HAL
open science

Hybrid Classification System for Uncertain Data

Zhun-Ga Liu, Quan Pan, Jean Dezert, Grégoire Mercier

To cite this version:

Zhun-Ga Liu, Quan Pan, Jean Dezert, Grégoire Mercier. Hybrid Classification System for Uncertain Data. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016, 47 (10), pp.2783-2790. 10.1109/TSMC.2016.2622247 . hal-01848997

HAL Id: hal-01848997

https://hal.science/hal-01848997v1

Submitted on 17 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybrid Classification System for Uncertain Data

Zhun-ga Liu[1], Quan Pan[1], Jean Dezert[2], Gregoire Mercier[3]
1. School of Automation, Northwestern Polytechnical University, Xi'an, China.
Email: liuzhunga@gmail.com
2. ONERA - The French Aerospace Lab, F-91761 Palaiseau, France.
Email: jean.dezert@onera.fr
3. Telecom Bretagne, CNRS UMR 6285 Lab-STICC/CID, Brest, France.
Email: Gregoire.Mercier@telecom-bretagne.eu

***Abstract*—In classification problem, several different classes may be partially overlapped in their borders. The objects in the border are usually quite difficult to classify. A Hybrid Classification System (HCS) is proposed to adaptively utilize the proper classification method for each object according to the K nearest neighbors, which are found in the weighting vector space obtained by Self-Organizing Map (SOM) in each class. If the K close weighting vectors (nodes) are all from the same class, it indicates that this object can be correctly classified with high confidence, and the simple hard classification will be adopted to directly classify this object into the corresponding class. If the object likely lies in the border of classes, it implies that this object could be difficult to classify, and the credal classification working with belief functions is recommended. The credal classification allows the object to belong to both singleton classes and sets of classes (meta-class) with different masses of belief, and it is able to well capture the potential imprecision of classification thanks to the meta-class and also reduce the errors. Fuzzy classification is selected for the object close to the border and hard to clearly classify, and it associates the object with different classes by different membership (probability) values. HCS generally takes full advantage of the three classification ways and produces good performance. Moreover, it requires quite low computational burden compared with other K-NN based methods due to the use of SOM. The effectiveness of HCS is demonstrated by several experiments with synthetic and real data sets.**

Keywords: belief function, evidence theory, pattern classification, uncertain data, Dempster-Shafer theory.

I. Introduction

The K-nearest neighbors (K-NN) classifier [1] remains an important nonparametric method for data classification. The original voting K-NN classifier [1] consists to directly classify the object (test sample) into the majority class of its KNNs in the training data space. This method is called a hard classification technique because it answers to the classification question by only *yes* or *no* (i.e. the object belongs, or not to the class under concern). In the voting K-NN, the K selected neighbors are considered with equal weight. In order to improve the performance of voting K-NN, the fuzzy version of K-NN (denoted by FK-NN) [2], [3] was developed taking into account the distance between the object and its neighbors[1], and the neighbor with the smaller distance to the object has bigger weight (contribution) in the classification. In this way, the object is allowed to belong to different classes with different membership (probability) values, and such kind of classification is referred as fuzzy classification in the literature. There exist various K-NN based classifiers working with probability framework, and they attempt to acquire the better classification performance using different ways. For example, some methods e.g. [4] focus on the selection of distance measure, and the proper metric is obtained for weighing each feature in the calculation of distance to the neighborhoods. The kernels are also introduced in some other methods e.g. [5] to estimate the curved local neighborhood. In [6], a nearest neighbor algorithm is presented to reduce the negative influence of the bad contributing samples, and the object is classified according to the distance between the object and the local categorical probability centers, and the posterior probability of the object. These methods can provide good performances in some particular cases.

The classification methods working with belief functions framework [7]–[9] as an extension of probability framework have been also developed to deal with the uncertainty of classification. In the classification of uncertain data, the different classes can partially overlap in their borders, and the object in the border[2] becomes hard to correctly classify according to the used attributes, because these classes appear undistinguishable for the object. Belief function theory [7]–[9] also called Dempster-Shafer Theory (DST) can well model such uncertainty and imprecision [10], and it has been applied in many fields, such as classification [11], [12], [14]–[21], clustering [22]–[24], decision making support [25]–[30], and so on. Particularly, the Evidential K-NN (EK-NN) [11] and Evidential Neural Network (ENN) classifiers [12] have been introduced by Denœux based on DST, and they generally produce good performance in data classification thanks to the use of belief functions for modeling the ignorant information. A recent evidential classifier

[1]K-NN denotes the classifier, whereas KNNs is the acronym of K-nearest neighbors.

[2]If the KNNs of one object belong to several different classes, this object will be considered quite likely lying in the border of these several classes.

[15] consisting of a feature selection procedure and a two-step classification strategy has been further developed by Lian and Ruan to improve the classification accuracy. We have also proposed several credal classification methods [16], [19] inspired by the previous evidential classifiers [11], [12] to deal with the partly overlapping classes in different cases, and the object can be committed not only to singleton classes but also to sets of any classes (i.e. meta-class) with different masses of beliefs. The credal classification is able to well characterize the imprecision of classification due to the meta-class, and can also reduce the misclassification errors. The recent Belief K-NN (BK-NN) classifier [16] works with credal classification, but it requires a quite high computational burden because of the complicate construction and fusion of the Basic Belief Assignments (BBA's).

