Case Retrieval Using Nonlinear Feature-Space Transformation

Rong Pan[1], Qiang Yang[2], and Lei Li[1]

[1] Software Engineering Institute
Zhngshan University
Guangzhou, China
gzpanrong@etang.com,lncsri07@cs.zsu.edu.cn
[2] Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon Hong Kong, China
qyang@cs.ust.hk

Abstract. Good similarity functions are at the heart of effective case-based reasoning. However, the similarity functions that have been designed so far have been mostly linear, weighted-sum in nature. In this paper, we explore how to handle case retrieval when the case base is *nonlinear* in similarity measurement, in which situation the linear similarity functions will result in the wrong solutions. Our approach is to first transform the case base into a feature space using kernel computation. We perform correlation analysis with maximum correlation criterion(MCC) in the feature space to find the most important features through which we construct a feature-space case base. We then solve the new case in the feature space using the traditional similarity-based retrieval. We show that for nonlinear case bases, our method results in a performance gain by a large margin. We show the theoretical foundation and empirical evaluation to support our observations.

Keywords: Similarity, Case Base Transformation, Nonlinear Case Bases.
Paper type: Research.

1 Introduction

Case-based reasoning (CBR) is a problem-solving strategy that uses previous cases to solve new problems ([5], [6]). Over the years, CBR has enjoyed tremendous success as a technique for solving problems related to knowledge reuse. Several practical systems and applications [15] highlight the use of similarity based functions to find relevant cases from case bases. In building a case base, important descriptors of the case, which distinguish between the cases, are singled out and represented as features. The features are typically combined in some numerical computation for similarity. When a new problem is input, its features will be extracted to compute its similarity measure to other cases in the case base. The cases with the most similar measure will be retrieved for further analysis and adaptation ([5], [6]).

The quality of the retrieved case in a CBR system depends heavily on how to use the features to compute similarity measures. Various methods have been proposed to

P. Funk and P.A. González Calero (Eds.): ECCBR 2004, LNAI 3155, pp. 361–374, 2004.

compute the similarity ([2], [1], [6], [12], [13], [14]), where most approaches rely on linear combination of features to perform this function. However, when the nature of the case base is *nonlinear*, where similar cases cannot be found by a linear combination of the features, such a method will fail to deliver the most relevant cases. In this paper, we present a solution to solving this problem.

As an example, suppose that in a problem domain there are N different features. If the similarity in the domain is based on a high-order polynomial function of the features' values, then the similarity of the features cannot be explained by a simple weighted sum of the input features alone. A real world example of this nature is when we define the similarity of two boxes by their weight. Suppose the input features given are the boxes' three dimensions x_1, x_2 and x_3 and the density d of the material that makes up the boxes. Then the computation of the boxes' weight which defines the similarity function is not a linear weighted sum of the three dimensions; instead, it involves the multiplication of the four features x_1, x_2 ,x_3 and d.

One can argue that in the above example, one can input the nonlinear features such as $x_1 * x_2 * x_3$ directly as part of an input feature, but we cannot expect the designer of case bases to have this insight for every domain that he encounters. We would rather have the system find out these nonlinear features directly using an automatic method. This issue is the focus of our paper.

In this paper, we present a kernel-based method by which we transform a case base from the original space to a feature space with the *kernel trick*. For a nonlinear target case base, we propose nonlinear feature-extraction methods with a Maximum Correlation Criterion(MCC). With this criterion, one can find in feature space those features that have the highest correlation to target solution. We call this method the Kernel Case Correlation Analysis (KCCA). Our empirical results show that for many nonlinear domains, our KCCA method outperforms the traditional linear similarity functions applied in the original case space.

2 Transformation of a Case Base to Feature Case Space

In this paper, we focus on a dot-product formulation of the similarity computation. Consider a given case base $D = \{(x_i, y_i), i = 1, \dots, M, x_i \in \mathbb{R}^N, y_i \in \mathbb{R}\}$, where $\mathbb{R}$ is the real domain, x_i is a vector of input attributes (features), and y_i is the case-solution which corresponds to a target variable. For generality, we assume that the target variable is a continuous variable; discrete variables that are ordinal can also be converted to continuous ones. Then a popular method for computing the similarity between two cases is as follows: for an input problem $\overrightarrow{\mathbf{c}}$, the similarity between a case $\overrightarrow{\mathbf{x}}$ in the case base and the input case is computed as the $\mathbb{S}$ function:

$$S(\overrightarrow{\mathbf{c}}, \overrightarrow{\mathbf{x}}) = \frac{\overrightarrow{\mathbf{w}} \cdot (\overrightarrow{\mathbf{c}} - \overrightarrow{\mathbf{x}})}{|\overrightarrow{\mathbf{w}}|}$$

where $\overrightarrow{\mathbf{w}}$ is a weight vector. Then, the cases with the largest value of the above similarity function are chosen as a candidate case. These cases are *adapted* to obtain a new solution. In this paper, we consider a simplified situation where we choose a highest ranked case by the similarity function and use the target value of that case as a recommended solution for the input case. This corresponds to using a 1-NN method for case retrieval. Our work can be easily extended to k-NN computations. In cases where the case solution is a compound structure, such as in the case of planning [3], our solution corresponds to predicting a solution index for the corresponding target case.

