Pp. 413–427 in *Advances in Case-Based Reasoning* (Proceedings of the 3rd European Workshop), edited by I. Smith and B. Faltings. Lecture Notes in Artificial Intelligence, Volume 1168, Springer-Verlag, Berlin Heidelberg, 1996.

A Bayesian Framework for Case-Based Reasoning

Henry Tirri, Petri Kontkanen, and Petri Myllymäki

Complex Systems Computation Group (CoSCo)
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland

Abstract. In this paper we present a probabilistic framework for case-based reasoning in data-intensive domains, where only weak prior knowledge is available. In such a probabilistic viewpoint the attributes are interpreted as random variables, and the case base is used to approximate the underlying joint probability distribution of the attributes. Consequently structural case adaptation (and parameter adjustment in particular) can be viewed as prediction based on the full probability model constructed from the case history. The methodology addresses several problems encountered in building case-based reasoning systems. It provides a computationally efficient structural adaptation algorithm, avoids overfitting by using Bayesian model selection and uses directly probabilities as measures of similarity. The methodology described has been implemented in the D-SIDE software package, and the approach is validated by presenting empirical results of the method's classification prediction performance for a set of public domain data sets.

1 Introduction

In principle any computational system capable of reasoning can be placed somewhere in a spectrum based on the "knowledge-intensiveness" of the underlying approach. At one end of the spectrum are the systems relying on deep causal models [7], closely followed by the classic Knowledge-Based Systems (KBS). At the opposite end of the spectrum are the learning systems with weak prior domain knowledge which base their reasoning on models built from data [8]. Along this spectrum the reasoning process itself gradually changes from the deductive reasoning of knowledge-intensive approaches to inductive reasoning of data-intensive approaches, i.e., from logical inference to statistical inference.

In its most general form, *case-based reasoning* (CBR) can be viewed as an approach to computational reasoning that solves new problems by adapting previously successful solutions to similar problems [20, 39]. This general definition does not by itself place CBR systems in any particular place in the spectrum discussed above—the adaptation step in case-based reasoning can embed a substantial amount of domain knowledge. However, while CBR has its roots in cognitive modeling [34, 20], from an engineering point of view it can be seen as an attempt to address many of the problems recognized in the design of traditional knowledge-intensive systems, in particular the brittleness problem [2]. Consequently, one of the advocated potential advantages of case-based reasoning has been the fact that CBR systems can be built in cases where only weak prior domain knowledge is available [39], and the "top-down" approaches with explicit model representations cannot be applied. In such domains CBR systems construct *implicit domain models*. These models are defined by a case history (a set of past cases) together with the accompanying domain-specific procedures for case matching and adaptation. Such implicit models are then used to solve new problems in the same domain. If we further restrict ourselves to domains where cases can be represented as vectors of attribute values, and allow some of the recorded past cases to be incorrect (both being very realistic assumptions in industrial applications), the CBR approach in such domains can clearly be placed in the data-intensive end of the spectrum. This type of case-based reasoning approaches can be paralleled to the fundamental quest in statistical inference, i.e., finding the most probable model given data and weak prior knowledge. In this view, structural case adaptation [20] (and parameter adjustment in particular) can be seen as probabilistic prediction based on the model constructed from the case history. This observation has been our starting point for developing a probabilistic foundation for data-intensive case-based reasoning.

As discussed in [20], no old situation is ever exactly the same as a new one, hence old solutions must usually be adapted (for more discussion on the importance of adaptation in CBR systems see e.g., [19]). Traditional case-based reasoning systems base their structural adaptation on adjustment heuristics applied to the values of interest in relevant cases. This set of relevant cases is determined during the case retrieval phase typically by using a nearest neighbor matching [32] or indexes created by induction algorithms [33]. For many problem domains the basic approach of storing all the cases suffers from several drawbacks. First, the run-time computational costs for matching algorithms are high when the size of the case base grows large. However, a more fundamental problem in storing all cases in real applications is related to the well-known "overfitting" problem inherent to any inductive model construction method for noisy domains. Finally, the adaptation performance of a case-based system can be very sensitive to the selection of a proper similarity function [39]. In order to address some of these drawbacks of the basic approach, many suggestions for structuring the case storage have been proposed in the literature. From our point of view the most interesting is the "conceptual clustering" approach where *generalized episodes* [26] are used to group similar cases under a more general structure.

The approach presented in this paper adopts a probabilistic viewpoint and argues that the case base stored can be seen to form a probabilistic model of the problem domain, and that (structural) adaptation in such a probabilistic view is equal to prediction based on the model in question. In this approach the attributes are interpreted as random variables, and the case base is used to approximate the underlying joint probability distribution of the attributes. Case adaptation can now be given a rigorous interpretation as follows. Given a new case with some unknown attributes which, if known, could be used to solve the problem at hand, the predictive distributions for these attributes can be constructed by summing the predictions of individual cases weighted by the probability that the new case is a distorted version of the case in question. In this view each case is seen as a component distribution, which contributes to the joint distribution, i.e., the joint distribution is represented as a mixture of "case distributions". This basic case-based approach in numerical domains can in probabilistic terms be understood as a form of using kernel density estimators for inference (see e.g., [35]).

