Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers

Russell Greiner
Dept of Computing Science
University of Alberta
Edmonton, AB T6G 2H1 Canada
greiner@cs.ualberta.ca

Wei Zhou
Dept of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1, Canada
w2zhou@math.uwaterloo.ca

Abstract

Bayesian belief nets (BNs) are often used for classification tasks — typically to return the most likely "class label" for each specified instance. Many BN-learners, however, attempt to find the BN that maximizes a different objective function (viz., likelihood, rather than classification accuracy), typically by first learning an appropriate graphical structure, then finding the maximal likelihood parameters for that structure. As these parameters may not maximize the classification accuracy, "discriminative learners" follow the alternative approach of seeking the parameters that maximize *conditional likelihood* (CL), over the distribution of instances the BN will have to classify. This paper first formally specifies this task, and shows how it relates to logistic regression, which corresponds to finding the optimal CL parameters for a naïve-bayes structure. After analyzing its inherent (sample and computational) complexity, we then present a general algorithm for this task, ELR, which applies to arbitrary BN structures and which works effectively even when given the incomplete training data. This paper presents empirical evidence that ELR works better than the standard "generative" approach in a variety of situations, especially in common situation where the BN-structure is incorrect.

Keywords: (Bayesian) belief nets, Logistic regression, Classification, PAC-learning, Computational/sample complexity

1 Introduction

Many tasks require producing answers to questions — *e.g.*, identifying the underlying fault from a given set of symptoms in context of expert systems, or proposing actions on the basis of sensor readings for control systems. An increasing number of projects are using "(Bayesian) belief nets" (BN) to represent the underlying distribution, and hence the stochastic mapping from evidence to response.

When this distribution is not known *a priori*, we can try to *learn* the model. Our goal is an *accurate* BN — *i.e.*, one that returns *the correct answer as often as possible*. While a perfect model of the distribution will perform optimally for any possible query, learners with limited training data are unlikely to produce such a model; moreover, this is impossible for learners constrained to a restricted range of possible distributions that excludes the correct one (*e.g.*, instantiations of a given BN-structure).

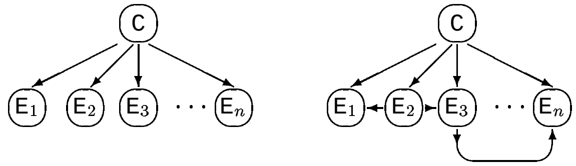

Figure 1: (a) NaïveBayes structure; (b) TAN structure

Here, it makes sense to find the parameters that do well with respect to the queries posed. This "discriminative learning" task differs from the "generative learning" that is used to learn an overall model of the distribution [Rip96]. Following standard practice, our discriminative learner will seek the parameters that maximize the *conditional likelihood* (CL) over the data, rather than simple likehood — that is, given the data $\{\langle c_i, \mathbf{e}_i\rangle\}$, we will try to find parameters Θ that maximize $\sum_i \log P_\Theta(\, c_i \,|\, \mathbf{e}_i\,)$, rather that the ones that maximize $\sum_i \log P_\Theta(\, c_i, \mathbf{e}_i\,)$ [Rip96].

Optimizing the CL of the root node of a *naïve-bayes* structure can be formulated as a standard logistic regression problem [MN89; Jor95]. General belief nets extend naïve-bayes-structures by permitting additional dependencies between the attributes. This paper provides a general discriminative learning tool ELR that can learn the parameters for an arbitrary structure, even given incomplete training data. It also presents empirical evidence, from a large number of datasets, that demonstrates that ELR works effectively.

Section 2 provides the foundations — overviewing belief nets, then defining our task: discriminative learning the parameters (for an arbitrary fixed belief net structure) that maximize CL. Section 3 formally analyses this task, providing both sample and computational complexity; we also note how this compares with corresponding results for generative learning. Seeing that our task is NP-hard in general, Section 4 presents a gradient-descent discriminative learning algorithm for general BNs, ELR. Section 5 reports empirical results that demonstrate that our ELR often performs better than the standard learning algorithms (which maximize likelihood), over a variety of situations: In particular, when the learner has complete data, we show that ELR is often superior to the standard "observed frequency estimate" (OFE) approach [CH92], and when given partial data, we show ELR is often superior to the EM [Hec98] and APN [BKRK97] systems. We also demonstrate that the ELR is especially useful in the common situation where the given BN-structure is incorrect. [GZ02] provides the proofs of the theorems, as well as a comprehensive literature survey.

2 Framework

We assume there is a stationary underlying distribution $P(\,\cdot\,)$ over n (discrete) random variables $\mathcal{V} = \{V_1, \ldots, V_n\}$; which we encode as a "(Bayesian) belief net" (BN) — a directed acyclic graph $B = \langle \mathcal{V}, A, \Theta \rangle$, whose nodes $\mathcal{V}$ represent variables, and whose arcs A represent dependencies. Each node $D_i \in \mathcal{V}$ also includes a conditional-probability-table (CPtable) $\theta_i \in \Theta$ that specifies how D_i's values depend (stochastically) on the values of its parents. In particular, given a node $D \in \mathcal{V}$ with immediate parents $\mathbf{F} \subset \mathcal{V}$, the parameter $\theta_{d|\mathbf{f}}$ represents the network's term for $P(\,D\!=\!d \mid \mathbf{F}\!=\!\mathbf{f}\,)$ [Pea88].

