작은 데이터에 대한 베이지안망 분류기(BNC)의 베이지안 모델 평균화(BMA) 성능 평가

황규백 장병탁

서 울 대 학 교 컴 퓨 터 공 학 부 바 이 오 지 능 연 구 실

kbhwang@bi.snu.ac.kr btzhang@cse.snu.ac.kr

Evaluation of Bayesian Model Averaging (BMA) of Bayesian Network Classifiers (BNCs) on Small Datasets

Kyu-Baek Hwang and Byoung-Tak Zhang

Biointelligence Lab, School of Computer Science and Engineering, Seoul National University

요 약

작은 데이터에서 베이지안망 분류기(Bayesian network classifier, BNC)를 학습할 때, 과대적합(overfitting)으로 인한 일반화 성능의 저하가 초래된다. 이런 경우, 베이지안 모델 평균화(Bayesian model averaging, BMA)는 모델 자체에 대한 불확실성을 분석 과정에서 고려함으로써, 성능 저하를 피할 수 있는 수단을 제공한다. 본 논문에서는 BNC의 BMA의 작은 데이터에 대한 성능을 평가 및 분석한다. 특히, 노드의 순서에 대한 평균화의 효과가 연구된다. 인공데이터에 대한 실험 결과, 노드의 순서가 BNC의 BMA의 분류 성능에 미치는 영향은 지대하며, 이는 데이터의 크기가 극히 작은 경우의 성능 저하에 직접적인 원인이 된다.

1. 서론

베이지안망 분류기(Bayesian network classifier, BNC)[5, 6]는 데이터 속성들 사이의 관계를 시각화해서 표현할 수 있는 특징을 가지고 있다. 이는 의학진단[3, 14]이나 유전자발현양상분류[7, 12]와 같이 분류 성능과 함께 결론이 도출되는 과정에 대한 이해 또한 중요한 경우에 유용하다.

한편, 데이터의 크기가 작은 경우, BNC의 일반화 성능은 낮아지게 된다. 그 이유는 다수의 서로 다른 베이지안망이 작은 학습 데이터를 잘 서술할 수 있으며, 이들 중 좋은 모델을 선택하기 위해 각각의 일반화 성능을 측정하는 것이 불가능하기 때문이다. 이의 해결을 위한 하나의 수단으로 베이지안 모델 평균화(Bayesian model averaging, BMA)를 들 수 있다[11]. BMA는 가능한 모든 가설(모델)들을 결합함으로써 모델 자체의 불확실성에 대한 원칙적인 해결책을 제시한다. 이는 또한 과대적합(overfitting) 문제의 해결과 일반화 성능의 향상에 도움을 준다. 하나의 문제점은 가능한 베이지안망의 개수가 노드의 수의 지수승 이상으로, 일반적인 경우, BNC의 BMA는 불가능하다는 점이다. 하지만 최근, 베이지안망의 BMA에 대한 근사화 기법들이 연구되고 있다[8, 4].

본 논문에서는 작은 데이터에 대한 BNC의 BMA의 성능을 평가 및 분석한다. 특히, 이전에는 시도된 바가 없는 노드 순서에 대한 BMA를 BNC에 적용한다. 인공데이터에 대한 실험을 통해, 다양한 데이터의 크기, BMA의 근사화의 정도(평균화에 이용되는 순서의 개수)가 분류 성능에 미치는 영향이 분석된다.

2. 베이지안망 분류기의 베이지안 모델 평균화

2.1 문제 정의

본 논문에서는 n개의 이산 변수 $\{X_1, X_2, \ldots, X_n\}$(= $\mathbf{X}$)로 구성된 문제를 다룬다. 변수 집합 $\mathbf{X}$는 하나의 클래스 변수 X_1과 $n-1$개의 속성 변수(feature variable)로 구성된다. 베이지안망 분류기(BNC)는 완전한(complete) 학습데이터 $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$에서 학습된다. 베이지안망은 망구조 G와 파라미터집합 $\boldsymbol{\Theta}$로 구성된다. G의 최대 in-degree는 상수 k로 제한된다. 베이지안망의 각 지역확률분포는 다항분포이며 그 파라미터는 디리슈레분포(Dirichlet distribution)를 따른다. 이제, 클래스 레이블이 없는 새 예제 $\{x_2, x_3, \ldots, x_n\}$(= $\mathbf{x}_{M+1}\backslash x_1$)의 클래스는 D에 기반해서 다음과 같이 예측할 수 있다.

