# Feature Selection by Markov Chain Monte Carlo Sampling – A Bayesian Approach

Michael Egmont-Petersen

Institute of Information and Computing Sciences, Utrecht University,
Padualaan 14, De Uithof, The Netherlands
Michael@cs.uu.nl

**Abstract.** We redefine the problem of feature selection as one of model selection and propose to use a Markov Chain Monte Carlo method to sample models. The applicability of our method is related to Bayesian network classifiers. Simulation experiments indicate that our novel proposal distribution results in an ignorant proposal prior. Finally, it is shown how the sampling can be controlled by a regularization prior.

## 1 Introduction

The problem of feature selection has been targeted regularly in publications appearing in the literature on datamining and statistical pattern recognition. Important key references include [1–4]. Whether one wants to learn a statistical classifier from data, a graphical model or perform clustering in high-dimensional space, confining the number of feature variables included in the model by either feature selection or feature transformation, is often necessary. Inclusion of too many feature variables leads to over-fitting. Within the context of feature selection, over-fitting causes the so-called *peaking phenomenon* to occur [5]. Peaking refers to the fact that the performance of a statistical model (e.g., the error rate of a classifier) on an independent test dataset generally peaks when utilizing only a *subset* of the available feature variables. This is counterintuitive, as adding extra non-informative feature variables to a statistical model should not, intuitively, lead to a performance decrease. However, as the model parameters are estimated from a training dataset of a finite size, variance associated with the parameter estimates leads to fitting random variations of the non-informative features and hence to a decrease in performance.

The problem of feature selection is more complex than often stated in the literature, because the use of different feature subsets inevitably imposes different models (e.g., a different topology of a neural network [6]). Hence, feature selection implies model selection. Model selection is a complex problem that, even in the simple case where models are compared with *one* assessment criterion (e.g., the likelihood of the model, its classification error rate or its residual variance), entails a trade-off between bias and variance. On the one hand, allowing the inclusion of a large number of features/parameters, may lead to an accurate model. However, the parameters will because of their large variance result in a model performance that is prone to noise. On the other hand, limiting the number of parameters is more likely to bias model performance, but

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 1034–1042, 2004.

it results in less variance. In this paper, we suggest to sample different models from a statistical distribution in order to make such trade-offs explicit. So instead of looking for "that one particular model with the best performance", we propose to sample the posterior distribution of models using a Markov Chain Monte Carlo method [7, 8].

## 2 Background

Feature selection has been approached within statistical pattern recognition at an early stage. In 1971, it was conjectured that the peak in performance (1 – error rate) of statistical classifiers solely occurred when the features were dependent [9]. Later, Trunk [10] managed to prove that even for $n_{all}$ independent normally distributed feature variables, peaking can occur when $n_{all} \rightarrow \infty$. It is furthermore clear that increasing the size $m$ of the training dataset solely shifts the position of the peak, allowing the model to utilize an increasing number of feature variables. Also within multivariate statistics, approaches for variable selection for linear regression [11], linear and quadratic discriminant analysis [12] have been developed.

Algorithms for feature selection rely on a *search scheme* and an *assessment criterion* $J_s(X, n)$ for comparing feature subsets. Within the pattern recognition literature, much research focused on search schemes [4, 13] and assessment criteria $J_s$ [6]. Solely exhaustive search is guaranteed to result in an optimal feature subset with the maximal score $J_s(X, n)$ on a test set, when the assumption of monotony of $J_s(X, n)$ with respect to an increase in $n$ does not hold [5].

Conclusively, three interrelated obstacles impede efficient feature selection: the peaking phenomenon, the combinatorial complexity of exhaustive search resulting in $2^{n_{all}} - 1$ nonempty feature subsets and the fact that feature selection entails model selection.

