Learning Class-Discriminative Dynamic Bayesian Networks

John Burge LAWNGUY@CS.UNM.EDU
Terran Lane TERRAN@CS.UNM.EDU
Department of Computer Science, University of New Mexico

Abstract

In many domains, a Bayesian network's topological structure is not known a priori and must be inferred from data. This requires a scoring function to measure how well a proposed network topology describes a set of data. Many commonly used scores such as BD, BDE, BDEU, etc., are not well suited for class discrimination. Instead, scores such as the class-conditional likelihood (CCL) should be employed. Unfortunately, CCL does not decompose and its application to large domains is not feasible. We introduce a decomposable score, *approximate conditional likelihood* (ACL) that is capable of identifying class discriminative structures. We show that dynamic Bayesian networks (DBNs) trained with ACL have classification efficacies competitive to those trained with CCL on a set of simulated data experiments. We also show that ACL-trained DBNs outperform BDE-trained DBNs, Gaussian naïve Bayes networks and support vector machines within a neuroscience domain too large for CCL.

1. Introduction

Our primary contribution is a decomposable Bayesian network scoring function that favors class-discriminative structures and is computationally tractable for large Bayesian networks (BNs). BNs are a graphical modeling framework capable of concisely representing a joint probability distribution (JPD) by taking advantage of independencies among random variables (RVs). They have long been used for their powerful inference capabilities and their ability to model dependencies among RVs. However, in many systems, knowledge of which RVs are correlated is not available a priori. The structure for BNs used to model these systems must be elicited from the data. This requires a scoring function to measure how well a proposed topology describes the RV dependencies within the data. Commonly used scoring methods such as BD (Cooper & Herskovits 1992), BDE (Heckerman, Geiger & Chickering 1995), BDEU (Buntine 1991), etc., are not well suited for class-discriminating since they score proposed structures on how likely the structures are given the data. Highly-likely structures are not necessarily class-discriminative structures. Instead, scoring methods such as the class-conditional likelihood (CCL) should be employed (Grossman & Domingos 2004).

Unfortunately, the CCL score does not decompose into an aggregation of independent scores for separate topological sub-structures of the BN, as most other commonly used scores do. This significantly increases the amount of computation required for structure searches. For instance, in the neuroscience domain we are interested in, structure search using the BDE score can take several hours whereas an equivalent search using CCL could take several months. We introduce a scoring function, *approximate conditional likelihood* (ACL), that both decomposes and identifies class-discriminating structures.

We set up a series of simulated data experiments designed to mimic qualities present in many domains, including our neuroscience domain. Within the data, strong RV correlations exist that are not helpful in class-discrimination. Instead, weaker correlations, whose dynamics change between classes, must be preferred. We show that BDE performs badly under these conditions. We compare the classification efficacies of ACL and CCL in three sets of experiments and find that ACL can identify more subtle differences among classes, CCL is more robust to intra-class noise and the difference in accuracies between the two scores remains relatively constant as the network sizes increase.

Our neuroscience problem is too large for CCL. For this domain, we compare ACL-trained BNs with a set of commonly employed machine learning techniques: BDE-trained BNs, Gaussian naïve Bayesian networks and support vector machines. We find ACL-trained BNs outperform these techniques in classification accuracy and dominates in an ROC cost-analysis.

2. Background

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

2.1 Bayesian networks

For an introductory overview of Bayesian networks (BNs), we refer the reader to the aptly titled "*Bayesian Networks without Tears*" (Charniak 1991). For a more detailed analysis see (Jensen 2001; Heckerman, Geiger & Chickering 1995).

BNs are directed acyclic graphs (DAGs) that explicitly represent independence relationships among RVs. They contain nodes for each RV and a link between any two statistically correlated nodes. The node originating the directed link is a *parent* and the terminating node a *child*. A child and its set of parents are a *family*. Each node contains a conditional probability table (CPT) that describes the relationship between it and its parents.

If the topology is unknown, i.e., the independence relations among RVs is unknown, an appropriate structure must be elicited from the data. This process is referred to as *structure search* and is well understood (Heckerman, Geiger & Chickering 1995) and known to be NP hard (Chickering, Geiger & Heckerman 1994). Structure search boils down to proposing as many hypothesis structures as possible and measuring the goodness of fit between each structure and the data. The method used to measure this fit is a *structure scoring function.*

As there are generally too many structures to score exhaustively, the following heuristic is generally employed. Starting with a topology with no links, iteratively score all legal modifications to the topology. A legal modification is a link addition, removal or reversal that does not result in a cycle. Choose the modification that resulted in the highest score. Repeat until no modifications yield improvements.

The complexity of this algorithm is polynomial in n, the number of nodes, but the degree depends on the score. Decomposable scores can be calculated as the aggregation of independent family scores,

$$Score(B \mid D) = \prod_{i=1}^{n} Score\left(X_i \mid Pa\left(X_i\right)\right). \qquad (1)$$

Modifying the structure or parameters within a single family only affects that family's score. Thus, after modifying the highest scoring family F in the current iteration of a structure search, the scores for the other families in subsequent iterations remain the same. The next search iteration will only have to score new modifications for family F. $\Theta(n^2)$ scores are initially computed and an additional $\Theta(n)$ scores are computed for each iteration. Assuming the number of iterations is proportional to the number of nodes, the algorithm computes $\Theta(n) \cdot \Theta(n) + \Theta(n^2) = \Theta(n^2)$ family scores.

