Inducing Relational Tree-Augmented Naive Bayes (RTAN) Networks: An Application to Learning in Backgammon

Scott P. Sanner
Department of Computer Science
University of Toronto
Toronto, ON M5S 3G4
ssanner@cs.toronto.edu

Abstract

In this research, we look at the task of inducing relational tree-augmented naive Bayes network structure for very large Bayesian networks. By inducing relational Bayes net structure, the search space is considerably reduced over the original propositional Bayes net structure, for which structure search is nearly intractable. Furthermore, this approach leverages structural regularities that may exist in the feature space but that may not be identifiable in a limited amount of training data. We apply our proposed algorithm to the task of learning feature structure for move prediction in Backgammon and show that adding feature structure suggested by our algorithm generally improves performance whereas adding random feature structure actually tends to decrease performance (since this increases the prediction noise due to increased variance).

1 Introduction

Previous work in learning to play Backgammon has shown that naive Bayes feature combination methods can prove quite effective for learning to predict the relative win/loss probability of a set of moves in Backgammon [1]. An extension of this work [2] has shown that a set of expert-crafted features can allow a naive Bayes approach to achieve results approach GrandMaster, on par with the early versions of TD-Gammon [3].

However, while these approaches certainly show that a straightforward technique such as naive Bayes inference can yield excellent results when used with good features, they do not explain how these features were learned in the first place. The purpose of this paper is to propose a general algorithm that can be provided with only the most primitive features in a domain that can effectively infer useful relational feature structure from a set of training data. The applicability of this algorithm will be especially germane when the task of inferring structure over the given propositional domain would be intractable due to domain size.

We will apply our algorithm to the domain of Backgammon, which has a huge primitive feature space (200,000 possible features), and show that the system can effectively learn useful feature structure that substantially increases performance. We compare our results to an algorithm that simply adds random feature structure to demonstrate that any algorithm intended to augment feature structure should choose only those feature augmentations which are likely to offer a substantial gain in predictive performance.

Although it would seem that additional feature structure could never hurt, it is important to remember that we only have a limited amount of training data. Consequently random feature augmentations tend to lead to higher variance, potentially unstable feature predictions that can negatively impact learning performance in the presence of a limited amount of data.

2 Related Work

There has been a variety of recent work on feature induction but most of the work directly addressing this topic is inapplicable to the domain of predicting good moves in Backgammon for a number of reasons.

Viola et al. [4, 5] have researched the topic of learning visual features for enhanced pattern recognition. However, their work primarily involves finding pose-invariant features and it's not clear that there is a direct and straightforward generalization to learning relational structure in general.

Della Pietra et al. [6] present an algorithm that can learn arbitrary features in Markov random fields. While the results of their algorithm are encouraging, the problem with applying this work to large relational models is that *each* potential feature augmentation evaluation requires thousands of Monte Carlo samples and an application of a root finding algorithm to optimize the weight settings. This approach would be very difficult for the size of the training data and the number of potential features we have in Backgammon. Ideally we would just like to cache sufficient statistics of the data and perform very simple calculations for feature evaluation..

Friedman et al. propose two of the main ideas that we will integrate in our approach to feature learning. First, Friedman et al. [7] introduce the Tree-Augmented Naive Bayes (TAN) network structure for learning classifiers. This work is propositional and we extend it to the general case of relational Bayes nets. Second, we leverage the general framework given by Friedman et al. [8] for learning probabilistic relational models. Our work is at the intersection of these two papers, applying TAN learning to a relational network while restricting the structure of a PRM to that for which very efficient search algorithms for optimal structure can be derived.

Although some of the foundations of this work have already been laid out, this paper presents three novel contributions:

1. This paper presents a restriction of learning probabilistic relational models to the TAN structure within features. This is an efficiency optimization that may be necessary to get these algorithms to perform well in practice.
2. This paper presents an example of learning relational structure and using it successfully in a large, delayed reward reinforcement learning (RL) model. So far there have been very few applications of learning probabilistic relational structure and this is one of the first empirical evaluations using this model within an RL framework.
3. This paper is perhaps the first work to offer an explanation as to how explicit, complex feature structures can be learned in Backgammon.[1]

[1]Certainly TD-Gammon had to learn complex structured features to achieve its level of play, but

