Neurocomputing 151 (2015) 1468–1476

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric

Lianmeng Jiao [a,b], Quan Pan [a,*], Xiaoxue Feng [a,c]

[a] School of Automation, Northwestern Polytechnical University, 710072 Xi'an, PR China
[b] UMR CNRS 7253, Heudiasyc, Université de Technologie de Compiègne, 60205 Compiègne, France
[c] UMR CNRS 6279, ICD-LM2S, Université de Technologie de Troyes, 10010 Troyes, France

ARTICLE INFO

Article history:
Received 8 February 2014
Received in revised form
24 May 2014
Accepted 13 October 2014
Communicated by Xianbin Cao
Available online 28 October 2014

Keywords:
Pattern classification
Weighted distance metric
Multi-hypothesis nearest-neighbor classifier
Dempster–Shafer theory

ABSTRACT

The performance of nearest-neighbor (NN) classifiers 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 class-conditional weighted (CCW) distance metric related to both the class labels of the prototypes and the query patterns is proposed. Compared with the existing distance metrics, the proposed metric provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized. Based on the proposed CCW distance metric, a multi-hypothesis nearest-neighbor (MHNN) classifier is developed. The scheme of the proposed MHNN classifier is to classify the query pattern under multiple hypotheses in which the nearest-neighbor sub-classifiers can be implemented based on the CCW distance metric. Then the classification results of multiple sub-classifiers are combined to get the final result. Under this general scheme, a specific realization of the MHNN classifier is developed within the framework of Dempster–Shafer theory due to its good capability of representing and combining uncertain information. Two experiments based on synthetic and real data sets were carried out to show the effectiveness of the proposed technique.

1. 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 prototypes) $T=\{(\mathbf{x}^{(1)},\omega^{(1)}),\ldots,(\mathbf{x}^{(N)},\omega^{(N)})\}$ with input vector $\mathbf{x}^{(i)}\in\mathbb{R}^D$ and class label $\omega^{(i)}\in\{\omega_1,\ldots,\omega_M\}$, the NN rule classifies a query pattern $\mathbf{y}\in\mathbb{R}^D$ 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 numbers of prototypes is well known [2]. However, in many practical pattern classification applications, only a small number of prototypes are available. Typically, under such a scarce-prototype framework, the ideal asymptotical behavior of the NN classifier degrades dramatically [3]. This problem has driven 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 and local distance metric learning [4]. The first learns the distance metric in a global sense, i.e., to share the same simple weighted (SW) distance metric for all of 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. Based on the above distance metric, the feature weights learning in [5,6] is formulated as a linear programming 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. [7] introduce an approach to learn the feature weights that maximize the clustering accuracy of objects in the

* Corresponding author at: School of Automation, Northwestern Polytechnical University, 710072 Xi'an, PR China Tel.: +86 29 88431307; fax: +86 29 88431306.
E-mail addresses: lianmeng.jiao@utc.fr (L. Jiao), panquannpu@gmail.com (Q. Pan), xiaoxue.feng@utt.fr (X. Feng).

http://dx.doi.org/10.1016/j.neucom.2014.10.039
0925-2312/

training set, and similarly, the classification error rate of objects in the training set is employed to evaluate the feature weights in [8]. Although the above global distance metric learning methods are intuitively correct, they are too coarse, as the feature weights of the distance metric are irrelevant with the prior-known class labels of the prototypes. 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 less relevant for others) [9]. Thus, many methods [10–14] have been developed to learn a distance metric in a local setting, i.e., the feature weights may be different for different prototypes. The most representative method is the class-dependent weighted (CDW) distance metric proposed by Paredes and Vidal [15,16], 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 $\mathbf{x}$. Although the above CDW distance metric provides more freedom than the SW metric, the following example illustrates that 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 $\mathbf{y}_1$, when calculating the distance between $\mathbf{x}^{(2)}$ and $\mathbf{y}_1$, intuitively, to avoid classifying it to Class B mistakenly, the feature value in the X-axis should be given a larger weight. However, in classifying data $\mathbf{y}_2$, it is reasonable that the feature value in the Y-axis should be given a larger weight to determine the distance between $\mathbf{x}^{(2)}$ and $\mathbf{y}_2$. That is, the feature weights should also be related to the class labels of the query patterns to be classified.

Motivated by the above consideration, in this paper we propose a more general distance metric that associates with both the class labels of the prototypes and the query patterns. As in classification problems the class label of the query pattern is not prior-known, this general distance metric only makes sense when conditioned on the assumption that the query pattern belongs to a specified class. Therefore, we define this type of variant as a class-conditional weighted (CCW) distance metric. Compared with the existing distance metrics mentioned above, the CCW metric provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized.

Based on the CCW distance metric, this paper develops a multi-hypothesis nearest-neighbor (MHNN) classifier. The main scheme

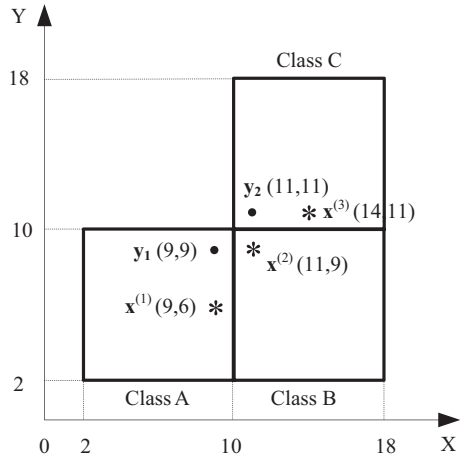


Fig. 1. A three-class classification example.

of this method is to classify the query pattern under multiple hypotheses in which the nearest-neighbor sub-classifiers can be implemented based on the CCW distance metric and then to combine the classification results of multiple sub-classifiers to obtain the final result. A variety of schemes have been proposed for deriving a combined decision from individual decisions, such as majority voting [17], Bayes combination [18], multilayered perceptrons [19], and the Dempster–Shafer theory (DST) [20–22]. In this paper, a specific realization of the MHNN classifier is developed within the framework of DST due to its good capability of representing and combining uncertain information which is always encountered in classification problems.

The rest of this paper is organized as follows. In Section 2, the class-conditional weighted distance metric is defined and then both a heuristic method and a parameter optimization procedure are designed to derive the involved feature weights. The multi-hypothesis nearest-neighbor classifier is designed and realized within the framework of DST in Section 3. Two experiments are given to evaluate the performance of the proposed method in Section 4. Finally, Section 5 concludes the paper.

2. Class-conditional weighted distance metric

2.1. Definition

Before defining the class-conditional weighted distance metric for the purpose of classification, we will first give a general weighted distance metric between two patterns with prior-known class labels as follows.