With non-decomposable scores, such as the CCL, modifying family F alters the contribution to the score of every other family. Subsequent iterations must rescore modifications to every family again. In all, $\Theta(n) \cdot \Theta(n^2) + \Theta(n^2) = \Theta(n^3)$ scores must be calculated—an increase in complexity that renders many domains intractable.

2.2 Notation

$\boldsymbol{X}$ represents a set of n fully observable RVs, $\{X_1, X_2, \ldots, X_n\}$ with arities $r_1, r_2, \ldots, r_n$. Y represents the class associated with a given observation. A *data point* consists of fully observable RVs and a class RV: $d = \{\boldsymbol{X}, Y\}$. $\boldsymbol{X_d}$ and Y_d refer to the observable RVs and the class of data point d, respectively. We assume binary classification such that the domain of $Y = \{1, 2\}$. A *dataset*, D, is a collection of m data points, $\{d_1, \ldots, d_m\}$. D_j denotes a dataset containing all of (and only) the data points for a specific class, i.e., $D_j = \{d : Y_d = j\}$.

B denotes a Bayesian network containing nodes for RVs $\{\boldsymbol{X}, Y\}$. The parent set for a RV X_i in B is denoted $\mathrm{Pa}_B(X_i)$ or just $\mathrm{Pa}(X_i)$ if the BN can be inferred. q_i is the number of possible configurations for the RVs in $\mathrm{Pa}(X_i)$. The joint probability distribution for BN B is given by,

$$P_B\left(\boldsymbol{X}, Y\right) = P_B\left(Y \mid Pa\left(Y\right)\right)\prod_{i=1}^{n} P_B\left(X_i \mid Pa\left(X_i\right)\right). \qquad (2)$$

Θ^B is the set of CPT parameters for BN B. Θ_i^B is the CPT for node X_i in BN B, $\Theta_{i,j}^B$ is the multinomial $P_B(X_i \mid \mathrm{Pa}(X_i)=j)$. $\Theta_{i,j,k}^B$ is the CPT element $P_B(X_i = k \mid \mathrm{Pa}(X_i) = j)$.

2.3 Dynamic Bayesian Networks

We are particularly interested in modeling temporal processes via the *dynamic* BN (DBN) representation. For an overview of DBNs, we refer the reader to (Murphy 2002). In the most general case, DBNs include one column of RVs for every time step in the system and one node in each column for every RV in the system. For most real world problems, such DBNs are intractably large. We make the *stationary* and *Markov order 1* assumptions and assume no isochronal links. The topology for these DBNs is composed of two columns, t and $t+1$. The nodes in each column do not represent absolute time points but instead represent behavior of a RV averaged across time. Links are allowed to originate in the left column and terminate in the right (skip to Figure 2 for an example).

Notation for temporal systems is slightly modified. X_i^t (shorthand for X_i^{t+0}) and X_i^{t+1} represents the i^{th} RV in columns t and $t+1$, respectively. The set of DBN RVs is denoted $\boldsymbol{X}^{0:1}=\{X_i^t, X_i^{t+1} : 1 \le i \le n\}$. The parameters for $\mathrm{P}_B(X_i^{t+e} = k \mid \mathrm{Pa}(X_i^{t+e}) = j)$ are denoted $\Theta_{e,i,j,k}^B$, $e \in \{0,1\}$. Y represents the class associated with the *entire* time series.

2.4 Multinets

When using BNs for class discrimination, a single BN with a *class* node is often learned. This is not optimal when the topology depends on the class. A fundamental weakness of the BN framework is that the topology remains static, i.e., the existence of a link cannot change based on the values of RVs. This complicates structure search by introducing redundant parameters and reduces the comprehensibility of the resulting structure, Figure 1.

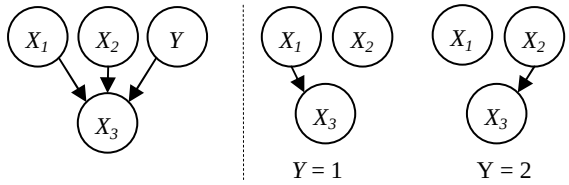

Figure 1. A system in which X_3 is correlated with X_1 *xor* X_2 based on *Y*. (left) The BN required to model the system. The CPT contains $\Theta(r^4)$ parameters where *r* is the arity of the nodes. (right) The pair of multinets required to represent the system. The CPTs contain only $\Theta(r^3)$ parameters.

Instead, separate BNs can be used to model each class, B_1 for class 1 and B_2 for class 2. B_1 and B_2 are referred to as multinets. Dataset D_α is B_α's *intra-class* data and D_β is B_α's *extra-class* data, $\beta \neq \alpha \in \{1,2\}$. An event $\xi^{\beta}_{i,j,k}$ is defined to be the occurrence of a data point $d \in D_\beta$ that has $X_i = k$ and $\text{Pa}(X_i) = j$. The value $\eta^{\beta}_{i,j,k}$ is the count of the number of times event $\xi^{\beta}_{i,j,k}$ occurs.

Work in multinets has been formalized in (Heckerman 1991) and (Geiger & Heckerman 1996). They have also been specifically applied to DBNs in (Bilmes 2000) and used in speech processing (Bilmes et al. 2001).

