Bayesian Network Learning for Rare Events

Samuel G. Gerssen, Leon J. M. Rothkrantz

Department of Information Technology and Systems
Delft University of Technology
samuel@ch.tudelft.nl, l.j.m.rothkrantz@ewi.tudelft.nl

March 5, 2004

Abstract

Parameter learning from data in Bayesian networks is a straightforward task. The average number of observed occurrences is stored in a conditional probability table, from which future predictions can be calculated. This method relies heavily on the quality of the data. A data set with 'rare events' will not yield statistically reliable estimates. Bayesian networks allow prior and posterior learning. In this paper, new prior assessment techniques are introduced to obtain stable priors for a conditional probability table. These learning algorithms are implemented and tested, and the results will be presented.

1 Introduction

A Bayesian network [8, 10] is a directed acyclic graph with each node representing a variable and each arc representing a causal relation between two variables. Variables are characterized by a probability distribution for each value. The probability distribution of each node is influenced by the states (for discrete nodes) or values (for continuous nodes) of the parent node. The conditional probabilities of a node are stored in a conditional probability table (CPT). The CPT is needed to calculate any conditional probability in the model, inference [5]. The size of the CPT depends on the number of states (s), the number of parents (p), and the number of parent states (s_p) in the following way:

$$\texttt{size}(CPT) = s \cdot (s_p)^p \tag{1}$$

For every possible combination of parent states, there is an entry listed in the CPT. Notice that for a large number of parents the CPT will expand drastically. Assume the variables in the Bayesian network illustrated in Figure 1 are binary.

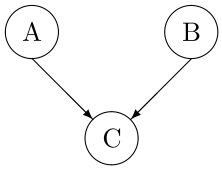

Figure 1: Example Bayesian network

The conditional probability table of node C will have eight entries, with four degrees of freedom. The CPT for node C is given in Table 1.

Table 1: CPT for node C

$p(C \mid AB)$	c	$\bar{c}$
a, b	θ_{ab}	$1-\theta_{ab}$
$a, \bar{b}$	$\theta_{a\bar{b}}$	$1-\theta_{a\bar{b}}$
$\bar{a}, b$	$\theta_{\bar{a}b}$	$1-\theta_{\bar{a}b}$
$\bar{a}, \bar{b}$	$\theta_{\bar{a}\bar{b}}$	$1-\theta_{\bar{a}\bar{b}}$

The number of degrees of freedom of a CPT with number of states (s), number of parents (p), and number of parent states (s_p) is $s_p^p \cdot (s-1)$. Values for θ can be obtained from the data by parameter learning. θ_{ab} is the number of occurrences (c, ab) divided by the number of occurrences (ab), or $(c, ab) + (\bar{c}, ab)$. In general, if for a variable$X$, with states $x_1 \dots x_N$, $\mathbf{p_i}$ is the i-th combination of parent states, then θ_{ij} denotes the probability of x_j given $\mathbf{p_i}$. Let $o(\mathbf{p_i}, x_j)$ be the number of observations $(\mathbf{p_i}, x_j)$ in the data set. θ_{ij} can now be calculated as follows:

$$\theta_{ij} = \frac{o(\mathbf{p_i}, x_j)}{\sum_{k=1}^{N} o(\mathbf{p_i}, x_k)} \tag{2}$$

Parameter learning is basically counting the number of observations of a specific event. Notice that the accuracy of θ heavily depends on the number of observed events. A large number of observations will result in a more accurate estimate than just a small sample. Therefore, large data sets usually provide good, stable models.

Unfortunately many real-world data sets are imbalanced; some states of the response variable may be dozens to thousands of times less likely than other states. This is the case in, for example, customer bankruptcies in banks, international armed conflicts, or epidemiological infections. Thus the effect of having a large data set available is canceled by the fact that it contains only a small number of 'interesting' records. Therefore, a model based on such data may be severely biased or highly unstable.

Outline The remainder of this article is organized as follows. Section 2 gives account of previous work. The new theory is described in Section 3. Results are listed in Section 4, and Section 5 gives the conclusions.

2 Previous Work

Rare event problems occur in generalized linear models, such as logistic regression, but also in models learned from data, such as neural networks and Bayesian networks. Especially in the case of generalized linear models, techniques such as prior correction and weighting have been developed to discard most of the 'uninteresting' part of the data set without much performance loss [4]. Additionally there are ways of reducing bias and variance [9]. In Bayesian networks, small sample problems are usually solved by setting a good prior distribution [1, 6], based on expert knowledge. However, if this expert knowledge is not available, an acceptable prior, based on the data, needs to be set. [7] proposes noisy-OR to obtain priors. This approach uses cutting to shift from prior to posterior estimate. Naive Bayes is a very good classifier, as described in [2].

