Fusion of Pairwise Nearest-Neighbor Classifiers Based on Pairwise-Weighted Distance Metric and Dempster-Shafer Theory

Lianmeng Jiao
School of Automation
Northwestern Polytechnical University
Xi'an, 710072, P. R. China, and
UMR CNRS 7253, Heudiasyc
Université de Technologie de Compiègne
60205 Compiègne, France
Email: lianmeng.jiao@utc.fr

Thierry Denœux
UMR CNRS 7253, Heudiasyc
Université de Technologie de Compiègne
60205 Compiègne, France
Email: thierry.denoeux@hds.utc.fr

Quan Pan
School of Automation
Northwestern Polytechnical University
Xi'an, 710072, P. R. China
Email: quanpan@nwpu.edu.cn

***Abstract*—The performance of the nearest-neighbor (NN) classifier is known to be very sensitive to the distance metric used in classifying a query pattern, especially in scarce-prototype cases. In this paper, a pairwise-weighted (PW) distance metric related to pairs of class labels is proposed. Compared with the existing distance metrics, it provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized. Base on the proposed PW distance metric, a polychotomous NN classification problem is solved by combining several pairwise NN (PNN) classifiers within the framework of Dempster-Shafer theory to deal with the uncertain output information. Two experiments based on synthetic and real data sets were carried out to show the effectiveness of the proposed method.**

***Keywords*—pattern classification, nearest-neighbor classifier, pairwise-weighted distance metric, Dempster-Shafer theory.**

I. Introduction

The nearest-neighbor (NN) rule, first proposed by Fix and Hodges [1], is one of the most popular and successful pattern classification techniques. Given a set of N labeled samples (or phototypes) $\mathcal{T} = \{(\mathbf{x}^{(1)}, \omega^{(1)}), \cdots, (\mathbf{x}^{(N)}, \omega^{(N)})\}$ with input vector $\mathbf{x}^{(i)} \in \mathcal{R}^D$ and class label $\omega^{(i)} \in \Omega = \{\omega_1, \cdots, \omega_M\}$, the NN rule classifies a query pattern $\mathbf{y} \in \mathcal{R}^D$ to the class of its nearest neighbor in the training set T. The basic rationale of the NN rule is both simple and intuitive: patterns close in feature space are likely to belong to the same class. The good behavior of the NN rule with unbounded number of prototypes is well known [2]. However, in many practical pattern classification applications only a small number of prototypes is available. Typically, in such a scarce-prototype situation, the ideal asymptotical behavior of NN classifier degrades dramatically [3]. This motivates the growing interest in finding variants of the NN rule and adequate distance measures (or metrics) that help improve the NN classification performance in small data set situations.

As the core of the NN rule, the distance metric plays a crucial role in determining the classification performance. To overcome the limitations of the original Euclidean (L2) distance metric, a number of adaptive methods have recently been proposed to address the distance metric learning issue. According to the structure of the metric, these methods can be mainly divided into two categories: global distance metric learning [4], [5], [6], and local distance metric learning [7], [8], [9], [10], [11]. The first approach learns the distance metric in a global sense, i.e., to share the same simple-weighted (SW) distance metric for all the prototypes:

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

where, $\mathbf{x}$ is a prototype in the training set, $\mathbf{y}$ is a query pattern to be classified, and λ_j is the weight of the j-th feature. Although the above global distance metric is intuitively appealing, it is too coarse 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 (for example, one feature may be more discriminative for some classes, but irrelevant for others). So, many methods have been developed to learn a distance metric in a local setting, i.e., the feature weights may be different for different patterns. The most representative one is the class-dependent weighted (CDW) distance metric [7], [8], which is related to the class index of the prototype:

$$d_{CDW}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{D} \lambda_{c,j}^2 (x_j - y_j)^2}, \tag{2}$$

where c is the class index of prototype $\mathbf{x}$. Though the above CDW distance metric provides more freedom than the SW one, however, as illustrated in the following example, this distance metric is insufficient to reflect the local specifics in feature space for query patterns in different classes. Fig.1 illustrates a simple three-class classification problem, where the data in each class are uniformly distributed. $(\mathbf{x}^{(1)}, A)$, $(\mathbf{x}^{(2)}, B)$ and $(\mathbf{x}^{(3)}, C)$ are two-dimensional data points in training set T. $\mathbf{y}_1$ and $\mathbf{y}_2$ are the query data to be classified. Considering the classification of data $\mathbf{y}_1$ between Class A and Class B, when calculating the distance of $\mathbf{y}_1$ to $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, intuitively, to avoid classifying it as Class B mistakenly, feature X should be