2.5 Class Conditional Likelihood

If the goal is classification accuracy, choosing the model that maximizes CCL is optimal (Duda & Hart 1973). The CCL structure score is,

$$CCL(B \mid D) = P_B(Y \mid X) = \prod_{d \in D} \frac{P_B(Y_d, \boldsymbol{X}_d)}{P_B(\boldsymbol{X}_d)}. \qquad (3)$$

For DBNs, $\boldsymbol{X_d}$ is replaced by $\boldsymbol{X}_d^{0:1}$. Separating the data into one set for each class, D_1 and D_2, and representing the single BN with two multinets, B_1 and B_2,

$$CCL(B \mid D) = PCCL(B_1 \mid D) \cdot PCCL(B_2 \mid D), \qquad (4)$$

$$PCCL(B_\alpha \mid D) = \frac{\prod_{d \in D_\alpha} P_{B_\alpha}(\boldsymbol{X}_d)}{\prod_{d \in D_\beta} \left[P_{B_1}(\boldsymbol{X}_d) + P_{B_2}(\boldsymbol{X}_d) \right]},$$

where $\beta \neq \alpha \in \{1,2\}$. The sum of probabilities from differing multinets in the denominator prevents CCL from decomposing. Because of this, there is no known closed form solution for computing the parameters that maximize CCL. For a given topology, Greiner and Zhou (2002) have proposed the ELR algorithm based on gradient descent heuristics for computing CCL parameters. But for structure search, this would require a gradient descent for each proposed structure, which was shown to be prohibitively expensive by Grossman and Domingos (2004). They suggest using ML parameters while scoring with CCL and were capable of classifying with higher accuracy than the likelihood-based BD score, as well as several other BN methods.

3. Approximate Conditional Likelihood

CCL is not decomposable. As such, it cannot be applied to datasets with a large number of RVs. Most likelihood-based scores are decomposable but do not favor class-discriminative structures. For discrimination, it is important that the score not rank structures based solely on how well they improve likelihood. Instead, the score should *increase* as the likelihood with respect to the intra-class data *increases*, and *decrease* as the likelihood with respect to the extra-class data *increases*. Such a score will produce networks that discriminate between classes since high-scoring structures will be indicative of relationships that are strong in one class, but weak in another. One such score is the ratio of a multinet's likelihood given its intra-class data and its likelihood given its extra-class data,

$$ACL(B \mid D) = PACL(B_1 \mid D) \cdot PACL(B_2 \mid D), \qquad (5)$$

$$PACL(B_\alpha \mid D) = \prod_{d \in D_\alpha} P_{B_\alpha}(\boldsymbol{X}_d) \Big/ \prod_{d \in D_\beta} P_{B_\alpha}(\boldsymbol{X}_d), \alpha \neq \beta.$$

$PACL(B_\alpha \mid D)$ fully represents the contribution to the overall score for multinet α. It is similar to the *PCCL* term in Equation (4), except the sums in the denominator are now single probabilities. Due to this similarity, we refer to this score as *approximate conditional likelihood* (ACL), though, if ACL is considered as an approximation to CCL, it is an unbounded one.

There are consequences for not including the probability summations in the denominators of the ACL score. If the likelihood of B_1 given D_1 is high, the $P_{B_1}(\boldsymbol{X}_d)$ term in the *PCCL*'s denominator in Equation (4) would lower the $PCCL(B_2 \mid D)$ score. As this term does not exist in the ACL score, the $PACL(B_2 \mid D)$ score is not lowered and may score the proposed structure too highly.

Even so, ACL has significant computational advantages over CCL. It is decomposable while still favoring discriminating structures over high-likelihood ones—a feature important to classification. In addition, closed form solutions for parameters that maximize ACL exist.

3.1 ACL Parameters

Each *PACL* term can be equivalently written in terms of CPT parameters,

$$PACL(B_\alpha \mid D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(\Theta^{\alpha}_{i,j,k} \right)^{\left(\eta^{\alpha}_{i,j,k} - \eta^{\beta}_{i,j,k} \right)}, \qquad (6)$$

subject to $\sum_{j=1}^{r_i} \Theta^{\beta}_{i,j,k} = 1$, $\alpha \neq \beta$.

The derivative of each *PACL* term with respect to a single parameter can be calculated using Lagrangian multipliers (Burge 2005), resulting in the following parameters,

$$\Theta^{\alpha}_{i,j,k} = \begin{cases} \frac{1}{\omega}\left(\eta^{\alpha}_{i,j,k} - \eta^{\beta}_{i,j,k} + \varepsilon \right) & \text{if } (\eta^{\alpha}_{i,j,k} > \eta^{\beta}_{i,j,k}) \\ \varepsilon & \text{otherwise} \end{cases}, \qquad (7)$$

where ω is a normalizing term and ε is a Laplacian smoothing constant. Using these parameters increases the probability mass in a multinet's CPT parameters for events occurring frequently in the intra-class data, but infrequently the extra-class data. It also reduces probability mass in parameters to ε for events that occur more frequently in the extra-class data. However, unlike CCL, maximizing ACL's parameters will not necessarily result in a classification accuracy increase.

While setting parameters with Equation (7) increases the ACL score, it also results in assigning equal probability to all events that are more frequent in the extra-class data. However, it could be that for two such events, one is far less common than the other. This information is restored in the following parameter setting (Burge 2005),

$$\Theta^{\alpha}_{i,j,k} = \frac{1}{\omega}\left[\left(\eta^{\alpha}_{i,j,k} - \eta^{\beta}_{i,j,k}\right) + \min_k\left(\eta^{\alpha}_{i,j,k} - \eta^{\beta}_{i,j,k}\right) + \varepsilon\right]. \qquad (8)$$

The resulting CPT's multinomials will only have a single parameter set to ε, corresponding to the event that occurs most disproportionately often in the extra-class data.

