Competence Driven Case-Base Mining

Rong Pan [(1)] and **Qiang Yang** [(1)] and **Jeffrey Junfeng Pan** [(1)] and **Lei Li** [(2)]
[(1)] Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon Hong Kong, China
[(2)] Software Institute, Zhongshan University, Guangzhou, China
({panrong,qyang,panjf}@cs.ust.hk, lncsri07@cs.zsu.edu.cn)

Abstract

We present a novel algorithm for extracting a high-quality case base from raw data while preserving and sometimes improving the competence of case-based reasoning. We extend the framework of Smyth and Keane's case-deletion policy with two additional features. First, we build a case base using a statistical distribution that is mined from the input data so that the case-base competence can be preserved or even increased for future problems. Second, we introduce a nonlinear transformation of the data set so that the case-base sizes can be further reduced while ensuring that the competence be preserved and even increased. We show that Smyth and Keane's deletion-based algorithm is sensitive to noisy cases, and that our solution solves this problem more satisfactorily. We show the theoretical foundation and empirical evaluation on several data sets.

Introduction

Case-based reasoning (CBR) is a problem-solving strategy that uses previous cases to solve new problems (Kolodner 1993; Leake, Kinley, & Wilson 1997; Watson 1997). Over the years, CBR has enjoyed tremendous success in solving problems related to knowledge reuse. A major issue in CBR is how to automatically obtain quality case bases from input data. A quality case base must have high competence in the sense that most future problems can be matched to a case which has a solution that can be adapted to solve it. In addition, the discovered case bases must be compact; it should not contain too much redundant knowledge.

One approach to mining cases from a raw database is to adapt Smyth and Keane's seminal work on competence-preserving case deletion (Smyth & Keane 1995). In this work, the authors defined case-base competence via how cases in an existing case base are related to each other. A case-deletion policy is applied after all cases are classified into several types. The case deletion policy aims at removing as much redundancy from a case base as possible by reducing the overlap between cases or groups of cases. This work inspired several subsequent works in case-base maintenance and case mining. For example, researchers have considered case-base maintenance and case mining from different angles in (Leake *et al.* 2001). The work of (Leake, Kinley, & Wilson 1995) analyzed case-base maintenance problems and cast them into a general framework of revising the content and structure of an existing case base.

Despite significant progress in case-mining research, serious problems still remain. Several previous approaches for case base mining and maintenance can be sensitive to the presence of noise and outliers. Noisy cases are those whose descriptions are stochastic in nature and if retained in a case base, may cause the solutions to be wrong. Outlier cases occur rarely in future queries, but are still considered as pivotal cases in (Smyth & Keane 1995) and are kept in the final case bases. Furthermore, case-deletion and addition policies (Zhu & Yang 1999) tend to retain cases with large coverage. However, in many practical applications, many cases may have approximately uniform coverage; thus, it is difficult to distinguish between these cases by coverage. Many of the cases have very small coverage as well.

In this paper, we present a case-mining algorithm known as kernel-based greedy case-mining algorithm (KGCM) that solves the above problems. To solve the problem of noisy and outlier cases, we apply a Fisher Discriminant Analysis (FDA) in order to find the statistically significant pivotal cases based on the problem distribution so that the effect of noisy and outlier cases can be reduced. To solve the problem of small and uniform coverage of the cases, we apply a kernel-based transformation on the input data to project the input space into a nonlinear feature space. This transformation makes it possible to enlarge the case coverage. Our experiments confirm that the KGCM algorithm outperforms the previous coverage-based algorithms in preserving case-base competence in many application domains when dealing with classification problem.

