Kernel Functions Derived from Fuzzy Clustering and Application to VQ Clustering

Jeong Sik Hwang
Graduate School of Systems and Information Engineering
University of Tsukuba
Tsukuba, Ibaraki 305–8573, Japan
Email: jeongsik@soft.risk.tsukuba.ac.jp

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

Abstract—Kernel functions are widely known to be useful tools in many areas of data analysis, while it has not yet shown that fuzzy models have close relation to kernel functions. We propose three basic functions derived from fuzzy c-means and possibilistic clustering. And then, we prove that the functions are able to be kernel functions, because those functions are positive-definite. The proof is based on the completely monotone function defined by Schöenberg. To verify that the present functions are useful to classify data sets with nonlinear cluster boundaries, we conduct vector quantization clustering based on the kernel functions. Moreover, we compare performances of kernel algorithms of vector quantization with different choices of the present kernel functions and their parameters.

I. INTRODUCTION

The goal of clustering is to group unlabeled data points into clusters, where data points within a cluster are more similar than data points in different clusters. We consider two clusters with nonlinear cluster boundary as Fig. 1 and Fig. 2 which we will discuss as examples herein. It has been known that traditional partitioning methods can not separate two cluster in this figure. A kernel function method to classify nonlinear data sets is now studied by many researchers in recent years. Among various types of kernel functions that have been studied so far, the Gaussian kernel(also RBF kernel) is popular and is said to have good performances overall [9], [10], [20].

We note that the Gaussian kernel is based on the Gaussian error function $e^{-t^2}$ which is used as the most standard probability density function. By analogy, we consider another function $\frac{1}{1+t^2}$ of a similar shape but with longer tails which is related to the Cauchy distribution. These two functions are typical symmetric functions for probability distributions on the line of real-values. Incidentally, the Gaussian function is a positive-definite function. It is natural for us to have a question that the other types of functions are also positive-definite functions. Another interesting property is that both types of functions are closely related to fuzzy c-means and possibilistic clustering [11], [12], [14], [15], [16], [21], as we will see in this paper.

Kohonen introduced the concept of self-organizing feature maps for visual display of certain one and two dimensional data sets and also started study of the prototype generation algorithm called learning vector quantization. The two methods are not clustering methods, however they often provide clustering algorithms [1], [3], [4], [18], which we will use in this paper.

The remainder of this paper is organized as follows. We first describe vector quantization clustering, kernel function and vector quantization clustering based on kernel function in Section 2. In section 3, we prove that three basic func-
tions defined in fuzzy clustering are positive-definite, and the proof is based on the completely monotone functions by Schönberg [22]. Moreover, regularization of squared Euclidean distance introduced by Ichihashi [21] is necessary to prove positive-definite property. Then in section 4, we show numerical results in which the derived functions are applied to kernel vector quantization clustering. It is shown that the non-Gaussian kernel functions discussed here outperform the Gaussian kernel in illustrative examples. Finally, we conclude this paper in Section 5.

II. VECTOR QUANTIZATION CLUSTERING AND KERNEL FUNCTION

A. VQ Clustering

The goal of vector quantization resides in determining a mapping \( \mathcal{M}: \mathbb{R}^m \rightarrow V \) from the \( m \)-dimensional Euclidean space \( \mathbb{R}^m \) to a finite subset \( V = \{ v_1, \ldots, v_c \} \). Each \( v_i \) is called a codevector and the set \( V \) is called a codebook. The vector quantizer is chosen to minimize the average distortion measure \( D(\mathbb{R}^m, V) = E[d(x, \mathcal{M}(x))] \) resulting from the approximation of the input \( x \in \mathbb{R}^m \) by the codevector \( \mathcal{M}(x) \), where \( d(\cdot) \) is usually taken as the standard squared Euclidean distance \( d(x, y) = \|x - y\|^2 \), and \( E(\cdot) \) denotes the expectation operator. The mapping of a vector quantizer \( \mathcal{M} \) determines a partition of the input space \( \mathbb{R}^m \) into \( c \) disjoint regions \( G_1, \ldots, G_c \), such that \( G_i = \{ x \in \mathbb{R}^m : \mathcal{M}(x) = v_i \} \) [5], [6], [7].