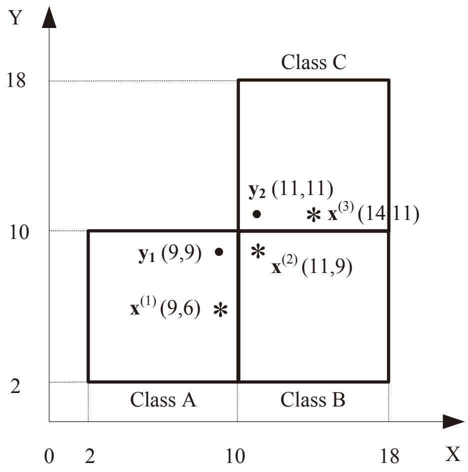

Fig. 1. A three-class classification example.

given a larger weight. However, when classifying data $\mathbf{y}_2$ as Class B or Class C, feature Y should be given a larger weight to determine the distance of $\mathbf{y}_2$ to $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$. In other words, the feature weights should be related to the labels of class pairs to be classified.

Motivated by the above consideration, in this paper we propose a pairwise-weighted (PW) distance metric related to the labels of the class pairs to be classified. Because only two classes are considered for each PW distance metric, the feature weights can be learnt in a more local way. Based on each PW distance metric, a pairwise NN (PNN) classifier can be designed to separate two classes. Then a polychotomous NN classification problem can be solved by fusing several PNN classifiers.

A variety of schemes has been proposed for deriving a combined decision from individual ones, such as majority voting [12], Bayes combination [13], multilayered perceptrons [14]. Considering that the output of each PNN classifier may have uncertainty, in this paper, the PNN classifiers are combined within the framework of Dempster-Shafer theory (DST) [15], [16] due to its well capability of representing and combining uncertain information [17].

The rest of the paper is organized as follows. In Section II, the PW distance metric is defined and a parameter optimization procedure is designed to learn the involved feature weights. Based on the proposed PW distance metric, the corresponding PNN classifiers are combined within the framework of DST in Section III and then two experiments are given to evaluate the performance of the proposed method in Section IV. At last, Section V concludes the paper.

II. Pairwise-Weighted Distance Metric

A. Definition

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

$$d_{PW}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{D} \lambda_{p,q,j}^2 (x_j - y_j)^2}, \tag{3}$$

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

This definition includes, as particular cases, the distance metrics revisited in the introduction. If $\lambda_{p,q,j} = 1$ for all $p = 1, \cdots, M$, $q = 1, \cdots, M$, $j = 1, \cdots, D$, the above distance metric is just the L2 distance metric. Besides, the SW and CDW distance metrics correspond to the cases where the metric weights are irrelevant of the class labels or only dependent on the class label of the first pattern, respectively. Therefore, the above weighted distance metric provides a more general dissimilarity measure than the L2, SW or CDW because the weights depend on the labels of two considered classes.

Remark 1: Compared with the existing distance metrics reviewed above, the PW one provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized. Take the three-class classification problem studied in the introduction for example. In Fig.1, using the PW distance metric, to classify Class B from Class A, $\lambda_{B,A,X}$ (the first two subscripts denote the class labels, the third subscript denotes the feature index) can takes much larger value than $\lambda_{B,A,Y}$, while one can assign smaller value for $\lambda_{B,C,X}$ than $\lambda_{B,C,Y}$ to classify Class B from Class C.

B. Feature Weights Learning

In the previous subsection, the definition of the PW distance metric was given and the advantages of this proposed distance metric were also analyzed. The only open parameters in the PW distance metric are the feature weights related to the labels of the two considered classes. In this subsection we aim to learn the feature weights $\lambda_{p,q,j}$ ($1 \leq p < q \leq M$, $j = 1, \cdots, D$) from the training set via 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 with each other [18].

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

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

where I_k is the set of indices of the training data $\mathbf{x}^{(i)}$ belonging to the class ω_k.