The user interacts with the belief net by asking *queries*, each of the form "$P(\,C\!=\!c \mid \mathbf{E}\!=\!\mathbf{e}\,) = ?$" where $C \in \mathcal{V}$ is a single "query variable", $\mathbf{E} \subset \mathcal{V}$ is the subset of "evidence variables", and c (resp., $\mathbf{e}$) is a legal assignment to C (resp., $\mathbf{E}$). We will focus on the case where all queries involve the same variable; *e.g.*, all queries ask about `Cancer` (but see the ALARM example in Section 5.6).

Following standard practice, we will assume there is a single distribution from which we can draw instances that correspond to queries with their labels, and let $sq(\,c\,;\,\mathbf{e}\,)$ be the probability of the unlabeled query $\mathbf{e}$ being asked and c being the response. ([GGS97] explains the need to distinguish $sq(\,\cdot\,)$ from $P(\,\cdot\,)$; see also Section 5.6.)

Given any unlabeled query $\{\mathbf{E}_i = \mathbf{e}_i\}$, the belief net B will produce a distribution over the values of the query variable; our associated H_B classifier system will then return the value $H_B(\mathbf{e}) = \text{argmax}_c\{B(C\!=\!c|\mathbf{E}\!=\!\mathbf{e})\}$ with the largest posterior probability.

A good belief net classifier is one that produces the appropriate answers to these unlabeled queries. We will use "classification error" (aka "0/1" loss) to evaluate the resulting B-based classifier H_B

$$\text{err}(\,B\,) \;=\; \textstyle\sum_{\langle \mathbf{e},c\rangle} sq(\,\mathbf{e}\,;\,c\,) \times \mathcal{I}(\,H_B(\mathbf{e}) \neq c\,) \quad (1)$$

where $\mathcal{I}(\,a \neq b\,) = 1$ if $a \neq b$, and $= 0$ otherwise.[1]

Our goal is a belief net B^* that minimizes this score, with respect to the query+response distribution $sq(\,\cdot\,;\,\cdot\,)$. While we do not know this distribution *a priori*, we can use a sample drawn from this sq distribution, to help determine which belief net is optimal. (This sq-based "training data" is the same data used by other classifiers.) This paper focuses on the task of learning the optimal CPtable Θ for a given BN-structure $G = \langle \mathcal{V}, A \rangle$.

Conditional Likelihood: Our actual learner attempts to optimize a slightly different measure: The "(empirical) log conditional likelihood" of a belief net B is

$$\text{LCL}_{sq}(\,B\,) \;=\; \textstyle\sum_{\langle \mathbf{e},c\rangle} sq(\,\mathbf{e}\,;\,c\,) \times \log(B(\,c \mid \mathbf{e}\,)) \quad (2)$$

Given a sample S, we can approximate this as

$$\widehat{\text{LCL}}^{(S)}(\,B\,) \;=\; \tfrac{1}{|S|} \textstyle\sum_{\langle \mathbf{e},c\rangle \in S} \log(\,B(\,c \mid \mathbf{e}\,)\,) \quad (3)$$

[MN89; FGG97] note that maximizing this score will typically produce a classifier that comes close to minimizing the classification error (Equation 1). Note also that many research projects, including [BKRK97], use this measure when evaluating their BN classifiers.

While this $\widehat{\text{LCL}}^{(S)}(\,B\,)$ formula closely resembles the "log likelihood" function

$$\widehat{\text{LL}}^{(S)}(\,B\,) \;=\; \tfrac{1}{|S|} \textstyle\sum_{\langle \mathbf{e},c\rangle \in S} \log(\,B(\,c, \mathbf{e}\,)\,) \quad (4)$$

used by many BN-learning algorithms, there are some critical differences. As noted in [FGG97],

$$\widehat{\text{LL}}^{(S)}(\,B\,) = \frac{1}{|S|}\Big[\sum_{\langle c,\mathbf{e}\rangle \in S} \log(\,B(\,c \mid \mathbf{e}\,)\,) + \sum_{\langle c,\mathbf{e}\rangle \in S} \log(\,B(\,\mathbf{e}\,)\,)\Big]$$

where the first term resembles our $\text{LCL}(\,\cdot\,)$ measure, which measures how well our network will answer the relevant queries, while the second term is irrelevant to our task. This means a BN B_α that does poorly wrt the first "$\widehat{\text{LCL}}^{(S)}(\,\cdot\,)$-like" term may be preferred to a B_β that does better — *i.e.*, if $\widehat{\text{LL}}^{(S)}(\,B_\alpha\,) < \widehat{\text{LL}}^{(S)}(\,B_\beta\,)$, while $\widehat{\text{LCL}}^{(S)}(\,B_\alpha\,) > \widehat{\text{LCL}}^{(S)}(\,B_\beta\,)$.

