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

Russell Greiner[1] Wei Zhou[2] Xiaoyuan Su[3] Bin Shen[1]

[1]: Dept of Computing Science, University of Alberta, Edmonton, AB T6G 2H1 Canada

[2]: Dept of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

[3]: Dept of Electrical & Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4 Canada

April 23, 2004

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 extends standard logistic regression. After analyzing its inherent sample and computational complexity, we present a general algorithm for this task, ELR, which applies to arbitrary BN structures and which works effectively even when given incomplete training data. This paper presents empirical evidence that ELR produces better classifiers than are produced by the standard "generative" algorithms in a variety of situations, especially in common situations where the given BN-structure is incorrect.

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

1 Introduction

Many tasks — including fault diagnosis, pattern recognition and forecasting — can be viewed as *classification*, as each requires assigning the class ("label") to a given instance, which is specified by a set of attributes. 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, optimality may be impossible for learners constrained to a restricted range of possible distributions that excludes the correct one (*e.g.*, when only considering parameterizations of a given BN-structure).

*This paper extends the earlier results that appear in [SSG+03] and [GZ02].

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 $\{\langle c_i, \mathbf{e}_i \rangle\}$ (each class label $C = c_i$ associated with evidence $\mathbf{E} = \mathbf{e}_i$), a discriminative learner will try to find parameters Θ that maximize

$$CL_{\{\langle c_i, \mathbf{e}_i \rangle\}}(\,\Theta\,) \quad = \quad \sum_i \log P_\Theta(\,c_i \,|\, \mathbf{e}_i\,) \tag{1}$$

rather than the ones that maximize $\sum_i \log P_\Theta(\,c_i, \mathbf{e}_i\,)$ [Rip96].

Optimizing the CL of the root node (given the other attributes) 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, among other generalizations. This paper provides a general discriminative learning tool `ELR`[1] that can learn the parameters for an arbitrary structure:

NaïveBayes : General Belief Net :: Simple Logistic Regression : `ELR` (2)

Moreover, while most algorithms for learning logistic regression functions require *complete* training data, the `ELR` algorithm can accept *incomplete* data. We also present empirical evidence, from a large number of datasets, that demonstrate that `ELR` works effectively.

Section 2 provides the foundations, overviewing belief nets then defining our task: discriminative learning the parameters (for a fixed belief net structure, G) that maximize CL. Section 3 formally analyses this task, providing both sample and computational complexity, and noting how these results compare 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 which demonstrate that our `ELR` often produces a classifier that is superior to ones produced by 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. As the `ELR` behavior can depend on how the given BN-structure G relates to the true structure T, we consider three regimes:[2]

- when G is *simpler* than T — *i.e.*, when G has too few links
- when G is *approximately equal* to T (*e.g.*, G is produced by a structure-learning algorithm)
- when G is *more complicated* than T — *i.e.*, when G has too many links

Section 6 provides a brief survey of the relevant literature, and the appendix provides the proofs of our theoretic claims.

2 Framework

We assume there is a stationary underlying distribution $P(\,\cdot\,)$ over n (discrete) random variables $\mathcal{V} = \{V_1, \ldots, V_n\}$; For example, perhaps V_1 is the "`Cancer`" random variable, whose value ranges over $\{$`true`, `false`$\}$; V_2 is "`Gender`" $\in$ $\{$`male`, `female`$\}$, V_3 is "`Age`" $\in$ $\{0, .., 100\}$, etc. We refer to this joint distribution as the "underlying distribution" or the "event distribution".

[1] for "Extended Logistic Regression"

[2] We are grateful to Professor Peter Hooper for this elegant breakdown.

We can encode this 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 immediate 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 $P(\,D = d \,|\, \mathbf{F} = \mathbf{f}\,)$ [Pea88].

The user interacts with the belief net by asking *queries*, each of the form "What is $P(\,C = c \,|\, \mathbf{E} = \mathbf{e}\,)$?" — *e.g.*, What is $P($`Cancer = true`$|$`Gender=female, Smoke=true`$)$? — 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., $\mathbf{E}$). This paper focuses on the case where all queries involve the same variable; *e.g.*, all queries ask about `Cancer`. Moreover, we will follow standard practice by assuming the distribution of conditioning events matches the underlying distribution. This means there is a single distribution from which we can draw instances, which correspond to a set of labeled instances (aka "labeled queries").[3] Note this corresponds to the data sample used by standard learning algorithms.

Given any unlabeled instance $\{\mathbf{E}_i = \mathbf{e}_i\}$, the belief net B will produce a distribution over the values of the query variable; perhaps $B($ `Cancer = true` $|\,\mathbf{E} = \mathbf{e}\,) = 0.3$ and $B($ `Cancer = false` $|\,\mathbf{E} = \mathbf{e}\,) = 0.7$. In general, the 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 — here return $H_B(\,\mathbf{E} = \mathbf{e}\,) =$ false as $B($ `Cancer = false` $|\,\mathbf{E} = \mathbf{e}\,) > B($ `Cancer = true` $|\,\mathbf{E} = \mathbf{e}\,)$.

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

$$\text{err}(\,B\,) \quad = \quad \sum_{\langle \mathbf{e}, c\rangle} P(\,\mathbf{e},\, c\,) \;\times\; \mathcal{I}(\,H_B(\mathbf{e}) \neq c\,) \tag{3}$$

where $\mathcal{I}(\,a \neq b\,) \;=\; 1$ if $a \neq b$, and $= 0$ otherwise.[4]

Our goal is a belief net B^* that minimizes this score, with respect to the true distribution $P(\,\cdot\,|\,\cdot\,)$. While we do not know this distribution *a priori*, we can use a sample drawn from this distribution, to help determine which belief net is optimal. 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 "log conditional likelihood" of a belief net B is

$$\text{LCL}_P(\,B\,) \quad = \quad \sum_{\langle \mathbf{e}, c\rangle} P(\,\mathbf{e},\, c\,) \;\times\; \log(B(\,c\,|\,\mathbf{e}\,)) \tag{4}$$

Given a sample S, we can approximate this as

$$\widehat{\text{LCL}}^{(S)}(\,B\,) \quad = \quad \frac{1}{|S|} \sum_{\langle \mathbf{e}, c\rangle \in S} \log(\,B(\,c\,|\,\mathbf{e}\,)\,) \tag{5}$$

[MN89, FGG97] note that maximizing this score will typically produce a classifier that comes close to minimizing the classification error (Equation 3). 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 (empirical) "log likelihood" function

$$\widehat{\text{LL}}^{(S)}(\,B\,) \quad = \quad \frac{1}{|S|} \sum_{\langle \mathbf{e}, c\rangle \in S} \log(\,B(\,c, \mathbf{e}\,)\,) \tag{6}$$

[3] See [GGS97] for an alternative position, and the challenges this requires solving.

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

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

$$\widehat{\mathrm{LL}}^{(S)}(\,B\,) \;=\; \frac{1}{|S|}\left[\sum_{\langle c,\mathbf{e}\rangle\in S}\log(\,B(\,c\,|\,\mathbf{e}\,)\,) \;+\; \sum_{\langle c,\mathbf{e}\rangle\in S}\log(\,B(\,\mathbf{e}\,)\,)\right]$$

where the first term resembles our $\widehat{\mathrm{LCL}}^{()}(\,\cdot\,)$ 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 "$\widehat{\mathrm{LCL}}^{(S)}(\,\cdot\,)$-like" term may be preferred to a B_β that does better — *i.e.*, if $\widehat{\mathrm{LL}}^{(S)}(\,B_\alpha\,) < \widehat{\mathrm{LL}}^{(S)}(\,B_\beta\,)$, while $\widehat{\mathrm{LCL}}^{(S)}(\,B_\alpha\,) > \widehat{\mathrm{LCL}}^{(S)}(\,B_\beta\,)$. (Section 5.4 provides other arguments explaining why our approach may work better than the $\widehat{\mathrm{LL}}^{(S)}(\,\cdot\,)$-based approaches; and Section 6 surveys other relevant literature.)

Finally, all of our algorithms are producing parameters for a given belief net structure, G. We therefore define $\mathcal{BN}(G) = \{\langle G, \Theta\rangle\}$ to be the set of all belief nets that use the structure G; our goal is the set of CPtable parameters $\Theta \in [0,1]^k$, appropriate for this G structure, that produces the optimal classification performance. Notice we can view $\mathcal{BN}(G)$ as the set of these parameters.

3 Theoretical Analysis

How many "labeled instances" are enough — *i.e.*, given any values $\epsilon, \delta > 0$, how many labeled instances are needed to insure that, with probability at least $1-\delta$, an algorithm can produce a classifier that is within ϵ of 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

$$\mathcal{BN}_{\Theta\succeq\gamma}(G) \quad = \quad \{\, B \in \mathcal{BN}(G) \mid \forall \theta_{d|\mathbf{f}} \in \Theta,\ \theta_{d|\mathbf{f}} \geq \gamma\,\} \tag{7}$$

be the subset of BNs whose CPtable values are all at least γ.[5] We now restrict our attention to these belief nets, and in particular, let

$$B^*_{G,\Theta>\gamma} \quad = \quad \operatorname{argmax}\{\mathrm{LCL}_P(\,B\,) \mid B \in \mathcal{BN}_{\Theta\succeq\gamma}(G)\} \tag{8}$$

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

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 5) with respect to a sample of*

$$M_{\gamma,N,K}(\epsilon,\delta) \quad = \quad 18\left(\frac{N\,\ln\gamma}{\epsilon}\right)^2\left[\log\frac{2}{\delta} \;+\; K\,\log\frac{6\,K}{\gamma\,\epsilon}\right] \quad = \quad O\left(\frac{N^2\,K}{\epsilon^2}\ln(\frac{K}{\epsilon\delta})\ \log^3(\frac{1}{\gamma})\right) \tag{9}$$

labeled queries drawn from $P(\,\cdot\,)$. Then, with probability at least $1-\delta$, $\hat{B}$ will be no more than ϵ worse than $B^*_{G,\Theta>\gamma}$ (from Equation 8). ∎

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

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.

[5]This $\theta_{d|\mathbf{f}} \geq \gamma$ constraint is trivially satisfied by any parameter learner that uses Laplacian correction, or that produces the posterior distribution from uniform Dirichlet priors, using $\gamma = 1/m$ where m is the number of training instances [Hec98].

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

$$\frac{288\,N^2\,K}{\epsilon^2}\ln^2\left(1+\frac{3\,N}{\epsilon}\right)\ln\frac{18\,N\,K\ln(1+3\,N/\epsilon)}{\epsilon\,\delta} \quad = \quad O\left(\frac{N^2\,K}{\epsilon^2}\,\ln(\frac{K}{\epsilon\,\delta})\,\ln^3(N)\,\ln^2(\frac{1}{\epsilon})\right) \tag{10}$$

complete tuples are sufficient to learn the parameters to a fixed structure that are with ϵ of the optimal *likelihood* (Equation 6). While comparing upper bounds is only suggestive, it is interesting to note that, ignoring the $\ln^{\ell}(\cdot)$ terms for $\ell > 1$, these bounds are asymptotically identical.

One asymmetry is that only our Equation 9 bound includes the γ term, which corresponds to the smallest CPtable entry allowed. While [Das97] (following [ATW91]) can avoid this term by "tilting" the empirical distribution, this trick does not apply in our discriminative task: Our task inherently involves computing *conditional* likelihood, which requires *dividing* by some CPtable values, which is problematic when these values are near 0. This observation also means our proof is *not* an immediate application of the standard PAC-learning approaches. Of course, our sample complexity remains polynomial in the size (N, K) of the belief net even if this γ is exponentially small, $\gamma = O(1/2^N)$.

Note finally that the parameters that optimize (or nearly optimize) likelihood will not optimize our objective of *conditional* likelihood, which means Equation 10 describes the convergence to parameters that are typically inferior to the ones associated with Equation 1, especially in the unrealizable case; see [NJ01].

The second question is computational: How hard is it to find these best parameters values, given this sufficiently large sample. Here, there is both good news and bad news:

Observation 2 (Good News) *Given a fixed belief net structure G whose parameters Θ are of size $|\Theta|$, a* complete *data sample S of size $|S|$, and $\epsilon \in (0,1)$, there is an algorithm that can, in time polynomial in $1/\epsilon$, $|\Theta|$ and $|S|$, find the values of the CPtables of G whose (empirical) conditional likelihood (Equation 5) with respect to S, is within ϵ of optimal.* ∎

Proof (sketch)[6] **:** It suffices to show that minimizing "negative log conditional likelihood" is a convex optimization problem, as we can then simply use standard interior point methods [NN94, BV04] to find a solution in polyominal time. To see this, observe that the negation of each term in Equation 5 is $-\ln P(\,C = c_i\,|\,\mathbf{e}\,) \;=\; -\ln P(\,C = c_i, \mathbf{e}\,) + \sum_j \ln P(\,C = c_j, \mathbf{e}\,)$. In the complete data case, each $P(\,C = c_i, \mathbf{e}\,)$ is a product, of the form $\prod_k \theta_k$. In the binary case, this corresponds to $e^{(\mathbf{e}\cdot\beta)}$ for some parameters β (which basically correspond to $\ln\theta_+/\theta_-$). Hence each term in Equation 5 corresponds to $-(\mathbf{e}\cdot\beta) + \ln\sum_k \mathbf{e}\cdot\beta_k$. which is convex. (We can use standard means to scale beyond binary values.) ∎

Unfortunately…

Theorem 3 (Bad News) *It is NP-hard to find the values for the CPtables of a fixed BN-structure that produce the smallest (empirical) conditional likelihood (Equation 5) for a given sample. This holds even if we consider only BNs in $\mathcal{BN}_{\Theta\succeq\gamma}(G)$ for $\gamma = O(1/N)$.* ∎

Of course, given Observation 2, this relies on *incomplete* instances. Moreover, the class of structures used to show hardness are more complicated than the naïve-bayes and `TAN` structures (see below), but see Section 5.2.

