Intelligent Data Analysis 4 (2000) 213–227
IOS Press

Supervised model-based visualization of high-dimensional data

Petri Kontkanen, Jussi Lahtinen, Petri Myllymäki, Tomi Silander and Henry Tirri
Complex Systems Computation Group (CoSCo)[1] *P.O.Box 26, Department of Computer Science, FIN-00014 University of Helsinki, Finland*

Received 8 October 1999
Revised 22 November 1999
Accepted 24 December 1999

Abstract. When high-dimensional data vectors are visualized on a two- or three-dimensional display, the goal is that two vectors close to each other in the multi-dimensional space should also be close to each other in the low-dimensional space. Traditionally, closeness is defined in terms of some standard geometric distance measure, such as the Euclidean distance, based on a more or less straightforward comparison between the contents of the data vectors. However, such distances do not generally reflect properly the properties of complex problem domains, where changing one bit in a vector may completely change the relevance of the vector. What is more, in real-world situations the similarity of two vectors is not a universal property: even if two vectors can be regarded as similar from one point of view, from another point of view they may appear quite dissimilar. In order to capture these requirements for building a pragmatic and flexible similarity measure, we propose a data visualization scheme where the similarity of two vectors is determined indirectly by using a formal model of the problem domain; in our case, a Bayesian network model. In this scheme, two vectors are considered similar if they lead to similar predictions, when given as input to a Bayesian network model. The scheme is supervised in the sense that different perspectives can be taken into account by using different predictive distributions, i.e., by changing what is to be predicted. In addition, the modeling framework can also be used for validating the rationality of the resulting visualization. This model-based visualization scheme has been implemented and tested on real-world domains with encouraging results.

Keywords: Data visualization, Multidimensional scaling, Bayesian networks, Sammon's mapping

1 Introduction

When displaying high-dimensional data on a two- or three-dimensional device, each data vector has to be provided with the corresponding 2D or 3D coordinates which determine its visual location. In traditional statistical data analysis (see, e.g., [8, 6]), this task is known as *multidimensional scaling*. From one perspective, multidimensional scaling can be seen as a data compression or data reduction task, where the goal is to replace the original high-dimensional data vectors with much shorter vectors, while losing as little information as possible. Consequently, a pragmatically sensible data reduction scheme is such that two vectors close to each other in the multidimensional space are also close to each other in the three-dimensional space. This raises the question of a distance measure — what is a meaningful definition of similarity when dealing with high-dimensional vectors in complex domains?

[1] URL: http://www.cs.Helsinki.FI/research/cosco/, E-mail: cosco@cs.Helsinki.FI

Traditionally, similarity is defined in terms of some standard geometric distance measure, such as the Euclidean distance. However, such distances do not generally properly reflect the properties of complex problem domains, where the data typically is not coded in a geometric or spatial form. In this type of domains, changing one bit in a vector may totally change the relevance of the vector, and make it in some sense quite a different vector, although geometrically the difference is only one bit. In addition, in real-world situations the similarity of two vectors is not a universal property, but depends on the specific focus of the user — even if two vectors can be regarded as similar from one point of view, they may appear quite dissimilar from another point of view. In order to capture these requirements for building a pragmatically sensible similarity measure, in this paper we propose and analyze a data visualization scheme where the similarity of two vectors is not directly defined as a function of the contents of the vectors, but rather determined indirectly by using a formal model of the problem domain; in our case, a Bayesian network model defining a joint probability distribution on the domain. From one point of view, the Bayesian network model can be regarded as a module which transforms the vectors into probability distributions, and the distance of the vectors is then defined in probability distribution space, not in the space of the original vectors.

It has been suggested that the results of model-based visualization techniques can be compromised if the underlying model is bad. This is of course true, but we claim that the same holds also for the so-called model-free approaches, as each of these approaches can be interpreted as a model-based method where the model is just not explicitly recognized. For example, the standard Euclidean distance measure is indirectly based on a domain model assuming independent and normally distributed attributes. This is of course a very strong assumption that does not usually hold in practice. We argue that by using formal models it becomes possible to recognize and relax the underlying assumptions that do not hold, which leads to better models, and moreover, to more reasonable and reliable visualizations.

A *Bayesian (belief) network* [20, 21] is a representation of a probability distribution over a set of random variables, consisting of an acyclic directed graph, where the nodes correspond to domain variables, and the arcs define a set of independence assumptions which allow the joint probability distribution for a data vector to be factorized as a product of simple conditional probabilities. Techniques for learning such models from sample data are discussed in [11]. One of the main advantages of the Bayesian network model is the fact that with certain technical assumptions it is possible to marginalize (integrate) over all parameter instantiations in order to produce the corresponding predictive distribution [7, 13]. As demonstrated in, e.g., [18], such marginalization improves the predictive accuracy of the Bayesian network model, especially in cases where the amount of sample data is small. Practical algorithms for performing predictive inference in general Bayesian networks are discussed for example in [20, 19, 14, 5].

In our supervised model-based visualization scheme, *two vectors are considered similar if they lead to similar predictions*, when given as input to the same Bayesian network model. The scheme is supervised, since different perspectives of different users are taken into account by using different predictive distributions, i.e., by changing what is to be predicted. The idea is related to the Bayesian distance metric suggested in [17] as a method for defining similarity in the case-based reasoning framework. The reason for us to use Bayesian networks in this context is the fact that this is the model family we are most familiar with, and according to our experience, it produces more accurate predictive distributions than alternative techniques. Nevertheless, there is no reason why some other model family could not also be used in the suggested visualization scheme, as long as the models are such that they produce a predictive distribution. The principles of the generic visualization scheme are described in more detail in Section 2.

Visualizations of high-dimensional data can be exploited in finding regularities in complex domains; in other words, in data mining applications. As our visualization scheme is based on Bayesian network models, one can say that we are using Bayesian networks for data mining. However, it is important to realize that our approach is fundamentally different from the more direct application of Bayesian networks for data mining, discussed in, e.g., [12]: as Bayesian network models represent regularities found in the problem domain, they can obviously be used for data mining tasks by examining the properties of the model. In our case, on the other hand, data mining is not performed by examining the properties of the model itself, but by visualizing the predictive distributions produced by the model. We would like to emphasize that we are not suggesting not to use Bayesian networks for data mining by examining the model properties; rather, we see our visualization scheme as a complementary technique that can be used in addition to normal application of Bayesian networks.