3 Theoretical Analysis

How many "labeled queries" are enough — *i.e.*, given any values $\epsilon, \delta > 0$, how many labeled queries are needed to insure that, with probability at least $1 - \delta$, we are within ϵ to optimal? While we believe there are general comprehensive bounds, our specific results require the relatively benign technical restriction that all CPtable entries must be bounded away from 0. That is, for any $\gamma > 0$, let

$$\mathcal{BN}_{\Theta \succeq \gamma}(G) \;=\; \{\, B \in \mathcal{BN}(G) \mid \forall \theta_{d|\mathbf{f}} \in \Theta,\ \theta_{d|\mathbf{f}} \geq \gamma \,\} \quad (5)$$

be the subset of BNs whose CPtable values are all at least γ; see [NJ01]. We now restrict our attention to these belief nets, and in particular, let

$$B^*_{G,\Theta > \gamma} \;=\; \text{argmax}_B\{\text{LCL}_{sq}(\,B\,) \mid B \in \mathcal{BN}_{\Theta \succeq \gamma}(G)\} \quad (6)$$

be the BN with optimal score among $\mathcal{BN}_{\Theta \succeq \gamma}(G)$ with respect to the true distribution $sq(\,\cdot\,)$.

Theorem 1 *Let G be any belief net structure with K CPtable entries $\Theta = \{\theta_{d_i|\mathbf{f}_i}\}_{i=1..K}$, and let $\hat{B} \in \mathcal{BN}_{\Theta \succeq \gamma}(G)$ be the BN in $\mathcal{BN}_{\Theta \succeq \gamma}(G)$ that has maximum* empirical *log conditional likelihood score (Equation 3) with respect to a sample of*

$$M_{\gamma,K}(\epsilon, \delta) \;=\; O\left(\frac{K}{\epsilon^2} \ln(\frac{K}{\epsilon\delta})\ \log^3(\frac{1}{\gamma})\right)$$

labeled queries drawn from $sq(\,\cdot\,)$. Then, with probability at least $1 - \delta$, $\hat{B}$ will be no more than ϵ worse than $B^*_{G,\Theta > \gamma}$. ∎

[1] When helpful, we will also consider mean squared error: $MSE(B) = \sum_{\langle \mathbf{e},c\rangle} sq(\,\mathbf{e}\,;\,c\,) \times [B(c|\mathbf{e}) - P(\,c \mid \mathbf{e}\,)]^2$.

A similar proof show that this same result holds when dealing with $\text{err}(\,\cdot\,)$ rather than $\text{LCL}(\,\cdot\,)$.

This PAC-learning [Val84] result can be used to bound the learning rate — *i.e.*, for a fixed structure G and confidence term δ, it specifies how many samples M are required to guarantee an additive error of at most ϵ — note the $O(\frac{1}{\epsilon^2}[\log\frac{1}{\epsilon}])$ dependency.

As an obvious corollary, observe that the sample complexity is polynomial in the size (K) of the belief net even if the underbound γ is exponentially small $\gamma = O(1/2^N)$.

For comparison, Dasgupta [Das97, Section 5] proves that

$$O\left(\frac{n\,K}{\epsilon^2}\,\ln(\frac{K}{\epsilon\,\delta})\,\ln^3(n)\,\ln^2(\frac{1}{\epsilon})\right) \tag{7}$$

complete tuples are sufficient to learn the parameters to a fixed structure that are with ϵ of the optimal *likelihood* (Equation 4). This bound is incomparable to ours for two reasons: First, as noted above, the parameters that optimize (or nearly optimize) likelihood will not optimize our objective of *conditional* likelihood, which means Equation 7 describes the convergence to parameters that are typically inferior to the ones associated with Equation 1, especially in the unrealizable case; see [NJ01]. Second, our Equation 1 includes the unavoidable γ term.[2] Nevertheless, ignoring this γ, our asymptotic bound is a factor of $O(n\ln^3(n)\ln^2(1/\epsilon))$ smaller; we attribute this reduction to the fact that our conditional-likelihood goal is more focused than Dasgupta's likelihood objective.[3]

The second question is computational: How hard is it to find these best parameters values, given this sufficiently large sample. Unfortunately...

Theorem 2 *It is NP-hard to find the values for the CP-tables of a fixed BN-structure that produce the smallest (empirical) conditional likelihood (Equation 3) for a given sample.*[4] *This holds even if we consider only BNs in $\mathcal{BN}_{\Theta\succeq\gamma}(G)$ for $\gamma = O(1/N)$.* ■

By contrast, note that there is an extremely efficient algorithm for the generative learning task of computing the parameters that optimize simple *likelihood* from complete data; see OFE, below. (Although the algorithms for optimizing likelihood from *incomplete* data are all iterative.)

4 Learning Algorithm

Given the intractability of computing the optimal CPtable entries, we defined a simple gradient-descent algorithm, ELR, that attempts to improve the empirical score $\widehat{\text{LCL}}^{(S)}(\,B\,)$ by changing the values of each CPtable entry $\theta_{d|\mathbf{f}}$. To incorporate the constraints $\theta_{d|\mathbf{f}} \geq 0$ and $\sum_d \theta_{d|\mathbf{f}} = 1$, we used a different set of parameters — "$\beta_{d|\mathbf{f}}$" — where each

$$\theta_{d|\mathbf{f}} \quad = \quad \frac{e^{\beta_{d|\mathbf{f}}}}{\sum_{d'} e^{\beta_{d'|\mathbf{f}}}} \tag{8}$$

As the β_is sweep over the reals, the corresponding $\theta_{d_i|\mathbf{f}}$'s will satisfy the appropriate constraints. (In the naïve-bayes case, this corresponds to what many logistic regression algorithms would do, albeit with different parameters [Jor95]: Find α, χ that optimize $P_{\alpha,\chi}(\,C = c\,|\,\mathbf{E} = \mathbf{e}\,) = e^{\alpha_c+\chi_\mathbf{c}\cdot\mathbf{e}}/\sum_j e^{\alpha_j+\chi_\mathbf{j}\cdot\mathbf{e}}$. Recall that our goal is a more general algorithm — one that can deal with *arbitrary* structures.)

