Maintaining Case-Based Reasoning Systems: A Machine Learning Approach

Niloofar Arshadi[1] and Igor Jurisica[1,2]

[1] Department of Computer Science, University of Toronto,
10 King's College Road, Toronto, Ontario M5S 3G4, Canada
niloofar@cs.toronto.edu
[2] Ontario Cancer Institute, Princess Margaret Hospital,
University Health Network, Division of Cancer Informatics
610 University Avenue, Toronto, Ontario M5G 2M9, Canada
ij@uhnres.utoronto.ca

Abstract. Over the years, many successful applications of case-based reasoning (CBR) systems have been developed in different areas. The performance of CBR systems depends on several factors, including case representation, similarity measure, and adaptation. Achieving good performance requires careful design, implementation, and continuous optimization of these factors. In this paper, we propose a maintenance technique that integrates an ensemble of CBR classifiers with spectral clustering and logistic regression to improve the classification accuracy of CBR classifiers on (ultra) high-dimensional biological data sets.
Our proposed method is applicable to any CBR system; however, in this paper, we demonstrate the improvement achieved by applying the method to a computational framework of a CBR system called *TA3*. We have evaluated the system on two publicly available microarray data sets that cover leukemia and lung cancer samples. Our maintenance method improves the classification accuracy of *TA3* by approximately 20% from 65% to 79% for the leukemia and from 60% to 70% for the lung cancer data set.

1 Introduction

Case-based reasoning (CBR) has been successfully applied to a wide range of applications, such as classification, diagnosis, planning, configuration, and decision-support [1]. CBR can produce good quality solutions in weak-theory domains, such as molecular biology, where the number and the complexity of the rules affecting the problem are very high, there is not enough knowledge for formal representation, and the domain understanding evolves over time [2]. In addition, similarly as other learning systems, CBR systems can suffer from the *utility problem*, which occurs when knowledge learned in an attempt to improve a system's performance degrades it instead [3].

These issues can be addressed by continuous *case-based reasoner maintenance* (CBRM) [4, 5], where the contents of one or more knowledge containers

are revised in order to improve future reasoning for a particular set of performance objectives [6]. According to Richter's definition, there are four containers in which the knowledge could be stored in a CBR system: the vocabulary used, the similarity measure, the solution transformation, and the case-base [7]. During maintenance, the contents of each of the four knowledge containers may be revised in order to improve the performance objectives, e.g., improving the quality of the proposed solution.

Although several methods have been proposed for revising the case-base to reduce the number of stored cases [8, 9, 10], relatively little work has been carried out on revising the case-base to reduce the number of attributes [1] of stored cases. The problem, known as the "curse of dimensionality", occurs in (ultra) high-dimensional domains with tens of thousands of attributes and only a few hundred cases (samples). Such domains include microarray data sets, which measure the activity of tens of thousands of genes simultaneously. Microarrays are used in medical domains to produce molecular profiles of diseased and normal tissues, and thus increase the level of detail that can be stored about every patient. That is useful for understanding various diseases, and the resulting patient profiles support more accurate analogy-based diagnosis, prognosis, and treatment planning. Microarray data sets are represented by an $N \times M$ matrix, where M is the number of genes for the N samples, and they are labeled using clinical profiles (or phenotypes).

Clustering and *feature selection* techniques have been applied to many domains including microarrays [11, 12, 13]. Clustering groups samples (cases) into partitions, such that samples within a cluster are similar to one another and dissimilar to samples in other clusters. Clustering techniques can be categorized into *partitional* and *hierarchical* methods [14]. Partitional-based clustering techniques attempt to break a data set into k clusters, such that each cluster optimizes a given criterion, e.g., minimizes the sum of squared distance from the mean within each cluster. Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones (agglomerative approach), or by splitting larger clusters (divisive approach).

The goal of feature selection is to identify "informative" features among thousands of available features, i.e., relevant features that improve CBR performance for a given reasoning task. In microarray data sets, "informative" features comprise genes with expression patterns that have meaningful biological relationships to the classification labels of samples (analogously, it could represent sample vectors that have meaningful biological relationship to the classification labels of genes). For microarray data sets, mining a subset of genes that distinguishes between cancer and normal samples can play an important role in disease pathology and drug discovery. Removing "non-informative" features helps overcome the "curse of dimensionality" and improves the prediction accuracy of classifiers.