## 3 A Formalism for Model and Feature Selection

We formalize the joint problem of feature and model selection. In the sequel, individual stochastic variables are denoted with capital letters $A, B, \ldots$, sets of stochastic variables with bold capital letters, $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$. Outcomes of the variables are denoted with small letters, e.g., $P(A = a)$. Correspondingly, $P(\boldsymbol{X} = \boldsymbol{x})$ denotes the probability that the variables $\boldsymbol{X}$ have the outcomes $\boldsymbol{x}$. In general, we use $\boldsymbol{X}$ to denote the set of feature variables and $C$ the outcome variable. The statistical model $M$ consists of a structural part represented by the graph $G$ and a set of parameters $\boldsymbol{\theta}$, $M = (G, \boldsymbol{\theta})$. The model is used to implicitly estimate the class-conditional probability distribution $P(\boldsymbol{X} \mid C, M)$, when the variables $\boldsymbol{X}$ are discrete and the density $p(\boldsymbol{X} \mid C, M)$ when the variables $\boldsymbol{X}$ are continuous. For simplicity, we henceforward solely address the situation where $C$ is *discrete*. With $\boldsymbol{D}$, we denote a training database with $m$ training instances, $\boldsymbol{D} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m)$. Each instance $\boldsymbol{d}$ is represented by the discrete feature vector $\boldsymbol{x}$ and class label $c$, yielding $\boldsymbol{d} = (\boldsymbol{x}, c)^T$.

The search for the best feature subset can be illustrated with the lattice graph introduced in [14]. Let $G_U = (\boldsymbol{V}, \boldsymbol{E})$ be an *undirected* graph with $\boldsymbol{V}$ indicating the vertices and $\boldsymbol{E}$ the edges connecting pairs of vertices: $E_{ijk} = 1$ if the vertices $V_i$ and $V_j$ are

connected (indicating feature subsets with a Hamming distance of 1), with $k$ the model index, whereas $E_{ijk} = 0$ when $V_i$ and $V_j$ are not connected in model $k$. Figlure 1 shows an example of a nested lattice structure that represents both feature subsets and models. It is clear that model selection can be performed separately, for a given feature subset, but that feature selection necessitates model selection.

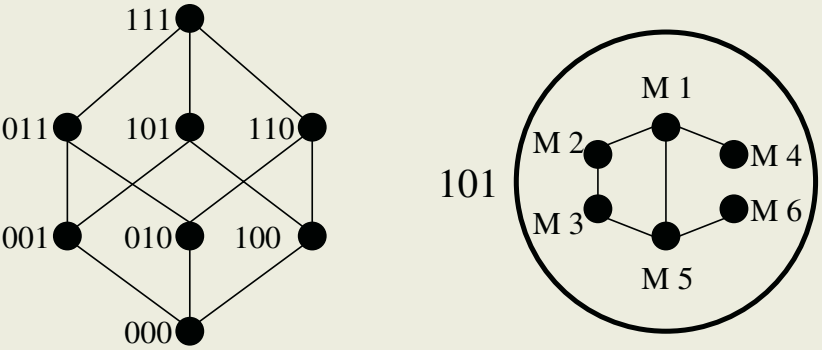


**Fig. 1.** (a) Lattice structure connecting the feature subsets with a feature set Hamming distance of 1. (b) A close-up of the *model subspace* associated with feature subset '101' reveals that several alternative models ($M1 - M6$) may be applied to the feature subset 101. The edges indicate models with a model Hamming distance of 1. The model subspace associated with the empty set 000 below (not shown) contains solely one model, namely the distribution of the predicted variable $C$ in the training set.

The nested lattice structure depicted in figure 1, with an *upper layer* representing the feature subsets and a *lower layer* representing the possible models that utilize the associated feature subset, is unnecessary complex to work with. Instead, we redefine the feature selection problem as one of model selection. Consequently, we propose considering the score $J_s$ as a random variable and set as goal to sample the joint distribution

$$p(J_s, \boldsymbol{Y}, \hat{M}|\boldsymbol{D}), \;\; \boldsymbol{Y} \in \mathcal{P}(\boldsymbol{X}), \; M \; \in \; \mathcal{M} \tag{1}$$

where the score $J_s$ is a continuous stochastic variable. The generic score function $J_s$ may indicate, for example, the likelihood $s = L$, the Bhattacharyya distance $s = \mu$, the error rate $s = \epsilon$, or another measure depending on how the model $M$ should be scored for the particular application at hand. The hat-notation indicates that the parameters $\boldsymbol{\theta}$ of the model $M$ have been estimated from the training set, but they may also be integrated out, see e.g. [15]. $\mathcal{P}(\boldsymbol{X})$ denotes the power set of $\boldsymbol{X}$ and $\mathcal{M}$ the set of valid models that can be learned from it.