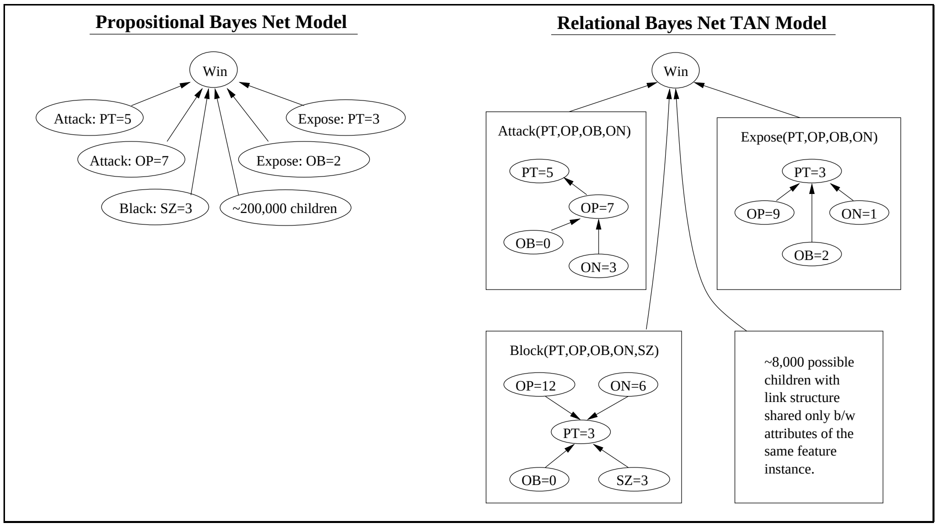

Figure 1: Diagrammatic representation of the propositional Bayes Net structure for the Backgammon feature space and a Relational Tree-Augmented Naive Bayes Net (RTAN) structure for the same feature space. (See Table 1 for the meaning of the feature attributes.) Note that in the RTAN model, the Bayes net links are a template for all feature instantiations (although we disallow links between templates and different template instances.

3 Approach

3.1 Preliminaries

The basic foundation for our learning architecture is given in [1]. The approach taken in this paper is relatively straightforward, however, in brief summary, the learning algorithm is simply naive Bayes where all features are assumed independent and the goal is to select a move that optimizes the *a posteriori* probability of a win given the feature set in a move afterstate. This conditional probability can be computed through two simple applications of Bayes rule and the naive independence assumption. A diagram corresponding to this Bayes network structure is given in the left half of Figure 1.

Since there are 200,000 base primitive features for Backgammon, the naive Bayes approach may seem implausible for this domain. Fortunately however, the move selection only involves a comparison between the features differing between moves. Thus, the probability contribution of all other features that have the same state (either mutually present or not present) cancel along with any noise that may have been present in these features. Empirically, naive Bayes performs very well in Backgammon even for simple feature representations. This is largely due to the low variance inherent in naive Bayes predictions.

3.2 Why do we need RTANs for Backgammon?

In this paper, our goal is to extend this naive Bayes representation to incorporate additional tree structured Bayes (TAN) network links. However, if we apply the TAN-building algorithm given by [7] directly to the current propositional network in the left half of Figure 1, we would need to compute $4 \cdot 10^{10}$ mutual information values. Clearly this is not practically tractable and furthermore, inferring 1,000's of Bayes net links from a limited set of

these feature detectors were encoded in the network weights and therefore not explicitly salient.

data could prove disastrous and highly noise prone. What we need is a learning algorithm that can make use of a limited amount of data to infer links that are of high-relevance and hopefully low variance.

This is where probabilistic relational models (PRMs) prove useful. In PRMS, we want to specify a relational Bayes net template that governs every instance of that template (and in the general case, every instance linked from that template as well). Thus, all templates are guaranteed to share the same structure and we do not need to learn different Bayes net structure for different instances of the same template. If in Backgammon, we consider the base feature types (i.e.. ATTACK, EXPOSE, BLOCK) to be these templates then we only have to learn the tree structure for each template. And furthermore, we can leverage the data from all template instances to determine whether to add a *single* template link thus making link inference robust in the presence of limited data. We call these structures, relational TAN structures or RTANs. An *example* RTAN structure for Backgammon is given in the right half of figure 1.