In practice a situation where all the case distributions are required for a good approximation is very rare, as the case base usually exhibits some cluster structure in the attribute domain space. In most cases the joint distribution can be approximated by a simpler mixture of distributions by giving a weighted sum of "cluster distributions", each of which gives the marginal attribute probability distributions conditioned by the cluster index. In fact, construction of prototypical structures such as Generalized Episodes by grouping cases sharing similar properties [26] can be understood as approximations to such simpler mixture structures [11, 38]. Thus the probabilistic approach offers also a natural way of structuring the case storage[1].

There are several advantages of using the above probabilistic view of CBR. If a good approximative representation of the problem domain distribution can be found, we can use a minimum risk Bayes decision rule in the adaptation phase [14]. It should also be observed that although the experimental results presented in Section 5 are only for classification tasks, in our approach *a full probability model is constructed.* Consequently, the predictive distributions can also be used directly for any adaptation task, since when the mixture model is constructed, all attributes are treated equally. In addition, such an adaptation computation can be performed efficiently [31, 30, 29].

As discussed above, to perform successful adaptation in an application with only weak domain knowledge, a model has to be extracted from the (noisy) set of cases. For constructing the component distributions from the case base, we have adopted the Bayesian approach (see e.g., [14]) which allows us to make a tradeoff between the complexity of our distribution structure and fit to the case data, thus resolving the overfitting problem. This combination of probabilistic finite mixture models with Bayesian model selection is akin to the approach adopted in the Autoclass system [5] with the notable difference in our focus to perform adaptation (prediction) rather than exploratory latent class analysis.

In this paper we describe a methodology for probabilistic case-based reasoning with discrete attribute values. The methodology addresses all the drawbacks discussed above: it provides a computationally efficient adaptation algorithm, avoids overfitting by using Bayesian model selection and uses directly probabilities as measures of similarity. The methodology described has been implemented in the D-SIDE software package. In order to validate our approach we present empirical results on a set of classification tasks (for discussion of exemplar-based classification see Chapter 13.1 in [20]). In order to be able to compare the probabilistic exemplar-based method with alternative classification approaches, we have used

[1] It should be observed that although not assumed in the discussion below, nothing in our approach prohibits storing also the individual cases in addition to the mixture distribution model. This can sometimes be useful for example for satisfying the needs of the explanation mechanisms.

public domain data sets (including data sets from the StatLog project [28]). Our results clearly demonstrate that the adaptation based on a probabilistic model is highly competitive for a wide spectrum of natural data sets with respect to other model construction approaches including decision trees or neural algorithms.

2 The finite mixture model

We model here the problem domain by m discrete random variables $X_1, \ldots, X_m$ (continuous attributes are at this point discretized by quantization). A *case* $\vec{d}$ is a data instantiation vector where all the variables X_i have been assigned a value, $\vec{d} = (X_1 = x_1, \ldots, X_m = x_m)$, where $x_i \in \{x_{i1}, \ldots, x_{in_i}\}$. A *case base* $D = (\vec{d}_1, \ldots, \vec{d}_N)$ is a set of N i.i.d. (independent and identically distributed) data instantiations, where each $\vec{d}_j$ is sampled from $\mathcal{P}$, the joint distribution of the variables $(X_1, \ldots, X_m)$.

We assume that the case base D is noisy, i.e., the data vectors can be regarded as distorted versions of some unknown "true" case. By noise we do not only mean incorrect or incomplete data caused by measurement errors or other failures in the data sampling process, but also the sampling errors caused by the finite size of the sample D. Consequently, this type of noise is inherently present in all the case bases available. Furthermore, we assume that the noise is not uniformly distributed, but the noisy cases form tightly clustered groups, where all the cases in a group correspond to the same unknown case. We have no means of determining the original unknown cases exactly, but we can form a probability distribution around each cluster of cases, based on the data vectors in the cluster. This probability distribution can be seen as a generalized case representation, a "prototypical case" representing all the data vectors in a cluster[2].

Let us now assume that we are given a clustering of the problem domain, i.e., the data instantiation space is partitioned into K groups of vectors. In finite mixture models [11, 38] the problem domain probability distribution is approximated as a weighted sum of mixture distributions:

$$P(\vec{d}) = \sum_{k=1}^{K} P(Y = y_k) P(\vec{d} | Y = y_k), \tag{1}$$

where the values of the discrete *clustering random variable* Y correspond to the separate clusters of the instantiation space, and each mixture distribution $P(\vec{d}|Y = y_k)$ represents the prototypical case corresponding to cluster k. As the data vectors belonging to the same cluster can be seen as noisy versions of some (unknown) point, we can assume the variables X_i inside each cluster to be independent and thus (1) becomes

$$P(\vec{d}) = P(X_1 = x_1, \ldots, X_m = x_m) = \sum_{k=1}^{K} P(Y = y_k) \prod_{i=1}^{m} P(X_i = x_i | Y = y_k).$$

[2] The prototypical cases reduce to normal cases if we allow only one data vector in each cluster.

It should be observed that the use of mixture distributions as prototypical cases implies that a given instance matches to several prototypical cases simultaneously with different probabilities, and thus the predictive distributions of variables can be computed as a weighted estimate from the marginal probability distributions for the variable in question, as shown in Section 3.