## 4 Markov Chain Monte Carlo Sampling

We will use the Metropolis-Hastings algorithm [16] to perform Markov Chain Monte Carlo sampling from a *target probability function* $\Pi$. More specifically, we propose to

sample the distribution $P(J, \boldsymbol{Y}, \hat{M}|\boldsymbol{D})$. The Metropolis-Hastings algorithm results in a discrete Markov Chain over a state space, $\mathcal{S}$. The transition probabilities, $P_q(S_k \rightarrow S_l)$, $S_k, S_l \in \mathcal{S}$ specify the probability of making a jump from state $S_k$ (associated with model $k$) to state $S_l$.

### 4.1 Probabilistic Network Classifiers

Markov Chain Monte Carlo techniques can be used to sample the posterior distribution of different types of classifiers. Here, we illustrate MCMC by probabilistic network classifiers [17, 18]. A probabilistic network classifier matches the general description of a statistical model given in Section 3, see Fig. 2.

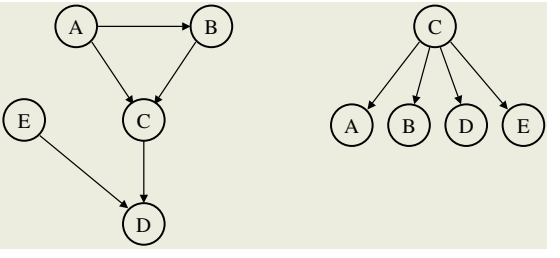


**Fig. 2.** (a) Directed acyclic graph specifying the direct dependencies in a Bayesian network classifier with 4 feature variables. The chain rule (Eq. (2)) specifies how the joint probability factorizes: $P(A, B, C, D, E) = P(A)P(B|A)P(C|A, B)P(D|C, E)P(E)$. The variables $A$ and $B$ are independent from $D$ and $E$, given the class label $C$. (b) Naive Bayesian classifier where the feature variables are independent, given the class label $C$. The joint probability factorizes into: $P(A, B, C, D, E) = P(A|C)P(B|C)P(C)(D|C)P(E|C)$.

A probabilistic network classifier $M = (G, \boldsymbol{\theta})$ consists of a structural model specification, the directed graph $G$, and the parameters, $\boldsymbol{\theta}$, with the (un)conditional probability $\theta_{i,j,\pi(i)} = P(D_i = d_j \mid \pi(D_i) = \boldsymbol{d}_{\pi(D_i)})$. The notation $\pi(D_i) = \boldsymbol{d}_{\pi(D_i)}$ indicates the values of the parents of node $D_i$ in the graph $G$ (the parents constitute the nodes with arcs pointing directly to node $D_i$). Computation of the posterior probability distribution $P(C = c|\boldsymbol{X} = \boldsymbol{x})$ is specified by the directed graph. It follows from the chain rule that the joint probability $P(\boldsymbol{d}) = P(c, \boldsymbol{x})$ is computed from

$$P(\boldsymbol{d}) = \prod_{i=1}^{n+1} P(D_i = d \mid \pi(D_i) = \boldsymbol{d}_{\pi(D_i)}) \tag{2}$$

(see Fig. 2). A little manipulation of Bayes formula yields the posterior probability associated with class label $c_j$ (where we omit denoting the variables)

$$P(c_j|\boldsymbol{x}) = \frac{P(c_j, \boldsymbol{x})}{\sum_k P(c_k, \boldsymbol{x})} \tag{3}$$

### 4.2 Approach by Madigan & York

Madigan & York have earlier presented an approach for sampling graphical models, based on the Markov Chain Monte Carlo scheme [8]. In their approach, the goal is to sample graphical models $G \in \mathcal{G}$ from the posterior distribution, $P(G \mid \boldsymbol{D})$. The likelihood $L(G \mid \boldsymbol{D}) \propto P(\boldsymbol{D} \mid G)$ with $P(\boldsymbol{D} \mid G)$ the probability that a particular structural model $G$ results in the dataset $\boldsymbol{D}$.