$$\text{The class label for } \mathbf{x}_{M+1}\backslash x_1 = \arg\max_{X_1}\{P(X_1 \mid \mathbf{x}_{M+1}\backslash x_1, D)\} \tag{1}$$

이때, 식(1)의 조건부확률은 확률의 정의에 의해 다음과 같이 계산된다.

$$P(X_1 \mid \mathbf{x}_{M+1}\backslash x_1, D) = \frac{P(X_1, \mathbf{x}_{M+1}\backslash x_1 \mid D)}{\sum_{X_1} P(X_1, \mathbf{x}_{M+1}\backslash x_1 \mid D)} \tag{2}$$

식(2)의 우변의 분모 및 분자를 계산하기 위한 BNC의 BMA는 다음과 같이 이루어진다.

$$\begin{aligned} P(\mathbf{X}=\mathbf{x} \mid D) &= \sum_G P(\mathbf{X}=\mathbf{x}, G \mid D) \\ &= \sum_G P(G \mid D)\cdot P(\mathbf{X}=\mathbf{x} \mid G, D) \\ &= \sum_G P(G \mid D)\cdot \int P(\mathbf{X}=\mathbf{x}, \boldsymbol{\Theta} \mid G, D)d\boldsymbol{\Theta} \\ &= \sum_G P(G \mid D)\cdot \int P(\boldsymbol{\Theta} \mid G, D)P(\mathbf{X}=\mathbf{x} \mid \boldsymbol{\Theta}, G)d\boldsymbol{\Theta} \end{aligned} \tag{3}$$

식(3)의 적분은 parameter modularity 및 parameter independence와 같은 적절한 가정하에 closed-form으로 계산된다[9, 10]. 문제는 n이 대략 8을 넘는 경우, G에 대한 합이 불가능하다는 점이다. G의 최대 in-degree를 적당한 k로 제한하더라도 가능한 망구조의 개수는 $2^{\Theta(kn\log n)}$이다[8]. [8]은 이를 해결하기 위해 특정한 노드 순서(이하 순서) $\prec$가 정해진 경우의 모델의 평균화와 모든 순서에 대한 평균화의 두 단계로 문제를 분할했다. 이는 다음과 같이 표현된다.

$$P(\mathbf{X}=\mathbf{x} \mid D) = \sum_{\prec} P(\mathbf{X}=\mathbf{x}, \prec \mid D) = \sum_{\prec} P(\prec \mid D)\cdot P(\mathbf{X}=\mathbf{x} \mid \prec, D) \tag{4}$$

2.2 순서가 고정된 경우의 모델 평균화

순서가 고정된 경우의 평균화는 다음과 같이 구해진다.

$$\begin{aligned} P(\mathbf{X}=\mathbf{x} \mid \prec, D) &= \frac{P(\mathbf{X}=\mathbf{x}, D \mid \prec)}{P(D \mid \prec)} = \frac{\sum_G P(\mathbf{X}=\mathbf{x}, G, D \mid \prec)}{\sum_G P(D, G \mid \prec)} \\ &= \frac{\sum_G P(G \mid \prec)\cdot P(D \mid G)\cdot P(\mathbf{X}=\mathbf{x} \mid G, D)}{\sum_G P(G \mid \prec)\cdot P(D \mid G)} \end{aligned} \tag{5}$$

식(5)의 계산을 위해 순서가 정해진 경우 망구조의 사전확률 $P(G|\prec)$가 정의되어야 한다. 본 논문에서는 계산상의 편의를 위해 G가 $\prec$와 불일치하는 경우는 0, 일치한다면 $\alpha P(G)$로 가정한다. 그러면, 식(5)는 다음과 같이 간략화된다.

$$P(\mathbf{X}=\mathbf{x}\,|<,D)=\frac{\sum_{G\in g_{k,<}}P(G)\cdot P(D\,|\,G)\cdot P(\mathbf{X}=\mathbf{x}\,|\,G,D)}{\sum_{G\in g_{k,<}}P(G)\cdot P(D\,|\,G)} \quad (6)$$