In our case both the cluster distribution $P(Y)$ and the intra-class conditional distributions $P(X_i|Y = y_k)$ are assumed to be multinomial. Thus a finite mixture model can be defined by first fixing K, the *model class* (the number of the mixing distributions), and then by determining the values of the model parameters $\Theta = (\alpha, \Phi), \Theta \in \Omega$, where $\alpha = (\alpha_1, \ldots, \alpha_K)$ and $\Phi = (\Phi_{11}, \ldots, \Phi_{1m}, \ldots, \Phi_{K1}, \ldots, \Phi_{Km})$, with the denotations

$$\alpha_k = P(Y = y_k), \Phi_{ki} = (\phi_{ki1}, \ldots, \phi_{kin_i}), \text{ where } \phi_{kil} = P(X_i = x_{il}|Y = y_k).$$

A detailed description of how to determine K and Θ, given a data sample D, can be found in Section 4.

3 Bayesian inference by finite mixture models

Given a finite mixture model, it is possible to solve various probabilistic reasoning tasks in a computationally efficient manner. The Bayesian *predictive inference* (see e.g. [3]) aims at predicting future (yet unobserved) quantities by means of already observed quantities. More precisely, if $\vec{d}$ is an unobserved data vector, its predictive distribution given observed data D is defined as

$$P(\vec{d}|D) = \int_\Omega P(\vec{d}|D, \Theta) P(\Theta|D) d\Theta. \tag{2}$$

So the predictive distribution can be considered as the expected distribution of $\vec{d}$ over all possible parameter settings (models). From (2) we see that the correct Bayesian procedure for making predictions would be to use all possible models by weighting them by their posterior density. However, we currently approximate the predictive distribution by using only a single parameter set $\hat{\Theta}$ which maximizes the density $P(\Theta|D)$, i.e. $\hat{\Theta}$ is a *maximum a posterior* (MAP) estimate of Θ. By setting the probability $P(\hat{\Theta}|D)$ to unity, the approximative predictive distribution becomes simply

$$P(\vec{d}|D) = P(\vec{d}|D, \hat{\Theta}) = P(\vec{d}|\hat{\Theta}), \tag{3}$$

where the last equality follows from the fact that $\vec{d}$ and D are conditionally independent given $\hat{\Theta}$.

With practical applications in mind, we are especially interested in finding the *conditional predictive distribution* of each uninstantiated X_i, given that some of the variables are instantiated, i.e. assigned a fixed value. Let $\mathcal{I} = \{i_1, \ldots, i_t\}$ be the indices of the instantiated variables, and let $\mathcal{X} = \{X_{i_s} = x_{i_s l_s}, s = 1, \ldots, t\}$ denote the corresponding assignments. Now we want to determine the distribution

$$P(X_i|D, \mathcal{X}) = \int_\Omega P(X_i|\Theta, \mathcal{X}) P(\Theta|D) d\Theta \tag{4}$$

for all $i \notin \mathcal{I}$. By using the approximation (3), (4) becomes

$$P(X_i = x_{il}|D, \mathcal{X}) = P(X_i = x_{il}|\hat{\Theta}, \mathcal{X}) = \frac{P(X_i = x_{il}, \mathcal{X}|\hat{\Theta})}{P(\mathcal{X}|\hat{\Theta})}$$

$$= \frac{\sum_{k=1}^{K}\left(P(Y = y_k|\hat{\Theta})P(X_i = x_{il}|Y = y_k, \hat{\Theta})\prod_{s=1}^{t}P(X_{i_s} = x_{i_s l_s}|Y = y_k, \hat{\Theta})\right)}{\sum_{k=1}^{K}\left(P(Y = y_k|\hat{\Theta})\prod_{s=1}^{t}P(X_{i_s} = x_{i_s l_s}|Y = y_k, \hat{\Theta})\right)}.$$

The conditional predictive distribution of X_i can clearly be calculated in time $\mathcal{O}(Ktn_i)$, where K is the number of clusters, t the number of instantiated variables and n_i the number of values of X_i. K is usually small compared to the sample size N, and thus the prediction computation can be performed very efficiently[3].

The predictive distributions can be used for classification and regression tasks. In classification problems, we have a special class variable X_c which is used for classifying data. We fix all the other variables and predict the value of X_c by using its conditional predictive distribution. With the simple 0/1-loss function, the predicted value is usually simply taken as the most probable value of the distribution in question, but alternative schemes emerge when using different loss functions. In more general regression tasks, we have more than one variable for which we want to compute the predictive distribution, given that the values of the other variables are instantiated in advance. Alternatively, finite mixture models can also be used for finding the maximal configuration (the most probable value assignment combination) for all the uninstantiated variables, given the values of the instantiated variables.

4 Learning finite mixture models from data

Let $D = (\vec{d}_1, \ldots, \vec{d}_N)$ be a random sample of N data instantiations, sampled from the problem domain distribution $\mathcal{P}$. By learning we mean here the problem of constructing a single finite mixture model $M_K(\Theta)$ which represents $\mathcal{P}$ as accurately as possible. The learning process can be divided into two separate phases: in the first phase we wish to determine the optimal value for K, the number of mixing distributions (the model class), and in the second phase we wish to find MAP parameter values $\hat{\Theta}$ for the chosen model class.

