IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 20, 2019, accepted April 3, 2019, date of publication April 9, 2019, date of current version April 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909752

BP*k*NN: *k*-Nearest Neighbor Classifier With Pairwise Distance Metrics and Belief Function Theory

LIANMENG JIAO, (Member, IEEE), XIAOJIAO GENG, AND QUAN PAN, (Member, IEEE)
School of Automation, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Xiaojiao Geng (xiaojiaogeng@mail.nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61790552 and Grant 61801386, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JQ6043, and in part by the Aerospace Science and Technology Foundation of China.

ABSTRACT The k-nearest neighbor (kNN) rule is one of the most popular classification algorithms in pattern recognition field because it is very simple to understand but works quite well in practice. However, the performance of the kNN rule depends critically on its being given a good distance metric over the input space, especially in small data set situations. In this paper, a new kNN-based classifier, called BPkNN, is developed based on pairwise distance metrics and belief function theory. The idea of the proposal is that instead of learning a global distance metric, we first decompose it into learning a group of pairwise distance metrics. Then, based on each learned pairwise distance metric, a pairwise kNN (PkNN) sub-classifier can be adaptively designed to separate two classes. Finally, a polychotomous classification problem is solved by combining the outputs of these PkNN sub-classifiers in belief function framework. The BPkNN classifier improves the classification performance thanks to the new distance metrics which provide more flexibility to design the feature weights and the belief function-based combination method which can better address the uncertainty involved in the outputs of the sub-classifiers. Experimental results based on synthetic and real data sets show that the proposed BPkNN can achieve better classification accuracy in comparison with some state-of-the-art methods.

INDEX TERMS Pattern classification, k-nearest-neighbor classifier, pairwise distance metric, belief function theory.

I. INTRODUCTION

Automatic classification of patterns is an important problem in a variety of engineering and scientific disciplines such as biology [1], psychology [2], medicine [3], marketing [4], military affairs [5], etc. Generally, complete statistical knowledge regarding the conditional density of each class is rarely available, which precludes applications of the optimal Bayes classification procedure [6]. In these cases, a good solution is to classify each new pattern using the evidence of nearby sample observations. One such non-parametric procedure has been introduced by Fix and Hodges [7], and has since become well-known in the pattern classification community as the k-nearest neighbor (kNN) rule. Cover and Hart [8] also provided a statistical justification of this procedure by showing that, as both the number of samples N and the number of neighbors k tend to infinity in such a way that $k/N \rightarrow 0$, the error rate of the kNN rule approaches the optimal Bayes error rate. However, in the case of finite number of samples, the classical k-NN rule is not guaranteed to be the optimal way of using the information contained in the neighborhood of query patterns.

In many practical pattern classification applications, the available data may be insufficient, and the real class-conditional probability distributions cannot be well characterized using the limited training samples [9], [10]. In such small data set situations (relative to the intrinsic dimensionality of the data involved), the ideal asymptotical behavior of the kNN rule degrades dramatically [11]. This is the reason why the improvement of this rule has remained an active research

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Li.

topic in the past 60 years. One of the directions is to find more adequate distance metrics that potentially improve the *k*NN classification performance in small data set situations.

The performance of the *k*NN rule depends critically on its being given a good distance metric over the input space, especially in small data set situations. To overcome the limitations of the original Euclidean (L2) distance metric, many adaptive distance metrics and learning methods have been proposed. According to the structure of the metric, these methods can be mainly divided into two categories: global distance metric learning [12]–[17], and local distance metric learning [18]–[23]. The main drawback of the global learning approach is that the learned single distance metric usually cannot separate all of the class pairs well. As one representative label-based local distance metric learning method, Paredes and Vidal [22], [23] proposed to learn a class-dependent weighted (CDW) distance metric adaptively for each class. However, as the learned CDW distance metric is only relevant to the class labels of the training samples, it is insufficient to reflect the local specificities in feature space for query patterns in different classes.

In this paper, we focus on the label-based local distance metric learning problem [1]. To overcome the limitations of the CDW distance metric, two types of pairwise distance metrics are defined, which can better characterize the local specificities in the input space. For a general polychotomous classification problem, a new *k*NN-based classifier, called BP*k*NN, is develop based on the pairwise distance metrics and the belief function theory [26], [27]. The idea is that instead of learning a global distance metric, we first decompose it into learning a group of pairwise distance metrics. Because only two classes are involved for each pairwise distance metric, the feature weights can be learned in a more local way. Then, based on each learned pairwise distance metric, a pairwise *k*NN (P*k*NN) sub-classifier can be adaptively designed to separate two classes. Finally, a polychotomous classification problem is solved by combining the uncertain outputs of the P*k*NN sub-classifiers in belief function framework. The main contributions of this paper are as follows.

1) A pairwise weighted (PW) distance metric related to the labels of the class pair to be classified is defined, and the corresponding parameter optimization procedure is designed based on the maximum likelihood principle. As a more general dissimilarity measure, the new PW distance metric can provide greater flexibility to design the feature weights.
2) The PW distance metric is further extended to the pairwise Mahalanobis (PM) distance metric, which can effectively address the potential feature correlations existed in many real-world applications.
3) A new BP*k*NN classifier is designed based on the pairwise distance metrics in belief function framework, which can better model and combine the uncertainty involved in the outputs of P*k*NN sub-classifiers.

Two types of experiments using both synthetic and real data sets have been developed to evaluate the performance of the proposed BP*k*NN classifier. In the synthetic data test, we employed different data generation distributions with different degrees of class overlapping. For all the cases, the proposed BP*k*NN classifier produced the highest classification accuracy, which demonstrated the generalizability of the proposed method on synthetic difficult data. In the real data test, twenty data sets varying greatly in the number of instances, features and classes, were selected from the UCI Machine Learning Repository [28] for evaluation. The comparison methods include those *k*NN classifiers based on other representative distance metrics and some of the most popular instance-based methods. The results reported show the competitive performance of the proposal for a variety of real tasks involving different data conditions.

The rest of this paper is organized as follows. Section II briefly reviews related background. After that, two types of pairwise distance metrics are defined and learned in Section III. The BP*k*NN classifier is then designed and realized based on the proposed pairwise distance metrics and the belief function theory in Section IV. The experiments to evaluate the performance of the proposed method are reported in Section V. Finally, Section VI concludes the paper.

II. BACKGROUND

This section briefly reviews the most important developments related to the distance metric learning in *k*NN classifiers as well as some other instance-based classifiers.

In the original *k*-NN rule [7], the Euclidean (L2) distance metric is used to compute the distance between two patterns as

$$d_{L2}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{P} (x_j - y_j)^2}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^P$ is a pattern in the training set, and $\mathbf{y} \in \mathbb{R}^P$ is a query pattern to be classified.

The main drawback of the L2 distance metric is that it does not take into account the weights of different features in the input space. To overcome this limitation, a number of distance metric learning methods have been proposed.

The first group of these methods learn the distance metric in a global sense, i.e., the same global weighted (GW) distance metric is defined for all the patterns by introducing feature weights as

$$d_{GW}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{P} \lambda_j^2 (x_j - y_j)^2}, \tag{2}$$

where λ_j is the weight of the j-th feature.

Based on the this GW distance metric, the feature weights learning in [12]–[14] was formulated as a linear programming

[1] A preliminary version of some of the ideas introduced here was presented in [24], [25]. The present paper is a deeply revised and extended version of this work, with several new results.

problem that minimizes the distance between the data pairs within the same classes subject to the constraint that the data pairs in different classes are well separated. Eick *et al.* [15] introduced an approach to learn the feature weights that maximizes the clustering accuracy of objects in the training set, and similarly, the classification error rate of objects in the training set was minimized to learn the feature weights in [16], [17].

Although the above global distance metric learning methods are intuitively appealing, they are still not fine enough, as the feature weights of the distance metric are irrelevant with the class labels of the patterns. This issue becomes more severe when some features behave distinctly for different classes (e.g., one feature may be more discriminative for some classes, but irrelevant for others). For classification problems with a large number of classes, it is hard to learn a GW distance metric that can simultaneously separate all of the class pairs with well performance.

Therefore, the local distance metric learning approach was developed to learn a local distance metric for some specific patterns. According to the types of used local information, this approach can be further divided into two subcategories: geometry-based local distance metric learning and label-based local distance metric learning.

For the geometry-based methods [18]–[21], the aim is to learn a locally adaptive distance metric in the neighborhood of each query pattern. Wang *et al.* [21] proposed an adaptive *k*NN algorithm which involves a locally adaptive distance measure by normalizing the ordinary Euclidean distance from a query pattern to each training sample by the shortest distance between the corresponding training sample to training samples of a different class.