While one may object that we have already given this system its feature structure, this is not entirely true for two reasons: 1) We have only unified[2] the most basic primitive features in the domain and these primitives would be known to even the most novice Backgammon player. 2) To simply combine all of these attributes and learn the joint feature probability could (and does) typically hurt performance in practice. As mentioned earlier, the increased variance due to sparse data distribution among the feature partitions often helps more than it hurts, so we want to choose our feature structure very carefully.

3.3 RTAN Induction Algorithm

To define our algorithm, we generalize from the TAN induction algorithm presented in [7].

Once we have the feature templates defined, we can cache all of the sufficient statistics for each basic propositional feature and every potential joint propositional feature within a template (we do not consider Bayes net links that can extend over general templates here, although this would not necessarily violate the TAN constraint). Building these sufficient statistics requires time linear in the amount of training data.

Next, we can generalize the log-likelihood score function from [7] to RTAN networks and rewrite it terms of mutual information as is commonly done in the literature. The following equation gives the total joint log-likelihood score of the RTAN network R in terms of a sum over the log likelihood of each template feature T_i (one for each feature i where $1 \leq i \leq |F|$) given the subset of feature sample data D_i.[3]

$$score_L(R : D) = \sum_{i=1}^{|f|} \left(M_i \sum_{j=1}^{|X_i|} MI_P(X_{ij}, Pa_{X_{ij}}) - M_i \sum_{j=1}^{|X_i|} H_P(X_{ij}) \right) \tag{1}$$

Note that we are learning each template individually with the subset of feature data corresponding to that feature class. Then we are summing over the total log likelihood for each

[2] The use of this term from logic is not accidental. In combining features to build these templates, we have essentially performed logical unification over the set of primitive features. There is more to explore here with this concept but this is beyond the scope of this paper. See Poole [9] for some deeper insights into the connection between logical unification and PRMs.

[3] MI is the mutual information function, H is the entropy function, $M_i = |D_i|$ is the number of instance examples for this template, and the probability P being measured is the joint probability of the given nodes and the Win/Loss indicator variable – the joint differs from the intended likelihood for each template by only a normalization constant α. $PA_{X_{ij}}$ indicates the parents of node X_{ij} in the TAN network template T_i being scored where X_i is the set of nodes for template T_i.

feature template to obtain the total joint log likelihood for all feature templates given the total data D. We can do this on account of the naive Bayes independence assumption made between different feature templates.

Given this log likeihood equation, it is now relatively clear how to modify Chow and Liu's algorithm used in [7] to choose the optimal tree structure in the RTAN. The insight is simply that one has to weight the mutual information scores for each template by the amount of data in that template in order to know the order in which to add the links from different templates.

The *modified* Chow and Liu procedure can be defined as follows:

1. Compute the MI_P values for each node pair within each template *weighted* by the number of instances for that template.
2. Build a complete undirected graph within each template, labelling each edge by the MI_P value.
3. Build a max weight spanning tree for this graph (we are allowing tree structure only within templates). This would almost be trivial since each template is a disconnected subgraph *except* that we want to retain the global order of node addition to determine which edges are advantageous to add first. This way when performing capacity control, we can add each additional link to *any* template so long as that link is the one that will most maximize the log likelihood out of the remaining links.

One could additionally modify this algorithm to take into account the MDL score for a naturally motivated approach to capacity control as [7] have done, but we leave this for future work.

4 Experimentation and Results

4.1 Training

We trained the above algorithm on features from 1,000 games of Backgammon between two non-learning stationary policy opponents. This yielded approximately 215,000 features to train on. Once trained, we then collected data on the learning performance of the induced RTAN network playing a stationary policy opponent on randomly selected games different from the training games, adding 1–7 edges that yielded the greatest increase in log likelihood. We also collected data on the performance of an RTAN network (same conditions) with 1–7 randomly added edges to compare adding an edge that most increases the log-likelihood with simply adding any random edge.

4.2 Qualitative Evaluation

We give the weighted mutual information edge values from step 2 of the RTAN induction algorithm for each template in Table 1.