Given a case base $\mathbb{D}$, we now consider how to transform the cases to the *feature space*. Our intuition is illustrated by the following example.

Consider a problem domain where the target $z = x^2 + 2y^2$, where x, y are the attributes. In the original space ($\mathbb{R}^2$), we cannot find a direction which correlates well with z, where the correlation coefficient is defined in $[-1, 1]$. Thus, if we use an 1-NN in the original space, we are not going to get good result.

Now consider the case in a nonlinear space induced by a 2-degree polynomial kernel [10]. The corresponding nonlinear map of the kernel is:

$$\phi : ([x], [y]) \mapsto \big([x]^2, [y]^2, [x][y], [y][x]\big)$$

With this kernel function, there exists $u = [x]^2 + 2[y]^2$, which is a linear transformation in the nonlinear feature space. We can see that u completely correlates to the target z. We can now solve the nonlinear case-base retrieval problem better by considering the correlation in a nonlinear feature space.

We now consider the general case. Let $\phi(x)$ be the nonlinear function which maps the input data into feature space, $\mathcal{F}$. Then in $\mathcal{F}$, we can define a matrix, in terms of a dot product in that space i.e. $K(i, j) = \langle\phi(x_i), \phi(x_j)\rangle$. Typically we select the matrix K based on our knowledge of the properties of the matrix rather than any knowledge of the function $\phi()$. The kernel trick allows us to define every operation in feature space in terms of the kernel matrix rather than the nonlinear function, $\phi()$.

Much research has been done in machine learning on feature selection and feature transformation in nonlinear feature space; some examples are Principal Component Analysis(PCA), single value decomposition(SVD)([4]), Kernel PCA, Sparse Kernel Feature Analysis, Kernel Projection Pursuit ([9], [10], [11]). However, in case-based reasoning, it is important to relate between the input and target variables and these works do not address this issue directly. In order to draw this relationship, we turn to Kernel Fisher Discriminant Analysis (KFDA)([7], [8]) which takes the class label of target into consideration. However, KFDA restricts the target to be of discrete values. In this paper, we present a novel nonlinear feature transformation method, by which we consider the correlation of input features with *a continuous valued target variable* in the feature space. Our questions are: first, for a given case base, how do we tell if a transformation to a feature space will give better result? Second, how do we perform feature selection in the feature space to result in maximal retrieval accuracy?

3 Kernel Correlation Analysis in the Feature Space

3.1 Review of Correlation Coefficient

In multivariate statistics, the correlation coefficient is used to measure the linear dependency between two random variables. Suppose that Y_1 and Y_2 are random variables with means μ_1 and μ_2 and with standard deviations σ_1 and σ_2, the correlation coefficient between Y_1 and Y_2 is defined as

$$\rho = \frac{\sum_1^M (Y_1 - \mu_1)(Y_2 - \mu_2)}{\sigma_1 \sigma_2} \tag{1}$$

It is easy to prove that the value of correlation coefficient ranges from -1 to 1. The larger the absolute value of ρ, the greater the linear dependence between Y_1 and Y_2. Positive values indicate that Y_1 increases with Y_2 ; negative values indicate that Y_1 decreases with Y_2 . A zero value indicates that there is no linear dependency between Y_1 and Y_2. (see Fig.1.) If we normalize Y_1 and Y_2 as

$$Y_1' = \frac{Y_1 - \mu_1}{\sigma_1}$$

and

$$Y_2' = \frac{Y_2 - \mu_2}{\sigma_2}$$

and define two vectors as follows

$$\mathbf{x_1} = (Y_{11}', Y_{12}', \ldots, Y_{1M}') \tag{2}$$

and

$$\mathbf{x_2} = (Y_{21}', Y_{22}', \ldots, Y_{2M}') \tag{3}$$

then, $\mathbf{x_1}$, $\mathbf{x_2}$ are identity vectors (whose 2-norms are equal to one) and the correlation coefficient is the inner product of $\mathbf{x_1}$, $\mathbf{x_2}$

$$\rho = \langle \mathbf{x_1}, \mathbf{x_2} \rangle$$

(see the right one of fig.1) On the left of the figure are two sets of scatter points(circles and dots) corresponding to Y_1 and Y_2 as they center around the mean point (μ_1, μ_2). If the scatter points mainly distribute in the 1-st and 3-rd quadrants(circle points), the