In the following, we consider learning the feature weights $\lambda_{p,q,j}$ ($j = 1, \cdots, D$) from training subsets $\mathcal{T}_p$ and $\mathcal{T}_q$. Let the set of data pairs from the same class be denoted by

$$\begin{aligned} \mathcal{S} = \ & \left\{(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) | m, n \in I_p; m < n\right\} \\ & \cup \left\{(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) | m, n \in I_q; m < n\right\}, \end{aligned}$$

and the set of data pairs from different classes by

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

Following the idea presented in [19], a logistic regression model can be assumed when estimating the probability for any data pair $(\mathbf{x}^{(m)}, \mathbf{x}^{(n)})$ to share the same class

$$\Pr(+|(\mathbf{x}^{(m)}, \mathbf{x}^{(n)})) = \frac{1}{1+\exp\left(d_{PW}^2(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) - \mu_{p,q}\right)}, \tag{4}$$

and then the probability for any data pair $(\mathbf{x}^{(m)}, \mathbf{x}^{(n)})$ to share different classes is

$$\begin{aligned} \Pr(-|(\mathbf{x}^{(m)}, \mathbf{x}^{(n)})) &= 1 - \frac{1}{1+\exp\left(d_{PW}^2(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) - \mu_{p,q}\right)} \\ &= \frac{1}{1+\exp\left(-d_{PW}^2(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) + \mu_{p,q}\right)} \end{aligned}, \tag{5}$$

where, "+" and "−" denote the 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 more probability to be in the same class when their PW distance is much less then the threshold $\mu_{p,q}$. In contrast, if their PW distance is much larger then the 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

$$\begin{aligned} &\mathcal{L}_g\left(\{\lambda_{p,q,j}\}_{j=1}^D, \mu_{p,q}\right) = \log \Pr(+|\mathcal{S}) + \log \Pr(-|\mathcal{D}) \\ &= -\sum_{(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) \in \mathcal{S}} \log\left(1+\exp\left(d_{PW}^2(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) - \mu_{p,q}\right)\right) \\ &- \sum_{(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) \in \mathcal{D}} \log\left(1+\exp\left(-d_{PW}^2(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) + \mu_{p,q}\right)\right) \end{aligned} \tag{6}$$

With the maximum likelihood estimation, the feature weights learning problem can be formulated as the following optimization problem

$$\begin{array}{cl} \max\limits_{\{\lambda_{p,q,j}\}_{j=1}^D, \mu_{p,q}} & \mathcal{L}_g\left(\{\lambda_{p,q,j}\}_{j=1}^D, \mu_{p,q}\right) \\ \text{s.t.} & \lambda_{p,q,j} \geq 0, j = 1, \cdots, D \text{ and } \mu_{p,q} \geq 0. \end{array} \tag{7}$$

The above constrained nonlinear optimization problem can be solved using Newton's method or some existing optimization software packages, such as the Matlab optimization toolbox. In a similar way, we can learn the feature weights $\lambda_{p,q,j}$, $j = 1, \cdots, D$, for all other class pairs $\Omega_{p,q}$ for $1 \leq p < q \leq M$, which are prepared for the following classifying process.

III. Fusion of PNN Classifiers in DST Framework

Base on the proposed PW distance metric concerning class pair $\Omega_{p,q}$, a pairwise NN (PNN) classifier can be designed to separate the two classes Ω_p and Ω_q. For an M-class classification problem, $M(M-1)/2$ PNN classifers $\mathcal{C}_{p,q}$ $(1 \leq p < q \leq M)$ can be designed and the final classification result can be obtained by combining the output of these PNN classifers. A popular method of combining the output of pairwise classifiers is the voting strategy, where each classifier gives a vote for the predicted class and the class with the largest number of votes is predicted. However, a classifier $\mathcal{C}_{p,q}$ is trained to distinguish only between classes Ω_p and Ω_q, thus its vote for an query pattern from a different class should be taken with care. In this section, we aim to overcome this difficulty by modeling the uncertainty related to the output of PNN classifiers in the framework Dempster-Shafer theory (DST).

A. Basics of DST

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

$$\mathrm{m}(\emptyset) = 0 \;\text{ and }\; \sum_{A \in 2^\Omega} \mathrm{m}(A) = 1. \tag{8}$$

Elements $A \in 2^\Omega$ having $\mathrm{m}(A) > 0$ are called *focal elements* of the BBA $\mathrm{m}(\cdot)$. The BBA $\mathrm{m}(A)$ 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 $\mathrm{m}(\Omega)$, is referred to as the degree of global ignorance.

