

Bayesian Network Classifiers versus k -NN Classifier using Sequential Feature Selection

Franz Pernkopf *

University of Washington, Department of Electrical Engineering
M254 EE/CSE Building, Box 352500, Seattle, WA, 98195-2500, USA
Graz University of Technology, Institute of Communications and Wave Propagation,
Inffeldgasse 12, A-8010 Graz, Austria, pernkopf@tugraz.at

Abstract

The aim of this paper is to compare Bayesian network classifiers to the k -NN classifier based on a subset of features. This subset is established by means of sequential feature selection methods. Experimental results show that Bayesian network classifiers more often achieve a better classification rate on different data sets than selective k -NN classifiers. The k -NN classifier performs well in the case where the number of samples for learning the parameters of the Bayesian network is small. Bayesian network classifiers outperform selective k -NN methods in terms of memory requirements and computational demands. This paper demonstrates the strength of Bayesian networks for classification.

Introduction

In most classification tasks the relevant features are often unknown a priori. Thus, many features are derived from a specific classification problem and those which do not contribute or even degrade the classification performance should be removed from the set of extracted features. Feature selection has become important for numerous pattern recognition and data analysis tasks (Jain & Zongker 1997), (Dash & Liu 1997), (Kohavi & John 1997). The main purpose of feature selection is to reduce the number of extracted features to a set of a few significant ones while maintaining the classification rate. The reduction of the feature set may even improve the classification rate by reducing estimation errors associated with finite sample size effects (Jain & Chandrasekaran 1982).

Another approach to achieve an improvement of the classification accuracy is to model statistical dependencies between attributes. Therefore, the framework of Bayesian networks (Pearl 1988) can be used to build classifiers (Friedman, Geiger, & Goldszmidt 1997), (Singh & Provan 1996), (Keogh & Pazzani 1999) to address this effectively. Bayesian network classifiers have been applied in many different domains. In this paper, we compare the performance

of these approaches to the selective k -NN classifier. The selective k -NN classifier uses a subset of features which maximize the classification performance. This subset is established by means of sequential feature selection methods.

Bayesian Network Classifier

A Bayesian network (Pearl 1988), (Cowell *et al.* 1999) $B = \langle G, \Theta \rangle$ is a directed acyclic graph G which models probabilistic relationships among a set of random variables $\mathbf{U} = \{X_1, \dots, X_n, \Omega\} = \{U_1, \dots, U_{n+1}\}$, where each variable in \mathbf{U} has specific states or values denoted by lower case letters $\{x_1, \dots, x_n, \omega\}$. The symbol n denotes the number of attributes of the classifier. Each vertex (node) of the graph represents a random variable, while the edges capture the direct dependencies between the variables. The network encodes the conditional independence relationship that each node is independent of its nondescendants given its parents. These conditional independence relationships reduce the number of parameters needed to represent a probability distribution. The symbol Θ represents the set of parameters which quantify the network. Each node contains a local probability distribution given its parents. The joint probability distribution of the network is uniquely determined by these local probability distributions. Basically, two different techniques for parameter learning are available, the maximum likelihood estimation and the Bayesian approach (Cowell *et al.* 1999). In this paper, the parameters of the network are estimated by the maximum likelihood method. In the following, three different types of Bayesian network classifiers are presented, the naïve Bayes classifier, the tree augmented naïve Bayes classifier, and the selective unrestricted Bayesian network classifier.

Naïve Bayes Classifier

The naïve Bayes (NB) decision rule (Duda, Hart, & Stork 2000) assumes that all the attributes are conditionally independent given the class label. As reported in the literature (Friedman, Geiger, & Goldszmidt 1997), the performance of the naïve Bayes classifier is surprisingly good even if the independence assumption between attributes is unrealistic in most of the data sets. Independence between the features ignores any correlation among them. The attribute values of X_i and X_j ($X_i \neq X_j$) are conditionally independent given the class label of node Ω . Hence, x_i