We close by observing that this situation resembles the generative case (for computing the parameters that optimize simple *likelihood* from complete data), as in each case there is an efficient algorithm for dealing with complete data (see `OFE`, below), but only iterative approaches for dealing with the incomplete case; see Section 5.1.3.

[6] We are grateful to John Lafferty for suggesting this proof.

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

$$\theta_{d|\mathbf{f}} \quad = \quad \frac{e^{\beta_{d|\mathbf{f}}}}{\sum_{d'} e^{\beta_{d'|\mathbf{f}}}} \tag{11}$$

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 $P_{\alpha,\chi}(\, C = c \,|\, \mathbf{E} = \mathbf{e}\,) \;=\; e^{\alpha_c + \chi_{\mathbf{c}} \cdot \mathbf{e}} / \sum_j e^{\alpha_j + \chi_{\mathbf{j}} \cdot \mathbf{e}}$.[7] Recall that our goal is a more general algorithm — one that can deal with *arbitrary* structures; see Equation 2.)

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

Proposition 4 *For the labeled query* $[\mathbf{e}; c]$ *and each "softmax" parameter* $\beta_{d|\mathbf{f}}$*,*

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

Our ELR also incorporates several enhancement to speed-up 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. Finally, our empirical studies show that the number of hill-climbing iterations is crucial — the algorithm needs enough to get the improvements, but must stop before it overfits. We therefore use a type of cross validation to determine this number.

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

5 Empirical Studies

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 CPtable 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 6) for a given structure [CH92]. For example, if 75 of the 100 $C = 1$ instances have $X_3 = 0$, then OFE sets

$$\theta_{X_3=0|C=1} = \frac{75}{100} \tag{12}$$

(Some versions use a Laplacian correction to avoid $0/0$ issues.) When the data is *incomplete*, we compare ELR to the standard Expectation Maximization algorithm EM [Hec98] and to APN [BKRK97], which descends to parameter values whose likelihood is locally optimal.

Here we present only the results of the ELR=ELR$_\beta$ algorithm, which used the β terms (Equation 11), as we found its performance strictly dominated the ELR$_\theta$ version which used θ directly. Similarly, while the original APN$_\theta$ [BKRK97] climbed in the space of parameters $\Theta = \{\theta_i\}$, we instead used a modified APN$_\beta$ system that uses the $\beta = \{\beta_i\}$ values (Equation 11), as we found it worked better as well.

Traditional wisdom holds that discriminative learning (ELR) is most relevant (*i.e.*, better than generative learning: OFE, APN, EM) when this underlying model $G = \langle \mathcal{V}, A \rangle$ is "wrong", that is, not an I-map of the true distribution T — which here means the graph structure does not include some essential arcs [Pea88]. The situation, which we denote "$G < T$", is fairly common as many learners consider only structures as simple as naïve-bayes, or the class of TAN structures (defined below), which are typically much simpler than T.[8] §5.1 deals with this situation, considering both the complete and imcomplete data cases.

§5.2 then considers another standard situation: where we employ a structure-learning algorithm to produce a structure that is similar to the truth; *i.e.*, where $G \approx T$. Here, we use the POWERCONSTRUCTOR system [CG02, CG99] for the first step (to learn the structure), then compare the relative effectiveness of algorithms for finding parameters for this structure.

We next consider the uncommon situation where the model G is more *complicated* than the truth T; *i.e.*, $G > T$. §5.3 uses artificial data to compare ELR vs OFE, APN and EM, in this context.

Finally, §5.4 summarizes all of these empirical results.

Notation: The notation "NB+ELR" will refer to the NB structure, whose parameters are learned using ELR; in general, we will use x+y to refer to the y instantiation of the x structure. Below we will compare various pairs of learners, in each case over a common dataset; we will therefore use a one-sided paired t-test [Mit97]. When this result is significant at the $\rho < 0.05$ level, we will write $\alpha \Leftarrow_{(p<\rho)} \beta$ — *e.g.*, we will soon see NB+ELR $\Leftarrow_{(p<0.005)}$ NB+OFE. For values larger than 0.05, we will use the notation $\alpha \leftarrow_{(p<\rho)} \beta$. (That is, we regard $p < 0.05$ as the cut-off for statistical significance.) Note the arrow will point to the learner that (appears) to be better.

While our main emphasis is comparing x+ELR to x+OFE (and to x+APN and x+EM) for various structures x, where relevant we will also compare across structure classes; *e.g.*, comparing x+ELR to y+ELR for different structures x and y.

[8] Note the $G < T$ notation does *not* mean the arcs of G are a subset of T's, as G may also include arcs that are not in T.

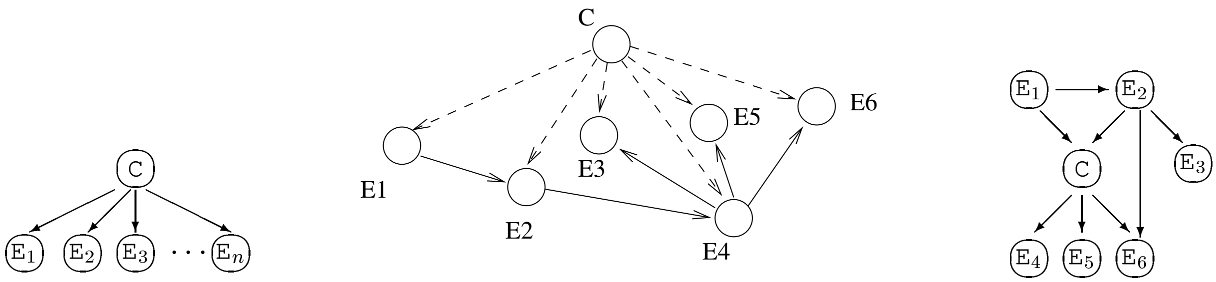


Figure 1: (a) NaïveBayes structure (b) TAN structure ([FGG97, Fig 3]) (c) Arbitrary GBN structure

5.1 Model is Simpler than the Truth ($G < T$)

§5.1.1 (resp., §5.1.2) compares algorithms for learning the parameters for a naïve-bayes model (resp., a TAN model) given *complete* data; §5.1.3 then considers learning these models given *incomplete* data. §5.1.4 uses a simple controlled study, on artificial data, to further investigate the earlier claim.

5.1.1 NaïveBayes — Complete, Real World Data

Our first experiments deal with the simplest situation: learning the NaïveBayes parameters from complete data. Recall that the NaïveBayes structure requires that the attributes are independent given the class label; see Figure 1(a). It is well-known that ELR, with this structure corresponds to standard logistic regression [NJ01].

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; see Table 1, which also specifies how we computed our accuracy values — based on 5-fold cross validation for small data, and holdout method for large data [Koh95]. To deal with continuous variables, we implemented supervised entropy discretization [FI93]. Table 2 summarizes the results. (§5.2.2 will later explain the line separating the first 20 datasets from the final 5.)

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 (a.k.a. "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 label "ELR-OFE" emphasizes that we used OFE to initialize the parameters, then used the ELR-gradient-descent. As we found this was beneficial, especially for small samples, we used it for all studies.[9])

Figure 3(a) provides a more comprehensive comparison, across all 25 datasets. (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.

5.1.2 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,

[9] Many discriminative learners initialize the parameters with the OFE values, especially when (like here) these values are "plug-in parameters" [Rip96].

Table 1: Description of data sets used in the experiments; see also [FGG97]

	Dataset	# Attributes	# Classes	# Instances	
				Train	Test
1	AUSTRALIAN	14	2	690	CV-5
2	BREAST	10	2	683	CV-5
3	CHESS	36	2	2130	1066
4	CLEVE	13	2	296	CV-5
5	CORRAL	6	2	128	CV-5
6	CRX	15	2	653	CV-5
7	DIABETES	8	2	768	CV-5
8	FLARE	10	2	1066	CV-5
9	GERMAN	20	2	1000	CV-5
10	GLASS	9	7	214	CV-5
11	GLASS2	9	2	163	CV-5
12	HEART	13	2	270	CV-5
13	HEPATITIS	19	2	80	CV-5
14	IRIS	4	3	150	CV-5
15	LETTER	16	26	15000	5000
16	LYMPHOGRAPHY	18	4	148	CV-5
17	MOFN-3-7-10	10	2	300	1024
18	PIMA	8	2	768	CV-5
19	SHUTTLE-SMALL	9	7	3866	1934
20	VOTE	16	2	435	CV-5
21	SATIMAGE	36	6	4435	2000
22	SEGMENT	19	7	1540	770
23	SOYBEAN-LARGE	35	19	562	CV-5
24	VEHICLE	18	4	846	CV-5
25	WAVEFORM-21	21	3	300	4700

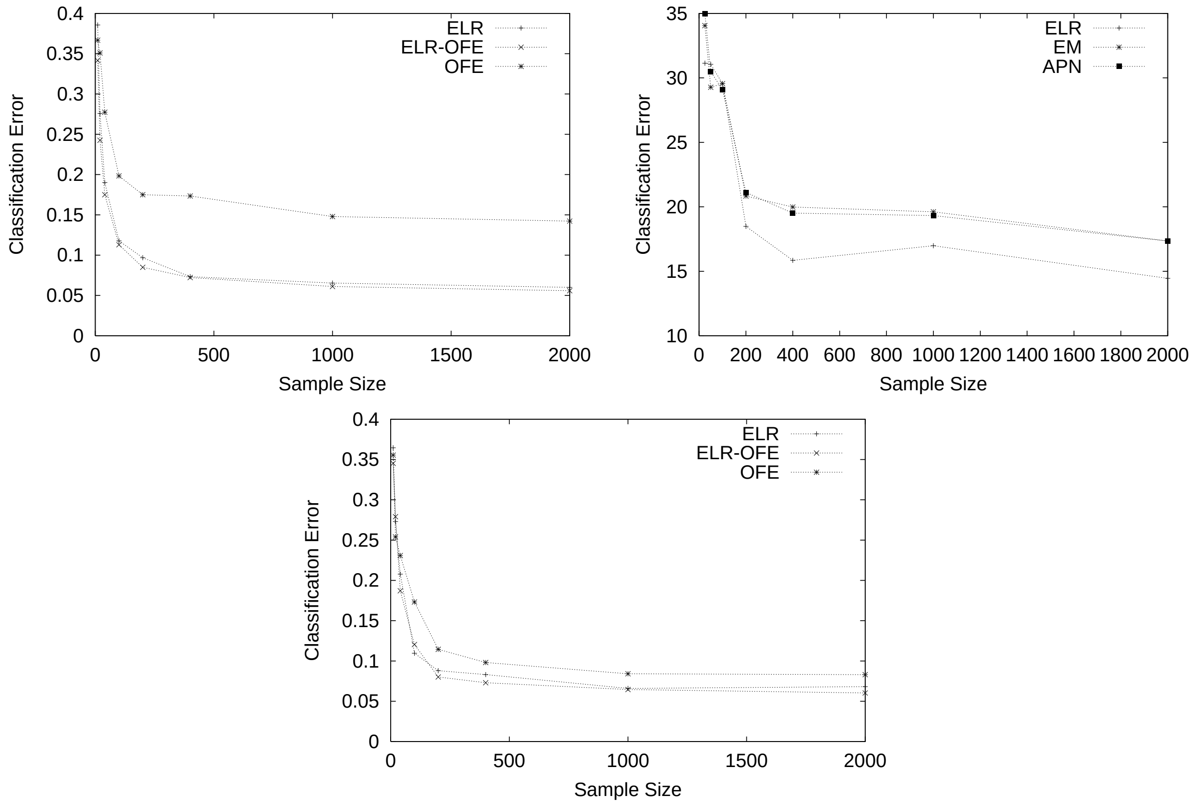


Figure 2: CHESS domain: (a) ELR vs OFE, complete data, structure is "incorrect" (naïve-bayes); (b) ELR vs EM, APN on incomplete data, structure is "incorrect" (naïve-bayes) (c) ELR vs OFE, complete data, structure is "≈correct" (POWERCONSTRUCTOR)