[1] In this paper, we use attributes and features interchangeably, unless otherwise specified.

Feature selection techniques are classified into *filter* and *wrapper* methods [15]. The filter approach selects feature subsets that are independent of the induction algorithm, while the wrapper approach evaluates the subset of features using the inducer itself.

Our main challenge is to interpret the molecular biology data to find similar samples to eventually use them in case-based medicine, and to identify those genes whose expression patterns have meaningful relationships to their classification labels. Clustering and feature selection techniques have been successfully applied to CBR maintenance [16, 10]; however, in this paper, we show how those techniques can further improve the prediction accuracy of a CBR classifier when combined with mixture of experts to analyze microarray data sets.

Our CBR maintenance approach has three main components: ensemble of CBR systems, clustering, and feature selection. We use an ensemble of CBR systems, called *mixture of experts* (MOE) to predict the classification label of a given (input) case. A gating network calculates the weighted average of votes provided by each expert. The performance of each CBR expert is further improved by using clustering and feature selection techniques. We apply spectral clustering [17] to cluster the data set into k groups, and the logistic regression model [18] is used to select a subset of features in each cluster. Each cluster is considered as a case-base for the k CBR experts, and the gating network learns how to combine the responses provided by each expert.

Although the proposed method is applicable to any CBR system, we demonstrate the improvement achieved by applying it to a specific implementation of a CBR system, called *TA3* [19]. *TA3* is a computational framework for CBR based on a modified nearest-neighbor technique and employs a variable context, a similarity-based retrieval algorithm, and a flexible representation language.

The rest of the paper is organized as follows. Section 2 reviews case-based reasoner maintenance techniques. In Section 3, we present the MOE4CBR method that uses a mixture of experts of CBR to classify high-dimensional data sets. Also, we discuss the proposed maintenance method in terms of Leake and Wilson's case-base maintenance framework [20]. Section 4 introduces the *TA3* CBR system, which is used as a framework for evaluating MOE4CBR. In Section 5, we demonstrate experimental results of the proposed method on two publicly microarray data sets.

2 Related Work

In this section, we explain the algorithms employed to maintain the contents of the four knowledge containers – vocabulary used, the similarity measure, the solution transformation, and the case-based knowledge container – introduced by Richter [7].

Case-base maintenance (CBM) policies differ in the approach they use and the schedule they follow to perform maintenance. Leake and Wilson categorize maintenance policies in terms of how they gather data relevant to maintenance,

how they decide when to trigger maintenance, the types of maintenance operations available, and how selected maintenance operations are executed [20].

Smyth and McKenna propose a method, that edits the case-base, such that the range of the problems that can be solved remains unchanged while the size of the case-base is minimized [8]. Their method is based on *condensed nearest neighbor* (CNN). This method builds up an edited set of training examples by incrementally adding examples to the set if they cannot be correctly classified by the current edited test [21].

In the CBM method proposed by Shiu and Yeung, a large case-base is transformed to a small case-base together with a group of adaptation rules that are generated by fuzzy decision trees [9]. These adaptation rules play the role of complementing the reduction of cases. Yang and Wu propose a method that does not remove cases from the case-base as they may be useful in the long run. Instead, the method reduces the size of case-base by creating small case-bases that are located on different sites [10].

DRAMA is an example of an interactive CBR system for *vocabulary maintenance* [22]. The cases in the system are conceptual aircraft designs, and the designers have freedom in defining new features to describe design cases. Each time before a new case is added to the case-base, the system examines the vocabulary container and suggests appropriate features that have been used previously. In this way, the vocabulary is built in parallel with the case-base.

Learning feature weights can be considered as an example of *similarity maintenance*. The system asks the user(s) to adjust feature weights for a set of cases, and applies the weights during case retrieval. Zhang and Yang propose a method for continually updating a feature-weighting scheme based on interactive user responses to the system's behavior [23].

Aha and Bankert discuss how using filter and wrapper techniques improve the classification accuracy of their case-based classifier on the cloud data set with 204 features and a few thousands data points [16]. Their results show that a wrapper FS method (called BEAM) applied to a nearest neighbor classifier IB1 improves its classification accuracy from 73% to 88%.

