the Relational Dependency Network (RDNs) and the Markov Logic Network (MLNs). The RIB and RBF extend the classical ensemble methods for

**Algorithm 2** Relational Boosted Forest

```
procedure RBF($DataStream, p$)
    Initialize empty $Forest$ & $Weights$, empty tree $\psi$ and $w \leftarrow 1$
    for each $d$ in $DataStream$ do
        After every $p$ examples do
            $\{Forest, Weights, S\} \leftarrow EvalCentre(Forest, Weights, d)$
        $\psi \leftarrow HRRT(\psi, d)$
        if $StabilityCheck(\psi)$ and $S$ then $\psi \leftarrow Boosting(\psi)$
            Add $\psi$ to $Forest$, $w$ to $Weights$ and reset $\psi,\ w \leftarrow 1$
        end if
    end for
    return $\{Forest, Weigths\}$
end procedure
```

the first time to relational scenarios for handling drifting concepts. As they are developed in the RFGB system, RIB and RBF can be naturally integrated with algorithms designed for RFGB system such as the Soft Margin [3] and the Structural EM [4] for modelling imbalanced and incomplete data in an incremental fashion.

In future work, we will evaluate these algorithms on some of the standard SRL benchmark datasets, and compare their performance against each other and with other state-of-the-art online structure learning algorithms.

## References

1. Natarajan, S., Khot, T., Kersting, K., Gutmann, B., Shavlik, J.: Gradient-based boosting for statistical relational learning: The relational dependency network case. Mach. Learn. 86, 25-56 (2012)
2. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Ann. Stat. 29, 1189-1232 (2001)
3. Yang, S., Khot, T., Kersting, K., Kunapuli, G., Hauser, K., Natarajan, S.: Learning from Imbalanced Data in Relational Domains: A Soft Margin Approach. Proc. - IEEE Int. Conf. Data Mining, ICDM. 2015-Janua, 1085-1090 (2015)
4. Khot, T., Natarajan, S., Kersting, K., Shavlik, J.: Gradient-based boosting for statistical relational learning: the Markov logic network and missing data cases. Mach. Learn. 100, 75-100 (2015)
5. Menezes, G., Zaverucha, G.: HTILDE-RT: Scaling up relational regression trees for very large datasets. Inductive Log. Program. (2011)
6. Kolter, J., Maloof, M.: Dynamic Weighted Majority : An Ensemble Method for Drifting Concepts. J. Mach. Learn. Res. 8, 2755-2790 (2007)
7. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 01. pp. 97-106. (2001)
8. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 00. pp. 71-80. (2000)