

A Method of Explicit Mappings for Kernel Data Analysis and Applications

Sadaaki Miyamoto
Department of Risk Engineering
University of Tsukuba
Ibaraki 305-8573, Japan
Email: miyamoto@risk.tsukuba.ac.jp

Keisuke Sawazaki
Master's Program in Risk Engineering
University of Tsukuba
Ibaraki 305-8573, Japan
Email: sawazaki@soft.risk.tsukuba.ac.jp

Abstract—The method of kernel data analysis is now a standard tool in modern data mining. An implicit mapping into a high-dimensional feature space is assumed in this method, in other words, an explicit form of the mapping is unknown but their inner product should be known instead. Contrary to this common assumption, we propose a method of explicit mappings. The reason why we use explicit mappings is as follows. (1) The use of these mappings does not lose any fundamental information in kernel data analysis. (2) We have the same formulas in kernel methods with and without the explicit mappings. (2) Usually the derivation becomes simpler by using these mappings. (3) New applications of the kernel methods become possible using these mappings. Two types of the mappings are proposed, one of which uses $\Phi(x_k) = e_k$ ($k = 1, \dots, N$) while the second type uses $\Phi(x_k) = K^{\frac{1}{2}} e_k$, where $K = (K(x_i, x_j))$ is $N \times N$ matrix. As an application we consider L_1 space fuzzy c -means clustering and lower dimensional approximation of kernel K . The effectiveness of the proposed method is shown by numerical examples.

Keywords—data mining; kernel function; explicit mapping; clustering.

I. INTRODUCTION

Kernel functions [7] have been used in support vector machines [8] and are now becoming a standard tool in many applications of data mining. The *kernel trick* is commonly used in this method which means that we consider a mapping, e.g. $\Phi(x)$, from a data space into a high-dimensional feature space of which an explicit form is unknown, but we know the explicit functional form of their inner product $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$ in the feature space. The function $K(x, y)$ is called a kernel function. Hence formulas in data analysis should be described in terms of the inner product of $\Phi(x)$, since the functional form of $\Phi(x)$ is unknown.

Kernel functions are known to be useful in many applications, but the derivations are generally complicated, since original formulas should be rewritten by the inner product forms. A typical example is kernel fuzzy c -means clustering [4] and kernel VQ clustering [5].

Contrary to this common understanding, we propose a method of kernel functions using explicit mappings. The idea is simple: the original idea is to assume a mapping $\Phi: \mathbf{R}^p \rightarrow H$, where $\mathbf{R}^p$ is the data space and H is the feature space. Instead, we assume $\Phi: \{x_1, \dots, x_N\} \rightarrow \mathbf{R}^p$, where $\{x_1, \dots, x_N\}$ is the set of objects to be analyzed. As a result we do not need to have the full functional form of $K(x, y)$; it is sufficient to

know $K(x_i, x_j)$ ($1 \leq i, j \leq N$). However, we do not know the actual form of $\Phi(x_k)$ but we know only $K = (K(x_i, x_j))$. How to handle $\Phi(x_k)$? The answer is simple: we consider two mappings:

- 1) $\Phi(x_k) = e_k$;
- 2) $\Phi(x_k) = K^{\frac{1}{2}} e_k$.

By these mappings the derivation of a kernel method becomes simpler, and some new techniques can be developed, as we will see later.

To illustrate the effectiveness of the present method, a typical result of kernel fuzzy c -means clustering with L_2 and L_1 norms will be shown. Further, a low dimensional approximation of a kernel is shown.

II. EXPLICIT MAPPINGS FOR KERNEL FUNCTIONS

A. Preliminary consideration

A set of objects $X = \{x_1, \dots, x_N\} \subset \mathbf{R}^p$ is assumed to be given. They are points in the p -dimensional real space: $x_k = (x_k^1, \dots, x_k^p)^\top \in \mathbf{R}^p$. We consider a mapping into a high-dimensional feature space $\Phi: \mathbf{R}^p \rightarrow H$ and associated kernel function as usual:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

where $\langle \cdot, \cdot \rangle$ is the inner product of H . We also assume $\|\cdot\|_H$ is the norm of H .

