

On Bayesian Case Matching

Petri Kontkanen, Petri Myllymäki, Tomi Silander, and Henry Tirri

Complex Systems Computation Group (CoSCo)
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland
`cosco@cs.Helsinki.FI`
<http://www.cs.Helsinki.FI/research/cosco/>

Abstract. Case retrieval is an important problem in several commercially significant application areas, such as industrial configuration and manufacturing problems. In this paper we extend the Bayesian probability theory based approaches to case-based reasoning, focusing on the case matching task, an essential part of any case retrieval system. Traditional approaches to the case matching problem typically rely on some distance measure, e.g., the Euclidean or Hamming distance, although there is no a priori guarantee that such measures really reflect the useful similarities and dissimilarities between the cases. One of the main advantages of the Bayesian framework for solving this problem is that it forces one to explicitly recognize all the assumptions made about the problem domain, which helps in analyzing the performance of the resulting system. As an example of an implementation of the Bayesian case matching approach in practice, we demonstrate how to construct a case retrieval system based on a set of independence assumptions between the domain variables. In the experimental part of the paper, the Bayesian case matching metric is evaluated empirically in a case-retrieval task by using public domain discrete real-world databases. The results suggest that case retrieval systems based on the Bayesian case matching score perform much better than case retrieval systems based on the standard Hamming distance similarity metrics.

1 Introduction

In Bayesian modeling, a given database of feature vectors is regarded as a random sample from an unknown problem domain probability distribution. From this perspective, traditional model-based machine learning methods estimate this distribution by constructing a single model, e.g., a decision-tree or a neural network, from the sample data. In contrast to this “eager” approach, the model-free *case-based reasoning (CBR)* systems [1, 3, 14, 19, 21] base their predictions directly on the sample data, without producing any explicit models of the problem domain. This type of machine learning is often also referred to as *lazy learning*, since the algorithms defer all the essential computation until the prediction phase [2].

The CBR algorithms typically consist of two separate phases. In the first, *case retrieval* phase, the system recalls the most relevant cases for the task in

question, by first performing *case matching* and then selecting the best subset. In the case matching phase typically some distance function (e.g., Hamming distance) is used for scoring the cases with respect to the given query. In the second, *case adaptation* phase, the retrieved cases are used in problem solving for example by using majority voting or some kind of averaging, in order to produce a solution to the task at hand. It has been shown in various studies (see e.g., [18] for references) that this type of an approach can in some cases perform well in tasks involving prediction when the results are compared to those of alternative predictive approaches, e.g., machine learning algorithms. The method suffers, however, from several drawbacks when applied in practice (see, e.g., the discussion in [24]). Most importantly, the performance of the algorithms seems to be highly sensitive to the selection of the distance function used in case matching [4, 11].

In [24, 23, 20] we proposed a Bayesian framework for CBR based on probability theory with a particular set of underlying distributional assumptions known as finite mixture assumptions [8, 25]. The approach suggested can be seen as a *partially lazy approach* [2], i.e., a hybrid between the traditional machine learning and the CBR approach, which is based solely on the given data. The studies were based on the probabilistic viewpoint, where groups of data vectors are transformed into distributions, which can be seen as sample points in a distribution space. The predictive distributions required for making predictions could then be computed by using the CBR approach in this distribution space, i.e., by introducing a probabilistic “distance metric”. Somewhat similar frameworks have been suggested in [9, 10, 13].

