Discriminative Parameter Learning of General Bayesian Network Classifiers

Bin Shen[1]
bshen@cs.ualberta.ca

Xiaoyuan Su[2]
xiaoyuan@ee.ualberta.ca

Russell Greiner[1]
greiner@cs.ualberta.ca

Petr Musilek[2]
musilek@ee.ualberta.ca
[2]*Electrical & Computer Engineering*
University of Alberta
Edmonton, AB, Canada, T6G 2V4

Corrine Cheng[1]
corrine@cs.ualberta.ca
[1]*Computing Science*
University of Alberta
Edmonton, AB, Canada, T6G 2E8

Abstract

Greiner and Zhou [1] presented ELR, a discriminative parameter-learning algorithm that maximizes conditional likelihood (CL) for a fixed Bayesian Belief Network (BN) structure, and demonstrated that it often produces classifiers that are more accurate than the ones produced using the generative approach (OFE), which finds maximal likelihood parameters. This is especially true when learning parameters for incorrect structures, such as Naïve Bayes (NB). In searching for algorithms to learn better BN classifiers, this paper uses ELR to learn parameters of more nearly correct BN structures – e.g., of a general Bayesian network (GBN) learned from a structure-learning algorithm [2]. While OFE typically produces more accurate classifiers with GBN (vs. NB), we show that ELR does not, when the training data is not sufficient for the GBN structure learner to produce a good model. Our empirical studies also suggest that the better the BN structure is, the less advantages ELR has over OFE, for classification purposes. ELR learning on NB (i.e., with little structural knowledge) still performs about the same as OFE on GBN in classification accuracy, over a large number of standard benchmark datasets.

1. Introduction

Many tasks – including pattern recognition and fault diagnosis – can be viewed as classification, as each requires identifying the class labels for instances, each typically described by a set of attributes. Learning accurate classifiers is an active research topic in machine learning and data mining. An increasing number of projects are using Bayesian belief net (BN) classifiers, whose wide use was motivated by the simplicity and accuracy of the naïve Bayes (NB) classifier [3]. While these NB learners find parameters that work well for a fixed structure, it is desirable to optimize structure as well as parameters, towards achieving an accurate Bayesian network classifier.

Most BN learners are *generative,* seeking parameters and structure that maximize *likelihood* [4]. By contrast, logistic regression (LR [5]) systems attempt to optimize the *conditional likelihood* (CL) of the class given the attributes; this typically produces better classification accuracy. Standard LR, however, makes the "naïve bayes" assumption: that the attributes are independent given the class. The discriminative learning tool, ELR (Extended Logistic Regression [1]) extends LR by computing the parameters that maximize CL for *arbitrary* structures, even given *incomplete* training data. [1] shows that ELR often produces better classifiers than generative learners: when the learner has complete data, ELR is often superior to the standard generative approach "Observed Frequency Estimate" (OFE) [6], and when given incomplete data, ELR is often better than the EM [7] and APN [8] systems. ELR appears especially beneficial in the common situations where the given BN-structure is incorrect.

Optimization of BN structure is also an important learning task. Conceivably, optimizing both structure and parameters would be a further improvement, producing BNs that are yet better classifiers. Our paper empirically explores this possibility.

Section 2 reviews the essentials of Bayesian network classifiers. Section 3 introduces Bayesian network learning, focusing on a particular conditional independence (CI) based algorithm for learning BN structure, and the ELR algorithm for learning BN parameters. Section 4 presents our empirical experiments and analyses, based on 25 standard benchmark datasets [9] and the data generated from the Alarm [10] and Insurance [8] networks.

We provide additional details, and additional data, in http://www.cs.ualberta.ca/~greiner/ELR.html.

2. Bayesian (network) classifiers

A Bayesian network (BN) is a probabilistic graph model $B = \langle N, A, \Theta \rangle$, where each network node $n \in N$ represents a random variable and each directed arc $a \in A$ between nodes represents a probabilistic association between variables, forming a directed acyclic graph. Associated with each node $n_i \in N$ is a conditional probability distribution (CPtable), collectively represented by $\Theta = \{\theta_{ij}\}$ which quantifies how much a node depends on its parents [11].

A classifier is a function that assigns a class label to instances, typically described by a set of attributes. Over the last decade or so, Bayesian networks have been used more frequently for classification tasks.

Bayesian classifiers, such as naïve Bayes (NB) classifier, Tree Augmented Naïve-Bayes (TAN) classifier and General Bayesian Networks (GBN) classifier etc. (defined below), are among those effective classifiers, in the sense that their predictive performance is competitive with state-of-the-art classifiers[1].

A naïve Bayes classifier has a simple structure with the class node as the parent of all the attribute nodes; see Figure 1(a). No connections between attribute nodes are allowed in a NB structure. NB is easy to construct and highly effective, especially when the features are not strongly correlated.