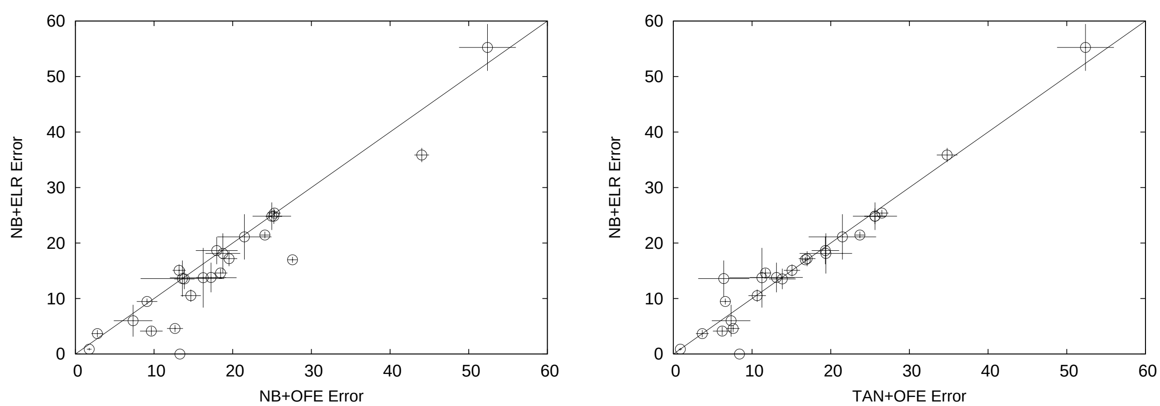


Figure 3: Comparing NB+ELR with (a) NB+OFE (b) TAN+OFE

Table 2: Empirical accuracy of classifiers learned from *complete* data

	Data set	NB+OFE	NB+ELR	TAN+OFE	TAN+ELR	GBN+OFE	GBN+ELR
1	AUSTRALIAN	86.81±0.84	84.93±1.06	84.93±1.03	84.93±1.03	86.38±0.98	86.81±1.11
2	BREAST	97.21±0.75	96.32±0.66	96.32±0.81	96.32±0.70	96.03±0.50	95.74±0.43
3	CHESS	87.34±1.02	95.40±0.64	92.40±0.81	97.19±0.51	90.06±0.92	90.06±0.92
4	CLEVE	82.03±2.66	81.36±2.46	80.68±1.75	81.36±1.78	84.07±1.48	82.03±1.83
5	CORRAL	86.40±5.31	86.40±3.25	93.60±3.25	100.00±0.00	100.00±0.00	100.00±0.00
6	CRX	86.15±1.29	86.46±1.85	86.15±1.70	86.15±1.70	86.00±1.94	85.69±1.30
7	DIABETES	74.77±1.05	75.16±1.39	74.38±1.35	73.33±1.97	75.42±0.61	76.34±1.30
8	FLARE	80.47±1.03	82.82±1.35	83.00±1.06	83.10±1.29	82.63±1.28	82.63±1.28
9	GERMAN	74.70±0.80	74.60±0.58	73.50±0.84	73.50±0.84	73.70±0.68	73.70±0.68
10	GLASS	47.62±3.61	44.76±4.22	47.62±3.61	44.76±4.22	47.62±3.61	44.76±4.22
11	GLASS2	81.25±2.21	81.88±3.62	80.63±3.34	80.00±3.90	80.63±3.75	78.75±3.34
12	HEART	78.89±4.08	78.52±3.44	78.52±4.29	78.15±3.86	79.63±3.75	78.89±4.17
13	HEPATITIS	83.75±4.24	86.25±5.38	88.75±4.15	85.00±5.08	90.00±4.24	90.00±4.24
14	IRIS	92.67±2.45	94.00±2.87	92.67±2.45	92.00±3.09	92.00±3.09	92.00±3.09
15	LETTER	72.40±0.63	83.02±0.53	83.22±0.53	88.90±0.44	79.78±0.57	81.21±0.55
16	LYMPHOGRAPHY	82.76±1.89	86.21±2.67	86.90±3.34	84.83±5.18	79.31±2.18	78.62±2.29
17	MOFN-3-7-10	86.72±1.06	100.00±0.00	91.60±0.87	100.00±0.00	86.72±1.06	100.00±0.00
18	PIMA	75.03±2.45	75.16±2.48	74.38±2.81	74.38±2.58	75.03±2.25	74.25±2.53
19	SHUTTLE-SMALL	98.24±0.30	99.12±0.21	99.12±0.21	99.22±0.20	97.31±0.37	97.88±0.33
20	VOTE	90.34±1.44	95.86±0.78	93.79±1.18	95.40±0.63	96.32±0.84	95.86±0.78
21	SATIMAGE	81.55±0.87	85.40±0.79	88.30±0.72	88.30±0.72	79.25±0.91	79.25±0.91
22	SEGMENT	85.32±1.28	89.48±1.11	89.35±1.11	89.22±1.12	77.53±1.50	77.40±1.51
23	SOYBEAN-LARGE	90.89±1.31	90.54±0.54	93.39±0.67	92.86±1.26	82.50±1.40	85.54±0.99
24	VEHICLE	55.98±0.93	64.14±1.28	65.21±1.32	66.39±1.22	48.52±2.13	51.95±1.32
25	WAVEFORM-21	75.91±0.62	78.55±0.60	76.30±0.62	76.30±0.62	65.79±0.69	65.79±0.69

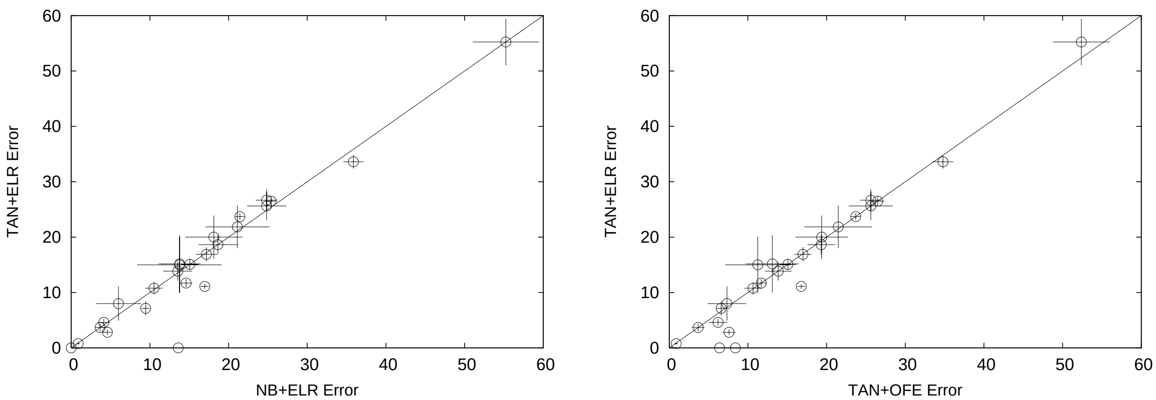


Figure 4: Comparing TAN+ELR vs (a) NB+ELR (complete data) (b) TAN+OFE (complete data)

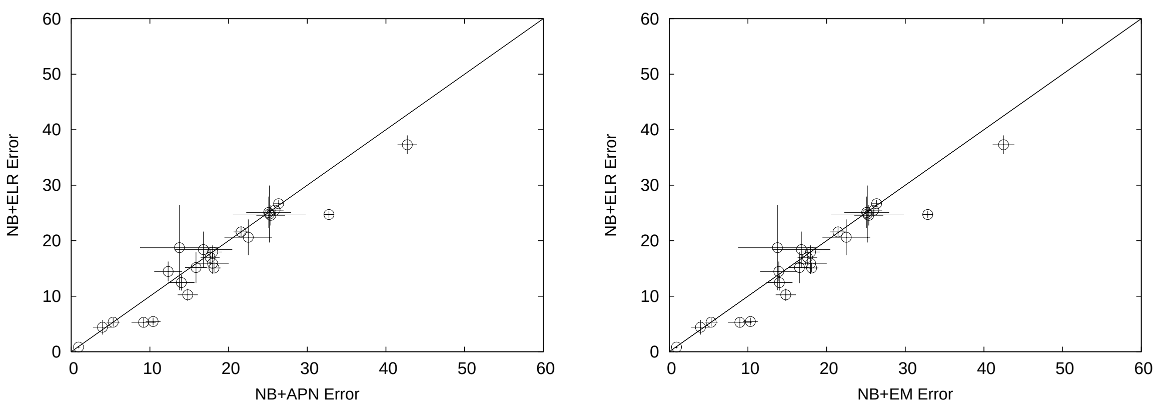


Figure 5: Incomplete Data: Comparing NB+ELR with (a) NB+APN; (b) NB+EM

connecting attributes to each other, form a tree; see Figure 1(b). (Hence this representation allows each attribute to have at most one "attribute parent", and so this class of structures strictly generalize NaïveBayes.) [FGG97] provides an efficient algorithm for learning such TAN structures, given complete data: This algorithm, based on [CL68], first computes the mutual information between each pair of attributes, conditioned on the class variable, then finds the minimum-weighted spanning tree within this complete graph of the attributes. (Each mutual information quantity is based on the empirical sample.)

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 4(a) suggests that TAN+ELR is slightly better than NB+ELR, this is not significant: only at the $p < 0.2$ level. However, Figure 4(b) 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.

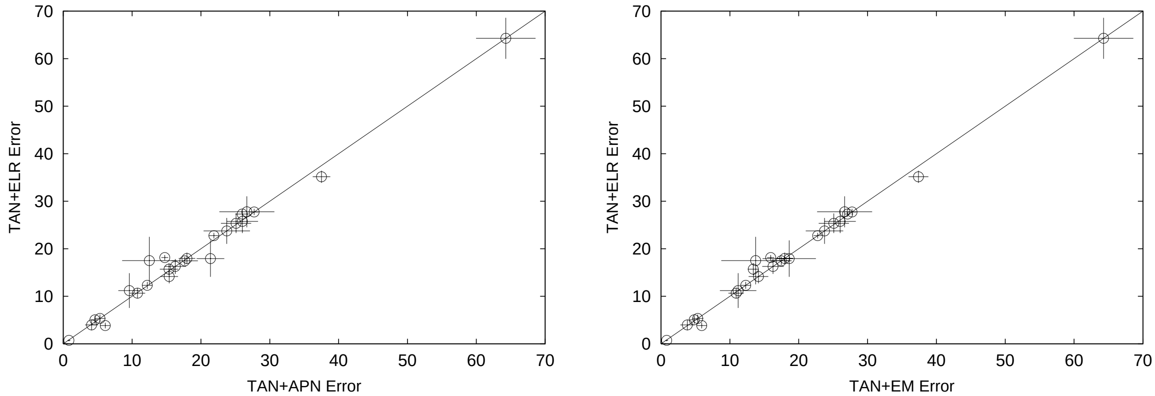


Figure 6: Incomplete data: Comparing TAN+ELR with (a) TAN+APN; (b) TAN+EM;

5.1.3 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(b) shows the learning curve for the CHESS domain, comparing ELR to APN and EM. We see that ELR does better for essentially every sample size. We also compared these algorithms over the rest of the 25 datasets; see Figures 5(a) and 5(b) 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 uses the mutual information between each pair of attributes, conditioned on the class variable. This is straightforward to compute 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 6(a) and 6(b) compare TAN+ELR to TAN+APN and to TAN+EM. We see that these systems are roughly equivalent: while TAN+ELR appears slightly better than TAN+EM, this is not significant (only at $p < 0.25$); similarly there is no significant difference between TAN+ELR and TAN+APN. Finally, we compared NB+ELR to TAN+ELR (Figure 7(a)), but found no significant difference here either.

Table 3 presents all of our empirical results related to missing data. (As we were concerned with the poor, but uniform, performance of all classifiers on the LETTER dataset, we further investigated their performances on this dataset, using various different missing-value rates. The results, shown in Figure 7(b), show that the error rate were not always so high (for smaller rates of missing data), and that various classifiers did have different performances when the missing data rate was smaller.)

5.1.4 "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; see above and [DP96].