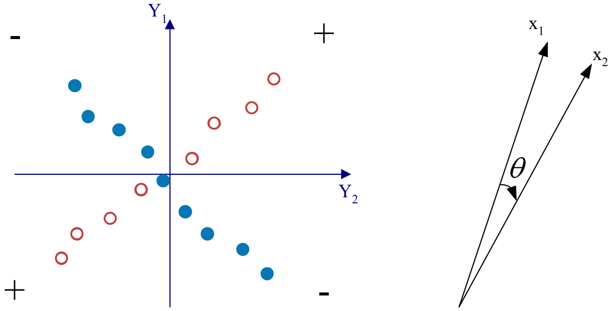


Fig. 1. Illustration of the correlation coefficient.

correlation coefficient is positive; if the scatter points mainly distribute in the 2-nd and 4-th quadrants, the correlation coefficient is negative(dots). If the scatter points equally distribute in the four quadrants, the correlation coefficient trends to zero. On the right of the figure are two vectors x_1 and x_2 as defined in (2) and (3), where θ is their angle. The correlation coefficient equals $\cos\theta$, where $\theta = 0$ means that they positively correlate, $\theta = \pi$ means that they completely negatively correlate, and $\theta = \pi/2$ means that they do not correlate.

3.2 Correlation Analysis on Input Case Base

We now propose a new feature extraction method similar to Fisher Linear Discriminant Analysis (FDA), extended to handle continuous target values. First, we consider the case in the input case space. Given an original case base with M cases:

$$D = \{(x_i, y_i), i = 1, \ldots, M, x_i \in \mathbb{R}^N, y_i \in \mathbb{R}\}$$

We assume that the attributes are centered around the origin and y_i is also normalized (assuming continuous attributes):

$$\sum_{i=1}^{M} x_i = 0\,, \sum_{i=1}^{M} y_i = 0\,, \sum_{i=1}^{M} y_i^2 = 1 \tag{4}$$

The correlation coefficient between the *j-th* coordinate $x^{(j)}$ and y is defined as follows:

$$cor\left(x^{(j)}, y\right) = \frac{\sum_{i=1}^{M} x_i^{(j)} y_i}{\sqrt{\sum_{i=1}^{M}\left(x_i^{(j)}\right)^2}\sqrt{\sum_{i=1}^{M} y_i^2}} = \frac{\sum_{i=1}^{M} x_i^{(j)} y_i}{\sqrt{\sum_{i=1}^{M}\left(x_i^{(j)}\right)^2}}$$

We now consider how to find features that best describe the correlation between attributes and the target. For many problems, there does not exist an independent variable whose correlation coefficient with the target variable is either 1 or -1. In such cases, we wish to find a new direction $\mathbf{w}$ in which the correlation coefficient between the projection of all cases on this direction and the target variable is maximized (absolute value maximizing). This new direction will serve as a *new feature* in the feature space and be used for computing case similarities. Suppose that $z_{\mathbf{w}}$ is the coordinate on the new direction $\mathbf{w}$ when a case $\mathbf{x}$ is projected on $\mathbf{w}$,

$$z_{\mathbf{w}} = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^{\tau}\mathbf{x}$$

Then the correlation coefficient of z and the target variable y is:

$$cor\left(z_{\mathbf{w}}, y\right) = \frac{\sum_{i=1}^{M} \langle \mathbf{w}, \mathbf{x}_i \rangle y_i}{\sqrt{\sum_{i=1}^{M} \langle \mathbf{w}, \mathbf{x}_i \rangle^2}\sqrt{\sum_{i=1}^{M} y_i^2}} = \frac{\sum_{i=1}^{M} \langle \mathbf{w}, y_i\mathbf{x}_i \rangle}{\sqrt{\sum_{i=1}^{M} \langle \mathbf{w}, \mathbf{x}_i \rangle^2}}$$

To increase the correlation between z_w and y is equivalent to maximizing the absolute value of the correlation coefficient between z_w and y. We know the following:

$$\arg\max_{\mathbf{w}} \left|cor\left(z_{\mathbf{w}}, y\right)\right| = \arg\max_{\mathbf{w}} \left(cor\left(z_{\mathbf{w}}, y\right)\right)^2$$