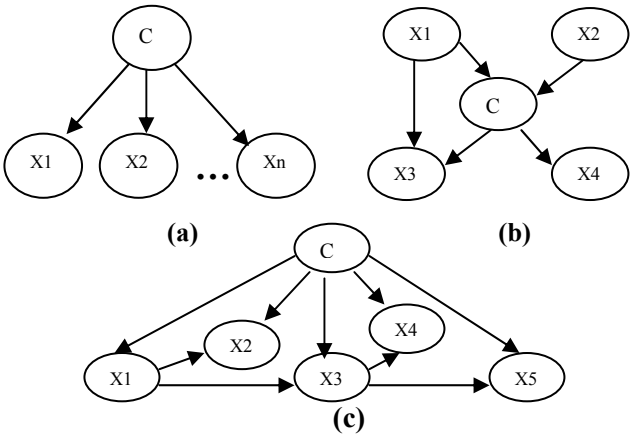


Figure 1. (a) Naïve Bayes (b) General Bayesian Net (c) Tree Augmented Naïve-Bayes

Tree Augmented Naïve-Bayes (TAN) is a natural extension to the naïve Bayes structure that allows some augmenting edges between attributes. Again the class variable C has no parents, and each attribute has the class variable as a parent. Here, however, an attribute can have at most one other attribute as a parent; these attribute-attribute links form a tree (Figure 1(c)). TAN classifiers can be learned in polynomial time by using the Chow-Liu algorithm [12]. TAN classifiers are attractive as they embody a good tradeoff between the quality of the approximation of correlations among attributes, and the computational complexity in the learning stage [9].

General Bayesian Network (GBN) is an unrestricted BN, which treats the class node as ordinary node (Figure 1(b)) – e.g., the class node can also be a child of some attribute nodes. See Section 3.1.1.

[1] There are other Bayesian network classifiers, such as the BN Augmented Naïve-Bayes (BAN) classifier, Bayesian Multi-net classifier, etc.; we will not consider them here.

3. Learning Bayesian networks

There are two major tasks in learning a BN: learning the graphical structure, and learning the parameters (CPtable entries) for that structure.

3.1 Learning Bayesian network structure

Learning structure is a model selection problem in the sense that each structure corresponds to a model (for which parameters have to be estimated) and we need to select a model based on the data.

In general, there are two general classes of approaches to learn the structure of a Bayesian network: conditional independence (CI) based algorithms (using an information theoretic dependency analysis), and search-&-scoring based algorithms [13]. We will focus on the first approach.

3.1.1 CI-based algorithm. The Bayesian network structure encodes a set of conditional independence relationships among the nodes, which suggests the structure can be learned by identifying the conditional independence relationships between the nodes.

Using information theory, the conditional mutual information of two nodes X and Y, with respect to a (possibly empty) conditioning set of nodes C, is defined as [14]:

$$I(X, Y \mid C) = \sum_{x,y,c} P(x, y, c) \log \frac{P(x, y \mid c)}{P(x \mid c) P(y \mid c)}$$

Of course, we do not have access to the true $P(a)$ distribution, but only a training sample S, from which we can compute empirical estimates $P_S(a) \approx P(a)$. We use this to approximate $I(X, Y \mid C)$ as $I_S(X, Y \mid C)$. When this $I_S(X, Y \mid C)$ is smaller than a certain threshold value $\varepsilon > 0$, we say that X and Y are d-separated (conditionally independent) given the condition set C.

Cheng et al. [14] developed a CI-based algorithm for GBN structure learning, the three-phase dependency analysis algorithm, TPDA. The three phases of the TPDA algorithm are *drafting*, *thickening* and *thinning*. The "*drafting*" phase produces an initial set of edges based on pair-wise mutual information, the "*thickening*" (resp., "*thinning*") phases adds (resp.,removes) arcs between

nodes respectively according to results of CI tests, e.g. "Is $I_S(X, Y| C)$ greater than ε?" [14].

This CI-based algorithm is correct (i.e. will produce the perfect model of distribution) given a sufficient quantity of training data D whenever the underlying model is monotone DAG-faithful [14]. This system can be downloaded as part of the Bayesian Belief Network Software package from

http://www.cs.ualberta.ca/~jcheng/bnsoft.htm.

3.2 Learning Bayesian network parameters

We assume there is an underlying distribution $P(.)$ over n (discrete) random variables $N = \{X_1, X_2, .., X_n\}$ (which includes the classification variable – i.e., $C = X_i$ for some i.). For example (just for illustration purpose), perhaps X_1 is the "SARS" random variable, whose value ranges over $\{true, false\}$, X_2 is "visitAsia" $\in \{true, false\}$, X_3 is "bodyTemperature" $\in \{37,...,44\}$, etc. We also assume the probability of asking any "What is $P(C \mid \mathbf{E=e})$?" query corresponds directly to natural frequency of the $\mathbf{E=e}$ event, which means we can infer this from the data sample S; see [15].

Our goal is to construct an effective Bayesian belief net (BN), $B = \langle N, A, \Theta \rangle$ for this classification task. Here, given a node $D \in N$ with immediate parents $\mathbf{F} \subset N$, the parameter $\theta_{d|f}$ represents the network's term for $P(D=d \mid \mathbf{F=f})$ [11].

Given any unlabeled instance, the belief net B will produce a distribution over the values of the query variable, e.g.

$P_B(SARS=true \mid visitAsia = false) = 0.2$

$P_B(SARS=false \mid visitAsia = false) = 0.8$

In general, the associated classifier system H_B will return the highest value:

$H_B(\mathbf{e}) = arg\ max_c\{B(C=c|\mathbf{E=e})\}$

hence, $H_B(visitAsia = false)$ = false.