Shafer [16] also defines the *belief function* and *plausibility function* of $A \in 2^\Omega$ as follows

$$\mathrm{Bel}(A) = \sum_{B \subseteq A} \mathrm{m}(B) \text{ and } \mathrm{Pl}(A) = \sum_{B \cap A \neq \emptyset} \mathrm{m}(B). \tag{9}$$

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

Several distinct bodies of evidence characterized by different BBAs can be combined using *Dempster's rule of combination* $\bigoplus$. Mathematically, the Dempster's rule of combination of two BBAs $\mathrm{m}_1(\cdot)$ and $\mathrm{m}_2(\cdot)$ defined on the same frame of discernment Ω is

$$\mathrm{m}(A) = \begin{cases} 0, & \text{for } A = \emptyset \\ \dfrac{\sum\limits_{B,C \in 2^\Omega; B \cap C = A} \mathrm{m}_1(B)\mathrm{m}_2(C)}{1 - \sum\limits_{B,C \in 2^\Omega; B \cap C = \emptyset} \mathrm{m}_1(B)\mathrm{m}_2(C)}, & \text{for } A \in 2^\Omega \setminus \emptyset. \end{cases} \tag{10}$$

As described in [16], Dempster's rule of combination is both commutative and associative.

For decision-making, the maximum plausibility (as defined in Eq.(9)) rule is usually utilized to make the final decision. Suppose $\mathrm{m} = \mathrm{m}_1 \bigoplus \mathrm{m}_2$, then

$$\mathrm{Pl}(\{\omega_i\}) \propto \mathrm{Pl}_1(\{\omega_i\})\mathrm{Pl}_2(\{\omega_i\}), \forall \omega_i \in \Omega. \tag{11}$$

That is, when combining several mass functions, 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.(11) to make the decision equivalently.

In order to manipulate the belief functions more effectively, some probabilistic operations (conditioning, deconditioning, etc) are introduced to DST framework [20]. Conditional beliefs represent knowledge which is valid provided that a hypothesis is satisfied. Let $\mathrm{m}^\Omega(\cdot)$ be a BBA on Ω, $S \subseteq \Omega$ an hypothesis and $\mathrm{m}_S^\Omega(\cdot)$ the categorical BBA such that $\mathrm{m}_S^\Omega(S) = 1$. Then the conditional belief function $\mathrm{m}^\Omega[S](\cdot)$ is

$$\mathrm{m}^\Omega[S] = \mathrm{m}^\Omega \bigoplus \mathrm{m}_S^\Omega. \tag{12}$$

The above operation is referred to as *Dempster's rule of conditioning*. In contrary, if we want to recover $\mathrm{m}^\Omega(\cdot)$ from the conditional belief function $\mathrm{m}^{\Omega}[S](\cdot)$, the following *deconditioning* operation can be used

$$\mathrm{m}^{\Omega}(A \cup \overline{S}) = \mathrm{m}^{\Omega}[S](A), \text{ for } A \in 2^{\Omega} \setminus \emptyset, \tag{13}$$

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

B. Combination of PNN Classifiers

Our aim is to use the Dempster-Shafer theory to model the uncertainty inherent in the pairwise classification. Now, we have a set of PNN classifiers $\mathcal{C}_{p,q}$ $(1 \leq p < q \leq M)$ and we first look for the representation of their output in terms of the DST.

For the output of each PNN classifier $\mathcal{C}_{p,q}$, there are two kinds of uncertainty. The first one is related to the fact that the real class label of query pattern $\mathbf{y}$ may actually not belong to class pair $\Omega_{p,q}$ (called *outer-pair uncertainty*). The second one is that even the real class label of query pattern $\mathbf{y}$ belongs to class pair $\Omega_{p,q}$, affected by the noises of the training patterns, the result of the classifier is not always accurate (called *inner-pair uncertainty*). So, the frame of discernment should be set as $\Omega = \{\omega_1, \cdots, \omega_M\}$, which can characterize both kinds of uncertainty.

For the PNN classifier $\mathcal{C}_{p,q}$, suppose the class label of the nearest neighbor is $\omega_p \in \Omega_{p,q}$. It can be seen as a piece of evidence that support the query pattern $\mathbf{y}$ belongs to ω_p. However, considering the *outer-pair uncertainty*, this piece of evidence is conditioned on the hypothesis $\omega_p \in \Omega_{p,q}$:

$$\mathrm{m}^{\Omega}[\Omega_{p,q}](\{\omega_p\}) = 1. \tag{14}$$

Further, due to the *inner-pair uncertainty*, this piece of evidence does not by itself provide 100% reliability. In DST formalism, this can be expressed by saying that only some part of the belief is committed to ω_p

$$\left\{ \begin{array}{lcl} \mathrm{m}^{\Omega}[\Omega_{p,q}](\{\omega_p\}) & = & \alpha_{p,q} \\ \mathrm{m}^{\Omega}[\Omega_{p,q}](\{\omega_q\}) & = & 1 - \alpha_{p,q}, \end{array} \right. \tag{15}$$

where, $\alpha_{p,q} \in [0, 1]$ represents the reliability that the PNN classifier $\mathcal{C}_{p,q}$ provides the right classification result between classes ω_p and ω_q.