4.1 Determining the model class

In the Bayesian framework, the optimal number of mixing distributions (clusters) can be determined by evaluating the posterior probability for each model class $\mathcal{M}_K$ given the data:

$$P(\mathcal{M}_K|D) \propto P(D|\mathcal{M}_K)P(\mathcal{M}_K), K = 1, \ldots, N,$$

[3] If massively parallel hardware is available, the computations can be made even faster since the algorithms can be parallelized easily [30, 29].

where the normalizing constant $P(D)$ can be omitted since we only need to compare different model classes. The number of clusters can safely be assumed to be bounded by N, since otherwise the sample size is clearly too small for the learning problem in question.

Assuming equal priors for the model classes, they can be ranked by evaluating the *evidence* $P(D|\mathcal{M}_K)$ for each model class,

$$P(D|\mathcal{M}_K) = \int_{\Omega} P(D|\Theta, \mathcal{M}_K) P(\Theta|\mathcal{M}_K)\ d\Theta. \tag{5}$$

This integral can also be given an interpretation in the information-theoretic framework, where the *stochastic complexity* of a model class can be defined as minus the logarithm of the evidence. The evidence integral is very hard to evaluate since the dimension of the parameter space is usually very large $(K(1+\sum_{i=1}^{m} n_i))$, but the evidence can be approximated by using e.g. Laplace's method [18]. The quality of the approximations has been tested empirically in the finite mixture framework by using both synthetic [23, 6] and natural [24] data.

4.2 Determining the model class parameters

Let us assume that the model class $\mathcal{M}_K$ is fixed. In the Bayesian approach, the goal is to find the MAP (maximum a posteriori) estimate of parameters by maximizing the posterior density $P(\Theta|D)$,

$$P(\Theta|D) = \frac{P(D|\Theta)P(\Theta)}{\int_{\Omega} P(D|\Theta)P(\Theta)\ d\Theta},$$

where the normalizing factor is the evidence term (5) discussed in the previous section.

We assume that the prior distributions of the parameters are from the family of Dirichlet densities. Assuming that the parameter vectors α and Φ_{ki} are independent, the joint prior distribution of all the parameters is

$$\mathrm{Di}\,(\mu_1, \ldots, \mu_K) \prod_{k=1}^{K} \prod_{i=1}^{m} \mathrm{Di}\,(\sigma_{ki1}, \ldots, \sigma_{kin_i}) \tag{6}$$

with density

$$P(\Theta) = \Gamma\left(\sum_{k=1}^{K} \mu_k\right) \prod_{k=1}^{K} \left(\frac{\alpha_k^{\mu_k - 1}}{\Gamma(\mu_k)}\right) \prod_{k=1}^{K} \prod_{i=1}^{m} \left(\Gamma(\sum_{l=1}^{n_i} \sigma_{kil}) \prod_{l=1}^{n_i} \frac{\phi_{kil}^{\sigma_{kil}-1}}{\Gamma(\sigma_{kil})}\right). \tag{7}$$

The family of Dirichlet densities is *conjugate* (see e.g. [9]) to the family of multinomials, i.e. the functional form of parameter distribution remains invariant in the prior-to-posterior transformation, thus the choice of Dirichlet priors is justified.

The likelihood $P(D|\Theta)$ has two different forms depending on whether we know to which cluster each data vector belongs to or not. Let us first consider the positive case, i.e. *supervised learning*. The likelihood term can now be conveniently represented by introducing the *cluster indicator variables* $Z = (z_1, \ldots, z_N)$, $z_j = (z_{j1}, \ldots, z_{jK})$, where

$$z_{jk} = \begin{cases} 1, & \text{if } \vec{d}_j \in y_k, \\ 0, & \text{otherwise.} \end{cases}$$

It follows that

$$P(D|\Theta, \mathcal{M}_K) = \prod_{k=1}^{K} \left(\alpha_k^{h_k} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \phi_{kil}^{f_{kil}} \right),$$

where $h_k = \sum_{j=1}^{N} z_{jk}$ is the number of instantiations in cluster k, and $f_{kil} = \sum_{j=1}^{N} z_{jk} v_{jil}$ is the number of instantiations in cluster k with variable X_i having value x_{il}, where

$$v_{jil} = \begin{cases} 1, & \text{if } d_{ji} = x_{il}, \\ 0, & \text{otherwise.} \end{cases}$$

From the conjugacy property it follows that the posterior distribution is also a product of Dirichlet's and is given by

$$\mathrm{Di}\,(\mu_1 + h_1, \ldots, \mu_K + h_K) \prod_{k=1}^{K} \prod_{i=1}^{m} \mathrm{Di}\,(\sigma_{ki1} + f_{ki1}, \ldots, \sigma_{kin_i} + f_{kin_i})$$

with the corresponding density $P(\Theta|D)$ being analogous to (7). The MAP estimate $\Theta = \hat{\Theta}$ is the mode of this density, and since the parameters α_k and ϕ_{kil} appear disjointly, the mode can be determined by finding the modes of the individual Dirichlet's and is obtained by setting

$$\hat{\alpha}_k = \frac{h_k + \mu_k - 1}{N + \sum_{k=1}^{K} \mu_k - K}, \quad \hat{\phi}_{kil} = \frac{f_{kil} + \sigma_{kil} - 1}{h_k + \sum_{l=1}^{n_i} \sigma_{kil} - n_i}. \tag{8}$$