*The author is grateful to Elisabeth Pernkopf and Prof. Djamel Bouchaffra for the valuable comments on this paper and to Ingo Reindl and Voest Alpine Donawitz Stahl for providing the data for the first experiment.
Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

is conditionally independent of x_j given class ω , whenever $P(x_i|\omega, x_j) = P(x_i|\omega)$ for all $x_i \in X_i, x_j \in X_j, \omega \in \Omega$, and when $P(x_j, \omega) > 0$. The structure of the naïve Bayes classifier represented as Bayesian network is illustrated in Figure 1. Feature selection is introduced to this network by removing irrelevant features by means of a search algorithm (see Section *Search-and-Score Structure Learning*). This extension is known as selective naïve Bayes classifier (SNB). The structure in Figure 1 shows that each attribute is

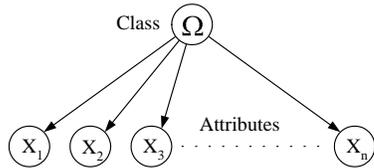


Figure 1: Structure of a naïve Bayes network.

conditionally independent of the remaining attributes given the class label ω of the class variable. The class variable Ω is the only parent for each attribute X_i denoted as $\text{pa}_{X_i} = \{\Omega\}$ for all $1 \leq i \leq n$. Hence, the joint probability distribution $P(X_1, \dots, X_n, \Omega)$ for this network is determined to be $P(X_1, \dots, X_n, \Omega) = \prod_{i=1}^{n+1} P(U_i|\text{pa}_{U_i}) = P(\Omega) \prod_{i=1}^n P(X_i|\Omega)$, and from the definition of conditional probability the probability for the classes in Ω given the values of the attributes is $P(\Omega|X_1, \dots, X_n) = \alpha P(\Omega) \prod_{i=1}^n P(X_i|\Omega)$, where α is a normalization constant.

Tree Augmented Naïve Bayes Classifier

Since the features may be correlated and the independence assumption of the naïve Bayes classifier is unrealistic, Friedman et al. (Friedman, Geiger, & Goldszmidt 1997) introduce the *tree augmented naïve Bayes classifier* (TAN). It is based on the structure of the naïve Bayes network where the class variable is the parent of each attribute. Hence, the posterior probability $P(\Omega|X_1, \dots, X_n)$ takes all the attributes into account. Additionally, edges (arcs) among the attributes are allowed in order to capture the correlations among them. Each attribute may have at most one other attribute as additional parent which means that there is an arc in the graph from feature X_i to feature X_j . This implies that these two attributes X_i and X_j are not independent given the class label. The influence of X_j on the class probabilities depends also on the value of X_i . An example of a tree augmented naïve Bayes network is shown in Figure 2. A tree augmented naïve Bayes network is initialized as naïve Bayes network. Additional arcs between attributes are found by means of a search algorithm (see Section *Search-and-Score Structure Learning*). The maximum number of arcs added to relax the independence assumption between the attributes is $n - 1$.

Selective Unrestricted Bayesian Network Classifier

The selective unrestricted Bayesian network classifier (SUN) (Singh & Provan 1996) (see Figure 3) can be viewed as generalization of the tree augmented naïve Bayes

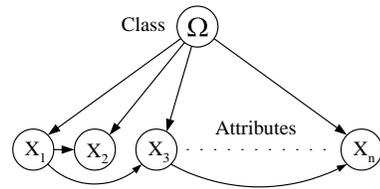


Figure 2: Structure of a tree augmented naïve Bayes network.

network. The class node is equally treated as an attribute node and may have attribute nodes as parents. The attributes need not be connected directly to the class node as for the tree augmented naïve Bayes network. After initialization the network consists of the nodes without any arcs. A search algorithm (see Section *Search-and-Score Structure Learning*) adds arcs to the network according to an evaluation criterion. If there is no arc between an attribute and the classifier network then the attribute is not considered during classification. During the determination of the network structure, irrelevant features are not included and the classifier is based on a subset of selected features. This unrestricted network structure maximizes the classification performance by removing irrelevant features and relaxing independence assumptions between correlated features. Since this network is unrestricted, the computational