Table 3: Empirical accuracy of classifiers learned from *incomplete* data

Data set	NB+ELR	NB+APN	NB+EM	TAN+ELR	TAN+APN	TAN+EM	GBN+ELR	GBN+APN	GBN+EM
AUSTRA-LIAN	78.41±1.01	78.41±0.96	78.55±1.01	77.25±0.59	78.12±0.74	77.25±0.59	74.06±1.06	74.06±1.06	74.78±0.74
BREAST	95.59±1.32	96.03±1.20	96.03±1.20	96.03±1.13	95.88±0.95	96.18±1.02	94.12±1.63	94.85±1.36	94.85±1.36
CHESS	94.56±0.69	89.59±0.94	89.68±0.93	96.15±0.59	93.90±0.73	94.09±0.72	90.34±0.90	90.06±0.92	90.06±0.92
CLEVE	84.07±1.90	82.03±2.05	82.03±2.05	83.73±1.57	83.73±1.57	83.73±1.57	83.05±1.93	81.36±2.34	83.39±1.89
CORRAL	81.60±3.25	83.20±3.67	83.20±3.67	88.80±3.67	90.40±1.60	88.80±2.65	92.00±1.79	88.80±2.65	92.00±1.79
CRX	87.54±1.43	86.00±1.67	86.00±1.67	85.85±1.43	84.62±1.29	85.85±1.43	86.15±1.67	87.23±1.10	86.92±0.97
DIABETES	75.42±1.84	74.64±1.83	74.64±1.83	74.64±2.06	74.90±2.19	74.90±2.19	73.46±1.99	73.20±1.99	72.81±1.79
FLARE	83.00±1.42	82.35±1.21	82.44±1.24	82.54±0.86	82.35±1.90	82.54±1.52	82.63±1.28	82.63±1.28	82.63±1.28
GERMAN	74.50±0.89	74.10±1.09	74.00±1.05	72.70±0.54	74.00±0.97	72.90±0.40	73.70±0.68	73.40±0.86	73.70±0.68
GLASS	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33	35.71±4.33
GLASS2	79.38±3.22	77.50±3.03	77.50±3.03	76.25±2.72	76.25±3.37	76.25±2.72	78.13±3.28	77.50±3.75	78.13±3.28
HEART	75.19±5.13	74.81±4.63	74.81±4.63	72.22±3.26	73.33±4.00	73.33±4.00	73.70±3.95	73.33±4.37	73.33±4.37
HEPATITIS	81.25±7.65	86.25±5.00	86.25±5.00	82.50±5.00	87.50±3.95	86.25±5.00	86.25±3.64	86.25±3.64	86.25±3.64
IRIS	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82	94.67±0.82
LETTER	75.28±0.61	67.24±0.66	67.14±0.66	81.86±0.54	85.25±0.50	84.07±0.52	72.80±0.63	69.81±0.65	68.60±0.66
LYMPHO-GRAPHY	84.83±2.80	84.14±1.38	83.45±1.29	82.07±3.84	78.62±2.01	81.38±3.87	78.62±2.29	78.62±2.29	79.31±2.18
MOFN-3-7-10	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20	82.03±1.20
PIMA	74.90±2.85	74.90±2.85	74.90±2.85	74.25±2.45	73.99±2.28	73.99±2.28	73.99±2.06	74.64±2.25	74.77±2.31
SHUTTLE-SMALL	99.17±0.21	99.07±0.22	99.07±0.22	99.28±0.19	99.17±0.21	99.17±0.21	99.22±0.20	98.04±0.32	98.04±0.32
VOTE	94.71±0.86	90.80±1.54	91.03±1.52	94.94±0.86	95.40±0.51	95.17±0.67	95.17±0.76	95.63±0.92	95.17±0.76
SATIMAGE	84.90±0.80	81.85±0.86	81.90±0.86	87.70±0.73	87.80±0.73	87.70±0.73	73.95±0.98	76.35±0.95	76.30±0.95
SEGMENT	89.74±1.09	85.19±1.28	85.19±1.28	89.35±1.11	89.22±1.12	89.09±1.12	77.40±1.51	77.40±1.51	77.40±1.51
SOYBEAN-LARGE	85.54±1.79	87.68±1.77	86.07±2.37	84.29±1.25	84.64±1.34	86.61±0.80	50.54±1.61	50.18±1.75	48.21±2.43
VEHICLE	62.72±1.69	57.28±1.25	57.51±1.38	64.85±1.29	62.49±1.28	62.60±1.44	49.94±0.91	44.73±1.94	44.73±1.94
WAVEFORM-21	73.34±0.64	73.64±0.64	73.64±0.64	72.26±0.65	72.28±0.65	72.26±0.65	64.38±0.70	55.85±0.72	55.85±0.72

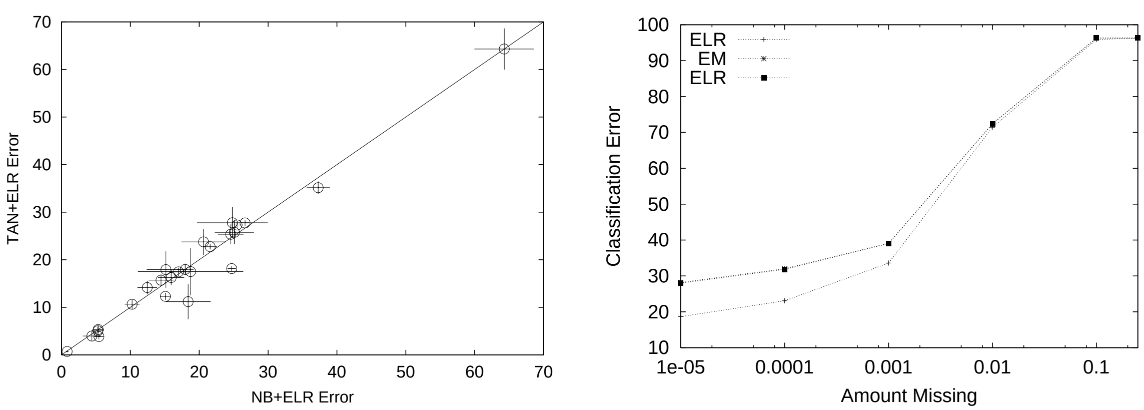
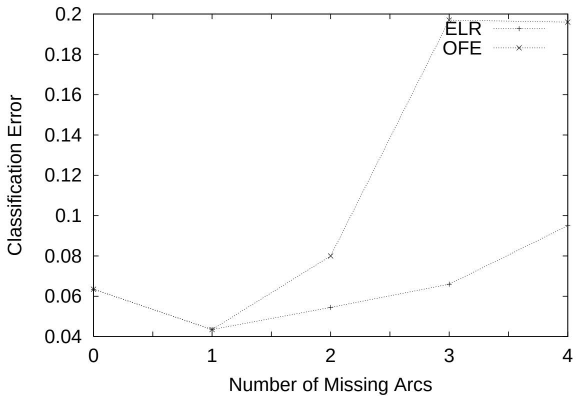


Figure 7: (a) Comparing TAN+ELR with NB+ELR on Incomplete Data (b) Comparing ELR to APN and EM on LETTER dataset, varying the quantity of missing data



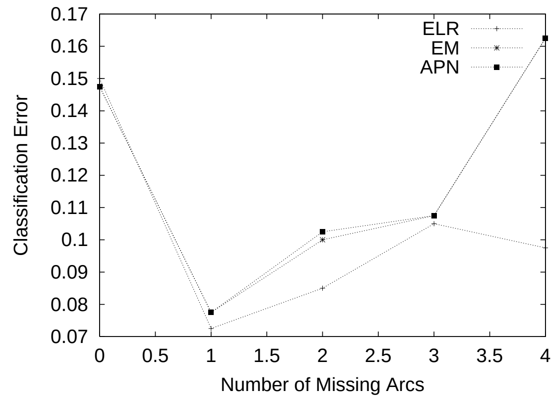


Figure 8: "Correctness of Structure": Comparing ELR to OFE, on increasingly incorrect structures for (a) Complete Data; (b) Incomplete Data;

We saw above that ELR is more robust than OFE, in that 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)

$$P(\,+c\,) = 0.9 \qquad P(\,+e_i \mid +c\,) = 0.2 \qquad P(\,+e_i \mid -c\,) = 0.8 \tag{13}$$

and our queries were all complete; *i.e.*, each instance of the form $\mathbf{E} = \langle \pm e_1,\, \pm e_2,\, \ldots,\, \pm e_k \rangle$.

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 $E_1 \equiv E_2 \equiv \ldots \equiv E_m$ (*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 13. 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 8(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.[10]

"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 8(b) compares ELR with the standard systems APN and EM; again we see that ELR is more accurate, in each case.

5.2 Model Approximates the Truth ($G \approx T$)

The previous section considered learners that were constrained to consider only some limited class of structures, such as NB or TAN. Other learners are allowed to first learn an arbitrary BN structure — seeking one that matches the underlying distribution — before learning the parameters of that structure, using ELR or OFE, etc. There are a number of algorithms for learning these BN structures, each of which will typically produce a structure that is close to correct. This paper considers the POWERCONSTRUCTOR sys-

[10]The "blip" at $m = 1$ is due to the quantization of selecting a classification value; we see a strictly monotonic relation if we instead consider "mean squared error"; see Footnote 4.

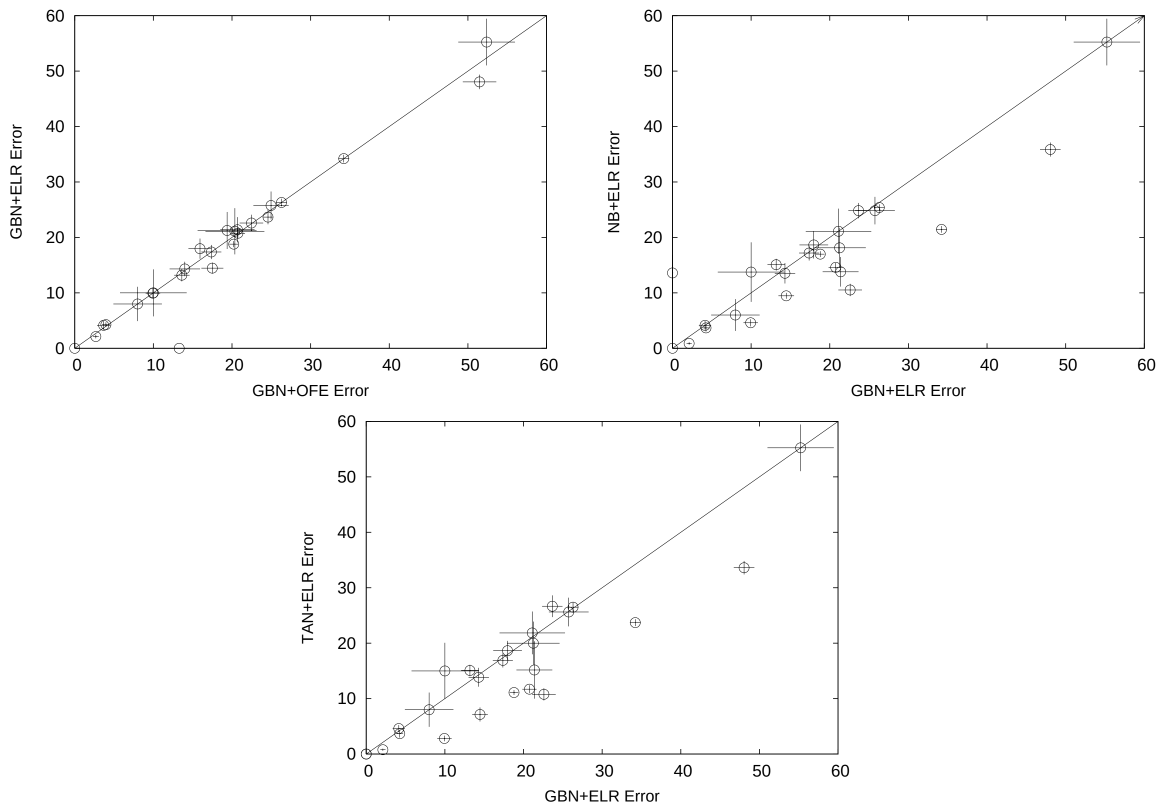


Figure 9: Comparing (a) `GBN+ELR` vs `GBN+OFE`; (b) `GBN+ELR` vs `NB+ELR`; (c) `GBN+ELR` vs `TAN+ELR` (over all 25 datasets)

tem [CG02, CG99], which uses mutual information tests to construct BN-structures from complete tuples. This algorithm is guaranteed to converge to the correct belief net structure, given enough data (and some other relatively benign assumptions). We will refer to the resulting POWERCONSTRUCTOR-produced structure as a "General Belief Net", or `GBN`; see Figure 1(c). This section explores the effectiveness of having such learned structures.

For each of the 25 datasets, we first used POWERCONSTRUCTOR to produce a structure for the given dataset, given all available (non–hold-out) data; we then asked `ELR` (resp., `OFE`) to find best parameters for this (presumably near optimal) structure (using the same non-hold-out data), and observed how well the resulting system performs, on the held-out data.

§5.2.1 compares `GBN+ELR` to `GBN+OFE`; §5.2.2 compares `GBN+ELR` to simpler models instantiated using `ELR`; and §5.2.3 compares the `OFE`-instantiation of `GBN` to `ELR`-instantiations of simpler models §5.2.4 investigates different algorithms for learning parameters (for these `GBN` structure) from *incomplete* data.

5.2.1 `GBN+ELR` vs `GBN+OFE`

The results shown in Figure 9(a) show that `GBN+ELR` is only insignificantly better than `GBN+OFE`: `GBN+ELR` $\leftarrow_{(p<0.2)}$ `GBN+OFE`. Hence, when considering reasonable structures, there appears to be little difference between `OFE` and `ELR`.