Definition 1 (*General weighted distance metric*). Suppose $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$ are two D-dimensional patterns with class labels ω_p and ω_q. A general weighted distance metric between $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(n)}$ can be defined as

$$d(\mathbf{x}^{(m)},\mathbf{x}^{(n)})=\sqrt{\sum_{j=1}^{D}\lambda_{p,q,j}^2(x_j^{(m)}-x_j^{(n)})^2}, \tag{3}$$

where $\lambda_{p,q,j}$ is a constant that weights the role of the j-th feature in the distance metric between class ω_p and class ω_q.

This definition includes, as particular cases, the distance metrics revisited in the Introduction. If $\lambda_{p,q,j}=1$ for all $p=1,\ldots,M$, $q=1,\ldots,M$, $j=1,\ldots,D$, the above distance metric is just the L2 distance metric. Moreover, the SW and CDW distance metrics correspond to the cases where the metric weights are not relevant to the class labels or are 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 metrics because the weights depend on both class labels of the two considered patterns.

In NN-based classification, the problem is to calculate the distance between a prototype and a query pattern, while the class label of the latter is not prior-known. So, for the purpose of classification, the above general distance metric only makes sense conditioned on the assumption that the query pattern belongs to a specified class ω_q. We will define this type of variant as follows.

Definition 2 (*Class-conditional weighted distance metric*). Let $T=\{(\mathbf{x}^{(1)},\omega^{(1)}),\ldots,(\mathbf{x}^{(N)},\omega^{(N)})\}$ be a set of prototypes. The class-conditional weighted (CCW) distance metric between a query pattern $\mathbf{y}$ and a prototype $\mathbf{x}\in T$ can be defined as

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

where p is the class index of the prototype $\mathbf{x}$ and q is the hypothesized class index of the query pattern $\mathbf{y}$.

Remark 1. The defined CCW distance metric has some advantages over the existing metrics. The most obvious advantage is that it 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 CCW distance metric, to classify $\mathbf{y}_1$ to Class A, $\lambda_{B,A,X}$ (the first two subscripts denote the class labels, while the third subscript denotes the feature index) can take a 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 $\mathbf{y}_2$ to Class C.

2.2. Heuristically deriving feature weights

In the previous subsection, the definition of the CCW distance metric was given and the advantages of this proposed distance metric were noted. The only open parameters in the CCW distance metric are the feature weights that associate with both the class labels of the prototypes and the query patterns. This section aims to derive the feature weights $\lambda_{p,q,j}$ $(p=1,\ldots,M,\ q=1,\ldots,M,\ j=1,\ldots,D)$ from the training set in a heuristic way.

We divide the training set T into M subsets T_k, $k=1,\ldots,M$, with each T_k containing all of the n_k training data labeled to the same class ω_k:

$$T_k=\{\mathbf{x}^{(i)}\}\ (i\in I_k), \tag{5}$$

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

Let us take into consideration the feature mean vector $\boldsymbol{\mu}_k$ and feature variance vector $\boldsymbol{\Sigma}_k$ estimated on the set T_k

$$\begin{aligned}\boldsymbol{\mu}_k&=\sum_{i\in I_k}\mathbf{x}^{(i)}/n_k,\\ \boldsymbol{\Sigma}_k&=\sum_{i\in I_k}(\mathbf{x}^{(i)}-\boldsymbol{\mu}_k)^2/(n_k-1),\end{aligned} \tag{6}$$

where the feature mean vector $\boldsymbol{\mu}_k$ denotes $\{\mu_{k,1},\ldots,\mu_{k,D}\}$ and the feature variance vector $\boldsymbol{\Sigma}_k$ denotes $\{\sigma^2_{k,1},\ldots,\sigma^2_{k,D}\}$.

Here, the feature weights associated with classes ω_p and ω_q are derived based on the following two commonsense rules:

- The closer the feature centers (approximately represented by the feature mean vectors), the more difficult the classification by this feature.
- The more dispersive the feature distributions (approximately represented by the feature variance vectors), the more difficult the classification by this feature.

Based on the above rules, in determining the CCW distance metric between two different classes ω_p and ω_q $(p\neq q)$, the feature weights should monotonically increase with the distance of the feature centers (i.e., $|\mu_{p,j}-\mu_{q,j}|$) and monotonically decrease with the sum of the feature distributions (i.e., $\sigma_{p,j}+\sigma_{q,j}$). Intuitively, we can construct $\lambda_{p,q,j}$ as

$$\lambda_{p,q,j}=|\mu_{p,j}-\mu_{q,j}|/(\sigma_{p,j}+\sigma_{q,j})\quad \text{for } j=1,\ldots,D. \tag{7}$$

When measuring the CCW distance between two patterns belong to the same class ω_p (i.e., $p=q$), the first commonsense rule no longer takes effect, and the feature weights $\lambda_{p,p,j}$ can be constructed as

$$\lambda_{p,p,j}=1/\sigma_{p,j}\quad \text{for } j=1,\ldots,D. \tag{8}$$

Lastly, the feature weights should be normalized so that $\sum_{j=1}^{D}\lambda_{p,q,j}=1$. Similarly, we can obtain the feature weights $\lambda_{p,q,j}$, $j=1,\ldots,D$, for all other class sets T_p, $p=1,\ldots,M$, and T_q, $q=1,\ldots,M$, which are prepared for the classifying process.

Remark 2. From Eqs. (7)–(8), we can see that $\forall p\in\{1,\ldots,M\}$ and $\forall q\in\{1,\ldots,M\}$, $\lambda_{p,q,j}=\lambda_{q,p,j}$ for all $j=1,\ldots,D$. That is, the proposed CCW metric satisfies the symmetry property, which reduces the number of parameters to $M(M+1)D/2$.

2.3. Feature weight optimization

In the above subsection, a heuristic derivation of feature weights $\lambda_{p,q,j}$ $(p=1,\ldots,M,\ q=1,\ldots,M,\ j=1,\ldots,D)$ was given, and it is supposed that the performance of the classification procedure can be further improved if these parameters are learned from the training set via optimizing certain 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 separate those data pairs from different classes far from each other.

Let the set of data pairs from the same class ω_p be denoted by

$$\mathcal{S}_{p,p}=\{(\mathbf{x}^{(m)},\mathbf{x}^{(n)})|m,n\in I_p;m\neq n\}$$

and the set of data pairs from different classes ω_p and ω_q be denoted by

$$\mathcal{D}_{p,q}=\{(\mathbf{x}^{(m)},\mathbf{x}^{(n)})|m\in I_p;n\in I_q;p\neq q\}.$$

To measure the CCW distance between two patterns belonging to the same class ω_p, the feature weights $\lambda_{p,p,j}$ $(j=1,\ldots,D)$ can be optimized via minimizing the inner-class distance criterion:

$$\begin{aligned}&\min_{\lambda_{p,p,j}}\ \sum_{(\mathbf{x}^{(m)},\mathbf{x}^{(n)})\in\mathcal{S}_{p,p}}d^2_{CCW}(\mathbf{x}^{(m)},\mathbf{x}^{(n)})\\ &\text{s.t.}\quad \lambda_{p,p,j}>0,\quad j=1,\ldots,D\quad\text{and}\quad\sum_{j=1}^{D}\lambda_{p,p,j}=1.\end{aligned} \tag{9}$$