In this section we first suppose, as usual, we do not know an explicit form of function $\Phi(\cdot)$ but we know $K(x, y)$. Specifically, the Gaussian kernel is frequently used:

$$K(x, y) = \exp(-\lambda \|x - y\|^2),$$

where $\lambda > 0$ and $\|x\|$ is the norm of $\mathbf{R}^p$.

1) *Kernel-based fuzzy c -means clustering*: Let us review the method of kernel-based fuzzy c -means clustering [4], [5]. In this section we do not use an explicit mapping.

An objective function of fuzzy c -means using the feature space H is the following.

$$J_H(U, W) = \sum_{i=1}^c \sum_{k=1}^N (u_{ki})^m \|\Phi(x_k) - W_i\|_H^2, \quad (m > 1)$$

where $U = (u_{ki})$ is $n \times c$ membership matrix of x_k to cluster i . U has the next constraint when it is optimized.

$$M = \{U = (u_{ki}) : \sum_{i=1}^c u_{ki} = 1, u_{\ell j} \geq 0, \forall \ell, j\},$$

while $W = (W_1, \dots, W_c)$ is a collection of cluster centers in H .

The iterative algorithm of fuzzy c -means clustering [1] is basically an alternative minimization of $J_H(U, W)$ with respect to U and W until convergence. We have the next solutions [5]:

$$u_{ki} = \left[\sum_{j=1}^c \left(\frac{D(x_k, W_i)}{D(x_k, W_j)} \right)^{\frac{1}{m-1}} \right]^{-1}, \quad (1)$$

$$W_i = \frac{\sum_{k=1}^N (u_{ki})^m \Phi(x_k)}{\sum_{k=1}^N (u_{ki})^m}, \quad (2)$$

where we put

$$D(x_k, W_i) = \|\Phi(x_k) - W_i\|_H^2. \quad (3)$$

Equations (1) and (2) should be repeated until convergence, but since $\Phi(x_k)$ is unknown, we use another formula for kernel fuzzy c -means clustering.

The formula is derived by eliminating (2) from iteration, i.e., we substitute (2) into (3) to have an updating formula for $D(x_k, W_i)$ [4]:

$$\begin{aligned} D(x_k, W_i) &= K(x_k, x_k) - \frac{2}{\sum_{k=1}^N (u_{ki})^m} \sum_{j=1}^N (u_{ji})^m K(x_j, x_k) \\ &+ \frac{1}{\{\sum_{k=1}^N (u_{ki})^m\}^2} \sum_{j=1}^N \sum_{\ell=1}^N (u_{ji} u_{\ell i})^m K(x_j, x_\ell). \end{aligned} \quad (4)$$

We thus repeat (1) and (4) until convergence in kernel-based fuzzy c -means clustering.

B. Two explicit mappings

The above derivation needs transformation of an original algorithm into another one, due to the fact that no explicit mapping is available.

In contrast, if we have an explicit mapping, then no additional derivation is needed and we can use an existing algorithm.

We show two explicit mappings. First mapping is very simple:

$$\Phi_1(x_k) = \mathbf{e}_k \quad (k = 1, 2, \dots, N) \quad (5)$$

where $H = \mathbf{R}^N$ and $\mathbf{e}_k$ is the k -th unit vector that has unity as i th component and all other components are zero. Note that $\Phi_1: X \rightarrow \mathbf{R}^N$, i.e., $\Phi(\cdot)$ is not defined on $\mathbf{R}^p$ but is limited

to the finite set X . We also assume that the inner product of $\mathbf{R}^N$ is

$$\langle \mathbf{e}_k, \mathbf{e}_\ell \rangle = K(x_k, x_\ell) \quad (6)$$

instead of the standard inner product $\langle \mathbf{e}_k, \mathbf{e}_\ell \rangle = \delta_{k\ell}$ ($\delta_{k\ell}$ is the Kronecker delta).

We have the next proposition.