Related Work on Case Mining

Related Work on Case-Base Maintenance

Case mining is closely related to case-base maintenance. In case-base maintenance, case-base structure, index and knowledge representation are updated to improve problem-solving performance. (Aamodt, Sandtorv, & Winnem 1998) presented learning algorithms for similarity functions used in case retrieval. In our formulation, case mining is targeted for large, un-indexed raw databases rather than an existing case base. Indexing problems have been an issue of intensive study (Bonzano, Cunningham, & Smyth 1997). Bonzano et al. assume that there is already a case base in existence, whereas in this paper we explore how to find the case base from historical databases. Zhu and Yang in (Zhu & Yang 1999) explored how addition-based policies can be applied to case-based maintenance. (Patterson *et al.* 2002) discussed how to apply the K-means algorithm for case-base maintenance, but it is an unsupervised framework, whereas in our situation, we have a supervised problem.

Some recent works have proposed to use kernel transformations for case-based reasoning. For example, Corchado and Fyfe et al. did some works on case-base construction with unsupervised kernel method (Fyfe & Corchado 2001). (Pan, Yang, & Li 2004) proposed to use nonlinear similarity measures for case retrieval. However, this work did not address the problem of case mining.

One technique we used is to apply a kernel based method. In the machine learning literature, much research has been done on feature space transformation with kernel functions. Some examples are Kernel Principle Component Analysis (Kernel PCA), and Kernel Independent Component Analysis (Kernel ICA) (Schölkopf, Smola, & Müller 1998; Bach & Jordan 2002). However, in case-based reasoning, it is important to relate the input and target variables. Thus, these previous works that rely on a unsupervised framework do not address this issue directly. In order to use this relationship, we apply Kernel Fisher Discriminant Analysis (KFDA) (Mika *et al.* 1999; Roth & Steinhage 1999) to a case base by taking into account both the similarity measures and the distribution of cases in the kernel space.

Case-Deletion Based Case Mining Algorithms

We first review the case-deletion policy of Smyth and Keane. We define a case as a problem-solution pair. That is, each element C of a case base is a pair $C = (x, s)$, where $s \in S$ is a corresponding solution to a problem description x.

When the size of a case base gets large, there is a need to select and keep a subset of the cases. To address this problem, Smyth and Keane (Smyth & Keane 1995) suggest a case-deletion-based approach. The premise of this approach is that each case in the case base should be classified according to its competence. These classifications are made according to two key concepts: *coverage* and *reachability*. Coverage refers to the set of problems that each case can solve. Reachability is the set of cases that can be used to provide solutions for a problem (see Definitions 1 and 2 (Smyth & Keane 1995)).

Definition 1 Coverage (Smyth and Keane)
Given a case base $C = \{c_1, ..., c_n\}, \forall c \in C$
$Coverage(c) = \{c' \in C : Adaptable(c, c')\}$
where Adaptable(c, c') means that c can be adapted from c' within a given cost limit.

Definition 2 Reachability (Smyth and Keane
Given a case base $C = \{c_1, ..., c_n\}, \forall c \in C$
$Reachability(c) = \{c' \in C : Adaptable(c', c)\}$

Cases that represent unique ways to answer a specific query are *pivotal* cases. *Auxiliary* cases are those which are completely subsumed by other cases in the base. In between these two extremes are *spanning* cases which link together areas covered by other cases, and *support* cases which exist in groups to support an idea. The deletion algorithm deletes cases in the order of their classifications : auxiliary, support, spanning and then pivotal cases (Smyth & Keane 1995). The definition for pivotal and auxiliary cases are as follows (see also Figure 1). A case c is a

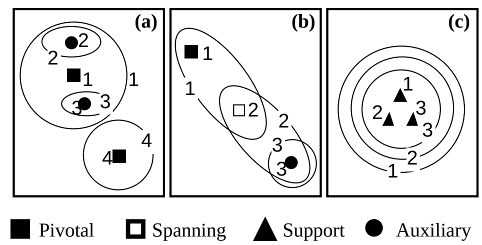


Figure 1: Case Categories (Adopted from (Smyth & Keane 1995)).