A system for disaster response planning called DIAL is an example of *solution transformation maintenance* [24]. The solution transformation container in DIAL comprises a set of adaptation cases and rules, which can be adjusted over time. In order to adapt a solution, the system checks for applicable adaptation case(s); if no adaptation cases apply, new adaptation cases can be learned by recording traces of rule-based or interactive manual adaptation.

Unlike speed-up learners that have first-principle problem solvers in addition to control rules, pure CBR systems do not usually have first-principle rules. Thus, without cases similar to a problem at hand, they cannot solve new problems. Therefore, in maintaining CBR systems, competence criteria (i.e., the range of target problems that a given CBR system can solve) should be considered, as well as efficiency criteria, which is the main focus of maintaining speed-up learners. However, based on which criterion had a higher optimization priority during

maintenance, different CBR maintenance methods can be categorized into two groups [5]:

1. *Competence-directed* CBM methods attempt to maintain the case-base to provide the same (or better) quality solution to a broader range of problems
2. *Efficiency-directed* methods consider the processing constraints, and modify knowledge containers to improve efficiency of storage or scalability of retrieval.

3 The MOE4CBR Method

The goal of our maintenance method is to improve the prediction accuracy of CBR classifiers, and at the same time reduce the size of the case-base knowledge container. According to Smyth's categorization [5], our maintenance method – Mixture Of Experts for CBR systems (MOE4CBR) – is both competence-directed, since the range of the problems the system can solve increases and efficiency-directed, since the size of case-base decreases.

The performance of each expert in MOE4CBR is improved by using clustering and feature selection techniques. Based on our initial analysis [25], we selected spectral clustering [17] for clustering the case-base, and the logistic regression model [18] as a filter feature selection for the *TA3* classifier. Given a labeled training data set, the system predicts labels for the unseen data (test set) following the process described below.

3.1 Clustering

Of the many clustering approaches that have been proposed, only some algorithms are suitable for domains with large number of features and a small number of samples. The two clustering approaches widely used in microarray data analysis [26, 27] are k-means clustering [14] and self-organizing maps (SOMs) [28]. Our earlier evaluation suggests that spectral clustering [17] outperforms k-means clustering and SOMs [25]. The comparison was based on two criteria:

1. **Dunn's index** [29], which does not require class labels and identifies how "compact and well separated" clusters are. It is defined as follows:

$$D = \min_{i=1,\ldots,k} \{ \min_{j=i+1,\ldots,k} (\frac{d(c_i, c_j)}{\max_{l=1,\ldots,k} diam(c_l)}) \},$$

where k denotes the number of clusters and $d(c_i, c_j)$ is the dissimilarity function between two clusters c_i and c_j defined as:

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y)$$

The diameter of a cluster c, represented by $diam(c)$, is considered as a measure of dispersion and is defined as follows:

$$diam(c) = \max_{x,y \in c} d(x, y)$$

2. **Precision** and **recall** [30] that compare the resulting clusters with pre-specified class labels [25]. Precision shows how many data points are classified (clustered) correctly, and recall shows how many data points the model accounts. They are defined as follows:

$$P_i = \frac{\mid c_i \cap g_i \mid}{\mid c_i \mid} \quad and \quad R_i = \frac{\mid c_i \cap g_i \mid}{\mid g_i \mid},$$

where c_i is the cluster output by a clustering algorithm for the i^{th} data point, and g_i is the pre-specified classification label of that data point, and $1 \leq i \leq T$, where T is the number of data points. Precision and recall of clustering is defined as the weighted average of the precision and recall of each cluster. More precisely:

$$P = \sum_{i=1}^{k} \frac{\mid g_i \mid}{T} P_i \quad and \quad R = \sum_{i=1}^{k} \frac{\mid g_i \mid}{T} R_i$$

The classification error, E, is defined as:

$$E = \sum_{i=1}^{k} \mid g_i \cap \overline{f(g_i)} \mid,$$

where $\mid g_i \cap \overline{f(g_i)} \mid$ denotes the number of data points in class g_i which labeled wrong, k shows the number of clusters, and $f(g)$ is a one to one mapping from classes to clusters, such that each class g_i is mapped to the cluster $f(g_i)$.