Given a set of labeled queries, ELR descends in the direction of the total derivative wrt these queries, which of course is the sum of the individual derivatives:

Lemma 3 *For the labeled query* $[\mathbf{e}; c]$, $\frac{\partial\,\widehat{LCL}^{([\mathbf{e};c])}(\,B\,)}{\partial\,\beta_{d|\mathbf{f}}} = \theta_{d|\mathbf{f}}\,[B(\,\mathbf{f}\,|\,c,\,\mathbf{e}\,) - B(\,\mathbf{f}\,|\,\mathbf{e}\,)] \;-\; [B(\,d, \mathbf{f}\,|\,\mathbf{e},\,c\,) - B(\,d, \mathbf{f}\,|\,\mathbf{e}\,)]$.

Our ELR also incorporates several enhancement to speedup this computation. First, we use line-search and conjugate gradient [Bis98]; Minka [Min01] provides empirical evidence that this is one of the most effective techniques for logistic regression. Another important optimization stems from the observation that this derivative is 0 if D and $\mathbf{F}$ are *d*-separated from $\mathbf{E}$ and C — which makes sense, as this condition means that the $\theta_{d|\mathbf{f}}$ term plays no role in computing $B(\,c\,|\,\mathbf{e}\,)$. We can avoid updating these parameters for these queries, which leads to significant savings for some problems [GZ02].

5 Empirical Exploration

The ELR algorithm takes, as arguments, a BN-structure $G = \langle\mathcal{V}, A\rangle$ and a dataset of labeled queries (aka instances) $S = \{\langle\mathbf{e}_i, c_i\rangle\}_i$, and returns a value for each parameter $\theta_{d|\mathbf{f}}$. To explore its effectiveness, we compared the $\text{err}(\,\cdot\,)$ performance of the resulting Θ_{ELR} with the results of other algorithms that similarly learn CPtable values for a given structure.

We say the data is "complete" if every instance specifies a value for every attribute; hence "$E_1 = e_1, \ldots, E_n = e_n$" is complete (where $\{E_1, \ldots, E_n\}$ is the full set of evidence variables) but "$E_2 = e_2, E_7 = e_7$" is not. When the data is complete, we compare ELR to the standard "observed frequency estimate" (OFE) approach, which is known to produce the parameters that maximize likelihood (Equation 4) for a given structure [CH92]. (*E.g.*, if 75 of the 100 $C = 1$ instances have $X_3 = 0$, then OFE sets $\theta_{X_3=0|C=1} = 75/100$. Some versions use a Laplacian correction to avoid $0/0$ issues.) When the data is incomplete, we compare ELR

[2] Unfortunately, we cannot use the standard trick of "tilting" the empirical distribution to avoid these near-zero probabilities [ATW91]: Our task inherently involves computing *conditional* likelihood, which requires *dividing* by some CPtable values, which is problematic when these values are near 0. This also means our proof is *not* an immediate application of the standard PAC-learning approaches. See [GZ02].

[3] Of course, this comparison of upper bounds is only suggestive. Note also that our bound deals only with a single query variable; in general, it scales as $O(k^2)$ when there are k query variables.

[4] The class of structures used to show hardness are more complicated than the naïve-bayes and TAN structures considered in the next sections. Moreover, our proof relies on *incomplete* instances (defined below). While we do not know the complexity of finding the optimal-for-CL parameters for naïve-bayes structures given complete instances, the fact that there are a number of *iterative* algorithms here (for the equivalent task of logistic regression [Min01]) suggests that it, too, is intractable.

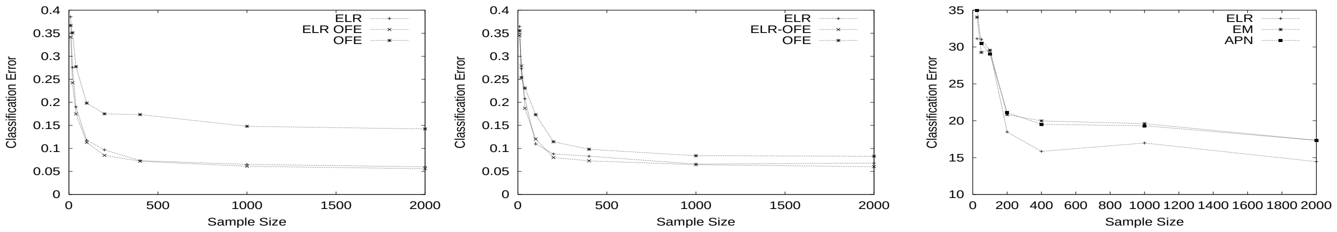

Figure 2: CHESS domain: (a) ELR vs OFE, complete data, structure is "Incorrect" (naïve-bayes); (b) ELR vs OFE, complete data, structure is "Correct" (POWERCONSTRUCTOR); (c) ELR vs EM,APN, complete data, structure is "Incorrect" (naïve-bayes)