In [16] we presented a new, improved probabilistic formalization of CBR of a purely lazy nature. The framework extended our earlier results by presenting a Bayesian approach for making (discrete) predictions directly from data, *without the transformation step between the original sample space and the distribution space*. Intuitively speaking, the new approach is based on the following notion: if we wish to make predictions by using only the sample data given, avoiding the notion of individual models, from the Bayesian point of view we can take this as a requirement for marginalizing, i.e., integrating, out the individual models. Quite interestingly, in [16] we showed that with the Bayesian CBR approach, the case adaptation problem can be solved directly in one pass, without the standard two-phase CBR methodology where the cases are first ordered according to some scoring function, and the best cases are then used in the adaptation phase for making predictions. Nevertheless, there are many situations where it would also be important to retrieve the most relevant cases from a given database. For example, the list of the relevant cases could be used by an explanation mechanism for increasing the credibility of the predictions. This raises the question of how to define and solve the case matching problem in the general Bayesian CBR setting. In [15] we described a probabilistic solution to this problem.

In this paper we extend and elaborate the work on the Bayesian CBR framework suggested, and focus on the Bayesian case matching metric. In Section 2 we first review the basic idea underlying Bayesian model-free inference. In Section 3

we present the Bayesian solution for the case matching problem, and discuss the reasons for choosing the metric proposed. In order to make our approach viable in practice, we have to make some restricting assumptions about the problem domain. As an example, in Section 4 we show how to implement the suggested approach with a set of simple, generally applicable independence assumptions. At this point it is important to emphasize that although the Bayesian approach requires one to fix some set of assumptions, which in some sense can be regarded as a “model”, the same applies for any CBR system: any distance metric used in a CBR system is based on the (implicit) assumption that the cases with a high matching score are relevant for the prediction task at hand. Consequently, strictly speaking there is no such thing as model-free inference, and all systems are based on some set implicit or explicit of assumptions. One of the main advantages of the Bayesian CBR framework is that it forces one to explicitly recognize all the assumptions made about the problem domain, a fact which helps in analyzing the performance of the resulting system.

Our earlier results [16] show that the Bayesian model-free inference scheme can perform extremely well in prediction tasks on a variety of different classification problems. Evaluating the Bayesian case matching metric empirically is a much more complicated issue. In Section 5.1 we discuss this question, and suggest a case retrieval setting that can be used for this purpose. In Section 5.2, we present results of a series of experiments, where the Bayesian case matching metric is evaluated empirically in the suggested case retrieval task by using publicly available real-world databases. The results suggest that case retrieval systems based on the Bayesian case matching score perform much better than case retrieval systems based on the standard Hamming distance similarity metrics used for comparison.

2 Bayesian model-free inference

Similarly to most of the prediction oriented CBR research, we will restrict ourselves here to the situation where cases are represented as feature vectors. From the Bayesian perspective, the case base D denotes a random sample of N i.i.d. (independent and identically distributed) data vectors $\mathbf{d}_1, \dots, \mathbf{d}_N$. For simplicity, we assume in the following that the data is coded by using discrete, finite-valued symbolic attributes from a set $X = \{X_1, \dots, X_m\}$, although the Bayesian approach described here can be easily extended to continuous (numeric) attributes as well, or even to a mixed-attribute case with both symbolic and numeric attributes. In probabilistic modeling we regard each discrete attribute X_i as a random variable with possible values from the set $\{x_{i1}, \dots, x_{in_i}\}$. Consequently, each data vector \mathbf{d} , i.e., case, is represented as a value assignment of the form $(X_1 = x_1, \dots, X_m = x_m)$, where $x_i \in \{x_{i1}, \dots, x_{in_i}\}$.

The underlying idea here is that the data vectors are assumed to be distributed according to some unknown probability distribution \mathcal{P} . Therefore, given a case base D and a new case with some unknown features, if we knew \mathcal{P} , we could predict values of the unknown features based on this distribution. More

precisely, given sample data D and the values of a subset U (the *clamped* variables) of the variables in X , the predictive distribution for the *free* variables $V = X \setminus U$ is now

$$\mathcal{P}(V \mid \mathbf{U} = \mathbf{u}, D). \quad (1)$$

Furthermore, each possible value combination \mathbf{v} for the variables V is assumed to be associated with some value called *utility* or *gain* $U(\mathbf{v})$. Informally, the utility gives us a measure of how beneficial this particular combination is, for example the profit of a particular sale action if the transaction is completed. Now we wish to establish a procedure which always gives us maximal gain in this decision-making setting. We regard the problem described above as a decision-theoretic formulation of the *case adaptation* problem: given some data, and the values for some of the domain attributes, the task is to determine the values for the remaining attributes in such a way that the utility function is maximized.