Furthermore, a learning algorithm for SOM was proposed by Kohonen. The learning algorithm can also be used for design vector quantizers. A significant difference of the VQ from the SOM is that the update is restricted to the winner of codebook vectors. VQ and SOM are not clustering algorithms in themselves, but application of them to clustering is similar to the hard \( c \)-means algorithm, thus we will call it VQ clustering.

\textbf{VQC (clustering by VQ) Algorithm}

\textbf{VQC1.} Set initial value for codevector \( v_i \) (e.g. Select \( c \) objects randomly as \( v_i, i = 1, \ldots, c \)).

Set \( t = 0 \) and Repeat from VQC2 to VQC4 until convergence.

\textbf{VQC2.} \ Select \( x(t) \in X \).

\textbf{VQC3.} \ Let \( v_i(t) = \arg \min_{1 \leq i \leq c} \| x(t) - v_i(t) \| \).

\textbf{VQC4.} \ Update \( v_1(t), \ldots, v_c(t) \):

\[ v_i(t + 1) = v_i(t) + \alpha[z(t) - v_i(t)], \]

\[ v_i(t + 1) = v_i(t), \quad i \neq l \]

Object represented by \( x(t) \) is allocated to \( G_l \).

\( t = t + 1 \).

\textbf{End VQC.}

In this algorithm, the parameter \( \alpha(t) \) should be taken to satisfy

\[ \sum_{t=1}^{\infty} \alpha(t) = \infty, \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \quad t = 1, 2, \ldots \]

B. Kernel-Based Clustering

The support vector machine with kernel functions conceptually implements the following idea: input data vectors are mapped to a high dimensional feature space. In this feature space, a linear cluster boundary is constructed. Such an SVM performs better than other classification algorithms for data sets having nonlinear cluster boundaries [8], [9].

A high dimensional feature space is denoted by \( H \) here which is called the feature space, whereas the original space \( \mathbb{R}^m \) is called the input data space. The feature space \( H \) is an infinite dimensional space. Its inner product is denoted by \( \langle \cdot, \cdot \rangle \); the norm of \( H \) is denoted by \( \| \cdot \|_H \). Notice that a function \( \Phi: \mathbb{R}^m \rightarrow H \) is employed and \( x \) is transformed into \( \Phi(x) \). An explicit representation of \( \Phi(x) \) is not usable, but the inner product \( \langle \Phi(x), \Phi(y) \rangle \) is expressed by a kernel function:

\[ K(x, y) = \langle \Phi(x), \Phi(y) \rangle, \]

where a function \( K \) is called a positive-definite kernel [23], [24]. Thus, we are capable of analyzing objects in the feature space \( H \) instead of the original space \( \mathbb{R}^m \).

\textbf{C. Kernel Vector Quantization Clustering}

Among various discussions of SOM and LVQ based on kernel function have been found [3], [4], [7], [9]. But we consider kernel VQ clustering proposed by Inokuchi et al [2].

Instead of the distance \( \| x(t) - v_i(t) \| \) in the input data space, the next distance in the high dimensional space is considered as below.

\[ d_{ik} = \| \Phi(x_k) - v_i(t) \|^2 \]

The updating equation of the codevector is as follows:

\[ v_i(t + 1) = v_i(t) + \alpha \Phi(x_k) - v_i(t) \],

where it should be noted that \( x_k \) is randomly selected input vector from \( X \). \( v_i \) is the codevector in the high dimensional space. The distance \( d_{ik}(t + 1) \) in the high dimensional space is calculated substituting Eq. (2) into Eq. (1). Thus, distance \( d_{ik}(t + 1) \) in the high dimensional space is given by the following expression.

\[ d_{ik}(t + 1) = (1 - \alpha) d_{ik}(t) - \alpha (1 - \alpha) d_{ki}(t) + \alpha (K_{kk} - 2 K_{kh} + K_{hh}), \]

where \( K_{kk} = K(x_k, x_k) \). The update of the high dimensional codevectors can be made indirectly. It is not necessary to keep the transformed codevectors which play the same role of the codevectors in the input space.

We do not know a concrete form of \( \Phi(x_k) \) and \( v_i \) explicitly. Nevertheless, we are able to know the distance between input data and codevector by using the kernel function.

We now have the KVQ clustering algorithm:

\textbf{KVQC (Kernel Vector Quantization Clustering) Algorithm.}

\textbf{KVQC1.} Fix \( c \), initialize \( d_{ik}, i = 1, \ldots, c, k = 1, \ldots, n \), learning rate \( \alpha \in (1, 0) \).
Set \( t = 0 \) and Repeat from KVQc2 to KVQc4 until convergence.