Proposition 1: If kernel function $K(x, y)$ is positive-definite, $\langle \mathbf{e}_k, \mathbf{e}_\ell \rangle$ defined by (6) satisfy the axioms of the inner product of $\mathbf{R}^N$, that is, $\mathbf{R}^N$ with (6) is an inner product space.

(Proof) The detailed proof is given in standard textbooks [7]. As a rough sketch of the proof, note that the Mercer condition [8]

$$\int \int K(x, y) \eta(x) \eta(y) dx dy \geq 0$$

for all $\eta(x)$ guarantees

$$\sum_{i,j} K(x_i, x_j) \zeta_i \zeta_j \geq 0$$

$\forall \zeta_i \in \mathbf{R}$, by putting $\eta(x) = \sum_i \zeta_i \delta(x - x_i)$. Thus the matrix $K = (K(x_i, x_j))$ is positive semi-definite. (QED)

We introduce second explicit mapping that may be more time consuming in calculation but useful in applications. Note that a positive definite matrix K can be divided into $K = K^{\frac{1}{2}} K^{\frac{1}{2}}$. We thus define the second mapping:

$$\Phi_2(x_k) = K^{\frac{1}{2}} \mathbf{e}_k \quad (k = 1, 2, \dots, N) \quad (7)$$

In this case, $\Phi_2: X \rightarrow \mathbf{R}^N$, i.e., the domain and the codomain are the same as Φ_1 , but now the inner product of $\mathbf{R}^N$ is standard:

$$\langle \mathbf{e}_k, \mathbf{e}_\ell \rangle = \delta_{k\ell}. \quad (8)$$

We now apply mappings Φ_i ($i = 1, 2$) to fuzzy c -means clustering. It is sufficient to show the optimal solution of W_i .

Proposition 2: For all positive semi-definite kernel $K(x, y)$ and mapping Φ_1 by (5), the cluster centers are given by

$$W_i = \left(\frac{(u_{1i})^m}{\sum_{k=1}^N (u_{ki})^m}, \dots, \frac{(u_{ni})^m}{\sum_{k=1}^N (u_{ki})^m} \right)^T, \quad i = 1, \dots, c \quad (9)$$

When Φ_2 by (7) is used, the cluster centers are given by

$$W_i = K^{\frac{1}{2}} \left(\frac{(u_{1i})^m}{\sum_{k=1}^N (u_{ki})^m}, \dots, \frac{(u_{ni})^m}{\sum_{k=1}^N (u_{ki})^m} \right)^T, \quad i = 1, \dots, c \quad (10)$$

(Proof) The proof of this proposition is not difficult by observing closely (2). Note that W_i given by (2) is the solution of

$$\min_{W_i} \sum_k (u_{ki})^m \|x_k - W_i\|^2$$

where the space can be an arbitrary inner product space, since the derivation of (2) uses a general variational principle valid for any Hilbert space. Note moreover that we substitute (9) into (1) to have the optimal solution of U . It should be noted that although optimal W_i is the same for all positive definite

kernel, optimal U differs because (6) give different values for different kernels. (QED)

The proof of the next proposition is almost trivial and is omitted.

Proposition 3: Substituting (9) or (10) into (3), we have (4).

That is, (9) (or (10)) derived from mapping (5) (or (7)) has all necessary and sufficient information for kernel fuzzy c -means clustering.

The mappings (5) and (7) are moreover useful for application to SOM [3] and VQ clustering [5], where vectors for updating quantization vectors should be based on learning. The explicit mappings enable vector representations in $\mathbf{R}^N$, and hence we can use every formulas in VQ and SOM.

C. L_1 space fuzzy c -means

Let us consider a method specific to an explicit mapping. Although kernel-based methods assume Euclidean spaces, we use an L_1 space clustering [5], which may be useful in applications. The L_1 space fuzzy c -means clustering uses

$$J_H(U, W) = \sum_{i=1}^c \sum_{k=1}^N (u_{ki})^m \|\Phi(x_k) - W_i\|_{L_1}, \quad (m > 1),$$

where we assume the second mapping: $\Phi(x_k) = K^{\frac{1}{2}} \mathbf{e}_k$. Hence

$$\|\Phi(x_k) - W_i\|_{L_1} = \sum_{j=1}^N |(K^{\frac{1}{2}} \mathbf{e}_k)^j - W_i^j|$$

where z^j is j -th component of z .