Pivotal Case if $Reachable(c) - \{c\}\emptyset$. It is an **Auxiliary Case** if $\exists c' \in Reachable(c) - \{c\} : Coverage(c) \subsetneq Coverage(c')$. It is a **Spanning Case** if $\neg Pivotal(c) \wedge Coverage(c) \bigcap \bigcup_{c' \in Reachable(c) - \{c\}} Coverage(c) \neq \emptyset$. Finally, it is a **Support Case** if $\exists c' \in Reachable(c) - \{c\} : Coverage(c') \subset Coverage(c)$.

From these definitions, it is easy to see that pivotal problems are the most unique, spanning problems are less unique, but auxiliary problems are not unique. Smyth and Keane's footprint deletion (FD) and footprint-utility deletion (FUD) policy delete auxiliary problems first, then support problems, then spanning problems, and finally pivotal problems. This approach is better than a random deletion policy for preserving competence. The competence of a case base built by Smyth and Keane's footprint deletion (FD) or footprint-utility deletion (FUD) policy is not guaranteed to be preserved.

The case-deletion policy is motivated by the need to delete cases in order to maintain the competency of a case base at a reasonable size. However, no mention is made of why auxiliary cases should be deleted first, and how the quality of the resulting case base is ensured after the update is done. In some situations, when an auxiliary case is near the centroid of a local area where the solutions of most cases are the same, and a pivotal case may be noisy cases or outlier cases, the auxiliary case is a representative case rather than the pivotal cases. Therefore, deleting the other type of cases before the pivotal cases only offers an intuitive solution to

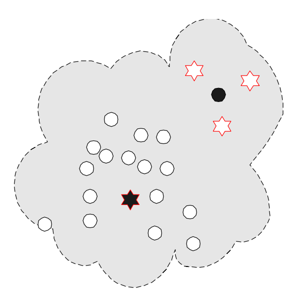


Figure 2: Illustrating noisy cases

the case-base maintenance problem; there is no guarantee of the level of true competence preservation. For example, Figure 2 shows the shortcoming of the case-deletion policy. In the figure, each circle or star is a case. Assume that each case is only adaptable to its neighbors. Then, according to the definition of pivotal cases, both dark circle and star are pivotal cases. If they were retained by the case-deletion policy, many future problems will be classified wrongly, because these two cases are also noise ones. In the next section, we show how to remove these noise cases from consideration.

Case Mining in Kernel Space

Overview of our Framework for Case Mining

We now consider how to mine cases from an original database. First, we consider an original database with l cases $D = \{(\overrightarrow{x_i}, s_i), i = 1, \ldots, l, \overrightarrow{x_i} \in \mathbb{R}^N, s_i \in solutionset\}$. Here $\overrightarrow{x_i}$ is the input of a case and the s_i is solution of the case.

Our input is a data set D consisting of problem and solution pairs. The problem part of a record consists of a set of attribute values. The solution can be categorized into a number of classes in the discrete solution case, or a numerical value in a continuous case. For the former, an example is finding a corresponding design to a set of requirements. For the latter, an example is to predict housing prices based on different factors. From D, we wish to construct a new case base C, such that future problems can be solved by consulting C based on a similarity measure.

Our case-mining framework functions in three phases. In the first phase, for a given input data set, we perform a kernel space transformation to enable the subsequent operation to be effective even for cases whose similarity is nonlinear; a nonlinear problem has the characteristic that similar problems using the input features may not lead to similar solutions. Then, in the second phase, in a transformed space, we perform a supervised maximum correlation analysis to find the best set of features (in kernel space) for distinguishing the solutions. After this, in phase three, we discover the most representative cases in each cluster via correlation analysis. These cases must be *globally diverse* in the sense that they represent the distribution of the cases in a corresponding category with a reduced set of features. The cases thus found are output into the final case base.

Once a case base is constructed, it must then be verified to be certain of its quality. Thus, an evaluation process follows in which the cases are used to solve new problems. In this manner, the learned case base serves as a problem solving model that is "trained" from the training data. The model must be cross validated on test data to be sure of its competence.