In summary, the K-NN methods can be broadly grouped into three families according to the framework working with: 1) the hard classification methods [1], 2) the fuzzy classification methods [2] and 3) the credal classification methods [16]. The hard classification approach is simple but it generates a high risk of error for the uncertain data (like the KNNs from different classes). The fuzzy classification captures the uncertainty of classification using fuzzy membership, whereas the meta-class (i.e. disjunction of several classes), which can well represent the partial ignorant knowledge of classification especially for the data lying in the overlapping border of several classes, is not taken into account in the Frame of Discernment (FoD). The credal classification approach is able to deal with the uncertain and ignorant information using the belief functions defined over the power-set of FoD, but it usually requires high computational complexity, which is "price" one has to pay for attaining the power of expressiveness of the power-set.

In fact, the simple hard classification is suitable for dealing with the object far from the border of class and easy to clearly classify with quite low complexity. Credal classification should be employed when the object is hard to classify and becomes undistinguishable with respect to several different classes. Other objects close to the border could be classified using the fuzzy classification way to properly characterize the uncertainty of classification with the fuzzy membership. So a Hybrid (i.e. Hard-Fuzzy-Credal) Classification System (HCS) is proposed in this work which allows to automatically select the proper classification method according to the KNNs of the object to classify.

The main drawback of K-NN based method is the high computational burden because of the calculation of the distances of each object to all the training samples especially when the number of the training data is big. Various methods have been developed to circumvent this drawback based on data reduction technique [31]. Self-Organizing Map (SOM) technique [32] has been successfully applied to reduce the computational burden of K-NN. In [34], a method referred by SOM-KNN was presented, and SOM is employed for the pre-processing of data set. One can define the potential similar patterns region according to the distance between the object and the weighting vectors of SOM, and then the neighborhoods of the object can be found in the given region. By doing this, one reduces substantially the execution time for seeking the neighborhoods. Another K-NN classification method consisting of the feature reduction based on SOM and the prototype selection using the neighborhood chains has also been developed in [33] to reduce the complexity when working with high dimensional data sets. SOM coupled with K-NN technique seems also very helpful here for automatic selection of a suitable (hard, fuzzy or credal) classification rule in different contexts. So HCS classification is presented based on SOM and K-NN in order to make the problems of classification of uncertain data tractable with admissible computational burden. SOM can produce a small number of weighing vectors that are able to preserve the topological properties of the input data. Therefore, SOM will be applied in each training class, and the optimized weighting vectors are used to represent the corresponding class. By doing this, the object can be classified according to the K_o nearest weighting vectors, and we just need to calculate the distances to the small amount of weighting vectors rather than the original training data, which can greatly improve the computation efficiency.

In HCS, the optimized weighting vectors corresponding to the nodes in SOM are found at first to characterize each training class. The nodes whose K nearest neighbors are from different classes quite probably lie in the overlapping zone of different classes, and they will be labeled as border nodes. The border nodes are considered providing different support degrees to the associated close classes. Then the K_o nearest nodes of the object are found. If the K_o nodes all come from the same class say w_i, the hard classification is adopted, and the object will be directly classified into the class w_i. If the K_o nodes are from different classes, but the nearest one is not in the border, which indicates that this object is not very likely in the border either, then the fuzzy classification is adopted. One can construct K_o BBA's based on the distances to these K_o nodes and the support degree functions of these nodes associated with different classes, and the combination of the K_o BBA's by Dempster's rule[3] of combination [7] will be used for the fuzzy classification. If the K_o nodes originate from different classes and the nearest node lies in the border of several classes, the credal classification is employed and the proper meta-class is used to capture the imprecision of classification (if necessary). In the credal classification of the object, the K_{y_1}[4] nearest original training data of the object, which can provide more refined neighborhood information than the nodes, will be found in close neighborhood area of the nearest node. Then the K_{y_1} BBA's obtained from the K_{y_1} nearest training data will be fused by the proposed rule in which the conflicting beliefs are conditionally transferred to the meta-class, and the credal classification of the object will

[3]At this step, other combination rules could be used as proposed in [8] but these rules requires higher complexity in general that is why we choose Dempster's rule in this work.

[4]Usually, one takes $K_{y_1} = K_o$.

be done based on the global fusion result.

This paper is organized as follows. After a brief introduction of the background knowledge including the basics of SOM and belief function theory in section II, the proposed Hybrid Classification System (HCS) is presented in section III. This new method is then tested and evaluated in section IV with respect to several other methods. Section V concludes and gives some perspectives for related works.

II. Background knowledge of SOM and belief functions

In order to reduce the computational burden of K-NN method, SOM technique is adopted to find the optimized weighting vectors, which can represent the corresponding class. Belief Function Theory (BFT) is also introduced here to deal with the uncertain and imprecise information in the classification. So the basic knowledge on SOM and BFT is briefly recalled.