The scheme suggested by Bishop and Tipping [2] represents another interesting possibility for applying Bayesian network models in data visualization and data mining. Bishop and Tipping regard the visual coordinates as missing information, and estimate this information by using standard techniques for learning latent variable models. In a sense, this approach resembles the approach presented here, as the visualization is essentially based on the predictive distribution of the latent variable. However, the scheme is completely unsupervised — the visualizations are always made with respect to the latent variable — while in our supervised case the predictive distribution concerns one or more of the original variables. Another difference is that our scheme is flat in the sense that the result of the visualization process is one picture, while the latent variable method described in [2] produces a hierarchy of pictures at different levels. Adapting the idea of hierarchical visualization to our supervised model-based visualization scheme would make an interesting research topic.

Kohonen's *self-organizing maps* (SOMs) [15] offer yet another method for visualizing high-dimensional data, but it should be noted that the approach is fundamentally different from the scheme suggested here: a SOM is based on a neighborhood-preserving topological map tuned according to geometric properties of sample vectors, instead of exploiting probabilistic distributions produced by a formal model. Consequently, although the scheme is fully unsupervised like the scheme suggested by Bishop and Tipping, strictly speaking the method is not model-based in the same sense as the Bayesian network based approach discussed above. A more involved discussion on this topic can be found in [1].

As discussed in [2], when comparing Bayesian model-based visualization approaches to the projection pursuit and related visualization methods (see, e.g., [4] and the references therein), we can make the following observations: Firstly, the Bayesian approach is parameter-free in the sense that the user is not required to participate in the data visualization process, but the process can be fully automated. Nevertheless, the prior distributions of a Bayesian model offer the user a theoretically justifiable method for affecting the model construction process, and hence the resulting visualization, if such intervention is considered useful. Secondly, in contrast to traditional "hard" hierarchical data partitioning and visualization methods, Bayesian methods work with "soft" uncertainties (probabilities), and can hence be expected to produce more smooth and robust visualizations.

After producing a 2D or 3D visualization of a complex domain, an obvious question concerns the quality of the result: how do we know whether the visual picture represents the problem domain in a reasonable manner? This question can of course be partly answered through the results of a data mining process: if the user is capable of discovering new, interesting regularities in the data based on the visualization obtained, we can say that the visualization is in some sense reasonable, at least from a pragmatic point of view. However, we would like to come up with a more theoretically rigorous, statistical methodology for estimating the quality of different visualizations. This important question is discussed in Section 3.

The model-based visualization scheme discussed above has been implemented and tested on real-world domains with encouraging results. Illustrative examples with public-domain classification datasets are presented in Section 4. More involved projects for analyzing real-world data with domain experts are currently in progress in co-operation with the Helsinki University Central Hospital (a medical diagnosis problem related to the Helsinki Heart Study project) and several domestic industrial partners. One of the partners has also started commercializing the visualization method as part of a generic Bayesian data mining tool to be released later.

2 Supervised Model-Based Visualization

Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote a collection of N vectors each consisting of values of n attributes $X_1, \ldots, X_n$. For simplicity, in the sequel we will assume the attributes X_i to be discrete. Now let $\mathbf{X}^{'}$ denote a new data matrix where each n-component data vector $\mathbf{x}_i$ is replaced by a two- or three-component data vector $\mathbf{x}_i^{'}$. As this new data matrix can easily be plotted on a two- or three-dimensional display, we will call $\mathbf{X}^{'}$ the visualization of data $\mathbf{X}$. Consequently, for visualizing high-dimensional data, we need to find a transformation (function) which maps each data vector in the domain space to a vector in the visual space. In order to guarantee the usefulness of the transformation used, an obvious requirement is that two vectors close to each other in the domain space should also be close to each other in the visual space. Formally, we can express this requirement in the following manner:

$$\text{Minimize} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(d(\mathbf{x}_i, \mathbf{x}_j) - d'(\mathbf{x}_i^{'}, \mathbf{x}_j^{'})\right)^2, \qquad (1)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the domain space, and $d'(\mathbf{x}_i^{'}, \mathbf{x}_j^{'})$ the distance between the corresponding vectors in the visual space. Finding visual locations by minimizing this criterion is known as Sammon's mapping (see [15]).

The visual space is used for graphical, geometric representation of data vectors, so the geometric Euclidean distance seems a natural choice for the distance metric $d'(\cdot)$. Nevertheless, it is important to realize that there is no a priori reason why this distance measure would make a good similarity metric in the high-dimensional domain space. As a matter of fact, in many complex domains it is quite easy to see that geometric distance measures reflect poorly the significant similarities and differences between the data vectors. The main problems one encounters when using Euclidean distance in these domains are the following:

Difficulties in handling discrete data. Many data sets contain nominal or ordinal attributes, in which case finding a reasonable coding with respect to the Euclidean distance metric is a difficult task.

Lack of focus. Euclidean distance is inherently unsupervised in the sense that all attributes are treated equally.

Dependence of attribute scaling. As all attributes are treated as equal, it is obvious that an attribute with a scale of, say, between -1000 and 1000, is more influential than an attribute with a range between -1 and 1.

Implicit assumptions. Although at first sight it would seem that Euclidean distance is model-free in the sense that the similarities are not based on a any specific domain model, this view is flawed: it is easy to see that when summing over the pairwise distances between different attribute values independently, *we are already implicitly using a model with global* *independence between normally distributed model attributes*, although we have not stated this (and other) assumptions explicitly.

We argue that although the Euclidean distance is an obvious choice for the distance metric $d'(\cdot)$, in general $d(\cdot)$ should be different from $d'(\cdot)$.