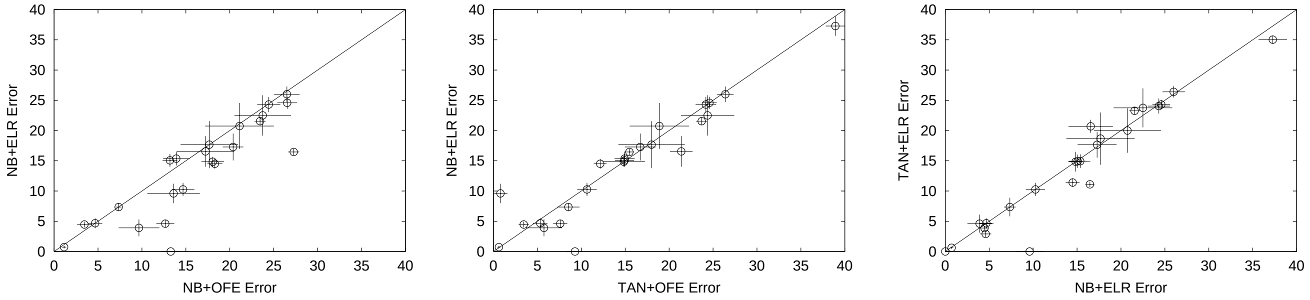

Figure 3: Comparing NB+ELR with (a) NB+OFE (b) TAN+OFE (c) TAN+ELR

to EM [Hec98] and APN [BKRK97],[5] which descends to parameter values whose likelihood is locally optimal.

This short article only reports on a few experiment, to illustrate the general trends; see [GZ02] for an exhaustive account of our experiments. Here we present only the results of the ELR$=$ELR$_\beta$ algorithm, which used the β terms (Equation 8), as we found its performance strictly dominated the ELR$_\theta$ version which used θ directly

5.1 NaïveBayes — Complete, Real World Data

Our first experiments dealt with the simplest situation: learning the NaïveBayes parameters from complete data — which corresponds to standard logistic regression [NJ01].[6] Recall that the NaïveBayes structure requires that the attributes are independent given the class label; see Figure 1(a).

Here, we compared the relative effectiveness of ELR with various other classifiers, over the same 25 datasets that [FGG97] used for their comparisons: 23 from UCIrvine repository [BM00], plus "MOFN-3-7-10" and "CORRAL", which were developed by [KJ97] to study feature selection. To deal with continuous variables, we implemented supervised entropy discretization [FI93]. Our accuracy values were based on 5-fold cross validation for small data, and holdout method for large data [Koh95]. See [GZ02],[FGG97] for more information about these datasets.

We use the CHESS dataset (36 binary or ternary attributes) to illustrate the basic behaviour of the algorithms. Figure 2(a) shows the performance, on this dataset, of our NB+ELR ("NaïveBayes structure + ELR instantiation") system, versus the "standard" NB+OFE, which uses OFE to instantiate the parameters. We see that ELR is consistently more accurate than OFE, for any size training sample. We also see how quickly ELR converges to the best performance. The ELR-*OFE* line corresponds to using OFE to initialize the parameters, then using the ELR-gradient-descent. We see this has some benefit, especially for small sample sizes.

Figure 3(a) provides a more comprehensive comparison, across all 25 datasets. (In each of these scatter-plot figures, each point below the $x = y$ line is a dataset where NB+ELR was better than other approach — here NB+OFE. The lines also express the 1 standard-deviation error bars in each dimension.) As suggested by this plot, NB+ELR is significantly better than NB+OFE at the $p < 0.005$ level (using a 1-sided paired-t test [Mit97]).

5.2 TAN — Complete, Real World Data

We next considered TAN ("tree augmented naïve-bayes") structures [FGG97], which include a link from the classification node down to each attribute and, if we ignore those class-to-attribute links, the remaining links, connecting attributes to each other, form a tree; see Figure 1(b).

Figure 3(b) compares NB+ELR to TAN+OFE. We see that ELR, even when handicapped with the simple NB structure, performs about as well as OFE on TAN structures. Of course, the limitations of the NB structure may explain the poor performance of NB+ELR on some data. For example, in the CORRAL dataset, as the class is a function of four interrelated attributes, one must connect these attributes to predict the class. As NaïveBayes permits no such connection, NaïveBayes-based classifiers performed poorly on this data. Of course, as TAN allows more expressive structures, it has a significant advantage here. It is interesting to note that our NB+ELR is still comparable to TAN+OFE, in general.

Would we do yet better by using ELR to instantiate TAN structures? While Figure 3(c) suggests that TAN+ELR is

[5] While the original APN$_\theta$ [BKRK97] climbed in the space of parameters θ_i, we instead used a modified APN$_\beta$ system that uses the β_i values (Equation 8), as we found it worked better.

[6] While the obvious tabular representation of the CPtables involves more parameters than appear in this logistic regression model, these extra BN-parameters are redundant.