Theoretically speaking, there exists an optimal procedure for solving the case adaptation problem as formulated above. Namely, it can be shown (see e.g., [5]) that if we always choose the value assignment maximizing the expected utility, then the resulting procedure is optimal in the sense that in the long run, the expected amount of overall utility gained will be maximized. Nevertheless, determining the expected utility requires computing the predictive probability (1) for each possible value assignment $\mathbf{V} = \mathbf{v}$. As the “true” domain probability distribution \mathcal{P} is typically not known, we are left with approximative approaches.

In traditional machine learning, the problem domain probability distribution is typically approximated by

$$\mathcal{P}(V \mid \mathbf{U} = \mathbf{u}, D) \approx P(V \mid \mathbf{U} = \mathbf{u}, \Theta),$$

where Θ is a *model*, e.g., a decision-tree or a neural network, constructed from the sample data D . In contrast to this model-based approach, in the *case-based reasoning (CBR)* approach [1, 3, 14, 19, 21], the learning algorithms base their predictions directly on the sample data, without producing any specific models of the problem domain. Nevertheless, it should be noted that the original predictive distribution (1) does *not* contain any notion of individual models, so it can be regarded as the Bayesian (or decision-theoretic) formalization of the intuitive idea of model-free case-based reasoning. Furthermore, in the Bayesian framework, (1) can be written as

$$\mathcal{P}(V \mid \mathbf{U} = \mathbf{u}, D) \approx \int P(V \mid \mathbf{U} = \mathbf{u}, D, \Theta) P(\Theta \mid \mathbf{U} = \mathbf{u}, D) d\Theta. \quad (2)$$

Consequently, from the Bayesian point of view, we can regard the CBR approach as a requirement for marginalizing, i.e., integrating, out all the individual models. In the following, by *Bayesian CBR* we mean methods based on (2).

To make the Bayesian CBR approach formally valid, the models considered in the integral (2) must be limited to some restricted set of models — considering all the possible models in all their possible varying forms corresponds to considering all the possible probability distributions, which is of course infeasible.

It should be observed, however, that in practice we are always forced to make some restricting assumptions about the problem domain, although this fact is not always explicitly recognized. For example, in many traditional CBR systems, the algorithms typically use a distance function (e.g., Euclidean distance) for the feature vectors in order to determine the most relevant data items for the prediction task in question. The use of a specific distance function implicitly assumes that the distance function is relevant with respect to the problem domain probability distribution, and hence restricts the set of distributions considered.

In the following, let Ψ denote the set of assumptions made about the problem domain. If the assumptions Ψ define a parametric model class \mathcal{M} , i.e., a parametric model form (e.g., a specific neural network topology), where each model (parameter instantiation) $\theta \in \mathcal{M}$ defines a probability distribution on the problem domain, then we can rewrite (2) in a more formal manner as

$$\mathcal{P}(V | U = \mathbf{u}, D) \approx \int P(V | U = \mathbf{u}, D, \theta, \Psi) P(\theta | U = \mathbf{u}, D, \Psi) d\theta, \quad (3)$$

where the integration goes over all the models in \mathcal{M} . Consequently, as we are uncertain about which of the probability distributions $P(V | U = \mathbf{u}, D, \theta, \Psi)$ is the most accurate representation of the problem domain distribution \mathcal{P} , in the Bayesian setting this problem is solved by averaging over all the possible distributions. In this infinite sum, each probability distribution is weighted by $P(\theta | U = \mathbf{u}, D, \Psi)$, the probability that the distribution generated by model θ coincides with the problem domain probability distribution \mathcal{P} where D and \mathbf{u} are generated from.