There have been several attempts to circumvent the above weaknesses by using various coding schemes or variants of the Euclidean distance measure, such as the Mahalanobis distance (see, e.g., [6]). However, the proposed approaches either use ad hoc methodologies with no theoretical framework to support the solutions presented, or are based on relatively simple implicit assumptions that do not usually hold in practice. As an example of the latter case, it is easy to see that the Euclidean distance is based on an underlying model with normally distributed, independent variables, while the Mahalanobis distance assumes the multivariate normal model. These models are clearly too simple for modeling practically interesting, complex domains, especially without the explicit, formal theoretical framework that can be used for determining the model parameters.

We propose that in order to overcome the problems listed above, our assumptions concerning the domain space should be explicitly listed in a formal model of the problem domain, instead of using implicit models defined by distance measures. By a model M we mean here a parametric model form so that each parameterized instance (M, θ) of the model produces a probability distribution $P(X_1, \ldots, X_n | M, \theta)$ on the space of possible data vectors $\mathbf{x}$. To make our presentation more concrete, for the remainder of the paper we assume that the models M represent different *Bayesian network structures* (for an introduction to Bayesian network models, see e.g., [20, 19, 14, 5]).

In our supervised model-based visualization scheme, *two vectors are considered similar if they lead to similar predictions*, when given as input to the same Bayesian network model M. To make this idea more precise, let us assume that we wish to visualize our data with respect to *m target attributes* $X_1, \ldots, X_m$. Given a data vector $\mathbf{x}_i$ and a model M, we can now compute the predictive distribution for the target attributes:

$$P(X_1, \ldots, X_m \mid \mathbf{x}_i^{-m}, M) = P(X_1, \ldots, X_m \mid X_{m+1} = x_{m+1}^i, \ldots, X_n = x_n^i, M), \tag{2}$$

where $\mathbf{x}_i^{-m}$ denotes the values of $\mathbf{x}_i$ outside the target set, and x_k^i is the value of attribute X_k in data vector $\mathbf{x}_i$. Data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ are now considered similar if the corresponding predictive distributions are similar. It is easy to see that distance measures based on this type of similarity measures are supervised, as we can easily change the focus of the metric by changing the target set $X_1, \ldots, X_m$. The scheme is also scale invariant as we have moved from the original attribute space to the probability space where all the numbers lie between 0 and 1. This also allows us to handle different types of attributes (discrete or continuous) in the same consistent framework. Furthermore, the framework is theoretically on a more solid basis, as our domain assumptions must be formalized in the model M.

The above scheme still leaves us with the question of defining a similarity measure between two predictive distributions. The standard solution for computing the distance between two distributions is to use the Kullback-Leibler divergence (see, e.g, [10]). However, this asymmetric measure is not a distance metric in the geometric sense, and what is more, it has an infinite range which leads easily to computational problems with practical implementations. For these reasons, we propose the following distance metric to be used in practice:

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1.0 - P(\text{MAP}(\mathbf{x}_i) = \text{MAP}(\mathbf{x}_j)), \tag{3}$$

where $\text{MAP}(\mathbf{x}_i)$ denotes the *maximum posterior probability* assignment for the target attributes $X_1, \ldots, X_m$ with respect to the predictive distribution (2). In Section 4, we present practical examples that were obtained by using this distance measure.

3 Validating the Visualization Results

The practical relevance of a visualization $\mathbf{X}^{'}$ can be indirectly measured through a data mining process, where domain experts try to capture interesting regularities from the visual image. From this point of view, no visualization can be considered "wrong" — some visualizations are just better than the others. However, as demonstrated below, there are also statistical methods for validating the goodness of different visualizations. Some examples of the results obtained by using the suggested validation scheme are reported in the next section.

Given two visualizations $\mathbf{X}_1^{'}$ and $\mathbf{X}_2^{'}$ of the same high-dimensional data set $\mathbf{X}$, we can use criterion (1) for deciding between $\mathbf{X}_1^{'}$ and $\mathbf{X}_2^{'}$. However, although this criterion can be used for comparing alternative visualizations, it does not directly tell us how good the visualizations are. Of course, if we manage to find the global optimum of criterion (1), we have found the best possible visualization with respect to criterion (1) and distance measures $d(\cdot)$ and $d'(\cdot)$, but this is rarely the case in practice. In practical situations we have to settle for approximations of (1), and as the scale of this criterion varies with different coding schemes and distance measures, it would be difficult to give any general guidelines for validating the visualizations based on the absolute value of criterion (1).

On the other hand, as discussed earlier, data visualization can be considered as a data compression/transformation task where the goal is to maintain as much information as possible when converting $\mathbf{X}$ to $\mathbf{X}^{'}$. From this point of view, different visualizations could be evaluated by calculating the amount of information in the resulting reduced data set $\mathbf{X}^{'}$, and by comparing the result to the amount of information in the original data set $\mathbf{X}$. However, although this approach is theoretically valid, implementing it in practice would be technically demanding. Moreover, even if we could calculate the absolute information content of a data set, the result would not constitute a very intuitive measure. For these reasons, we suggest a different approach for validating different data visualizations, based on the idea of estimating the predictive accuracy in the reduced data set $\mathbf{X}^{'}$.

Recall from the previous section that our visualization scheme is based on the predictive distribution on the set of target attributes. This predictive distribution is computed by using the values of the attributes outside the target set. Consequently, the reduced data set $\mathbf{X}^{'}$ can be seen to contain information from those columns in $\mathbf{X}$ that contain the values of attributes $X_{m+1}, \ldots, X_N$. If we have succeeded in our data reduction process, we should be able to predict the target data $\mathbf{X}^m$ from the visual data $\mathbf{X}^{'}$, where $\mathbf{X}^m$ denotes the first m columns of $\mathbf{X}$. A natural prediction method for this purpose is the *nearest neighbor* method, where, given a partial input vector as a query, the closest vectors to the query vector in $\mathbf{X}^{'}$ are first identified, and the prediction is performed by using the target data in the corresponding rows in the matrix $\mathbf{X}^m$. The overall predictive accuracy can then be estimated by using some empirical technique, such as the *crossvalidation* method [22, 9]. Note that the suggested validation method, based on the nearest neighbor prediction accuracy measurement as described above, can be used for comparing two alternative visualizations $\mathbf{X}_1^{'}$ and $\mathbf{X}_2^{'}$, even in cases where the visualizations are produced with a different distance function $d(\cdot)$.