여기서, $g_{k,<}$는 최대 in-degree가 k이며 순서 $<$를 따르는 망구조들만의 집합이다. 하지만, $g_{k,<}$의 크기 역시 $2^{\Theta(kn\log n)}$이기 때문에 식(6)의 계산은 여전히 거의 불가능하다. 그런데, 식(6)에서 marginal likelihood $P(D|G)$와 새 예제에 대한 조건부확률 $P(\mathbf{X}=\mathbf{x}|G,D)$는 디리슈레를 포함한 앞의 가정들에 의해 다음과 같이 노드별로 분리된다[9, 10].

$$P(D\,|\,G)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\cdot\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (7)$$

$$\text{and}\quad P(\mathbf{X}=\mathbf{x}\,|\,G,D)=\prod_{i=1}^{n}\frac{\alpha_{ij_{\mathbf{x}}k_{\mathbf{x}}}+N_{ij_{\mathbf{x}}k_{\mathbf{x}}}}{\alpha_{ij_{\mathbf{x}}}+N_{ij_{\mathbf{x}}}}$$

이제, 노드별로 분리가능한 망구조의 prior probability, $P(G)=\Pi_i\rho(X_i|\mathbf{Pa}_G(X_i))$(단, $\mathbf{Pa}_G(X_i)$는 G에서 X_i의 부모)를 사용하면, 각 노드에 대한 prior probability, marginal likelihood 및 조건부확률의 해당 부분을 아래와 같이 함께 표현할 수 있다.

$$\begin{aligned}&score(X_i,\mathbf{Pa}_G(X_i)\,|\,D)\\&=\rho(X_i\,|\,\mathbf{Pa}_G(X_i))\cdot\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\cdot\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}\end{aligned} \quad (8)$$

이제, 식(6)은 식(7)과 식(8)에 의해 다음과 같이 표현된다.

$$P(\mathbf{X}=\mathbf{x}\,|<,D)=\frac{\sum_{G\in g_{k,<}}\prod_i score(X_i,\mathbf{Pa}_G(X_i)\,|\,D)\cdot\frac{\alpha_{ij_{\mathbf{x}}k_{\mathbf{x}}}+N_{ij_{\mathbf{x}}k_{\mathbf{x}}}}{\alpha_{ij_{\mathbf{x}}}+N_{ij_{\mathbf{x}}}}}{\sum_{G\in g_{k,<}}\prod_i score(X_i,\mathbf{Pa}_G(X_i)\,|\,D)} \quad (9)$$

[2, 8, 4]는 식(9)가 아래와 같은 변형을 통해 효율적으로 계산될 수 있음을 보였다.

$$P(\mathbf{X}=\mathbf{x}\,|<,D)=\frac{\prod_i\sum_{\mathbf{U}\in u_{i,<}}score(X_i,\mathbf{U}\,|\,D)\cdot\frac{\alpha_{ij_{\mathbf{x}}k_{\mathbf{x}}}+N_{ij_{\mathbf{x}}k_{\mathbf{x}}}}{\alpha_{ij_{\mathbf{x}}}+N_{ij_{\mathbf{x}}}}}{\prod_i\sum_{\mathbf{U}\in u_{i,<}}score(X_i,\mathbf{U}\,|\,D)} \quad (10)$$

식(10)에서 $u_{i,<}$는 주어진 순서 $<$에서 가능한 X_i의 부모들을 나타낸다. 식(9)를 식(10)으로 변형하는 것은 순서가 고정된 경우 각 노드의 부모를 선택하는 일이 서로 독립이기 때문에 가능하다.

2.3 순서에 대한 평균화

이 절에서는 순서에 대한 평균화(식(4))에 대해 설명한다. 노드의 개수가 n일 때, 가능한 순서는 $n!$개이므로, 이에 대해 평균화하는 것은 불가능하다. [8]에서는 이를 근사하기 위해 MCMC(Markov chain Monte Carlo) 기법을 이용하였다. 이 기법은 순서의 posterior probability $P(<|D)$에 기반한 순서를 생성하는 사슬을 이용하여 T개의 순서 $\{<_1, <_2, \ldots, <_T\}$를 생성해 낸다. 이제 식(4)는 아래와 같이 근사화된다.

$$\sum_{<}P(<|\,D)\cdot P(\mathbf{X}=\mathbf{x}\,|<,D)\approx\frac{1}{T}\sum_{t=1}^{T}P(\mathbf{X}=\mathbf{x}\,|<_t,D) \quad (11)$$

본 논문에서는 이러한 마코프사슬을 만드는데 메트로폴리스 알고리즘[13]을 이용했다. 이 알고리즘은 아래의 식에 따라서 현 순서 $<$에서 다음 순서 $<'$(자신도 포함)으로의 이동을 결정한다.

$$\min\left[1,\frac{P(<'\,|\,D)q(<|<')}{P(<|\,D)q(<'\,|<)}\right] \quad (12)$$

$<$에서 $<'$으로의 이동을 결정하는 proposal probability $q(<'|<)$은 uniform을 이용하였으며, 순서의 이동은 임의의 두 노드를 선정해 위치를 바꾸는 방법을 이용했다.