Assuming that the values of the cluster indicators are part of the training data D results in the simple naive Bayes model (see e.g. [16]). If, however, we do not consider the values of cluster indicators to be known, the learning problem becomes much more complex. In this *unsupervised learning* case we regard z_j's as random variables and assume that Z is a random sample of size N from a multinomial distribution with parameters α. The joint likelihood of D and Z is

$$P(D, Z|\Theta) = \prod_{j=1}^{N} \prod_{k=1}^{K} \left(\alpha_k \prod_{i=1}^{m} \phi_{kid_{ji}} \right)^{z_{jk}}, \tag{9}$$

and the marginal likelihood of D is obtained by summing over Z corresponding to all possible clusterings of data into K clusters. These clusterings are, however, exponential (K^N) in number, so exact MAP estimate cannot be found, and we are forced to use numerical approximation methods. The usual approach is to regard Z as missing data and estimate it iteratively [27]. This estimation can be performed by using the *Expectation-Maximization (EM)* algorithm [10]. The EM algorithm is an iterative algorithm, which monotonically increases the expected value of the posterior corresponding to incomplete data. It consists of two steps: Expectation(E) and Maximization(M). In E step, the expected values of the complete data sufficient statistics given the incomplete data and current parameter estimates are determined. In M step, the parameter values are updated in such a way that the obtained expected value of the posterior is maximized. The detailed derivation of the update formulas in our mixture case is somewhat involved and omitted here, but it is akin to the use of EM in [17]. The detailed derivation of the update formulas in our mixture case can be found in [25].

5 Empirical results

The approach described above has been implemented as part of a more general software package D-SIDE[4] for building systems capable of probabilistic inference. As already discussed in the introduction, we are motivated in finding models with good adaptation performance. In addition we are interested in evaluating the approach against other popular method families such as neural networks or decision tree algorithms.

Below we report results from an ongoing extensive experimentation with the probabilistic case-based reasoning method using publicly available datasets for classification problems. For comparison purposes, we have also collected performance results from the literature for alternative methods using the same datasets. We make no claims that the list of the results of alternative algorithms is exhaustive, however for each dataset we have included the best results we have found in the literature. The selection of datasets was done on the basis of their reported use, i.e., we have preferred datasets that have been used for testing many different methods over datasets with only isolated results. Many of the results are from the StatLog project [28], but we have also included more recent results. The descriptions of the datasets, the testing procedures used, and the best model classes (the number of mixtures) found for each data set are given in Table 1. The last column, the default value, denotes the success rate of a simple classifier, which classifies all the instances to the most common class.

It should be observed that with the exception of the DNA dataset, all our current results are crossvalidated, and when possible (for the StatLog datasets) we have used the same crossvalidation scheme as described in [28]. The same does not hold for many of the results for the other methods. In many cases the testing procedure either was not reported, or the best result with a single test set was given. The current performance results of all the algorithms (measured as classification success percentage) for individual datasets are presented as barcharts in Figures 1 and 2.

Even from these preliminary empirical results it can clearly be seen that our approach performs well not only when compared to memory-based methods

[4] A running Java™ demo of the D-SIDE software can be accessed through our WWW homepage at URL "http: //www.cs.Helsinki.FI/research/cosco/".