Considering the results of our comparison, we apply spectral clustering, which has been successfully used in many applications including computer vision and VLSI [17]. In this approach, data points are mapped to a higher dimensional space prior to being clustered. More precisely, the k eigenvectors associated with the k largest eigenvalues of matrix X are clustered by k-means, where k represents the number of clusters and is set by the user. Matrix X is a transformation of the *affinity* matrix – the matrix holding the Euclidean distance between any two data points. In the next step, data point s_i is assigned to cluster j if and only if row i of the matrix X was assigned to cluster j, where $1 \leq i \leq N$, $1 \leq j \leq k$, and N is the number of data points.

3.2 Feature Selection

The goal of feature selection is to improve the quality of data by removing redundant and irrelevant features, i.e., those features whose values do not have meaningful relationships to their labels, and whose removal improves the prediction accuracy of the classifier. Feature selection techniques are classified into *filter* and *wrapper* methods [31]. The main difference is that the latter use the final classifier to evaluate the subset of features, while the former do not.

Fisher criterion and standard t-test are two statistical methods that have been successfully applied to feature selection problem in (ultra) high-dimensional data sets [32]. In order to select a suitable feature selection approach for CBM, we have evaluated performance of Fisher criterion, t-test, and the logistic regression model [18] when used in a CBR classifier [25]. Namely, we have applied the three feature selection techniques to the *TA3* classifier, and measured the improvement in *accuracy* and *classification error*. Accuracy measures the number of correctly classified data points, and classification error counts the number of misclassified data points. Based on our evaluation, logistic regression applied to feature selection outperforms Fisher and standard t-test techniques [25].

Assuming that classifier x is the logistic of a linear function of the feature vector, for two classes, the logistic regression model has the following form:

$$Pr(y = 0|x, w) = \frac{1}{1 + e^{-w^T x}}, \tag{1}$$

where w is a $p + 1$ column vector of weights, and p is the number of features [18]. Logistic regression has been successfully applied to classifying (ultra) high-dimensional microarrays [33]. However, we use logistic regression as a filter feature selection method. In order to select a subset of features (genes), the logistic regression classifier is trained on the training set using the above formula, and features corresponding to the highest ranking magnitude of weights are selected. The data sets are normalized, such that all features (regressor variables) have the same mean and the same variance. Since there are thousands of features in the microarray data sets, features are eliminated in chunks; however, better results might be obtained by removing one feature at a time, and training the classifier on the remaining features.

3.3 Mixture of Experts

The mixture of experts approach is based on the idea that each expert classifies samples separately, and individual responses are combined by the gating network to provide a final classification label [18]. A general idea of the mixture of experts approach is depicted in Fig. 1. In order to combine the responses of k experts, the following formulas are used [18]:

$$Pr(y = Y|x_i) = \sum_{j=1}^{k} Pr(C_j|x_i) \times Pr(y = Y|C_j, x_i), \tag{2}$$

with the constraint that:

$$\sum_{j=1}^{N} Pr(C_j|x_i) = 1, \tag{3}$$

where x_i represents the unseen data (test data), $\{C_1, ..., C_k\}$ denote the clusters, and Y is the class label. $Pr(C_j|x_i)$ is calculated as follows. Given a test data

x_i, the l similar cases are retrieved from the case-base, where l can be chosen by the user. Then $Pr(C_j|x_i)$ is calculated by dividing the number of retrieved cases belonging to C_j (represented by S) by the total number of the retrieved cases (which is l). $Pr(y = Y|C_j, x_i)$ is the number of retrieved cases with class label Y belonging to C_j divided by S.

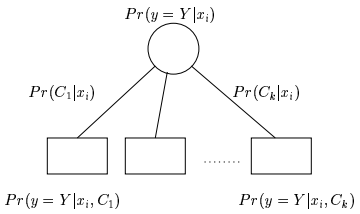

Fig. 1. Mixture of Experts: terminal nodes are experts, and the non-terminal node is the gating network. The gating network returns the probability that the input case x_i belongs to class Y.