ACL-ML refers to training with maximum likelihood parameters, *ACL-Max* refers to training with the parameters given in Equation (7) and *ACL-Mid* refers to training with parameters given in Equation (8).

3.2 Classification

After DBNs have been learned for each class, a cost function ratio can be used to classify data points,

$$\begin{cases} Y_d = 1 & \text{if } \left(c_1 + c_2 \cdot \ell\left(B_1 \mid d_t\right) + c_3 \cdot \ell\left(B_2 \mid d_t\right) > 0\right) \\ Y_d = 2 & \text{otherwise} \end{cases},$$

where d_t is the testing data point, ℓ is the likelihood function, Y_d is the class assigned to the data point and c_1, c_2 and c_3 are the parameters for the classification boundary in likelihood-space. For BDE and CCL, these parameters are set to $c_1 = 0$, $c_2 = 1$, $c_3 = -1$. For ACL, they must be learned. We used values that minimize the squared error (MSE) between the classification boundary and the training data points. To ensure that ACL did not gain an unfair advantage by having MSE classification boundary parameters, we tested both BDE and CCL with MSE parameters and found no improvement in classification accuracy.

3.3 Multiple Class Classification

In this paper, we limit ourselves to binary classification, though a brief discussion on multiple class classification is warranted. The most straightforward approach—including additional summation terms in the *PACL* denominators for the extra classes' data—is problematic and renders ACL non-decomposable.

Instead, ACL generalizes to the one-versus-many classification paradigm in which each class's extra-class data is the composition of data for all other classes, i.e., the extra class data for class $\alpha = \bigcup_{z \in Z/\alpha} D_z$, where Z is the set of all classes. Just as before, a single DBN would be learned for each class. This results in DBNs that discriminate between a class α and a class γ which represents the composition of all other classes.

Classification would then consist of multiple cost function ratio tests, one for each class. In each test, B_2 would represent the composite class γ. Each ratio test would indicate whether a data point should be classified as class α or as the composite class not including α. It is possible that more than a single ratio test would classify a data point as the non-composite class, i.e., more than a single ratio test would result in $Y_d = 1$. This ambiguity could be resolved in several ways, e.g., by using the cost function ratio with the largest positive value to classify the point.

4. Experiments

4.1 Neuroscience Domain

Functional magnetic resonance imaging (fMRI) has become widely used in the study and diagnosis of mental illness. It is a non-invasive technique measuring the activity of small cubic regions of brain tissue (voxels). Psychologists frequently use fMRI data to test hypotheses about changing neural activity caused by mental illness.

An fMRI scanning session can result in hundreds of 3D images, each consisting of 65,000 or more voxels. As there is too much data collected to analyze directly, we abstract from the voxel level to a *region of interest* (ROI) level. To do this, we use the Talairach database (Lancaster et al. 2000) since it is widely accepted in the neuroscience community. Each 3D image is converted into an activity snapshot detailing the momentary activation of 150 ROIs. A detailed time series is built from the snapshots that accounts for the activity of each ROI. Each ROI is treated as a temporal RV, and the system is modeled with a stationary Markov order 1 DBN containing the nodes $\boldsymbol{X}^{0:1}=\{X_i^t, X_i^{t+1} : 1 \le i \le 150\}$.

We analyze data collected by Buckner et al. (2000) in an experiment theorized to elicit different neural responses from healthy and demented elderly patients. In the original analysis, Buckner et al. found little difference between the two groups using a general linear model. Using BDE-trained DBNs, Burge et al. (2004) found significant differences between groups not identified in the original study. In Section 5.2, we compare the classification efficacy and ROC cost-analysis of BDE-trained BNs, ACL-trained BNs, Gaussian naïve Bayesian networks and support vector machines.

4.2 Simulated Data

The large number of variables in the fMRI domain prohibits learning with CCL. Thus, a comparison of the classification accuracies between ACL and CCL cannot be performed. Instead, we compare ACL and CCL's

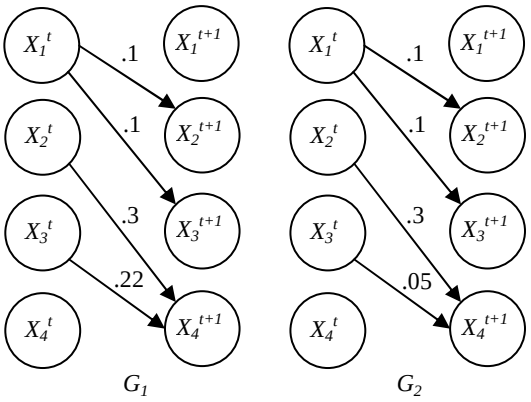

Figure 2. Topology for the two generative DBNs, G_1 and G_2. The number above each link indicates the correlational strength between the nodes. Higher numbers indicate stronger correlations. The only class-discriminating link is $X_3^t \rightarrow X_4^{t+1}$.

classification efficacy on a set of simulated data experiments designed to capture qualities inherent in the fMRI domain (as well as other domains).

Two generative DBNs, G_1 and G_2, are constructed, one for each class of data. The relations among RVs in the generative DBNs can vary in strength. The dynamics of a correlation between a parent and child can change across classes without requiring changes to the dynamics between that same child and its other parents. Changes such as these are present in the fMRI domain. E.g., strong correlations between ROIs *A* and *C* may not differ, but weak correlations between ROIs *B* and *C* may.