Madigan & York define a homogeneous, stationary and reversible Markov chain. This chain specifies the transition probabilities, $P_q(G \rightarrow G\prime)$, that a jump is made from the model $G$ to the model $G\prime$. The possible jumps from $G$ to $G\prime$ consist of all acyclic graphs that can be constructed by adding one arc to $G$ or by deleting one arc from $G$. Hence, $G\prime \in NB(G)$, the neighborhood of $G$ defined by

$$NB(G) = \left\{ \begin{array}{ll} \bigcup_{i \in I(D)}, \bigcup_{j \in (I(D) \setminus i \mid (X_i \rightarrow X_j) \notin G)} & Add_{AC}(G, (X_i \rightarrow X_j)) \ \bigcup \\ \bigcup_{i \in I(G)}, \ \bigcup_{j \in (I(G) \setminus i \mid (X_i \rightarrow X_j) \in G)} & Del(G, (X_i \rightarrow X_j)) \end{array} \right\} \tag{4}$$

with $I(D)$ denoting the indices of the nodes representing all $n+1$ variables, and $Add_{AC}$ a function that adds an arc to $G$ *iff* the resulting graph $G\prime$ is *acyclic*. Together, the requirements of homogeneity and reversibility and the fact that the Markov Chain is stationary, make it feasible to use the transition kernel (proposal distribution) $q(G \rightarrow G\prime)$ in the Metropolis-Hastings algorithm [8]. The *proposal probability* $P_q(G \rightarrow G\prime) = |NB(G)|^{-1}$, whereas the probability of the reverse proposal is $P_q(G\prime \rightarrow G) = |NB(G\prime)|^{-1}$. The *transition probability* $P(G\prime|G)$ is modelled as $P(G\prime|G) = P_q(G \rightarrow G\prime)\alpha(G, G\prime)$, $G \neq G\prime$. The detailed balance, which ensures reversibility, is obtained by using the normalization factor $\alpha(G, G\prime)$

$$\alpha(G, G\prime) = \min \left[ 1 \, , \, \frac{P(\boldsymbol{D}|G\prime)\, P(G\prime)}{P(\boldsymbol{D}|G)\, P(G)} \frac{P_q(G\prime \rightarrow G)}{P_q(G \rightarrow G\prime)} \right] \tag{5}$$

Sampling from the proposal distribution $q(G \rightarrow G\prime)$ and normalising by $\alpha(G, G\prime)$ result in a posterior distribution $P(G|\boldsymbol{D})$ where more likely models appear more often than unlikely ones.

### 4.3 Naive Bayes classifiers

A special type of probabilistic network classifiers are the naive Bayesian classifiers, see Fig. 2 (b). MCMC can be modified to sample naive Bayesian classifiers. Because no model selection takes place in this simple case, each node in the lattice graph (Fig. 1 (a)) contains solely one model. Instead of using the likelihood $P(\boldsymbol{D}|G)$ as assessment criterion, we suggest to use the general criterion $J_s$, which may be the likelihood, 1–error rate, or another metric that measures discriminative performance. We define the *add-one-delete-one neighborhood* to include only Naive Bayes classifiers

$$NB_I(G) = \left\{ \begin{array}{ll} \bigcup_{i \in I(X) \mid (C \rightarrow X_i) \notin G} & Add(G, (C \rightarrow X_i)) \ \bigcup \\ \bigcup_{i \in I(G) \setminus C} & Del(G, (C \rightarrow X_i)) \end{array} \right\} \tag{6}$$

Following the approach by Madigan & York, it would be natural to set the proposal probability to $P_q(G \rightarrow G\prime) = |NB_I(G)|^{-1}$. However, such a choice leads nonuniform proposal distribution $P(N)$ with respect to the size of the feature subsets that are compared. The following theorem formalizes this

**Theorem 1** *A Markov Chain Monte Carlo scheme for proposing feature subsets, where each subset in the add-one-delete-one neighborhood $NB_I$ has the same probability of being proposed, $P_q(G \rightarrow G\prime) = |NB_I(G)|^{-1}$, this results in a proposal probability imposing a nonuniform prior $P(N)$ that is maximal for $N = n, n \in \{n_{all}/2 - 1, n_{all}/2, n_{all}/2 + 1\}$, depending on whether $n_{all}$ is even or odd.*