In determining the CCW distance metric between two different classes ω_p and ω_q $(p\neq q)$, the feature weights $\lambda_{p,q,j}$ $(j=1,\ldots,D)$ can be optimized via maximizing the between-class distance criterion

$$\begin{aligned}&\max_{\lambda_{p,q,j}}\ \sum_{(\mathbf{x}^{(m)},\mathbf{x}^{(n)})\in\mathcal{D}_{p,q}}d^2_{CCW}(\mathbf{x}^{(m)},\mathbf{x}^{(n)})\\ &\text{s.t.}\quad \lambda_{p,q,j}>0,\quad j=1,\ldots,D\quad\text{and}\quad\sum_{j=1}^{D}\lambda_{p,q,j}=1.\end{aligned} \tag{10}$$

Remark 3. For the above two optimization problems, the objectives are quadratic functions of the feature weights $\lambda_{p,p,j}$ or $\lambda_{p,q,j}$, and both constraints are easily verified to be convex. Thus, the optimization problems are convex and can be solved using existing optimization software packages, such as the MATLAB optimization toolbox [23]. Moreover, the feature weights derived heuristically in the previous subsection can be used as the initial estimates to shorten the optimization time.

3. Multi-hypothesis nearest-neighbor classifier

As in the classification process the class label of the query pattern is not prior-known, the proposed CCW distance metric should be used based on the hypothesized class label. For this consideration, this section develops a multi-hypothesis nearest-neighbor (MHNN) classifier based on the CCW distance metric. This approach first classifies the query pattern under multiple hypotheses in which the nearest-neighbor sub-classifiers can be implemented based on the proposed CCW distance metric and then combines the classification results of multiple sub-classifiers to obtain the final result. In Section 3.1, the general scheme of the proposed MHNN classifier is illustrated and analyzed, and then a specific realization within the framework of Dempster–Shafer theory is developed in Section 3.2.

3.1. General scheme

The general scheme of the proposed multi-hypothesis nearest-neighbor (MHNN) classifier is shown in Fig. 2, where $C(\mathbf{y})=\omega_i$

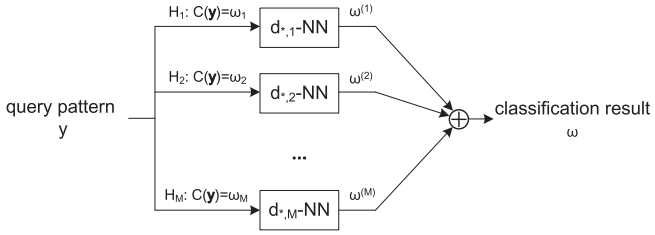


Fig. 2. General scheme of the MHNN classifier.

represents that the class label of **y** is ω_i, and $d_{*,i}$–NN denotes the nearest-neighbor sub-classifier based on the CCW distance metric under hypothesis H_i. The proposed MHNN classifier mainly includes the following two stages:

- *Stage*1: *Nearest-neighbor classification under multiple hypotheses.* M hypotheses $(H_1, H_2, \ldots, H_M)$ are constructed for the class labels of the query pattern **y**. Under each hypothesis H_i, the nearest-neighbor sub-classifier is implemented based on the proposed CCW distance metric with the corresponding weights $\lambda_{p,i,j}$, $p=1,\ldots,M$, $j=1,\ldots,D$.
- *Stage*2: *Combination of multiple sub-classifiers.* The classification results $\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(M)}$ of the M sub-classifiers under different hypotheses are fused to obtain the final result.

Remark 4. As illustrated in Section 2, the proposed CCW distance metric provides more local specifics in feature space than the CDW metric, so the sub-classifier with the correct hypothesis in MHNN will have a lower classification error rate than the existing NN classifier based on CDW distance metric (CDW-NN for short) intuitively. Moreover, the other $M-1$ sub-classifiers with incorrect hypotheses are actually equivalent to CDW-NN because the CCW distance metric reduces to the CDW metric when the hypothesis about the class label of the query pattern fails (in which case, the class label of the query pattern has no effect for either distance metric in determining the feature weights). Therefore, any sub-classifier in MHNN has a better or at least equal classification performance compared with CDW-NN. Thus, it can be expected that through combination, the MHNN classifier will yield a much better performance.

3.2. Realization within the framework of Dempster–Shafer theory

Depending on the different combination methods employed in *Stage*2 of the MHNN classification scheme, different types of realizations can be obtained. In this paper, we combine the output of multiple sub-classifiers within the framework of Dempster–Shafer theory (DST) [24,25], as it provides powerful tools for representing and combining uncertain information.

Using DST to solve a specific problem generally involves three processes: constructing the mass functions (or basic belief assignments) representing the uncertainties in the problem with independent items of evidence, then combining multiple mass functions into a single one, and lastly, making a decision based on the combined result. Thus, after a brief introduction of the basics in DST, we will focus on the construction and combination of the basic belief assignments as well as the decision-making strategies in the MHNN classifier.

3.2.1. Basic concepts in DST

In DST, a problem domain is represented by a finite set $\Theta=\{\theta_1,\theta_2,\ldots,\theta_n\}$ of mutually exclusive and exhaustive hypotheses called the *frame of discernment*. A *mass function* or *basic belief assignment* (BBA) expressing the belief assigned to the elements of 2^Θ by a given source of evidence is a mapping function $m(\cdot): 2^\Theta \to [0,1]$, such that

$$m(\varnothing)=0 \text{ and } \sum_{A\in 2^\Theta} m(A)=1. \tag{11}$$

Elements $A\in 2^\Theta$ having $m(A)>0$ are called *focal elements* of the BBA $m(\cdot)$. The BBA $m(A)$ 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 all of the subsets of 2^Θ is summed to unity, and there is no belief left to the empty set. The belief assigned to Θ, or $m(\Theta)$, is referred to as the degree of global ignorance.

Shafer [25] also defines the *belief function* and *plausibility function* of $A\in 2^\Theta$ as follows:

$$\mathrm{Bel}(A)=\sum_{B\subseteq A} m(B) \quad \text{and} \quad \mathrm{Pl}(A)=\sum_{B\cap A\neq\varnothing} m(B). \tag{12}$$

$\mathrm{Bel}(A)$ represents the exact support to A and its subsets, and $\mathrm{Pl}(A)$ represents all of the possible support to A and its subsets. The interval $[\mathrm{Bel}(A), \mathrm{Pl}(A)]$ can be seen as the lower and upper bounds of support to A. The belief functions $m(\cdot)$, $\mathrm{Bel}(\cdot)$ and $\mathrm{Pl}(\cdot)$ are in one-to-one correspondence.