The goal of learning BN parameters is a Bayesian net that minimizes the classification error of the resulting B-based classifier H_B:

$$err(B) = \sum_{\langle \mathbf{e},c \rangle} P(\mathbf{e},c) \times I(H_B(\mathbf{e}) \neq c)$$

where $I(a \neq b) = 1$ if $a \neq b$, and = 0 otherwise.

The actual parameter learner attempts to optimize the *log conditional likelihood* of a belief net B. Given a sample S, it can be approximated as:

$$\hat{LCL}^{(S)}(B) = \frac{1}{S} \sum_{<e,c> \in S} \log(P_B(c \mid e)) \qquad (1)$$

[16] and [9] note that maximizing this score will typically produce a classifier that comes close to minimizing the classification error. Unfortunately, the complexity of finding the Bayesian network parameters that optimize Equation 1 is NP-hard [15].

For the complete dataset, people often use the Observed Frequency Estimate (OFE) approach that is known to produce the parameters that maximize *likelihood* for a given structure. For example, if 80 of the 100 $C=1$ instances have $X_2=0$, then OFE sets $\theta_{x2=0|C=1}=80/100$. (Some versions use a Laplacian correction to avoid 0/0 issues.) For incomplete datasets, algorithms such as Adaptive Probabilistic Network (APN) [8] or Expectation Maximization (EM) [7] are used to learn the parameters.

3.2.1 Discriminative Parameter Learning Algorithm. Greiner & Zhou implemented a discriminative parameter-learning algorithm, ELR, to maximize the *log conditional likelihood* (Equation 1) [1]. It produces better classifiers than the standard "generative" approach in a variety of situations, especially in common situation where the given BN-structure is incorrect.

Given the intractability of computing the optimal CPtable entries, ELR hill-climbs to improve the empirical score $\hat{LCL}^{(S)}(B)$ by changing the values of each CPtable entry $\theta_{d|f}$ [1]. To incorporate the constraints $\theta_{d|f} \geq 0$ and $\sum_d \theta_{d|f}=1$, we used a different set of parameters: the "softmax" (or "logistic") $\beta_{d|f}$, where

$$\theta_{d|f} = \frac{e^{\beta_{d|f}}}{\sum_{d'} e^{\beta_{d'|f}}}.$$

As the β_is sweep over the reals, the corresponding $\theta_{d_i|f}$'s will satisfy the appropriate constraints.

Given a set of labeled queries, ELR descends in the direction of the total derivative with respect to these queries, which is the sum of the individual derivatives.

Proposition [1]**:** For the tuple (labeled query) $[e;c]$ and each "softmax" parameter $\beta_{d|f}$,

$$\frac{\partial \hat{LCL}^{([e;c])}(B)}{\partial \beta_{d|f}} = [B(d,f \mid e,c) - B(d,f \mid e)] - \theta_{d|f}[B(f \mid c,e) - B(f \mid e)]$$

(Here $B(x)$ refers to probability that B assigns to x.)

For effective learning, parameters are initially set to their maximum *likelihood* values using observed frequency estimates before gradient descent. ELR uses line-search and conjugate gradient techniques, which are known to be effective for LR tasks [17]. Our empirical studies also show that the number of iterations is crucial. We therefore use a type of cross validation (called "cross tuning") to determine this number. ELR also incorporates several other enhancement to speed-up this computation, which leads to significant savings for some problems [1].

4. Empirical Experiments and Analyses

The main focus of this paper is to apply ELR to learn parameters of GBN structures learned from CI-based algorithms and compare (one-sided paired T-tests [18]) its classification performance with several other structure and parameter learning combinations. We evaluated various algorithms over the standard 25 benchmark datasets used

by Friedman et al. [9]: 23 from UCI repository [19], plus "mofn-3-7-10" and "corral", which were developed by [20] to study feature selection. We also used the same 5-fold cross validation and Train/Test learning schemas. As part of data preparation, continuous data are discretized using the supervised entropy-based approach [21].

As mentioned in Section 3, CI-based algorithms can effectively learn GBN structures from complete datasets, provided enough data instances are available. We used Cheng's Power Constructor (which implements the CI-based algorithm TPDA described above) to learn the graphical structure, then applied the ELR parameter optimization to the learned GBN structure. We experimentally compare the results from GBN learning to the results based on the NB and TAN structures, and considered both OFE and ELR parameter learners.

Section 4.1 investigates how the GBN structure (produced by TPDA) compares to NB and TAN for classification purposes, by comparing results of OFE parameter learning on these three classes of structures. Section 4.2 asks "Can ELR learning improve the classification results on GBN, more than the improvements on the relatively incorrect structures NB and TAN?" – does GBN+ELR improve GBN+OFE as much as NB+ELR improves on NB+OFE and TAN+ELR improves on TAN+OFE in classification tasks? Section 4.3 investigates how ELR learning on the (typically incorrect) NB model competes with OFE learning on the better GBN structure produced by TPDA. Empirical classification results over 25 benchmark datasets are listed in Table 1 and Figure 2. Finally, Section 4.4 applies ELR to learn parameters of *correct* structures, to determine if correct structures can further improve ELR learning in classification tasks.