*Proof (sketch)*
It follows from the Binomial theorem that the number of size $n \in \{1, \ldots, n_{all}\}$ feature subsets of $n_{all}$, is $\binom{n_{all}}{n}$. As $\binom{n_{all}}{n+1} \geq \binom{n_{all}}{n}$, $n < n_{all}/2$ because $|I(\boldsymbol{X}),\ (C \rightarrow X_i) \notin G| \geq |I(G) \setminus C|$ whereas $\binom{n_{all}}{n+1} \leq \binom{n_{all}}{n}$, $n > n_{all}/2$ because $|I(\boldsymbol{X}),\ (C \rightarrow X_i) \notin G| \leq |I(G) \setminus C|$, it follows that the prior $P(N)$ is maximal for feature subsets with a size $n_{all}/2$.

We will instead use a proposal distribution, that results in each size $n$ having the same (uniform) probability. Establish the partitioning of $NB_I(G)$ into two *disjoint* subsets, $NB_I(G) = \{NB_{IG(+1)}, NB_{IG(-1)}\}$, where $NB_{IG(+1)}(G)$ is the subset of graphical models in $NB_I(G)$ that results from *adding* one arc to $G$, and $NB_{IG(-1)}(G)$ is the subset of graphical models in $NB_I(G)$ that results from *deleting* one arc from $G$. We now suggest a two-step proposal distribution $q$: Draw a uniformly distributed number $u \sim \mathcal{U}(0,1)$. If $u \geq \frac{1}{2}$ then choose a model in $NB_{IG(+1)}$ with the probability $|NB_{IG(+1)}|^{-1}$, otherwise choose a model in $NB_{IG(-1)}$ with the probability $|NB_{IG(-1)}|^{-1}$. The normalization factor that ensures detailed balance, becomes

$$\alpha(G, G\prime) = \min\left[1\,,\ \frac{P(J|G\prime)\,P(G\prime)}{P(J|G)\,P(G)}\frac{P_q(G\prime \rightarrow G)}{P_q(G \rightarrow G\prime)}\right] \tag{7}$$

with the proposal probability $P_q(G \rightarrow G\prime) = 1$ and

$$P_q(G\prime \rightarrow G) = \begin{cases} \frac{1}{2}: & n(G) = 0 \\ \frac{1}{2}: & n(G) = n_{\max} \\ 1: & otherwise \end{cases} \tag{8}$$

where $n(G)$ indicates the number of features included in model $G$. $P_q$ restores detailed balance with respect to the number of features in relation to the two end points, 0 and $n_{\max}$, of the proposal interval. The parameter $n_{\max} \leq n_{all}$ such that the maximal size of a feature subset can be limited. The resulting posterior distribution implies a noninformative (uniform) prior, $P(N)$, on size $N = n$ of any feature subset. We conduct a simulation experiment to illustrate the practical implication of Theorem 1.

### 4.4 General Bayesian Network Classifiers

We now extend our MCMC approach to sample general Bayesian network classifiers. Hence, the lattice graph (Fig. 1 (a)) represents both different feature subsets and models. Define the *one-step look ahead* neighborhood of the graph $G$ consisting of the directed acyclic graphs resulting in valid probabilistic classifiers that can be constructed by adding one arc to $G$ or deleting one arc from $G$

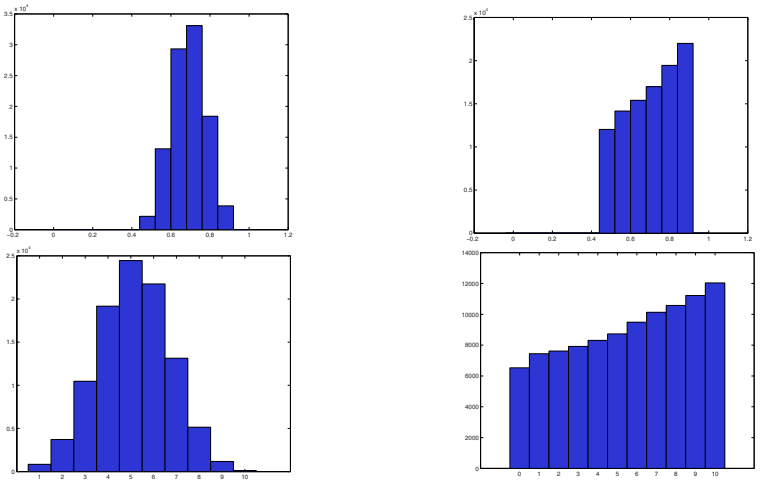

**Fig. 3.** (Upper left) Shows the posterior distribution of $J$ for MCMC scheme 1. (Lower left) Shows the distribution of feature subsets for MCMC scheme 1. (Upper right) Shows the posterior distribution of $J$ for MCMC scheme 2. (Lower right) Shows the distribution of feature subsets for MCMC scheme 2.