KVQc2. Select \( x(t) \in X \).

KVQc3. For \( k = 1, \ldots, n \)

a. \( d_{ik}(t) = \arg \min_{1 \leq i \leq c} d_{ik}(t) \)

b. Updating \( d_{ik}(t) \) by Eq. (3)

Allocate \( x_k \) to cluster \( l \).

KVQc4. Adjust learning rate \( \alpha \). \( t = t + 1 \)

End KVQc.

III. Kernel Functions Derived from Fuzzy Clustering

A. Fuzzy c-means Clustering

We begin with the review of fuzzy c-means and possibilistic clustering. Let objects to be clustered \( x_k = (x_{k1}, \ldots, x_{kp}) \in \mathbb{R}^p, k = 1, \ldots, N \), a vector in the \( p \)-dimensional Euclidean space. Cluster centers are \( v_i = (v_{i1}, \ldots, v_{ip})^T, i = 1, \ldots, c \), where \( c \) is the number of clusters. An abbreviated symbol \( V = (v_1, \ldots, v_c) \) is used for the whole collection of cluster centers. The membership matrix \( U = (u_{ki}), (i = 1, \ldots, c; k = 1, \ldots, N) \) is used, where \( u_{ki} \) means the degree of belongingness of object \( x_k \) to cluster \( i \). In fuzzy c-means clustering, the constraint for a fuzzy partition is:

\[
M_f = \{ U = (u_{ki}) : \sum_{i=1}^c u_{ki} = 1, \forall k; u_{kj} \geq 0, \forall k, j \}
\]

The dissimilarity for clustering is the standard squared Euclidean distance between an individual and a cluster center:

\[
D(x_k, v_i) = \| x_k - v_i \|^2
\]

It is well known that fuzzy c-means clustering is based on the optimization of an objective function. We consider three different types of objective function.

\[
J_1(U, V) = \sum_{k=1}^N \sum_{i=1}^c (u_{ki})^\ell (\epsilon + D(x_k, v_i)) \quad (\epsilon > 0)
\]

\[
J_2(U, V) = \sum_{k=1}^N \sum_{i=1}^c u_{ki} D(x_k, v_i) + \lambda^{-1} u_{ki} (\log u_{ki} - 1)
\]

\[
J_3(U, V) = \sum_{k=1}^N \sum_{i=1}^c (u_{ki})^\ell D(x_k, v_i) + \sum_{i=1}^c \eta_i \sum_{k=1}^N (1 - u_{ki})^m
\]

\( J_1 \) is similar to the well known objective function proposed by Bezdek [11] and Dunn [16], but a positive parameter \( \epsilon \) is added. This objective function has been proposed by Ichihashi [21]. When \( \epsilon \) approaches zero, \( J_1 \) approaches to the standard objective function, and it is also known that the solution converges to the standard solution.

\( J_2 \) is an entropy-based method proposed by several authors (e.g. [17], [18], [19]). \( J_3 \) is the objective function for the possibilistic clustering [14], [15], [21].

B. Three Basic Functions

Let us define basic functions \( F_{\ell}(x, y) \) (\( \ell = 1, \ldots, 3 \)) for the respective objective functions.

Basic functions for fuzzy c-means and possibilistic clustering

\[
F_1(x, y) = \frac{1}{(\epsilon + \|x - y\|^2)^{\frac{1}{\ell - 1}}}
\]

\[
F_2(x, y) = \exp(-\lambda \|x - y\|^2)
\]

\[
F_3(x, y) = \frac{1}{1 + (\|x - y\|^2/\eta_i)^{\frac{1}{\ell - 1}}}
\]

By using the ratio of basic functions, \( u_{ki}^{(\ell)}(\ell = 1, 2) \) are represented as

\[
u_{ki}^{(\ell)} = \frac{F_{\ell}(x_k; v_i)}{\sum_{j=1}^c F_{\ell}(x_k; v_j)}
\]

Note that \( J_3 \) cannot be used for fuzzy c-means clustering.

On the other hand, solutions of cluster centers are as follows.

\[
v_i = \frac{\sum_{k=1}^N (u_{ki})^m x_k}{\sum_{k=1}^N (u_{ki})^m}
\]

where \( m = 1 \) for \( J_2 \).