Accordingly the solution is

$$u_{ki} = \left[\sum_{j=1}^c \left(\frac{\|\Phi(x_k) - W_i\|_{L_1}}{\|\Phi(x_k) - W_j\|_{L_1}} \right)^{\frac{1}{m-1}} \right]^{-1},$$

while calculation of W_i needs an algorithm of an weighted median [5] of which the details are omitted for simplicity. We note, however, that calculation of an weighted median is frequently faster than the weighted centroid in the ordinary L_2 -space based method.

Note also that this algorithm is valid only for the explicit mapping, since it does not assume an Euclidean feature space.

D. Low dimensional approximation of kernels

Another method specific to the explicit mapping is a low dimensional approximation of kernels. As we will see in the next section of numerical examples, the effectiveness of a kernel data analysis frequently depends on a small dimension of those subspaces of a few top eigenvalues of the kernel matrix. This means that a low dimensional approximation of the kernel simplifies the analysis and yet preserves its effectiveness. Such approximation is also suitable when an original kernel matrix is not known to be positive-definite. Then we can take a subspace of a number of top positive eigenvalues.

Let us assume that the original $N \times N$ kernel matrix K is decomposed into

$$K = T^\top \Lambda T,$$

where T is an orthogonal matrix which is composed of eigenvectors, and

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$$

is the diagonal matrix of eigenvalues with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$. Let us consider D -dimensional approximation by taking

$$\hat{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_D, 0, \dots, 0),$$

where $D < N$ and $\lambda_D > 0$. Accordingly we put

$$\hat{K} = T^\top \hat{\Lambda} T,$$

and use $\hat{K}$ instead of K . We thus assume $\Phi_2(x_k) = \hat{K}^{\frac{1}{2}} \mathbf{e}_k$, where

$$\begin{aligned} \hat{K} &= T^\top \hat{\Lambda}^{\frac{1}{2}} T \\ &= T^\top \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_D}, 0, \dots, 0) T. \end{aligned}$$

It is easily seen that the feature space is now D -dimensional and when $D \ll N$, we have a low-dimensional approximation of the kernel space.

Note 1: This approximation appears similar to the kernel principal component analysis, but the concepts and the derivations are different. The method herein is an approximation of kernel K , while the kernel principal component analysis is approximation of a covariance matrix. Note also that, when D is two or three, we can visualize the ‘principal part’ of a kernel feature space. Thus the low dimensional approximation is useful as a visualization method for the feature space.

III. NUMERICAL EXAMPLES

A set of data shown in Fig. 1 was used to show the effectiveness of the proposed method. This example is well-known in the sense that a usual fuzzy c -means (abbreviated as FCM) algorithm cannot separate the outer circle and the inner ball, while a kernel-based c -means algorithm can. Figure 2 depicts the result of an ordinary FCM, while Fig. 3 with the kernel-based FCM with the explicit mappings of the Gaussian kernel successfully separated the two groups.

Moreover Fig. 4 shows two major principal components with mapping Φ_2 , where the method of the ordinary principal component analysis has been applied. In contrast, Figure 5 shows the two-dimensional approximation ($D = 2$) of the Gaussian kernel, which shows a similar, yet different configuration of the points.

The same result of perfect separation in Fig. 2 was obtained by the L_1 -based method with an L_1 exponential kernel:

$$K(x, y) = \exp(-\text{const} \|x - y\|_{L_1})$$

which has been proved to be a kernel function [8], [2]. The two-dimensional approximation with this kernel is shown in Fig. 6.

Note that both Φ_1 and Φ_2 give the successful separation in Fig. 2 because they are equivalent to (4), while Φ_1 cannot

be used for the principal component analysis, since the coordinates for Φ_1 does not satisfy $\langle e_k, e_\ell \rangle = \delta_{k\ell}$ which is necessary in the principal component analysis.