4 Empirical Results

To illustrate the validity of the suggested data visualization scheme, we performed a series of experiments with publicly available classification datasets from the UCI data repository [3]. The 24 data sets used and their main properties are listed in Table 1.

In each case, the data was visualized by using the class variable as the target set — in other words, the data vectors were visualized according to the classification distribution obtained by

Table 1:
The datasets used in the experiments.

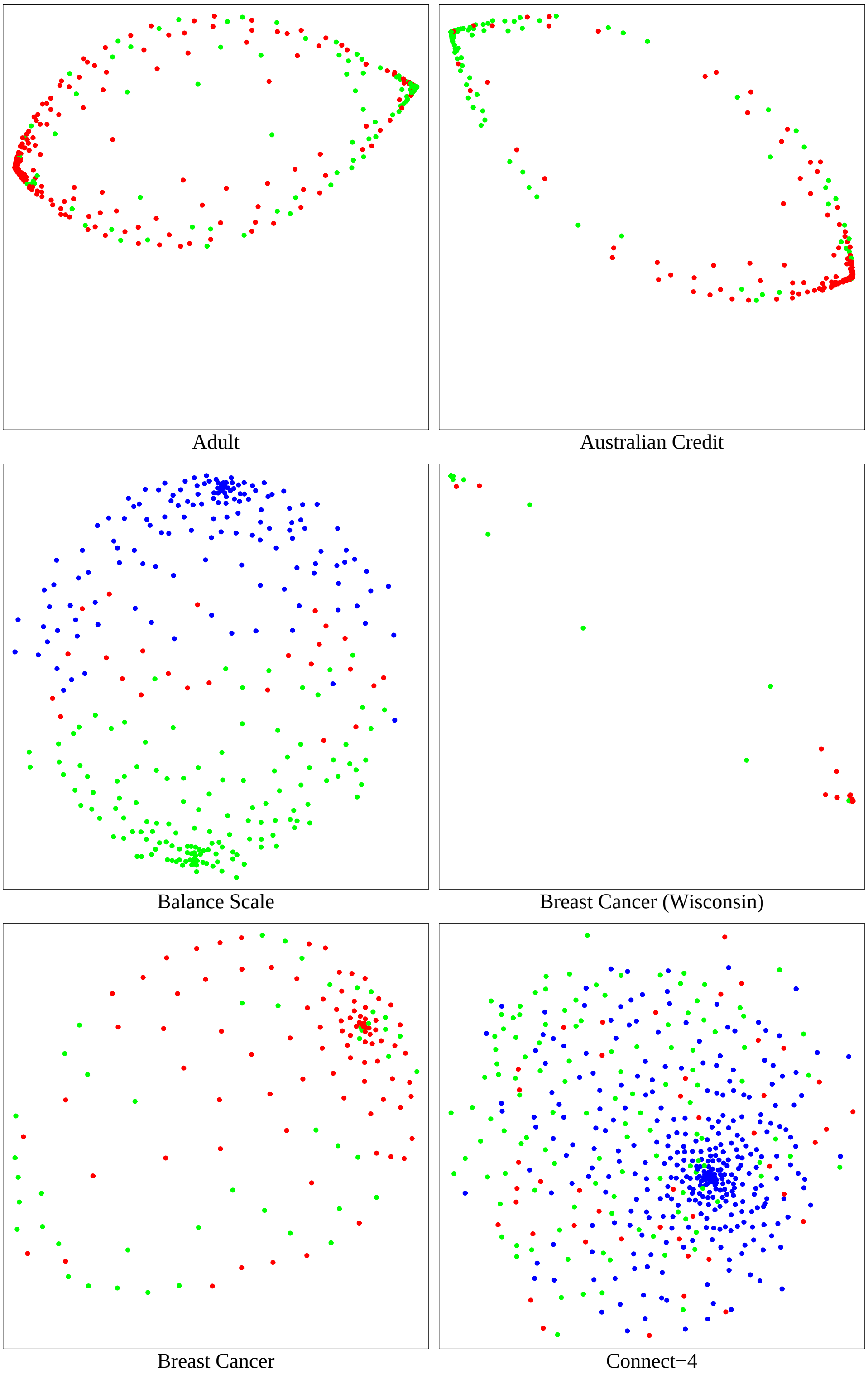
Dataset	Size	#Attrs.	#Classes
Adult	32561	15	2
Australian Credit	690	15	2
Balance Scale	625	5	3
Breast Cancer (Wisconsin)	699	11	2
Breast Cancer	286	10	2
Connect-4	67557	43	3
Credit Screening	690	16	2
Pima Indians Diabetes	768	9	2
German Credit	1000	21	2
Heart Disease (Hungarian)	294	14	2
Heart Disease (Statlog)	270	14	2
Hepatitis	155	20	2
Ionosphere	351	35	2
Iris Plant	150	5	3
Liver Disorders	345	7	2
Lymphography	148	19	4
Mole Fever	425	33	2
Mushrooms	8124	23	2
Postoperative Patient	90	9	3
Thyroid Disease	215	6	3
Tic-Tac-Toe Endgame	958	10	2
Vehicle Silhouettes	846	19	4
Congressional Voting Records	435	17	2
Wine Recognition	178	14	3

using a Bayesian network model M. As we in this empirical setup did not have access to any domain knowledge outside the data, the Bayesian network model had to be constructed from the data itself. This was done by splitting the data sets into two parts, of which the first part was used for constructing the Bayesian network model after which this data was thrown away and only the data in the second part was used for visualization. We understand that in practice it would make more sense to use all the data for constructing the model to be used, as well as for visualization, but by using this experimental setup we wanted to guarantee that we do not accidentally introduce any errors in the validation experiments exploiting the validation scheme described in the previous section.