A. Overview of Self-Organizing Map

Self-Organizing Map (SOM) also called Kohonen map [32], [33], [35] is a type of artificial neural network, and it is trained by unsupervised learning method. It defines a mapping from the input space to a low-dimensional (typically two-dimensional) grid of $M \times N$ nodes. So it can approximate the feature space dimension (e.g. a real input vector $\mathbf{x} \in \mathbb{R}^p$) into a projected 2D space. SOM allows to preserve the topological properties of the input space using a neighborhood function. The node at position $(i, j), i = 1, \ldots M, j = 1, \ldots, N$ corresponds to a weighting vector denoted by $\mathbf{o}(i, j) \in \mathbb{R}^p$. An input vector $\mathbf{x} \in \mathbb{R}^p$ is to be compared to each $\mathbf{o}(i, j)$, and the neuron whose weighting vector is the most close (similar) to $\mathbf{x}$ according to a given metric (e.g. Euclidean distance) is called the Best Matching Unit (BMU), which is defined as the output of SOM with respect to $\mathbf{x}$. The competitive learning is adopted in SOM, and the training algorithm is iterative. After the optimization procedure, the close input patterns will be mapped to the close nodes with the weighting vectors.

In this work, SOM is applied with each training class, and the normal Euclidean distance measure is used here. The optimized weighting vectors corresponding to the nodes can well represent this training class. The number of the weighting vectors (nodes) usually is much smaller than the original training samples. So computational burden of seeking the K nearest weighting vectors of the object could be quite low.

B. Basis of belief function theory

The Belief Function Theory (BFT) is also known as Dempster-Shafer Theory (DST) [7]–[9]. Let us consider a frame of discernment consisting of c exclusive and exhaustive hypotheses (classes) denoted by $\Omega = \{\omega_i, i = 1, 2, \ldots, c\}$. The power-set of Ω denoted 2^Ω is the set of all the subsets of Ω, empty set included. In BFT, one object can belong to different singleton elements and any sets of elements in the power-set with a Basic Belief Assignment (BBA). BBA is a function $m(.)$ from 2^Ω to $[0, 1]$ satisfying $m(\emptyset) = 0$ and the normalization condition $\sum_{A \in 2^\Omega} m(A) = 1$. The subsets A of Ω such that $m(A) > 0$ are called the focal elements of the belief mass $m(.)$. The credal classification (or partitioning) [22] is defined as n-tuple $M = (\mathbf{m_1}, \cdots, \mathbf{m_n})$ of BBA's, where $\mathbf{m}_i$ is the basic belief assignment of the object $\mathbf{x}_i \in X$, $i = 1, \ldots, n$ associated with the different elements in the power-set 2^Ω.

In BFT, the combination of multiple sources of evidence represented by a set of BBA's, is done with Dempster's rule (called DS rule for short) [7]. The combination of two BBA's $m_1(.)$ and $m_2(.)$ over 2^Ω using DS rule is defined by $m_{DS}(\emptyset) = 0$ and for $A \neq \emptyset, B, C \in 2^\Omega$ by

$$m_{DS}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} \tag{1}$$

DS rule is commutative and associative. The total conflicting mass $\sum_{B \cap C = \emptyset} m_1(B) m_2(C)$ is redistributed back to the focal elements through a classical normalization step in (1). However, this normalization can unfortunately yield unreasonable results in the high conflicting cases, as well as in some special low conflicting cases as well [36]. That is why different rules of combination have emerged to overcome its limitations [8].

III. Hybrid Classification System

Hybrid Classification System (HCS) includes three families of classification methods (hard classification, fuzzy classification and credal classification). One proper classification method will be automatically selected according to the context. If one object can be clearly classified with high confidence, the hard classification will be adopted. If one object lying in the border of different classes is difficult to correctly classify, one uses the credal classification to well capture the imprecision of classification and to reduce the risk of error. The border zone is determined according to the neighborhood information. The fuzzy (probabilistic) classification lies between the credal classification and hard classification, and it is applied for the object close to the border and hard to clearly classify.

A. Transformation of training data using SOM

Let us consider a set of test data (samples) $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ to be classified using a set of labeled training data $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_s\}$ over the frame of discernment $\Omega = \{\omega_1, \ldots, \omega_c\}$ of classes.

In the classification problem, the big computational burden is the main drawback of the K-NN classifier especially for the big training data set, since the distances between the object and all the training data have to be calculated. In order to solve this problem, we attempt to seek a small number of data points to well represent the training data set. SOM technique [32] will be applied in each training class, and the optimized weighing

vectors corresponding to nodes in SOM are able to preserve the topological properties of the input data. Each training class (i.e. $w_g, g = 1, \ldots, c$) is characterized by $z = M \times N$ weighting vectors (nodes) $\mathbf{O}^{w_g} = \{\mathbf{o}_1^{w_g}, \ldots, \mathbf{o}_z^{w_g}\}$ obtained by SOM, and the weighting vector space is denoted by $\mathbf{O} = \{\mathbf{O}^{w_1}, \ldots, \mathbf{O}^{w_c}\}$.