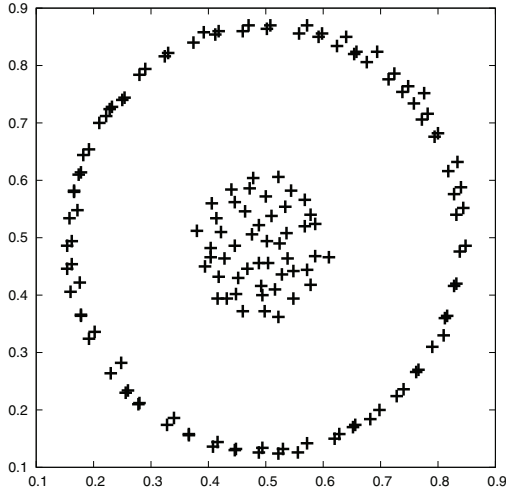


Fig. 1. A ball in a circle with 150 data points.

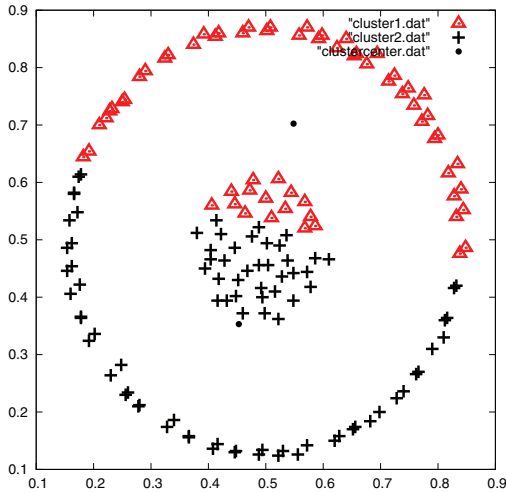


Fig. 2. The result of ordinary FCM applied to data in Fig. 1.

IV. CONCLUSIONS

We have proposed the method of explicit mappings for kernel based data analysis. Either one of Φ_1 and Φ_2 can be used, as there is no fundamental difference between the two, but the second mapping seems to be more useful in applications. Indeed, the L_1 -based method and the low dimensional approximation of kernels has been developed for the second mapping. The low dimensional approximation is also useful when we do not know whether a kernel function is positive-definite or not, since it may be difficult to prove positive-definiteness in real applications.

In summary, we note the following advantages of the present method.

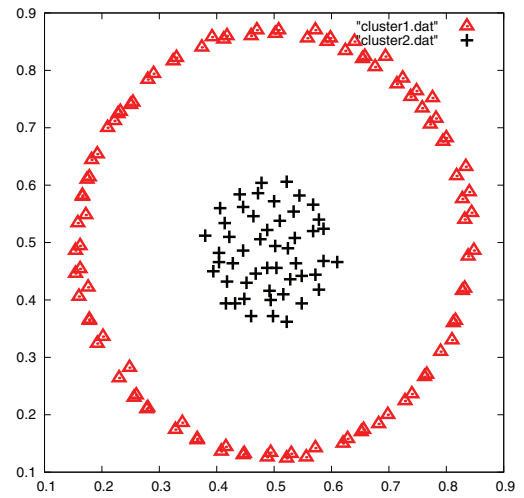


Fig. 3. The result of kernel-based FCM applied to data in Fig. 1.

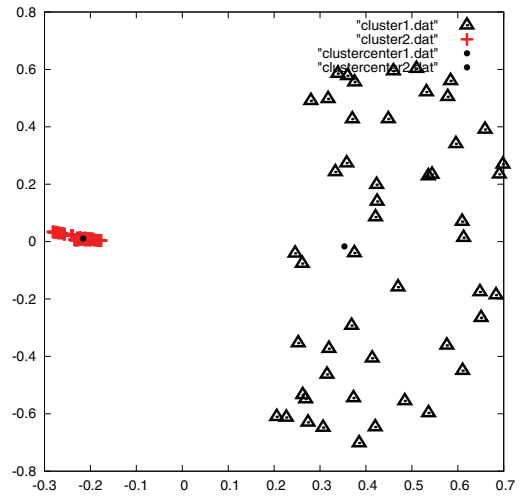


Fig. 4. The display of two major principal components to data in Fig. 1 with mapping $\Phi_2(x_k)$.