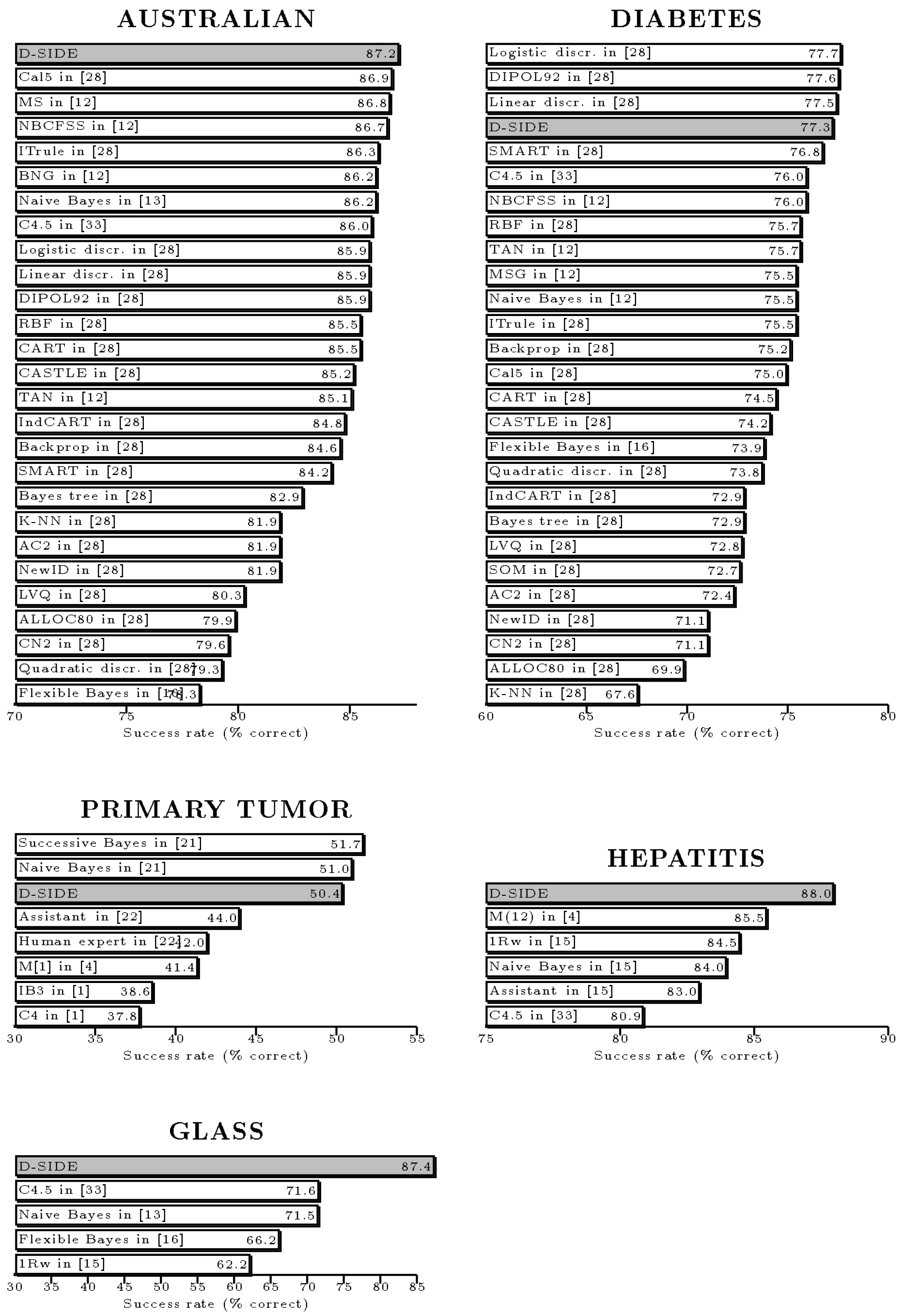

Fig. 1. Experimental results on the Australian, Diabetes, Primary tumor, Hepatitis and Glass datasets.