In contrast, Paredes and Vidal [22], [23] provided an idea to learn a label-based local distance metric. In their work, a class-dependent weighted (CDW) distance metric which is relevant to the class labels of the training samples was defined as

$$d_{CDW}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{P} {\lambda_c^j}^2 (x_j - y_j)^2}, \qquad (3)$$

where, λ_c^j is the weight of the j-th feature and c is the class index of training sample $\mathbf{x}$. The involved feature weights can be optimized by minimizing the leaving-one-out classification error of the given training set.

The CDW distance metric has become recently popular in *k*NN classifiers as well as some other instance-based classifiers. For example, in [29], a weighted data gravitation classification (DGC+) method was developed by employing a matrix of weights to describe the importance of each attribute in the classification of each class, which was used to weight the distance between data samples.

As observed, the CDW distance metric provides more freedom than the GW distance metric and can be learned adaptively for different classes of the training samples. However, as the learned CDW distance metric is only relevant to the class labels of the training samples, it is insufficient to characterize the local specificities in the input space for query patterns in different classes. This problem is enhanced as the numbers of features and classes increase.

III. PAIRWISE DISTANCE METRIC: DEFINITION AND LEARNING

To better characterize the local specificities in the input space, in Section III-A, we define a pairwise weighted distance metric and design a parameter optimization procedure to learn it based on the maximum likelihood principle. Then, in Section III-B, we extend the pairwise weighted distance metric to further consider the potential correlations between different features.

A. PAIRWISE WEIGHTED DISTANCE METRIC

Definition 1 (Pairwise weighted distance metric): Suppose $\mathbf{x}$ and $\mathbf{y}$ are two P-dimensional patterns whose labels belong to class pair $\{\omega_p, \omega_q\} \triangleq \Omega_{p,q}$. The pairwise weighted (PW) distance metric between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d_{PW}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{P} {\lambda_{p,q}^j}^2 (x_j - y_j)^2}, \qquad (4)$$

where $\lambda_{p,q}^j$ is a constant that weights the role of the j-th feature in the distance metric concerning class pair $\Omega_{p,q}$.

This definition includes, as particular cases, the distance metrics reviewed in the previous section. If $\lambda_{p,q}^j = 1$ for all $p = 1, \cdots, M$, $q = 1, \cdots, M$, $j = 1, \cdots, P$, the above defined PW distance metric reduces to the L2 distance metric. In addition, the GW and CDW distance metrics correspond to the cases where the metric weights do not depend on the class labels or are only dependent on the class label of the first pattern, respectively. Therefore, the PW distance metric is a more general dissimilarity measure than the L2, GW or CDW distance metrics and can provide greater flexibility to design the feature weights so that the local specificities in the input space can be well characterized.

In the above defined PW distance metric, the only free parameters are the feature weights related to the labels of the two considered classes. In the following part, we aim to learn feature weights $\lambda_{p,q}^j$ $(1 \leq p < q \leq M, j = 1, \cdots, P)$ from the training data by optimizing some criteria. A simple way of defining the criteria for the desired metric is to keep the data pairs from the same class close to each other while separating those data pairs from different classes far from each other [12].

We divide training set $\mathcal{T}$ into M subsets $\mathcal{T}_k, k = 1, \cdots, M$, with each $\mathcal{T}_k$ containing all of the N_k training data belonging to class ω_k:

$$\mathcal{T}_k = \{(\mathbf{x}_i, \omega_k) \mid i \in I_k\},$$

where I_k is the set of indices for training data $\mathbf{x}_i$ belonging to class ω_k.

We now consider learning feature weights $\lambda_{p,q}^j$ ($j = 1, \cdots, P$) from training subsets $\mathcal{T}_p$ and $\mathcal{T}_q$. Let us denote the set of data pairs from the same class as

$$\mathcal{S} = \left\{(\mathbf{x}_m, \mathbf{x}_n) \mid m, n \in I_p; m < n\right\} \cup \left\{(\mathbf{x}_m, \mathbf{x}_n) \mid m, n \in I_q; m < n\right\},$$

and the set of data pairs from different classes as

$$\mathcal{D} = \left\{(\mathbf{x}_m, \mathbf{x}_n) \mid m \in I_p; n \in I_q\right\}.$$

In order to estimate the probability for any data pair $(\mathbf{x}_m, \mathbf{x}_n)$ to share the same class or different classes, the logistic regression model is used here. The reasons for choosing this model are as follows. Firstly, The logistic function in this model can ensure that the predicted values are probabilities and are therefore restricted to (0, 1). Furthermore, because the logistic regression model predicts probabilities, rather than just classes, the parameters of this model can be easily fitted using the maximum likelihood estimation.

With the logistic regression model, the probability Pr for any data pair $(\mathbf{x}_m, \mathbf{x}_n)$ to share the same class is

$$\Pr(+ \mid (\mathbf{x}_m, \mathbf{x}_n)) = \frac{1}{1 + e^{d_{PW}^2(\mathbf{x}_m,\mathbf{x}_n)-\mu_{p,q}}}, \tag{5}$$

and then the probability Pr for any data pair $(\mathbf{x}_m, \mathbf{x}_n)$ to share different classes is

$$\begin{aligned}\Pr(- \mid (\mathbf{x}_m, \mathbf{x}_n)) &= 1 - \frac{1}{1 + e^{d_{PW}^2(\mathbf{x}_m,\mathbf{x}_n)-\mu_{p,q}}} \\ &= \frac{1}{1 + e^{\mu_{p,q}-d_{PW}^2(\mathbf{x}_m,\mathbf{x}_n)}},\end{aligned} \tag{6}$$

where "$+$" and "$-$" denote data pair $(\mathbf{x}_m, \mathbf{x}_n)$ belonging to the same class and different classes, respectively. Parameter $\mu_{p,q}$ is the threshold. The data pair $(\mathbf{x}_m, \mathbf{x}_n)$ will be assigned higher probability to be in the same class when their square PW distance is much smaller than threshold $\mu_{p,q}$. In contrast, if their square PW distance is much larger than threshold $\mu_{p,q}$, they will be given more probability to have different classes.

Then, the overall log-likelihood for both the data pairs in $\mathcal{S}$ and $\mathcal{D}$ can be written as

$$\begin{aligned}&\mathcal{L}_g\left(\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}\right) \\ &= \log \Pr(+ \mid \mathcal{S}) + \log \Pr(- \mid \mathcal{D}) \\ &= -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + e^{d_{PW}^2(\mathbf{x}_m,\mathbf{x}_n)-\mu_{p,q}}\right) \\ &\quad - \sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + e^{\mu_{p,q}-d_{PW}^2(\mathbf{x}_m,\mathbf{x}_n)}\right) \\ &= -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + e^{\sum_{j=1}^{P} {\lambda_{p,q}^j}^2 (x_{mj}-x_{nj})^2-\mu_{p,q}}\right) \\ &\quad - \sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + e^{\mu_{p,q}-\sum_{j=1}^{P} {\lambda_{p,q}^j}^2 (x_{mj}-x_{nj})^2}\right).\end{aligned} \tag{7}$$

With the maximum likelihood principle, the PW distance metric learning can be formulated as the following optimization problem:

$$\begin{aligned}&\max_{\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}} \mathcal{L}_g\left(\{\lambda_{p,q}^j\}_{j=1}^P, \mu_{p,q}\right) \\ &\text{s.t. } \lambda_{p,q}^j \geq 0,\ j = 1, \cdots, P, \text{ and } \mu_{p,q} \geq 0.\end{aligned} \tag{8}$$

This is a convex programming problem, which can be solved using Newton's method [30].

B. EXTENSION TO PAIRWISE MAHALANOBIS DISTANCE METRIC

In the above subsection, the distance metric was learned under the assumption that the P considered features are independent. However, in many real-world applications, this assumption is hardly tenable [31]. Therefore, in this subsection, we extend the PW distance metric to further consider the correlations between different features.

Definition 2 (Pairwise Mahalanobis distance metric): Suppose $\mathbf{x}$ and $\mathbf{y}$ are two P-dimensional patterns whose labels belong to class pair $\Omega_{p,q}$. The pairwise Mahalanobis (PM) distance metric between $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d_{PM}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{A}_{p,q}(\mathbf{x} - \mathbf{y})}, \tag{9}$$

where $\mathbf{A}_{p,q} \in \mathbf{R}^{P\times P}$ is a positive semi-definite matrix (i.e., $\mathbf{A}_{p,q} \succeq 0$) that weights the role of features in the distance metric concerning class pair $\Omega_{p,q}$.

This definition provides a more generalized pairwise distance metric by introducing the potential correlations between different features. If we restrict $\mathbf{A}_{p,q}$ to be diagonal, the defined PM distance metric just reduces to the PW distance metric in Definition 1.