- 1) Using this mapping, we do not lose any fundamental information in kernel data analysis.
- 2) Generally the derivation becomes simpler using this mapping.
- 3) New applications of the kernel methods become possible using this mapping.

The last statement should be further studied. In relation to fuzzy clustering, the method of fuzzy c -varieties [1] should be considered. We also have many research possibilities related to SOM and LVQ [3].

Moreover we have other possibility to use nonlinear methods of data analysis such as LLE [6] to the transformed data, where the above explicit mapping is essential. Without such a mapping, we cannot apply a nonlinear tool of data analysis.

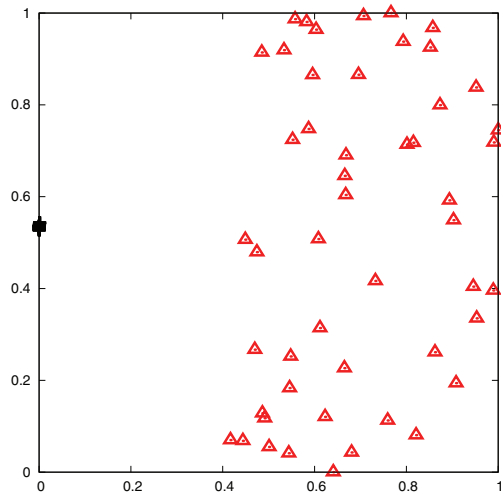


Fig. 5. The display of the low dimensional approximation to data in Fig. 1 with mapping $\Phi_2(x_k)$.

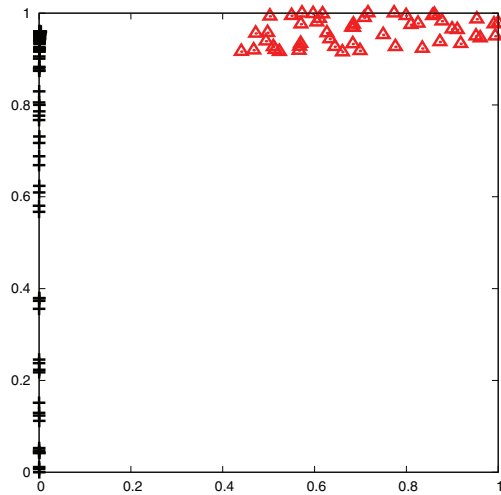


Fig. 6. The display of the low dimensional approximation to data in Fig. 1 with mapping $\Phi_2(x_k)$. L_1 -based FCM with L_1 -exponential kernel was applied.

ACKNOWLEDGMENT

This research has partly been supported by the Grant-in-Aid for Scientific Research, JSPS, Japan, No.19300074.

REFERENCES

- [1] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Functions, Plenum Press, New York, 1981.
- [2] N. Cristianini, J. Shawe-Talor, *An Introduction to Support Vector Machines*, Cambridge Univ. Press, 2000.
- [3] T. Kohonen, *Self-Organizing Maps*, 2nd Ed., Springer, Berlin, 1995.
- [4] S. Miyamoto, D. Suizu, Fuzzy c -means clustering using transformations into high dimensional spaces, Proc. of FSKD'02: 1st International Conference on Fuzzy Systems and Knowledge Discovery, Nov. 18-22, 2002, Singapore, Vol.2, pp. 656–660.
- [5] S. Miyamoto, H. Ichihashi, K. Honda, *Algorithms for Fuzzy Clustering*, Springer, Berlin, 2008.
- [6] S.T. Roweis, L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, Vol.290, No.22, pp.2323-2326, 2000.

- [7] B. Schölkopf, A.J. Smola, *Learning with Kernels*, the MIT Press, Cambridge, 2002.
- [8] V.N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.