3. 실험 및 평가

실험에서는 BNC의 BMA를 ALARM망[1]에서 생성된 데이터를 통해 평가했다. ALARM망은 37개의 이산 변수(노드)로 구성되어 있으며, 각 변수는 2 ~ 4개의 값을 가진다. 실험에서는 이진 변수인 *Catecholamine* 노드를 클래스 변수로 상정하였다. 학습데이터의 크기로는 25, 50, 100, 500, 1,000이 이용되었으며, 데이터 생성과정에 기인하는 편차를 고려하여 각 경우에서 10번씩 데이터를 생성 및 실험하였다. 분류성능의 평가를 위해서 3,000개의 예제를 가지는 평가데이터를 따로 생성하였다. 분류성능의 측정은 ROC(receiver operating characteristics) curve의 넓이를 이용하였다. 마지막으로, 망구조의 prior로는 복잡한 구조에 낮은 확률을 부여하는 아래의 일반적인 prior를 이용하였다.

$$P(G)\propto 2^{-\sum_i\left(\log n+\log\binom{n}{\mathbf{Pa}_G(X_i)}\right)} \quad (13)$$

식(7)의 디리슈레 prior α_{ijk}는 uninformative value인 1.0이 이용되었다.

3.1 순서표본의 추출

학습데이터의 크기에 따른 순서표본의 추출과정은 그림 1과 같다. log 점수로 평가하는 경우, 약 1만번 정도의 반복을 거치면 안정된 순서표본을 학습데이터의 크기에 상관없이 얻을 수 있음을 알 수 있다.

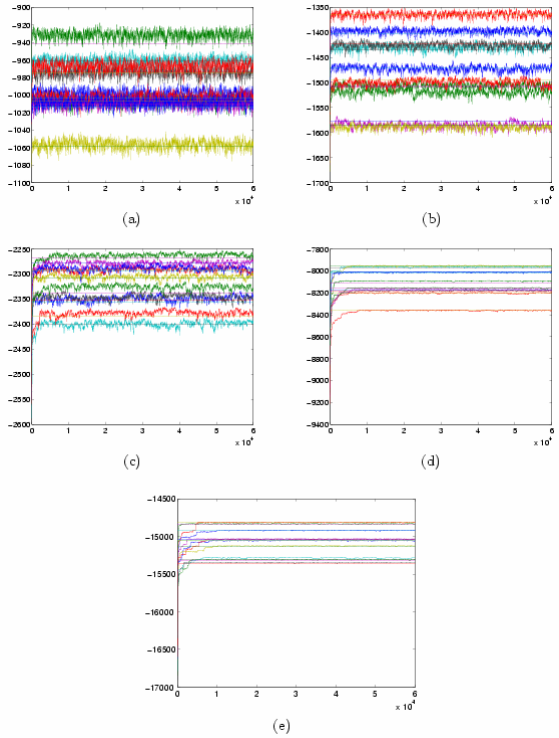

그림 1. 학습데이터의 크기에 따른 순서표본의 추출과정((a) 25, (b) 50, (c) 100, (d) 500, (e) 1000). 각 그래프의 *x*축은 표본추출과정의 진행을 나타낸다(0 ~ 6만번). *y*축은 추출된 순서의 log 점수($\log_2 P(<,D)$)를 나타낸다. 각 학습데이터의 크기에 대해서 10번씩 독립적인 데이터 생성을 통한 실험을 행하였다. 모든 경우에 약 10,000번의 시행 후에는 점수 상으로는 안정적인 모드에 접어듬을 알 수 있다.