If one node lies in the border (overlapped zone) of several classes, the class of object in the close neighborhood zone of this node will be considered uncertain because these different classes appear undistinguishable for the object. In other words, this node provides uncertain support degree for these different classes close together. In this work, the K-NN method is used to detect the border node. At first, K_o nearest neighbors (in the weighting vector space) of each weighting vector (e.g. $\mathbf{o}_i^{w_g}$) are found[5]. If the K_o neighbors coincide all with the same class (e.g. w_g) for this node, then this node defined by non-border node is considered totally reliable to support this class, and its support degree is given by $\tilde{m}_i^o(w_g) = 1$. If the K_o neighbors belong to distinct classes, this node will be labeled by *border node* because it seems very likely in the border zone of these different classes.

Let us consider a border node $\mathbf{o}_i^{w_g}$ obtained from training class w_g, and the K_o nearest neighbors of $\mathbf{o}_i^{w_g}$ are respectively from $w_l, \ldots, w_g$. The support degree of this border node for its close classes can be calculated by a normal classifier. The Evidential K-NN (EK-NN) method, which is an evidential extension of K-NN proposed in [11], is adopted here to classify the border node using the original training data in these close classes, since this K-NN based method generally produces good performance [11]. The output of EK-NN is denoted by $\tilde{m}_i^o(.)$, and it includes the singleton elements corresponding to these involved classes and the total ignorant class Ω. If the class mainly supported by $\tilde{m}_i^o(.)$ is w_g, i.e. $\max\{\tilde{m}_i^o(.)\} = \tilde{m}_i^o(w_g)$, it implies that $\tilde{m}_i^o(.)$ is consistent with the fact that the border node $\mathbf{o}_i^{w_g}$ originates from w_g. Consequently, $\tilde{m}_i^o(.)$ can be directly used to reflect the support degree of the node $\mathbf{o}_i^{w_g}$ for these associated close classes.

If the mainly supported class by $\tilde{m}_i^o(.)$ is not w_g, i.e. $\max\{\tilde{m}_i^o(.)\} \neq \tilde{m}_i^o(w_g)$, it indicates that $\tilde{m}_i^o(.)$ highly conflicts with the truth that the border node $\mathbf{o}_i^{w_g}$ is from w_g. In such case, this result $\tilde{m}_i^o(.)$ doesn't bring information for supporting the object classification in the specific class w_g, and one can just obtain the partial ignorance $m_i^o(w_l \cup \ldots \cup w_g) = 1$ from $\mathbf{o}_i^{w_g}$ in the classification. The support degree characterized by a BBA $m_i^o(.)$ of the node $\mathbf{o}_i^{w_g}$ for its close classes is given by

$$\begin{cases} m_i^o(A) = \tilde{m}_i^o(A), A \subseteq \Omega, & \text{if } \max\{\tilde{m}_i^o(.)\} = \tilde{m}_i^o(w_g) \\ m_i^o(w_l \cup \ldots \cup w_g) = 1, & \text{if } \max\{\tilde{m}_i^o(.)\} \neq \tilde{m}_i^o(w_g) \end{cases} \tag{2}$$

B. Data classification using K-nearest neighbors

The $K_\mathbf{o}$ nearest weighting vectors (also called nodes) in SOM of each object (say $\mathbf{x}_i, i = 1, \ldots, n$) are sought at first. In this step, one just needs to calculate the distances of the object to the $c \times z$ (c being the number of classes, and z being the number of nodes in each class) nodes rather than all the training data, which maintains the computational burden relatively low.

1) Hard classification: If the K_o selected nodes are all from the same class (say w_g), it indicates that this object is easy to clearly classify, and one can be very confident in committing directly this object to this class w_g in hard classification manner. This is a very simple way with quite low computational complexity.

2) Fuzzy classification: If the K_o nodes are from distinct classes but the closest one is not labeled as border node, it indicates that the object is likely close to these different classes. The hard classification of this object can cause high risk of error, and the fuzzy classification method is adopted.

The K_o pieces of support degree functions $m_{i,k}^o(.)(k = 1, \ldots, K_o)$ corresponding to the K_o nodes will be weighted combined for the classification of this object. The weighting factor of each support degree function $m_{i,k}^o(.)$ is determined by the distance of the object to the corresponding node $\mathbf{o}_{i,k}$. The weighting factor $\alpha_{i,k}$ is defined by

$$\alpha_{i,k} = e^{-d_{ik}\gamma_o} \tag{3}$$

with

$$\gamma_o = \frac{2 \sum_{\substack{i=1 \\ i \neq j}}^{cz} \sum_{j=1}^{cz} d(\mathbf{o}_i, \mathbf{o}_j)}{cz(cz-1)} \tag{4}$$

where d_{ik} is distance between object $\mathbf{x}_i$ and the node $o_{i,k}$, and γ_o is the average distance between each pair of weighting vectors. The parameter c is the number of classes, and z is the number of nodes in each class.