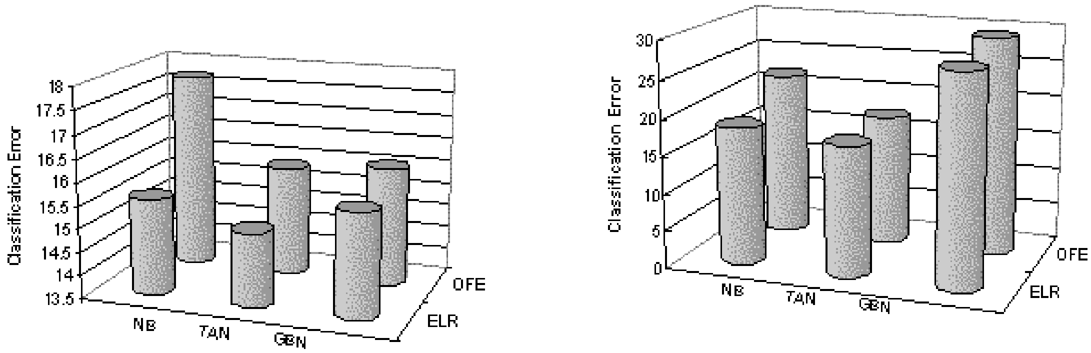


Figure 10: Average classification accuracy from various learning algorithms (a) results from 20 benchmark datasets (b) results from 5 problematic databases: SATIMAGE, SEGMENT, SOYBEAN-LARGE, VEHICLE and WAVEFORM-21

5.2.2 GBN+ELR vs NB+ELR, TAN+ELR

While TAN structures are more expressive than NB, we found earlier that TAN+ELR is *not* statistically better than NB+ELR. Could we produce a classifier that is superior to NB+ELR if we used the yet better expressive GBN structure. We therefore compared GBN+ELR to NB+ELR (Figure 9(b)), and to TAN+ELR (Figure 9(c)). Here, we were surprised to find that the simpler structures actually produced *better* classifiers than GBN did — NB+ELR $\Leftarrow_{(p<0.01)}$ GBN+ELR and TAN+ELR $\Leftarrow_{(p<0.008)}$ GBN+ELR !

One possible reason was that POWERCONSTRUCTOR was doing a poor job — producing structures that were *worse* than NaïveBayes. To test this, we considered the performances of their OFE-instantiations; *i.e.*, we compared GBN+OFE to NB+OFE. Here, we found that they were essentially the same: NB+OFE $\leftarrow_{(p<0.46)}$ GBN+OFE! Moreover, TAN+OFE was significantly better than GBN+OFE: TAN+OFE $\Leftarrow_{(p<0.02)}$ GBN+OFE!

A closer look reveals that GBN+OFE did particularly poorly in the 5 domains SATIMAGE, SEGMENT, SOYBEAN-LARGE, VEHICLE and WAVEFORM-21 — *e.g.*, on WAVEFORM-21 the accuracy of GBN+OFE learning is 0.658, compared to 0.759 from NB+OFE. See Figure 10. Moreover, we noticed that for those 5 domains (where 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 is because the small quantity of instances here is not sufficient for POWERCONSTRUCTOR to produce good GBN structures.

Over the remaining 20 benchmark datasets, we found that GBN+OFE is a significantly better classifier than NB+OFE: here, GBN+OFE $\Leftarrow_{(p<0.036)}$ NB+OFE. Moreover, GBN+OFE is comparable to TAN+OFE here; GBN+OFE $\leftarrow_{(p<0.4)}$ TAN+OFE. That is, for these 20 "good datasets" (where presumably the data is sufficient), POWERCONSTRUCTOR can produce a structure that is a reasonable model of the distribution, which can be used to produce an effective classifier.

When we focus on only these "good 20 datasets", we found that GBN+ELR was about the same as TAN+ELR and NB+ELR: GBN+ELR $\leftarrow_{(p<0.5)}$ NB+ELR and TAN+ELR $\leftarrow_{(p<0.18)}$ GBN+ELR. This is consistent with the earlier obseration that TAN+ELR was only comparable with NB+ELR.

Finally, we note that x+ELR remained slightly better than x+OFE, even for these "bad 5" databases; see Figure 10. In particular, ELR produced as good a classifier as OFE even when the GBN structure was lousy — *i.e.*, GBN+ELR $\leftarrow_{(p<0.1)}$ GBN+OFE.

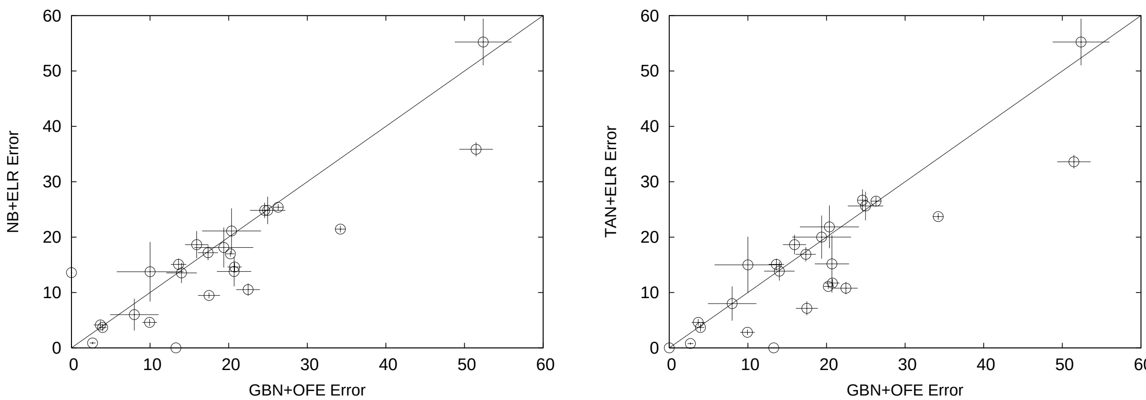


Figure 11: Comparing (a) GBN+OFE vs NB+ELR; (b) GBN+OFE vs TAN+ELR (over all 25 datasets)

Table 4: Comparison of Different Parameter Learners — *Complete* (UCI) Data; all 25 datasets

	GBN+ELR	GBN+OFE	TAN+ELR	TAN+OFE	NB+ELR
GBN+OFE	$\uparrow$ (0.2)				
TAN+ELR	$\Leftarrow$ (0.008)	$\Leftarrow$ (0.008)			
TAN+OFE	$\Leftarrow$ (0.04)	$\Leftarrow$ (0.02)	$\uparrow$ (0.12)		
NB+ELR	$\Leftarrow$ (0.01)	$\Leftarrow$ (0.03)	$\uparrow$ (0.2)	$\uparrow$ (0.45)	
NB+OFE	$\uparrow$ (0.36)	$\uparrow$ (0.46)	$\Uparrow$ (0.006)	$\Uparrow$ (0.002)	$\Uparrow$ (0.005)

Legend: Each $\langle i, j\rangle$ entry consists of both an arrow that points to the superior learner (using a double arrow $\Uparrow$ or $\Leftarrow$ if this is significant, and a single arrow $\uparrow$ or $\leftarrow$ otherwise); and the associated p-value in parentheses.

5.2.3 GBN+OFE vs NB+OFE, TAN+OFE

One approach to learning a good belief net (classifier) is to first find a good structure, then instantiate this structure using the trivial OFE algorithm. The first step can be hard — *e.g.*, NP-hard if seeking the structure that maximizes the BIC score [CGH94].[11] Another approach, suggested by our analysis, is to use a simple structure, such as NB or TAN, but then spend resources finding the best parameters, using ELR.

We therefore compared GBN+OFE to NB+ELR (Figure 11) and found NB+ELR to be significantly better: NB+ELR $\Leftarrow_{(p<0.03)}$ GBN+OFE. Moreover, TAN+ELR is yet stronger: TAN+ELR $\Leftarrow_{(p<0.008)}$ GBN+OFE.

Of course, GBN+OFE is at a real disadvantage when GBN is poor; recall from §5.2.2 that even NB+OFE is better than GBN+OFE in this situation. This suggests considering only the "good 20" databases. Here we found that, while NB+ELR (resp., TAN+ELR) remained slightly better than GBN+OFE, this difference was no longer significant: NB+ELR $\leftarrow_{(p<0.35)}$ GBN+OFE and TAN+ELR $\leftarrow_{(p<0.17)}$ GBN+OFE.

Table 4 presents a succinct summary of the results on the UCI data, over all 25 datasets. (Note this repeats many of the results from the previous sections.) The rows and columns are ordered based on our expectation that most of the entries would be $\Uparrow$'s. The corresponding table that used only the "good 20" databases was similar; however, here all of the $\Leftarrow$ arrows become $\leftarrow$'s — *i.e.*, none of the unanticipated differences is significant. (See [Gre04] for details.)

We also explicitly compared x+ELR to x+OFE, where $x \in \{$NB, TAN, GBN$\}$ over all of the 25 datasets considered, and found that ELR was significantly better: x+ELR $\Leftarrow_{(p<0.0015)}$ x+OFE.

[11] This BIC is a generative measure. We suspect finding the best "discriminative structure" would be as difficult. See also the iterative methods used in [GD03] for this task.

Table 5: Comparison of Different Parameter Learners — *InComplete* (UCI) Data; all 25 datasets

	GBN+ELR	GBN+APN	GBN+EM	TAN+ELR	TAN+APN	TAN+EM	NB+ELR	NB+APN
GBN+APN	$\Uparrow$ (0.05)							
GBN+EM	$\uparrow$ (0.09)	$\uparrow$ (0.25)						
TAN+ELR	$\Leftarrow$ (0.015)	$\Leftarrow$ (0.007)	$\Leftarrow$ (0.012)					
TAN+APN	$\Leftarrow$ (0.01)	$\Leftarrow$ (0.005)	$\Leftarrow$ (0.009)	$\uparrow$ (0.32)				
TAN+EM	$\Leftarrow$ (0.015)	$\Leftarrow$ (0.006)	$\Leftarrow$ (0.01)	$\uparrow$ (0.25)	$\uparrow$ (0.5)			
NB+ELR	$\Leftarrow$ (0.02)	$\Leftarrow$ (0.01)	$\Leftarrow$ (0.015)	$\uparrow$ (0.4)	$\uparrow$ (0.32)	$\uparrow$ (0.3)		
NB+APN	$\uparrow$ (0.08)	$\uparrow$ (0.06)	$\Uparrow$ (0.04)	$\uparrow$ (0.07)	$\uparrow$ (0.06)	$\Uparrow$ (0.04)	$\Uparrow$ (0.025)	
NB+EM	$\uparrow$ (0.06)	$\Uparrow$ (0.05)	$\Uparrow$ (0.04)	$\uparrow$ (0.055)	$\Uparrow$ (0.05)	$\Uparrow$ (0.035)	$\Uparrow$ (0.015)	$\uparrow$ (0.2)

5.2.4 GBN+x vs other classifiers, with Incomplete data

This section investigates the effectiveness of learning the parameters for GBN structures, from *incomplete* training data. As POWERCONSTRUCTOR is designed for complete data, we actually built each of the structures using complete data. We did this once, using all of the available data.

To produce the data used for learning and evaluating the parameters, we then removed the values of each evidence attribute for each tuple, with probability 0.25 — so again we are dealing with MCAR data [LR87].

The overall results appear in Table 5 where again we expected the majority of the entries to be $\Uparrow$'s. Most importantly, we see that x+ELR was never significantly worse than x+APN, or x+EM: and moreover, it was sometimes significantly better. There were some slight surprises "across the structure classes"; *e.g.*, TAN+x appears uniformly better than GBN+y. (of course this is consistent with the case for complete data.) We were also intrigued that NB+ELR (read' "standard logistic regression") seemed to be significantly better than GBN+y.

Finally, we compared x+ELR to x+APN and to x+EM, for all $x \in \{$NB, TAN, GBN$\}$ on all of the 25 datasets, and found that ELR was significantly better than both: x+ELR $\Leftarrow_{(p<0.0015)}$ x+APN and x+ELR $\Leftarrow_{(p<0.0015)}$ x+EM.

5.3 Model is More Complex than Truth ($G > T$)

§5.1 focused on the common situation where G (the BN-structure being instantiated) is presumedly *simpler* than the "truth" — *e.g.*, we used naïve-bayes when there probably were dependencies between the attributes. This section considers the opposite situation, where we allow the model "more degrees of freedom" than the truth. As this is atypical, we can only consider artificial data.

In our first experiment, we attempt to learn the parameters for a naïve-bayes model, when the truth is $C \equiv E_1$ — *i.e.*, the other attributes $E_2, \ldots, E_k$ are each irrelevant. We focus on $k = 6$ and $k = 7$ attributes. When the data is complete, we used first OFE and then ELR to instantiate the parameters of a given NaïveBayes model. Figure 12(a) shows the learning curve as we increase the sample size, over 10 different runs. (Each run used its own training sample.) We see that NB+OFE is consistently slightly beter than NB+ELR: averaged over all of the run, this is significant at $p < 0.002$.

We also weakened the $C \equiv E_1$ condition, to simply require C be highly correlated with E_1. Using the same set-up show above, even when the correlation is only 0.96, we still found NB+OFE $\Leftarrow_{(p<0.001)}$ NB+ELR.

The second experiment "reverses" the situations shown in §5.1.4. Here, the truth corresponds to a naïve-bayes structure (with no dependencies between the evidence E_i variables, conditioned on the class variable), but we attempt to find the parameters for a "P_m-based structure" — *i.e.*, a TAN structure that links $E_1 \equiv E_2 \equiv \ldots \equiv E_m$. These results appear in Figure 12(b), again this is averaged over 10 runs. (This difference is not significant.)