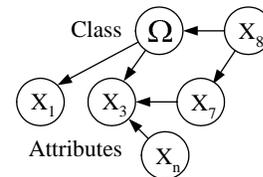


Figure 3: Structure of a selective unrestricted Bayesian network.

demands for determining the network structure is huge especially if there is a large number of attributes available. Additionally, the size of the conditional probability tables of the nodes increases exponentially with the number of parents. This might result in a more unreliable probability estimate of the nodes which have a large number of parents. The posterior probability distribution of Ω given the value of all attributes is only sensitive to those attributes which form the Markov blanket of node Ω (Pearl 1988). The Markov blanket of the class node Ω consists of the direct parents of Ω , the direct successors (children) of Ω , and all the direct parents of the direct successors (children) of the class node Ω . All the features outside the Markov blanket do not have any effect on the classification performance. Introducing this knowledge into the search algorithm reduces the search space and the computational effort for determining the structure of the classifier.

Search-and-Score Structure Learning

In order to learn the structure of the Bayesian network classifiers, Keogh and Pazzani propose hill climbing search (Keogh & Pazzani 1999). An improvement of the hill climbing search is to apply the classical floating search algorithm (CFS) used for feature selection applications (Pudil, Novovičová, & Kittler 1994). This algorithm is adopted for learning the network structure of tree augmented naïve Bayes classifiers and selective unrestricted Bayesian network classifiers (Pernkopf & O’Leary 2003). We use the cross-validation classification accuracy estimate as scoring function J for evaluating the performance of the networks. For the selective unrestricted Bayesian network this learning approach enables simultaneous feature selection and structure learning. The main disadvantage of hill climbing search is that once an arc has been added to the network structure, the algorithm has no mechanism for removing the arc at a later stage. Hence, this algorithm suffers from the *nesting* effect (Kittler 1978). To overcome this drawback, Pudil et al. (Pudil, Novovičová, & Kittler 1994) present a floating search method for finding significant features which optimize the classification performance in feature selection tasks. This algorithm allows conditional exclusions of previously added attributes and/or arcs from the network. Hence, this algorithm is able to correct disadvantageous decisions which were performed in previous steps. Therefore, it may approximate the optimal solution in a better way than hill climbing search, especially in case of data of great complexity and dimensionality. However, this search strategy uses more evaluations to obtain the network structure and therefore is computationally less efficient than hill climbing search. This floating search algorithm facilitates the correction of disadvantageous decisions made in previous steps.

Sequential Feature Selection Algorithms

Sequential feature selection algorithms search in a sequential deterministic manner for the best feature subset (sub-optimal). Basically, forward and backward algorithms are available. The forward methods start with an empty set and add features until a stopping criterion concludes the search. The backward algorithms begin with all features and remove features iteratively. The well-known suboptimal sequential algorithms are listed in the following.

- Sequential forward selection (SFS): The sequential forward selection algorithm (Kittler 1978) is a bottom up search method. With each iteration one feature among the remaining features is added to the subset, so that the subset maximizes the evaluation criterion J .
- Sequential backward selection (SBS): Sequential backward selection (Kittler 1978) is the counterpart of the sequential forward selection. In each step one feature is rejected so that the remaining subset gives the best result of the evaluation criterion J .
- Plus l -take away r selection (PTA(l, r)): The PTA(l, r) algorithm (Kittler 1978) partially avoids nesting of feature sets by allowing a low level backtracking in the selection process. The subset is enlarged by adding l features using the SFS algorithm. Afterwards, r features are rejected by

using the SBS method. Both steps are repeated until a particular predetermined subset size is obtained.