Thus, we can get

$$
\begin{aligned}
(cor\,(z_{\mathbf{w}}, y))^2 &= \frac{\left(\sum_{i=1}^{M} \langle \mathbf{w},\mathbf{x}_i \rangle y_i\right)^2}{\sum_{i=1}^{M} \langle \mathbf{w},\mathbf{x}_i \rangle^2} \\
&= \frac{\left\langle \mathbf{w},\sum_{i=1}^{M} y_i \mathbf{x}_i \right\rangle^2}{\sum_{i=1}^{M} w^{\tau} \mathbf{x}_i \mathbf{x}_i^{\tau} \mathbf{w}} \\
&= \frac{\mathbf{w}^{\tau}\left(\sum_{i=1}^{M} y_i \mathbf{x}_i\right)\left(\sum_{i=1}^{M} y_i \mathbf{x}_i\right)^{\tau}\mathbf{w}}{\mathbf{w}^{\tau}\left(\sum_{i=1}^{M} \mathbf{x}_i \mathbf{x}_i^{\tau}\right)\mathbf{w}}
\end{aligned}
$$

If we define $\mu = \sum y_i \mathbf{x}_i$, $C = \sum \mathbf{x}_i \mathbf{x}_i^{\tau}$, we can get a new *Rayleigh coefficient*:

$$
J(\mathbf{w}) = \frac{(\mathbf{w}^{\tau}\mu)^2}{\mathbf{w}^{\tau} C \mathbf{w}}
$$

Finally, to obtain the important directions which mostly correlate with the target variable and be used as the new feature, we compute $\arg\max_{\mathbf{w}} J(\mathbf{w})$. We call this the *Maximum Correlation Criterion*(abr. MCC).

To provide some intuition on how Correlation Analysis generates new feature, we show an experiment with an artificial 3-d linear target-function case base in Fig. 2. In this example, the input variables' x, y-values are elliptically distributed as the righthand figure shows. The target z-values are generated from $\mathbf{z} = \mathbf{x} - \mathbf{2y} + \xi$, where ξ is the white noise with a standard deviation of 0.2. The lefthand figure shows the 3-d coordinate of the case base. The right hand figure is the projection on the $x - y$ plane and illustrates the Principle Component Analysis (PCA) and Correlation Analysis for this case base. PCA does not consider the target variable and simply returns the direction of statistical maximum variance as the first eigenvector. Correlation Analysis, on the other hand, returns a direction $(1, -2)$ that correlate to the continuous target variable the most.

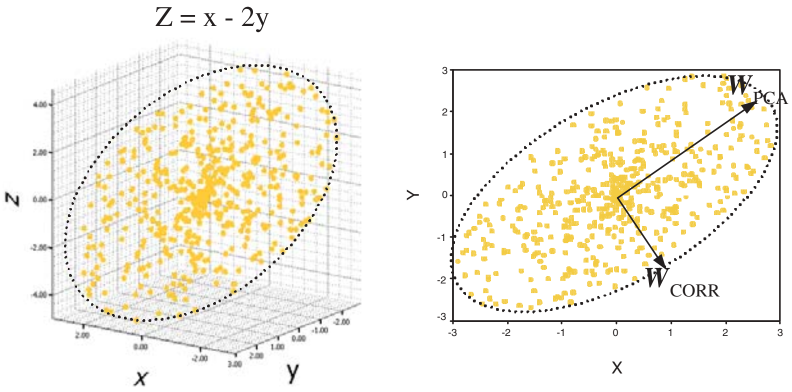


Fig. 2. 2-dimension input and 1-dimension target artificial example, with 500 cases generated.

3.3 Transformation of Case Base by KCCA

In a nonlinear target domain, it is often difficult to find one or several directions that correlates well with the target variable. With the "kernel trick", we now consider projecting a case base into a nonlinear feature space. Then, we attempt to transpose the linear correlation analysis to the nonlinear case base also with MCC. We call this method *Kernel Case Correlation Analysis* (abr. KCCA). In the next subsection, we give some examples of an artificial domain to demonstrate the merit of KCCA with 1-NN.

Given an original case base with M centered observations, the new case base can be obtained in a feature space FS by:

$$\Phi(D) = \{(\phi(x_i), y_i), i = 1, \ldots, M, x_i \in \mathbb{R}^N, y_i \in \mathbb{R}\}$$

where $\phi(x)$ and y_i are centered on the origin:

$$\sum_{i=1}^{M} \phi(x_i) = 0\,, \sum_{i=1}^{M} y_i = 0\,, \sum_{i=1}^{M} y_i^2 = 1$$

We project our input case in the new direction $\mathbf{w}$ in the feature space. Like the Kernel PCA [9], we assume that $z_{\mathbf{w}}$ is a new coordinate:

$$\mathbf{w} = \sum_i \alpha_i \phi(x_i) \quad \text{and} \quad z_{\mathbf{w}} = \langle \mathbf{w}, \phi(x) \rangle$$