Figure 2 illustrates the DBNs used to generate the simulated data. The number above each link indicates the strength of the correlation as a normalized mutual information score (NMIS). The higher the score, the stronger the correlation. The only difference between G_1 and G_2 is the NMIS for link $X_3^t \rightarrow X_4^{t+1}$ is 0.22 in G_1 and 0.05 in G_2. This indicates that $P_{G_1}(X_4^{t+1} \mid X_3^t) \neq P_{G_2}(X_4^{t+1} \mid X_3^t)$. All other links with equal NMIS's indicate that the relationships between parent and child do not change across classes. While the CPT $P(X_4^{t+1} \mid X_2^t, X_3^t)$ changes between the G_1 and G_2 classes, the marginalized CPT, $P(X_4^{t+1} \mid X_2^t)$ does not.

If a non-discriminative score, such as BDE, was used to classify data generated from G_1 and G_2, the highest scoring parent for node X_4^{t+1} would be X_2^t since the NMIS for $X_2^t \rightarrow X_4^{t+1}$ is 0.3 and only 0.22 for $X_3^t \rightarrow X_4^{t+1}$. Thus, DBNs trained using BDE would include the non-discriminating $X_2^t \rightarrow X_4^{t+1}$ link and would be ineffective in class discrimination. The $X_3^t \rightarrow X_4^{t+1}$ link should be favored, even though it corresponds to a weaker correlation. Unlike BDE, both CCL and ACL will identify the correct discriminating link.

The exact method for generating a CPT for node X_i^{t+e} that conforms to a set of NMIS's is outside the scope of this paper. We will refer to it as the distribution $P(\Theta_{e,i}^B \mid S_{e,i}^B)$, where $e = \{0,1\}$, $\Theta_{e,i}^B$ is the set of CPT parameters for node X_i^{t+e} in DBN *B* and $S_{e,i}^B = \{ s_{e,i,1}^B, \ldots, s_{e,i,p}^B \}$ is a list of NMIS's, one NMIS for each of the *p* parents of X_i^{t+e}. E.g., $S_{1,4}^{G_1}$ in Figure 2 = {.3 .22}. A given $\Theta_{e,i}^B$ can also be modified to produce a new CPT, $\Theta_{e,i}'^B$, compliant with a different list of NMIS's, $S_{e,i}'^B$. This generator is referred to as the distribution $P(\Theta_{e,i}'^B \mid \Theta_{e,i}^B, S_{e,i}'^B)$. The closer $S_{e,i}'^B$ is to $S_{e,i}^B$, the smaller the Kullback-Leibler (KL) divergence between multinomials in $\Theta_{e,i}'^B$ and $\Theta_{e,i}^B$ will be. If $S_{e,i}'^B = S_{e,i}^B$, then $\Theta_{e,i}'^B = \Theta_{e,i}^B$.

4.2.1 Diminishing Inter-Class Differences

The first simulated data experiment measures the ability for each score to identify decreasing magnitudes of discriminating behavior. This is an important feature within many domains, as class-discriminating behaviors need not be profound.

The CPTs in G_1 are drawn from their generators given the NMIS scores listed in Figure 2. G_2 is constructed as a copy of G_1, except the $X_3^t \rightarrow X_4^{t+1}$ NMIS is changed to $0.22 - c$, $0 \leq c \leq 0.22$, and the CPT for X_4^{t+1} is drawn from $P(\Theta_{1,4}^{G_2} \mid \Theta_{1,4}^{G_1}, \{0.3\ (0.22 - c)\})$. Since the NMIS for the $X_2^t \rightarrow X_4^{t+1}$ link does not change across classes, the marginal CPT $P(X_4^{t+1} \mid X_2^t)$ also does not change. For classification to be successful, the scoring function must score the $X_3^t \rightarrow X_4^{t+1}$ link higher than other candidate links—including the more strongly correlated $X_2^t \rightarrow X_4^{t+1}$ link. As *c* increases, the KL divergence between G_1's and G_2's X_4^{t+1} CPT also increases, simplifying classification. At $c = 0$ classification is impossible.

4.2.2 Increasing Intra-Class Differences

The second set of experiments is designed to measure a score's tolerance to *intra-class noise*. Like the first set of experiments, this set is designed to mimic traits in the fMRI domain. A demented patient's neural activity may fundamentally differ from that of healthy patients, but it will also differ among other demented patients. This characteristic is manifested in other domains. Take an example from automated speech recognition. There may be distinct differences in how *bat* and *vat* are pronounced, but not everyone pronounces *bat* the same either. Successfully classifying the utterance of *bat* or *vat* requires identification of the differences discriminating the two words, not the differences that exist among separate utterances of the same word.

For this set of simulated data, G_1 and G_2 are treated as each class's *base-line* model. Each data point is generated from a modified version of the base line models. Both G_1 and G_2 are generated as they were in the first experiment with $c = 0.17$ (the classification accuracy for both scoring methods was 100% at this value of *c*). The model for class α's g^{th} generated data point, G_α^g, starts as a copy of G_α. ρ random $< j,k >$ tuples are chosen, $0 \leq \rho \leq 300$, and 0.1 is added to $\Theta_{1,4,j,k}^{G_\alpha^g}$. When $\rho = 0$, there is no intra-class noise. As ρ increases, intra-class differences increase and discrimination is more difficult.