4.1 GBN + OFE vs. NB, TAN + OFE

Given a typically incorrect structure such as NB, OFE can perform poorly [22] in classification tasks. We were therefore surprised when our experimental results (Table 1) showed OFE parameter learning on NB structure (NB+OFE) performed just about the same as GBN+OFE in classification accuracy over the 25 benchmark datasets. A closer look reveals that GBN+OFE did particularly poorly in 5 domains (satimage, segment, soybean-large,

Table 1. Empirical accuracy of classifiers learned from *complete* data

	Data set	GBN+ELR	GBN+OFE	NB+ELR	NB+OFE	TAN+ELR	TAN+OFE
1	australian	**86.81** ± 1.11	86.38 ± 0.98	84.93 ± 1.06	**86.81** ± 0.84	84.93 ± 1.03	84.93 ± 1.03
2	breast	95.74 ± 0.43	96.03 ± 0.50	96.32 ± 0.66	**97.21** ± 0.75	96.32 ± 0.70	96.32 ± 0.81
3	chess	90.06 ± 0.92	90.06 ± 0.92	95.40 ± 0.64	87.34 ± 1.02	**97.19** ± 0.51	92.40 ± 0.81
4	cleve	82.03 ± 1.83	**84.07** ± 1.48	81.36 ± 2.46	82.03 ± 2.66	81.36 ± 1.78	80.68 ± 1.75
5	corral	**100.00** ± 0.00	**100.00** ± 0.00	86.40 ± 3.25	86.40 ± 5.31	**100.00** ± 0.00	93.60 ± 3.25
6	crx	85.69 ± 1.30	86.00 ± 1.94	**86.46** ± 1.85	86.15 ± 1.29	86.15 ± 1.70	86.15 ± 1.70
7	diabetes	**76.34** ± 1.30	75.42 ± 0.61	75.16 ± 1.39	74.77 ± 1.05	73.33 ± 1.97	74.38 ± 1.35
8	flare	82.63 ± 1.28	82.63 ± 1.28	82.82 ± 1.35	80.47 ± 1.03	**83.10** ± 1.29	83.00 ± 1.06
9	german	73.70 ± 0.68	73.70 ± 0.68	74.60 ± 0.58	**74.70** ± 0.80	73.50 ± 0.84	73.50 ± 0.84
10	glass	44.76 ± 4.22	**47.62** ± 3.61	44.76 ± 4.22	**47.62** ± 3.61	44.76 ± 4.22	**47.62** ± 3.61
11	glass2	78.75 ± 3.34	80.63 ± 3.75	**81.88** ± 3.62	81.25 ± 2.21	80.00 ± 3.90	80.63 ± 3.34
12	heart	78.89 ± 4.17	**79.63** ± 3.75	78.89 ± 4.08	78.52 ± 3.44	78.15 ± 3.86	78.52 ± 4.29
13	hepatitis	**90.00** ± 4.24	**90.00** ± 4.24	86.25 ± 5.38	83.75 ± 4.24	85.00 ± 5.08	88.75 ± 4.15
14	iris	92.00 ± 3.09	92.00 ± 3.09	**94.00** ± 2.87	92.67 ± 2.45	92.00 ± 3.09	92.67 ± 2.45
15	letter	81.21 ± 0.55	79.78 ± 0.57	83.02 ± 0.53	72.40 ± 0.63	**88.90** ± 0.44	83.22 ± 0.53
16	lymphography	78.62 ± 2.29	79.31 ± 2.18	86.21 ± 2.67	82.76 ± 1.89	84.83 ± 5.18	**86.90** ± 3.34
17	mofn-3-7-10	**100.00** ± 0.00	86.72 ± 1.06	**100.00** ± 0.00	86.72 ± 1.06	**100.00** ± 0.00	91.60 ± 0.87
18	pima	74.25 ± 2.53	75.03 ± 2.25	**75.16** ± 2.48	75.03 ± 2.45	74.38 ± 2.58	74.38 ± 2.81
19	shuttle-small	97.88 ± 0.33	97.31 ± 0.37	99.12 ± 0.21	98.24 ± 0.30	**99.22** ± 0.20	99.12 ± 0.21
20	vote	95.86 ± 0.78	**96.32** ± 0.84	95.86 ± 0.78	90.34 ± 1.44	95.40 ± 0.63	93.79 ± 1.18
21	*satimage	79.25 ± 0.91	79.25 ± 0.91	85.40 ± 0.79	81.55 ± 0.87	**88.30** ± 0.72	**88.30** ± 0.72
22	*segment	77.40 ± 1.51	77.53 ± 1.50	**89.48** ± 1.11	85.32 ± 1.28	89.22 ± 1.12	89.35 ± 1.11
23	*soybean-large	85.54 ± 0.99	82.50 ± 1.40	90.54 ± 0.54	90.89 ± 1.31	92.86 ± 1.26	**93.39** ± 0.67
24	*vehicle	51.95 ± 1.32	48.52 ± 2.13	64.14 ± 1.28	55.98 ± 0.93	**66.39** ± 1.22	65.21 ± 1.32
25	*waveform-21	65.79 ± 0.69	65.79 ± 0.69	**78.55** ± 0.60	75.91 ± 0.62	76.30 ± 0.62	76.30 ± 0.62