In a similar way as described in the above subsection, the PM distance metric learning can also be formulated as a nonlinear optimization problem. However, in the case of learning a full matrix $\mathbf{A}_{p,q}$, the positive semi-definite constraint of $\mathbf{A}_{p,q}$ becomes difficult to enforce, and Newton's method often becomes prohibitively expensive (requiring $O(P^6)$ time to invert the Hessian over P^2 parameters). To simplify the computation, we model the matrix $\mathbf{A}_{p,q}$ using the eigenspace of the training samples [32]. Based on training subsets $\mathcal{T}_p$ and $\mathcal{T}_q$, the covariance matrix between any two features is computed as

$$\mathbf{M}_{p,q} = \frac{1}{n_{p,q} - 1} \sum_{i\in I_{p,q}} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \tag{10}$$

where, $I_{p,q}$ is the set of indices for training sample $\mathbf{x}_i$ belonging to class ω_p or ω_q, $n_{p,q}$ is the number of training samples, and $\bar{\mathbf{x}}$ is the mean feature vector over the $n_{p,q}$ training samples. Denoting $\mathbf{v}_{p,q}^k$, $k = 1, 2, \cdots, K$, the top K ($K \leq P$) eigenvectors of matrix $\mathbf{M}_{p,q}$, we model $\mathbf{A}_{p,q}$ as a linear combination of the top K eigenvectors as

$$\mathbf{A}_{p,q} = \sum_{k=1}^{K} \gamma_{p,q}^k \mathbf{v}_{p,q}^k {\mathbf{v}_{p,q}^k}^T, \tag{11}$$

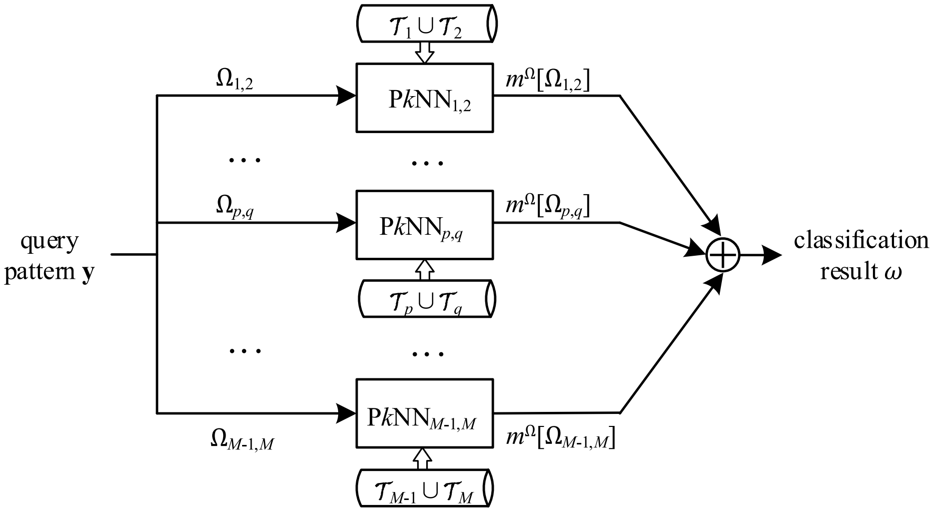

$\mathcal{T}_1 \cup \mathcal{T}_2$
$\Omega_{1,2}$
PkNN$_{1,2}$
$m^{\Omega}[\Omega_{1,2}]$
$\Omega_{p,q}$
PkNN$_{p,q}$
$m^{\Omega}[\Omega_{p,q}]$
query pattern $\mathbf{y}$
classification result ω
$\mathcal{T}_p \cup \mathcal{T}_q$
PkNN$_{M\text{-}1,M}$
$\Omega_{M\text{-}1,M}$
$m^{\Omega}[\Omega_{M\text{-}1,M}]$
$\mathcal{T}_{M\text{-}1} \cup \mathcal{T}_M$

FIGURE 1. Scheme of kNN classification with pairwise distance metrics and belief function theory.

where $\gamma_{p,q}^k$, $k = 1, 2, \cdots, K$, are the non-negative weights for linear combination.

Then, with the above decomposition of matrix $\mathbf{A}_{p,q}$, the overall log-likelihood for both the data pairs in $\mathcal{S}$ and $\mathcal{D}$ can be written as

$$\begin{aligned}
&\mathcal{L}_g\left(\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}\right)\\
&\quad = \log \Pr(+ \mid \mathcal{S}) + \log \Pr(- \mid \mathcal{D})\\
&\quad = -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + e^{d_{PM}^2(\mathbf{x}_m,\mathbf{x}_n)-\mu_{p,q}}\right)\\
&\qquad - \sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + e^{\mu_{p,q}-d_{PM}^2(\mathbf{x}_m,\mathbf{x}_n)}\right)\\
&\quad = -\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{S}} \log\left(1 + e^{\sum_{k=1}^{K}\gamma_{p,q}^k \nu_{m,n}^k - \mu_{p,q}}\right)\\
&\qquad - \sum_{(\mathbf{x}_m,\mathbf{x}_n)\in\mathcal{D}} \log\left(1 + e^{\mu_{p,q}-\sum_{k=1}^{K}\gamma_{p,q}^k \nu_{m,n}^k}\right), \qquad (12)
\end{aligned}$$

with $\nu_{m,n}^k = (\mathbf{x}_m - \mathbf{x}_n)^T \mathbf{v}_{p,q}^k {\mathbf{v}_{p,q}^k}^T (\mathbf{x}_m - \mathbf{x}_n)$.

With the maximum likelihood principle, the PM distance metric learning can be formulated as the following optimization problem:

$$\begin{aligned}
&\max_{\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}} \quad \mathcal{L}_g\left(\{\gamma_{p,q}^k\}_{k=1}^K, \mu_{p,q}\right)\\
&\text{s.t. } \gamma_{p,q}^k \geq 0,\ k = 1, \cdots, K, \text{ and } \mu_{p,q} \geq 0, \qquad (13)
\end{aligned}$$

which is similar to the optimization of the PW distance metric learning, and can be solved using the same optimization method.

IV. KNN CLASSIFICATION WITH PAIRWISE DISTANCE METRICS AND BELIEF FUNCTION THEORY

With the proposed pairwise distance metrics concerning class pair $\Omega_{p,q}$, a pairwise kNN sub-classifier PkNN$_{p,q}$ can be designed to separate the two classes ω_p and ω_q based on the training subset $\mathcal{T}_p \cup \mathcal{T}_q$. Then, for an M-class classification problem, $M(M-1)/2$ sub-classifiers PkNN$_{p,q}$ $(1 \leq p < q \leq M)$ can be designed and the final classification result is obtained by combining the outputs of these PkNN sub-classifiers. A variety of schemes have been proposed for deriving a combined decision from individual ones, such as voting rule [33], Bayes combination [34], neural networks [35], and evidential combination [36]–[40]. Considering that the output of each PkNN sub-classifier may involve great uncertainty, the PkNN sub-classifiers are combined in belief function framework [26], [27] due to its well capability of modeling and combining uncertain information. Figure 1 shows the scheme of kNN classification with pairwise distance metrics and belief function theory.

Using belief function theory to solve a specific problem generally involves three processes: evidence representation, evidence combination, and decision making based on the combined results. Thus, after a brief recall of the basics of belief function theory in Section IV-A, we will show how to represent pieces of evidence for the outputs of PkNN sub-classifiers in Section IV-B, and how to combine the generated pieces of evidence and make a final decision for classification in Section IV-C.

A. BASICS OF BELIEF FUNCTION THEORY

The belief function theory [26], [27], also known as Dempster-Shafer theory or evidence theory, is a generalization of probability theory. It offers a well-founded and workable framework to represent and combine a large variety of uncertain information. The prerequisite of reasoning in belief function framework is the representation of the available information, which is usually called *evidence*. This is done based on some basic functions used to represent our knowledge about the considered problem. At a glance, there are three main functions: mass, belief and plausibility functions. The mass function is the most basic and intuitive way of expressing someone's degrees of belief. The belief and plausibility functions are often used to compute intervals in order to bound the uncertainty.

In belief function theory, a problem domain is represented by a finite set $\Omega = \{\omega_1, \omega_2, \cdots, \omega_M\}$ of mutually exclusive and exhaustive hypotheses called the *frame of discernment*. A *mass function* expressing the belief committed to the elements of 2^Ω by a given source of evidence is a mapping function m: $2^\Omega \to [0, 1]$, such that

$$m(\emptyset) = 0 \text{ and } \sum_{A \in 2^\Omega} m(A) = 1. \tag{14}$$

The mass function $m(A)$ measures the degree of belief exactly assigned to a proposition A and represents how strongly the proposition is supported by evidence. The belief assigned to Ω, or $m(\Omega)$, is referred to as the degree of global ignorance.