Then the correlation coefficient of $z_{\mathbf{w}}$ and y is:

$$\begin{aligned}(cor(z_{\mathbf{w}}, y)) &= \frac{\sum_i \langle \mathbf{w}, y_i \phi(x_i) \rangle}{\sqrt{\sum_i \langle \mathbf{w}, \phi(x_i) \rangle^2}} = \frac{\sum_i \sum_j \alpha_j \langle \phi(x_i), \phi(x_j) \rangle y_i}{\sqrt{\sum_i \left(\sum_j \alpha_j \langle \phi(x_j), \phi(x_i) \rangle\right)^2}} \\ &= \frac{\alpha^{\tau} K y}{\sqrt{\sum_i (\alpha^{\tau} K_i)^2}} = \frac{\alpha^{\tau} K y}{\sqrt{\alpha^{\tau} \sum_i \left(K_i K_i^{\tau}\right) \alpha}} = \frac{\alpha^{\tau} K y}{\sqrt{\alpha^{\tau} K K^{\tau} \alpha}}\end{aligned}$$

where K is the Kernel Matrix, and $\alpha = (\alpha_1, \ldots, \alpha_M)^{\tau}$.

Next, we consider the Rayleigh coefficient

$$J(\alpha) = (cor(z, y))^2 = \frac{(\alpha^{\tau} K y)^2}{\alpha^{\tau} K K^{\tau} \alpha}$$

where K is the kernel matrix and $y = (y_1, \ldots, y_M)^{\tau}$. Let $\mu = Ky$, $\mathbb{M} = \mu\mu^{\tau}$, and $N = KK^{\tau}$. Finally we obtain an expression for

$$J(\alpha) = \frac{(\alpha^{\tau} \mu)^2}{\alpha^{\tau} N \alpha} = \frac{\alpha^{\tau} \mathbb{M} \alpha}{\alpha^{\tau} N \alpha} \tag{5}$$

[10] presents several equivalent ways of the similar problems of maximizing Equation (5). One method is to solve the generalized eigenvalue problem and then selecting eigenvectors α with maximal eigenvalues λ, as follows:

$$\mathbb{M}\alpha = \lambda N \alpha \tag{6}$$

Like Kernel PCA, we can compute the projection on the eigenvectors $\mathbf{w}^k$ in the feature space as follows:

$$\left(\mathbf{w}^k, \phi\left(x\right)\right) = \sum \alpha_i^k \left(\phi\left(x_i\right), \phi\left(x\right)\right) = \sum \alpha_i^k K\left(x_i, x\right) \tag{7}$$

Each eigenvector then corresponds to an attribute that we can select in the feature space for defining the cases. Let $< X_1, X_2, \ldots, X_n >$ be the selected attributes in the feature space, where the target value remains the same. We can then build a feature-space case base for case based reasoning. In particular, our feature-space case-based reasoning algorithm is shown as follows:

Algorithm	Kernel Case Correlation Analysis(KCCA)
Step1.	Transform the case base by solving the Eq.(6) and computing the selected attributes for the case base with Eq.(7).
Step2.	For an input case *c*, transform c to the feature space using the Eq.(7). The weight is determined by the correlation coefficient between the nonlinear feature and the target.
Step3.	Compute the weighted similarity between *c* and every case in the Case Base
Step4.	Select the case x_i with the largest similarity value.
Step5.	Return the target value y of x_i as the solution.

The KCCA algorithm is based on an 1-NN computation in the feature space. However, it would be straightforward for us to extend it to a k-NN algorithm.