* These benchmark datasets may not have sufficient data instances for CI-based algorithms to construct good GBN structures

vehicle and waveform-21 – e.g., the accuracy rate of GBN+OFE learning on waveform-21 is 0.658 compared to 0.759 from NB+OFE), which severely affected the statistical significance of the overall comparison results. Like all learners, CI-based structure learners are sensitive to the coverage of the sample over the underlying distribution of instances. We noticed that for those 5 domains, GBN+OFE did not perform well, the classification statistics have large standard deviations and some of the median values are quite different from the mean values, which indicate the skewness of the underlying distributions of data. We suspect that this small quantity of instances is not sufficient for TPDA to produce good GBN structures. Therefore, all our comparisons involving GBN in the rest of the analyses will be based on the remaining 20 benchmark datasets. Our studies only related to NB and TAN still involve all the 25 benchmark datasets. We also found the learned GBN structures have a slightly different number of edges from their NB counterparts for the 5 domains that GBN+OFE performed particularly poorly (Table 2).

Table 2. GBN structures vs. NB structures (in terms of the number of edges) for satimage, segment, soybean-large, vehicle and waveform-21 domains

Data set / number of BN edges	GBN	NB
satimage	45	36
segment	24	19
soybean-large	27	35
vehicle	24	18
waveform-21	26	21

GBN+OFE consistently outperforms NB+OFE in classification error over the 20 benchmark datasets at significance $p<0.036$, which indicates GBN is a better structure for classification over NB whenever TPDA has produced a good GBN model; see Figure 3(a). Moreover, GBN+OFE performs about as well as TAN+OFE. (In all scatter plot figures, points below the $\mathbf{y} = \mathbf{x}$ diagonal are datasets for which algorithm **y** achieved better classification results than **x**. Moreover, the error-bars reflect the standard deviation of each dimension; see data in Table 1.)

4.2 GBN+ELR vs. NB, TAN+ELR

OFE parameter training produces better classifiers with better structures; can the same be said of ELR parameter training? Classification results from GBN+ELR, NB+ELR and TAN+ELR (Figure 3(b) and 3(c)) refute this over the 20 benchmark datasets, as we see that GBN+ELR, TAN+ELR and NB+ELR all perform comparably to each other. This suggests that a discriminative learner is more robust against degradation of structures, while structural improvements seems more beneficial to a generative learner. However, we suspect that these 20 benchmark datasets may not be sufficient for TPDA to learn a GBN that is accurate enough to make a difference for ELR parameter training. Of course, this learner was attempting to find a structure that optimized *likelihood*; it would be useful to instead use a learner that sought structures that optimized *conditional likelihood*, and then sought appropriate parameters for that structure, using either ELR or OFE [23].

We also notice the performance gaps in classification error between ELR and OFE *shrink* with better structural models. NB+ELR consistently yields better classification results over NB+OFE at significance $p<0.005$ (Figure 4(a)), TAN+ELR performs a little better than TAN+OFE (but not significantly, as only $p<0.12$) (Figure 4(b)). However, GBN+ELR performs very much the same as GBN+OFE (Figure 4(c)).

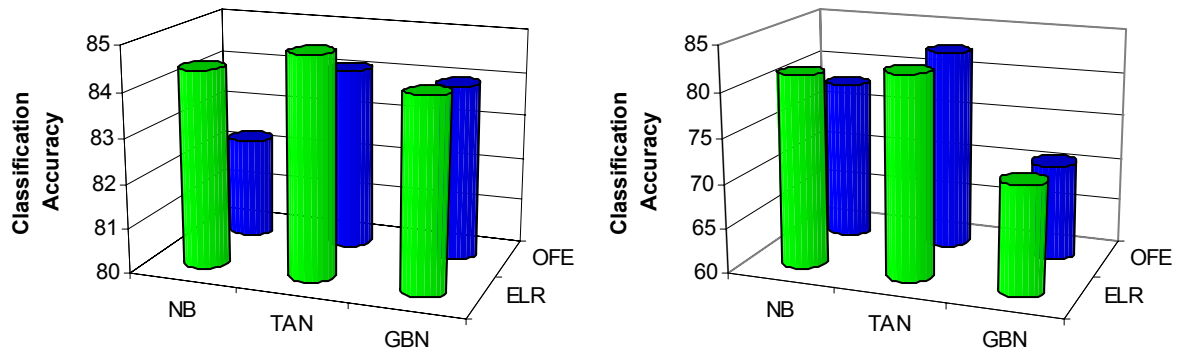
4.3 NB+ELR vs. GBN+OFE

We next investigated how well discriminative parameter training on an incorrect structure competes with a generative parameter training on a more correct structure. Our empirical results show that they are just about the same, with NB+ELR very slightly better than GBN+OFE over the 20 benchmark datasets (Figure 4(d)). This suggests that ELR, even when given essentially no structural knowledge, can compete effectively with generative learner on a much better structure, which is finding the parameters that optimize *likelihood*.