This data matches common intuitions about the utility of joint features. For example, for the EXPOSE feature, the highest mutual information value is for PT (the point on the board at which the expose occurs) and OB (the number of opponents on the bar - if OB ¿ 0 this indicates that the opponent cannot attack until he removes his pieces from the bar). This reflects the intuition that the utility of exposing a piece to be attacked with no opponents on the bar depends on the point being exposed (lower points can be exposed with very little danger even though no opponents on the bar means that the exposed piece could be

Table 1: Mutual information edge values (from step 2 of the RTAN induction algorithm) for each attribute of the three feature classes weighted by the number of samples for each class (ATTACK: $13,600$, EXPOSE: $104,300$, and BLOCK: $99,700$). These values are $\times 1e6$. The attribute abbreviations are: PT - Backgammon board point on which the move occurs (range: 1-24), OP - Number of opponents ahead of point (range: 0-15), OB - Number of opponents on the bar (range: 0-15), ON - Number of opponents ahead within a distance of 7 points (range: 0-15), and SZ - For BLOCK only, the width in points of the block (range: 1-7).

ATTACK	**PT**	**OP**	**OB**
OP	0.1627		
OB	0.1653	0.1540	
ON	0.1629	0.1518	0.1543

EXPOSE	**PT**	**OP**	**OB**
OP	3.3507		
OB	3.4007	3.1855	
ON	3.3612	3.1505	3.1924

BLOCK	**PT**	**OP**	**OB**	**ON**
OP	3.3001			
OB	3.3123	3.1064		
ON	3.3001	3.0942	3.1064	
SZ	2.8888	2.6851	2.6900	2.6851

attacked). However, exposing is entirely safe on any point when OB ¿ 0. Thus, this joint feature can prove more informative than the individual feature.

Other examples of the intuitions behind the mutual information values follow:

- The opponents ahead (OP), the opponents near (ON), and the opponents on the bar (OB) all have a huge impact on the danger of an EXPOSE except when the EXPOSE occurs on a point near the beginning (then they don't matter).
- The BLOCK attribute PT has the most information when considered in conjunction with OB. This is a very useful observation since blocking an opponent on a high point while he is on the bar prevents him from getting off the bar and making any other moves. For low points, the number of opponents on the bar does not matter.
- For the BLOCK feature, the size is *not* the best indicator of utility (low mutual information) since the simple fact that a BLOCK of any size exists is the important part (any pieces in a block are effectively safe).

4.3 Quantitative Evaluation

As mentioned above, we collected data on the running and mean performance of the original naive Bayes classifier, the RTAN version with 1–7 added Bayes net links, and a version with 1–7 randomly added links.

The mean final performance of each of these algorithms training on unseen games is shown in Figure 2. This shows that adding the links that yield the greatest log-likelihood increase generally help learning performance while adding random links can decrease performance.

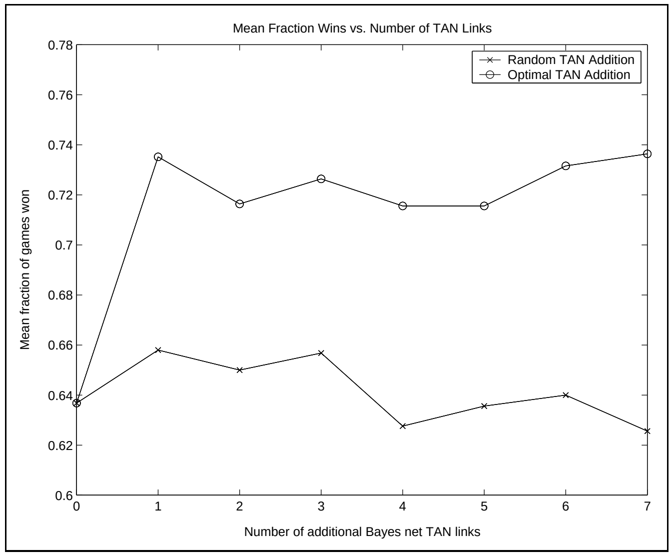

Figure 2: Mean winning percentage during training vs. the number of tree-augmented naive Bayes (TAN) links added to the original representation. Features were added using 1) the optimal link structure as given by the TAN algorithm and 2) randomly selecting a feature to add. Note that random feature additions hurt performance due to the fact that they increase the variance of feature probability estimates while lending very little utility to the predictive capability of the Bayesian net. There is a standard deviation of ± 0.01 for all of these values.