4.2.3 Increasing Network Size

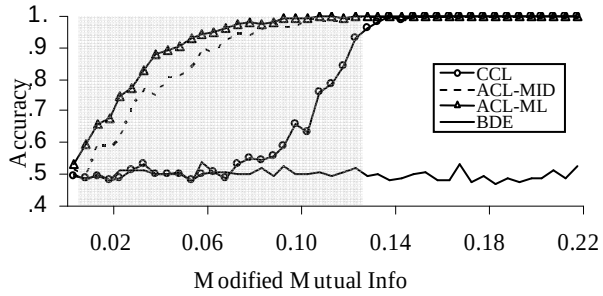

Figure 3. The scores' ability to recognize changes of diminishing magnitudes. As *c* increases, the magnitude of difference between the classes increase, simplifying classification. Gray areas indicate ranges where ACL's accuracy is significantly higher than CCL's ($p = 0.05$).

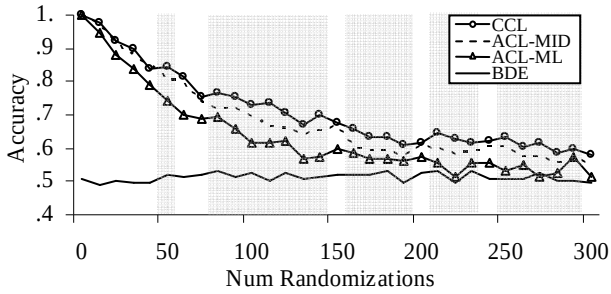

Figure 4. The scores' tolerance to intra-class noise. Gray areas indicate ranges where CCL's accuracy is significantly higher than ACL-Mid's ($p = 0.05$).

This experiment is designed to test how increasing the size of the generative networks changes CCL and ACL's relative accuracies. Simulated data is drawn from randomly generated DBNs with *n* nodes in each of the *t* and *t+1* columns, $4 \leq n \leq 40$. Each node in the *t+1* column was randomly assigned two parents in the *t* column with NMISs of 0.1 and 0.05. A single node is randomly chosen as the class discriminating node. In class 1's generative DBN, the NMIS's for the discriminating node's parents are set to 0.3 and 0.22. In class 2's DBN, they are set to 0.3 and 0.05 (as was done in Figure 2). Intra-class noise (Section 4.2.2) was added to the generated data. Based on the results of the previous two experiments, CCL's accuracy with these settings should be slightly higher than ACL's.

5. Results / Discussion

5.1 Simulated Data

All simulated data experiments consist of the following procedure. CPTs are first drawn for G_1 and G_2. Five training data points, each containing a series of 500 time points, are then drawn for each class. A structure search is performed to locate two new BNs, L_1 and L_2, using the generated data points as training data and one of the following scores: ACL-ML, ACL-Mid, ACL-Max, BDE or CCL. The structure search finds the single highest scoring parent for each X_i^{t+1} node.

Only a single parent is allowed because every score will select the discriminating link as either the first or second link for node X_4^{t+1}. By allowing a single link, we test whether a score favors discriminating links over non-discriminating links. Further, only a single link is needed for class discrimination.

Five testing data points are then generated for each class and classified via a cost function ratio test between L_1 and L_2. Each experiment is run multiple times to determine the average classification accuracy for a fixed *c* or ρ. The *t* test for two dependent samples (Sheskin 2003) was applied for each experiment ($p = 0.05$) to determine if the observed accuracy averages differed significantly. 95% confidence intervals can be computed from the *t* test, but due to the large number of experiments, these intervals are small and have been omitted from Figures 3, 4 and 5 for clarity. In their place, gray regions indicate ranges where the difference between ACL and CCL's accuracy was statistically significant.

5.1.1 Diminishing Inter-Class Differences

Figure 3 illustrates the results of the first set of simulated data experiments. Each point is the average of 100 runs. As expected, BDE-trained DBNs favored the $X_2^t \rightarrow X_4^{t+1}$ link due to its higher NMIS, and could not classify the data. This highlights the need for a discriminative score when high-likelihood structural similarities exist between classes. ACL-Max was insignificantly different from ACL-Mid, and has been omitted from the results.

CCL classified perfectly when $c \geq 0.13$, below which its ability to recognize differentiating behavior between the two classes began to falter. At $c \approx 0.11$, CCL began scoring the $X_3^t \rightarrow X_4^{t+1}$ link the same as the other non-discriminating links, and its accuracy began falling dramatically. By $c \approx 0.08$ CCL's classification accuracy became statistically indistinguishable from guessing.

On the other hand, ACL-ML was able to maintain perfect classification when $c \geq 0.114$ and was able to classify with better than 50% accuracy all the way down to $c = 0.01$. ACL-ML continued scoring the $X_3^t \rightarrow X_4^{t+1}$ link higher than the other links far longer than CCL. However, using the ACL-Mid parameters resulted in a significant drop in accuracy. The difference between the score for the $X_3^t \rightarrow X_4^{t+1}$ link and the scores for non-discriminating links using ACL-Mid was lower than the difference when using ACL-ML. This likely accounts for ACL-Mid's lower accuracy. However, regardless of the parameters chosen, ACL was clearly capable of discerning more subtle differences between classes than CCL was.

5.1.2 Increasing Intra-Class Differences

Figure 4 illustrates the results of adding intra-class noise. Each point is the average of 100 runs. In the presence of increasing differences among data points in the same class, CCL performed significantly better than ACL. As

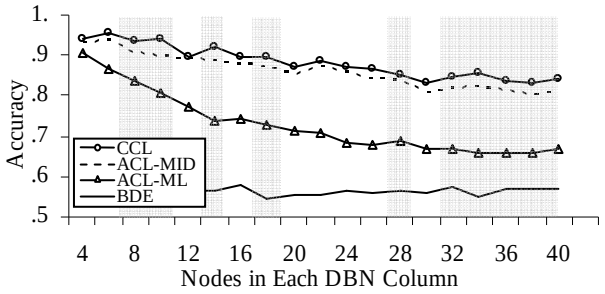

Figure 5. The scores' tolerance to increasing network size. Each point is the average accuracy of 150 runs. Gray areas indicate ranges where CCL's accuracy was significantly higher than ACL-Mid's ($p = 0.05$).