Similar to (Smyth & Keane 1995), we use competence to evaluate the quality of a generated case base. Consider solving classification problems using a new case base. Let $A_n(T)$ be the accuracy of our new case base when it is applied to a test problem set T, and $A_i(T)$ be that of the input database. Then we define the competence of the new case base as

$$Comp_n(T) = \frac{A_n(T)}{A_i(T)}$$

Note that in this definition, the competence $Comp_n(T)$ depends on the future problem set T. This definition captures the problem solving experience in future problems, rather than based on the current static view of the database, as is done using coverage only. In the sequel, we will rely on this definition of competence in our discussion.

Feature Extraction with Kernels

Case mining aims to extract a case base from an original database where cases can cover the future problem space. We also wish to maintain high competence for the mined case base. In the last subsection, we introduced our three-phase case-mining framework. In this section, we perform a nonlinear feature extraction to find candidate cases. In the next section, we discuss the case-base construction problem.

Fisher Discriminant Analysis (FDA) was first presented by Fisher (Fisher 1936). We wish to go beyond the linear analysis to cope with nonlinear data sets. Mika and Roth et al. (Mika *et al.* 1999; Roth & Steinhage 1999) introduced the Kernel Fisher Discriminant (KFD) Analysis. The key idea of KFD is to minimize the intra-class scatter around target classes in the feature space, while maximizing the inter-class scatter.

Assume that we are given a raw dataset $X = \{c_1, c_2, \ldots, c_l\}$ where c_i denotes a case which is a pair of attributes and solution $(\overrightarrow{x_i}, s_i)$ where $\overrightarrow{x_i} \in \mathbb{R}^N$. KFD first maps the attribute part of the cases into a feature space with a function ϕ. The resulting set is $\mathfrak{X} = \{\phi(\overrightarrow{x_1}), \phi(\overrightarrow{x_2}), \ldots, \phi(\overrightarrow{x_l})\}$. Suppose that $\mathfrak{X}_i\{\phi(\overrightarrow{x_1^i}), \phi(\overrightarrow{x_2^i}), \ldots, \phi(\overrightarrow{x_{l^i}^i})\}$ is a set of cases belonging to the same solution s_i in the feature space. The centroid $\overrightarrow{\mathbf{m_i}}$ of $\mathfrak{X}_i$ can be calculated as $\overrightarrow{\mathbf{m_i}} = l_i^{-1} \sum_{j=1}^{l_i} \phi(\overrightarrow{x_j^i})$. Then the total differences of the cases corresponding to the same solution s_i can be expressed as $S_w = \sum_{i \in solutionset} \sum_{j=1}^{l_i} (\phi(\overrightarrow{x_j^i}) - \overrightarrow{\mathbf{m_i}})(\phi(\overrightarrow{x_j^i}) - \overrightarrow{\mathbf{m_i}})^T$, and the sum of distances between any two centroids of different solutions can be expressed $S_B = \sum_{i,j \in solutionset} (\overrightarrow{\mathbf{m_i}} - \overrightarrow{\mathbf{m_j}})(\overrightarrow{\mathbf{m_i}} - \overrightarrow{\mathbf{m_j}})^T$. KFD attempts to find a new direction $\overrightarrow{\mathbf{w}}$ such that the projection of S_B can be maximized while the projection of S_W can be minimized. That is, given a vector $\overrightarrow{\mathbf{w}}$, we need to maximize the following formula

$$J(\overrightarrow{\mathbf{w}}) = \frac{\overrightarrow{\mathbf{w}}^T S_B \overrightarrow{\mathbf{w}}}{\overrightarrow{\mathbf{w}}^T S_W \overrightarrow{\mathbf{w}}}. \quad (1)$$

The computational details of KFD can found in (Mika *et al.* 1999; Roth & Steinhage 1999).

Case-Base Construction

With the eigenvectors found from solving Equation (1), we wish to then find a case base of size k. Suppose that we have found the most correlating direction $\overrightarrow{\mathbf{w}}$ extracted by KFD. Consider how to construct a new case base that has a smaller size than the original database D for efficient case retrieval. To achieve this, we need to find the most representable cases for the same solutions. We define a *global diversity* among the mined case base with respect to the original data base D as the spread of the cases among the data. We wish to maximize global diversity.