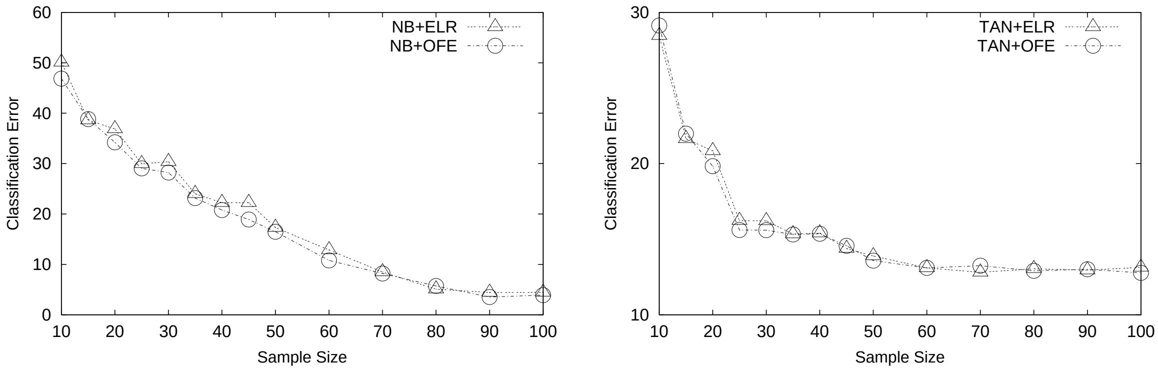


Figure 12: $G > T$ situations, complete data. (a) #1: Model is NB; Truth is $C \equiv E_1$; (b) #2: Model is TAN; Truth is NaïveBayes. (Each point is averaged over 10 runs)

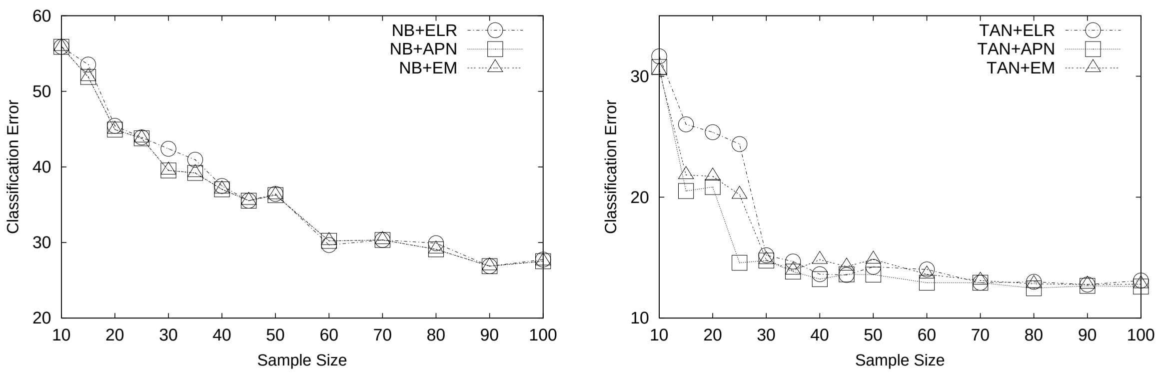


Figure 13: $G > T$ situations, *incomplete* data. (a) #1: Model is NB; Truth is $C \equiv E_1$; (b) #2: Model is TAN; Truth is NaïveBayes. (Each point is averaged over 10 runs.)

We next considered the same two situations, but in the *incomplete* data case. In particular, here we blocked a value of any entry with probability 0.2.

The results are shown in Figure 13; we see that the generative measures (NB+APN and NB+EM) dominated the discriminant NB+ELR: NB+APN $\Leftarrow_{(p<0.02)}$ NB+ELR and NB+EM $\Leftarrow_{(p<0.015)}$ NB+ELR. (Moreover, NB+EM $\Leftarrow_{(p<0.025)}$ NB+APN.)

In a nutshell, we observed that discriminative learning (here ELR) will often overfit in this "model is more complex than truth" situation, and so produce results that are often inferior to the results of a generative learner; this was true for both the complete and the incomplete data situations.

5.4 Discussion

This section has presented a large number of empirical results, all in the context of producing a good belief-net based classifiers for a fixed structure. The main take-home messages are...

- In the unusual situation where you are forced to use a model G that is more complex than the truth T, it is better to use the generative learners (OFE, APN, EM) — §5.3. However...
- in essentially all other complete-data situations, the discriminative learner ELR is at least as good, and often superior, to OFE. See in particular the x+ELR vs x+OFE entries in Table 4.

- We see similar results in the *incomplete* data case; here, the discriminative learner ELR is at least as good, and often superior, to APN and EM. See Table 5.
- While we typically found more expressive models produced better classifiers (*i.e.*, for each parameter-learner z, GBN+z was better than TAN+z, and TAN+z was better than NB+z), this was not universal. This comparison often goes the other way when the structure learner produces a low quality structure — *e.g.*, when POWERCONSTRUCTOR produce a structure that is effectively *worse* than NB!

  (Of course, this is not really POWERCONSTRUCTOR's fault: recall it was designed to find a structure that optimized *likelihood*, but our task is more related to optimizing *conditional likelihood*; see §1. It would be useful to instead use a learner that sought structures that optimized this measure, before then finding good parameters [NJ01].)
- In some cases, ELR is so much better than the associated generative parameter learner that x+ELR was better than y+OFE, where y is more expressive than x. Indeed, we found that ELR can do well with even the simplest of structure NB; note that NB+ELR was often superior to GBN+OFE!

Role of Structure Learner (POWERCONSTRUCTOR): One could argue that much of the problem with GBN stems from POWERCONSTRUCTOR, and that some of the problems would disappear if we used a different algorithm for learning the structure [Hec98, GSSK87]. Note, however, that we are comparing the parameter learning algorithms, and are seeking one that works effectively in all common situations — even if the structure is wrong. (And of course, any structure learner will work suboptimally in some situations.)

Why ELR Works Well: We found that ELR worked effectively in many situations, and it was especially advantageous (*i.e.*, typically better than the alternative ways to instantiate parameters) whenever the BN-structure was *incorrect* — *i.e.*, whenever it is not an 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 [Hec98, VG00]), or because they are considering only a restricted class of structures (*e.g.*, NaïveBayes [DH73], poly-tree [CL68, Pea88], Tree-Augmented Network [FGG97], etc.) which is not guaranteed to contain the correct structure.

To understand why a bad structure is problematic for OFE, recall that OFE is designed to produce the parameter values that yield the optimal likelihood value *for the structure G*, given the data S. However, if the structure G is incorrect, even the optimal-likelihood-for-G parameters might yield a fairly poor model of the true tuple distribution, which means it might return incorrect values for queries. By contrast, the ELR algorithm is not as constrained by the specific structure, and so may be able to produce parameters that yield fairly accurate answers, even if the structure is sub-optimal. (See the standard comparison between discriminative versus generative training, overviewed in Section 6 below.)

Computational Efficiency: ELR typically required a handful of iterations to converge for the small datasets, and dozens of iterations for the larger ones. APN and EM typically used slightly more iterations. Our current ELR implementation is in unoptimized JAVA code. Its time per iteration varied, from around 0.5 seconds per iteration for the smaller datasets through a few minutes for larger datasets on a PentiumIII-800MHz. Again, this is roughly comparable to the performance of the incomplete data algorithms, APN and EM.

These times are, of course, considerable more than required by OFE, which is arguably the most efficient possible algorithm. We are currently investigating whether there could be a more efficient algorithm for our task (computing parameters that optimize conditional likelihood) in the complete data case.

Tradeoff: Our results, in general, suggest an interesting tradeoff: Most BN-learners spend most of their time learning a near-optimal structure [CGH94], then use a simple algorithm (OFE) to fill in the CPtables. When the goal is classification accuracy, our empiricial studies suggest instead quickly producing trivial structures — such as NaïveBayes — then spending time learning good parameters, using ELR.

6 Related Results

There are a number of researchers providing techniques, and insights, related to learning belief nets. Much of their work focuses on learning the best *structure*, either for a general belief net, or within the context of some specific class of structures (*e.g.*, TAN-structures, or selective NaïveBayes); see [Hec98, Bun96] for extensive tutorials. By contrast, this paper suggests a way to learn the *parameters* for a given structure.

Most of those structure-learning systems also learn the parameters. Essentially all use the `OFE` algorithm here (Equation 12). This is well motivated in the generative situation, as these parameter values do optimize the likelihood of the data [CH92].

As noted earlier, our goal is different, as we are seeking the optimal *classifier* — *i.e.*, *discriminative learning*. While a perfect model of the underlying distribution would also be the optimal classifier, the converse is not true; *i.e.*, we are happy with parameters that yield a good classifier, even if those parameters do not reflect the true underlying distribution. That is, the eventual performance system will be expected to address a certain range of questions — *e.g.*, about the probability of cancer given gender, smoking habits, and other specified fetures. We consider our learner good if it produces parameters that provide appropriate answers to these questions, even if the overall distribution would return completely wrong answers to other (unasked) questions, *e.g.*, about the conditional probability of smoking given gender, etc.

There have been many other systems that also considered discriminant learning of belief nets [KMST99, JMJ00, FGG97, CG99]. These systems, however, focused on structure learning; and usually used `OFE` to instantiate the resulting parameters.

Our results also relate closely to the work on *discriminant learning of Hidden Markov Models (HMMs)* [SMK$^+$97, CJL92]. In particular, much of that work uses "Generatized Probabilistic Descent", which resembles our `ELR` system by descending along the derivative of the parameters, to maximize the conditional likelihood of the hypothesis (which typically are words) given the observations — which they call "Maximimum Mutual Information" criterion.

This relates directly to the large literature on *discriminant learning* in general; see [CS89, Jor95, Rip96]. One standard model is Linear Discriminant Analysis (LDA), which typically assumes $P(\,\mathbf{E}\,|\,C=c\,)$ is multivariate normal — *i.e.*, $P(\,\mathbf{E}\,|\,C=c\,) \sim \mathcal{N}(\mu_c, \Sigma)$ where each μ_c mean is dependent on the class $C=c$, and the covariance matrix Σ is the same for all classes. The LDA system then estimates the relevant $\{\mu_c,\ \Sigma,\ \hat{P}(\,C=c\,)\}$ parameters from a body of data, seeking the ones that maximize the likelihood of the data relevant to those parameters. Given these parameters, we can then use Bayes Rule to compute the conditional distribution of $P(\,C\,|\,\mathbf{E}=\mathbf{e}'\,)$ given new evidence $\mathbf{E}=\mathbf{e}'$.

While this approach attempts to estimate the entire $\langle C;\mathbf{E}\rangle$ joint distribution, Multiple Logistic Regression (MLR) estimates only the parameters explicitly associated with the conditional distribution, seeking the $\{\alpha_j, \beta_j\}$ parameters that maximize the conditional likelihood

$$P(\,C=c\,|\,\mathbf{E}=\mathbf{e}\,) \quad = \quad \frac{\exp(\alpha_c + \beta_c \cdot \mathbf{e})}{\sum_j \exp(\alpha_j + \beta_j \cdot \mathbf{e})}$$

Note this form corresponds to the multivariate conditions used by LDA; indeed, this form is appropriate whenever $P(\,\mathbf{E}\,|\,C\,)$ is in the exponential family.

We can view LDA as being generative (aka "causal" or "class-conditional" [Jor95], or "sampling" [Daw76]), as it is attempting to fit parameters for the entire joint distribution, while MLR is discriminant (aka "diagnostic", "predictive" [Jor95]), as it focuses only on the conditional probabilities. We can therefore identify LDA with (generative) `OFE`, and MLR with (discriminant) `ELR`.

Our results echo the common wisdom obtained by these prior analyses of discriminant systems. In particular, (1) accuracy: discriminative training typically produces more accurate classifiers than generative training (see the comparative studies throughout Section 5); (2) robustness: typically discriminative systems are more robust against incorrect models than generative one (see Section 5); (3) efficiency: typically generative is more efficient than discriminative (compare the efficient OFE with the iterative ELR). Due to the final point, many discriminative learners initialize their parameters based on generative (read "maximum-likelihood") estimates, especially as the latter are often "plug-in parameters" [Rip96]; our OFE-ELR algorithm incorporates this idea as well.

The work reported in this paper has significant differences, of course. First, we are dealing with a different underlying model, based on *discrete* variables (rather than continuous, say normally distributed, ones), in the context of a *specified belief net structure*, which corresponds to a given set of independency claims. We also describe the inherent computational complexity of this task, produce algorithms specific to our task, and provide empirical studies to demonstrate that our algorithms works effectively, given either complete or incomplete training data.

Finally, our companion paper [GGS97] also considers learning the parameters of a given structure towards optimizing performance on a distribution of queries. Our results here differ, as we are considering a different learning model: [GGS97] tries to minimize the squared-error score (a variant of the equation in Footnote 4) which is based on two different types of samples — one with tuples, to estimate $P(\,C\,|\,\mathbf{E}\,)$, and the other with queries, to estimate the probability of seeing each "What is $P(\,C\,|\,\mathbf{E}=\mathbf{e}\,)$?" query. By contrast, the current paper tries to minimize classification error (Equation 3) by seeking the optimal "conditional likelihood" score (Equation 4), wrt a single sample of labeled instances. Also, of course, our current paper includes new theoretical results, a different algorithm, and completely new empirical data.