In determining the reliability factor $\alpha_{p,q}$, the probability model in Eq.(4) for data pair from the same class can be used

$$\alpha_{p,q} = \frac{1}{1 + \exp\left(d_{PW}^2 - \mu_{p,q}\right)}, \tag{16}$$

where, d_{PW} denotes the PW distance between the query pattern $\mathbf{y}$ and its nearest neighbor and $\mu_{p,q}$ is the threshold learned in Eq.(7).

In order to combine the output of different PNN classifiers in a uniform framework, the conditional BBA constructed as Eq.(15) should be deconditioned using Eq.(13) as

$$\left\{ \begin{array}{lcl} \mathrm{m}^{\Omega}_{p,q}(\overline{\{\omega_q\}}) = \mathrm{m}^{\Omega}[\Omega_{p,q}](\{\omega_p\}) & = & \alpha_{p,q} \\ \mathrm{m}^{\Omega}_{p,q}(\overline{\{\omega_p\}}) = \mathrm{m}^{\Omega}[\Omega_{p,q}](\{\omega_q\}) & = & 1 - \alpha_{p,q}. \end{array} \right. \tag{17}$$

Because the mass function and plausibility function are in one-to-one correspondence, so we can compute the plausibility function $\mathrm{Pl}_{p,q}(\cdot)$ from the above constructed and deconditioned BBA $\mathrm{m}^{\Omega}_{p,q}(\cdot)$ using Eq.(9) as

$$\mathrm{Pl}_{p,q}(\{\omega_i\}) = \left\{ \begin{array}{ll} \alpha_{p,q}, & \text{if } i = p \\ 1 - \alpha_{p,q}, & \text{if } i = q \\ 1, & \text{others .} \end{array} \right. \tag{18}$$

In order to decrease the computation complexity, instead of combining the $M(M-1)/2$ BBAs $\mathrm{m}^{\Omega}_{p,q}(\cdot)$ $(1 \leq p < q \leq M)$ using Dempsters rule of combination, we can compute the combined plausibility function $\mathrm{Pl}(\cdot)$ directly using Eq.(11) to make the decision as follows

$$\mathrm{Pl}(\{\omega_i\}) \propto \mathrm{Pl}'(\{\omega_i\}) = \prod_{1 \leq p < q \leq M} \mathrm{Pl}_{p,q}(\{\omega_i\}), \forall \omega_i \in \Omega. \tag{19}$$

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

Remark 2: As can be seen from above, the combination process of PNN classifiers is quite time-saving. So, when classifying a query pattern, the time is mainly consumed in the classification process of multiple PNN classifiers. Even though the number of PNN classifiers is of M^2 order (with $M(M-1)/2$ classifiers), each classifier only uses the training samples from the corresponding classes (about $2N/M$ samples averagely). Hence the total number of the computed samples is about $N(M-1)$, which is just $M-1$ times larger than the original NN classifier. To most classification problems, such as the benchmark data sets studied in next section, the number of considered classes is not very large, so the computation cost of the proposed method is not a big problem.

IV. Numerical Experiments

The capabilities of the proposed NN classification method based on the PW distance metric and Dempster-Shafer theory (denoted as PNN-DST) was empirically assessed through two different types of experiments. In the first experiment, a synthetic data set was used to show the behavior of the proposed method in controlled setting. In the second one, several standard benchmark data sets from the well-known UCI Repository of Machine Learning Databases [21] were considered, with the aim to show that the proposed technique is uniformly adequate for a variety of tasks involving different data conditions: large/small data sets, high/low dimensionality, etc.

A. Synthetic Data

A three-class classification problem was used to compare our method with the original NN classifier based on L2 distance metric (L2-NN) [1] and the NN classifier based on CDW distance metric (CDW-NN) [7], [8]. The following class-conditional normal distributions were assumed. Class A: $\mu_A = (6, 6), \Sigma_A = diag(2, 2)$; Class B: $\mu_B = (14, 6), \Sigma_B = diag(2, 2)$; Class C: $\mu_C = (14, 14), \Sigma_C = diag(2, 2)$.