We define an evaluation function for cases that can integrate two factors: the importance of a case in terms of its projection onto an eigenvector $\overrightarrow{w}$, and its diversity score with respect to the cases that are extracted already. These functions are defined below. In the functions, I is a case base.

$$Eval(I) = weight(I) * globaldiv(I), \quad (2)$$

$$weight(I) = \prod_{c_i \in I} (w_i), \quad (3)$$

$$globaldiv(I) = \min_{c_i, c_j \in I} dissimilarity(c_i, c_j). \quad (4)$$

In this paper, $dissimilarity(.,.)$ is defined as the Euclidean distance of the two cases.

Based on these functions, we present a case-mining algorithm in Table 1.

Table 1: Kernel Greedy Case Mining (KGCM) Algorithm.

	KGCM(D,k,α**)** D is the input database, k is the number of cases to be mined, and α is used for selecting seed cases.
1	Run KFD on D to find the direction $\overrightarrow{w}$
2	**for** every $D_i = \{\overrightarrow{x_j^i} \mid \overrightarrow{x_j^i} \in D \wedge$ $\overrightarrow{x_j^i}$ `has the same solution` $s_i\}$ **do**
3	Compute the projection w_j^i of each data record $\overrightarrow{x_j^i}$ onto $\overrightarrow{w}$; sort D_i by w_j^i in descending order
4	$CB_i = \{\}$;M= First m data of D_i, where $m = \alpha * \lvert D_i \rvert$
5	Select the case c with *largest* w_j^i, $CB_i = CB_i + \{c\}$, $M = M - \{c\}$
6	**while** $M! = \phi$ **and** $\lvert CB_i \rvert = k * (\lvert D_i \rvert / \lvert D \rvert)$ **do** Select the case c with largest evaluation value: $Eval(CB_i \bigcup \{c\})$

7	$CB_i = CB_i + \{c\}$, $M = M - \{c\}$
	end while
	end for

8	$CB = \bigcup_i CB_i$
9	**return** CB

In Table 1, the *KGCM algorithm* first finds a case c_i with the largest projection onto a direction $\overrightarrow{w}$. This case is added to the case base. Then, we find the one whose diversity evaluation through the $Eval(.)$ function is the largest. We add this case to the case base. We continue until k cases are seleted. This greedy algorithm operates in the kernel space, and always looks for a next case that is highly representative of the population.

An Example

Consider an example to demonstrate our case-mining algorithm KGCM. The dataset is an artificially generated spiral dataset used to illustrate the effect of a nonlinear case base, where the data set is adapted from (Duda, Hart, & Stork 2000). This data consist of two categories: positive and negative. The positive data and negative data comprise 1000 instances. In this example, we wish to extract a subset of k data as a case base which can accurately classify new data into either a positive or a negative category. Therefore, for this example, the task for a CBR system is to classify the input data.

We perform the KGCM algorithm on this dataset (see Figure 3), where "oripos" means the original data that belongs

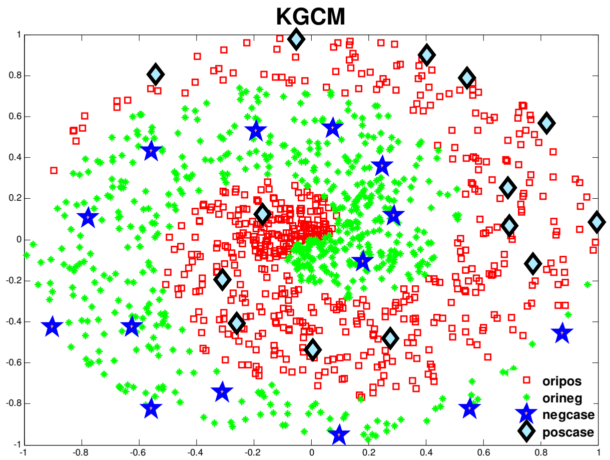


Figure 3: Spiral data with 14 cases found, per class, by the KGCM algorithm.