- Generalized sequential forward selection (GSFS(r)) (Kittler 1978): At each stage r features are added simultaneously instead of adding just one feature to the subset at a time like the SFS method does. This algorithm and the following counterpart does not avoid nesting entirely, since successive added sets may be nested.
- Generalized sequential backward selection (GSBS(r)): This algorithm is the counterpart of the GSFS method.
- Generalized plus l -take away r selection (GPTA(l, r)) (Kittler 1978), (Devijver & Kittler 1982): The difference between the PTA and the GPTA method is that the former approach employs the SFS and the SBS procedures instead of the GSFS and GSBS algorithms.
- Sequential forward floating selection (SFFS): The sequential forward floating algorithm is a bottom up search procedure introduced by Pudil et al. (Pudil, Novovičová, & Kittler 1994). This method was adapted for learning the structure of Bayesian network classifiers (see Section *Search-and-Score Structure Learning*) (Pernkopf & O’Leary 2003). The algorithm includes new features which maximize the criterion J by means of the SFS procedure starting from the current feature set. Afterwards, conditional exclusions of the previously updated subset take place. If no feature can be excluded anymore, the algorithm proceeds again with the SFS algorithm. The floating methods are allowed to correct wrong decisions made in previous steps and they approximate the optimal solution in a better way than the sequential feature selection methods described above.
- Sequential backward floating selection (SBFS). This algorithm is the counterpart to the SFFS method.
- Adaptive sequential forward floating selection (ASFFS(r_{max}, b, d)): This search algorithm suggested by Somol et al. (Somol et al. 1999) utilizes the best of both, generalized strategies and floating methods. This algorithm is similar to the SFFS procedure, where the SFS and the SBS methods are replaced by their generalized versions GSFS(r) and GSBS(r). The optimal value of r is determined dynamically. The maximum generalization level which is used is restricted by a user defined bound r_{max} . The current generalization level r changes during the search depending on the subset size k . Therefore, the parameters d and b are necessary. For a detailed description of the algorithm refer to Somol et al. (Somol et al. 1999). This algorithm is initialized with an empty subset.
- Adaptive sequential backward floating selection (ASBFS(r_{max}, b, d)) (Somol et al. 1999). This is the counterpart of the ASFFS method.

Experiments

A comparative study has been performed on data of a surface inspection task (Pernkopf 2003) and data sets from the UCI repository (Merz, Murphy, & Aha 1997). Throughout the experiments, five-fold cross-validation classification accuracy has been used for learning the structure of the Bayesian

networks. Similarly, the five-fold cross-validation classification rate of the k -NN approach is used as scoring function J for the feature selection methods. All the structure learning and feature selection experiments are based on exactly the same cross-validation folds of the data sets. The Bayesian network classifiers use discretized features, which were discretized using recursive minimal entropy partitioning (Fayyad & Irani 1993). The partition boundaries for discretization were established only through the training data set. Zero probabilities of the conditional probability tables of the Bayesian network classifiers are replaced with a small epsilon $\varepsilon = 0.00001$. The following abbreviations are used for the different classification approaches:

- **NB**: Naïve Bayes classifier.
- **CFS-SNB**: Selective Naïve Bayes classifier using the classical floating search.
- **HCS-TAN**: Tree augmented naïve Bayes classifier using hill-climbing search.
- **CFS-TAN**: Tree augmented naïve Bayes classifier using classical floating search.
- **CFS-SUN**: Selective unrestricted Bayesian network using classical floating search.
- **SFFS- k -NN-C**: k -nearest neighbor classifier using continuous-valued data and the SFFS method ($k \in \{1, 3, 5, 9\}$).
- **SFFS- k -NN-D**: k -nearest neighbor classifier using discrete-valued data and the SFFS method ($k \in \{1, 3, 5, 9\}$).

The k -NN classifier requires a scaling of the features, whereby, the scaling parameters were determined only through the training data set of the corresponding cross-validation folds of the data set. Each feature is scaled to zero mean and unit variance (Kaufman & Rousseeuw 1990).

First Experiment

Data Experiments have been performed on a data set \mathcal{S} consisting of 516 surface segments from a surface inspection task, uniformly distributed into three classes. Each sample (surface segment) is represented by 42 features which are described in more detail in (Pernkopf 2003). The data set is divided into six mutually exclusive subsets $\mathcal{S} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5, \mathcal{H}\}$. The data set parts $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$, and \mathcal{D}_5 are used for finding the optimal classifier (five-fold cross-validation). Each part is comprised of 90 samples. The established classifiers are validated on a separate hold-out data set \mathcal{H} which was never employed during the optimization experiments (Kohavi & John 1997).