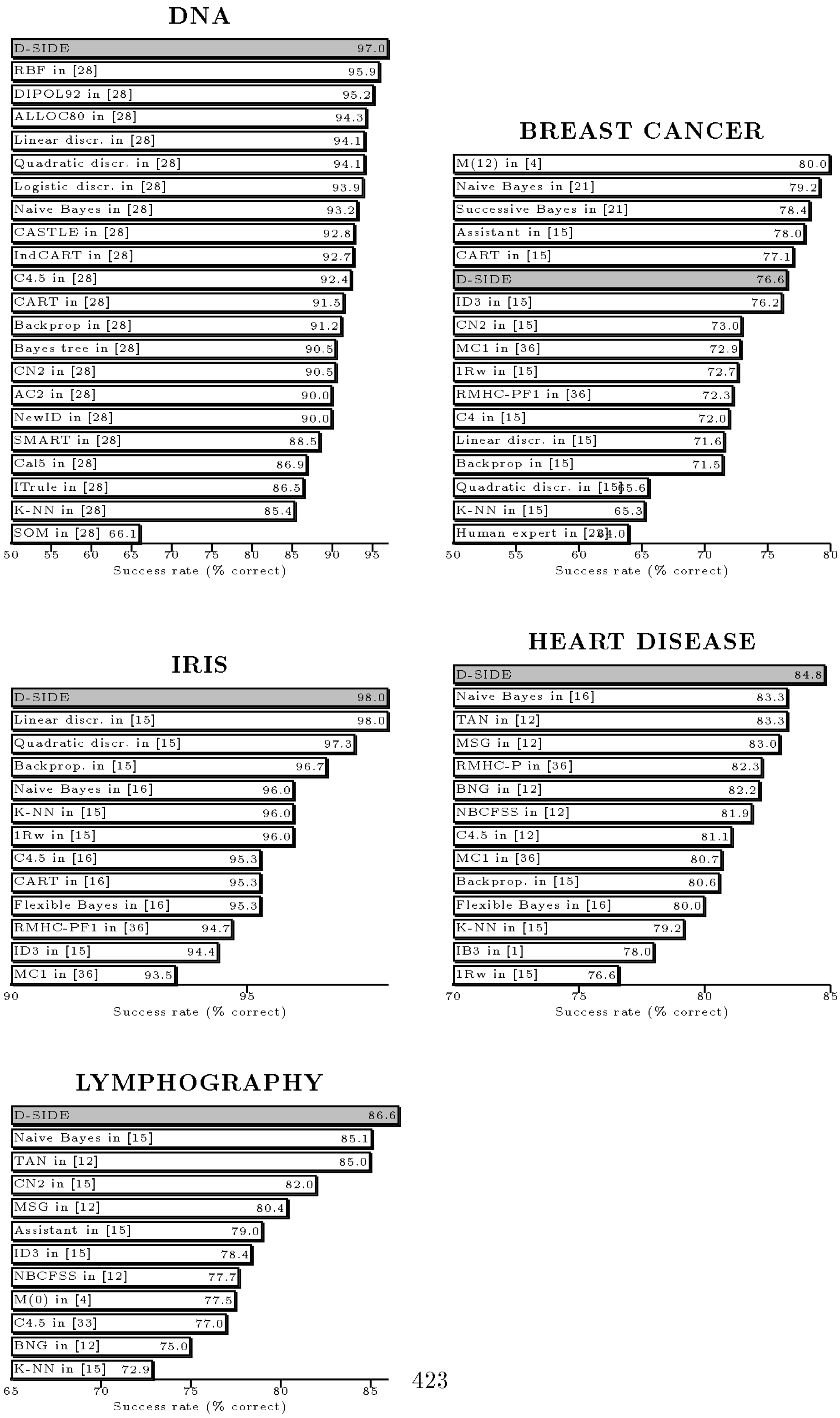

Fig. 2. Experimental results on the DNA, Breast cancer, Iris, Heart disease and Lymphography databases.

Table 1. The datasets and testing methods used in our experiments.

Data set	Size	#Attrs	#Classes	#Clusters	Test method	Default
Australian	690	15	2	17	10-fold CV	56.0
Breast cancer	286	10	2	21	11-fold CV	70.3
Diabetes	768	9	2	20	12-fold CV	65.0
DNA	3186	181	3	13	train&test	50.8
Glass	214	10	6	30	7-fold CV	40.7
Heart disease	270	14	2	8	9-fold CV	79.4
Hepatitis	150	20	2	9	5-fold CV	55.6
Iris	150	5	3	4	5-fold CV	33.3
Lymphography	148	19	4	19	5-fold CV	54.7
Primary tumor	339	18	21	21	10-fold CV	24.8

such as K-NN, IB3, or ALLOC80, but also with respect to decision tree methods and common neural network approaches such as backpropagation. The goodness of the results with respect to the memory based methods are not especially surprising as they can be seen as approximations to the finite mixture approach presented here [37]. A more interesting observation is that the EM algorithm finds MAP estimates that outperform also all other Bayesian approaches present in the StatLog comparison, including CASTLE, Naive Bayes, and Bayes tree algorithms.

In addition to the adaptation performance, it is also interesting to study if the constructed clusters can indeed be interpreted as "prototypical cases". Hence we investigated the clusters found with the D-SIDE environment, which provides a flexible graphical user interface for displaying predictive distributions and the mixture structure (see Figure 3). Since the meaning of the attributes in the data sets used for the empirical evaluation above were not documented, and thus could not be interpreted by a layman, for this cluster interpretation we used another public domain data set in a credit assessment domain. In this German credit data set we could give clear interpretations for 6 out of the 9 clusters. For example the most influential prototype case could be described consisting of working single women (age < 30, $4 - 7$ years in the current job), living in an apartment they owned, and requesting a small loan (less than 1800 DM) for furniture or home appliances with very little savings. It should be emphasized that since, like any case-based system, the mixture model built allows calculation of any predictive distribution, it can also be used explore dependencies between attributes. For example one can investigate what is the effect of being just recently employed to the purpose of the loan.

6 Conclusion

We have discussed the use of a probabilistic framework for building case-based reasoning systems. The presented Bayesian framework is realized by using the

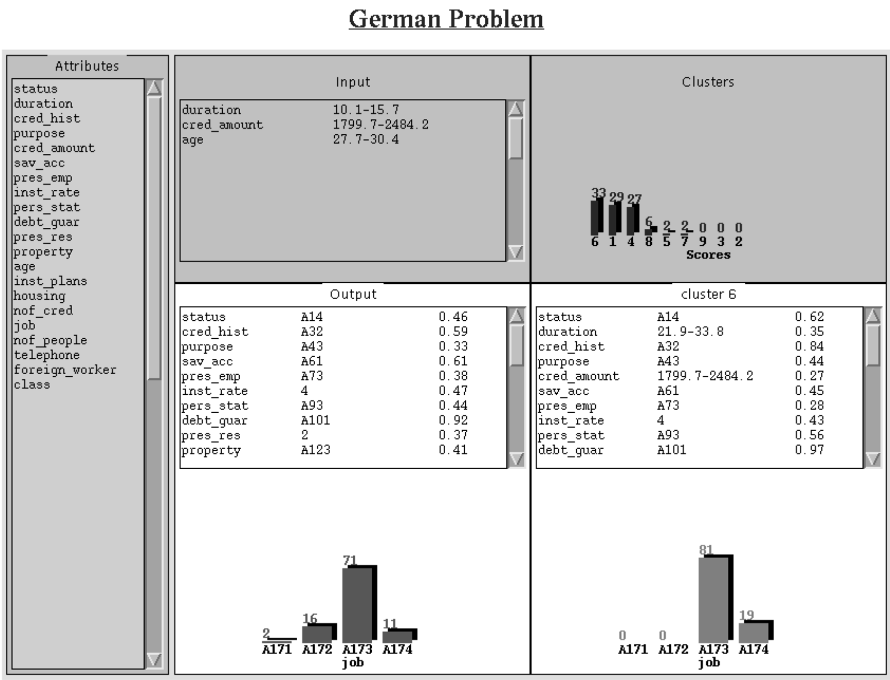

Fig. 3. A snapshot of the interface of the D-SIDE reasoning engine.