For decision-making support, Smets [26] proposed the *pignistic probability* $\mathrm{BetP}(A)$[1] to approximate the unknown probability in $[\mathrm{Bel}(A), \mathrm{Pl}(A)]$, given by

$$\mathrm{BetP}(A)=\sum_{\substack{B\subseteq\Theta\\ A\cap B\neq\varnothing}} \frac{|A\cap B|}{|B|} m(B), \tag{13}$$

where $|X|$ stands for the cardinality of the set X.

Several distinct bodies of evidence characterized by different BBAs can be combined using Dempster's rule. Mathematically, Dempster's rule of combination of two BBAs $m_1(\cdot)$ and $m_2(\cdot)$ defined on the same frame of discernment Θ is

$$m(A)=\begin{cases} 0 & \text{for } A=\varnothing \\ \dfrac{\sum_{B,C\in 2^\Theta; B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B,C\in 2^\Theta; B\cap C=\varnothing} m_1(B)m_2(C)} & \text{for } A\in 2^\Theta \quad \text{and} \quad A\neq\varnothing. \end{cases} \tag{14}$$

As described in [25], Dempster's rule of combination is both commutative and associative.

3.2.2. The construction of BBAs in the MHNN classifier

This section aims to construct the BBAs from the M sub-classifier output $\omega^{(i)}, i=1,\ldots,M$. From the view of DST, the class set $\Omega=\{\omega_1,\ldots,\omega_M\}$ can be regarded as the frame of discernment of the problem.

Being affected by the noises of the patterns, the classification results of the sub-classifiers do not always have full accuracy. For any sub-classifier under hypothesis H_i, the result $\omega^{(i)}=\omega_q$ can be regarded as a piece of evidence that increases the belief that the query pattern **y** belongs to ω_q. However, this piece of evidence does not by itself provide 100% certainty. In DST formalism, this can be expressed by saying that only some part of the belief is committed to ω_q. Because $\omega^{(i)}=\omega_q$ does not point to any other particular class, the rest of the belief should be assigned to the frame of discernment Ω representing global ignorance. Therefore, this item of evidence can be represented by a BBA $m(\cdot|H_i)$ verifying:

$$\begin{cases} m(\{\omega_q\}|H_i)=\alpha_i \\ m(\Omega|H_i)=1-\alpha_i \\ m(A|H_i)=0, \quad \forall A\in 2^\Omega\setminus\{\Omega,\{\omega_q\}\}, \end{cases} \tag{15}$$

[1] From the Latin word *pignus* meaning a bet.

where $\alpha_i \in [0,1]$ represents the belief that the sub-classifier under the hypothesis H_i provides the correct classification result.

Remark 5. Eq. (15) can be seen as a discounting operation that discounts the classification result of each sub-classifier with its classification reliability. After discounting, only some part of the belief is assigned to the classification result of the sub-classifier and the rest of the belief is assigned to the frame of discernment which represents global ignorance. Thus, with the discounting operation, the conflict among different pieces of evidence from different sub-classifiers can be highly reduced.

In determining the belief factors $\alpha_i, i=1,\ldots,M$, we divide the M sub-classifiers into two groups: sub-classifier with the correct hypothesis (i.e., $\omega^{(i)}=\omega_i$), and those with incorrect hypotheses (i.e., $\omega^{(i)}\neq\omega_i$). As indicated in Section 3.1, the sub-classifiers with incorrect hypotheses reduce to CDW-NN and thus, on average, have larger classification error rates than the one with the correct hypothesis which is based on the proper CCW distance metric. The belief factors can therefore be determined as

$$\alpha_i = \begin{cases} \alpha^{CCW} & \text{if } \omega^{(i)}=\omega_i \\ \alpha^{CDW} & \text{if } \omega^{(i)}\neq\omega_i \end{cases} \text{for } i=1,\ldots,M, \tag{16}$$

where α^{CCW} and α^{CDW} denote the classification accuracy based on the CCW and CDW distance metrics, respectively. These parameters can be obtained via the leave-one-out (LOO) test[2] on the training set.

3.2.3. The combination of BBAs in the MHNN classifier

To make a decision regarding the output of the M sub-classifiers, the corresponding BBAs constructed in the above part can be combined using Dempster's rule. However, as indicated in [27], the direct use of the Dempster's rule will result in an exponential increase in computational complexity for the reason of enumerating all subsets or supersets of a given subset A of Ω, and the operation becomes impractical when the frame of discernment has more than 15–20 elements. The following part is intended to develop an operational algorithm for evidence combination, which reduces the computational complexity to linear time considering the fact that the focal elements of each associated BBA are all singletons except the ignorance set Ω.

Define $I(i)$ as the set of the former i hypotheses

$$I(i)=\{H_1,H_2,\ldots,H_i\}. \tag{17}$$

Let $m(\cdot|I(i))$ be the BBA after combining all of the former i BBAs associated with $I(i)$. Given the above definitions, a recursive evidence combination algorithm can be developed as follows:

$$\begin{aligned} m(\{\omega_q\}|I(i+1)) &= K_{I(i+1)}\left[m(\{\omega_q\}|I(i))\cdot m(\{\omega_q\}|H_{i+1}) + m(\Omega|I(i))\cdot m\right. \\ &\left.\times(\{\omega_q\}|H_{i+1}) + m(\{\omega_q\}|I(i))\cdot m(\Omega|H_{i+1})\right], \quad q=1,2,\ldots,M \end{aligned}$$

$$\begin{aligned} m(\Omega|I(i+1)) &= K_{I(i+1)}\left[m(\Omega|I(i))\cdot m(\Omega|H_{i+1})\right] \\ K_{I(i+1)} &= \left[1-\sum_{j=1}^{M}\sum_{p=1,p\neq j}^{M} m(\{\omega_j\}|I(i))\cdot m(\{\omega_p\}|H_{i+1})\right]^{-1} \\ i &= 1,2,\ldots,M-1, \end{aligned} \tag{18}$$

where $K_{I(i+1)}$ is a normalizing factor, so that $\sum_{q=1}^{M} m(\{\omega_q\}|I(i+1)) + m(\Omega|I(i+1)) = 1$.

Note that $m(\{\omega_q\}|I(1)) = m(\{\omega_q\}|H_1)$ for $q=1,2,\ldots,M$ and $m(\Omega|I(1)) = m(\Omega|H_1)$, so this recursive evidence combination algorithm can initiate with the BBA under the first hypothesis H_1. Accordingly, when the recursive index i comes to $M-1$, the final results $m(\{\omega_q\}|I(M))$ and $m(\Omega|I(M))$ ($m(\{\omega_q\})$ and $m(\Omega)$ for short, respectively) are obtained by combining all of the M BBAs.