In possibilistic clustering, the constraint \( M_f \) of fuzzy c-means is not imposed on \( U \). Rather, we assume

\[
M = (\mathbb{R}^p)^n
\]

C. Kernels and Completely Monotone Functions

Recently positive-definite kernel functions and their application to data analysis have been studied by many researchers, e.g., [10], [13], [23], [24].

It has also been known that completely monotone functions define a class of positive-definite kernels.

We introduce a function \( f : [0, +\infty) \rightarrow \mathbb{R} \) and note the next definition.
Definition 1. (Schönberg [22])
A function \( f: [0, +\infty) \rightarrow \mathbb{R} \) is said to be completely monotone if \( f^{(2n)}(t) \geq 0 \) and \( f^{(2n-1)}(t) \leq 0 \) \( (n = 1, 2, 3, \ldots) \) for \( t > 0 \) and \( f(0) = f(+0) \).

The following theorem has been proved by Schönberg.

Theorem 1. (Schönberg [22])
Let \( H \) be a Hilbert space with norm \( \| \cdot \|_H \). The function

\[
K(x, y) = f(\|x - y\|_H)
\]

is positive-definite if and only if \( f(t) \) is completely monotone.

It is easily seen that the Gaussian kernel is associated with the function \( f(t) = e^{-t} \), which is obviously completely monotone. We thus know basic function \( F_2(x, y) = \exp(-\lambda\|x - y\|^2) \) is positive-definite.

We consider whether the other basic functions

\[
F_1(x, y) = \frac{1}{(c + \|x - y\|^2)^{\frac{1}{m-1}}}
\]

and

\[
F_3(x, y) = \frac{1}{1 + (\|x - y\|^2/\eta)^{\frac{1}{m-1}}}
\]

are positive-definite functions or not.

We first observe that the next lemma holds.

Lemma 1. Let \( g: [0, +\infty) \rightarrow \mathbb{R} \) be

\[
g(t; \gamma, \delta) = (t + c)^{-\gamma} t^{-\delta},
\]

where \( 0 < b < 1 \) and \( c > 0 \) are constants, while \( \gamma \) and \( \delta \) are positive parameters. Notice that \( g(t; \gamma, \delta) > 0 \). Then

\[
g'(t; \gamma, \delta) = -\gamma bg(t; \gamma + 1, \delta + 1 - b) - \delta g(t; \gamma, \delta + 1)
\]

In particular, we have \( g'(t; \gamma, \delta) < 0 \).

The proof is straightforward and is omitted. We then have the following proposition.

Proposition 1. A function \( f: [0, \infty) \rightarrow \mathbb{R} \)

\[
f(t) = (t^b + c)^{-a},
\]

where \( a > 0, 0 < b < 1 \) and \( c > 0 \) are constants, is completely monotone.

Proof: We apply Lemma 1 repeatedly. Since higher order derivatives of \( f(t) \) have terms of the form \( g(t; \gamma, \delta) \) multiplied by constants, and the sign of constants changes with differentiation, we have \( f^{(2n)}(t) > 0 \) and \( f^{(2n-1)}(t) < 0 \) for all positive \( n \). Since \( \lim_{t \to +0} f(t) = e^{-a} = f(0) \), the proposition is proved.

We hence have the next proposition.

Proposition 2. The three basic functions \( F_\ell(x, y) \) \( (\ell = 1, 2, 3) \) are all positive definite.

Proof: The above form (14) includes

\[
f_1(t) = (t + c)^{-\frac{1}{m-1}}
\]

and

\[
f_3(t) = (t^{-\frac{1}{m-1}} + 1)^{-1}.
\]

We hence have the desired conclusion.

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>THREE BASIC FUNCTIONS ARE POSITIVE-DEFINITE KERNELS</td>
</tr>
<tr>
<td>( F_1 )-Kernel</td>
</tr>
<tr>
<td>( F_2 )-Kernel</td>
</tr>
<tr>
<td>( F_3 )-Kernel</td>
</tr>
</tbody>
</table>

IV. APPLICATION TO VQ CLUSTERING

We have found that the three basic functions in fuzzy clustering are positive-definite kernels. In this section, we consider an application of three basic functions to kernel VQ clustering discussed by Inokuchi [2] (see also [3],[4]). We show numerical results of non-kernel VQ clustering and kernel VQ clustering using the basic functions. Moreover we compare the results from kernel functions \( F_1(x, y) \) and \( F_3(x, y) \) with those from the Gaussian kernel \( F_2(x, y) \).