Fig.2 shows classification error rates with different training set sizes, ranging from 5 to 100 prototypes per class. For each size, each classification algorithm runs 100 times with different

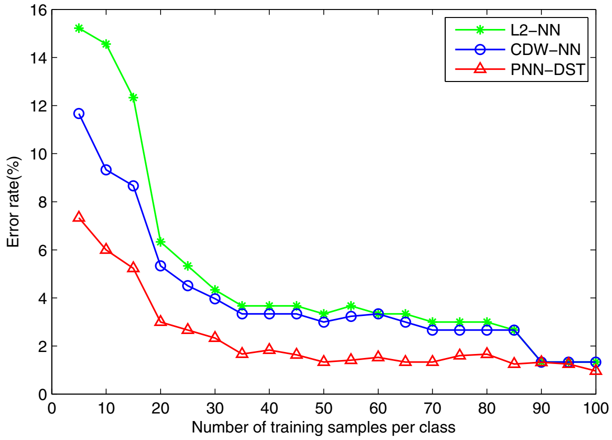

Fig. 2. Classification error rates of our proposed method compared with other methods with different training set sizes.

training sets randomly drawn from the above distributions. A fixed test set of 300 query patterns, independently drawn from the same distributions, is used for error statistics. As can be seen from the result, the CDW-NN classifier is just slightly better than the original L2-NN classifier for small training sets. This is mainly because the CDW distance metric only characterizes the local specifics of the features with respect to the prototypes, while, in this simulation scenario, for each class, the statistical variances of the prototypes in different features (i.e. X axis, Y axis) are almost the same. Under these circumstances, the CDW distance metric asymptotically approximates to the L2 distance metric as the amount of training data increases. The proposed PNN-DST classifier produces the lowest error rate as the PW distance metric provides more local specifics of the features for each PNN sub-classifier and further in the combination process the output uncertainty of those PNN sub-classifiers is well addressed. Besides, the performance improvement is more significant for small training sets, in which case the ideal asymptotical behavior of NN classifier degrades dramatically.

To better illustrate the superiority of the proposed PNN-DST classifier, the classification results of different methods for one Monte Carlo run with 30 training data and 30 test data are given in Fig.3. As can be seen in Fig.3(a), the test data $\mathbf{y_1}, \mathbf{y_2}$ are quite close to the boundary between Class B and Class C, and in this scarce-prototype condition, it is quite difficult to make the right classification. The L2-NN and CDW-NN just misclassify these two data points as shown in Fig.3(b) and (c), because based on L2 and CDW distance metrics their nearest neighbors are the training data $\mathbf{x_1}$ labeled Class C and $\mathbf{x_2}$ labeled Class B, respectively. However, as shown in Fig.3(d), the test data $\mathbf{y_2}$ is correctly classified based on the PNN-DST classifier. The PNN-DST classifier classifies the test data $\mathbf{y_2}$ by combining the results of three PNN sub-classifiers:

$$\begin{aligned}
&\mathrm{Pl}_{A,B}(\{A\}) = 0.27,\ \mathrm{Pl}_{A,B}(\{B\}) = 0.73,\ \mathrm{Pl}_{A,B}(\{C\}) = 1;\\
&\mathrm{Pl}_{A,C}(\{A\}) = 0.29,\ \mathrm{Pl}_{A,C}(\{B\}) = 1,\ \mathrm{Pl}_{A,C}(\{C\}) = 0.71;\\
&\mathrm{Pl}_{B,C}(\{A\}) = 1,\ \mathrm{Pl}_{B,C}(\{B\}) = 0.31,\ \mathrm{Pl}_{B,C}(\{C\}) = 0.69.
\end{aligned}$$

Then after the fusion of multiple PNN sub-classifiers within DST framework, we can get the combined result:

$$\mathrm{Pl}'(\{A\}) = 0.08,\ \mathrm{Pl}'(\{B\}) = 0.23,\ \mathrm{Pl}'(\{C\}) = 0.49.$$

Finally, based on the maximum plausibility rule we get the final classification result Class C with hard partition.

B. *Benchmark Data Sets*

In this second experiment, ten well-known benchmark data sets from UCI repository were used to evaluate the performance of the PNN-DST classifier. The main characteristics of the data sets are summarized in Table I. In order to evaluate the effectiveness of the combination process with DST, apart from the L2-NN and CDW-NN methods, we also consider the method of fusing the PNN classifiers with voting strategy (denoted as PNN-VOTE).