The model structure used in this set of experiments was fixed to be the structurally simple naive Bayes model; this allowed us to avoid the problem of searching for a good model structure. A description of this model can be found in, e.g., [16]. Of course, using more sophisticated Bayesian network model structures would probably improve the results further, with the cost of increased computational requirements. For minimizing criterion (1), we used a very straightforward stochastic greedy algorithm. The resulting two-dimensional plots are given in Figures 1–4. Vectors with different class labels are shown with different types of markers.

An example of the corresponding three-dimensional plots is given in Figure 5, modified for grayscale printing. Different classes are shown with different colors. The colored versions of these images, produced by the POV-Ray™ software package, can be obtained through the CoSCo-group home page.

For estimating the quality of the visualizations produced, we used the validation scheme described in the previous section. The prediction methods used are listed in Table 2, and the corresponding prediction accuracies are shown in Table 3. It should be noted that results are



Adult

Australian Credit

Balance Scale

Breast Cancer (Wisconsin)

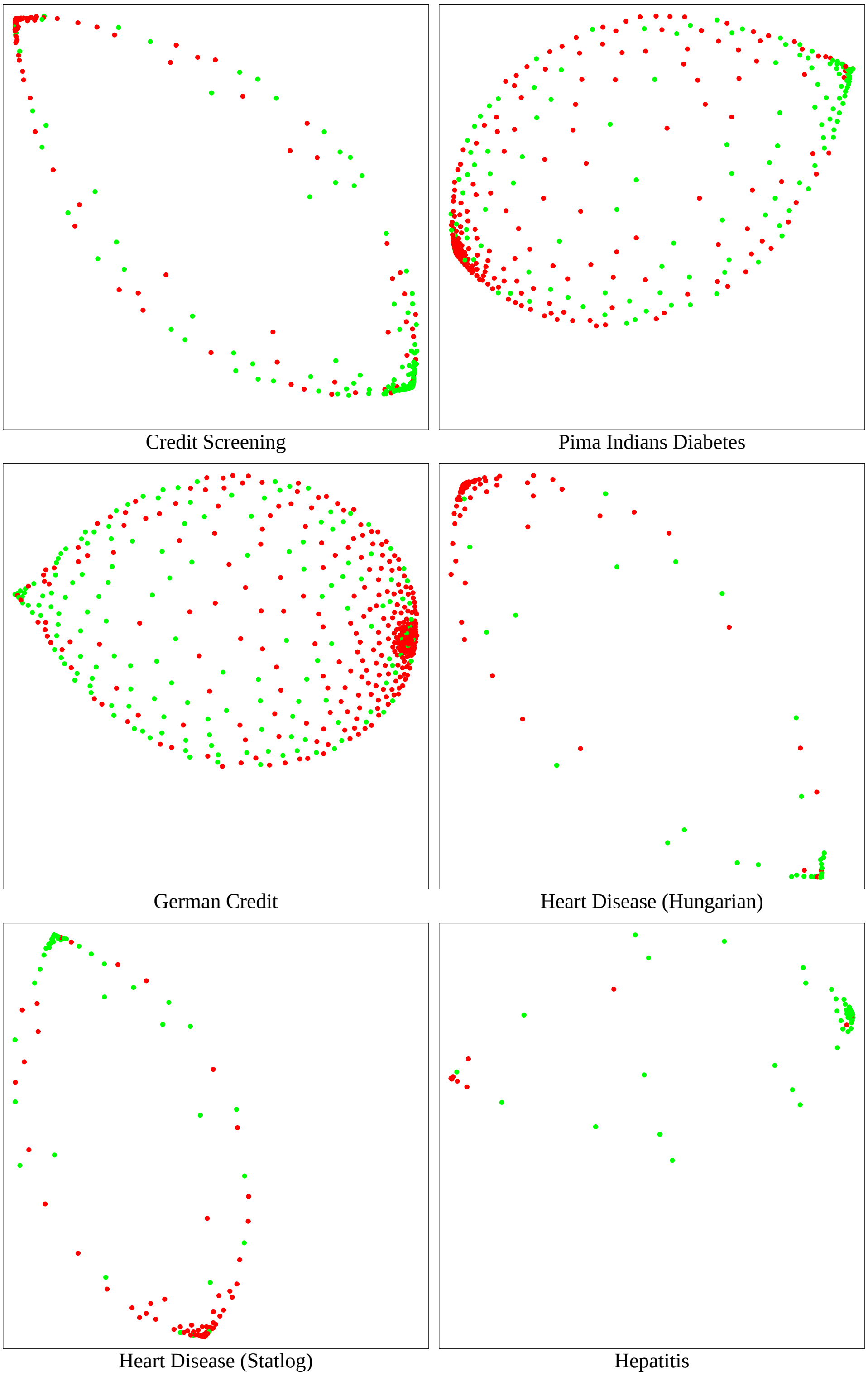
Breast Cancer

Connect−4

Figure 1: Two-dimensional visualizations of datasets 1–6 in Table 1.

not comparable to similar crossvalidation reported earlier, as we used some of the data for constructing the Bayesian network model, and this data was not used in the validation process.

From these results we can make two observations. Firstly, the visualizations obtained by