to the positive class, and "orineg" the data that belongs to the negative class. Similarly, "negcase" corresponds to the data points that are mined from the negative class categories, and likewise for positive cases. Here, we use the Gaussian kernel with the $\sigma^2 = 4$. The ratio $\alpha = 0.3$ is chosen, which means that 30% of the best classification data are used as candidates for the first round of Greedy selection. This example shows that when we mine cases from an original database, we need to consider both the sparsity and the representativeness of the cases. First, we can see that this problem space is nonlinear in nature, such that any naive method for selecting representative cases, such as a random selection method or simple selection of cases on either side of the chart, may fail to find a good case base. Second, although the problem is nonlinear, it can be transformed into a linear case base by means of a kernel transformation. In the feature space, it is then possible to perform a linear separation of the cases. Finally, when we select cases for this problem, we should avoid selecting cases that belong to a single region; instead, we should choose cases by maximizing global diversity. The resultant case base shown in the figure, which consists of the stars and diamonds, are the output of the KGCM algorithm when $k = 14$. As we can see, this case base is of high quality.

Experimental Analysis

We claim that KGCM benefits from superior effectiveness and performance in terms of competence. We test our algorithms on a number of case bases. Our experiments are performed on several publicly available datasets. They are:

wdbc, *optdigits*, *pendigits*, *shuttle*, *satellite* , *ionosphere*, *scale*, *sonar*, and *spiral* [1]. In all experiments we compare the competence of the case base against the percentage of the size of the case base to the original database; we call this percentage the "ratio of case base size" in the experiments.

For each application domain, we validate our KGCM with 1-NN as the underlying CBR scheme. In every experiment, we use 1-NN to classify the testing data with all the training data. Then we use three case-mining algorithms on the training data. Next, we validate the mined case base on the testing data. The three case-mining algorithms are: our KGCM, and LGCM and case deletion policy, where LGCM is to apply FDA without using kernels. LGCM is compared to because we wish to test the effectiveness of using FDA and kernels separately.

For the spiral dataset, we split the original 2000 dataset into 1200 training data and 800 test data. Figures 4 shows the

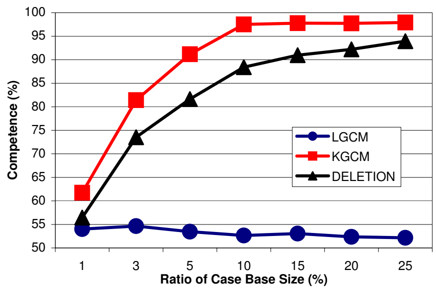


Figure 4: Spiral dataset: Competence on the case base mined by KGCM and other Case Mining algorithms using 1-NN.

result. In the figure, we compare the case-mining algorithms on the 1-NN accuracy.

The experimental result indicates that for the classification problem, the KGCM algorithm has higher competence than the case-deletion algorithm in the original space, as well as LGCM algorithm. We attribute this success to the result of the nonlinear transformation using Gaussian kernels. This demonstrates that for the nonlinear distribution problems such as the "*spiral*" dataset, KGCM gives better performance. As the ratio of case base size increases from 1% to 5%, we can see that KGCM rises most rapidly towards the optimal competence level, which is achieved by the input data set. Comparatively, the other two algorithms are rather slow to respond. When they all reach 25% in size, the difference between KGCM and case-deletion policy is very small.

In the related work section , we mentioned that the case-deletion policy is sensitive to noise. In the following experiment, we compare the competence of the "*spiral*" case base against the percentage of the amount of noise added to the original database; we call this percentage the "noise ratio" in this experiment. Figure 5 shows the comparison performance between the case-deletion policy and our KGCM algorithm when there is some noise in the raw database. The figure shows the competence curves over different noise ratio for the KGCM algorithm and case-deletion policy. We

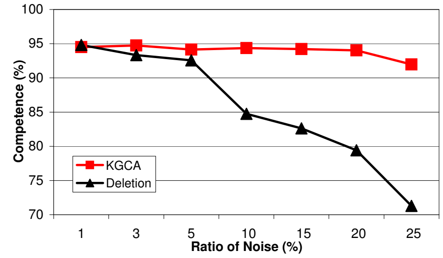


Figure 5: Competence trends of KGCM and case-deletion policy with different noise ratio.

can see that the KGCM algorithm is more robust than the case-deletion policy when noise increases.