$$NB_C(G) = \left\{ \begin{array}{ll} \bigcup_{i \in I(\mathbf{X}),\, C \notin \sigma(X_i),\, C \notin \pi(X_i)} & Add(G, (X_i \rightarrow C)) \ \bigcup \\ \bigcup_{i \in I(\mathbf{X}),\, C \notin \sigma(X_i),\, C \notin \pi(X_i)} & Add(G, (C \rightarrow X_i)) \ \bigcup \\ \bigcup_{j \in I(\sigma(C))} \bigcup_{i \in I(\mathbf{X}) \setminus \pi(X_j)} & Add(G, (X_i \rightarrow X_j)) \ \bigcup \\ \bigcup_{i \in I(G)} \bigcup_{j \in I(G) \mid (X_i \rightarrow X_j) \in G} & Del(G, (X_i \rightarrow X_j)) \end{array} \right\} \quad (9)$$

with $\pi(X_i)$ the children of node $X_i$ and $\sigma(X_i)$ the parents of node $X_i$. The function $I(\sigma(C))$ denotes the indices of the children of the classification node $C$. The neighborhood $NB_C(G)$ is subdivided into four disjoint subsets

$$NB_C(G) = \{NB_C(G + 1_F), NB_C(G - 1_F), NB_C(G + 1_M), NB_C(G - 1_M)\} \quad (10)$$

The subset $NB_C(G + 1_F)$ contains the graphical models in $NB_C(G)$ where the addition of an arc implies that $G\prime$ contains one feature variable more than $G$. The subset $NB_C(G - 1_F)$ contains the models in $NB_C(G)$ where the deletion of an arc implies that $G\prime$ contains one feature variable less than $G$. The subset $NB_C(G + 1_M)$ contains the models in $NB_C(G)$ where the addition of an arc increases the complexity of $G\prime$, but where $G$ and $G\prime$ include the same feature variables. $NB_C(G - 1_M)$ contains the models in $NB_C(G)$ where the deletion of an arc decreases the complexity of $G\prime$, but where $G$ and $G\prime$ include the same feature variables.

We define the proposal distribution $q_C$ as follows:

$$q_C(G \rightarrow G\prime) = \left\{ \begin{array}{ll} u < \frac{1}{4} & q_1(|NB_C(G + 1_F)|^{-1}) \\ \frac{1}{4} \leq u < \frac{1}{2} & q_2(|NB_C(G - 1_F)|^{-1}) \\ \frac{1}{2} \leq u < \frac{3}{4} & q_3(|NB_C(G + 1_M)|^{-1}) \\ \frac{3}{4} \leq u & q_4(|NB_C(G - 1_M)|^{-1}) \end{array} \right. \quad (11)$$

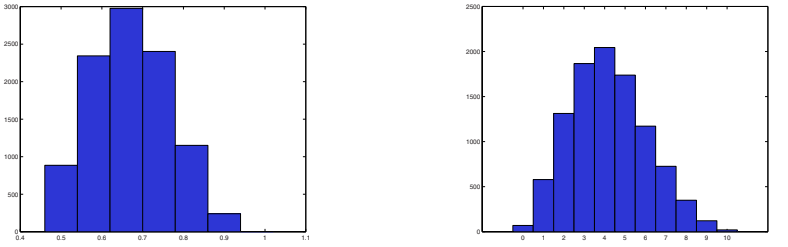

**Fig. 4.** (a) Shows the posterior distribution of $J$ when using a Poisson prior, $P(G)$. The left tail contains models with the maximal performance 0.9. (b) Shows the distribution of feature subsets when the same prior is used.

with $u \sim \mathcal{U}(0,1)$. So in each proposal, the MCMC-algorithm with the same probability chooses to add a feature, delete a feature, increase the model complexity or simplify the model (the two latter moves keep the same feature subset).