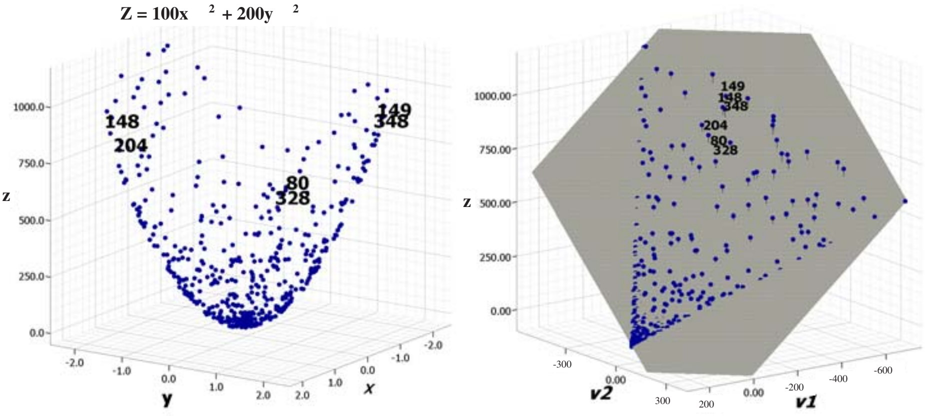
3.4 An Example for KCCA

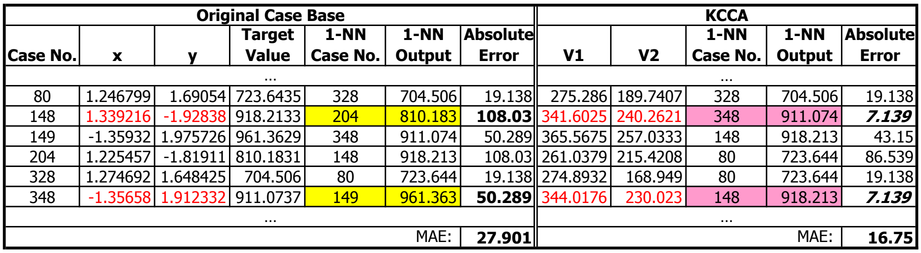
To give some intuition on how KCCA generates new case base and the merit of KCCA, we show in Fig. 3 an experiment on an artificial case base with two input dimensions and one target dimension using a polynomial kernel of degree two.

In this example, we have 500 data randomly generated cases in the following way: the input variable's (x,y) values are distributed in a circle, and the target z-values are generated from $z = 100x^2 + 200y^2 + \xi$, where ξ is the white noise with a standard deviation 0.2. The top left figure shows the 3-d coordinate of the case base. The top right one is the result of our KCCA on this case base. V_1, V_2 are the first two directions with which the linear regression plane (the hexagonal plane) is a good fit for the actual values. The table at the bottom shows the result of a segment with 6 cases in 500 cases before and after applying KCCA. The case numbers are also marked in the top figures. In this table, we can find that the overall MAE (Mean Absolute Error) of 1-NN with KCCA is about 40% lower than the overall MAE of 1-NN with original case base. Moreover, we can find that the nearest neighbors in the original case base of case no. 148 and case no. 348 are no. 204 and no. 149. The errors are respectively 108.03 and 50.289. In contrast, KCCA put these cases (symmetrical in original case base) together, so that the errors reduce to 7.139.

4 Experimental Analysis

We claim that KCCA benefits from superior efficiency and performance in terms of retrieval accuracy. We test our algorithms on a number of case bases. Our experiments





Original Case Base							KCCA				
Case No.	x	y	Target Value	1-NN Case No.	1-NN Output	Absolute Error	V1	V2	1-NN Case No.	1-NN Output	Absolute Error
...							...				
80	1.246799	1.69054	723.6435	328	704.506	19.138	275.286	189.7407	328	704.506	19.138
148	1.339216	-1.92838	918.2133	204	810.183	**108.03**	341.6025	240.2621	348	911.074	***7.139***
149	-1.35932	1.975726	961.3629	348	911.074	50.289	365.5675	257.0333	148	918.213	43.15
204	1.225457	-1.81911	810.1831	148	918.213	108.03	261.0379	215.4208	80	723.644	86.539
328	1.274692	1.648425	704.506	80	723.644	19.138	274.8932	168.949	80	723.644	19.138
348	-1.35658	1.912332	911.0737	149	961.363	**50.289**	344.0176	230.023	148	918.213	***7.139***
...							...				
					MAE:	**27.901**				MAE:	**16.75**

Fig. 3. A 2-dimension input and 1-dimension target artificial example.

are performed on an artificial case base and several publicly available case bases; in this paper, the case bases are: *Wine Recognition*, *Boston House*, *bank*, *travel* and *comp-activ*[1]. For each application domain, we validate our KCCA with linear regression and 1-NN respectively.

We first used an artificial domain where there are three attributes and a numerical target value. This domain is designed to be a nonlinear one, and is aimed at showing the validity of our KCCA algorithm to show that for nonlinear domains where the linear regression technique cannot produce good results, our KCCA method can indeed make a dramatic improvement. For each system the retrieval similarity criterion is based on a cosine-similarity threshold; an input problem was successfully solved if the similarity between the problem and the retrieved case exceeded the threshold.

4.1 Testing KCCA

Artificial Domain Experiments. We now conduct experiments to test the effect of KCCA in several domains, to demonstrate how effective the new algorithm is. The first

[1] The Wine and Boston domains are available in http://www.ics.uci.edu/∼mlearn/MLSummary.html. The bank and comp-activ domains are available in http://www.cs.utoronto.ca/∼delve/data/datasets.html. For the Travel domain, we thank Lorraine McGinty and Barry Smyth for the data.

experiment is concerned with evaluating the KCCA method in an artificially generated domain(same domain of fig.3). The common feature of these problem domains is that they all exhibit nonlinear nature in terms of feature descriptions. To demonstrate this point, we first show, in Figures (4) and (5), the result of linear regression, 1-NN and 1-NN with KCCA in these problem domains. “Actual” means the true value of the target function. Thus, the closer a method is to the actual value, the better the method is. As we can see, linear regression performs poorly in these domains.