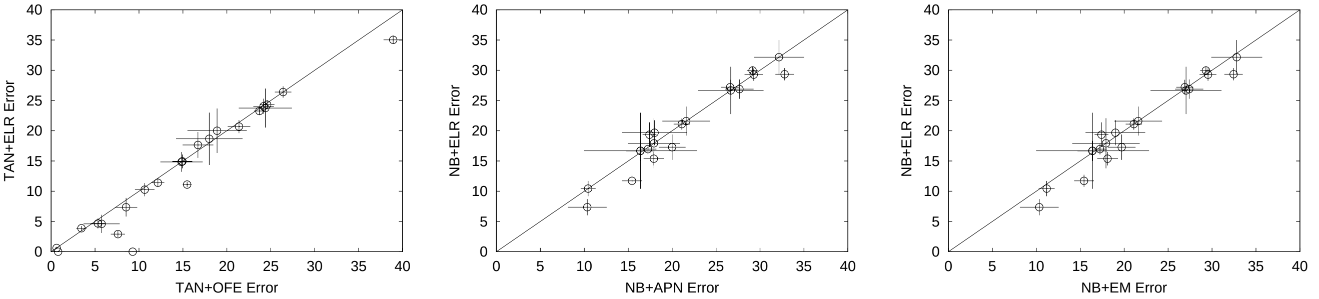

Figure 4: (a) Comparing TAN+ELR vs TAN+OFE;(b,c) Incomplete Data: Comparing NB+ELR with (b) NB+APN; (c) NB+EM

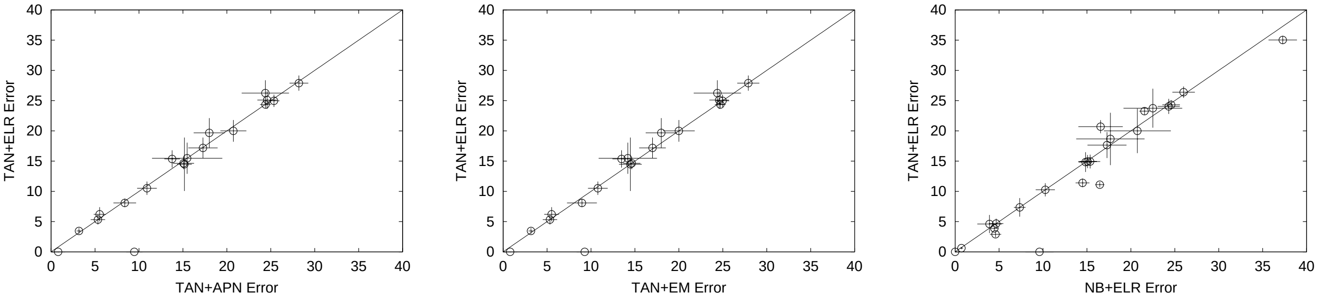

Figure 5: Incomplete data: Comparing TAN+ELR with (a) TAN+APN; (b) TAN+EM; (c) NB+ELR

slightly better than NB+ELR, this is not significant (only at the $p < 0.2$ level). However, Figure 4(a) shows that TAN+ELR does consistently better than TAN+OFE — at a $p < 0.025$ level. We found that TAN+ELR did perfectly on the the CORRAL dataset, which NB+ELR found problematic.

5.3 NB, TAN — Incomplete, Real World Data

All of the above studies used *complete* data. We next explored how well ELR could instantiate the NaïveBayes structure, using *incomplete* data.

Here, we used the datasets investigated above, but modified by randomly removing the value of each attribute, within each instance, with probability 0.25. (Hence, this data is missing completely at random, MCAR [LR87].) We then compared ELR to the standard "missing-data" learning algorithms, APN and EM. In each case — for ELR, APN and EM — we initialize the parameters using the obvious variant of OFE that considers only the records that include values for the relevant node and all of its parents.

Here, we first learned the parameters for the NaïveBayes structure; Figure 2(c) shows the learning curve for the CHESS domain, comparing ELR to APN and EM. We see that ELR does better for essentially any sample size.

We also compared these algorithms over the rest of the 25 datasets; see Figures 4(b) and 4(c) for ELR vs APN and ELR vs EM, respectively. As shown, ELR does consistently better — in each case, at the $p < 0.025$ level.

We next tried to learn the parameters for a TAN structure. Recall the standard TAN-learning algorithm computes the mutual information between each pair of attributes, conditioned on the class variable. This is straightforward when given complete information. Here, given *incomplete* data, we approximate mutual information between attributes A_i and A_j by simply ignoring the records that do not have values for both of these attributes. Figures 5(a) and 5(b) compare TAN+ELR to TAN+APN and to TAN+EM. We see that these systems are roughly equivalent: TAN+ELR is perhaps slightly better than TAN+EM (but only at $p < 0.1$), but it is not significantly better than TAN+APN. Finally, we compared NB+ELR to TAN+ELR (Figure 5(c)), but found no significant difference.

5.4 "Correctness of Structure" Study

The NaïveBayes-assumption, that the attributes are independent given the classification variable, is typically incorrect. This is known to handicap the NaïveBayes classifier in the standard OFE situation [DP96].

We saw above that ELR is more robust than OFE, which means it is not as handicapped by an incorrect structure. We designed the following simple experiment to empirically investigate this claim.