3.2.4. Decision-making strategies

For decision-making with hard partition, the belief function Bel(·), plausibility function Pl(·) and pignistic probability BetP(·) are common alternatives. Because the focal elements of the combined BBA $m(\cdot)$ are all singletons except the ignorance set Ω, the credibility, plausibility and pignistic probability of each class ω_q are derived as follows:

$$\begin{aligned} \mathrm{Bel}(\{\omega_q\}) &= m(\{\omega_q\}) \\ \mathrm{Pl}(\{\omega_q\}) &= m(\{\omega_q\}) + m(\Omega) \\ \mathrm{BetP}(\{\omega_q\}) &= m(\{\omega_q\}) + \frac{m(\Omega)}{M} \end{aligned} \tag{19}$$

for $q=1,2,\ldots,M$. It is supposed that, based on this evidential body, a decision has to be made assigning query pattern $\mathbf{y}$ to one of the classes in Ω. Denote A_q the action of assigning $\mathbf{y}$ to class ω_q. Further suppose the loss incurred in the case of wrong classification is equal to one, while the loss corresponding to correct classification is equal to zero. Then, the lower, upper and pignistic expected losses [28] associated to action A_q are given as follows:

$$\begin{aligned} R_*(A_q) &= 1-\mathrm{Bel}(\{\omega_q\}) \\ R^*(A_q) &= 1-\mathrm{Pl}(\{\omega_q\}) \\ R_{bet}(A_q) &= 1-\mathrm{BetP}(\{\omega_q\}). \end{aligned} \tag{20}$$

Because of the particular structure of the combined BBA (i.e., the focal elements are either singletons or the whole frame Ω), it can be easily discovered that

$$\begin{aligned} \omega &= \arg\min_{\omega_q\in\Omega} R_*(A_q) \\ &= \arg\min_{\omega_q\in\Omega} R^*(A_q) \\ &= \arg\min_{\omega_q\in\Omega} R_{bet}(A_q) \\ &= \arg\max_{\omega_q\in\Omega} m(\{\omega_q\}). \end{aligned} \tag{21}$$

That is, the three strategies minimizing R_*, R^*, and R_{bet} lead to the same decision, in which case the pattern is assigned to the class with the maximum basic belief assignment.

3.2.5. Some merits of the realization

Combining multiple sub-classifiers within the framework of DST has some potential benefits.

First, due to the noises of the patterns and the unfitness of the distance metrics, great uncertainty may exist in the classification results of the multiple sub-classifiers. Compared with the previously mentioned majority voting, Bayes combination and multi-layered perceptions methods, the DST framework provides more powerful tools for representing and combining uncertain information [21]. On the one hand, by representing each sub-classifier output with a BBA structure, different types of uncertainty (probability uncertainty and ignorance) can be well characterized. On the other hand, with Dempster's rule of combination, the uncertain information of multiple pieces of evidence can be well considered to obtain the final results.

Second, this method can tackle the classification problem in which the class memberships of the training data are only partially known.[3] Well-labeled data are often difficult to obtain due to limitations of the underlying equipment, partial or uncertain responses in surveys, and so on [29,30]. In this case, the classification results of the sub-classifiers will have imprecise and/or

[2] In the LOO test, one sample in the training set is selected randomly and is classified based on the remaining training set. This procedure is repeated until all the samples in the training set have been tested.

[3] In this case, the class membership of training data $\mathbf{x}^{(i)}$ has the form $\omega^{(i)}=\{(\omega_1,p_{1,i}),\ldots,(\omega_M,p_{M,i})\}$, where $p_{q,i}$ is the probability that the training data $\mathbf{x}^{(i)}$ belongs to class ω_q.

uncertain soft labels. Because both soft and crisp labels can easily be recast as BBAs, the uncertainty of the training sample can be well addressed within the framework of DST.

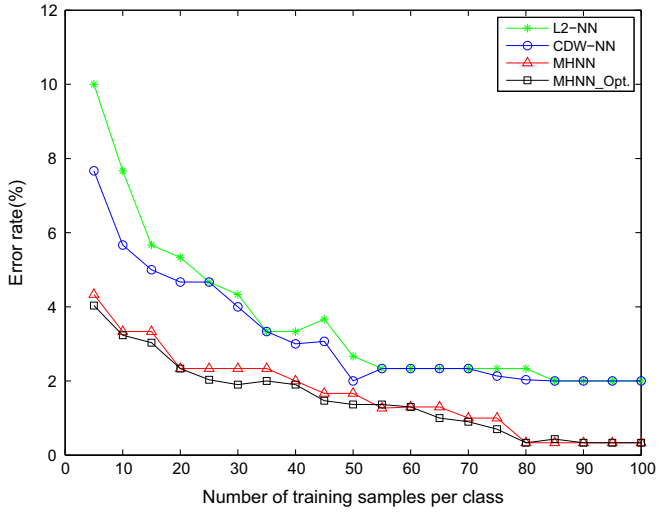


Fig. 3. Classification error rates of our proposed method in comparison with other methods with different training set sizes.

Third, the combined result based on DST gives more freedom to make decisions. In addition to the hard partition studied above, the soft partition [31] is occasionally also needed when the conditional error of making a hard decision is high. In some applications (especially those related to defense and security, as in target classification [32]), it is better to obtain a robust (and eventually partially imprecise) result that will become more precise with additional techniques than to obtain a precise result directly with high risk [33,34]. Fortunately, the DST-based sub-classifier combination result is expressed in the form of BBA, which is originally a type of soft partition.

Fourth, this DST-based sub-classifier combination method can be applied to classify a query pattern with priori information if it is available, such as when a query pattern belonging to some classes with larger probability is known in advance. For our method, the priori information can be used to make decisions by providing an additional sub-classifier with a proper output in the form of BBA, and this priori information can take effect in the combination process.

4. Experiments

The capabilities of the proposed MHNN classifier based on the CCW distance metric will be empirically assessed through two