Credit Screening

Pima Indians Diabetes

German Credit

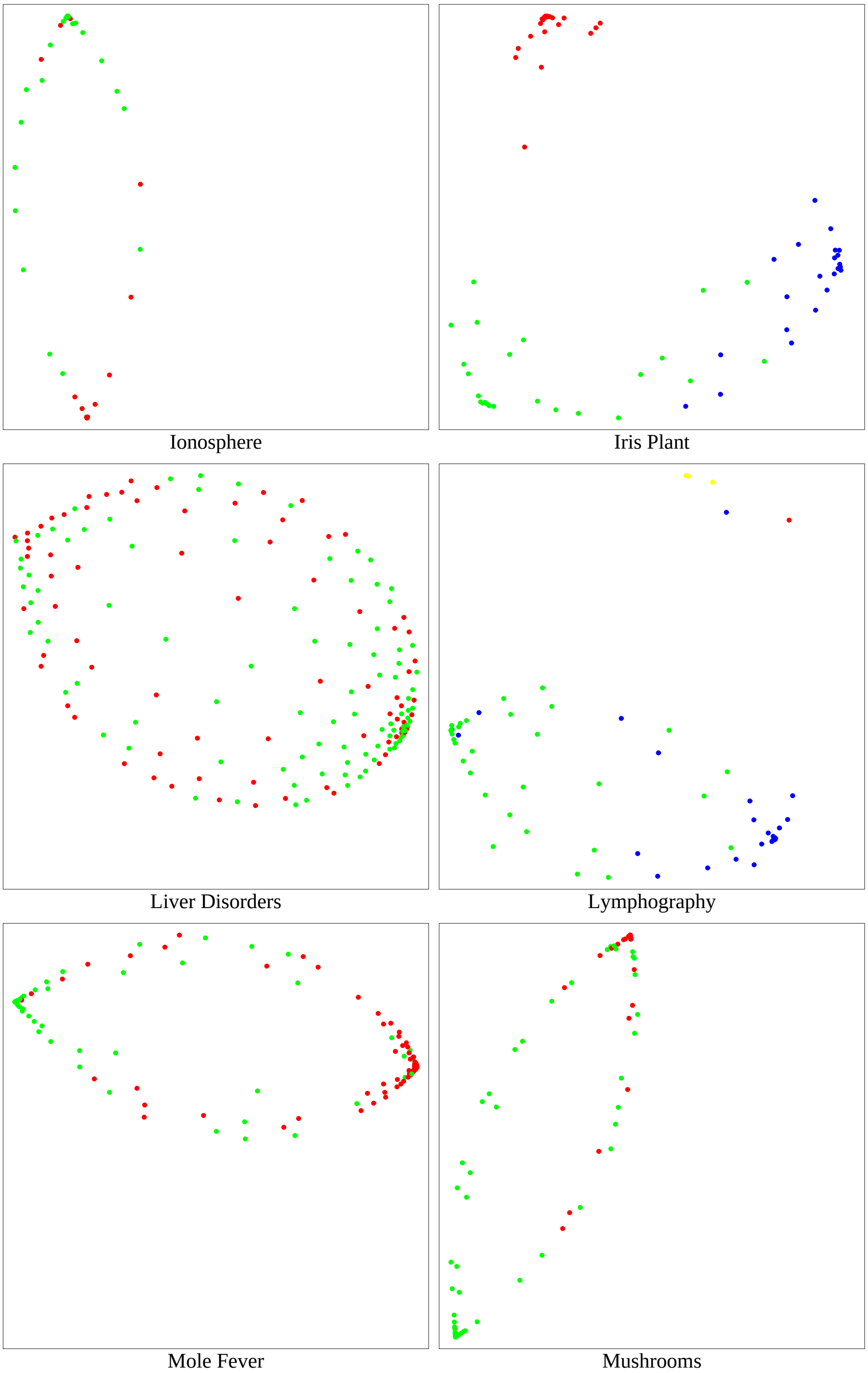
Heart Disease (Hungarian)

Heart Disease (Statlog)

Hepatitis

Figure 2: Two-dimensional visualizations of datasets 7–12 in Table 1.

the supervised model-based approach generally perform better in nearest neighbor classification than the visualizations produced by Euclidean multidimensional scaling. In the two-dimensional case, the model-based approach yields better results with 15 datasets (with one tie), and in the