Results First, the sequential forward feature selection algorithms are compared in terms of the achieved cross-validation classification performance using the 3 – NN classifier (see Figure 4). Generally, the floating algorithms perform better than the methods without floating property. The sequential forward floating algorithm (SFFS) performs in a similar manner as the more complex adaptive floating method (ASFFS(3,4,5)) (With the given parameter setting of the ASFFS algorithm the classification performance of a subset size of 5 within a neighborhood of 4 is optimized

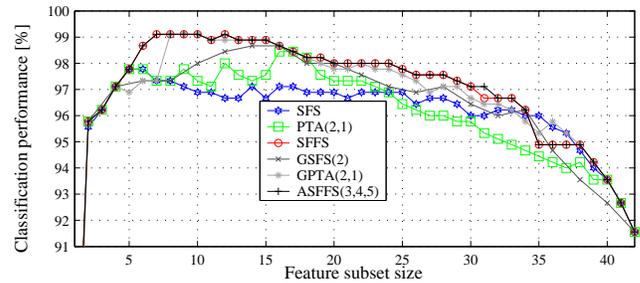


Figure 4: Cross-validation classification performance of different sequential forward feature selection methods for a subset size up to 42.

more thoroughly.). There is only a marginal difference of the result for the feature subset size of 31. However, the number of classifier evaluations used for establishing the optimal subset for classification is for the SFFS only 5086 compared to the ASFFS with 7768. For the floating algorithms, the computational costs depend on the characteristics of the data set due to the floating property. The generalized algorithms (GSFS and GPFA) perform slightly worse than the SFFS method with a computational requirement of 6391 and 13201, respectively. The number of evaluations used for obtaining the optimal subset for all non-floating algorithms is fixed for a given parameter setting. The PTA and SFS search strategies achieve the lowest scores for different sizes of subsets. Therefore, 2623 and 903 evaluations are necessary. Since the SFFS algorithm achieves a good tradeoff between computational demands and achieved classification score and due to the fact that the structure of the Bayesian network classifiers was trained with an equivalent algorithm further feature selection results consider only this method.

Table 1 compares the feature selection results of the SFFS approach to the Bayesian network classification methods. The table shows the five-fold cross validation classification accuracy estimate (%CV5) and the performance on the hold-out data set (%H). It also depicts the number of classifier evaluations (#Evaluations) used for search, the number of independent probabilities (#Parameters), and the number of features (#Features) and/or arcs (#Arcs) used to achieve this classification accuracy estimate. The best achieved classification accuracy is emphasized by boldface letters.

The selective naïve Bayes classifier (CFS-SNB) achieves a better %CV5 classification accuracy estimate than the naïve Bayes (NB) approach based on all available attributes. However, the performance on the hold-out data is similar. The computational demands for establishing the CFS-SNB classifier is relatively small. For the tree augmented naïve Bayes classifier the same result is achieved either with hill climbing or with the classical floating search algorithm. This means that the CFS method does not perform backward steps during the search. This is also observable in the number of used classifier evaluations. The TAN classifier uses all extracted features and 12 arcs are added to the naïve Bayes structure (#Arcs=52). The TAN classifier uses 533 independent probabilities which have to be estimated from the data set and