We used synthesized data, to allow us to vary the "incorrectness" of the structure. Here, we consider an underlying distribution P_0 over the $k + 1$ binary variables $\{C, E_1, E_2, \ldots, E_k\}$ where (initially)

$$P(\,+C\,) = 0.9 \quad P(\,+E_i \mid +C\,) = 0.2 \quad P(\,+E_i \mid -C\,) = 0.8 \tag{9}$$

and our queries were all complete; *i.e.*, each instance of the form $\vec{\mathbf{E}} = \langle \pm E_1, \pm E_2, \ldots, \pm E_k \rangle$.

We then used OFE (resp., ELR) to learn the parameters for the NaïveBayes structure from a data sample, then used the resulting BN to classify additional data. As the structure was correct for this P_0 distribution, both OFE and ELR did quite well, efficiently converging to the optimal classification error.

We then considered learning the CPtables for this NaïveBayes structure, but for distributions that were *not* consistent with this structure. In particular, we formed the $m\!-\!th$ distribution P_m by asserting that $E_1 \equiv E_2 \equiv \ldots \equiv$

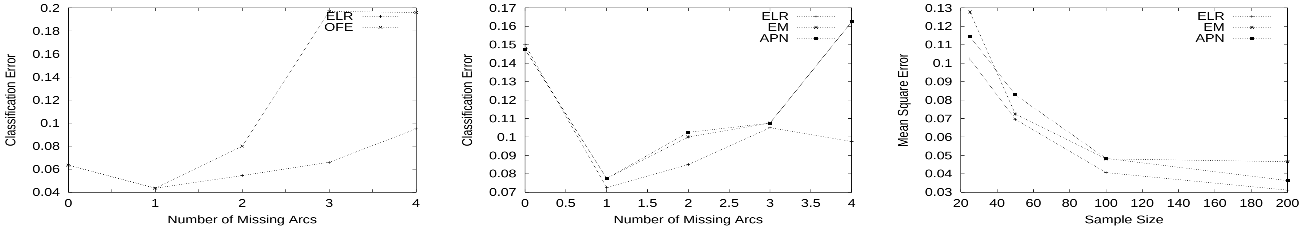

Figure 6: (a,b) Comparing ELR to OFE, on increasingly incorrect structures for (a) Complete Data; (b) Incomplete Data; (c) Using range of query values, and "incomplete" data on ALARM;

E_m (*i.e.*, $P(+E_i \mid +E_1) = 1.0$, $P(+E_i \mid -E_1) = 0.0$ for each $i = 2..m$) in addition to Equation 9. Hence, P_0 corresponds to the $m = 0$ case. For $m > 0$, however, the m-th distribution cannot be modeled as a NaïveBayes structure, but could be modeled using that structure augmented with $m - 1$ links, connecting E_{i-1} to E_i for each $i = 2..m$.

Figure 6(a) shows the results, for $k = 5$, based on 400 instances. As predicted, ELR can produce reasonably accurate CPtables here, even for increasingly wrong structures. However, OFE does progressively worse.

5.5 "Correctness of Structure", Incomplete Data

We next degraded this training data by randomly removing the value of each attribute, within each instance, with probability 0.5. Figure 6(b) compares ELR with the standard systems APN and EM; we see that ELR is more accurate, in each case.

5.6 Nearly Correct Structure — Real World Data

We next asked whether ELR could find the best parameters for more complicated structures. Due to space limitation, this paper will report on only two situations; [GZ02] presents other examples. First, we considered the more-nearly-correct structures learned using the POWERCONSTRUCTOR system [CG02; CG99]. We know that this system will converge to the correct belief net, given enough data (and some other relatively benign assumptions).

Figure 2(b) shows the results for CHESS: again ELR works effectively — and better than OFE. (Comparing Figure 2(a) to Figure 2(b) shows that ELR was not that hampered by the poorness of the NaïveBayes structure, but OFE was.)

To see how well ELR would perform on the correct structure, but given *incomplete* training data, we considered the ALARM network B_{alarm} [BSCC89], which has a known structure involving 36 nodes, 47 links and 505 parameters.

Here, we had to define the appropriate query distribution. From [HC91], we know that 8 of the ALARM variables typically appear as query variables, and a disjoint set of 16 variables can appear as evidence. We therefore generated queries by uniformly selecting, as query variable, one of the 8 query variables, and then, for each of the 16 evidence variables, including it with probability $1/2$ — hence on average a query will include $16/2$ evidence variables. (Note that different instances used different variables as the class label [CPT97]; here it was critical to distinguish $sq(\cdot)$ from $P(\cdot)$ [GGS97].) We then specify values for these evidence variables based on the natural joint distribution for these evidence variables. Figure 6(c) shows that ELR works more effectively here.

5.7 Other Experiments

The studies so far focus on the common situation where the model ("BN-structure") we are instantiating is likely simpler than the "truth" — *e.g.*, we used naïve-bayes when there probably were dependencies between the attributes. Here, we have a great deal of evidence that our ELR algorithm, which tries to optimize conditional likelihood, works better than generative algorithms, which optimize likelihood. [GZ02] considers other (less typical) situations, where the model is more complex than the truth. In a nutshell, we observed, as expected, that discriminative learning (here ELR) will often over-fit in this situation, and so produce results that are often inferior to the generative learners. We were able to reduce this effect by initializing the parameters with the OFE values (in the complete data case); notice many discriminative learners do this, especially when (like here) these values are "plug-in parameters [Rip96].