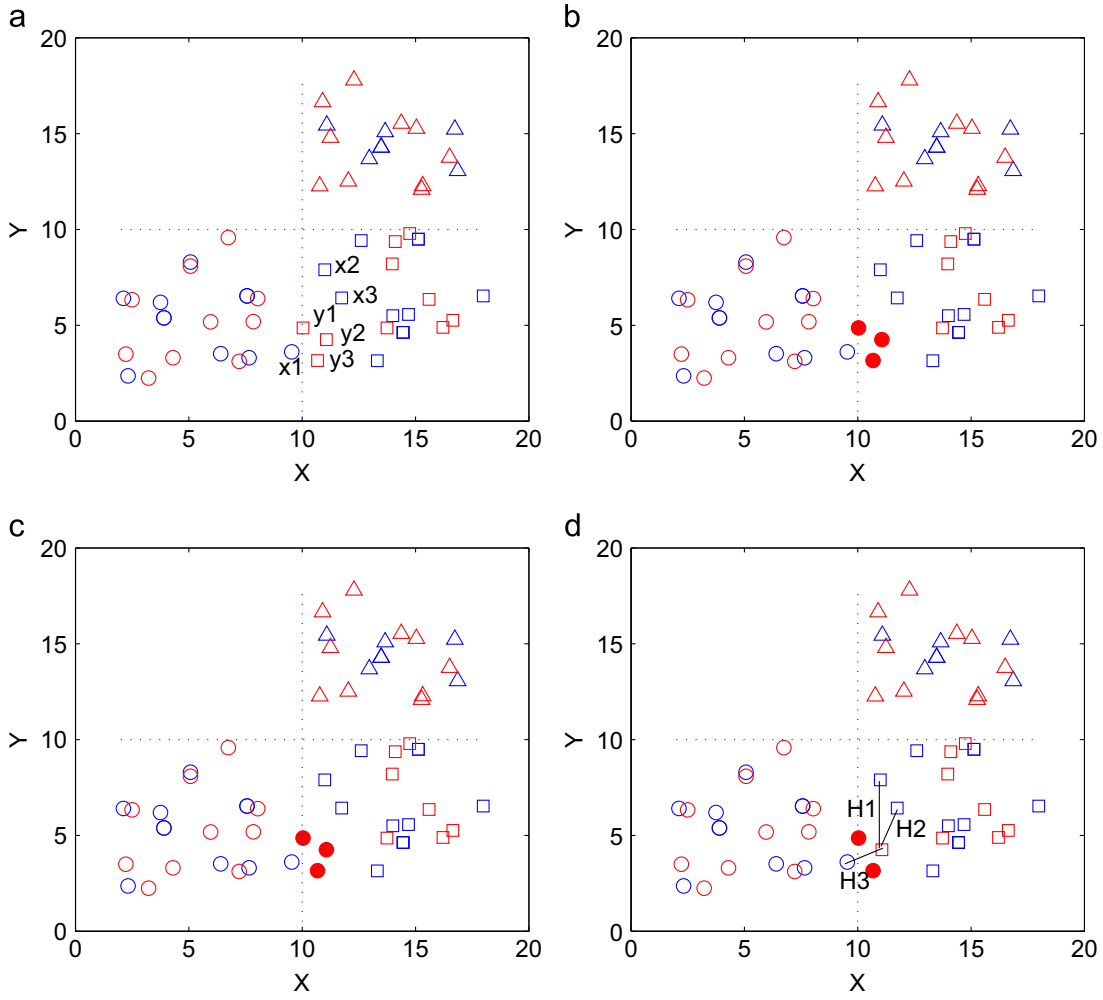


Fig. 4. Classification results for different methods with 30 training data and 30 test data. (a) Training data and test data. (b) Classification results by L2-NN. (c) Classification results by CDW-NN. (d) Classification results by MHNN. (The blue makers represent the training data, and the red makers 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.) (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

different types of experiments. In the first experiment, a synthetic data set is used to show the behavior of the proposed approach in a controlled setting. In the second experiment, several standard benchmark data sets from the well-known UCI Repository of Machine Learning Databases [35] are considered to show that the proposed technique is uniformly adequate for a variety of tasks involving different data conditions, such as large/small training sets and large/small dimensionality.

4.1. Synthetic data

The three-class classification problem mentioned in the Introduction is evaluated here to compare our method with the original NN classifier based on the L2 distance metric (L2-NN) [1] and the NN classifier based on the CDW distance metric (CDW-NN) [15,16]. The following class-conditional uniform distributions are assumed. Class A: $[2,10]\times[2,10]$; Class B: $[10,18]\times[2,10]$; and Class C: $[10,18]\times[10,18]$.

Fig. 3 shows classification error rates of L2-NN, CDW-NN, MHNN and MHNN with optimized feature weights (MHNN_Opt.) for different training set sizes. For each size, each classification algorithm runs 100 times with different 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 seen from the result, the CDW-NN classifier is only 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 particular simulation scenario, for each class the statistical variances of the prototypes in different features (i.e., *X*-axis and *Y*-axis) are nearly the same. Under this circumstance, the CDW distance metric becomes asymptotically approximate to the L2 distance metric as the amount of training data increases. The proposed MHNN classifier produces a much lower error rate, as the CCW distance metric provides more local specifics of the features related to both the prototypes and the query patterns. Moreover, the performance improvement is particularly significant for small training sets, in which case the ideal asymptotical behavior of NN classifier degrades dramatically. We can also see that through feature weight optimization, the performance of the proposed method can be further improved, but not significantly. Thus, for highly time-constrained classification problems, the heuristic feature weight derivation method may be a better choice for considering both the classification rate and the computation burden.

To better illustrate the superiority of the proposed MHNN classifier, the classification results of different methods for one Monte Carlo run with 30 training data and 30 test data are given in Fig. 4. As can be seen in Fig. 4(a), the test data $\mathbf{y1}, \mathbf{y2}, \mathbf{y3}$ are quite close to the boundary between Class A and Class B, and in this scarce-prototype condition, it is quite difficult to make the right classification. The L2-NN and CDW-NN just classify these three data points to Class A as shown in Fig. 4(b) and (c) because, based on L2 and CDW distance metrics, their nearest neighbors are the same training data **x1** labeled Class A. However, as shown in Fig. 4(d), the test data **y2** is correctly classified based on the MHNN classifier. The MHNN classifier classifies the test data **y2** by combining the results of three sub-classifiers under the hypothesis $H_1: C(\mathbf{y2})=A$, $H_2: C(\mathbf{y2})=B$, and $H_3: C(\mathbf{y2})=C$, respectively. For hypothesis H_2(the right hypothesis), in calculating the distance between test point **y2** and the training points in Class B, the value in the *X*-axis has a nearly equivalent weight with that in the *Y*-axis ($\lambda_{B,B,X}=0.55$ and $\lambda_{B,B,Y}=0.45$), while in calculating the distance with the training points in Class A, the value in the *X*-axis has a much larger weight than that in the *Y*-axis ($\lambda_{B,A,X}=0.87$ and $\lambda_{B,A,Y}=0.13$). With this distance metric, the nearest neighbor of the test point **y2** is **x3** labeled Class B, which is quite different from the results based on the L2 or CDW distance measures. Accordingly, the nearest neighbors of the test point **y2** under hypotheses H_1 and H_3 are **x2** labeled Class B and **x1** labeled Class A, respectively. Then, in the combination process of multiple sub-classifiers within the DST framework, the result under the correct hypothesis H_2 is assigned larger belief ($\alpha^{CCW}=0.97$ by LOO test on the training set) than the results under incorrect hypotheses ($\alpha^{CDW}=0.9$ by the same way). Lastly, based on the combined BBA, we obtain the classification result Class B with hard partition.

4.2. Benchmark data sets

In this second experiment, 10 well-known benchmark data sets from UCI repository are used to evaluate the performance of the

Table 1
Statistics of the benchmark data sets used in the experiment.

Data set	# of instances	# of features	# of classes
Balance	625	4	3
Cancer[a]	683	9	2
Diabetes	768	8	2
Glass	214	9	6
Letter	20,000	16	26
Liver	345	6	2
Satimage	6435	36	6
Segment	2310	19	7
Vehicle	846	18	4
Wine	178	13	3

[a] For *Cancer* data set, the samples with missing feature values are discarded.