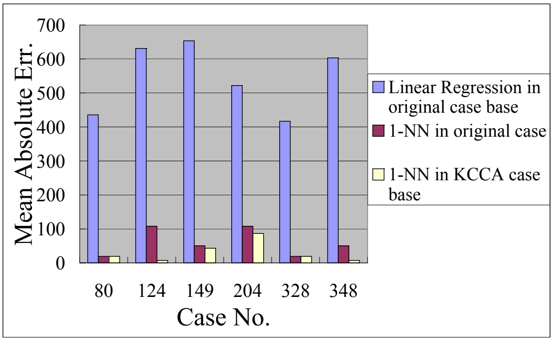


Fig. 4. Comparison of KCCA with linear regression and 1-NN in the original space.

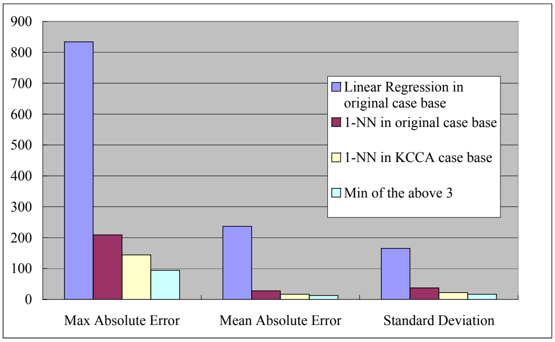


Fig. 5. The retrieval result of 6 cases from the artificial data set, when compared with Linear Regression and 1-NN in the original space and 1-NN with KCCA.

To test the efficiency of the KCCA method, we plotted the mean absolute error of KCCA as a function of the number of features. The result is shown in Figure (6). As we can see, the first several eigenvectors found by KCCA are in fact much better features than the rest in the feature space. This gives us confidence as to the use of KCCA in case-based reasoning.

Public Domain Experiments. We used the Bank domain, the com-activ domain and the Wine domain (available from the UCI Machine Learning Repository). The result is

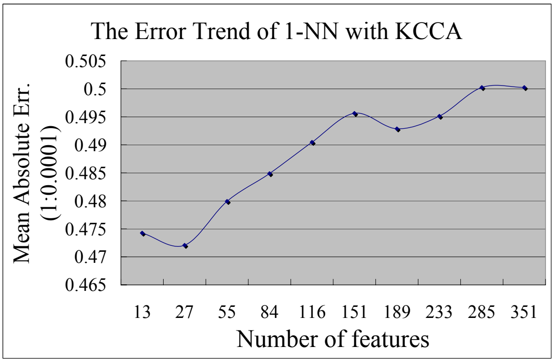


Fig. 6. The trend of average error as a function of different number of features computed by KCCA.

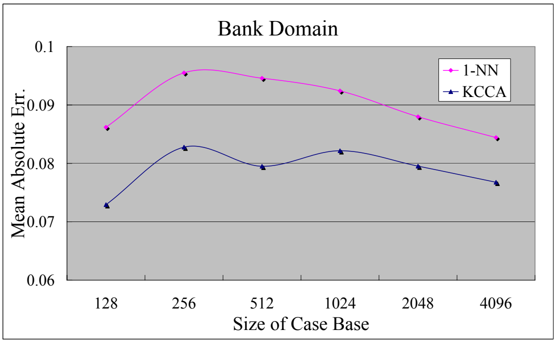


Fig. 7. Test KCCA on the Bank Base. The figure shows the average error as a function of different sizes of the case base.

shown in Figure (7), where we compare the mean absolute error KCCA and 1-NN in the original space as a function of different case-base size. As can be seen, using the KCCA uniformly outperforms 1-NN in terms of the MAE, and the difference in the error rate is the largest when the case base size reaches 512. Figure (8) shows the test result on the Travel database used often in case-based reasoning testing [14], where the objective is defined on the Price attribute, and Figure (9) shows the result of the com-activ base. As can be seen from both bases, the KCCA method outperforms 1-NN in the original space.

Table (1) shows a similar comparison with different kernels for the KCCA. As we can see, the MAE for the 1-NN method in the original space is as large as 172.944, whereas for the Gaussian kernel with the appropriately chosen parameters the MAE can be made smaller. One interesting fact is that the polynomial kernel in fact results in larger MAE error; this indicates to us that the Wine domain is in fact a linear domain, and thus 1-NN in the original space will perform just fine. It also indicated to us that the performance of the KCCA method is sometimes sensitive to the choice of kernel functions.