Shafer [27] also defines the *belief function* and *plausibility function* of $A \in 2^\Omega$ as

$$Bel(A) = \sum_{B \subseteq A} m(B) \text{ and } Pl(A) = \sum_{B \cap A \neq \emptyset} m(B). \tag{15}$$

$Bel(A)$ represents the exact support to A and its subsets, and $Pl(A)$ represents all the possible support to A and its subsets. The functions m, Bel and Pl are in one-to-one correspondence.

Several distinct bodies of evidence characterized by different mass functions can be combined using *Dempster's rule of combination* $\bigoplus$. Mathematically, the Dempster's rule of combination of two mass functions m_1 and m_2 defined on the same frame of discernment Ω is

$$m(A) = \begin{cases} 0, & \text{for } A = \emptyset \\ \dfrac{\sum\limits_{B,C \in 2^\Omega; B \cap C = A} m_1(B) m_2(C)}{1 - \sum\limits_{B,C \in 2^\Omega; B \cap C = \emptyset} m_1(B) m_2(C)}, & \text{for } A \in 2^\Omega \setminus \emptyset. \end{cases} \tag{16}$$

As described in [27], Dempster's rule of combination is both commutative and associative.

For decision-making, the maximum plausibility (as defined in Eq.(15)) rule is usually utilized to make the final decision. Suppose $m = m_1 \bigoplus m_2$, then

$$Pl(\{\omega_i\}) \propto Pl_1(\{\omega_i\}) Pl_2(\{\omega_i\}), \forall \omega_i \in \Omega. \tag{17}$$

That is, when combining several pieces of evidence, we do not need to compute the complete mass function using Dempster's rule of combination. Instead, we can compute the combined plausibility using Eq.(17) to make the decision equivalently.

In order to manipulate the belief functions more effectively, some probabilistic operations (conditioning, deconditioning, etc) are introduced to belief function framework [41]. Conditional beliefs represent knowledge which is valid provided that a hypothesis is satisfied. Let m^Ω be a mass function on Ω, $S \subseteq \Omega$ an hypothesis and m_S^Ω the categorical mass function such that $m_S^\Omega(S) = 1$. Then the conditional mass function $m^\Omega[S]$ is

$$m^\Omega[S] = m^\Omega \bigoplus m_S^\Omega. \tag{18}$$

The above operation is referred to as *Dempster's rule of conditioning*. In contrary, if we want to recover m^Ω from the conditional mass function $m^\Omega[S]$, the following *deconditioning* operation can be used:

$$m^\Omega(A \cup \overline{S}) = m^\Omega[S](A), \text{ for } A \in 2^\Omega \setminus \emptyset, \tag{19}$$

where $\overline{S}$ denotes the complement of set S with respect to set Ω.

B. EVIDENCE REPRESENTATION FOR THE OUTPUTS OF PkNN SUB-CLASSIFIERS

Our aim is to use the belief function theory to model the uncertainty inherent in the pairwise classification. Now, with a set of sub-classifiers PkNN$_{p,q}$ $(1 \leq p < q \leq M)$, we first study the representation of their outputs in belief function framework. As reviewed in the above subsection, there are three main functions for evidence representation, i.e., mass, belief and plausibility functions. In this work, the frequently used mass function is selected for its easiness and intuitiveness of representing evidence.

For the output of each sub-classifier PkNN$_{p,q}$, two types of uncertainty are involved as follows.

- **Outer-pair uncertainty**. This type of uncertainty is caused by the fact that the real class label of query pattern $\mathbf{y}$ may actually not belong to class pair $\Omega_{p,q}$. Therefore, the output of sub-classifier PkNN$_{p,q}$ should be represented within the global frame of discernment $\Omega = \{\omega_1, \cdots, \omega_M\}$.
- **Inner-pair uncertainty**. When the real class label of query pattern $\mathbf{y}$ belongs to class pair $\Omega_{p,q}$, affected by the noise of the training samples, the classification result of sub-classifier PkNN$_{p,q}$ is not always accurate. This type of uncertainty can be modeled within the local frame of discernment $\Omega_{p,q} = \{\omega_p, \omega_q\}$.

When classifying query pattern $\mathbf{y}$ using sub-classifier PkNN$_{p,q}$, suppose $\mathbf{x}_j$ is one of its k nearest neighbors in the training subset $\mathcal{T}_p \cup \mathcal{T}_q$, and its class label is $\omega_i \in \Omega_{p,q}$. It can be seen as a piece of evidence that supports the query pattern $\mathbf{y}$ belonging to ω_i. However, considering the *outer-pair uncertainty*, this piece of evidence should be constructed conditioned on the hypothesis $\omega_i \in \Omega_{p,q}$ as

$$m^\Omega[\Omega_{p,q}](\{\omega_i\} \mid \mathbf{x}_j) = 1. \tag{20}$$

Further, due to the *inner-pair uncertainty*, this piece of evidence does not by itself provide 100% reliability. In the formalism of belief function theory, this can be expressed by saying that only some part of the belief is committed to ω_i. Because the class label $\omega_i \in \Omega_{p,q}$ does not point to other particular class, the rest of the belief should be assigned to the local frame of discernment $\Omega_{p,q}$ representing local ignorance.

Therefore, this piece of evidence can be represented by the following mass function:

$$\begin{cases} m^{\Omega}[\Omega_{p,q}](\{\omega_i\} \mid \mathbf{x}_j) = \alpha_j \\ m^{\Omega}[\Omega_{p,q}](\Omega_{p,q} \mid \mathbf{x}_j) = 1 - \alpha_j, \end{cases} \tag{21}$$

where $\alpha_j \in [0, 1]$ is the belief that sample $\mathbf{x}_j$ and query pattern $\mathbf{y}$ share the same class, and can be determined using the PW distance metric-based logistic regression model in Eq. (5) when the features are assumed to be independent:

$$\alpha_j = \frac{1}{1 + \exp\left(d^2_{PW}(\mathbf{x}_j, \mathbf{y}) - \mu_{p,q}\right)}. \tag{22}$$

Otherwise, the following PM distance metric-based logistic regression model is selected:

$$\alpha_j = \frac{1}{1 + \exp\left(d^2_{PM}(\mathbf{x}_j, \mathbf{y}) - \mu_{p,q}\right)}. \tag{23}$$

C. EVIDENCE COMBINATION AND DECISION MAKING FOR CLASSIFICATION

For the sub-classifier PkNN$_{p,q}$, based on the k nearest neighbors of query pattern $\mathbf{y}$, we can calculate all the corresponding k mass functions as the way developed in the above subsection. As the items of evidence from different neighbors are independent, the k mass functions are combined using Dempster's rule defined by Eq. (16) to form a resulting mass function synthesizing the overall conditional mass regarding the label of $\mathbf{y}$ as

$$\begin{aligned} m^{\Omega}[\Omega_{p,q}] = m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_1}) \oplus m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_2}) \\ \oplus \cdots \oplus m^{\Omega}[\Omega_{p,q}](\cdot \mid \mathbf{x}_{i_k}), \end{aligned} \tag{24}$$

where $i_1, i_2, \cdots, i_k$ are the indices of the k nearest neighbors of query pattern $\mathbf{y}$.

In a similar way, based on the outputs of the $M(M-1)/2$ sub-classifiers PkNN$_{p,q}$ $(1 \leq p < q \leq M)$, we can calculate all the corresponding $M(M-1)/2$ conditional mass functions. In order to combine these conditional mass functions in a uniform framework, the conditional mass function constructed as Eq. (24) should be deconditioned using Eq. (19) as

$$\begin{cases} m^{\Omega}_{p,q}(\overline{\{\omega_q\}}) = m^{\Omega}[\Omega_{p,q}](\{\omega_p\}) \\ m^{\Omega}_{p,q}(\overline{\{\omega_p\}}) = m^{\Omega}[\Omega_{p,q}](\{\omega_q\}) \\ m^{\Omega}_{p,q}(\Omega) = m^{\Omega}[\Omega_{p,q}](\Omega_{p,q}). \end{cases} \tag{25}$$

where $\overline{\{\omega_p\}}$ and $\overline{\{\omega_q\}}$ denote the complement of set $\{\omega_p\}$ and $\{\omega_q\}$ with respect to set Ω, respectively.

Because the mass and plausibility functions are in one-to-one correspondence, we can compute the plausibility function $Pl_{p,q}$ from the above deconditioned mass function $m^{\Omega}_{p,q}$ using Eq. (15) as

$$Pl_{p,q}(\{\omega_i\}) = \begin{cases} 1 - m^{\Omega}[\Omega_{p,q}](\{\omega_q\}), & \text{if } i = p \\ 1 - m^{\Omega}[\Omega_{p,q}](\{\omega_p\}), & \text{if } i = q \\ 1, & \text{otherwise }. \end{cases} \tag{26}$$

In order to decrease the computation complexity, instead of combining the $M(M-1)/2$ mass functions $m^{\Omega}_{p,q}$ $(1 \leq p < q \leq M)$ using Dempster's rule of combination, we can compute the combined plausibility function Pl directly using Eq. (17) for decision making as follows:

$$Pl(\{\omega_i\}) \propto Pl'(\{\omega_i\}) = \prod_{1 \leq p < q \leq M} Pl_{p,q}(\{\omega_i\}), \ \forall \omega_i \in \Omega. \tag{27}$$

Note that the combined plausibility function Pl is proportional to Pl', so the maximum plausibility rule can be used for Pl' equivalently to make a decision. Finally, the class label of query pattern $\mathbf{y}$ is assigned to the class with maximum plausibility.