3 A Bayesian metric for case matching

In the previous section we discussed the Bayesian approach for case-based inference. If the integral in (3) can be solved analytically, it means that the corresponding predictive distribution can be used as a tool for solving the case adaptation problem in an optimal manner, with respect to the set of distributions defined by the assumptions made. However, it should be noted that the Bayesian approach does not follow the standard two-phase CBR methodology, where the cases are first ordered according to some scoring function, and the best cases are then used in the adaptation phase for finding configurations \mathbf{v} so that the gained utility will be high. This means that the Bayesian case adaptation approach described in [16] cannot be used for case matching tasks, i.e., for ranking the stored cases $\mathbf{d} \in D$ with respect to their similarity to the given query. Nevertheless, it is quite evident that there are many situations where instead of trying to predict the values of the free variables V as described above, a more important task would be to retrieve the most relevant cases from a database D , given a query $U = \mathbf{u}$. For example, the list of the relevant cases could be used by an explanation mechanism for increasing the credibility of the predictions. This raises the question of how to define and solve a “pure” case retrieval problem

in the general Bayesian CBR setting, in a similar way as the case adaptation problem was solved above.

An intuitively appealing solution to the above mentioned problem is to define a case matching score \mathcal{S} for a case \mathbf{d}_j as

$$\mathcal{S}(\mathbf{d}_j | \mathbf{u}) \stackrel{\text{def}}{=} P(\mathbf{d}_j | \mathbf{u}, D, \Psi).$$

Unfortunately, there are several reasons why using this measure is not reasonable in practice: Firstly, the measure produces a non-zero score only to those cases which are fully consistent with the given query \mathbf{u} . Nevertheless, in many practical case retrieval applications, this kind of behavior is not acceptable. On the other hand, it should also be noted that this score favors cases with high “prior” probability $P(\mathbf{d}_j | D)$ (probability given the sample data without a query). This follows from the fact that the score tries to find the most probable case vector consistent with the query from the probability distribution defined by the case base D . This, however, does not match our intuitive notion of the case matching task, where the goal is to find from the case base D the case most *similar to* the query \mathbf{u} , even if the solution would be a case with a very low prior probability.

In order to avoid these problems, we instead propose the following Bayesian case matching scoring function:

$$\mathcal{S}(\mathbf{d}_j | \mathbf{u}) \stackrel{\text{def}}{=} P(\mathbf{u} | \mathbf{d}_j, D, \Psi). \quad (4)$$

Intuitively speaking, the suggested case matching score ranks the cases according to the following question: “Which of the cases in D should be duplicated if one wishes to maximize the probability of the given query $\mathbf{U} = \mathbf{u}$?”. On the other hand, noting that

$$\mathcal{S}(\mathbf{d}_j | \mathbf{u}) \stackrel{\text{def}}{=} P(\mathbf{u} | \mathbf{d}_j, D, \Psi) \propto \frac{P(\mathbf{d}_j | \mathbf{u}, D, \Psi)}{P(\mathbf{d}_j | D, \Psi)},$$

we see that the cases are ranked according to their “posterior” probability (probability given the query and the sample data), normalized by their “prior” probability. This means that also cases with very low initial probability can get a high score if they match the given query well, which agrees with our intuitive notion of a good case matching score.

It should be noted that the scoring metric suggested is “soft” in the sense that the winning case may violate one or more of the variable-value assignments in the query $\mathbf{U} = \mathbf{u}$. Obviously, if the query \mathbf{u} can not be regarded as a “wish-list”, but as a set of absolute constraints which should not be violated, then this type of hard constraint situation can be handled easily by restricting the search for the matching case to those cases consistent with the given query.