Table 2
Classification accuracy (%) of our proposed method in comparison with other NN-based methods.

Data set	L2-NN	CDW-NN	MHNN
Balance	75.15	**80.12**[a]	**85.56**[b]
Cancer	95.24	95.48	98.06
Diabetes	70.05	69.96	**75.33**
Glass	69.36	71.15	73.72
Letter	95.15	96.60	96.08
Liver	64.36	60.88	**68.12**
Satimage	89.09	88.45	90.29
Segment	95.42	95.90	96.92
Vehicle	69.42	69.55	73.60
Wine	76.45	**89.40**	**94.38**

[a] The results typeset in boldface are significantly better over the baseline L2-NN method at the 5% level.
[b] The results underlined correspond to the best accuracy.

Table 3
Classification CPU time (s) of our proposed method in comparison with other NN-based methods.

Data set	L2-NN	CDW-NN	MHNN
Balance	0.5616	0.9360	1.8308
Cancer	0.5772	1.0140	1.3472
Diabetes	0.9984	1.4976	2.1840
Glass	0.0836	0.1248	0.5192
Letter	311.20	724.45	8325.4
Liver	0.1560	0.2496	0.3358
Satimage	53.414	88.450	340.22
Segment	6.5520	14.945	48.965
Vehicle	0.8580	1.8720	3.8360
Wine	0.0468	0.0780	0.1516

Table 4
Classification accuracy (%) of our proposed method in comparison with other well-known methods.

Data set	ALLOC80	CART	C4.5	Discrim	NBayes	Quadisc	Cal5	MHNN
Diabetes	69.9(8)[a]	74.5(4)	73.0(7)	77.5(1)	73.8(5)	73.8(5)	75.0(3)	75.3(2)
Letter	93.6(2)	–	86.8(4)	69.8(6)	47.1(7)	88.7(3)	74.7(5)	96.1(1)
Satimage	86.8(2)	86.2(3)	85.0(4)	82.9(7)	71.3(8)	84.5(6)	84.9(5)	90.3(1)
Segment	97.0(1)	96.0(3)	96.0(3)	88.4(6)	73.5(8)	84.3(7)	93.8(5)	96.9(2)
Vehicle	82.7(2)	76.5(4)	73.4(6)	78.4(3)	44.2(8)	85.0(1)	72.1(7)	73.6(5)

[a] The numbers in brackets denote the accuracy orders of the eight methods.

MHNN classifier. The main characteristics of the 10 data sets are summarized in Table 1.

As displayed in Table 1, some UCI data sets are small. In these cases, *B-Fold Cross-Validation* (B-CV) [36] is applied to estimate the error rates by different classification methods. Each data set is divided into B blocks using $B-1$ 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 simplest 2-CV here, because it has the advantage that both the training and test sets are large, and each sample is used for both training and testing on each fold. The 2-CV test is repeated 10 times, and the average classification result on the test data is calculated. For the relatively larger data sets, such as *Letter* and *Satimage*, the single partition into training and test sets specified in the UCI repository is adopted.

The classification results of the 10 benchmark data sets are shown in Table 2. The significance of the differences between results is evaluated using a *Mc Nemar* test [37] at the 5% level. For each data set, the best classification accuracy is underlined, and those that are significantly improved over the baseline L2-NN method are printed in bold. As seen from these results, the MHNN classifier (with feature weights derived heuristically) presented in this paper outperforms the baseline L2-NN or CDM-NN in most of the data sets. Moreover, for the *Balance*, *Diabetes*, *Liver*, and *Wine* data sets, the improvements are statistically significant because the local distance metric plays a more crucial role in determining the NN-based classification performance for these scarce-prototype and large-dimensionality cases.

To evaluate the computational consumption of the proposed MHNN method, the average classification CPU times of the different methods are listed in Table 3. The numerical experiments are executed by MATLAB7.12 in an HP EliteBook 8570p with an Intel (R) Core(TM) i7-3540 M CPU @3.00 GHz and 8 GB memory. From the results, it can be seen that the proposed MHNN classifier has a relatively higher computational consumption, approximately M (the total number of classes) times higher than the original L2-NN classifier (because the MHNN classifier is quite time-saving in both the heuristic feature weight derivation and the multiple sub-classifier combination process, the time is mainly consumed in the classification process under multiple hypotheses). Fortunately, for most classification problems, such as the benchmark data sets studied here, the number of considered classes is not very large, so the computation cost of the MHNN classifier is not a significant problem.

To verify the performance of the proposed method more generally, we compare our results against several state-of-the-art classification methods tested in [38]. These methods can be categorized into decision tree based methods (CART, Cal5, and C4.5) and statistical methods (ALLOC80, Discrim, NBayes, and Quadisc). For a complete description of these methods, the reader may refer to [38]. Table 4 compares our results with those using the above methods on several data sets under the same experimental arrangement.[4] From these comparisons, it can be seen that the proposed MHNN classifier exhibits a uniformly good behavior for all of the data sets, while other procedures may work very well for some data sets but typically tend to worsen (dramatically in many cases) for the rest.

[4] Only those methods that have results in many data sets, and those data sets for which results with many methods are available have been chosen for the comparisons in Table 4.

5. Conclusion

To improve the performance of the NN-based classifier in small data set situations, a new distance metric, called the CCW distance metric, is proposed in this paper. Compared with the existing distance metrics, the CCW metric provides more flexibility to design the feature weights so that the local specifics in feature space can be well characterized. Based on the CCW distance metric, a MHNN classifier is developed that mainly includes two stages. In the first stage, the query pattern is classified under multiple hypotheses in which the nearest-neighbor sub-classifiers can be implemented based on the proposed CCW distance metric. In the second stage, the classification results of multiple sub-classifiers are combined within the framework of DST to obtain the final result. From the results reported in the last section, we can conclude that the proposed technique achieved a uniformly good performance when applied to a variety of classification tasks, including those with high dimensionality and sparse prototypes.

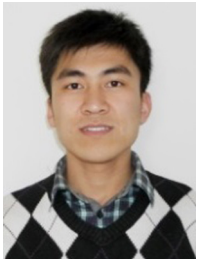
Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading and in-depth criticisms and suggestions. This work is partially supported by China Natural Science Foundation (Nos. 61135001, 61374159 and 61403310) and the Doctorate Foundation of Northwestern Polytechnical University (No. CX201319).