As Fig. 2 depicts, the MOE4CBR maintenance method has two main steps: first, the case-base of each expert is formed by clustering the data set into k groups, then each case-base is maintained "locally" using feature selection techniques. Each of the k obtained sets will be considered as a case-base for our k CBR experts. We use formulas 2 and 3 to combine the responses of the k experts. Each expert applies the *TA3* model to decide on the class label, and the gating network uses *TA3* to assign weights to each classifier, i.e., to determine which class the input case most likely belongs to.

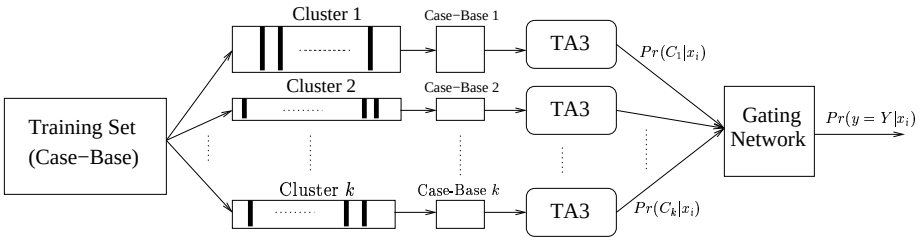

Fig. 2. Mixture of Experts for Case-Based Reasoning: training set is grouped into k clusters, and after selecting a subset of features for each group (shown with vertical bars), each group will be used as a case-base for the k experts of CBR. The gating network combines the responses provided by each *TA3* expert considering the weights of each expert (weights are shown on the arrows that connect *TA3* experts to the gating network).

Next, we describe the proposed maintenance method in terms of the maintenance framework introduced by Leake and Wilson [20]:

- **Type of data**: *none.* The method does not collect any particular data to decide when CBM is needed.
- **Timing**: *ad hoc, conditional.* The removal of "non-informative" attributes from case representation is performed at irregular intervals. To make the system more autonomous, we plan to make it conditional, i.e., whenever there is a change in the task, or whenever new cases are added to the case-base.
- **Integration**: *offline.* The maintenance is performed offline during a pause in reasoning.
- **Triggering**: *conditional.* Maintenance will be triggered whenever there is a change in the task, or new cases are added to the case-base.
- **Revision level**: *knowledge level.* Maintenance is mainly focused on removing "non-informative" features.
- **Execution**: *none.* The system makes no changes when the maintenance is needed, the maintenance method is invoked manually.
- **Scope of maintenance**: *broad.* The whole case-base is affected by the CBM operations.

It should be noted that unlike in other CBR application domains, where cases are usually added to the case-base one by one, cases are added in a batch mode in molecular biology domains. Thus, the maintenance can be performed offline. As a result, the complexity issues of the maintenance method are less important.

4 An Introduction to the TA3 Case-Based Reasoning System

We used the *TA3* CBR system as a framework to evaluate our method, although our maintenance method can be applied to any CBR system. The *TA3* system has been applied successfully to biology domains, such as *in vitro* fertilization (IVF) [34] and protein crystal growth [35]. We briefly describe the main features of the *TA3* CBR system.

4.1 Case Representation in *TA3*

A case C corresponds to a real world situation, represented as a finite set of attribute/value pairs [34]:

$$C = \{<a_0 : V_0>, <a_1 : V_1>, ..., <a_n : V_n>\}.$$

There are two types of cases: (1) an input case (target) that describes the problem and is represented as a case without a solution; and (2) a source case, which is a case stored in a case-base that contains both a problem description and a solution.

Using the information about the usefulness of individual attributes and information about their properties, attributes are grouped into two or more Telos-style categories [36]. This enhancement of case representations is used during the retrieval process to increase the accuracy of classification and flexibility of retrieval, and to improve system's performance. In classification tasks, each case has at least two categories: problem description and class. The problem description characterizes the problem and the class gives a solution to a given problem. Additional categories can be used to group attributes into separate equivalence partitions, and the system can treat each partition separately during case retrieval.