4.4 Correct Model+ELR vs. GBN, NB+ELR

Section 4.2 shows that GBN+ELR does not significantly outperform NB+ELR and TAN+ELR in classification accuracy over 20 benchmark datasets, which suggests that GBNs learned from this CI-based algorithm may not be a good model to improve ELR parameter learning for classification. We designed the following experiments to further evaluate how the ELR learner responds in classification performance given better structural knowledge of the data.

We applied ELR to learn parameters of two correct structures: the ALARM network, a 37 variable (8 query variables, 16 evidence variables) BN for monitoring patients in the intensive care unit [10] and the INSURANCE network, a 27 variable (3 query variables, 12 evidence variables) BN for evaluating car insurance risks [8]. Complete datasets with all variables are sampled from the original networks [24]. We generated queries from these datasets by fixing one query variable and including all the evidence variables (all the other variables were removed) – e.g., a query generated from the ALARM network will include 17 variables (one query



(a) (b)

Figure 2. Average classification accuracy from various algorithms (a) results from 20 benchmark datasets (b) results from satimage, segment, soybean-large, vehicle and waveform-21

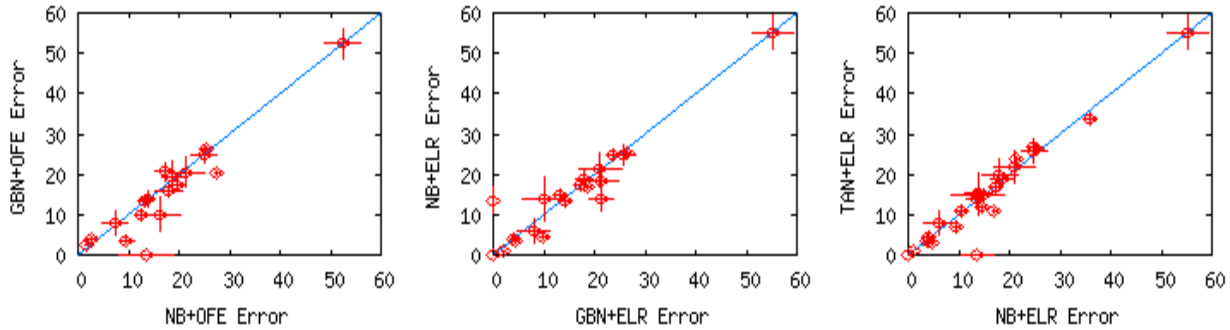


Figure 3. Comparing (a) GBN+OFE vs. NB+OFE over 20 benchmark datasets (b) GBN+ELR vs. NB+ELR over 20 benchmark datasets (c) TAN+ELR vs. NB+ELR over 25 benchmark datasets

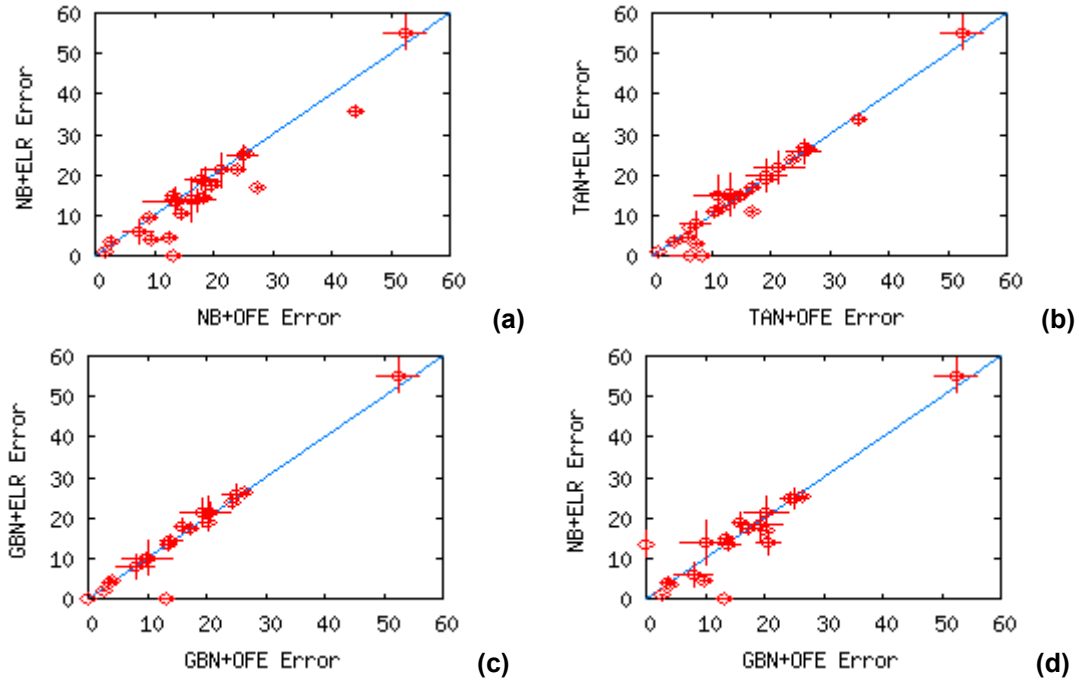


Figure 4. Comparing (a) NB+ELR vs. NB+OFE over 25 benchmark datasets (b) TAN+ELR vs. TAN+OFE over 25 benchmark datasets (c) GBN+ELR vs. GBN+OFE over 20 benchmark datasets (d) NB+ELR vs. GBN+OFE over 20 benchmark datasets