## 5 Simulation Experiments

We performed a simulation experiment in order to compare the two sampling schemes proposed for the Naive Bayesian classifier in Section 4.3. We set the scoring metric $J(X) = 1 - \epsilon(X)$ and chose to simulate with a feature set of 10 features. Five of the (independent) features could each lead to an increase in $J(X)$ of 0.08, yielding a maximum of 0.9. The performance resulting from the empty feature set is 0.5. We sampled 100.000 feature subsets (naive Bayes classifiers) using the scheme based on the Madigan & York approach (scheme 1), and 100.000 using our novel proposal distribution (scheme 2).

Our second proposal distribution based on the neighborhood $NB_I$ behaves as could be expected and support Theorem 1. The more features a subset contains, the higher the resulting score $J$ will be. To cope with the curse of dimensionality, we experimented with using the discrete Posson distribution as prior, $P(N)$. Setting $\lambda = 4$, we obtained the results as shown in Fig. 4.

In our third experiment, we implemented the proposal distribution for general Bayesian network classifiers, Eq. (11). We sampled 100 training cases from the probability distribution specified by the graph in Fig. 2. MCMC was set to run for 1000 iterations. The simulation resulted in 757 nonempty feature sets. The most frequent nonempty feature set included all 5 features, and was found 166 times. The second most likely feature set consisting only of feature 1, was sampled 63 times. So the correct feature set was also most frequently sampled in the markov chain.

## 6 Discussion

We have presented a method for sampling statistical models in general, and pattern classifiers in particular, using an ignorant proposal distribution. It was shown how a regularization prior can be used to restrain the maximal dimensionality of the sampled models. Finally, we showed how general Bayesian classifiers can be sampled using the novel proposal distribution.

## References


1. Foroutan, I., Sklansky, J.: Feature selection for automatic classification of non-gaussian data. IEEE Transactions on Systems, Man, and Cybernetics **17** (1987) 187–198
2. Kittler, J.: Computational problems of feature selection pertaining to large data sets. In: Proceedings of Pattern Recognition in Practice. (1980) 405–414
3. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE transactions on pattern analysis and machine intelligence **19** (1997) 153–158
4. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern recognition **33** (2000) 25–41
5. Waller, W.G., Jain, A.K.: On the monotonicity of the performance of a bayesian classifier. IEEE Transactions on Information Theory **24** (1978) 120–126
6. Egmont-Petersen, M., Talmon, J., Hasman, A., Ambergen, A.: Assessing the importance of features for multi-layer perceptrons. Neural networks **11** (1998) 623–635
7. Giudici, P., Castelo, R.: Improving markov chain monte carlo model search for data mining. Machine learning **50** (2003) 127–158
8. Madigan, D., York, J.: Bayesian graphical models for discrete-data. International statistical review **63** (1995) 215–232
9. Chandrasekaran, B.: Independence of measurements and the mean recognition accuracy. IEEE Transactions of Information Theory **17** (2002) 452–456
10. Trunk, G.: A problem of dimensionality: a simple example. IEEE Transactions of Pattern Analysis and Machine Intelligence **1** (1979) 306–307
11. Forsythe, A., Engleman, L., Jennrich, R., May, P.R.A.: A stopping rule for variable selection in multiple regression. Journal of the American Statistical Association **68** (1973) 75–77
12. McLachlan, G.J.: Discriminant analysis and Statistical Pattern Recognition. John Wiley & Sons, New York (1992)
13. Siedlecki, W., Sklansky, J.: On automatic feature selection. Journal of Pattern Recognition and Artificial Intelligence **2** (1988) 197–220
14. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters **10** (1989) 335–347
15. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine learning **9** (1992) 309–347
16. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. American statistician **49** (1995) 327–335
17. Baesens, B., Egmont-Petersen, M., Castelo, R., Vanthienen, J.: Learning bayesian network classifiers for credit scoring using markov chain monte carlo search. In: Proceedings of the International Conference on Pattern Recognition, Piscataway, IEEE Computer Society (2002) 49–52
18. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine learning **29** (1997) 131–163