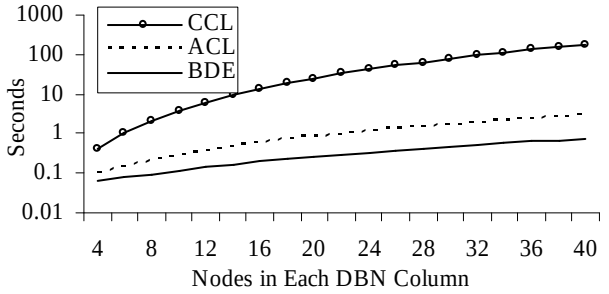

Figure 6. Semi-logarithmic graph of the running times for structure search with an increasing number of nodes. Choice of ACL parameters does not dramatically alter running times.

soon as $\rho > 0$, the accuracy of CCL and ACL-Mid was significantly higher than that of ACL-ML. At $\rho = 80$, CCL's accuracy became significantly higher than ACL-Mid's, and remained higher for most subsequent ρ. Again, BDE was incapable of accurate classification.

ACL-Mid had a significantly higher accuracy than ACL-ML. This is because choosing parameters that increase the ACL score had the effect of increasing the average distance of a data point to the classification boundary in likelihood space. Thus, intra-class noise was less likely to cause a data point to cross the classification boundary and be misclassified.

5.1.3 Increasing Network Size

Figure 5 illustrates the accuracy differences between ACL and CCL as the network size increases. Each point is the average of 150 runs. As expected, CCL's performance was slightly higher than ACL-Mid's. However, as the size of the networks grew, the difference between CCL's and ACL-Mid's accuracy remained essentially constant. This indicates that as the size of the simulated networks increases, ACL remains competitive with CCL.

However, ACL-ML's accuracy dropped dramatically as the network size increased. In Section 5.1.2, ACL-ML was shown to handle intra-class noise worse than ACL-Mid and CCL. We conclude that the increasing amounts of noise in this experiment, in the form of non-discriminating links, paired with ACL-ML's inability to deal with intra-class noise, accounts for its decreased accuracy. Additional experiments (not shown due to space limitations) without intra-class noise have shown that ACL-ML's accuracy does not always drop off as network size increases.

Figure 6 shows the running time of ACL versus CCL. As expected, CCL requires significantly more time than ACL. E.g., for 40-dimensional data, CCL is over 60 times slower than ACL.

5.2 Neuroscience fMRI domain

The DBNs used to model the fMRI domain contained 150 nodes in the *t+1* column, each of which were allowed to have a small number of parents, p. This domain is significantly too large to use CCL. We compare the classification efficacy of ACL-trained DBNs with BDE-trained DBNs, a Gaussian naïve Bayesian network (GNBN) and a support vector machine (SVM) with a linear kernel. Gaussian and quadratic kernels were also employed, but resulted in lower accuracies and their results have been omitted.

Each X_i^{t+1} node's optimal parent set was found independently from the other nodes using the following greedy algorithm. The node is initially set to have no parents. Each X_i^t node is then individually added as a parent, scored, and removed. The X_i^t node with the highest score is then permanently added as a parent. This process is repeated until the node has p parents, or no new parent improves the score. Not all families will be helpful for classification, so only the top κ families with the highest scores are used. All accuracies are computed via leave one out cross-validation and p and κ are found empirically.

The leaders in classification accuracy were ACL-ML and ACL-Mid, achieving an 80% accuracy. BDE and the GNBN achieved 73% accuracy and were the best classifiers reported by Burge et al. The least accurate classifiers were ACL-Max and the SVM, achieving only 70% and 65% accuracies, respectively. As can be seen in the ROC curves given in Figures 7 and 8, ACL-ML virtually dominated all other classification methods in a cost-analysis, only beaten by ACL-Mid in a narrow range.

6. Conclusions

Commonly used BN scoring functions such as BDE are inadequate for purposes of class discrimination. One alternative is to use CCL as a scoring function. However, the application of CCL is limited by its non-decomposable nature to small or mid-sized BNs. We introduced the *approximate conditional likelihood* (ACL) score capable of identifying discriminating structures while remaining decomposable.

We compared the classification efficacy of DBNs trained with CCL and ACL on a class of simulated data that is

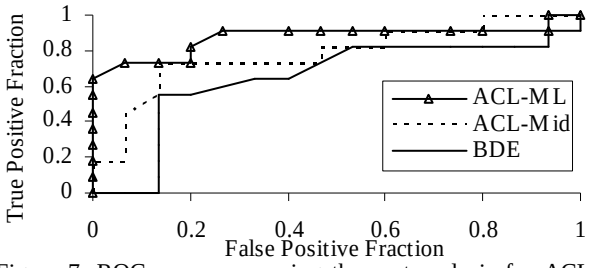

Figure 7. ROC curve comparing the cost-analysis for ACL-ML and ACL-Mid trained BNs, and BDE-trained BNs. A true positive is the correct classification of a demented patient, a false positive is the incorrect classification of a healthy patient as demented. ACL-ML dominates through most of the plot.