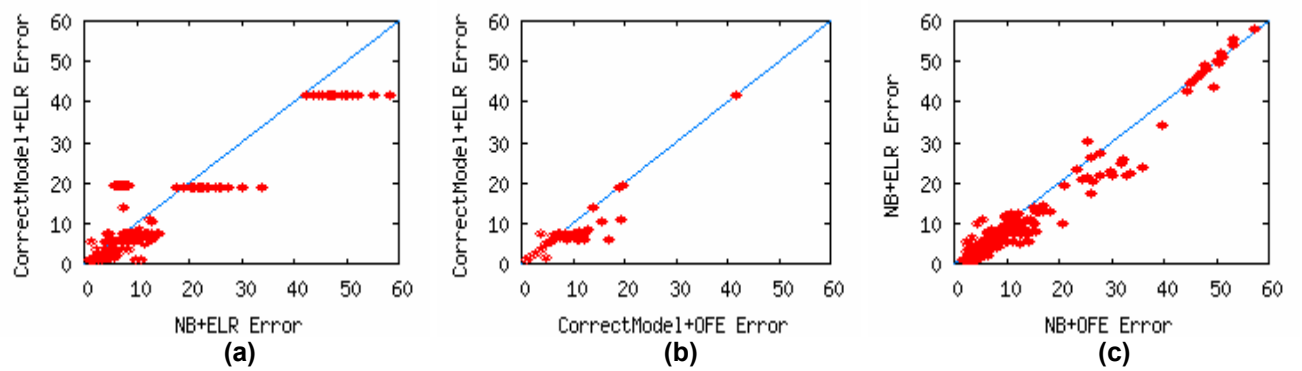


Figure 5. Comparing algorithms in learning correct structures (a) CorrectModel+ELR vs. NB+ELR (b) CorrectModel+ELR vs. CorrectModel+OFE (c) NB+ELR vs. NB+ OFE (from experimental results on 99 sample queries generated from 11 query forms – 8 on ALARM and 3 on INSURANCE with sample size: 20, 30, 40, 50, 60, 70, 80, 90 and 100)

variable and 16 evidence variables). All query variables (8 from ALARM and 3 from INSURANCE) were considered in generating queries – for each query form corresponding to one query variable, 9 training sample queries were generated with sizes from 20 to 100 (step 10); therefore, there were 99 training samples overall for experiments in each run here. Using each training sample, we applied ELR and OFE to learn parameters of the true structure, and the NB, TAN, GBN constructed from the same sample. The resulting systems were then evaluated based on a 3000 tuple testing dataset computed analytically from the true model. Here the GBN was first learned from a 500 tuple training dataset (with all variables) sampled from the true model. When learning parameters of the GBN using the sample from a particular query form, the original GBN was truncated to exclude those variables that were not presented in that training sample, we are aware this truncated GBN is only an approximation of the original learned GBN.

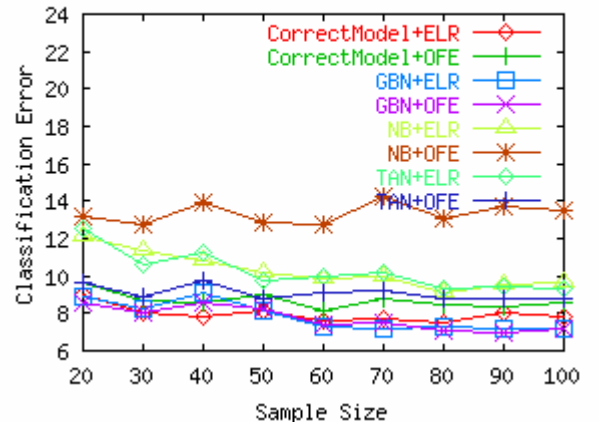


Figure 6. Alarm domain: Comparing competing algorithms among CorrectModel+ELR, CorrectModel+OFE, GBN+ELR, GBN+OFE, NB+ELR, NB+OFE, TAN+ELR, TAN+OFE

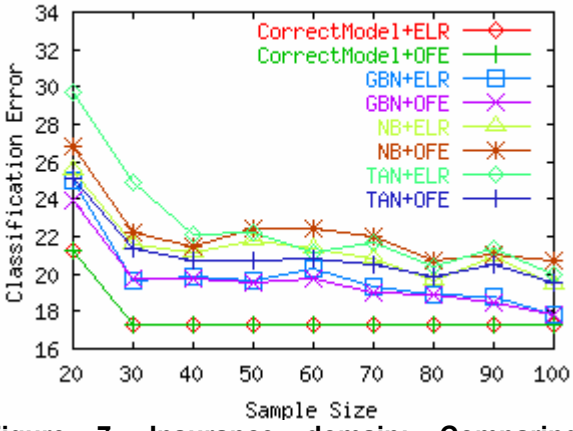


Figure 7. Insurance domain: Comparing competing algorithms among CorrectModel+ELR, CorrectModel+OFE, GBN+ELR, GBN+OFE, NB+ELR, NB+OFE, TAN+ELR, TAN+OFE

Figure 5(a) shows CorrectModel+ELR outperforms NB+ELR consistently at significance $p<0.02$. With a correct structure, ELR still produces better classification accuracies than OFE at $p<0.001$ (Figure 5(b) in Alarm domain), while NB+ELR has an even better performance than NB+OFE at $p<0.000001$ here, indicating better structures will lower the advantage of ELR over OFE in classification tasks.