	%CV5	%H	#Arcs	#Features	#Evaluations	#Parameters
NB	89.11 ± 2.8727	95.45	40	40	1	275
CFS-SNB	96.44 ± 2.11	96.96	20	20	2098	122
HCS-TAN	97.11 ± 2.02	96.96	52	40	17893	533
CFS-TAN	97.11 ± 2.02	96.96	52	40	17958	533
CFS-SUN	98.66 ± 0.81	98.48	14	12	4097	230
SFFS-1-NN-C	99.33 ± 0.60	98.48	-	8	2873	450 samples × 8 features
SFFS-3-NN-C	99.11 ± 0.49	98.48	-	8	5086	450 samples × 8 features
SFFS-5-NN-C	98.44 ± 0.60	98.48	-	7	4346	450 samples × 7 features
SFFS-9-NN-C	98.44 ± 0.99	98.48	-	9	5086	450 samples × 9 features
SFFS-1-NN-D	96.22 ± 2.02	86.36	-	15	6003	450 samples × 15 features
SFFS-3-NN-D	96.44 ± 2.53	90.90	-	23	4803	450 samples × 23 features
SFFS-5-NN-D	96.22 ± 2.02	93.94	-	26	5302	450 samples × 26 features
SFFS-9-NN-D	96.00 ± 1.26	93.94	-	20	4745	450 samples × 20 features

Table 1: Comparison of classification approaches (Experiment 1).

it is only slightly better than the selective naïve Bayes classifier. However, the CFS-SNB classifier has a much simpler structure and a smaller number of parameters is required. The selective unrestricted Bayesian network (CFS-SUN) achieves the best classification accuracy estimate on the five-fold cross-validation and hold-out data set among the Bayesian network classifiers. For achieving this result, 230 probabilities have to be estimated and the structure consists of 12 attributes and 14 arcs, whereas the TAN and NB structure do not enable feature selection. Additionally, the number of classifier evaluations used for determining the structure of the TAN network is high compared to establishing the structure of the CFS-SUN since the Markov blanket is used during the search for the SUN network structure. The selective k -NN classifier on continuous attributes slightly outperforms the CFS-SUN classifier. However, the achieved classification performance on the hold-out data set is the same for both. As mentioned above, the Bayesian network classifiers use a discretized feature space. For discretized features the performance of the SFFS- k -NN-D classifier degrades. Basically, the k -NN decision rule searches through a labeled reference set for the nearest neighbors which might be time-consuming in case of a large number of samples. Additionally, a large amount of memory is required. Bayesian network classifiers outperform selective k -NN methods in terms of memory requirements and computational demands during classification. Especially, the CFS-SUN is simple to evaluate but still maintains high predictive accuracy.

Second Experiment

The second experiment has been performed on 8 data sets from the UCI repository (Merz, Murphy, & Aha 1997). The main characteristics are summarized in Table 2. The attributes in the data sets are multinomial and continuous-valued. Table 3 compares the CFS-SUN classifier to the SFFS method using the k -NN classifier. We select $k \in \{1, 3, 5, 9\}$ which gives the largest classification performance. Both classification approaches are based on the equivalent search algorithm. The table shows the five-fold cross validation classification accuracy estimate (%CV5),

Data set	#Features	#Classes	#Instances	Attributes
australian	14	2	690	mixed
flare	10	2	1066	mixed
glass	9	7	214	continuous
glass2	9	2	163	continuous
heart	13	2	270	continuous
pima	8	2	768	continuous
vote	16	2	435	discrete
vehicle	18	4	846	continuous

Table 2: Data sets (Experiment 2).

the number of classifier evaluations (#Evaluations) used for search, the number of independent probabilities (#Parameters), the number of nearest neighbors k , and the number of features (#Features) and/or arcs (#Arcs) used to achieve this classification accuracy estimate. The best achieved classification accuracy is emphasized by boldface letters. The CFS-SUN classifier outperforms the selective k -NN approach five times. In one case the selective k -NN classifier using discretized attributes achieves the best classification performance. The selective k -NN classifier performs well in domains such as Glass, Glass2, and Heart where the number of samples for learning the parameters (probabilities) of the Bayesian network is small (see Table 2). In general, it is interesting how few parameters are used by the CFS-SUN classifier, especially for the data sets Flare, Glass2, Heart, and Pima. The number of independent probabilities used for classifying the Vehicle data set is large. Basically, the size of the conditional probability tables of the nodes in the network increases exponentially with the number of parents. In this case, this might provide probability estimates that are not robust since the data set is relatively small.

Conclusion

This paper compares Bayesian network classifiers to the selective k -NN classifier. The selective k -NN classifier uses a subset of features which is established by means of sequential feature selection methods. In order to learn the structure of the Bayesian networks, hill climbing search and the sequential forward floating algorithm are used.