4 Computing the Bayesian case matching metric

Similarly to the Bayesian model-free inference scheme described in Section 2, from the axioms of probability theory we can deduce that the Bayesian case

matching metric (4) can be computed by marginalizing (integrating) over all the possible models consistent with the given assumptions Ψ :

$$\mathcal{S}(\mathbf{d}_j | \mathbf{u}) = P(\mathbf{u} | \mathbf{d}_j, D, \Psi) = \int_{\Theta \in \mathcal{M}(\Psi)} P(\mathbf{u} | \Theta, \Psi) P(\Theta | \mathbf{d}_j, D, \Psi) d\Theta. \quad (5)$$

In order to be able to use this formula in practice, the assumptions Ψ have to be such that the integral in (5) can be solved analytically, or at least approximated well. One commonly used simplifying assumption is that the other variables are independent given the value of a special class variable, denoted here by X_m . This same assumption is also used in constructing a model called the *Naive Bayes classifier*. In this case, the joint probability distribution for a data vector \mathbf{d} can be written as

$$P(\mathbf{d} | \Theta) = P(X_m = x_m) \prod_{i=1}^{m-1} P(X_i = x_i | X_m = x_m).$$

Consequently, a single distribution P can be uniquely determined by fixing the values of the parameters $\Theta = (\alpha, \Phi)$, where $\alpha = (\alpha_1, \dots, \alpha_K)$ and $\Phi = (\Phi_{11}, \dots, \Phi_{1,m-1}, \dots, \Phi_{K1}, \dots, \Phi_{K,m-1})$. Here K denotes the number of values of the class variable X_m , α_k the probability $P(X_m = k)$, and Φ_{ki} denotes a vector $(\phi_{ki1}, \dots, \phi_{kin_i})$, where $\phi_{kil} = P(X_i = x_{il} | X_m = x_m)$.

For being able to implement the Bayesian case matching in practise, we also need some technical assumptions. In the following we assume that $\alpha_k > 0$ and $\phi_{kil} > 0$ for all k, i , and l , and that both the class variable distribution $P(X_m)$ and the intra-class conditional distributions $P(X_i | k) = P(X_i | X_m = k)$ are multinomial, i.e., $X_m \sim \text{Multi}(1; \alpha_1, \dots, \alpha_K)$, and $X_i | k \sim \text{Multi}(1; \phi_{ki1}, \dots, \phi_{kin_i})$. Since the family of Dirichlet densities is *conjugate* (see e.g. [7]) to the family of multinomials, it is convenient to assume that the prior distributions of the parameters are from this family. More precisely, we let $(\alpha_1, \dots, \alpha_K) \sim \text{Di}(\mu_1, \dots, \mu_K)$, and $(\phi_{ki1}, \dots, \phi_{kin_i}) \sim \text{Di}(\sigma_{ki1}, \dots, \sigma_{kin_i})$, where $\{\mu_k, \sigma_{kil} | k = 1, \dots, K; i = 1, \dots, m-1; l = 1, \dots, n_i\}$ are the *hyperparameters* of the corresponding distributions. Finally, assuming that the parameter vectors α and Φ_{ki} are independent, and applying the results in [6, 12], the Bayesian case matching score (5) for case \mathbf{d}_j can be computed by

$$\begin{aligned} \mathcal{S}(\mathbf{d}_j | \mathbf{u}) &= \sum_{k=1}^K P(\mathbf{u}, X_m = k | \mathbf{d}_j, D, \Psi) \\ &= \sum_{k=1}^K \frac{h_k + \mu_k}{N + \sum_{c=1}^K \mu_c} \prod_i \frac{f_{k i u_i} + \sigma_{k i u_i}}{h_k + \sum_{l=1}^{n_i} \sigma_{k i l}}, \end{aligned} \quad (6)$$

where the product goes over all the variables instantiated in the query \mathbf{u} , and h_k and f_{kil} are the sufficient statistics of the data $D \cup \mathbf{d}_j$, i.e., h_k is the number of cases where $X_m = k$, and f_{kil} is the number of cases where $X_m = k$ and $X_i = x_{il}$. If no expert knowledge about the problem domain is available, it is

usually reasonable to use the noninformative uniform prior, in which case all hyperparameters μ_k and σ_{kij} are set to 1. A more detailed discussion of the priors can be found in [17].