A. Numerical Examples

In the numerical examples, we demonstrate the usefulness of the present kernel functions for synthetic data sets and several real-world data sets.

We used two kinds of synthetic data sets.

- A ball in a circle: 100 data points from two classes, which is comprised of 40 points in an inner ball and 60 points in a surrounding outer circle as Fig. 1.
- Two crescents: 100 data points from two classes like a crescent as Fig. 2.

We also report results for several real-world data sets, which is taken from UCI machine learning repository [25].
Iris data sets: Iris data sets are used as a test for clustering algorithms. It contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two classes but the latter classes are not linearly separable from each other.

Wisconsin Breast Cancer data sets: 683 instances from two classes which is comprised of 444 benign instances and 239 malignant instances. We removed 16 instances with missing attribute values in original data sets. Each instance has 11 attributes including an ID number.

Glass Identification data sets: 214 instances from two classes which is comprised of 163 window glass (building windows and vehicle windows) and 51 non-window glass. Each instance has 10 attributes including an ID number.

B. Synthetic Data Sets

The results of KVQ clustering using the present kernel functions are shown in Fig. 3 and Fig. 4. Figure 3 shows the data sets classified perfectly into the ball and the circle. It is well known that an ordinary fuzzy and hard c-means and VQ clustering cannot divide the whole set into such clusters, because the classification boundary is nonlinear. It is also known that the result from a kernel function \( F_2(x, y) \) (Gaussian kernel) can be perfect as shown. It has been confirmed that the other kernel functions \( F_1(x, y) \) and \( F_3(x, y) \) can also divide these data sets. Two crescents are also classified, producing results with errors (see Fig. 4). Generally it is more difficult to divide this data sets into two clusters of crescents. KVQ clustering using kernel functions \( F_2(x, y) \) and \( F_3(x, y) \) can not divide this data sets at all.

We calculated the rate of good classifications and the running time in each time, where we considered that the ratio of error under 10% is also a good classification. We presented the results of KVQ clustering using the present kernel functions for the synthetic data sets in Table II.

C. Real-World Data Sets

For VQ clustering, setosa (class 1) of IRIS data sets is classified from versicolor (class 2) and virginica (class 3).

![Fig. 4. Results of KVQ clustering using function \( F_1(x, y) \)](image)

![Fig. 5. VQ clustering for Iris data sets](image)

![Table II](image)

However it is difficult to classify versicolor and virginica, since those classes can not be linearly separated from each other as Fig. 5.

The kernel functions discussed above work on classification for data sets with nonlinear boundary. It has been observed that the performance of these kernel functions depends on their parameters: \( F_1 \) and \( F_3 \) kernel functions are characterized by two parameter. For \( F_1 \) kernel function, when \( M \) is greater, all data points are assigned to the same cluster. If \( \epsilon \) is smaller, data points are assigned sparsely as Fig. 6. On the other hand, If \( \eta \) is greater, all data points are assigned to one cluster. For \( F_3 \) kernel function, when \( M \) is smaller, they are assigned sparsely as Fig. 8. By using a proper value of their parameters, however, Iris data sets are well-classified as Fig. 7 and Fig. 9.

The ratio of the number of good classification to the number of all trial is expressed as a percentage in Table III. In this way, the performance of kernel functions depends on strongly parameters. KVQ clustering based on kernel \( F_1(x, y) \) and \( F_3(x, y) \) have frequently shown good classification and fast running time on synthetic datasets. Likewise on real-world data sets, it have been observed that data sets are frequently well-classified and faster than the Gaussian kernel.

V. CONCLUSION

We have discussed the three basic functions that arise in fuzzy c-means clustering and possibilistic clustering as positive definite kernels, using the completely monotone properties by Schönberg [22]. This fact itself is useful in the sense that we have a larger class of kernel functions. Moreover we have shown that the non-Gaussian kernel functions work well in typical clustering examples of nonlinear cluster boundaries.
They work better than the Gaussian kernel in the numerical examples.

In the future, we will further investigate the usefulness of the present kernel functions in variety of real-world data sets.

REFERENCES


[11] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algo-