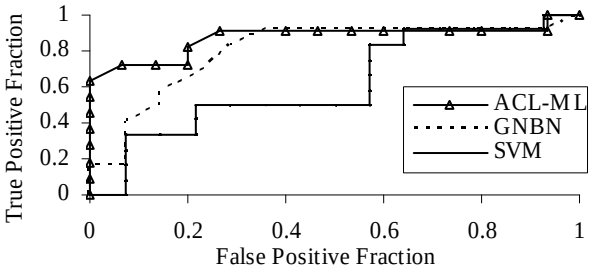

Figure 8. ROC cost-analysis for ACL-ML BNs, the Gaussian naïve Bayesian network and the support vector machine. A true positive is the correct classification of a demented patient, a false positive is the incorrect classification of a healthy patient as demented. ACL-ML dominates.

difficult for non-discriminative scores, such as BDE, to classify. While there was no dominance of one score over the other in all the simulated data tests, CCL was found to be more tolerant to intra-class noise whereas ACL was capable of identifying more subtle differences between classes. Further, as the network sizes increased, the difference in accuracies between ACL and CCL remained essentially constant.

We applied ACL to the neuroscience problem of classifying elderly patients as either healthy or demented based on functional magnetic resonance imaging data. The DBNs used to model this domain are significantly too large for the application of CCL, so we compared ACL's classification accuracies with a host of other commonly used machine learning methods: BDE-trained DBNs, Gaussian naïve Bayesian networks and support vector machines with linear, quadratic and Gaussian kernels. The ACL score effectively identified discriminating structures and achieved the highest observed classification accuracy. Further, from a cost-analysis point of view, the ACL virtually dominated every other algorithm.

Future work involves identifying better methods for training ACL's parameters, as our results showed that ACL-ML and ACL-Mid performed well in different situations. Specifically, we are investigating parameters that maximize the classification margin in likelihood space. We also plan on applying ACL to non-dynamic BNs as there is no fundamental limitation preventing this.

Acknowledgements

We would like to thank Dr. Vincent P. Clark, our collaborating neuroscientist, for his invaluable assistance, Shibin Qiu and Hamilton Link for their work on the SVM and GNBN modeling experiments and the anonymous reviewers that gave us excellent feedback on our paper. This project was partially funded through grant DA012852 from the National Institute of Drug Abuse, NIH, and supported in part by The MIND Institute.

References

Bilmes, J.A. Dynamic Bayesian Multinets. (2000). *Proceedings of the 16th conference on Uncertainty in Artificial Intelligence*. pg. 38-45.

Bilmes, J., Zweig, G., Richardson, T., Filali, K., Livescu, K., Xu, P., Jackson, K., Brandman, Y., Sandness, E., Holtz, E., Torres, J., & Byrne, B. (2001). Discriminatively structured graphical models for speech recognition Tech. Report. Center for Language and Speech Processing, Johns Hopkins Univ., Baltimore, MD.

Buckner, R. L., Snyder, A., Sanders, A., Marcus, R., Morris, J. (2000). Functional Brain Imaging of Young, Nondemented, and Demented Older Adults. *Journal of Cognitive Neuroscience*, 12, 2. 24-34.

Burge, J. (2005) Dynamic Bayesian Networks: Class Discriminative Structure Search with an Application to Functional Magnetic Resonance Imaging Data. Tech Report. University of New Mexico. June, 2005.

Burge, J., Clark, V.P., Lane, T., Link, H., Qiu, S. (2004). Bayesian Classification of FMRI Data: Evidence for Altered Neural Networks in Dementia. In submission to *Human Brain Mapping*.

Buntine, W. (1991). Theory Refinement on Bayesian Networks. *Proceedings of the Seventh Conference on UAI*. 52-60.

Charniak, E. (1991). Bayesian Networks Without Tears. AI Magazine, 12 (4).

Chickering, D., Geiger, D., Heckerman, D. (1994). Learning Bayesian Networks is NP-Hard (Technical Report MSR-TR-94-17). Microsoft.

Cooper, G., Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.

Duda, R. O., Hart, P. E. (1973). Pattern classification and scene analysis. New York, NY: Wiley.

Geiger, D., Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian Multinets. *Artificial Intelligence*. v. 82, pg. 45-74.

Greiner, R., Zhou, W. (2002). Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Proc. 18th Natl. Conf. On Artificial Intelligence*. pg. 167-173.

Grossman, D., Domingos, P. (2004) Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. *International Conference on Machine Learning*, 21. 361-368

Heckerman, D. (1991). Probability Similarity Networks. MIT Press, 1991.

Heckerman, D., Geiger, D., Chickering, D.M. (1995). Learning Bayesian Networks: the Combination of Knowledge and Statistical Data. *Machine Learning*, 20, 197-243.

Jensen, F. V., (2001). Bayesian Networks and Decision Graphs. Springer-Verlag, New York.

Lancaster J.L., Woldorff M.G., Parsons L.M., Liotti M., Freitas C.S., Rainey L., Kochunov PV, Nickerson D., Mikiten S.A., Fox P.T. (2000). Automated Talairach Atlas labels for functional brain mapping. *Human Brain Mapping* 10,120-131.

Murphy, K. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. PhD dissertation, Berkeley, University of California, Computer Science Division.

Pearl, J. (1986) Fusion, Propagation and Structuring in Belief Networks. *Artificial Intelligence*, v. 29, n. 3, 241-288.

Sheskin, D.J. (2003). Handbook of Parameter and Nonparametric Statistical Procedures, 3rd Edition. Chapman & Hall/CRC. QA276.25.S54 2003.