5.8 Summary of Empirical Data

Our empirical studies using the UCI datasets suggest, when given complete training data,

$$\boxed{\begin{array}{lllll} \text{TAN+ELR} & > & \text{TAN+OFE} & & \\ & & \text{NB+ELR} & > & \text{NB+OFE} \end{array}}$$

and when dealing with incomplete data,

$$\boxed{\text{NB+ELR} \quad > \quad \left\{ \begin{array}{c} \text{NB+APN} \\ \text{NB+EM} \end{array} \right\}}$$

where ">" indicates statistical significance at the $p < 0.05$ level or better. (While many of the other comparisons suggest an ELR-based systems worked better, those results were not statistically significant.)

We see that ELR proved especially advantageous when the BN-structure was *incorrect* — *i.e.*, whenever it is not a I-map of the underlying distribution by incorrectly claiming that two dependent variables are independent [Pea88]. This is a very common situation, as many BN-learners will produce incorrect structures, either because they are conservative in adding new arcs (to avoid overfitting the data), or because they are considering only a restricted class of structures (*e.g.*, naïve-bayes [DH73], poly-tree [CL68; Pea88], TAN [FGG97], etc.) which is not guaranteed to contain the correct structure.

6 Conclusions

This paper overviews the task of discriminative learning of belief net parameters for general BN-structures. We first describe this task, and discuss how it extends that standard logistic regression process by applying to arbitrary structures, not just naïve-bayes. Next, our formal analyses shows that discriminative learning can require fewer training instances than generative learning to converge, and that it will often converge to a superior classifier. The computational complexity is harder to compare: While we know our specific task — finding the optimal CL parameters for a given general structure, from incomplete data — is NP-hard, we do not know the corresponding complexity of finding the parameters that optimize likelihood. We suspect that discriminative learning may be faster as it can focus on only the relevant parts of the network; this can lead to significant savings when the data is incomplete. Moreover, if we consider the overall task, of learning both a structure and parameters, then we suspect discriminative learning may be more efficient that generative learning, as it can do well with a simpler structure.

We next present an algorithm ELR for our task, and show that ELR works effectively over a variety of situations: when dealing with structures that range from trivial (naïve-bayes), through less-trivial (TAN), to complex (Alarm, and ones learned by POWERCONSTRUCTOR). We also show that ELR works well when given *partial* training data, and even if different instances use different query variables. (This is one of the advantages of using a general belief net structure.) We also include a short study to explain why ELR can work effectively, showing that it typically works better than generative methods when dealing with models that are less complicated than the true distribution (which is a typical situation).

While statisticians are quite familiar with the idea of discriminative learning (*e.g.*, logistic regression), this idea, in the context of belief nets, is only beginning to make in-roads into the general AI community. We hope this paper will help further introduce these ideas to this community, and demonstrate that these algorithms should be used here, as they can work very effectively.

Acknowledgements

We thank Lyle Ungar, Tom Dietterich, Adam Grove, Peter Hooper, and Dale Schuurmans for their many helpful suggestions. Both authors were partially funded by NSERC; RG was also funded by Siemens Corporate Research; and WZ, by Syncrude.

References

[ATW91] N. Abe, J. Takeuchi, and M. Warmuth. Polynomial learnability of probablistic concepts with respect to the Kullback-Leibler divergence. In *COLT*, pages 277–289. 1991.

[Bis98] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford, 1998.

[BKRK97] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

[BM00] C. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[BSCC89] I. Beinlich, H. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *ECAI-Medicine*, August 1989.

[CG99] J. Cheng and R. Greiner. Comparing bayesian network classifiers. In *UAI'99* pages 101–107. August 1999.

[CG02] J. Cheng and R. Greiner. Learning bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 2002. to appear.

[CH92] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[CL68] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, pages 462–467, 1968.

[CPT97] R. Caruana, L. Pratt, and S. Thrun. Multitask learning. *Machine Learning*, 28:41, 1997.

[Das97] S. Dasgupta. The sample complexity of learning fixed-structure bayesian networks. *Machine Learning*, 29, 1997.

[DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[DP96] P. Domingo and M. Pazzani. Beyond independence: conditions for the optimality of the simple bayesian classier. In *ICML*, 1996.

[FGG97] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[FI93] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1027, 1993.

[GGS97] R. Greiner, A. Grove, and D. Schuurmans. Learning Bayesian nets that perform well. In *UAI* 1997.

[GZ02] R. Greiner and W. Zhou. Beyond logistic regression. Technical report, UofAlberta, 2002.

[HC91] E. Herskovits and C. Cooper. Algorithms for Bayesian belief-network precomputation. In *Methods of Information in Medicine*, pages 362–370, 1991.

[Hec98] D. Heckerman. A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, 1998.

[Jor95] M. Jordan. Why the logistic function? a tutorial discussion on probabilities and neural networks, 1995.

[KJ97] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2), 1997.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, 1995.

[LR87] J. Little and D. Rubin. *Statistical Analysis with Missing Data*. Wiley, New York, 1987.

[Min01] Tom Minka. Algorithms for maximum-likelihood logistic regression. Technical report, CMU CALD, 2001. http://www.stat.cmu.edu/∼minka/papers/logreg.html.

[Mit97] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[MN89] P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

[NJ01] A. Ng and M. Jordan. On discriminative versus generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 2001.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Rip96] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.