The classical Shafer's discounting method [7] is employed here to discount the support degree function $m_{i,k}^o(.), k = 1, \ldots, K_o$ obtained from $\mathbf{o}_{i,k}$ using the weighting (discounting) factors $\alpha_{i,k}$, and the discounted BBA is given by

$$\begin{cases} m'_{i,k}(w_g) = \alpha_{i,k} m_{i,k}^o(w_g), & \text{for } w_g \in \Omega \\ m'_{i,k}(\Omega) = 1 - \alpha_{i,k} + \alpha_{i,k} m_{i,k}^o(\Omega) \end{cases} \tag{5}$$

These discounted BBA's $m'_{i,k}(.)$ will be combined using DS rule by eq.(1) for the fuzzy classification of the object, and it is given by $A, B_k \in 2^\Omega$:

$$m_i(A) = \frac{\sum_{\bigcap_{k=1}^{K_o} B_k = A} \prod_{k=1}^{K_o} m'_{i,k}(B_k)}{1 - \sum_{\bigcap_{k=1}^{K_o} B_k = \emptyset} \prod_{k=1}^{K_o} m'_{i,k}(B_k)} \tag{6}$$

Due to the particular structure of BBA's to combine with DS rule, the combination result here contains only the singleton elements (classes) and the total ignorant class denoted by Ω which usually represents the ignorance (i.e. outliers). The mass

[5] K_o being a small number, e.g. two or three.

of belief on Ω can be redistributed to other focal elements if the outliers (noisy data) is not applicable. To get a Bayesian BBA (homogeneous to a probability measure) denoted $p_i(.)$ to make the fuzzy classification, one just needs to normalize the masses of singleton classes by $1 - m_i(\Omega)$. More precisely, the output of the fuzzy classification say $p_i(.)$ is a probabilistic measure given by

$$p_i(w_g) = \frac{m_i(w_g)}{1 - m_i(\Omega)} \tag{7}$$

3) Credal classification: If the closest node (e.g. $\mathbf{o}_{i,l}$) of one object is in the border of different classes and the K_o selected nodes of this object belong to different classes, it indicates that the object probably lies in the overlapped zone of these classes, and it becomes very difficult to correctly classify this object. In such case, the credal classification must be employed to capture the imprecision of classification and to reduce the misclassification errors.

The object in the border of classes must be treated more cautiously, and a more refined neighborhood information is necessary for making the classification. So, it could be better to seek the K_{y_1} (K_{y_1} usually is equal to K_o) nearest neighbors of this object in the original training data space Y than in the node (weighting vector) space $\mathbf{O}$, because the original training data space can provide more specific information in the local area than the nodes of SOM.

However, finding the K_{y_1} nearest neighbors of $\mathbf{x}_i$ directly in the whole training data set can be a very time consuming task which is not satisfactory for real applications. In fact, the K_{y_1} nearest neighbors to find generally lie in the close neighborhood of the nearest node $\mathbf{o}_{i,l}$. In this work, the $K_{y_2} = \rho K_{y_1}$ nearest neighbors of $\mathbf{o}_{i,l}$ in the training data space are used to characterize the close neighborhood (local area) information around $\mathbf{o}_{i,l}$, and fortunately this step can be done off-line during the transformation of training data for the SOM construction. The tuning parameter $\rho > 1$ is a positive number with a recommended default value of $\rho = 3$. So we can choose the K_{y_1} nearest neighbors of $\mathbf{x}_i$ from the K_{y_2} selected training data around $\mathbf{o}_{i,l}$, and we just need to calculate the distances between the object and the K_{y_2} selected training data rather than the whole training data set, which reduces drastically the computational complexity of the search.

Each selected training sample can provide a BBA's of the object $\mathbf{x}_i$ associated with the class this training sample belongs to. The BBA's of the object $\mathbf{x}_i$ associated with one selected neighbor $\mathbf{y}_k, k = 1, \ldots, K_{y_1}$ labeled by the class w_g can be defined by.

$$\begin{cases} \hat{m}_{i,k}(w_g) = e^{-d_{ik}\gamma_y} \\ \hat{m}_{i,k}(\Omega) = 1 - e^{-d_{ik}\gamma_y} \end{cases} \tag{8}$$

with

$$\gamma_y = \frac{2 \sum\limits_{\substack{i=1 \\ i \neq j}}^{s} \sum\limits_{j=1}^{s} d(\mathbf{y}_i, \mathbf{y}_j)}{s(s-1)} \tag{9}$$

where d_{ik} is the distance between the object $\mathbf{x}_i$ and $\mathbf{y}_k$. One can see that the smaller distance d_{ik} leads to the more mass of belief on the class w_g and less belief on the ignorance. However, if the neighbor is far from the object $\mathbf{x}_i$, it can provide low support degree for $\mathbf{x}_i$ belonging to the corresponding class w_g, and the ignorance degree will become high. $\gamma_{\mathbf{y}}$ is the average distance between each pair of training samples, and s is the number of training samples. The K_{y_1} BBA's can be obtained based on the distance of the object to each neighbor in the similar way. The K_{y_1} BBA's will be globally fused for the credal classification of this object.