4.2 Case Retrieval in *TA3*

The retrieval component is based on a modified nearest-neighbor matching [37]. Its modification includes: (1) grouping attributes into categories of different priorities so that different preferences and constraints can be used for individual categories during query relaxation; (2) using an explicit context (i.e., set of attribute and attribute value constraints) during similarity assessment; (3) using an efficient query relaxation algorithm based on incremental context transformations [19].

Similarity in *TA3* is determined as a closeness of values for attributes defined in the *context*. Context can be seen as a view or an interpretation of a case, where only a subset of attributes are considered relevant. Formally, a context is defined as a finite set of attributes with associated constraints on their values:

$$\Omega = \{<a_0 : CV_0>, ..., <a_k : CV_k>\},$$

where a_i is an attribute name and the constraint CV_i specifies the set of "allowable" values for attribute a_i. By selecting only certain features for matching and imposing constraints on feature values, a context allows for controlling what can and what cannot be considered as a partial match: all (and only) cases that satisfy the specified constraints for the context are considered similar and are relevant with respect to the context. Machine learning and knowledge-mining techniques may be applied to determine an optimal context: selecting features which are most "relevant" for a given task and specifying characteristic values for them.

4.3 Case Adaptation in *TA3*

Considering the characteristics of microarray data sets, the current implementation uses only simple adaptation. Namely, for case-base classification, the average class label of the similar retrieved cases is considered as the class label for the input case.

5 Experimental Results

Here we demonstrate the results of applying the MOE4CBR method to the *TA3* classifier.

5.1 Data Sets

The experiments have been performed on the following microarray data sets:

1. **Leukemia**: The data set contains data of 72 leukemia patients, with 7,129 expression levels for each sample[1] [26]. 46 samples belong to type I Leukemia (called Acute Lymphoblastic Leukemia) and 25 samples belong to type II Leukemia (called Acute Myeloid Leukemia).
2. **Lung**: The data set taken from the *Ontario Cancer Institute* [2] contains 39 samples, with 18,117 expression levels for each sample. Samples are pre-classified into recurrence and non-recurrence. Missing values were imputed using KNNimputed software, which is based on the weighted k-nearest neighbor method [38].

5.2 MOE4CBR Results

Table 1 depicts the results of applying the MOE4CBR maintenance method to the leukemia and lung data sets. When there is a tie, the *TA3* classifier cannot decide on the label of data points; resulting cases are categorized as "undecided" in the table. As the table shows, before the maintenance method is applied, the classification accuracy of the $TA3_{Leukemia}$ [3] and $TA3_{Lung}$ is 65% and 60%, respectively. However, after our maintenance method selects a subset of 712 out of 7129 genes for leukemia and a subset of 1811 out of 18117 genes for the lung data sets, and combines *TA3* classifiers using mixture of experts, the accuracy improves to 79% and 70%, respectively. In our experiments, the number of clusters, k, was assigned to be the number of classification labels, i.e., k was set to be 2 for both the leukemia and lung data sets.

Table 1. Accuracy of *TA3* before and after maintenance

Leukemia Data Set			
Method	Accuracy	Error	Undecided
No Maintenance	65%	35%	0%
MOE4CBR	79%	21%	0%
Lung Data Set			
Method	Accuracy	Error	Undecided
No Maintenance	60%	30%	10%
MOE4CBR	70%	30%	0%

We used the training and test data set suggested by the data set provider for the leukemia data set (38 samples in the training and 34 samples in the test

[1] http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_menu.cgi

[2] http://www.cs.toronto.edu/juris/publications/data/CR02Data.txt

[3] $TA3_X$ denotes application of *TA3* into a domain X.

set). The leave-one-out cross-validation (LOOCV) method was used for the lung data set, and results are averaged over 20 trials.

The lung data set has also been analyzed by Jones et al. [39]. They developed a model-based clustering prior to using the SVM classifier. Their results show that the classification accuracy of SVM prior to applying the proposed model-based clustering is 72% using 10-fold cross-validation for evaluation. After the proposed method is applied, which selects 20 meta-genes, the classification accuracy drops to 67%. Considering that our method improves the classification accuracy of *TA3* on the lung data set from 60% to 70% after we reduce the size of the data-set by 90%, our results are encouraging.