References

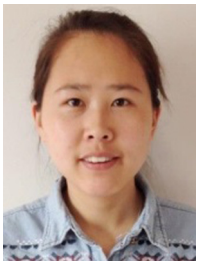
[1] E. Fix, J. Hodges, Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties, Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
[2] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[3] L. Devroye, L. Gyorfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer-Verlag, New York, 1996.
[4] W. Wang, B. Hu, Z. Wang, Globality and locality incorporation in distance metric learning, Neurocomputing 129 (2014) 185–198.
[5] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning distance functions using equivalence relations, in: Proceedings of the Uncertainty in Artificial Intelligence, 2003, pp. 11–13.
[6] Z. Zhang, J. Kwok, D. Yeung, Parametric distance metric learning with label information, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2003, pp. 1450–1452.
[7] C.F. Eick, A. Rouhana, A. Bagherjeiran, R. Vilalta, Using clustering to learn distance functions for supervised similarity assessment, Eng. Appl. Artif. Intell. 19 (2006) 395–401.
[8] E.H. Han, G. Karypis, V. Kumar, Text categorization using weight-adjusted nearest-neighbor classification, in: Proceedings of the Knowledge Discovery and Data Mining, 2001, pp. 53–65.
[9] J.H. Friedman, Flexible Metric Nearest Neighbor Classification, Technical Report, Department of Statistics, Stanford University, 1994.
[10] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1281–1285.

[11] J. Wang, P. Neskovic, L.N. Cooper, Improving nearest neighbor rule with a simple adaptive distance measure, Pattern Recognit. Lett. 28 (2007) 207–213.
[12] M.Z. Jahromi, E. Parvinnia, R. John, A method of learning weighted similarity function to improve the performance of nearest neighbor, Inf. Sci. 179 (2009) 2964–2973.
[13] H. Zhang, J. Yu, M. Wang, Y. Liu, Semi-supervised distance metric learning based on local linear regression for data clustering, Neurocomputing 93 (2012) 100–105.
[14] L. Jiao, Q. Pan, X. Feng, F. Yang, An evidential k-nearest neighbor classification method with weighted attributes, in: Proceedings of the 16th International Conference on Information Fusion, 2013, pp. 145–150.
[15] R. Paredes, E. Vidal, A class-dependent weighted dissimilarity measure for nearest neighbor classification problems, Pattern Recognit. Lett. 21 (2000) 1027–1036.
[16] R. Paredes, E. Vidal, Learning weighted metrics to minimize nearest-neighbor classification error, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1100–1110.
[17] D. Ruta, G. Gabrys, Classifier selection for majority voting, Inf. Fusion 6 (2005) 63–81.
[18] L. Kuncheva, J. Bezdek, R. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognit. 34 (2001) 299–314.
[19] C.Y. Suen, Y.S. Huang, K. Liu, The combination of multiple classifiers by a neural network approach, Int. J. Pattern Recognit. Artif. Intell. 9 (1995) 579–597.
[20] B. Quost, T. Denœux, M.-H. Masson, Classifier fusion in the Dempster–Shafer framework using optimized t-norm based combination rules, Int. J. Approx. Reason. 52 (2011) 353–1374.
[21] M. Tabassian, R. Ghaderi, R. Ebrahimpour, Combination of multiple diverse classifiers using belief functions for handling data with imperfect labels, Expert Syst. Appl. 39 (2012) 1698–1707.
[22] M.P. Naeini, B. Moshiri, B.N. Araabi, M. Sadeghi, Learning by abstraction: hierarchical classification model using evidential theoretic approach and Bayesian ensemble model, Neurocomputing 130 (2014) 73–82.
[23] T. Coleman, M.A. Branch, A. Grace, Optimization Toolbox—For Use With MATLAB, The Mathworks Inc., Natick, MA, 1999.
[24] A. Dempster, Upper and lower probabilities induced by multivalued mapping, Ann. Math. Stat. 38 (1967) 325–339.
[25] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.
[26] P. Smets, Decision making in the TBM: the necessity of the pignistic transformation, Int. J. Approx. Reason. 38 (2005) 133–147.
[27] N. Wilson, Algorithms for Dempster–Shafer theory, in: D.M. Gabbay, P. Smets (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management, Kluwer, Boston, MA, 2000, pp. 421–475.
[28] T. Denœux, A k-nearest neighbor classification rule based on Dempster–Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (1995) 804–813.
[29] T. Denœux, Maximum likelihood estimation from uncertain data in the belief function framework, IEEE Trans. Knowl. Data Eng. 25 (2013) 119–130.
[30] L. Jiao, Q. Pan, Y. Liang, X. Feng, F. Yang, Combining sources of evidence with reliability and importance for decision making, Cent. Eur. J. Oper. Res., 2013, in press, http://dx.doi.org/10.1007/s10100-013-0334-3.
[31] C.K. Chow, On optimum recognition error and reject tradeoff, IEEE Trans. Inf. Theory 16 (1970) 41–46.
[32] L. Jiao, Q. Pan, X. Feng, F. Yang, A belief-rule-based inference method for carrier battle group recognition, in: Z. Sun, Z. Deng (Eds.), Lecture Notes in Electrical Engineering, vol. 254, Springer, Berlin, Heidelberg, 2013, pp. 261–271.
[33] Z. Liu, Q. Pan, J. Dezert, A new belief-based k-nearest neighbor classification method, Pattern Recognit. 46 (2013) 834–844.
[34] Z. Liu, Q. Pan, J. Dezert, Classification of uncertain and imprecise data based on evidence theory, Neurocomputing 133 (2014) 459–470.
[35] C.J. Merz, P.M. Murphy, 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.
[36] S. Raudys, A. Jain, Small sample effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 252–264.
[37] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (1998) 1895–1923.
[38] D. Michie, D.J. Spiegelhalter, C.C. Taylor, Machine Learning, Neural and Statistical Classification, Ellis Horwood, New York, 1994.



Lianmeng Jiao received the B.E. and M.E. degrees from Northwestern Polytechnical University, Xi'an, China, in 2009 and 2011, respectively. Currently, he is a co-supervised Ph.D. candidate at the School of Automation, Northwestern Polytechnical University, Xi'an, China and the UMR CNRS 7253, Heudiasyc, Université de Technologie de Compiègne, Compiègne, France. His research work focuses on belief functions theory and its application in machine learning and decision making.

Quan Pan received the B.E. degree from Huazhong Institute of Technology, Wuhan, Chinain 1991 and the M.E. and Ph.D. degrees from Northwestern Polytechnical University (NPU), Xi'an, China, in 1991 and 1997, respectively. Since 1998, he has been a professor at the School of Automation, NPU, and he has been the chair of School of Automation, NPU, since 2010. He is a member of IEEE, a member of the International Society of Information Fusion (ISIF), and a board member of the Chinese Association of Automation. His research interests are decision making, information fusion, hybrid system estimation theory, belief function theory, and image processing.



Xiaoxue Feng received the B.E. and M.E. degrees from Northwestern Polytechnical University, Xi'an, China, in 2010 and 2012, respectively. She started the Ph.D. course at the School of Automation, Northwestern Polytechnical University, Xi'an, China, since September 2012. Currently, she is a joint Ph.D. student at the UMR CNRS 6279, ICD-LM2S, Université de Technologie de Troyes, Troyes, France. Her research interests include bio-inspired optimization and information fusion.