Parameter learning is explained in detail by David Heckerman in [3]. The Bayesian approach calculates the belief of a probability θ. Beliefs of probabilities of binary variables are assumed to have a Beta distribution with parameters $\alpha > 0$ and $\beta > 0$:

$$\begin{aligned} p(\theta) &= \mathrm{Beta}(\theta \mid \alpha, \beta) \\ &\equiv \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} \end{aligned} \tag{3}$$

The Beta distribution is very convenient in a couple of ways. Suppose the belief of a probability needs to be updated after an observation. If the (binary) observation was an event, then the probability distribution is multiplied by θ. However, if a nonevent is observed, the probability distribution of θ is multiplied by $(1-\theta)$. After the multiplication, the distribution needs to be normalized again. By definition, the resulting distribution is again Beta, with parameters $(\alpha + x_{N+1}, \beta + (1 - x_{N+1}))$. Therefore, the parameters α and β represent the number of observed events and the number of observed nonevents, respectively.

The second advantage of Beta distributions is that calculation of the expectation is:

$$\int \theta \ \mathrm{Beta}(\alpha, \beta) d\theta = \frac{\alpha}{\alpha + \beta} \tag{4}$$

Notice that this formula is equivalent to the parameter assessment in equation 2. Because distributions of CPT parameters are assumed to be Beta, the learning process merely comes down to counting the number of events and nonevents. These values are the parameters in a Beta distribution, from which the expectation is calculated using equation 4.

For variables with multiple states, the generalized form of the Beta distribution is used: the Dirichlet distribution. A Dirichlet distribution for a variable with N states and parameters $(\alpha_1 \dots \alpha_N)$ is defined as:

$$\begin{aligned} p(\theta) &= \mathrm{Dir}(\theta \mid \alpha_1 \dots \alpha_N) \\ &\equiv \frac{\Gamma(\sum_{i=1}^{N} \alpha_i)}{\prod_{i=1}^{N} \Gamma(\alpha_i)} \prod_{i=1}^{N} \theta_i^{\alpha_i - 1} \end{aligned} \tag{5}$$

Inference in a Bayesian network is the calculation of conditional probabilities, given the probabilities in a CPT. Inference is based on two rules. The first one is Bayes' rule, which is defined as:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \tag{6}$$

The second rule is the expansion rule, which is defined for binary variables ($p(\bar{x}) = 1 - p(x)$) as:

$$\begin{aligned} p(x) =& p(x \mid y) \cdot p(y) + p(x \mid \bar{y}) \cdot p(\bar{y}) \\ p(x) =& p(x \mid yz) \cdot p(yz) + p(x \mid \bar{y}z) \cdot p(\bar{y}z) + \\ & p(x \mid y\bar{z}) \cdot p(y\bar{z}) + p(x \mid \bar{y}\bar{z}) \cdot p(\bar{y}\bar{z}) \\ =& \sum_{Y} \sum_{Z} p(x \mid yz) \cdot p(yz) \\ & \text{etc..} \end{aligned} \tag{7}$$

Using these two rules, in the example in Figure 1, $p(a \mid c)$ can now be calculated:

$$\begin{aligned} p(a \mid c) =& \frac{p(a) \cdot p(c \mid a)}{p(c)} \\ =& \frac{p(a) \cdot \sum_B p(c \mid ab)p(b)}{\sum_A \sum_B p(c \mid ab)p(ab)} \end{aligned} \tag{8}$$

Inference will be used for the assessment of naive Bayes priors.

3 CPT Calculation

Deriving CPT parameters using a prior-posterior approach consists of three stages:

1. prior assessment
2. posterior assessment
3. merging

The posterior assessment is equivalent with CPT parameter learning, shown in equation 2, and will not be discussed here. First, the merging method is described and then the prior assessment.