Expectation-Maximization (EM) algorithm for finding good models. We also presented empirical results on the method's adaptation performance for a set of public domain classification data sets, and compared the results to the performance of various machine learning and neural network methods. Although the experimentation is still ongoing, the current results clearly demonstrate that our Bayesian approach is highly competitive, and offers a very consistent performance over different types of data.

One should observe that the derivation presented here is given for discrete attributes (but continuous model spaces). The approach extends also to the case where attributes are real-valued. However, some limited experimentation we performed indicated that moving from discrete to continuous values does not necessarily improve the adaptation performance of the model. This is due to the additional assumptions of the distribution form (e.g., modeling continuous attributes with independent Gaussian distributions with appropriate conjugate priors) that are required, as well as to the fact that eventually when computations are performed, only limited numerical accuracy is available. We are currently investigating these issues.

Acknowledgments This research has been supported by the Technology Development Center (TEKES). The Primary Tumor, the Breast Cancer and the Lymphography data

were obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklič for providing the data.

References

1. D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.
2. R. Barletta. An introduction to case-based reasoning. *AI Expert*, pages 42–49, 1991.
3. J.M. Bernardo and A.F.M Smith. *Bayesian theory.* John Wiley, 1994.
4. B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In Y. Kodratoff, editor, *Machine Learning EWSL-91*, pages 138–150. Springer-Verlag, 1991.
5. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 54–64, Ann Arbor, June 1988.
6. D.M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. Technical Report MSR-TR-96-08, Microsoft Research, Advanced Technology Division, 1996.
7. W. Clancey. Heuristic classification. *Artificial Intelligence*, 27:289–350, 1985.
8. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.
9. M.H. DeGroot. *Optimal statistical decisions.* McGraw-Hill, 1970.
10. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
11. B.S. Everitt and D.J. Hand. *Finite Mixture Distributions.* Chapman and Hall, London, 1981.
12. N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *Proceedings of AAAI-96 (to appear)*, 1996.
13. N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference (to appear).* Morgan Kaufmann Publishers, 1996.
14. A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.
15. R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.
16. G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In P. Besnard and S. Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann Publishers, 1995.
17. M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
18. R.E. Kass and A.E. Raftery. Bayes factors. Technical Report 254, Department of Statistics, University of Washington, 1994.
19. M. Keane. Analogical asides on case-based reasoning. In S. Wess, K.-D. Althoff, and M Richter, editors, *Topics in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 21–32. Springer-Verlag, 1994.

20. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, 1993.
21. I. Kononenko. Successive naive Bayesian classifier. *Informatica*, 17:167–174, 1993.
22. I. Kononenko and I. Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6:67–80, 1991.
23. P. Kontkanen, P. Myllymäki, and H. Tirri. Unsupervised Bayesian learning of discrete finite mixtures. Manuscript, submitted for publication.
24. P. Kontkanen, P. Myllymäki, and H. Tirri. Comparing Bayesian model class selection criteria by discrete finite mixtures. In *Proceedings of the ISIS (Information, Statistics and Induction in Science) Conference*, Melbourne, Australia, August 1996. (To appear.).
25. P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report C-1996-9, University of Helsinki, Department of Computer Science, February 1996.
26. P. Koton. *Using experience in learning and problem solving*. PhD thesis, Massachusetts Institute of Technology, 1989.
27. R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley, 1987.
28. D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, London, 1994.
29. P. Myllymäki and H. Tirri. Bayesian case-based reasoning with neural networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 422–427, San Francisco, March 1993. IEEE, Piscataway, NJ.
30. P. Myllymäki and H. Tirri. Massively parallel case-based reasoning with probabilistic similarity metrics. In S. Wess, K.-D. Althoff, and M Richter, editors, *Topics in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 144–154. Springer-Verlag, 1994.
31. P. Myllymäki and H. Tirri. Constructing computationally efficient Bayesian models via unsupervised clustering. In A.Gammerman, editor, *Probabilistic Reasoning and Bayesian Belief Networks*, pages 237–248. Alfred Waller Publishers, Suffolk, 1995.
32. C. Owens. Integrating feature extraction and memory search. *Machine Learning*, 10(3):311–340, 1993.
33. J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
34. R. Schank. *Dynamic Memory: A theory of reminding and learning in computers and people*. Cambridge University Press, 1982.
35. D.W. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization.* John Wiley & Sons, New York, 1992.
36. D. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 293–301, 1994.
37. H. Tirri, P. Kontkanen, and P. Myllymäki. Probabilistic instance-based learning. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 507–515. Morgan Kaufmann Publishers, 1996.
38. D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, New York, 1985.
39. I. Watson and F. Marir. Case-based reasoning: A review. *The Knowledge Engineering Review*, 9(4):327–354, 1994.

This article was processed using the LaTeX macro package with LLNCS style