In the fusion process, the BBA's obtained from the the same class are grouped into one cluster at first, and we can get several distinct groups $G_1, G_2, \ldots, G_r$ corresponding to class $w_1, w_2, \ldots, w_r$. The BBA's in the same cluster (e.g. G_g) are generally consistent, since they all mainly support the same class (e.g. w_g). We propose to combine these BBA's with DS rule, denoted symbolically by $\oplus$, according to Eq. (1) because the conflict between them should be relatively low, more precisely one will get for $A \subseteq \Omega$, and with $\hat{\mathbf{m}}_{i,h} \in G_g, h = j, \ldots, k$

$$\hat{\mathbf{m}}_i^{w_g}(A) = [\hat{\mathbf{m}}_{i,j} \oplus \ldots \oplus \hat{\mathbf{m}}_{i,k}](A) \tag{10}$$

The sub-combination results of BBA's in the different clusters (i.e. $\hat{\mathbf{m}}_i^{\omega_g}, g = 1, \ldots, r$) will be globally fused for the final classification. These sub-combination results $\hat{\mathbf{m}}_i^{w_g}, g = 1, \ldots, r$ are usually in high conflict because they strongly support distinct classes. If DS rule is still directly applied in this global fusion, the conflicting beliefs will be totally distributed to other focal elements. Nevertheless, the conflicting beliefs are very important to reveal the imprecision of classification for the object in the border of classes. We argue that the conflicting beliefs should be conditionally preserved according to the distinguishability degree of the object with respect to the different classes.

The class having the biggest mass of beliefs among these pieces of sub-combination results will be considered as the most expected one in classification of the object, and it will obtained by:

$$\hat{m}_i^{w_t}(w_t) \triangleq \max\{\hat{m}_i^{w_1}(w_1), \ldots, \hat{m}_i^{w_r}(w_r)\} \tag{11}$$

where w_t is the most expected class.

The distinguishability degree of one object between the class w_t and an other class is defined by:

$$\kappa_{t,g} \triangleq \hat{m}_i^{w_t}(w_t) - \hat{m}_i^{w_g}(w_g) \tag{12}$$

If the condition $\kappa_{t,g} \leq \epsilon$ (ϵ being a small threshold number) is satisfied, it indicates the classes w_g and w_t cannot be clearly distinguished for the object with respect to the threshold ϵ. The undistinguishable class set is defined by $\Psi \triangleq \{w_g | \kappa_{t,g} \leq \epsilon\}$. The subsets of Ψ called meta-classes should be kept to reflect the imprecision of classification, and the associated conflicting beliefs will be transferred to the proper meta-class to reflect the imprecision degree.

Therefore, the global fusion rule for these sub-combination

results is defined by:

$$\widetilde{m}_i(A) = \begin{cases} \sum\limits_{\bigcap_{j=1}^{r} B_j = A} \prod\limits_{j=1}^{|A|} \hat{m}_i^{w_j}(B_j), A \in \Omega, \text{or } A = \Omega \\ \sum\limits_{\substack{\bigcap_{j=1}^{|A|} w_j = \emptyset \\ \bigcup_{j=1}^{|A|} w_j = A}} [\prod\limits_{j=1}^{|A|} \hat{m}_i^{w_j}(w_j) \prod\limits_{g=|A|+1}^{r} \hat{m}_i^{w_g}(\Omega)], A \subseteq \Psi_i \end{cases} \tag{13}$$

Because not all partial conflicting masses of belief are transferred into the meta-classes through the global fusion formula (13), the combined BBA is normalized by the classical way as follows before making a decision:

$$m_i(A) = \frac{\widetilde{m}_i(A)}{\sum\limits_{B_i \subseteq \Omega} \widetilde{m}_i(B_j)} \tag{14}$$

The object is then classified into the class A (A being a singleton class or meta-class) according to the maximum mass of belief, that is if $m_i(A) = \max\{m_i(.)\}$. This procedure is called hard credal classification.

Guideline for tuning the parameters: In SOM training procedure, the number of the nodes (weighting vectors) $z = M \times N$ is determined according to the number of the training data points, and the default value 5×6 is generally recommended. The distinguishability degree threshold $\epsilon \in (0, 1)$ is usually a very small positive value, and the smaller value will lead to fewer number of objects in the meta-class. So ϵ can be tuned according to the imprecision degree one can accept in the classification.

IV. Experiments

Three experiments are presented to evaluate the performance of HCS with respect to five other K-NN based methods (i.e. K-NN [1], FK-NN [2], EK-NN [11], SOM-KNN [34] and BK-NN [16]), and ENN classifier [12]. The different methods have been programmed and tested with Matlab™ software. The parameters of EK-NN were automatically optimized using the method introduced in [18]. The tuning threshold ϵ in HCS is optimized using the training data. The optimized value corresponds to a suitable compromise between error rate and imprecision rate. The training data preparation with the parameter optimization can be done off-line.