3.1 Merging

The merging process of priors and posteriors can be handled a in couple of ways. [7] uses a cut value for the number of observations, in that paper called smoothing. If, in the data set, a combination of parent states $\mathbf{p_i}$ occurs less than the cut value, the prior will be used in the CPT. Otherwise, the posterior will be used. Let the prior for θ_{ij} be p_{ij}, the posterior r_{ij}, and the cut value c, the smoothing merging is defined as:

$$\theta_{ij} = \begin{cases} p_{ij} & \text{if } \sum_{k=1}^{N} o(\mathbf{p_i}, x_k) < c \\ r_{ij} & \text{otherwise} \end{cases} \tag{9}$$

Substitution of equations 2 and 9 yields:

$$\theta_{ij} = \begin{cases} p_{ij} & \text{if } \sum_{k=1}^{j} o(\mathbf{p_i}, x_k) < c \\ \frac{o(\mathbf{p_i}, x_j)}{\sum_{k=1}^{N} o(\mathbf{p_i}, x_k)} & \text{otherwise} \end{cases} \tag{10}$$

In this paper, a gentle transition is proposed, by weighting the priors. The priors will receive an integer value, weight w, which is equivalent to a number of observations. The resulting CPT entries will be a weighted average of the priors and the posteriors:

$$\theta_{ij} = \frac{(w \cdot p_{ij}) + o(\mathbf{p_i}, x_j)}{w + \sum_{k=1}^{j} o(\mathbf{p_i}, x_k)} \tag{11}$$

If $o(\mathbf{p_i}, x_j)$ is low, the CPT value will mainly be based on the prior, however, if $(\mathbf{p_i}, x_j)$ is often observed in the data set, the data will influence the CPT value more than the prior. Basically this effect is the same as smoothing, only the transition is more gentle.

In smoothing and weighting, the values for c and w, respectively, need to be set. The tests in Section 4 show that these parameters have the greatest effect if a value between 5 and 80 is chosen, preferably between 10 and 40. If the values are too low, the procedure will be similar to normal CPT learning. If the values are too high, the posteriors will have very little effect. Notice that for a model with extremely good priors, the weight or cut value should be very high.

3.2 Prior Assessment

A CPT contains very detailed information about conditional probabilities for all possible parent states. As shown in the introduction, the number of degrees of freedom expands rapidly if the number of parents and the number of parent states increase. A stable prior should therefore be derived from a low number of degrees of freedom, but as accurate as possible. [7] propose the usage of noisy-MAX priors, because it is a good modeling technique for rare events. Noisy-MAX makes the assumption that the state of the response variable is a logical combination of the states of the input parameters. In practice, for a lot of data sets, noisy-MAX shows high performance, because in many cases, an effect is an addition or multiplication of the causes. An example of a binary noisy-MAX (noisy-OR) model is given in Figure 2.

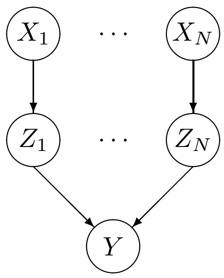

Figure 2: Noisy-MAX network

Aside from the input variables $X_1 \ldots X_N$ and response variable Y, there is a set of inhibitor nodes $Z_1 \ldots Z_N$. X_i can cause Z_i to be present with a probability c_i, but absence of X_i always implies absence of Z_i. The CPT of each Z_i is given in Table 2.

The CPT of node Y is similar to a logical MAX gate (Table 3).

Naive Bayes is in many cases a superior classifier [2]. Just as noisy-MAX, naive Bayes has a relatively low number of degrees of freedom. The performance as a classifier is slightly better than noisy-MAX. The network structure is completely opposite from a regular CPT network. It may be against intuition to use priors from an opposite network. In the prior assessment, the causalities are less relevant than how $p(y \mid \mathbf{p_i})$ is calculated.

Table 2: CPT of nodes $Z_1 \ldots Z_N$

$p(Z_i \mid X_i)$	z_i	$\bar{z}_i$
x_i	c_i	$1 - c_i$
$\bar{x}_i$	0	1

Table 3: CPT of node Y

$p(Y \mid Z_1 \ldots Z_n)$	y	$\bar{y}$
$z_1 \ldots z_n$	1	0
$\ldots$	$\ldots$	$\ldots$
$z_1, \bar{z_2} \ldots \bar{z_n}$	1	0
$\bar{z_1} \ldots \bar{z_n}$	0	1

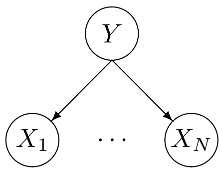

Figure 3: Naive Bayes network

A naive Bayes network (Figure 3) makes very unrealistic assumptions about causality. It assumes the predictive variables to be dependent on the response variable. Also, it assumes conditional independence between the predictive variables, meaning that given the value of Y, $X_1 \ldots X_N$ are independent. Aside from these assumptions, a naive Bayes model is a very powerful classifier.