To validate the KGCM algorithm, we do more experiments on several UCI datasets with KGCM algorithm, LGCM algorithm and case-deletion policy (see Figure 6). In many cases, the KGCM algorithm has higher competence than the case-deletion policy. One reason for this is KFD on the cases ensures that most important features are used for the determination of case coverage, rather than using the full set of features given in the input. This makes it possible to focus on the most important features of the case in mining the case base.

Furthermore, when the input problem distributions are nonlinear, the coverage of each case is relatively small when its reachability is small. Therefore, after a nonlinear transformation, the coverage of the case can become much larger. This essentially makes it possible to use a smaller number of cases to achieve the same level of competence. In the experiments, we also compare KGCM algorithm and LGCM algorithm. We also find that KGCM outperforms LGCM, which implies that using kernels in the case-mining algorithm is important for most of these problems.

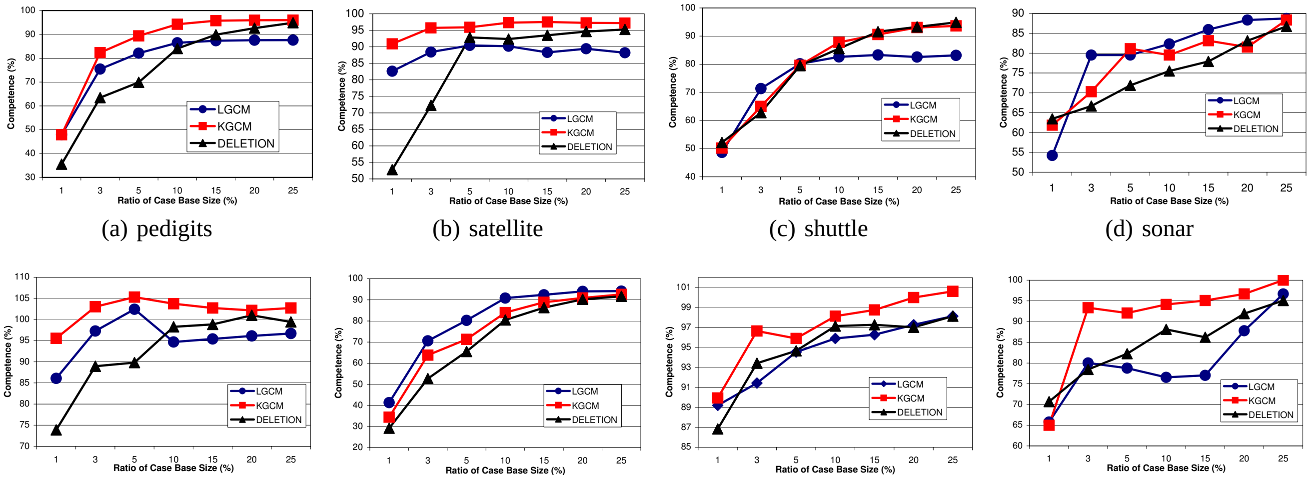
In addition, in the experiments on the "*scale*" and "*wdbc*" datasets (Figure 6(e) & 6(g)), we found that the competence of KGCM increased over 100% when the ratio of case base size reach 3% and 25% respectively. This shows that, in some situations, the competence of the case base generated by KGCM algorithm can increase over that of the original database.