7 Conclusions

7.1 Future Work

This paper investigtes the challenges of filling in the CPtables of a given BN-structure. While this is an important subtask, a general learner should be able to learn that structure as well — perhaps using *conditional likelihood* as the selection criterion; see [KMST99, JMJ00]. We plan to investigate ways to synthesize these approaches; see [GD03].

So far, the goal is classification accuracy. This measure is not as useful when the dataset is imbalanced (*i.e.*, many more instances of one class than another) or when different misclassifications have different penalties. Here, it is common to seek the classifier (here, set of parameters) that maximize the area-under-ROC-curve (AUC) measure [HT01]. We plan to explore this approach.

7.2 Contributions

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— see Equation 2. Our formal analyses then show that discriminative learning will often converge to a classifier superior to one learned generatively; moreover, we show that it will converge (to this better classifier) at essentially the same $O(\cdot)$ rate (ignoring polylog terms). The computational complexity is also comparable: In both cases, finding the optimal parameter values is in P when given a complete datasample, but the corresponding tasks given incomplete data appears harder. (We know that our specific task — finding the optimal CL parameters for a given general structure, from incomplete data — is NP-hard, but 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 it is possible that discriminative learning may be more efficient than 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 (NB), through less-trivial (TAN), to complex (ones learned by POWERCONSTRUCTOR). We also show that ELR works well when given *incomplete* training data. Our empirical evidence suggests that ELR can be inferior to the standard generative models only in the unusual situation where the model is more complex than the truth. In essentialy every other situations, however, we see that ELR is at least as good, and often better, than the other contenders. We also include a short study to explain why ELR is so effectivey, showing that it typically works better than generative methods when dealing with models that are less complicated than the true distribution, which is a very common 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.

For more information, including all of the data used for the experiments, see [Gre04].

Acknowledgements

We thank Corrine Cheng, Tom Dietterich, Adam Grove, Peter Hooper, Chris O'Brien, Dale Schuurmans and Lyle Ungar for their many helpful suggestions, and Jie Cheng for allowing us to use his POWERCONSTRUCTOR system for our GBN studies. RG and WZ were partially funded by NSERC; RG was also funded by Siemens Corporate Research and the Alberta Ingenuity Centre for Machine Learning; 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 *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 277–289. Morgan Kaufmann, 1991.

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

[BKRK97] John Binder, Daphne Koller, Stuart Russell, and Keiji 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. Technical report, Dept. Info. & Comp. Sci., Univ. Calif. at Irvine, 2000. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[Bun96] Wray Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 1996.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CG99] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 101–107. Morgan Kaufmann Publishers, August 1999.

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

[CGH94] David M. Chickering, Dan Geiger, and David Heckerman. Learning bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Research, November 1994.

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

[CJL92] W. Chou, B. Juang, and C. Lee. Segmental GPD training of HMM based speech recognizer. In *ICASSP*, volume 1, pages 473–476, 1992.

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

[Coo90] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990.

[CS89] D. R. Cox and E. J. Snell. *Analysis of Binary Data*. Chapman & Hall, London, 1989.

[Das97] Sanjoy Dasgupta. The sample complexity of learning fixed-structure bayesian networks. *Machine Learning*, 29:165–180, 1997.

[Daw76] A. P. Dawid. Properties of diagnostic data distributions. *Biometrics*, 32:647–658, 1976.

[DH73] Richard O. Duda and Peter E. 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 *Proc. 13th International Conference on Machine Learning*, 1996.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning Journal*, 29:131–163, 1997.

[FI93] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteeth International Joint Conference on Artificial Intelligence*, pages 1022–1027, San Francisco, CA, 1993. Morgan Kaufmann.

[GD03] D. Grossman and P Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA., 2003.

[GGS97] Russell Greiner, Adam Grove, and Dale Schuurmans. Learning Bayesian nets that perform well. In *Uncertainty in Artificial Intelligence*, 1997.

[Gre04] 2004. http://www.cs.ualberta.ca/$\sim$greiner/ELR.

[GSSK87] Clark Glymour, Richard Scheines, Peter Spirtes, and Kevin Kelly. *Discovering Causal Structure*. Academic Press, Inc., London, 1987.

[GZ02] Russell Greiner and Wei Zhou. Structural extension to logistic regression: Discriminant parameter learning of belief net classifiers. In *Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence (AAAI-02)*, pages 167–173, Edmonton, August 2002.

[Hec98] David E. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*, 1998.

[HT01] David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning Journal*, 45:171–186, 2001.

[JMJ00] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *NIPS2000*, 2000.

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

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

[KMST99] Petri Kontkanen, Petri Myllymäki, Tomi Silander, and Henry Tirri. On supervised selection of bayesian networks. In *UAI99*, pages 334–342, 1999.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, San Francisco, CA, 1995. Morgan Kaufmann.

[LR87] J. A. Little and D. B. 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] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

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

[NJ01] A. Y. Ng and M. I. Jordan. On discriminative versus generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, 2001.

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Methods in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

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

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

[SMK+97] R. Schlüter, W. Macherey, S. Kanthak, H. Ney, and L. Welling. Comparison of optimization methods for discriminative training criteria. In *Proc. EUROSPEECH'97*, pages 15–18, 1997.

[SSG+03] Bin Shen, Xiaoyuan Su, Russell Greiner, Petr Musilek, and Corrine Cheng. Discriminative parameter learning of general bayesian network classifiers. In *Proceedings of the Fifteenth IEEE International Conference on Tools with Artificial Intelligence (ICTAI-03)*, Sacramento, November 2003.

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

[VG00] Tim Van Allen and Russell Greiner. Model selection criteria for learning belief nets: An empirical comparison. In *ICML'00*, pages 1047–1054, 2000.

A Proofs

Proof of Theorem 1: As the set $\mathcal{BN}_{\Theta\succeq\gamma}(G)$ is uncountably infinite, we cannot simply apply the standard techniques for PAC-learning a finite hypothesis set. We can, however, partition this uncountable space into a finite number $L = L(K, \gamma, \epsilon)$ of sets, such that any two BNs within a partition have similar conditional log-likelihood scores. We can then, in essense, simultaneously estimate the scores of all members of $\mathcal{BN}_{\Theta\succeq\gamma}(G)$ if we collect enough query samples to estimate the score for one representative of each partition.

Now for the details: We prove below that, if the CPtables for two BNs $B^{(1)}, B^{(2)} \in \mathcal{BN}_{\Theta\succeq\gamma}(G)$ have similar CPtables $\Theta^{(1)} = \{\theta^{(1)}_{d_i|\mathbf{f}_i}\}_i$ and $\Theta^{(2)} = \{\theta^{(2)}_{d_i|\mathbf{f}_i}\}_i$, then they will have similar LCL-scores wrt any query; *i.e.*,

$$\text{if} \quad \left|\theta^{(1)}_{d_i|\mathbf{f}_i} - \theta^{(2)}_{d_i|\mathbf{f}_i}\right| \;\le\; \frac{\gamma\,\epsilon}{6\,K} \quad \text{then} \quad \forall c, \mathbf{e} \;\; |\ln(B^{(1)}(\,c\,|\,\mathbf{e}\,)) \;-\; \ln(B^{(2)}(\,c\,|\,\mathbf{e}\,))| \;\le\; \frac{\epsilon}{6} \tag{14}$$

This of course implies the same bound on the difference between their overall LCL-scores

$$|\mathrm{LCL}_k(\, B^{(1)}\,) - \mathrm{LCL}_k(\, B^{(2)}\,)| \quad \leq \quad \frac{\epsilon}{6}$$

for any distribution $\mathrm{LCL}_k(\,\cdot\,)$ — both for the "true" query distribution $\mathrm{LCL}(\,\cdot\,)$, and for the distribution associated with any empirical sample $\widehat{\mathrm{LCL}}^{(S)}(\,\cdot\,)$.

We therefore partition the $\mathcal{BN}_{\Theta\succeq\gamma}(G)$ space into $L = (\frac{6\,K}{\gamma\,\epsilon})^K$ disjoint sets (where any two BNs from any partition will have close CPtables), then define the set $R = \{B_i\}_i$ to contain one representative from each partition. We prove below that a sample S of size

$$M_{ELR}\left(\frac{\epsilon}{6}, \frac{\delta}{L}\right) \quad = \quad 2\left(\frac{3N\log\gamma}{\epsilon}\right)^2 \ln\frac{2L}{\delta} \tag{15}$$

is sufficient to estimate each of these single representatives to within $\epsilon/6$ of correct, with probability of error at most δ/L; *i.e.*, such that, for each i,

$$P\left[\,\left|\widehat{\mathrm{LCL}}^{(S)}(\,B_i\,) - \mathrm{LCL}(\,B_i\,)\right| > \frac{\epsilon}{6}\,\right] \quad < \quad \frac{\delta}{L}$$

But since there are L representatives, we have a total probability of at most $L\frac{\delta}{L} = \delta$ that *any* of the representative's scores are mis-estimated by more than $\epsilon/6$.

This means we have, in effect, estimated the scores on *any* $B \in \mathcal{BN}_{\Theta\succeq\gamma}(G)$ to within $\epsilon/2$: For any $B \in \mathcal{BN}_{\Theta\succeq\gamma}(G)$, let $B' \in R$ be the representative in Bs partition. Observe

$$\begin{array}{rcccccccc} |\widehat{\mathrm{LCL}}(\,B\,) - \mathrm{LCL}(\,B\,)| & \leq & |\widehat{\mathrm{LCL}}(\,B\,) - \widehat{\mathrm{LCL}}(\,B'\,)| & + & |\widehat{\mathrm{LCL}}(\,B'\,) - \mathrm{LCL}(\,B'\,)| & + & |\mathrm{LCL}(\,B'\,) - \mathrm{LCL}(\,B\,)| \\ & \leq & \epsilon/6 & + & \epsilon/6 & + & \epsilon/6 \\ & = & \epsilon/2 & & & & \end{array}$$

This means, in particular, that our estimate of the scores of both $\widehat{B}$ and B^* are within $\epsilon/2$, and so

$$\begin{array}{rcccccccc} \mathrm{LCL}(\,\widehat{B}\,) - \mathrm{LCL}(\,B^*\,) & \leq & |\mathrm{LCL}(\,\widehat{B}\,) - \widehat{\mathrm{CLL}}(\,\widehat{B}\,)| & + & \widehat{\mathrm{CLL}}(\,\widehat{B}\,) - \widehat{\mathrm{CLL}}(\,B^*\,) & + & |\widehat{\mathrm{CLL}}(\,B^*\,) - \mathrm{LCL}(\,B^*\,)| \\ & \leq & \epsilon/2 & + & 0 & + & \epsilon/2 \end{array}$$

To complete the proof, we need only prove Equations 14 and 15. For Equation 14: Consider the sequence of BNs $B_0, B_1, \ldots B_K$ where the first i of B_i's CPtables come from $B^{(1)}$, and the remaining from $B^{(2)}$ — *i.e.*,

$$B_i \quad \sim \quad \{\theta^{(1)}_{d_1|\mathbf{f}_1},\ \ldots,\ \theta^{(1)}_{d_i|\mathbf{f}_i},\ \theta^{(2)}_{d_{i+1}|\mathbf{f}_{i+1}},\ \ldots,\ \theta^{(2)}_{d_K|\mathbf{f}_K}\}$$

Now observe

$$|\ln(B^{(1)}(\,c\,|\,\mathbf{e}\,)) - \ln(B^{(2)}(\,c\,|\,\mathbf{e}\,))| \quad \leq \quad \sum_{i=1}^{K} |\ln(B_i(\,c\,|\,\mathbf{e}\,)) - \ln(B_{i-1}(\,c\,|\,\mathbf{e}\,))|$$

and each $|\ln(B_i(\,c\,|\,\mathbf{e}\,)) - \ln(B_{i-1}(\,c\,|\,\mathbf{e}\,))|$ is based on changing a single CPtable entry. We therefore need only show $|\ln(B_i(\,c\,|\,\mathbf{e}\,)) - \ln(B_{i-1}(\,c\,|\,\mathbf{e}\,))| \leq \frac{\epsilon}{6\,K}$. For any value of $z = \theta_{d_i|\mathbf{f}_i}$, let $f(z) = \ln(B[z](\,c\,|\,\mathbf{e}\,))$, where $B[z]$ be the BN whose first $i-1$ CPtable entries come from $B^{(1)}$, whose final $K-i-1$ entries come from $B^{(2)}$, and whose i^{th} CPtable entries is z; hence $f(\theta^{(1)}_{d_i|\mathbf{f}_i}) = \ln(B_i(\,c\,|\,\mathbf{e}\,))$, and $f(\theta^{(2)}_{d_i|\mathbf{f}_i}) = \ln(B_{i+1}(\,c\,|\,\mathbf{e}\,))$. As this function is continuous, we know that