The classification results of the ten benchmark data sets are shown in Table II. The significance of the differences between results is evaluated using a *Mc Nemar test* [22] at level 5%. For each dataset, the best classification accuracy is underlined, and those that are significantly improved than the baseline L2-NN method are printed in bold. As can be seen from these results, the PNN-DST method, presented in this paper, outperforms the L2-NN or CDW-NN for most of the data sets. Additionally, for *Ecoli*, *Glass*, *Vehicle*, *Waveform*, *Wine* and *Yeast* data sets, the improvements are statistically significant than the baseline L2-NN method, because the local distance metric plays more crucial role in determining the NN based classification performance for these scarce-prototype and high-dimensionality cases. On the other hand, the PNN-DST method always performs better than PNN-VOTE, especially for those data sets with small number of classes, because the voting strategy will take great adventure when the total number of votes ($M(M-1)/2$, with M be the number of classes) is small.

V. Concluding Remarks

In order to improve the performance of the NN based classifier in small data set situations, a new distance metric called PW distance metric, has been proposed in this paper. Compared with the existing distance metrics, the PW one provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized. A parameter optimization procedure is designed to learn the feature weights from the training data set. Based on the PW distance metric, a PNN-DST classifier is developed, which combines the output of PNN classifiers using Dempster-Shafer theory. 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 dimensionality and sparse prototypes.

Acknowledgment

This work is partially supported by China Natural Science Foundation (No. 61135001, 61374159) and the Doctorate Foundation of Northwestern Polytechnical University (No. CX201319).

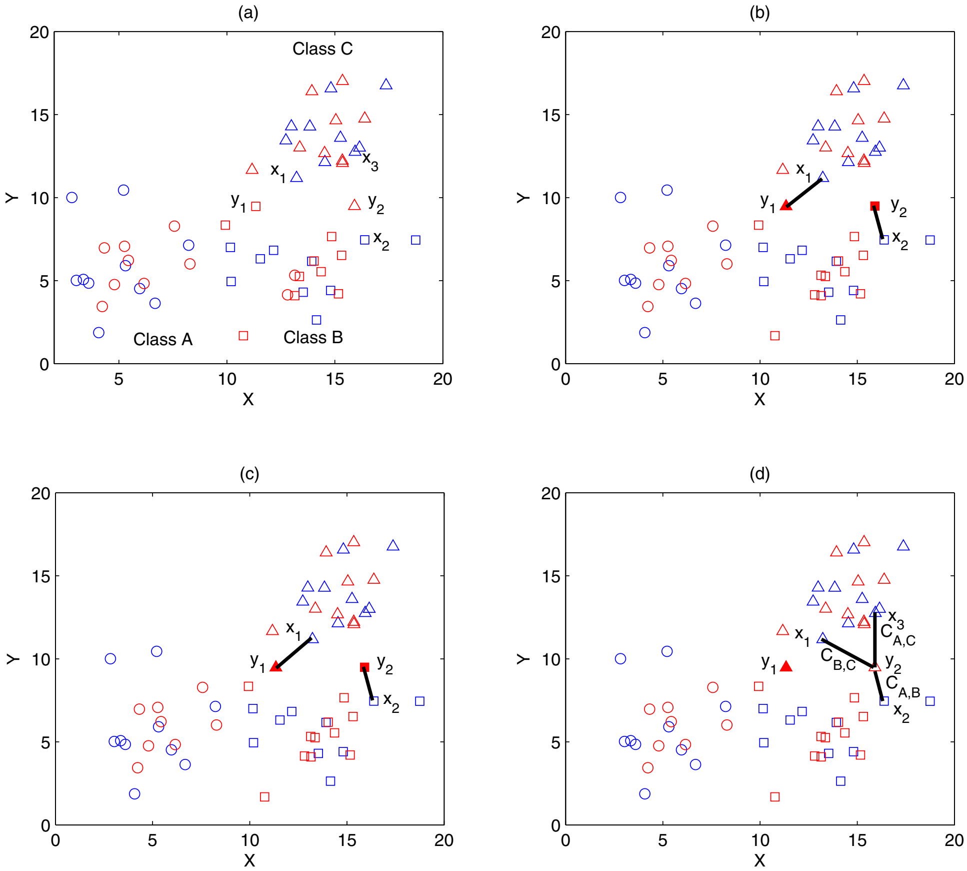

Fig. 3. Classification results for different methods with 30 training data and 30 test data. (a) Training data and test data. (b) Classification results by L2-NN. (c) Classification results by CDW-NN. (d) Classification results by PNN-DST. (The blue makers represent the training data, and the red makers represent the test data, with circle for class A and square for class B and triangle for class C, respectively. The filled makers represent the data mistakenly classified.)