Ionosphere

Iris Plant

Liver Disorders

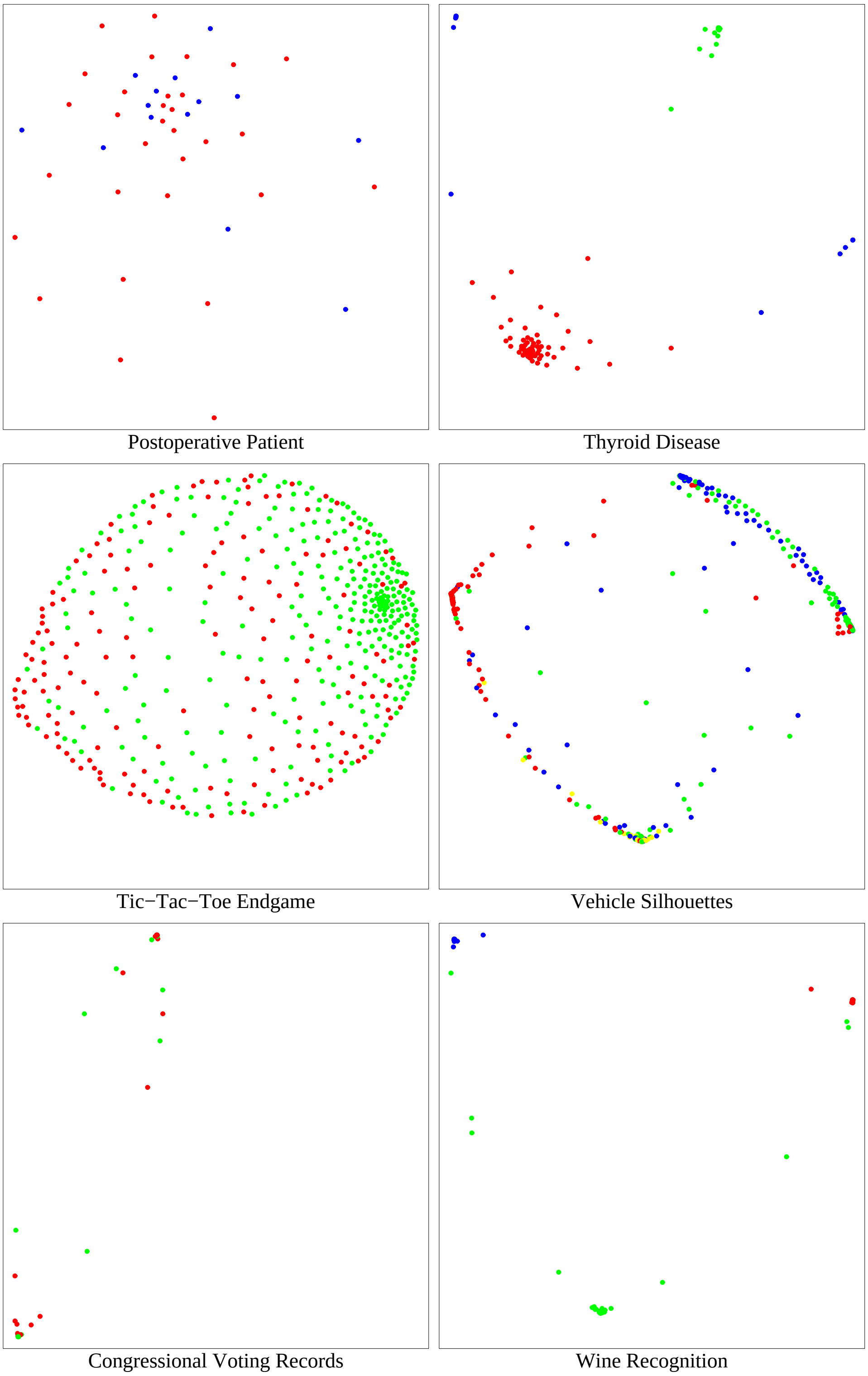
Lymphography

Mole Fever

Mushrooms

Figure 3: Two-dimensional visualizations of datasets 13–18 in Table 1.

three-dimensional case the results are better with 17 datasets (with 3 ties). Secondly, the overall classification accuracy with the reduced data sets is not significantly worse than with the methods exploiting the original multidimensional vectors. As a matter of fact, in some cases



Postoperative Patient

Thyroid Disease

Tic−Tac−Toe Endgame

Vehicle Silhouettes

Congressional Voting Records

Wine Recognition

Figure 4: Two-dimensional visualizations of datasets 19–24 in Table 1.

the classification accuracy can even be improved by using the visualization process. This means that the visualizations have been capable of preserving most of the information in the data, with respect to the supervised classification task chosen.

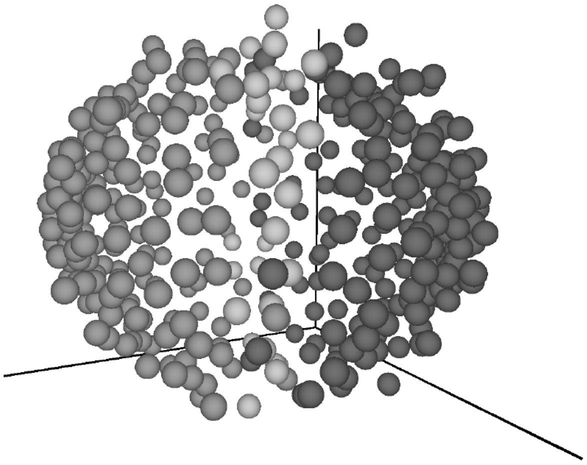


Figure 5: An example of a three-dimensional visualization, obtained with the "Balance" dataset.

The size of the neighborhood used in the nearest neighbor classifier in the experiments reported in Table 3 was 9. Runs with different size neighborhoods gave very similar results, as long as the neighborhood size was above one: it seems that the 1-nearest neighbor method does not measure very well the visual quality of the images produced. Visual inspection of the images produced conforms to the statistical analysis: the images produced by the model-based method look visually much better than the images produced by the Euclidean method. Actually, the model-based images look better even in cases where the nearest neighbor classifier gives worse crossvalidation results than the results obtained by using the Euclidean visualizations. An example of such a situation is given in Figure 6. This means that the nearest-neighbor based crossvalidation scheme does not reflect the visual quality of the images as well as we would have

Table 2:
The classification methods used

Label	Explanation
NB	Leave-one-out crossvalidated prediction accuracy of the Naive Bayes model with the original data set **X**.
NN	Leave-one-out crossvalidated prediction accuracy of the nearest neighbor classifier with the original data set **X**.
NN_{2D}^{NB}	Leave-one-out crossvalidated prediction accuracy of the nearest neighbor classifier with the two-dimensional data set obtained from the predictive distribution of the Naive Bayes model constructed from external training data.
NN_{3D}^{NB}	Leave-one-out crossvalidated prediction accuracy of the nearest neighbor classifier with the three-dimensional data set obtained from the predictive distribution of the Naive Bayes model constructed from external training data.
NN_{2D}^{EC}	Leave-one-out crossvalidated prediction accuracy of the nearest neighbor classifier with the two-dimensional data set obtained by Euclidean multidimensional scaling.
NN_{3D}^{EC}	Leave-one-out crossvalidated prediction accuracy of the nearest neighbor classifier with the three-dimensional data set obtained by Euclidean multidimensional scaling.

Table 3:
Crossvalidation results with the methods described in Table 2

Dataset	NB	NN	NN_{2D}^{NB}	NN_{3D}^{NB}	NN_{2D}^{EC}	NN_{3D}^{EC}
Adult	82.40	81.20	85.20	84.60	79.40	80.60
Australian Credit	84.93	84.35	85.51	86.09	83.77	81.74
Balance Scale	89.78	87.86	89.46	87.54	69.97	73.48
Breast Cancer (Wisconsin)	96.57	94.86	96.00	96.00	94.57	94.29
Breast Cancer	76.92	69.93	69.93	74.13	72.03	74.13
Connect-4	65.00	61.20	67.00	70.00	61.20	60.00
Credit Screening	84.64	84.06	80.29	81.45	81.16	80.87
Pima Indians Diabetes	74.48	76.04	72.66	73.96	70.83	73.18
German Credit	75.20	77.20	70.40	72.00	74.80	74.40
Heart Disease (Hungarian)	85.03	85.71	87.76	88.44	84.35	85.71
Heart Disease (Statlog)	87.41	85.93	83.70	86.67	83.70	87.41
Hepatitis	87.18	87.18	92.31	92.31	89.74	87.18
Ionosphere	90.34	77.84	92.61	92.05	84.66	85.23
Iris Plant	96.00	100.00	92.00	93.33	97.33	100.00
Liver Disorders	69.36	54.34	49.13	49.13	46.24	47.98
Lymphography	85.14	78.38	78.38	82.43	71.62	75.68
Mole Fever	89.67	78.87	85.45	89.67	75.59	80.75
Mushrooms	91.40	97.20	96.00	96.80	93.40	95.40
Postoperative Patient	64.44	55.56	55.56	60.00	62.22	60.00
Thyroid Disease	98.15	91.67	99.07	99.07	94.44	91.67
Tic-Tac-Toe Endgame	69.52	71.61	68.68	70.15	70.77	64.09
Vehicle Silhouettes	58.16	64.07	57.21	60.05	52.48	53.43
Congressional Voting Records	89.45	91.28	90.83	90.83	91.28	90.83
Wine Recognition	97.75	98.88	93.26	93.26	95.51	98.88

wanted, it can only be used for revealing general guidelines.