$$|f(a) - f(b)| \quad = \quad \frac{\partial f(z)}{\partial z}[b-a]$$

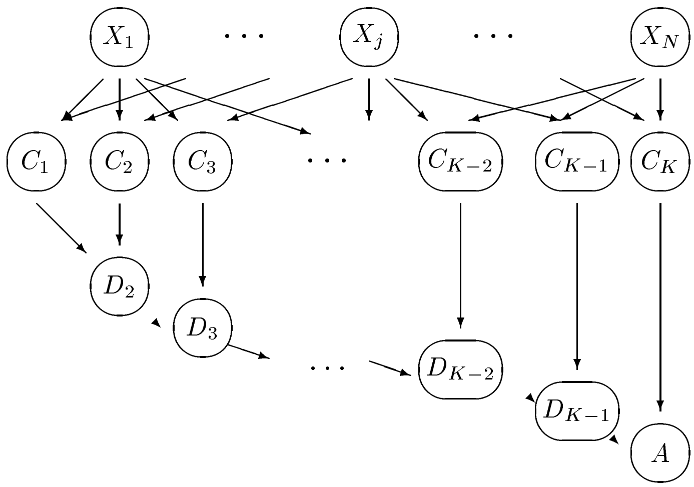


Figure 14: Belief Net structure for any SAT problem [Coo90]

for some $z \in [a, b]$. As $f(z) = \ln(B[z](\, c, \mathbf{e}\,)) - \ln(B[z](\, \mathbf{e}\,))$, we see that

$$\begin{aligned}
\frac{\partial f(z)}{\partial z} &= \tfrac{1}{B[z](\,c,\mathbf{e}\,)} B[z](\, c, \mathbf{e} \,|\, d_i, \mathbf{f}_i\,) \times B[z](\, \mathbf{f}_i\,) \; - \; \tfrac{1}{B[z](\,\mathbf{e}\,)} B[z](\, \mathbf{e} \,|\, d_i, \mathbf{f}_i\,) \times B[z](\, \mathbf{f}_i\,) \\
&= \tfrac{1}{z}[B[z](\, d_i, \mathbf{f}_i \,|\, c, \mathbf{e}\,) - B[z](\, d_i, \mathbf{f}_i \,|\, \mathbf{e}\,)]
\end{aligned}$$

which means that $|\frac{\partial f(z)}{\partial z}| \leq 1/z \leq 1/\gamma$. (The second inequality follows from the assumption that we are only considering $B \in \mathcal{BN}_{\Theta \succeq \gamma}(G)$.) Hence,

$$\begin{aligned}
|\ln(B_{i+1}(\, c \,|\, \mathbf{e}\,)) - \ln(B_i(\, c \,|\, \mathbf{e}\,))| &= |f(\theta^{(2)}_{d_i|\mathbf{f}_i}) - f(\theta^{(1)}_{d_i|\mathbf{f}_i})| \\
&\leq \tfrac{1}{\gamma} \times |\theta^{(2)}_{d_i|\mathbf{f}_i} - \theta^{(1)}_{d_i|\mathbf{f}_i}| \quad \leq \quad \tfrac{1}{\gamma} \times \tfrac{\gamma\,\epsilon}{6\,K} \quad = \quad \tfrac{\epsilon}{6\,K}
\end{aligned}$$

To prove Equation 15: Observe first that the probability of any event must be at least the product of N CPtable entries, and hence $B(\, c\,) \geq \gamma^N$ for any c and any $B \in \mathcal{BN}_{\Theta \succeq \gamma}(G)$. This means the value of $-\ln(B(\, c \,|\, \mathbf{e}\,)$, and hence $\text{LCL}_k(\, B\,)$ for any distribution sq, is between 0 and $-N \ln \gamma$.

As the queries $q = P(\, c\,,\, \mathbf{e}\,)$ are drawn at random from a stationary distribution, we can view the quantity $\ln B(\, q\,)$ as an iid random value, whose range is $[0,\, -N \ln \gamma]$ and whose expected value is $\text{LCL}(\, B\,)$. Hoeffding's Inequality bounds the chance that the empirical average score after M iid examples (here $\widehat{\text{LCL}}^{(S)}(\, B\,)$) will be far away from the true mean $\text{LCL}(\, B\,)$:

$$P(\,|\widehat{\text{LCL}}^{(S)}(\, B\,) - \text{LCL}(\, B\,)| > \frac{\epsilon}{6}\,) \quad < \quad 2 \exp -2M((\epsilon/6)/N \ln \gamma)^2\,. \tag{16}$$

Here, we want the right-hand-side to be under δ/L, which requires $M = M(\epsilon, \delta) = 2\left(\frac{3\,N\,\ln\gamma}{\epsilon}\right)^2 \ln(\frac{2L}{\delta})$. ∎

Proof of Theorem 3: We reduce 3SAT to our task, using a construction similar to the one in [Coo90]: Given any 3-CNF formula $\varphi \equiv \bigwedge C_i$, where each $C_i \equiv \bigvee \pm X_{ij}$, we construct the network shown in Figure 14, with one node for each variable X_i and one for each clause C_j, with an arc from X_i to C_j whenever C_j involves X_i — e.g., if $C_1 = x_1 \vee \bar{x}_2 \vee x_3$ and $C_2 = \bar{x}_1 \vee \bar{x}_3 \vee x_4$, then there are links to C_1 from each of X_1, X_2 and X_3, and to C_2 from X_1, X_3 and X_4. In addition, we include $K - 1$ other boolean nodes, $\{D_2, \ldots, D_{K-1}, A\}$, where D_j is the child of D_{j-1} and C_j, where D_1 is identified with C_1, and A is used for D_K.

X_1	X_2	X_3	X_4	$\cdots$	X_n	A
0	1	0				0
0		0	1			0
	$\vdots$					$\vdots$
0		1		1		0
						1

Figure 15: Queries used in Proof of Theorem 3

Here, we intend each C_i to be true if the assignment to the associated variables X_{i1}, X_{i2}, X_{i3} satisfies C_i; and A corresponds is the conjunction of those C_i variables. We do this using (all-but-the-final) instances in Figure 15. There is one such instance for each clause, with exactly the assignment (of the 3 relevant variables) that falsifies this clause. Hence, the first line corresponds to $C_1 \equiv x_1 \vee \bar{x}_2 \vee x_3$. The "label" of each instance always corresponds to the single variable A.

We now prove, in particular, that

> There is a set of parameters for the structure in Figure 14, producing the $\widehat{\text{LCL}}^{()}(\ )$-score, over the queries in Figure 15, of 0
>
> *iff*
>
> there is a satisfying assignment for the associated φ formula.

$\Leftarrow$: Just set each C_i to be the disjunction of the associated X_{i1}, X_{i2}, X_{i3} variables (its parents), with the appropriate sense. Eg, using $C_1 \equiv x_1 \vee \bar{x}_2 \vee x_3$, then C_1's CPtable would be

x_1	x_2	x_3	$P(+c_1 \mid x_1, x_2, x_3)$
0	0	0	1.0
0	0	1	1.0
0	1	0	1.0
0	1	1	0.0
1	0	0	1.0
1	0	1	1.0
1	1	0	1.0
1	1	1	1.0

.

Similarly set the CPtables for the D_j to correspond to the conjunction of its 2 parents $D_j = D_{j-1} \wedge C_j$; *e.g.*,

D_4	C_5	$P(+d_5 \mid D_4, C_5)$
0	0	0.0
0	1	0.0
1	0	0.0
1	1	1.0

.

Finally, set X_i to correspond to the satisfying assignment; *i.e.*, if $X_1 = 1$, then

$P(+x_1)$
1.0

; and if *i.e.*, if $X_4 = 0$, then

$P(+x_4)$
0.0

. Note that these CPtable values satify all $k+1$ of the labeled instances.

$\Leftarrow$: Here, we assume there is no satisfying assignment. Towards a contradiction, we can assume that there is a 0-LCL set of CPtable entries. This means, in particular, that $P(+a \mid x_{i1}, x_{i2}, x_{i3}) = 0$, where x_{i1}, x_{i2}, x_{i3} correspond to the assignment that violates the ith constraint. (*E.g.*, for $C_1 \equiv x_1 \vee \neg x_2 \vee x_3$, this would be $X_1 = 0$, $X_2 = 1$, $X_3 = 0$.)

Now consider the final labled instance, $P(a)$. As there is no satisfying assignment, we know that each assignment $\mathbf{x}$ violates at least one constraint. For notation, let $\gamma^{\mathbf{x}}$ refer to one of these violations (say the one with the smallest index). So if $\mathbf{x} = \langle 0, 1, 0, \ldots \rangle$, then $\gamma^{\langle 0,1,0,\ldots\rangle} = \langle X_1 = 0,\ X_2 = 1,\ X_3 = 0 \rangle$ corresponds to the violation of the first constraint C_1. We also let $\beta^{\mathbf{x}}$ refer to the rest of the assignment.

Now observe

$$\begin{aligned} P(\,+a\,) &= \textstyle\sum_{\mathbf{x}} P(\,+a,\ \mathbf{x}\,) \\ &= \textstyle\sum_{\mathbf{x}} P(\,+a \,|\, \gamma^{\mathbf{x}}\,) \cdot P(\,\gamma^{\mathbf{x}}\,) \cdot P(\,\beta^{\mathbf{x}} \,|\, +a,\ \gamma^{\mathbf{x}}\,) \\ &= \textstyle\sum_{\mathbf{x}} \quad 0 \quad \cdot P(\,\gamma^{\mathbf{x}}\,) \cdot P(\,\beta^{\mathbf{x}} \,|\, +a,\ \gamma^{\mathbf{x}}\,) \quad = \quad 0 \end{aligned}$$

which shows that the final instance will be mislabeled. This proves that there can be no set of CPtable values that produce 0 LCL-score when there are no satisfying assignments. ∎

Proof of Proposition 4: Below, we will use $P(\,\chi\,)$ to refer to $B(\,\chi\,)$, the value the belief net B will assign to the χ event. In general, for any assignment Z,

$$P(\,Z\,) \quad = \quad \sum_{\mathbf{f}'} \sum_{d'} P(\,Z \,|\, D{=}d', \mathbf{F}{=}\mathbf{f}'\,)\; P(\,D{=}d' \,|\, \mathbf{F}{=}\mathbf{f}'\,)\; P(\,\mathbf{F}{=}\mathbf{f}'\,) \tag{17}$$

As we assume the different CPtable rows are independent, and $\mathbf{F}$ are the parents of D, this means

$$\frac{\partial\, P(\,B \,|\, Z\,)}{\partial\, \beta_{d|\mathbf{f}}} \quad = \quad \sum_{d'} P(\,Z \,|\, d', \mathbf{f}\,)\; \frac{\partial\, P(\,d' \,|\, \mathbf{f}\,)}{\partial\, \beta_{d|\mathbf{f}}} P(\,\mathbf{f}\,)$$

Recalling $\theta_{d|\mathbf{f}} = P(\,d \,|\, \mathbf{f}\,) = e^{\beta_{d|\mathbf{f}}} / \sum_{d'} e^{\beta_{d'|\mathbf{f}}}$, observe that $\frac{\partial\, P(\,d \,|\, \mathbf{f}\,)}{\partial\, \beta_{d|\mathbf{f}}} = \theta_{d|\mathbf{f}}(1 - \theta_{d|\mathbf{f}})$, and when $d \neq d'$, $\frac{\partial\, P(\,d' \,|\, \mathbf{f}\,)}{\partial\, \beta_{d|\mathbf{f}}} = -\,\theta_{d|\mathbf{f}}\, \theta_{d'|\mathbf{f}}$. This means $\frac{\partial\, P(\,Z\,)}{\partial\, \beta_{d|\mathbf{f}}} = P(\,Z, d, \mathbf{f}\,) - \theta_{d|\mathbf{f}} P(\,Z, \mathbf{f}\,)$.

Hence, as $\ln P(\,c \,|\, \mathbf{e}\,) = \ln P(\,c, \mathbf{e}\,) - \ln P(\,\mathbf{e}\,)$,

$$\begin{aligned} \frac{\partial\, \ln P(\,c \,|\, \mathbf{e}\,)}{\partial\, \beta_{d|\mathbf{f}}} &= \frac{\partial\, \ln P(\,c, \mathbf{e}\,)}{\partial\, \beta_{d|\mathbf{f}}} - \frac{\partial\, \ln P(\,\mathbf{e}\,)}{\partial\, \beta_{d|\mathbf{f}}} \\ &= \frac{1}{P(\,c, \mathbf{e}\,)} \frac{\partial\, P(\,c, \mathbf{e}\,)}{\partial\, \beta_{d|\mathbf{f}}} - \frac{1}{P(\,\mathbf{e}\,)} \frac{\partial\, P(\,\mathbf{e}\,)}{\partial\, \beta_{d|\mathbf{f}}} \\ &= \frac{1}{P(\,c, \mathbf{e}\,)} [P(\,c, \mathbf{e}, d, \mathbf{f}\,) - \theta_{d|\mathbf{f}} P(\,c, \mathbf{e}, \mathbf{f}\,)] - \frac{1}{P(\,\mathbf{e}\,)} [P(\,\mathbf{e}, d, \mathbf{f}\,) - \theta_{d|\mathbf{f}} P(\,\mathbf{e}, \mathbf{f}\,)] \\ &= [P(\,d, \mathbf{f} \,|\, c,\ \mathbf{e}\,) - P(\,d, \mathbf{f} \,|\, \mathbf{e}\,)] - \theta_{d|\mathbf{f}}\, [P(\,\mathbf{f} \,|\, c,\ \mathbf{e}\,) - P(\,\mathbf{f} \,|\, \mathbf{e}\,)] \end{aligned}$$

∎