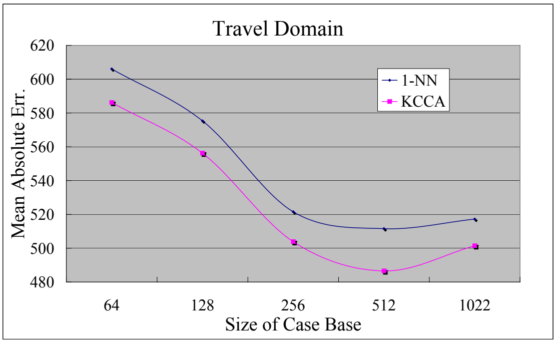


Fig. 8. Test KCCA on the Travel Database. The figure shows the average error as a function of different sizes of the case base.

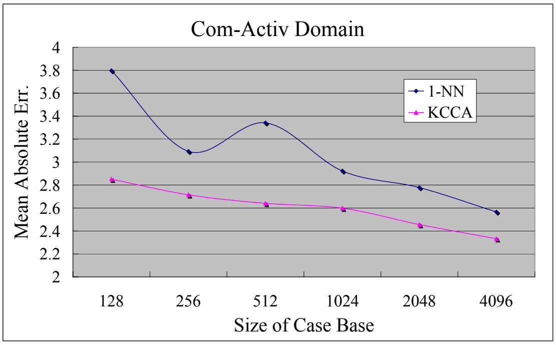


Fig. 9. Test KCCA on the Computer Database. The figure shows the average error as a function of different sizes of the case base.

We also noted that the time for the KCCA computation involves building the feature space case base and case correlation analysis. This is a one time cost. Once done, we can use the result to measure the similarity of each new case. This latter computation has the same time complexity as the original case based retrieval cost.

5 Conclusions and Future Work

In this paper we proposed a new solution to case base retrieval using a new nonlinear similarity function, when the nature of the problem domain is nonlinear. We used an FDA for finding the best attributes to compute a new case base in the feature space. We noted that the FDA cannot handle the continuous target case. We then proposed a new correlation analysis in the feature space, where we designed a new case based reasoning algorithm we call KCCA. Our approach is to first transform the case base into a feature space using kernel computation. We perform correlation analysis with maximum correlation criterion(MCC) in the feature space to find the most important features through which we construct a feature-space case base. We solve the new case in the feature space using the traditional similarity-based retrieval. We then empirically

Table 1. Test KCCA on the Wine data set. We compare the Mean Absolute Error with different kernels.

Original Space						
MAE	172.944					
KCCA with polynomial kernels of different parameters						
degree	2	3	4	5	6	7
MAE	217.96	165.6	217.9	179.8	267.5	191
KCCA with Gaussian kernels of different parameters						
Gamma	0.05	0.1	0.2	0.3	0.4	0.5
MAE	166.48	178.1	167.3	163.8	168.9	176

tested the KCCA for artificially generated data and for UCI data sets. Our result supports our initial claim that in nonlinear domains the KCCA will be more appropriate measure of similarity.

In the future we wish to extend this work to other methods for the construction of case bases. One important subject is to design kernel functions for the purpose of selecting cases from raw datasets, so that the CBR solution can be carried out. Another direction is to apply the kernel computation to more sophisticated kinds of target values, instead of just a single real value.

Acknowledgment

We thank Lorraine McGinty and Barry Smyth for the travel data. Rong Pan and Lei Li are supported by Zhong Shan University. Qiang Yang is supported by a Hong Kong RGC grant and an ITF grant, and a Chinese National 973 project.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–52, 1994.
2. D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.
3. K. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego, 1989.
4. I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 2002.
5. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.
6. D. Leake, A. Kinley, and D. Wilson. Case-based similarity assessment: Estimating adaptability from experience. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. AAAI Press, 1997.
7. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Kernel fisher discriminant analysis. In *Neural Networks for Signal Processing 9 – Proceedings of the 1999 IEEE Workshop*, New York, 1999. IEEE.
8. V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 568–574. MIT Press, 1999.
9. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

10. B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.
11. A. Smola, O. Mangasarian, and B. Schölkopf. Sparse kernel feature analysis, 1999.
12. B. Smyth and M. Keane. 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*, pages 377–382, San Francisco, August 1995. Morgan Kaufmann.
13. B. Smyth and M. Keane. Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102(2):249–293, 1998.
14. B. Smyth and E. McKenna. Footprint-based retrieval. In *Proceedings of the Third International Conference on Case-Based Reasoning*, pages 343–357, Berlin, 1999. Springer Verlag.
15. I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, San Mateo, CA, 1997.