To get an idea of the implementation performance of the above Bayesian case matching formula, we performed an experiment where we generated an artificial case base with 2 million data vectors, each consisting of 14 discrete attributes having on the average 3-4 values. Using a 200MHz Pentium machine running Linux, matching a given query against the whole case base, corresponding to performing 2 million case matches, took 27.23 seconds, i.e., about 13.6 microseconds per matching operation (approximately 74000 matches per second). The size of this case base serves also as an example of the scalability of the approach.

5 Empirical results

5.1 Experimental setup

Validating a case matching metric is a difficult problem. Of course, in practical situations a metric can be evaluated by building a case retrieval system, and by using a domain expert for evaluating the performance of the case retrieval process. This, however, does not produce an objective criterion for comparing different metrics, as the evaluations of domain experts are based on subjective considerations. On the other hand, in machine learning, objective measures of prediction accuracy, notably the *cross-validation* scheme [22], has been used frequently for comparing the performance of alternative classification methods. In leave-one-out cross-validation, each data vector is classified in turn by using the $N - 1$ remaining data vectors as the training data. The cross-validated classification accuracy is then the average of all the N individual classifications. Inspired by this scheme, we planned the following setup for comparing different case matching metrics.

In our setup, the idea is to make a duplicate of a case, distort it, and try to recover the original case by matching the distorted case against the whole case base. For measuring the success of the recovery process, we propose the following *rank measure*: Given a query (a distorted version of a duplicate of one of the cases), all cases in the case base are first ordered according to the similarity metric used, with respect to the query. In situations where there are several cases with the same value of the similarity score as with the correct case, the other cases are put ahead of the correct case in the ordering. The rank is now defined to be the position of the correct case in the resulting ordering. Consequently, a rank of 1 means that we have succeeded in recovering the original case, and a rank of 5 means that there are 4 cases (with a higher or equal value of similarity score) our similarity metrics prefers over the correct case. This whole procedure is then repeated N times, once for each case in the case base (as in leave-one-out crossvalidation), and the actual rank measure is then the average of the N ranks obtained.

For distorting the cases in this test setup, it may seem at first natural to use a procedure, where some of the components of the selected case are changed to

random values. This kind of a setup would however produce severe problems, as the following example illustrates. Say we take case \mathbf{d}_j to be the original case, and through the random distortion process produce a query vector equal to some other case, say \mathbf{d}_i . Now any reasonable matching metric surely would say that the closest vector nearest to the query vector \mathbf{d}_i (the distorted \mathbf{d}_j) would be \mathbf{d}_i , but we would obtain rank 1 by giving the case \mathbf{d}_j the highest similarity score. For this reason, the distortion process can not be based on randomly changing the components of the chosen vector. Instead, in our experiments we used a setup where we randomly removed some of the components of the chosen case, and used the remaining components as the query with which to match the cases in the case base. To see how the rank measure behaves as a function of the number of components in the query, we used a procedure where we first removed one component from the selected case, then two, and so on. As the outcome of this test setup is dependent on the ordering in which the components are removed, this whole process was repeated 100 times by using random orderings, and the rank was defined to be the average of the 100 ranks obtained. This means that testing a case metric on a case base with N vectors requires $100N^2$ case matching operations, and that the ranks reported are averages of this many numbers.

5.2 The results

To validate the suggested Bayesian case matching score (4), we used several public domain classification data sets of varying size from the UCI data repository¹. The datasets were all discretized by using a very simple discretization scheme where the data intervals were chosen so that each interval contained an equal number of data points. The datasets used in the experiments are described in Table 1. It should be emphasized that the datasets were chosen randomly, not by any selection process.

Table 1. The data sets used in the experiments.