3.2 분류 성능 평가

순서에 대한 평균화를 위해(식(11)), 1만번 이후의 순서들을 무작위로 선택해서 이용했다. T의 값으로는 1, 5, 10, 30을 이용했으며, 더 이상은 계산상의 비용문제로 사용할 수 없었다. 각 경우에 따른 분류성능은 그림 2와 같다. 우선 기본적으로 모든 경우에 분류 성능이 상당히 좋음을 알 수 있다(ROC 면적 평균 0.85 이상). 이는 학습데이터의 크기가 상당히 작았던 점을 고려하면 BMA의 성능이 매우 뛰어남을 알 수 있는 결과이다. 그림 2를 보면 학습데이터의 크기가 지극히 작은 경우(25, 50)는 원래의 노드순서(Original) 하나를 이용하는 경우의 성능이 제일 좋음을 알 수 있다. 하지만, 학습데이터의 크기가 적당히 작은 경우(100, 500, 1,000)는 5개 이상의 순서에 대해 평균화를 하는 것이 원래 순서를 이용하는 것과 대등한 성능을 냄을 알 수 있다. 특히, 학습데이터의 크기가 500 이상인 경우는 원래의 순서 하나를 이용하는 것보다 여러 개의 추출된 순서에 대한 평균화를 이용하는 것이 성능이 더 좋음을 알 수 있다.

우리는 이러한 현상의 분석을 위해, 추출된 순서들과 원래 순서 사이의 rank correlation coefficient를 계산하였다. Rank correlation coefficient에서 -1.0은 완전한 반대를 의미하며, 1.0은 완전한 일치를 의미한다. 이 결과는 그림 3에 제시되어 있다. 학습데이터의 크기가 극히 작은 경우(25, 50)에는 최대 값이 0.8이며 -0.5 ~ 0.6 사이의 순서가 대부분임을 알 수 있다. 그러나, 학습데이터의 크기가 적당히 작은 경우(100, 500, 1,000)에는 1만번 이하의 burn-in phase 부분을 제외하면 거의 모든 순서

가 0 이상의 correlation을 가짐을 알 수 있다. 이는, 분류 성능에 대한 학습데이터의 크기에 따른 경향과 일치하며, 좋은 분류 성능은 적절한 노드의 순서와 밀접한 연관이 있다는 실험적 증거로 해석될 수 있을 것이다.

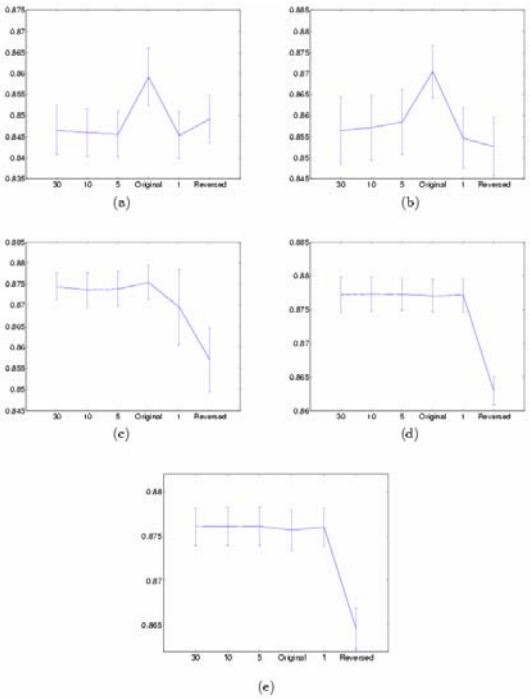

그림 2. 다른 크기의 학습데이터((a) 25, (b) 50, (c) 100, (d) 500, (e) 1000)에서 학습된 BNC의 BMA의 분류 성능(평균 및 표준편차). 성능은 최대값을 1로 normalize한 ROC curve의 넓이로 평가했다. *x*축은 평균화된 순서의 개수(식(11)의 *T*)를 나타낸다. "Original"은 ALARM망의 원래 순서를 이용한 것이며, "Reversed"는 이를 반대로 한 것을 의미한다.