In HCS, SOM will be trained in each class. For the class $\omega_i, i = 1, \ldots, c$, the corresponding SOM consists of $M \times N$ grid notes associated with $M \times N$ weighting vectors, which are obtained by the competitive learning (i.e. iterative training algorithm) using the training patterns in ω_i. Each training pattern is mapped by SOM to one node with the closest weighting vector to this pattern. So the close patterns will be mapped to the close nodes in SOM. Thus, SOM preserves the topological properties of the training patterns, and the nodes can well characterize the distribution of the corresponding class. Euclidean distance is adopted here. The choice of number of the nodes, i.e. $M \times N$, remains an open problem, and it mainly depends on the actual applications. If the number of nodes as $M \times N$ is too big, it will lead to high computation complexity here. Whereas, the too few nodes can not well characterize the whole training data space. In the experiments, the value of $M \times N$ is determined based on the cross validation (leave-one-out) in training data space, and the selected value can produce the good classification performance. For example, $M \times N = 5 \times 6$ generates good performance according to the cross validation using training data, and it is employed in the following experiments. The number of epochs can be set from 200 to 1000 according to the applications, but the big number of epochs can cause big computational burden. In this work, we take number of epochs as 250, which can yield good results with admissible complexity. In the class decision making step, the object with the hard classification is directly classified into a particular class, and the object with fuzzy classification is considered belonging to the class with maximum probability, and the object with credal classification is committed to the class (set) having the maximum mass of belief.

In the performance evaluation, the error rate, imprecision rate (related with the meta-classes), and the utility value [37] are used for the comparison between HCS and several other methods. For one object originated from w_i, if it is classified into A with $w_i \cap A = \emptyset$, it will be considered as an error. If $w_i \cap A \neq \emptyset$ and $A \neq w_i$, it will be considered with imprecise classification. The error rate denoted by R_e is calculated by $R_e = N_e/T$, where N_e is number of objects wrongly classified, and T is the total number of the objects tested. The imprecision rate denoted by R_{i_j} is calculated by $R_{i_j} = N_{i_j}/T$, where N_{i_j} is number of objects committed to the meta-classes with the cardinality value j. We take $j = 2$ in the following experiments, since there is no object committed to the meta-class with the cardinality value bigger than two. So it holds that the sum of the accuracy rate denoted by R_a[6] and error rate R_e and the imprecision rate R_i must be one as $R_a + R_e + R_i = 1$. The small error rate may cause high imprecision rate, since the object hard to classify may be committed to the meta-class (disjunction of several classes) by credal classification method.

In [37], the utility-discounted predictive accuracy was introduced to evaluate the credal classifier based on imprecise-probability models, and the measure of utility value is also adopted here. The utility value for the classification of one object can be given by

$$U_{65}(\mathbf{x}_i) = \begin{cases} 1, & \text{if } A = L(i) \\ 0, & \text{if } A \cap L(i) = \emptyset \\ 0.65, & \text{if } L(i) \in A, \text{and } |A| = 2 \end{cases} \tag{15}$$

where A is the classification result of the object $\mathbf{x}_i$ in HCS, and $L(i)$ is the real class that $\mathbf{x}_i$ belongs to. It indicates that the object correctly classified contributes the utility value 1, and the

[6] $R_a = \frac{N_a}{T}$, where N_a is the number of objects correctly classified.

object committed to the the proper meta-class with cardinality value 2 produces the utility value 0.65. The utility value for the misclassified object is zero. The utility value is denoted by U for short in the sequel.

A. Experiment 1

This experiment is used to clearly illustrate the use of hybrid classification with respect to the EK-NN and K-NN methods. A particular 3-class 2-D data set composed by three rings corresponding to three classes w_1, w_2 and w_3 are shown by Fig. 1-(a). Each class contains 303 training samples and 303 test samples. The centers and radiuses interval of the three rings are given by Table I below.

Table I
PARAMETERS OF SIMULATED ARTIFICIAL DATA SETS.

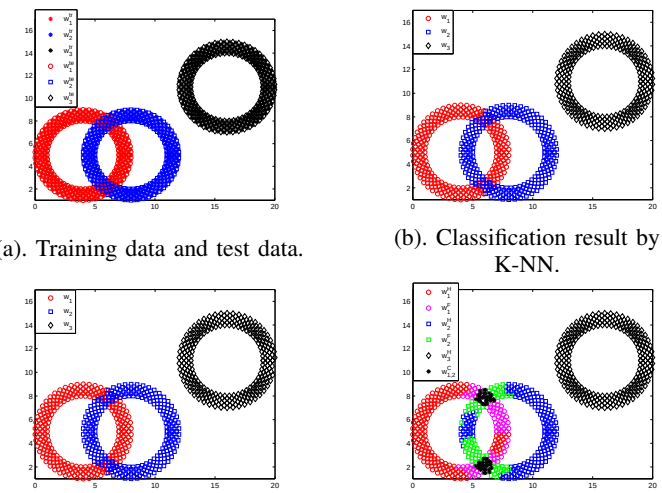
	Center	Radius interval
w_1	(4, 5)	[3, 4]
w_2	(8, 5)	[3, 4]
w_3	(16, 11)	[3, 4]

The particular values of $K_o = K_{y_1} = 4, \rho = 3$ in HCS and the default value $M \times N = 5 \times 6$ in SOM are used here because they provide good performances. The classification results of test data by K-NN, EK-NN and HCS are respectively shown in Fig. 1-(b)-(d). For notation conciseness, we have denoted $w^{te} \triangleq w^{test}$, $w^{tr} \triangleq w^{training}, w_{i,\ldots,k} \triangleq w_i \cup \ldots \cup w_k$ and $w_i^H, i = 1, 2, 3$ represents the class w_i obtained by the hard classification, $w_i^F, i = 1, 2, 3$ represents the class w_i obtained by the fuzzy classification, and $w_i^C, i = \{1, 2\}$ represents the class w_i obtained by the credal classification. The classification performance including the error rate R_e, imprecision rate R_{i_2}, the execution time T in millisecond (ms) and the utility value U as expressed in eq. (15) is given in table II with respect to different methods. "NA" means "No Applicable".