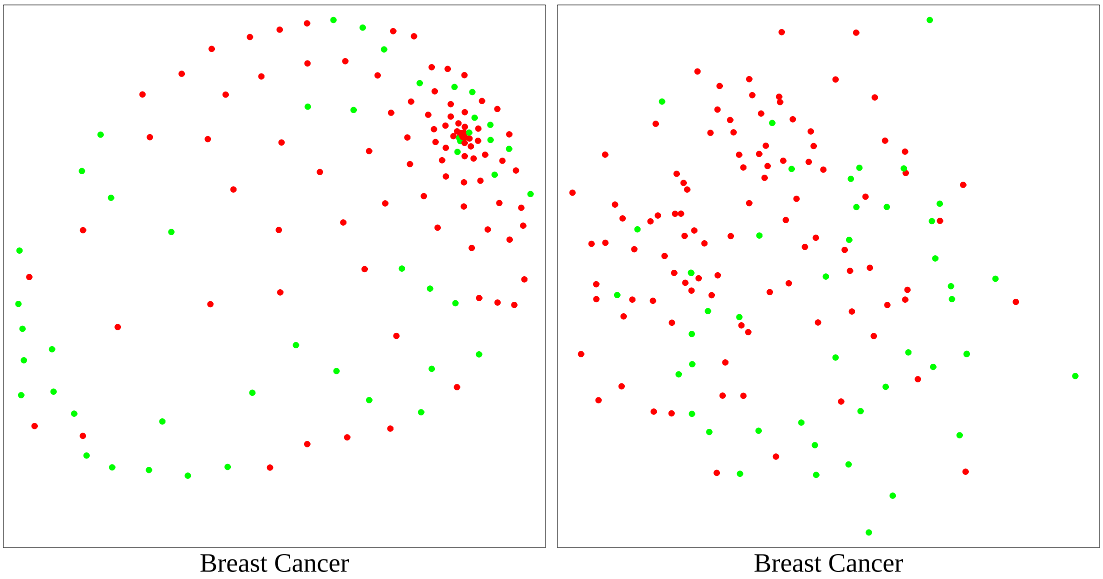


Figure 6: An example of the difference in the quality of the images produced by the model-based (left) and Euclidean (right) visualization.

One can of course argue that the basic Euclidean multidimensional scaling is not a very sophisticated data visualization method, and its performance can be substantially improved by using various techniques. This is without doubt true, but we can similarly argue that the

Bayesian network model used in this study was the structurally simple Naive Bayes model, and the performance of the model-based visualization scheme can most likely be improved a great deal by using more elaborate Bayesian network models. The algorithm used for minimizing criterion (1) was in both cases exactly the same, so the comparison should be fair in this sense also.

5 Conclusion

We have described a model-based visualization scheme based on the idea of defining a distance metric with respect to predictions obtained by a formal, probabilistic domain model. The scheme is supervised in the sense that the focus of the visualization can be determined by changing the target of the predictions. We also discussed methods for validating the quality of different visualizations, and suggested a simple scheme based on estimating the prediction accuracy that can be obtained by using the reduced, low-dimensional data. Our empirical results with publicly available classification datasets demonstrated that the scheme produces visualizations that are much better (with respect to the validation scheme suggested) than those obtained by classical Euclidean multidimensional scaling. Current examples were obtained by using a structurally simple Bayesian network model for producing the predictive distributions required, and a straightforward search algorithm for determining the visual locations of the data vectors. It is our belief that the results can be further improved by using more sophisticated methods in these tasks. This hypothesis will be examined in a project that aims at commercializing the ideas presented in a generic Bayesian data mining tool.

Acknowledgments

This research has been supported by the National Technology Agency (TEKES), and by the Academy of Finland.

References

[1] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, January 1998.

[2] C.M. Bishop and M.E. Tipping. A hierarchical latent variable model for data visualization. Technical Report NCRG/96/028, Neural Computing Research Group, Department of Computer Science, Aston University, 1998.

[3] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases, 1998. URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[4] A. Buja, D. Cook, and D.F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.

[5] E. Castillo, J. Gutiérrez, and A. Hadi. *Expert Systems and Probabilistic Network Models*. Monographs in Computer Science. Springer-Verlag, New York, NY, 1997.

[6] C. Chatfield and A. Collins. *Introduction to Multivariate Analysis*. Chapman and Hall, New York, 1980.

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

[8] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. John Wiley, 1973.

[9] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, June 1975.

[10] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[11] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[12] D. Heckerman. Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, 1(1):79–119, 1997.

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

[14] F. Jensen. *An Introduction to Bayesian Networks.* UCL Press, London, 1996.

[15] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[16] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. BAYDA: Software for Bayesian classification and feature selection. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 254–258. AAAI Press, Menlo Park, 1998.

[17] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. On Bayesian case matching. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop (EWCBR-98)*, volume 1488 of *Lecture Notes in Artificial Intelligence*, pages 13–24. Springer-Verlag, 1998.

[18] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. Comparing predictive inference methods for discrete domains. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 311–318, Ft. Lauderdale, Florida, January 1997.

[19] R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems.* John Wiley & Sons, New York, NY, 1990.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[21] R.D. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36(4):589–604, July-August 1988.

[22] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B)*, 36:111–147, 1974.