This latter observation is likely due to the high noise/low predictive utility tradeoff in learning the additional structure. See the figure caption for more explanation and discussion.

Figure 3 gives the running performance of a subset of the RTAN networks training against an opponent with a stationary policy on unseen games. This data indicates that it only takes a few RTAN log likelihood maximizing Bayes feature augmentations to substantially increase the performance while subsequent log likelihood maximizing features offer a more modest performance increase. See the figure caption for more explanation and discussion.

5 Conclusions and Future Work

In this paper we have motivated and examined the problem of learning relational tree-augmented naive Bayes (RTAN) network structure and applied it the domain of Backgammon. While this work has shown encouraging results, there is much future work remaining to be done:

- One could explore application of the MDL principle for capacity control to avoid overfitting. Such issues would need to be addressed for on-line algorithms that cannot test performance against independently generated validation data.
- One could explore more general RTAN models that enforce the TAN constraint over links between feature templates. One could also consider adding TAN links between different instances of the same template, e.g. to learn the probability that a joint double ATTACK is better than the product of the assumed indepedent ATTACK probabilities.
- One could allow template attributes to be combined to form a full joint distribution

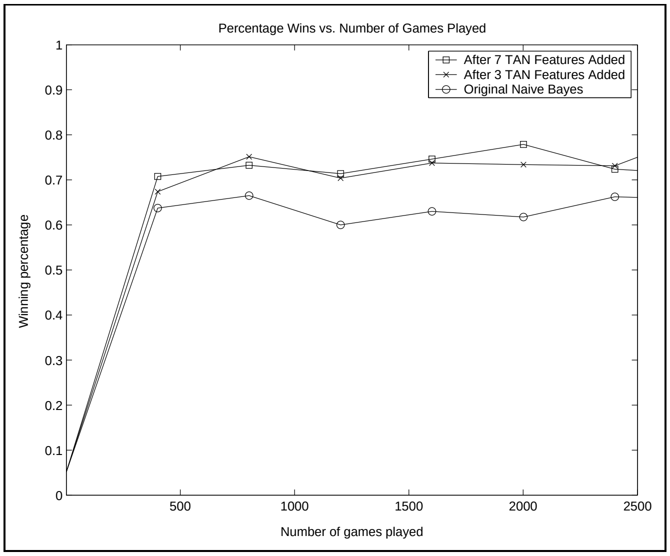

Figure 3: Winning percentage during training for the original naive Bayes representation and the 3 and 7 feature TAN feature representations. Note that the initial features cause the greatest increase in performance, subsequent features have a less substantial impact on performance (although they appear to help a little). Additionally note that good features do not slow the learning curve as the augmented representations outperform the naive Bayes representation both initially and asymptotically. There is a standard deviation of ± 0.025 for all of these values which helps to explain the non-monotonicity in the learning curves.

and then update the RTAN algorithm assuming this new joint feature attribute is primitive. This would allow for augmentations of any size (not just pairwise) while retaining the basic computational efficiency of the RTAN structure.

References

[1] Scott Sanner, John R. Anderson, Christian Lebiere, and Marsha Lovett. Achieving efficient and cognitively plausible learning in backgammon. In *ICML-2000*, pages 823–830, Stanford, 2000.

[2] Scott Sanner. Experimentation in naive bayes for learning in backgammon. In process for publication, 2003.

[3] Gerald Tesauro. Practical issues in temporal difference learning. In John E. Moody, Steve J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 259–266. Morgan Kaufmann Publishers, Inc., 1992.

[4] Paul Viola. Complex feature recognition: A bayesian approach for learning to recognize objects. Technical report, MIT, 1996.

[5] Thomas D. Rikert, Michael J. Jones, and Paul A. Viola. A cluster-based statistical model for object detection. In *ICCV (2)*, pages 1046–1053, 1999.

[6] Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[7] Nir Friedman and Moises Goldszmidt. Building classifiers using bayesian networks. In *AAAI/IAAI, Vol. 2*, pages 1277–1284, 1996.

[8] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.

[9] David Poole. First-order probabilistic inference. In *IJCAI-03*, pages 985–991, 2003.