Inference in a naive Bayes network is not as straightforward as in a CPT network. A network with predictive variables $X_1 \ldots X_N$ and response variable Y with states $y_1 \ldots y_M$ will have the probabilities $p(y \mid \mathbf{p_i})$ listed in the CPT. In a naive Bayes network, these values need to be calculated as follows:

$$\begin{aligned} p(y \mid \mathbf{p_i}) &= \frac{p(\mathbf{p_i} \mid y) \cdot p(y)}{p(\mathbf{p_i})} \\ &= \frac{\prod_i p(x_i \mid y) \cdot p(y)}{\prod_i p(x_i)} \\ &= \frac{\prod_i p(x_i \mid y) \cdot p(y)}{\prod_i \sum_j (p(x_i \mid y_j) \cdot p(y_j))} \end{aligned} \tag{12}$$

The values for $p(x_i \mid y)$ and $p(y)$ can be obtained by parameter learning in the naive Bayes network, described in Section 2. The p_{ij} from equation 11 can be replaced by the right part of equation 12. Now the formula for calculation of the CPT values is complete.

Table 4: Performance of models

MODEL	GINI
Naive Weighting W=1	76.4%
Naive Weighting W=5	77.0%
Naive Weighting W=10	77.1%
Naive Weighting W=20	77.2%
Naive Weighting W=40	77.3%
Naive Weighting W=80	77.3%
Noisy-MAX Weighting W=10	71.4%
Naive Smoothing C=10	75.4%
Naive Smoothing C=20	76.6%
Naive Smoothing C=40	77.0%
Noisy-MAX Smoothing C=20	71.4%

4 Results

The method above was implemented and tested on a bankruptcy data set from a bank. The bankruptcies were rare (around 2% of all records).

As a performance measure for scoring the models, the GINI index was used. In a graph where a curve is plotted as the cumulative bankrupt percentage of clients against the total percentage of clients when they are sorted on risk (low risk on the left, high risk on the right), the GINI index is defined as the area between the diagonal and the curve (the Lorentz curve) divided by the total area under the diagonal. It has a range between -1 and 1, or -100% to 100%. High scores indicate good models.

The GINI index is widely used in social sciences to measure the discriminative power of a model. More information about the GINI index and its applications can be found in [11].

A couple of modeling methods were compared, of which the results are listed in Table 4.

Apparently, for this data set, naive Bayes provides much better priors than noisy-MAX. This is partly due to the fact that a suboptimal learning algorithm for noisy-MAX is used. Even with EM learning, noisy-MAX scores will not be higher than 76%. Secondly, the weighting seems to perform better than the smoothing for this data set.

5 Conclusions

Two improvements to learning for rare event data were suggested in this paper. Firstly, weighted merging instead of cutting, which allows a more gentle balance between a prior and a posterior. Secondly, inference in a naive Bayes models can provide excellent priors for a CPT in a 'normal' Bayesian network.

Compared to existing methods, such as noisy-MAX priors, naive Bayes priors perform better on the test data set. Unfortunately, the best performing model on this data set was naive Bayes without any posterior learning. Therefore, more data sets to test these methods on are required for a definite statement. These initial results are very promising.

References

[1] Beck, N., G. King & L. Zeng. 'The Problem with Quantitative Studies of International Conflict.' http://web.polmeth.ufl.edu/papers/98/beck98.zip.

[2] Friedman, N., D. Geiger & M. Goldszmidt. 'Bayesian Network Classifiers.' *Machine Learning* 29(2-3):131-163, 1997.

[3] Heckerman, D. 'A Tutorial on Learning Bayesian Networks.' Technical Report, Microsoft Research, 1995.

[4] King, G. & L. Zeng. 'Logistic Regression in Rare Events Data.' *Political Analysis* 9(2):137-163, 2001.

[5] Lauritzen, S.L. & D.J. Spiegelhalter. 'Local computations with probabilities on graphical structures and their application to expert systems (with discussion).' *Journal of Royal Statistical Society, Series B* 50(2):157-224, 1988.

[6] Neal, R.M. *Bayesian Learning for Neural Networks.* Springer-Verlag, 1996.

[7] Onisko, A., M.J. Druzdzel & H. Wasyluk. 'Learning Bayesian network parameters from small data sets: application of Noisy-OR gates.' *International Journal of Approximate Reasoning* 27(2):165-182, 2001.

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

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

[10] Spirtes, P., C. Glymour & R. Scheines *Causation, Prediction and Search.* Lecture Notes in Statistics 81, Springer Verlag, 1993.

[11] Xu, K. 'How has the literature on Gini's Index evolved in the past 80 years?' Working paper, 2003.