Data set	Size	Attributes	Classes
Breast cancer	286	10	2
Glass	214	10	6
Heart disease	270	14	2
Hepatitis	150	20	2
Iris	150	5	3
Lymphography	148	19	4

In Figure 1, the average ranks are plotted as a function of the number of fixed variables, i.e., the variables that were not removed from the query in the distortion process described in the previous section. In this picture we see that with all six case bases, as the number of fixed variables increases, the case retrieval

¹ <http://www.ics.uci.edu/~mllearn/>.

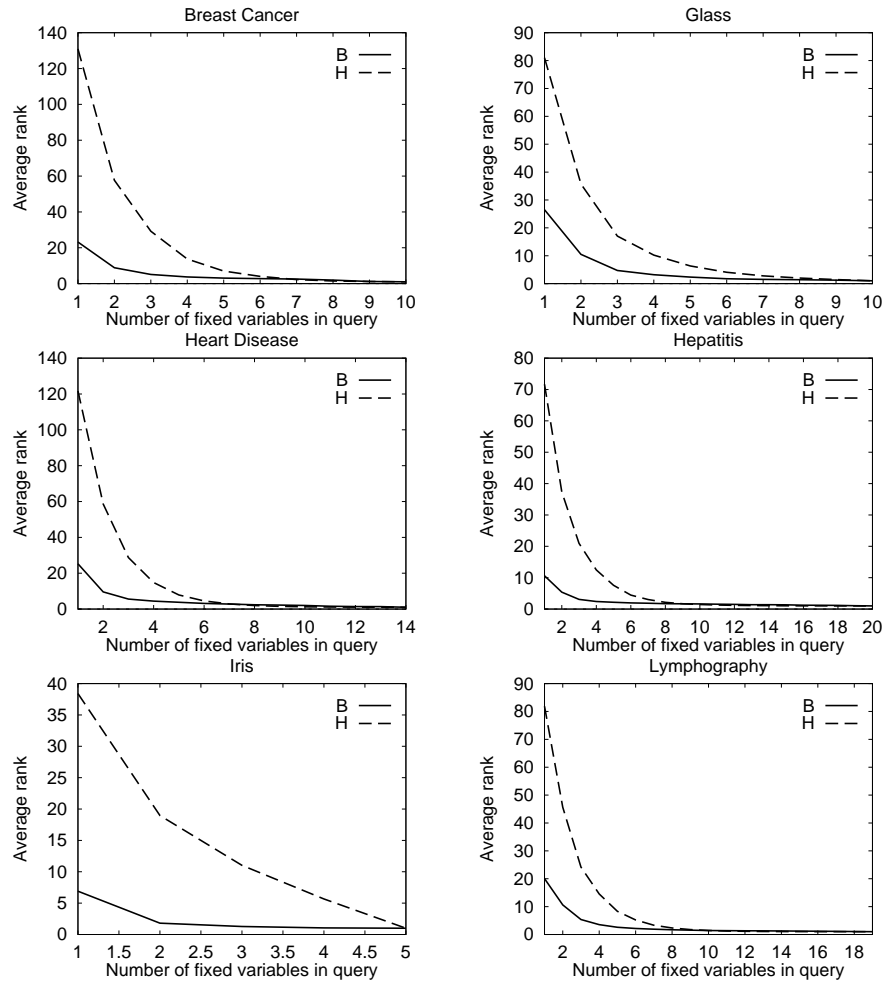


Fig. 1. Average ranks with the Bayesian (B) and Hamming (H) matching score as a function of the number of the fixed variables in the query.

system recovers the original case with both the Bayesian matching metric and the Hamming metric (the rank converges to one). However, with the Bayesian matching metric the rank converges to 1 much faster, which means that the metric is more efficient in extracting the relevant cases from the case base.