Conclusions and Future Work

In this paper we proposed a new case-base mining algorithm in a nonlinear feature space. We convert an input data set into a feature space through a kernel transformation, introducing a new case mining algorithm (KGCM) that uses this transformed data. Optimal feature selection is done in this space through a correlation analysis, and the best cases are chosen based on the global diversity constraint and a maximal correlation constraint. We tested our algorithm on "*spiral*" dataset and the UCI datasets. Our result supports that in nonlinear domains the KGCM will be more appropriate for extracting a case base.

In the future we wish to extend this work to other kernel methods for the construction of case bases. One important subject is to automatically find the number of cases for a case base. This number should be controlled by the distribution of the input data. Another important topic is to develop

[1]The spiral dataset is from the classification toolbox with the book(Duda, Hart, & Stork 2000). The other datasets are available in http://www.ics.uci.edu/ mlearn/MLRepository.html.



(a) pedigits (b) satellite (c) shuttle (d) sonar

(e) scale (f) optdigits (g) wdbc (h) ionosphere

Figure 6: Competence on the case base mined by KGCM and other Case Mining algorithms using 1-NN.

an operational criterion for detecting when an input data set belongs to the category of nonlinear ones.

Acknowledgment We would like to thank the support of Hong Kong RGC CA0304.EG01 and Hong Kong UST Post-doctoral Matching fund.

References

Aamodt, A.; Sandtorv, H. A.; and Winnem, O. M. 1998. Combining case based reasoning and data mining - a way of revealing and reusing rams experience. In *Proceedings of the International Conference on Safety and Reliability, ESREL'98*, 1345–1351.

Bach, F. R., and Jordan, M. I. 2002. Kernel independent component analysis. *Journal of Machine Learning Research* 3:1–48.

Bonzano, A.; Cunningham, P.; and Smyth, B. 1997. Using introspective learning to improve retrieval in CBR: A case study in air traffic control. In *Proceedings of the 2nd International Conference on Case-Based Reasoning (ICCBR-97)*, LNAI, 291–302. Springer.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. Wiley-Interscience, 2nd edition.

Fisher, R. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7:179–188.

Fyfe, C., and Corchado, J. M. 2001. Automating the construction of cbr systems using kernel methods. *International Journal of Intelligent Systems* 16(4):571–586.

Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Leake, D.; Smyth, B.; Yang, Q.; and Wilson, D., eds. 2001. *Maintaining Case-Based Reasoning Systems*. Blackwell. Special issue of *Computational Intelligence*, 17(2), 2001.

Leake, D.; Kinley, A.; and Wilson, D. 1995. Learning to improve case adaptation by introspective reasoning and CBR. In *Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95)*, 229–240. Berlin: Springer Verlag.

Leake, D.; Kinley, A.; and Wilson, D. 1997. Case-based similarity assessment: Estimating adaptability from experience. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*. AAAI Press.

Mika, S.; Rätsch, G.; Weston, J.; Schölkopf, B.; and Müller, K.-R. 1999. Kernel fisher discriminant analysis. In *Neural Networks for Signal Processing 9 — Proceedings of the 1999 IEEE Workshop*. New York: IEEE.

Pan, R.; Yang, Q.; and Li, L. 2004. Case retrieval using nonlinear feature-space transformation. In *Advances in Case-Based Reasoning, 7th European Conference (ECCBR-04)*, 361–374.

Patterson, D. W.; Rooney, N.; Galushka, M.; and Anand, S. S. 2002. Towards dynamic maintenance of retrieval knowledge in cbr. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, 126–131.

Roth, V., and Steinhage, V. 1999. Nonlinear discriminant analysis using kernel functions. In Solla, S.; Leen, T.; and Müller, K.-R., eds., *Advances in Neural Information Processing Systems*, volume 12, 568–574. MIT Press.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10:1299–1319.

Smyth, B., and Keane, M. 1995. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 377–382. San Francisco: Morgan Kaufmann.

Watson, I. 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Mateo, CA: Morgan Kaufmann.

Zhu, J., and Yang, Q. 1999. Remembering to add: Competence-preserving case-addition policies for case base maintenance. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 234–241. Morgan Kaufmann.