Figure 6 and 7 show that not only the *correct* structures, but also the GBNs help ELR find better parameters for classification tasks. It is important to note GBNs here were constructed from the data that accurately represented the underlying distribution as they were generated from the *true* models and sampled uniformly.

5. Conclusions and Future Work

These results suggest it would be helpful to find structures that maximize classification accuracy or conditional likelihood [25], especially when the data are

insufficient for generative structure learning. While our experiments dealt with complete data cases, further studies are needed to learn GBN+ELR on incomplete/partial training datasets.

Contributions: This paper demonstrates how the classification performance of discriminative parameter learning responds to a better structural model – here the GBN structure learned using a CI-based algorithm that identifies the independence relationships, which corresponds to the *generative* task of directly modeling the underlying probability distribution. We compared classification results on GBN, NB and TAN structures using both ELR and OFE parameter estimators. We showed that as the structure improves, in terms of better modeling of the underlying probability distribution, a generative parameter learner such as OFE will produce better classifiers; and OFE becomes more competitive compared to ELR for classification purposes. Our empirical studies of ELR learning on *correct* models reveal ELR can be greatly enhanced in classification performance given *true* structures or GBNs constructed from the training data that are good representation of the underlying distribution; considering in the real world, often the data obtained for classification do not cover the good portion of the *true* distribution, this suggests with insufficient training data, a structure optimizing *CL* rather than *likelihood* may have better chances to improve ELR parameter learning for classification purposes. Therefore, we also suspect one of the reasons why GBN is not helping ELR much in classification tasks under many circumstances is that ELR and CI-based algorithms are trying to optimize different objective functions.

Most of datasets and experimental results for this paper can be downloaded from:

http://www.cs.ualberta.ca/~bshen/elr.htm.

More comparative study results for evaluating classifiers based on probability estimation in terms of *conditional likelihood* and area-under-ROC-curve *AUC* [26] will be posted onto the above website.

Acknowledgement

We thank Dr. J. Cheng (Siemens Corporate Research), W. Zhou (U. of Waterloo), Dr. H. Zhang (U. of New Brunswick), X. Wu (U. of Alberta) and Dr. P. Domingos (U. of Washington) for the invaluable discussions and their generous help. RG was supported by NSERC and the Alberta Ingenuity Center for Machine Learning.

References

[1] R. Greiner and W. Zhou. Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. *AAAI*, 2002.

[2] J. Cheng and R. Greiner. Learning Bayesian Belief Network Classifiers: Algorithms and System, *CSCS101*, 2001.

[3] R. Duda and P. Hart. *Pattern classification and scene analysis*. New York, NY: Wiley, 1973.

[4] B. Ripley. *Pattern Recognition in Intelligence Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.

[5] M. Jordan. Why the logistic function? A tutorial discussion on probabilities and neural networks, 1995.

[6] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from the data. *Machine Learning*, 9:309-347, 1992.

[7] D. Heckerman. A Tutorial on Learning with Bayesian Networks. *Learning in Graphical Models*, 1998.

[8] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213-244, 1997.

[9] N. Friedman, D. Geiger and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning* 29:131-163, 1997.

[10] I. Beinlich, H. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *AIME-89,* pages 247-256. Springer Verlag, Berlin, 1989.

[11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[12] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 14:462-467, 1968.

[13] D. Heckerman. A Bayesian Approach to Learning Causal Networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 285-295), Morgan Kaufmann, August 1995.

[14] J. Cheng, R. Greiner and J. Kelly et al. Learning Bayesian network from data: An information-theory based approach. *Artificial Intelligence Journal* 137:43-90, 2002.

[15] R. Greiner, A. Grove and D. Schuurmans. Learning Bayesian Nets that Perform Well. *UAI97*, August 1997.

[16] P. McCullagh and J. Nelder. *Generalized Linear Models.* Chapman and Hall, 1989.

[17] T. Minka, Algorithms for maximum-likelihood logistic regression. *Statistics Technical Report 758*, CMU, 2001.

[18] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[19] UCI Repository of Machine Learning Databases and Domain Theories, 1995.
http://www.ics.uci.edu/~mlearn/MLRepository.html

[20] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:1–2, 1997.

[21] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, (pp. 1022–1027), 1993.

[22] P. Domingo and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. *Proc.13th International Conference on Machine Learning*, 1996.

[23] A. Ng and M. Jordan. On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naïve Bayes, *NIPS 15*, 2003.

[24] E. Herskovits and C. Cooper. Algorithms for Bayesian belief-network precomputation. In *Methods of Information in Medicine* (pp. 362–370), 1991.

[25] D. Grossman and P. Domingos. Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood, Department of Computer Science and Engineering, University of Washington, *Technical Report*, 2003.

[26] C. Ling, J. Huang, and H. Zhang. AUC: a Statistically Consistent and more Discriminating Measure than Accuracy. *Proceedings of IJCAI 2003.*