6 Conclusions

Molecular biology domain is a natural application for CBR systems, since CBR systems can perform remarkably well on complex and poorly formalized domains. However, due to the large number of attributes in each case, CBR classifiers, similarly as other learning systems, suffer from the "curse of dimensionality". Maintaining CBR systems can improve the prediction accuracy of CBR classifiers by clustering similar cases, and removing "non-informative" features in each group.

In this paper, we introduced the *TA3* case-based reasoning system, a computational framework for CBR systems. We proposed the mixture of experts for case-based reasoning (MOE4CBR) method, where an ensemble of CBR systems is integrated with clustering and feature selection to improve the prediction accuracy of the *TA3* classifier. Spectral clustering groups samples, and each group is used as a case-base for each of the k experts of CBR. To improve the accuracy of each expert, logistic regression is applied to select a subset of features that can better predict class labels. We also demonstrated that our proposed method improves the prediction accuracy of the *TA3* case-based reasoning system on two public lung and leukemia microarray data sets.

Although we have used a specific implementation of a CBR system, our results are applicable in general. Generality of our solution is also not degraded by the application domains, since many other life sciences problem domains are characterized by (ultra) high-dimensionality and a low number of samples. Further investigation may take additional advantage of Telos-style categories in *TA3* for classification tasks, and perform more experiments on several other data sets. The system may also benefit from new clustering approaches, and new feature selection algorithms.

Acknowledgments

This work is supported by IBM CAS fellowship to NA, and the National Science and Engineering Research Council of Canada (NSERC Grant 203833-02) and IBM Faculty Partnership Award to IJ. The authors are grateful to Patrick Rogers, who implemented the current version of *TA3*.

References

[1] Lenz, M., Bartsch-Sporl, B., Burkanrd, H., Wess, S., eds.: Case-based reasoning: experiences, lessons, and future directions. Springer (1998)

[2] Jurisica, I., Glasgow, J.: Application of case-based reasoning in molecular biology. Artificial Intelligence Magazine, Special Issue on Bioinformatics **25(1)** (2004) 85–95

[3] Francis, A.G., Ram, A.: The utility problem in case-based reasoning. In: Proceedings of the 1993 AAAI Workshop on Case-Based Reasoning, Washington, DC (1993)

[4] Leake, D.B., Wilson, D.C.: Remembering why to remember: performance-guided case-base maintenance. In Blanzieri, E., Portinale, L., eds.: Advances in Case-Based Reasoning, Fivth European Workshop on Case-Based Reasoning, Trento, Italy, Springer (2000) 161–172

[5] Smyth, B.: Case base maintenance. In Pobil, A.D., Mira, J., Ali, M., eds.: Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Volume 2., Castellon, Spain, Springer (1998) 507–516

[6] Wilson, D., Leake, D.: Maintaining case-based reasoners: Dimensions and directions. Computational Intelligence **17** (2001) 196–212

[7] Richter, M.M.: 1. In: Case-based reasoning: experiences, lessons, and future directions. Springer (1998) 1–15

[8] Smyth, B., McKenna, E.: Building compact competent case-bases. In Althoff, K.D., Bergmann, R., Branting, K., eds.: Proceedings of the 3rd International Conference on Case-Based Reasoning Research and Development (ICCBR-99), Seeon Monastery, Germany, Springer (1999) 329–342

[9] Shiu, S.C., Yeung, D.S.: Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance. Computational Intelligence **17** (2001) 295–314

[10] Yang, Q., Wu, J.: Keep it simple: a case-base maintenance policy based on clustering and information theory. In Hamilton, H., ed.: Advances in Artificial Intelligence, In Proceedings of the 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Montreal, Canada, Springer (2000) 102–114

[11] Xing, E.P.: Feature selection in microarray analysis. In Berrar, D., Dubitzky, W., Granzow, M., eds.: A Practical Approach to Microarray Data Analysis. Kluwer Academic publishers (2003) 110–131

[12] Quackenbush, J.: Computational analysis of microarray data. Nat Rev Genet **2** (2001) 418–427

[13] Molla, M., Waddell, M., Page, D., Shavlik, J.: Using machine learning to design and interpret gene-expression microarrays. AI Magazine **25** (2004) 23–44

[14] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kauffmann Publishers (2000)