It should be noted that by *not* restricting the search to those cases consistent with the given query, we allowed the assignments \mathbf{u} to be violated, although it is easy to see that in the case retrieval test setup described above, the hard constraint approach would always have given better results. However, as discussed earlier, our goal was not to produce a metric for the hard constrained case, and so this type of metric was not used in this set of experiments. A more natural experimental testing of the soft constraint approach is currently in progress

with one of our industrial partners by using real-world data and problem domain expert evaluations.

6 Conclusion

We have argued that Bayesian probability theory can be used as a formalization for the intuitively appealing case-based reasoning paradigm. In this paper we focused on the case matching problem, and proposed a probabilistic scoring metric for this task. We showed how the Bayesian scoring metric can be computed efficiently when a strong independence assumption between the domain variables is made. In the experimental part of the paper, we proposed a case retrieval test setup for measuring the goodness of different case matching scores, and evaluated empirically the Bayesian case matching score by using publicly available real-world case bases. The empirical results showed that when encountered with cases where some of the feature values have been removed, a relatively small number of remaining values is sufficient for retrieving the original case from the case base by using the proposed measure. In this task, the Bayesian case matching score produced much better results than the standard Hamming distance similarity score.

Acknowledgments. This research has been supported by the Technology Development Center (TEKES), and the Academy of Finland.

References

1. D. Aha. *A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Observations*. PhD thesis, University of California, Irvine, 1990.
2. D. Aha, editor. *Lazy Learning*. Kluwer Academic Publishers, Dordrecht, 1997. Reprinted from *Artificial Intelligence Review*, 11:1–5.
3. C. Atkeson. Memory based approaches to approximating continuous functions. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting. Proceedings Volume XII in the Santa Fe Institute Studies in the Sciences of Complexity*. Addison Wesley, New York, NY, 1992.
4. C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. In Aha [2], pages 11–73.
5. J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.
6. G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
7. M.H. DeGroot. *Optimal statistical decisions*. McGraw-Hill, 1970.
8. B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman and Hall, London, 1981.
9. D. Fisher. Noise-tolerant conceptual clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 825–830, Detroit, Michigan, 1989.

10. D. Fisher and D. Talbert. Inference using probabilistic concept trees. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 191–202, Ft. Lauderdale, Florida, January 1997.
11. J.H. Friedman. Flexible metric nearest neighbor classification. Unpublished manuscript. Available by anonymous ftp from Stanford Research Institute (Menlo Park, CA) at playfair.stanford.edu, 1994.
12. 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.
13. S. Kasif, S. Salzberg, D. Waltz, J. Rachlin, and D. Aha. Towards a better understanding of memory-based reasoning systems. In *Proceedings of the Eleventh International Machine Learning Conference*, pages 242–250, New Brunswick, NJ, 1994. Morgan Kaufmann Publishers.
14. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, 1993.
15. P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. A Bayesian approach for retrieving relevant cases. In P. Smith, editor, *Artificial Intelligence Applications (Proceedings of the EXPERSYS-97 Conference)*, pages 67–72, Sunderland, UK, October 1997. IITT International.
16. P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Bayes optimal instance-based learning. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98, Proceedings of the 10th European Conference*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 77–88. Springer-Verlag, 1998.
17. P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. Bayesian and information-theoretic priors for Bayesian network parameters. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98, Proceedings of the 10th European Conference*, Lecture Notes in Artificial Intelligence, Vol. 1398, pages 89–94. Springer-Verlag, 1998.
18. D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, London, 1994.
19. A. Moore. Acquisition of dynamic control knowledge for a robotic manipulator. In *Seventh International Machine Learning Workshop*. Morgan Kaufmann, 1990.
20. 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.
21. C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.
22. M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B)*, 36:111–147, 1974.
23. H. Tirri, P. Kontkanen, and P. Myllymäki. A Bayesian framework for case-based reasoning. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 413–427. Springer-Verlag, Berlin Heidelberg, November 1996.
24. 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.
25. D.M. Titterton, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, New York, 1985.