TABLE I. STATISTICS OF THE BENCHMARK DATA SETS USED IN THE EXPERIMENT.

Data set	# of instances	# of features	# of classes	# of training patterns	# of test patterns
Balance	625	4	3	500	125
Ecoli	336	7	8	200	136
Glass	214	9	6	139	75
Letter	20,000	16	26	15,000	5,000
Satimage	6,435	36	6	4,435	2,000
Segment	2,310	19	7	1,400	910
Vehicle	846	18	4	646	200
Waveform	5,000	21	3	3,500	1,500
Wine	178	13	3	75	103
Yeast	1,484	8	10	1000	484

TABLE II. CLASSIFICATION ACCURACY (IN %) OF OUR METHOD IN COMPARISON WITH OTHER NN-BASED METHODS.

Data set	L2-NN	CDW-NN	PNN-VOTE	PNN-DST
Balance	74.40	**79.20**[a]	75.20	77.60
Ecoli	80.88	83.82	**86.76**[b]	**88.24**
Glass	69.33	72.00	72.00	**74.67**
Letter	94.26	94.72	95.40	95.80
Satimage	89.30	88.55	91.55	92.55
Segment	95.05	96.15	96.70	96.92
Vehicle	69.00	69.50	70.50	**74.50**
Waveform	80.53	**83.87**	**84.53**	**85.87**
Wine	76.70	**89.32**	**87.38**	**90.29**
Yeast	51.65	**55.79**	**57.64**	**57.85**

[a] The results underlined correspond to the best accuracy.

[b] The results typeset in boldface are significantly better than the baseline L2-NN method at level 5%.

REFERENCES

[1] E. Fix and J. Hodges, "Discriminatory analysis, nonparametric discrimination: consistency properties," USAF School of Aviation Medicine, Randolph Field, Texas, Tech. Rep. 4, 1951.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[3] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[4] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proceeding of Uncertainty in Artificial Intelligence*, 2003.

[5] Z. Zhang, J. Kwok, and D. Yeung, "Parametric distance metric learning with label information," in *Proceeding of International Joint Conference on Artificial Intelligence*, 2003.

[6] C. F. Eick, A. Rouhana, A. Bagherjeiran, and R. Vilalta, "Using clustering to learn distance functions for supervised similarity assessment," *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 395–401, 2006.

[7] R. Paredes and E. Vidal, "A class-dependent weighted dissimilarity measure for nearest neighbor classication problems," *Pattern Recognition Letters*, vol. 21, pp. 1027–1036, 2000.

[8] ——, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1100–1110, 2006.

[9] J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognition Letters*, vol. 28, pp. 207–213, 2007.

[10] M. Z. Jahromi, E. Parvinnia, and R. John, "A method of learning weighted similarity function to improve the performance of nearest neighbor," *Information Sciences*, vol. 179, pp. 2964–2973, 2009.

[11] L. Jiao, Q. Pan, X. Feng, and F. Yang, "An evidential k-nearest neighbor classification method with weighted attributes," in *Proceeding of 16th International Conference on Information Fusion*, 2013, pp. 145–150.

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

[13] L. Kuncheva, J. Bezdek, and R. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, pp. 299–314, 2001.

[14] C. Y. Suen, Y. S. Huang, and K. Liu, "The combination of multiple classifiers by a neural network approach," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, pp. 579–597, 1995.

[15] A. Dempster, "Upper and lower probabilities induced by multivalued mapping," *Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1967.

[16] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

[17] L. Jiao, Q. Pan, Y. Liang, X. Feng, and F. Yang, "Combining sources of evidence with reliability and importance for decision making," *Cent. Eur. J. Oper. Res.*, pp. In press, doi: 10.1007/s10 100–013–0334–3, 2014.

[18] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *Advances in Neural Information Processing Systems*, vol. 15, pp. 521–528, 2003.

[19] L. Yang and R. Jin, "Distance metric learning: a comprehensive survey," Michigan State University, Tech. Rep., 2006.

[20] P. Smets, "Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem," *International Journal of Approximate Reasoning*, vol. 9, pp. 1–35, 1993.

[21] C. J. Merz, P. M. Murphy, and D. W. Aha, *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, University of California, Irvine. http://www.ics.uci.edu/mlearn/MLRepository.html, 1997.

[22] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.