Remark 1: The usage of belief function theory in kNN-based classification is not new. An evidential version of kNN, denoted by EkNN [42], has been proposed based on the belief function theory; it introduces the ignorance class to model the uncertainty. In [43], the EkNN rule was further extended to deal with uncertainty using rejection class and meta-classes. Recently, an evidential editing procedure was developed in [44] to preprocess the original training samples in order to model the imprecision and uncertainty of samples in overlapping regions or noisy patterns. However, neither of them has considered the distance metric learning problem in belief function framework, which is the subject of this work.

V. EXPERIMENTS

The performance of the proposed BPkNN classifier was assessed by two different types of experiments. In the first experiment, synthetic data sets were used to show the behavior of the proposed classifier in controlled settings. In the second one, twenty real data sets from the UCI Machine Learning Repository [28] were considered, with the aim to show that the proposed technique is adequate for a variety of real tasks involving different data conditions: large/small size, high/low dimension, etc.

A. SYNTHETIC DATA TEST

1) A DEMONSTRATION EXAMPLE

A two-dimensional three-class classification example was designed to show the process and advantage of the proposed BPkNN classifier. The following class-conditional normal distributions were assumed:

$$\begin{aligned} &\mu_A = (6, 6)^T, \quad \mu_B = (14, 6)^T, \ \mu_C = (14, 14)^T, \\ &\Sigma_A = 3\mathbf{I}_2, \ \Sigma_B = 3\mathbf{I}_2, \ \Sigma_C = 3\mathbf{I}_2. \end{aligned}$$

As shown in Figure 2(a), a total number of 60 training samples were generated using equal prior probabilities, and we consider the classification of one test sample $\mathbf{y}$, whose real label is Class B. We can see that the test sample $\mathbf{y}$ is quite close to the boundaries of the three classes, and in this small training data set condition, it is quite difficult to make the right classification. As can be seen from Figure 2(b), the original kNN classifier based on L2 distance metric (L2-kNN) just misclassifies this data point as Class A with the 1NN rule. Next, we will see how the proposed BPkNN classifier performs for this classification problem.

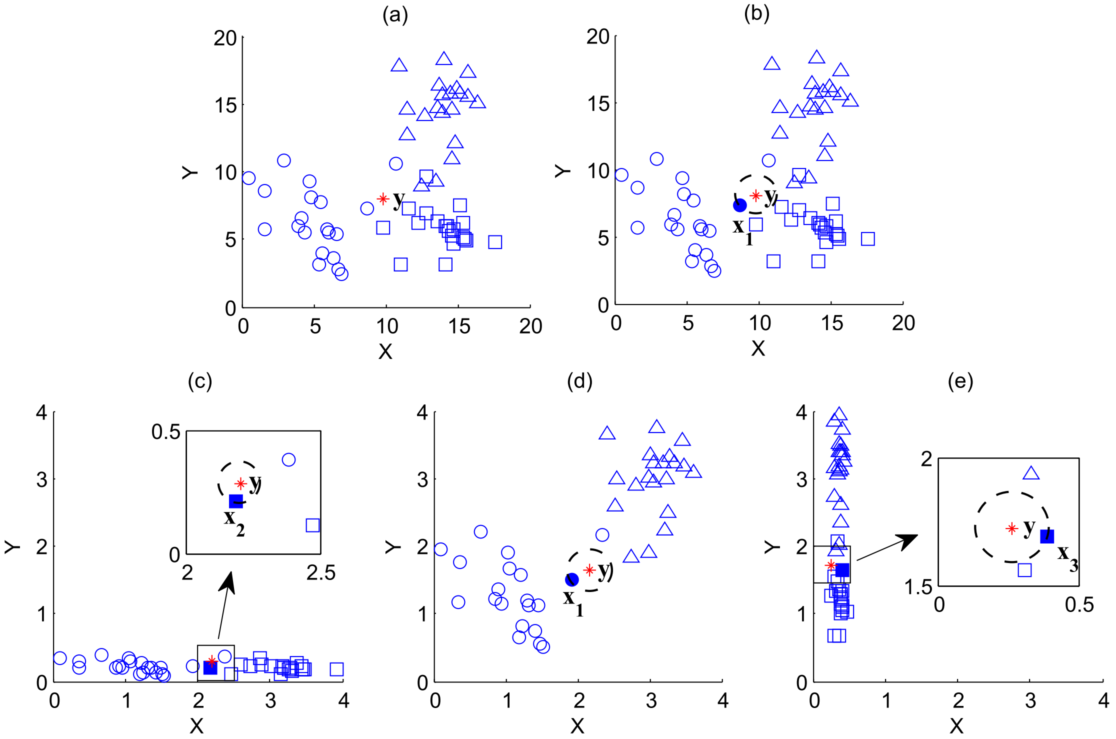


FIGURE 2. Classification results of test sample y with 60 training samples (with '○' for class A, '□' for class B and '△' for class C, respectively). (a) Data set. (b) Result of L2-kNN. (c) Result of PkNN$_{A,B|}$. (d) Result of PkNN$_{A,C|}$. (e) Result of PkNN$_{B,C|}$.

The first step of the BP*k*NN classifier is to make classification using pairwise sub-classifiers. For this three-class classification problem, three pairwise sub-classifiers can be designed with separately learned feature weights as follows.

$$\begin{aligned}&\mathrm{P}k\mathrm{NN}_{A,B}\colon\ \lambda^X_{A,B}=0.2231,\ \lambda^Y_{A,B}=0.0357;\\ &\mathrm{P}k\mathrm{NN}_{A,C}\colon\ \lambda^X_{A,C}=0.2200,\ \lambda^Y_{A,B}=0.2050;\\ &\mathrm{P}k\mathrm{NN}_{B,C}\colon\ \lambda^X_{B,C}=0.0265,\ \lambda^Y_{B,C}=0.2156.\end{aligned}$$

Figures 2(c)-(e) show the classification results of the test sample **y** using the above pairwise sub-classifiers with the 1NN rule. To visualize the results, each original point **x** is replaced by **Ax**, where **A** is a diagonal matrix filled by the learned feature weights. After this procedure, the classification problem is transformed into applying the standard Euclidean metric to the rescaled data to find the nearest neighbors. Thanks to the locally learned pairwise distance metrics, when classifying **y** between Class A and Class B, feature X is assigned larger weight, whereas in classifying **y** between Class B and Class C, feature Y is assigned larger weight. As shown in Figures 2(c) and (e), both the two sub-classifiers P*k*NN$_{A,B}$ and P*k*NN$_{B,C}$ provide the correct classification result as Class B. However, as shown in Figure 2(d), the sub-classifiers P*k*NN$_{A,C}$ misclassifies this data point as Class A.

The second step of the BP*k*NN classifier is to combine the outputs of these P*k*NN sub-classifiers in belief function framework to get the final classification result. For doing this, we should first build a mass function based on the output of each P*k*NN sub-classifier. Using Eqs.(20)-(22), the following three mass functions can be built as

$$\begin{aligned}&\mathrm{P}k\mathrm{NN}_{A,B}\colon && m^{\Omega}[\{A,B\}](\{B\})=0.8108,\\ & && m^{\Omega}[\{A,B\}](\{A,B\})=0.1892;\\ &\mathrm{P}k\mathrm{NN}_{A,C}\colon && m^{\Omega}[\{A,C\}](\{A\})=0.9920,\\ & && m^{\Omega}[\{A,C\}](\{A,C\})=0.0080;\\ &\mathrm{P}k\mathrm{NN}_{B,C}\colon && m^{\Omega}[\{B,C\}](\{B\})=0.8977,\\ & && m^{\Omega}[\{B,C\}](\{B,C\})=0.1023.\end{aligned}$$

Then, Using Eqs.(24)-(27), we get the combined result as

$$Pl'(\{A\})=0.17,\ Pl'(\{B\})=1,\ Pl'(\{C\})=0.09.$$

Finally, based on the maximum plausibility rule, we get the correct final classification result as Class B.