	Data set	Australian	Flare	Glass	Glass2	Heart	Pima	Vote	Vehicle
CFS-SUN	%CV5	89.58	84.07	74.70	82.66	86.29	75.90	98.64	76.03
	#Arcs	15	3	10	5	7	9	15	13
	#Features	10	3	7	4	5	7	10	13
	#Evaluations	1647	83	992	184	441	307	6055	1922
	#Parameters	58	15	273	8	18	29	495	1808
SFFS- k -NN-C	%CV5	-	-	78.50	88.96	86.29	75.26	-	75.53
	k	-	-	3	1	5	5	-	9
	#Features	-	-	5	5	7	7	-	9
	#Evaluations	-	-	117	87	316	108	-	1130
SFFS- k -NN-D	%CV5	88.26	83.95	68.69	79.14	87.03	68.88	97.24	68.32
	k	5	3	9	1	5	9	1	5
	#Features	8	3	4	3	7	5	6	12
	#Evaluations	465	122	87	87	283	165	377	522

Table 3: Comparison of classification approaches (Experiment 2).

Experiments were performed on data of a surface inspection task and data sets from the UCI repository. Bayesian network classifiers more often achieve a better classification rate on different data sets as selective k -NN classifiers. The k -NN classifier performs well in the case where the number of samples for learning the parameters of the Bayesian network is small. It has been observed that only few parameters are used by the selective unrestricted Bayesian network classifier for certain data sets. Bayesian network classifiers outperform selective k -NN methods in terms of memory requirements and computational demands. Especially, the performance of the selective unrestricted Bayesian network classifier demonstrates the strength of Bayesian network classifiers.

References

- Cowell, R.; Dawid, A.; Lauritzen, S.; and Spiegelhalter, D. 1999. *Probabilistic networks and expert systems*. Springer Verlag.
- Dash, M., and Liu, H. 1997. Feature selection for classification. *Intelligent Data Analysis* 1(3):131–156.
- Devijver, P., and Kittler, J. 1982. *Pattern recognition: A statistical approach*. Prentice Hall International.
- Duda, R.; Hart, P.; and Stork, D. 2000. *Pattern Classification*. John Wiley & Sons.
- Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1022–1027.
- Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine Learning* 29:131–163.
- Jain, A., and Chandrasekaran, B. 1982. *Dimensionality and sample size considerations in pattern recognition in practice*, volume 2 of *Handbook of Statistics*. Amsterdam: North-Holland.
- Jain, A., and Zongker, D. 1997. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(2):153–158.
- Kaufman, L., and Rousseeuw, P. 1990. *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons.
- Keogh, E., and Pazzani, M. 1999. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of 7th International Workshop on Artificial Intelligence and Statistics*, 225–230.
- Kittler, J. 1978. Feature set search algorithms. In Chen, C., ed., *Pattern Recognition and Signal Processing*. Sijtho and Noordho. 41–60.
- Kohavi, R., and John, G. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97:273–324.
- Merz, C.; Murphy, P.; and Aha, D. 1997. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, URL: www.ics.uci.edu/~mlearn/MLRepository.html.
- Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann.
- Pernkopf, F., and O’Leary, P. 2003. Floating search algorithm for structure learning of Bayesian network classifiers. *Pattern Recognition Letters* 24:2839–2848.
- Pernkopf, F. 2003. 3D surface analysis using Bayesian network classifiers. Technical report, Graz University of Technology.
- Pudil, P.; Novovičová, J.; and Kittler, J. 1994. Floating search methods in feature selection. *Pattern Recognition Letters* 15:1119–1125.
- Singh, M., and Provan, G. 1996. Efficient learning of selective Bayesian network classifiers. In *International Conference of Machine Learning*, 453–461.
- Somol, P.; Pudil, P.; Novovičová, J.; and Paclík, P. 1999. Adaptive floating search methods in feature selection. *Pattern Recognition Letters* 20:1157–1163.