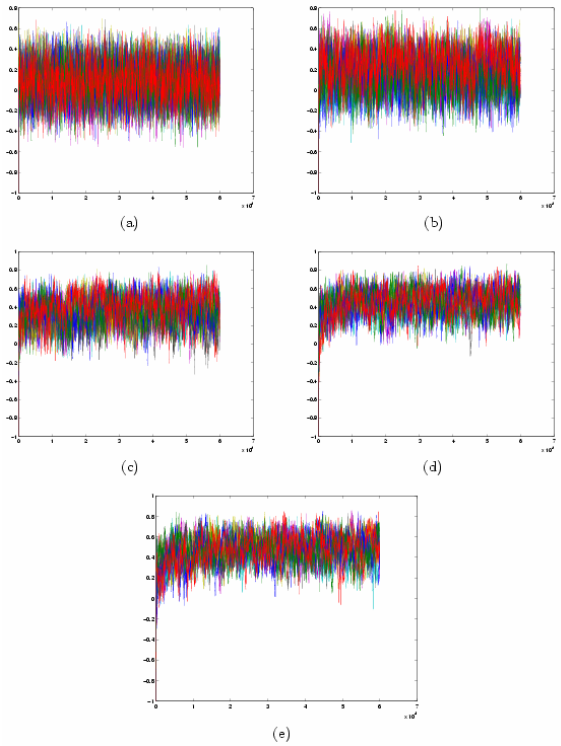

그림 3. 다른 크기의 학습데이터((a) 25, (b) 50, (c) 100, (d) 500, (e) 1000)에 따른 마코프체인을 이용해 추출된 순서들과 원래 순서의 rank correlation coefficient. 1.0은 완벽한 일치이며 -1.0은 완전한 반대를 의미한다. 학습데이터의 크기가 클수록 원래의 순서와 잘 일치하는 순서들이 추출됨을 알 수 있다.

4. 결론

본 논문에서는 베이지안망 분류기(BNC)의 베이지안 모델 평균화(BMA)의 작은 데이터에 대한 성능을 평가 및 분석하였다. 우선, BMA를 이용하는 경우 작은 데이터로도 뛰어난 분류 성능을 얻을 수 있었다. 또한, 노드 순서가 학습 성능에 미치는 영향이 지대함을 알 수 있었으며, 이는 학습데이터가 극히 작은 경우의 성능 저하의 원인임을 알 수 있었다. 따라서, 앞으로의 연구 방향은 작은 데이터에서 정확한 노드 순서를 효율적으로 얻어낼 수 있는 방법에 초점을 맞추어야 할 것이다. 적절한 순서를 얻을 수 있다면, 이러한 노드에 대한 효율적인 BMA를 통해 데이터의 크기가 작은 경우의 분류 문제를 잘 해결할 수 있다고 생각된다.

감사의 글

이 논문은 교육인적자원부의 BK21 사업과 과학기술부의 IMT-2000, NRL 및 BrainTech 사업에 의하여 지원되었음.

참고 문헌

[1] Beinlic, I., Suermondt, G., Chavez, R., Cooper, G.F.: The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. Proceedings of the Second European Conference on Artificial Intelligence in Medicine (1989) 247-256

[2] Buntine, W.: Theory refinement on Bayesian networks. Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI) (1991) 52-60

[3] Chickering, D.M., Pearl, J.: A clinician's tool for analyzing non-compliance. AAAI/IAAI **2** (1996) 1269-1276

[4] Dash, D., Cooper, G.F.: Model averaging with discrete Bayesian network classifiers. Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (2003)

[5] Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning **29**(2/3) (1997) 103-130

[6] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29**(2/3) (1997) 131-163

[7] Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. Journal of Computational Biology **7**(3/4) (2000) 601-620

[8] Friedman, N., Koller, D.: Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. Machine Learning **50**(1) (2003) 95-125

[9] Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning **20**(3) (1995) 197-243

[10] Heckerman, D: A tutorial on learning Bayesian networks. In: Jordan, M.I. (ed.): Learning in Graphical Models. MIT Press, Cambridge: MA (1999) 301-354

[11] Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: a tutorial. Statistical Science **14**(4) (1999) 382-417

[12] Hwang, K.-B., Cho, D.-Y., Park, S.-W., Kim, S.-D., Zhang, B.-T.: Applying machine learning techniques to analysis of gene expression data: cancer diagnosis. In: Lin, S.M., Johnson, K.F. (eds.): Methods of Microarray Data Analysis. Kluwer Academic Publishers, Norwell: MA (2002) 167-182

[13] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculation by fast computing machines. Journal of Chemical Physics **21**(6) (1953) 1087-1092

[14] Nikovski, D.: Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. IEEE Transactions on Knowledge and Data Engineering **12**(4) (2000) 509-516