[15] John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Machine Learning: Proceedings of the Eleventh International Conference, Morgan Kauffmann (1994) 121–129

[16] Aha, D.W., Bankert, R.: Feature selection for case-based classification of cloud types: an empirical comparison. In Aha, D.W., ed.: Proceedings of the AAAI-94 Workshop on Case-Based Reasoning, Menlo Park, CA: AAAI Press (1994) 106–112

[17] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In G. Dieterich, S. Becker, Z.G., ed.: Advances in Neural Information Processing Systems 14, MIT Press (2002)

[18] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Springer (2001)

[19] Jurisica, I., Glasgow, J., Mylopoulos, J.: Incremental iterative retrieval and browsing for efficient conversational CBR systems. International Journal of Applied Intelligence **12(3)** (2000) 251–268

[20] Leake, D., Wilson, D.: Categorizing case-base maintenance: dimensions and directions. In Smyth, B., Cunningham, P., eds.: Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning (EWCBR-98), Dublin, Ireland, Springer (1998) 196–207

[21] Hart, P.: The condensed nearest neighbor rule. IEEE on Information Theory **14** (1968) 515–516

[22] Leake, D.B., Wilson, D.C.: Combining CBR with interactive knowledge acquisition, manipulation and reuse. In Althoff, K.D., Bergmann, R., Branting, K., eds.: Proceedings of the 3rd International Conference on Case-Based Reasoning Research and Development (ICCBR-99), Seeon Monastery, Germany, Springer (1999) 203–217

[23] Zhang, Z., Yang, Q.: Dynamic refinement of feature weights using quantitative introspective learning. In: Proceedings of the fifteenth International Joint Conference on Artificial Intelligence (IJCAI 99), Quebec, Canada, Morgan Kauffmann (1999) 228–233

[24] Leake, D.B., Kinley, A., Wilson, D.C.: Acquiring case adaptation knowledge: a hybrid approach. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, AAAI Press, Menlo Park (1996) 648–689

[25] Arshadi, N., Jurisica, I.: Maintaining case-based reasoning in high-dimensional domains using mixture of experts. Technical Report CSRG-490, University of Toronto, Department of Computer Science (2004)

[26] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gassenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science **286** (1999) 531–537

[27] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Dmitrovsky, E., Lander, E., Golub, T.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. In: Proceedings of the National Academy of Science of the United States of America. Volume 96(6). (1999) 2907–2912

[28] Kohonen, T.: Self-Organizing Maps. Springer (1995)

[29] Dunn, J.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics **4** (1974) 95–104

[30] Baeza-Yates, R., Ribiero-Neto, B.: Modern information retrieval. Addison-Wesley (1999)

[31] Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324

[32] Jaeger, J., Sengupta, B., Ruzzo, W.: Improved gene selection for classification of microarrays. In: Pacific Symposium on Biocomputing. (2003) 8:53–64

[33] Xing, E.P., Jordan, M.L., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In Brodley, C.E., Danyluk, A.P., eds.: Proceedings of the Eighteenth International Conference on Machine Learning, Williamstown, MA, USA, Morgan Kauffmann (2001) 601–608

[34] Jurisica, I., Mylopoulos, J., Glasgow, J., Shapiro, H., Casper, R.F.: Case-based reasoning in IVF: prediction and knowledge mining. Artificial Intelligence in Medicine **12** (1998) 1–24

[35] Jurisica, I., Rogers, P., Glasgow, J., Fortier, S., Luft, J., Wolfley, J., Bianca, M., Weeks, D., DeTitta, G.: Intelligent decision support for protein crystal growth. IBM Systems Journal **40(2)** (2001) 394–409

[36] Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. ACM Transactions on Information Systems **8(4)** (1990) 325–362

[37] Wettschereck, D., Dietterich, T.: An experimental comparison of the nearest neighbor and nearest hyperrectangle algorithms. Machine Learning **19(1)** (1995) 5–27

[38] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics **17** (2001) 520–525

[39] Jones, L., Ng, S.K., Ambroise, C., McLachlan, G.: Use of microarray data via model-based classification in the study and prediction of survival from lung cancer. In Johnson, K., Lin, S., eds.: Critical Assessment of Microarray Data Analysis. (2003) 38–42