2) TEST OF DATA SETS WITH DIFFERENT CLASS OVERLAPPING

This experiment was designed to compare the proposed BP*k*NN classifier with those *k*NN classifiers based on other representative distance metrics, including the original *k*NN classifier based on L2 distance metric (L2-*k*NN) [7], the *k*NN classifier based on GW distance metric (GW-*k*NN) [12] and the *k*NN classifier based on CDW distance metric (CDW-*k*NN) [23]. A ten-dimensional twenty-class classification problem was considered. The multidimensional Gaussian mixture model was assumed for the data distributions. The mean value of each class-conditional Gaussian distribution was drawn randomly from the feature space $[-10, 10]^{10}$. For comparisons, we changed the covariance of each distribution to control the degree of class overlapping (larger covariances

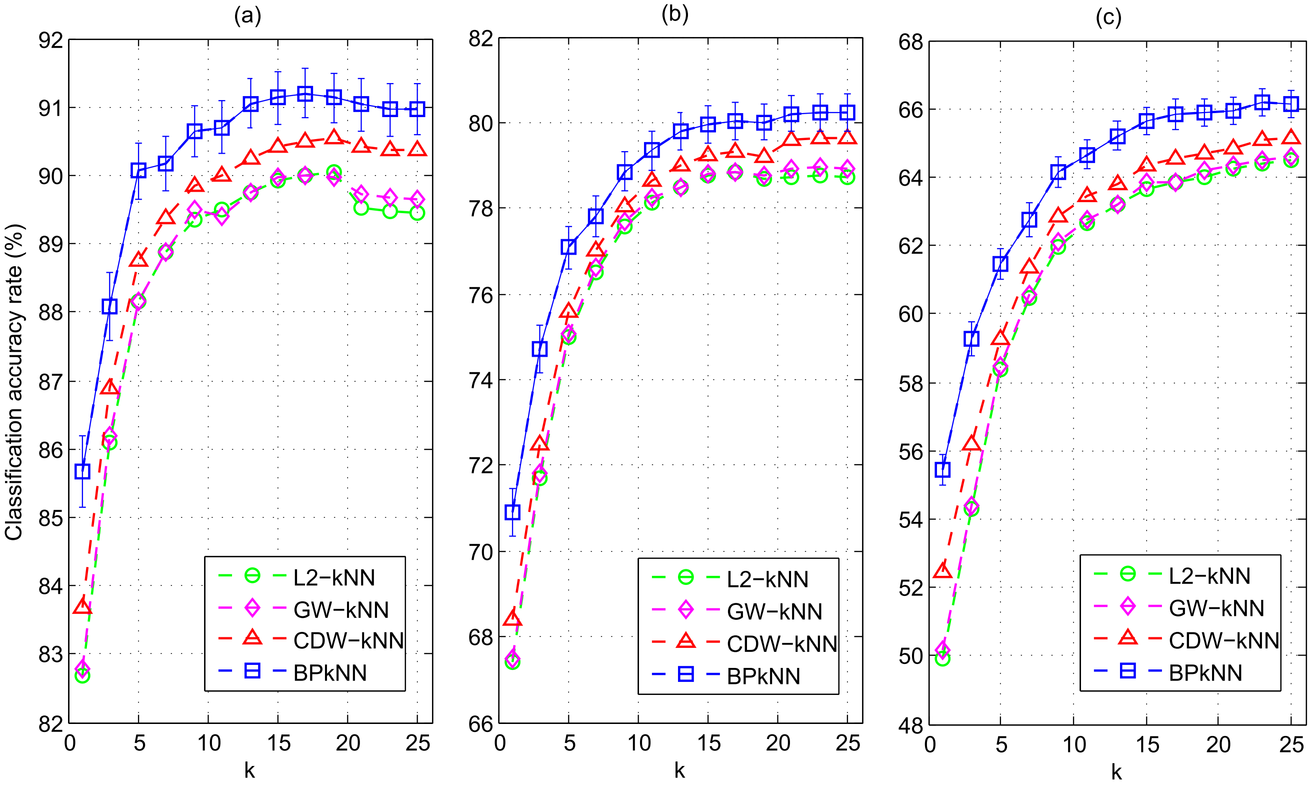


FIGURE 3. Classification accuracy rate (in %) for synthetic data sets with different class overlapping. (a) Case 1. (b) Case 2. (c) Case 3.

lead to a higher degree of class overlapping):

Case 1: $\Sigma_i = 4\mathbf{I}_{10}, \quad i = 1, 2, \cdots, 20;$
Case 2: $\Sigma_i = 5\mathbf{I}_{10}, \quad i = 1, 2, \cdots, 20;$
Case 3: $\Sigma_i = 6\mathbf{I}_{10}, \quad i = 1, 2, \cdots, 20.$

For each case, a training set of 200 samples were generated using equal prior probabilities, and a test set of 2000 samples was used for classification accuracy estimation. A total of 30 trials were performed with independently generated data sets. The average classification accuracy and the corresponding 95% confidence interval were calculated. For the proposed BPkNN classifier, as the features are independent from each other in this study, we used the PW distance metric. For all of the considered classifiers, values of k ranging from 1 to 25 have been investigated.

Figure 3 shows the classification accuracy of the considered classifiers for the three cases with different degrees of class overlapping. As can be seen from these results, the GW-kNN classifier shows similar performance as compared to the original L2-kNN classifier, which is mainly because the learned GW distance metrics have almost the same weights for the ten involved features. The CDW-kNN classifier, which is based on CDW distance metric, is just slightly better than the original L2-kNN classifier. The proposed BPkNN classifier produces the highest classification accuracy for all of the three cases, which demonstrates the generalizability of the proposed method on synthetic difficult data. The reason is that, for each PkNN sub-classifier, the PW distance metric characterizes more local specificities in the feature space, and further in the combination process, the output uncertainty of those PkNN sub-classifiers is well addressed. In addition, the performance improvement is more significant when the number of neighbors k takes smaller values, in which cases the distance metric plays a more important role in determining the performance of the kNN-based classifiers.

B. REAL DATA TEST

1) DATA SETS AND EXPERIMENTAL CONDITIONS

Twenty representative data sets from the UCI repository were selected to evaluate the performance of the proposed BPkNN classifier. The main characteristics of the 20 data sets are summarized in Table 1. It can be seen that the selected data sets vary greatly in the number of instances (from 80 to 12,690), the number of features (from 4 to 60), and the number of classes (from 2 to 11).

To develop the experiments, we consider the B-Fold Cross-Validation (B-CV) model. Each data set is divided into B blocks, with $B - 1$ blocks as a training set and the remaining block as a test set. Therefore, each block is used exactly once as a test set. We use the 10-CV here, i.e., ten random partitions of the original data set, with nine of them (90%) as the training set and the remainder (10%) as the test set. For each data set, we consider the average result of the ten partitions.

To assess whether significant differences exist among different methods, we adopt a nonparametric statistical analysis.

TABLE 1. Statistics of the real data sets used in the experiment.

Data set	# Instances	# Features	# Classes
Australian	690	14	2
Balance	625	4	3
Car	1,278	6	4
Contraceptive	1,473	9	3
Dermatology[a]	358	34	6
Ecoli	336	7	8
Glass	214	9	6
Hepatitis[a]	80	19	2
Ionosphere	351	33	2
Iris	150	4	3
Lymphography	148	18	4
Nursery	12,690	8	5
Page-blocks	5,472	10	5
Sonar	208	60	2
Thyroid	7,200	21	3
Vehicle	846	18	4
Vowel	990	13	11
Wine	178	13	3
Yeast	1,484	8	10
Zoo	101	16	7

[a] For the data sets containing missing values, instances with missing feature values are removed.

For conducting multiple statistical comparisons over multiple data sets, as suggested in [45], [46], the Iman and Davenport test and the corresponding *post hoc* Bonferroni-Dunn test were employed. For performing multiple comparisons, it is necessary to check whether the results obtained by different methods present any significant difference (Iman and Davenport test), and in the case of finding one, we can find out by using a *post hoc* test to compare the control method with the remaining methods (Bonferroni-Dunn test). We use $\alpha = 0.05$ as the significance level in all cases. For a detailed description of these tests, one can refer to [45], [46].

2) CLASSIFICATION ACCURACY EVALUATION

In this experiment, we aim to compare the classification accuracy of our proposed BPkNN classifier with those kNN classifiers based on other representative distance metrics including L2-kNN [7], GM-kNN [12], CDW-kNN [23], A-kNN [21], as well as the DGC+ classifier [29], which is a popular instance-based method based on weighted data gravitation classification. Besides, in order to evaluate the effectiveness of the combination process using the belief function theory, apart from the above compared methods, we also considere the method of combining the PkNN sub-classifiers using the voting rule (denoted as VPkNN). Settings of the comparison methods are summarized in Table 2.

TABLE 2. Settings of the comparison methods.

Method	Parameter	Value
L2-kNN	Number of neighbors Distance metric	3 Euclidean
GM-kNN	Number of neighbors Distance metric	3 Global Mahalanobis
CDW-kNN	Number of neighbors Distance metric	3 Class-dependent weighted Euclidean
A-kNN	Number of neighbors Distance metric	3 Adaptive Euclidean
DGC+	Distance metric Evolution parameters	Class-dependent weighted Euclidean same as [29]
VPkNN	Number of neighbors Distance metric Combination method	3 Pairwise Mahalanobis Voting rule
BPkNN	Number of neighbors Distance metric Combination method	3 Pairwise Mahalanobis Belief function theory

Table 3 shows the classification accuracy rates of our proposed BPkNN classifier in comparison with other state-of-the-art methods for real data sets. The numbers in brackets represent the rank of each method and the best classification accuracy for each data set is highlighted in bold. As can be seen from these results, the proposed BPkNN classifier outperforms the other methods in 9 of the 20 data sets and obtains competitive accuracy rates in the other data sets. In addition, the BPkNN classifier always performs better than the VPkNN classifier (except the data sets with only two classes, in which cases, only one pairwise sub-classifier is designed and no combination is needed), which demonstrates the effectiveness of the belief function theory for the combination of PkNN sub-classifiers.

To compare the results statistically, we carry out nonparametric tests for multiple comparisons based on the average ranks obtained over the considered data sets. First, we use the Iman and Davenport test to determine whether significant differences exist among all of the average values. The Iman and Davenport statistic (distributed according to the F-distribution with 6 and 114 degrees of freedom) is 10.39 for average ranks and the corresponding critical value is 2.18 for a significance level of $\alpha = 0.05$. Given that the Iman and Davenport statistic is clearly greater than the critical value, the test rejects the null hypothesis, and therefore, it can be said that there are significant differences among the accuracy results of the methods.

Then, we apply the *post hoc* Bonferroni-Dunn test to compare the best ranking method (i.e., BPkNN) with the remaining ones. Fig. 4 shows the test result of the average ranks with a significance level of $\alpha = 0.05$, in which case the calculated critical difference is 1.73. The critical difference value is represented as a thicker horizontal line, and those values that exceed this line are methods with significantly different results than the control method (i.e., BPkNN). It can be seen that the proposed BPkNN classifier performs significantly better than all those kNN classifiers based on other distance metrics, including L2-kNN, GM-kNN, CDW-kNN and A-kNN.

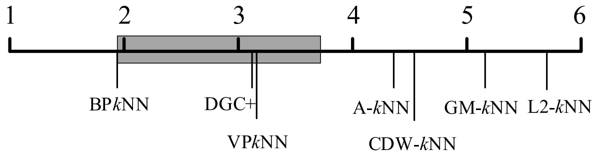


FIGURE 4. Bonferroni-Dunn test result of the average ranks.

TABLE 3. Classification accuracy rate (in %) for real data sets.

Data set	L2-kNN	GM-kNN	CDW-kNN	A-kNN	DGC+	VPkNN	BPkNN
Australian	88.78 (4)	**89.21** (1)	81.45 (7)	83.91 (5)	83.74 (6)	88.90 (2.5)	88.90 (2.5)
Balance	83.37 (7)	84.66 (6)	85.27 (5)	89.43 (2)	**89.66** (1)	88.60 (4)	89.26 (3)
Car	92.31 (4)	89.60 (6)	90.48 (5)	87.97 (7)	**95.23** (1)	92.48 (3)	94.05 (2)
Contraceptive	44.95 (7)	45.42 (6)	47.77 (5)	48.27 (4)	49.45 (2)	48.65 (3)	**49.87** (1)
Dermatology	96.90 (4)	96.60 (5)	97.01 (3)	95.21 (7)	95.44 (6)	97.25 (2)	**97.40** (1)
Ecoli	80.67 (7)	81.27 (5)	80.70 (6)	82.47 (3)	82.17 (4)	84.33 (2)	**84.80** (1)
Glass	70.11 (6)	70.67 (4)	73.61 (2)	64.98 (7)	70.36 (5)	73.10 (3)	**73.77** (1)
Hepatitis	82.51 (7)	84.55 (6)	84.98 (3)	85.08 (2)	**86.28** (1)	84.65 (4.5)	84.65 (4.5)
Ionosphere	85.18 (7)	92.58 (5)	92.39 (6)	**93.72** (1)	93.11 (4)	93.42 (2.5)	93.42 (2.5)
Iris	94.00 (6)	94.33 (5)	94.00 (6)	95.33 (2.5)	95.33 (2.5)	95.00 (4)	**95.67** (1)
Lymphography	77.39 (7)	79.06 (5)	78.86 (6)	80.85 (3)	**81.40** (1)	80.44 (4)	81.10 (2)
Nursery	92.54 (6)	92.86 (5)	93.61 (4)	86.10 (7)	**96.96** (1)	94.83 (3)	95.14 (2)
Page-blocks	95.91 (4)	95.26 (6)	95.76 (5)	96.29 (2)	95.08 (7)	96.16 (3)	**96.45** (1)
Sonar	83.07 (7)	88.25 (3)	86.81 (5)	87.98 (4)	84.87 (6)	**88.33** (1.5)	**88.33** (1.5)
Thyroid	93.89 (6)	92.60 (7)	96.14 (2)	93.96 (5)	**97.04** (1)	94.08 (4)	95.22 (3)
Vehicle	71.75 (3)	71.50 (4)	69.98 (6)	68.79 (7)	71.16 (5)	72.15 (2)	**74.30** (1)
Vowel	97.78 (6)	98.15 (5)	**99.49** (1)	96.87 (7)	98.24 (4)	98.98 (3)	99.10 (2)
Wine	95.49 (3)	93.60 (6)	94.38 (5)	96.63 (2)	**97.31** (1)	93.40 (7)	95.20 (4)
Yeast	53.17 (7)	55.27 (6)	56.47 (4)	55.39 (5)	**59.26** (1)	57.08 (3)	57.33 (2)
Zoo	92.81 (6)	91.33 (7)	93.47 (4.5)	94.47 (4.5)	95.53 (2)	95.40 (3)	**95.98** (1)
Avg. accuracy	83.63	84.34	84.63	84.19	85.88	85.86	86.50
Avg. rank	5.70	5.15	4.53	4.35	3.13	3.15	1.95

TABLE 4. Average runtime (in s) of BPkNN in comparison with L2-kNN for real data sets.

					Train time		Test time		
Data sets	# Train Instances	# Test Instances	# Features	# Classes	L2-kNN	BPkNN	L2-kNN	BPkNN	ratio
Balance	552	63	4	3	0	0.051	0.005	0.016	3.2
Ecoli	302	34	7	8	0	0.441	0.007	0.068	9.7
Glass	192	22	9	6	0	0.225	0.002	0.014	7.0
Nursery	11,421	1,269	8	5	0	1.365	0.241	1.320	5.4
Sonar	187	21	60	2	0	0.523	0.032	0.034	1.1
Vehicle	761	85	18	4	0	0.125	0.016	0.063	3.9
Vowel	891	99	13	11	0	0.732	0.022	0.261	11.9
Wine	160	18	13	3	0	0.028	0.001	0.003	3.0
Yeast	1,335	149	8	10	0	0.950	0.061	0.562	9.2

3) TIME COMPLEXITY ANALYSIS

In this experiment, a time complexity analysis of the proposed BPkNN classifier is provided to show to what extent the runtime depends on factors such as the number of instances, the number of features and the number of classes. Nine representative real data sets selected from Table 1 are considered for evaluation. The numerical experiment is executed by MATLAB 7.8.0 on a Lenovo desktop M4500 with an Intel(R) Core(TM) i5-4590 CPU @3.30 GHz and 4 GB memory.

Table 4 shows the average runtime of the proposed BPkNN classifier in comparison with L2-kNN classifier under 10-CV in the training and testing phases, respectively. It can be seen that, in the training phase, the L2-kNN classifier does not cost any time, whereas some time is consumed for the proposed BPkNN classifier in order to learn the pairwise distance metrics. The test time of the BPkNN classifier is positive correlated with the number of train instances, the number of features and the number of classes, but is acceptable even for those complex data sets, such as *Nursery*, *Vowel*, and *Sonar*. In the testing phase, the BPkNN classifier consumes more time than the L2-kNN classifier. For the BPkNN classifier, the fusion process of PkNN sub-classifiers is quite time-efficient, and therefore the time is mainly consumed in the classification process of multiple PkNN sub-classifiers. Even though the number of PkNN sub-classifiers is $M(M-1)/2$ (with M being the number of classes), each sub-classifier only uses the train instances from the corresponding two classes (about $2N/M$ instances averagely, with N being the number of train instances). Hence the total number of computed instances is about $N(M-1)$, which is just $M-1$ times larger than the original L2-kNN classifier approximately. The last column of Table 4 shows the runtime ratio of BPkNN to L2-kNN, which is in accordance with the above theoretical analysis. For most real classification problems, like those studied in this experiment, the number of classes is not very large, so the time complexity of the proposed method is not very high.

VI. CONCLUSION

In order to improve the performance of the kNN-based classifier in small data set situations, two types of pairwise distance metrics have been proposed in this paper. Compared with the existing distance metrics, the pairwise distance metrics provide greater flexibility to design the feature weights so that the local specificities in the input space can be well characterized. The parameter optimization procedures were designed to learn the pairwise distance metrics from the training data set. For a general polychotomous classification problem, a BP*k*NN classifier was developed, which combines the outputs of locally learned P*k*NN sub-classifiers in the belief function framework. From the results reported in the last section, we can conclude that the proposed method achieved a uniformly good performance when applied to a variety of classification tasks, including those with high dimension and small sample size, in which cases the training data set is not rich enough to well characterize the real class-conditional probability distributions.

REFERENCES

[1] P. Szacherski *et al.*, "Classification of proteomic MS data as Bayesian solution of an inverse problem," *IEEE Access*, vol. 2, pp. 1248–1262, 2014.

[2] S. Walter, J. Kim, D. Hrabal, S. C. Crawcour, H. Kessler, and H. C. Traue, "Transsituational individual-specific biopsychological classification of emotions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 4, pp. 988–995, Jul. 2013.

[3] U. Iqbal, T. Y. Wah, M. H. Ur Rehman, and Q.-U.-A. Mastoi, "Usage of model driven environment for the classification of ECG features: A systematic review," *IEEE Access*, vol. 6, pp. 23120–23136, 2018.

[4] P. O. L. Junior, L. G. de Castro Junior, and A. L. Zambalde, "Applying textmining to classify news about supply and demand in the coffee market," *IEEE Latin Amer. Trans.*, vol. 14, no. 12, pp. 4768–4774, Dec. 2016.

[5] L. Jiao, T. Denœux, and Q. Pan, "A hybrid belief rule-based classification system based on uncertain training data and expert knowledge," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 12, pp. 1711–1723, Dec. 2016.

[6] A. K. Jain, R. P. W. Duin, and J. C. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[7] E. Fix and J. Hodges, "Discriminatory analysis, nonparametric discrimination: Consistency properties," USAF School Aviation Med., Randolph Field, TX, USA, Tech. Rep. 4, 1951.

[8] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[9] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, "Local geometric structure feature for dimensionality reduction of hyperspectral imagery," *Remote Sens.*, vol. 9, no. 8, p. 790, Aug. 2017.

[10] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, to be published. doi: 10.1109/TCYB.2018.2810806.

[11] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York, NY, USA: Springer-Verlag, 1996.

[12] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 521–528.

[13] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 11–18.

[14] Z. Zhang, J. T. Kwok, and D. Y. Yeung, "Parametric distance metric learning with label information," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1450–1452.

[15] C. F. Eick, A. Rouhana, A. Bagherjeiran, and R. Vilalta, "Using clustering to learn distance functions for supervised similarity assessment," *Eng. Appl. Artif. Intell.*, vol. 19, no. 4, pp. 395–401, 2006.

[16] E. H. Han, G. Karypis, and V. Kumar, "Text categorization using weight adjusted *k*-nearest neighbor classification," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2001, pp. 53–65.

[17] L. Jiao, Q. Pan, X. Feng, and F. Yang, "An evidential *k*-nearest neighbor classification method with weighted attributes," in *Proc. 16th Int. Conf. Inf. Fusion*, Jul. 2013, pp. 145–150.

[18] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.

[19] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, Sep. 2002.

[20] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[21] J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognit. Lett.*, vol. 28, no. 2, pp. 207–213, 2007.

[22] R. Paredes and E. Vidal, "A class-dependent weighted dissimilarity measure for nearest neighbor classification problems," *Pattern Recognit. Lett.*, vol. 21, pp. 1027–1036, Nov. 2000.

[23] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, Jul. 2006.

[24] L. Jiao, T. Denœux, and Q. Pan, "Fusion of pairwise nearest-neighbor classifiers based on pairwise-weighted distance metric and dempster-shafer theory," in *Proc. Int. Conf. Inf. Fusion*, Jul. 2014, pp. 1–7.

[25] L. Jiao, "Classification of uncertain data in the framework of belief functions: Nearest-neighbor-based and rule-based approaches," M.S. theses, Dept. Comput. Eng., Université Technologie Compiègne, Compiègne, France, 2015. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01304280

[26] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Statist.*, vol. 38, no. 2, pp. 325–339, 1967.

[27] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton Univ. Press, 1976.

[28] D. Dua and E. K. Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[29] A. Cano, A. Zafra, and S. Ventura, "Weighted data gravitation classification for standard and imbalanced data," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1672–1687, Dec. 2013.

[30] M. Avriel, *Nonlinear Programming: Analysis and Methods*. Chelmsford, MA, USA: Courier Corporation, 2003.

[31] A. R. Webb, *Statistical Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2003.

[32] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, Tech. Rep. 4, 2006.

[33] D. Ruta and G. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, pp. 63–81, Mar. 2005.

[34] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 619–627.

[35] G. Giacinto and F. Roli, "Design of effective neural network ensembles for image classification purposes," *Image Vis. Comput.*, vol. 19, nos. 9–10, pp. 699–707, Aug. 2001.

[36] B. Quost, T. Denœux, and M.-H. Masson, "Classifier fusion in the Dempster–Shafer framework using optimized t-norm based combination rules," *Int. J. Approx. Reasoning*, vol. 52, pp. 353–374, Mar. 2011.

[37] M. Tabassian, R. Ghaderi, and R. Ebrahimpour, "Combination of multiple diverse classifiers using belief functions for handling data with imperfect labels," *Expert Syst. Appl.*, vol. 39, pp. 1698–1707, Feb. 2012.

[38] L. Jiao, Q. Pan, and X. Feng, "Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric," *Neurocomputing*, vol. 151, pp. 1468–1476, Mar. 2015.

[39] L. Jiao, Q. Pan, T. Denœux, Y. Liang, and X. Feng, "Belief rule-based classification system: Extension of FRBCS in belief functions framework," *Inf. Sci.*, vol. 309, pp. 26–49, Jul. 2015.

[40] Z. Liu, Q. Pan, J. Dezert, and A. Martin, "Combination of classifiers with optimal weight based on evidential reasoning," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1217–1230, Jun. 2018.

[41] P. Smets, "Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem," *Int. J. Approx. Reasoning*, vol. 9, no. 1, pp. 1–35, 1993.

[42] T. Denœux, "A *k*-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804–813, May 1995.

[43] Z. Liu, Q. Pan, and J. Dezert, "A new belief-based *k*-nearest neighbor classification method," *Pattern Recognit.*, vol. 46, pp. 834–844, Mar. 2013.

[44] L. Jiao, T. Denœux, and Q. Pan, "Evidential editing *k*-nearest neighbor classifier," in *Proc. 13th Eur. Conf. Symbolic Quant. Approaches Reasoning Uncertainty*, 2015, pp. 461–471.

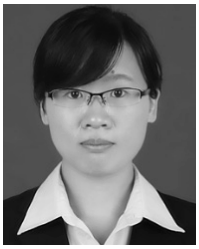
[45] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[46] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, pp. 2044–2064, May 2010.

LIANMENG JIAO (M'17) received the B.E. and M.E. degrees in control science and engineering from Northwestern Polytechnical University, Xi'an, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer science from the Université de Technologie de Compiègne, Compiègne, France, in 2015.

Since 2016, he has been an Assistant Professor with the School of Automation, Northwestern Polytechnical University. He has authored two books and more than 20 peer-reviewed journal and conference papers. His research interests include information fusion, belief function theory, and its application in machine learning and decision making.

Dr. Jiao has been served as the Section Chair for the 12th International Conference on Intelligent Unmanned Systems, an Organization Committee Member for the Fourth School on Belief Functions and their Applications, and an Editorial Board Member for the *American Journal of Artificial Intelligence*.



XIAOJIAO GENG received the B.S. degree in mathematics from Shandong Normal University, Jinan, China, in 2012, and the M.S. degree in applied mathematics from Northwestern Polytechnical University, Xi'an, China, in 2016, where she is currently pursuing the Ph.D. degree in control science and engineering. Her research interests include pattern classification and the development of expert systems under uncertainty using belief function theory.

QUAN PAN (M'05) received the B.E. degree from the Huazhong Institute of Technology, Wuhan, China, in 1991, and the M.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1991 and 1997, respectively.

Since 1998, he has been a Professor with the School of Automation. He has published five books, almost 100 international journal papers, and more than 60 international conference papers. His research interests include decision making, information fusion, hybrid system estimation theory, belief function theory, and image processing.

Dr. Pan was a recipient of the Sixth National Youth Award for Outstanding Contribution to Science and Technology, in 1998, and the Chinese National New Century Excellent Professional Talent, in 2000.