Table II
CLASSIFICATION RESULTS OF DIFFERENT METHODS IN EXP. 1 (IN%).

	K-NN [1]	EK-NN [11]	HCS
R_e	4.84	4.85	0.77
R_{i_2}	NA	NA	6.05
U	95.16	95.15	97.11
T	10.94	9.38	6.18

In Fig. 1-(a), one can see that the the classes w_1 and w_2 are crossed, and it is quite hard to correctly classify the objects in the crossed (overlapped) zones. Each object has been classified into a particular class w_1, w_2 or w_3 by K-NN and EK-NN according to the maximum probability (for K-NN) or belief (for EK-NN) measure, but this causes many misclassification errors for the objects in the overlapped zones. With HCS, the objects that can be easily classified have been directly classified into a specific class by hard classification, and this result is considered very confident for the decision making support. The objects close to the overlapped zone are cautiously classified using fuzzy classification way, and the object is assigned to a class with the maximum probability. For the objects in the overlapped zone, they are really difficult to be correctly classified, and it is more reasonable to commit them into the proper meta-class $w_1 \cup w_2$ thanks to the credal classification. By doing this, one is able to well characterize the imprecision of classification and also to reduce the risk of error. So HCS obtains the biggest unified value shown in table II compared with other methods. The credal classification can warn the user that the used attribute information is not sufficient for making a clear classification of the object, and other information source or technique should be necessary (if available) to get a more precise classification result.



(a). Training data and test data.

(b). Classification result by K-NN.

(c). Classification result by EK-NN.

(d). Classification result by HCS.

Figure 1. Classification results of a 3-class data set by K-NN, EK-NN and HCS .

B. Experiment 2

We evaluate the performance of HCS using a 4D data set which includes 3 classes ω_1, ω_2, and ω_3. The artificial data are generated from three 4D Gaussian distributions characterized by the following means vectors and covariance matrices, where $\mathbf{I}$ denotes the 4×4 identity matrix:

$$\mu_1 = (10, 50, 100, 100)^T, \mathbf{\Sigma}_1 = 10 \cdot \mathbf{I}$$
$$\mu_2 = (30, 30, 80, 90)^T, \mathbf{\Sigma}_2 = 15 \cdot \mathbf{I}$$
$$\mu_3 = (20, 80, 90, 130)^T, \mathbf{\Sigma}_3 = 12 \cdot \mathbf{I}$$

We have used n training samples, and n test samples (for $n = 500, 1000, 2000$) in each class. So there is a total of $N = 3 \times n$ training samples and $N = 3 \times n$ test samples. Three other methods K-NN, FK-NN and EK-NN are also applied here for the performances comparison. The average error rates,

imprecision rates and execution time with standard deviation are reported in Table III according to the 10 trials performed with 10 independent random generation of the data sets, with a number of neighbors K ranging from 5 to 20. In HCS, we still use the values $K_o = K_{y_1} = 4$, $\rho = 3$ with $M \times N = 5 \times 6$ in SOM, and the parameter ϵ has been optimized to obtain an acceptable compromise between error rate and the imprecision degree. The classification performance of the five tested methods (i.e. K-NN, FK-NN, EK-NN, SOM-KNN and HCS) consisting of error rate R_e, imprecision rate R_{i_2}, execution time T in millisecond (ms) and the utility value U are given in Table III.

One can see that the proposed HCS produces a lower error rate with respect to K-NN, FK-NN and EK-NN thanks to introduction of the meta-classes. The objects really hard to be clearly classified are committed to the proper meta-class, which brings the partial imprecision of classification in HCS. Moreover, HCS has a much shorter execution time than K-NN, FK-NN, EK-NN, and even SOM-KNN methods thanks to the use of SOM technique. In HCS, the KNNs of object are found in the weighting vectors space obtained by SOM, and one just needs to calculate the distances of object to the few ($3 \times 30 = 90$) weighting vectors. The hard classification is adopted for many objects that can be easily classified, and there is no need to calculate the distance between the object and the original training samples for hard classification. So the computational burden of HCS is much lower than these other K-NN methods especially when dealing with large training data sets.

C. Experiment 3

We use eight real data sets[7] from UCI [8] to test the performance of HCS with respect to the evidential neural network (ENN) [12] and five other K-NN based methods (i.e. K-NN [1], FK-NN [2], SOM-KNN [34], EK-NN [11] and BK-NN [16]). The basic information about the used data sets are given in Table IV.

Table IV
BASIC INFORMATION OF THE USED REAL DATA SETS.

Name	Classes	Attributes	Instances
Statlog Heart (H)	2	13	270
Breast(B)	2	9	699
Iris (I)	3	4	150
Seeds (S)	3	7	210
Wine (W)	3	13	178
Yeast (Y)	3	8	1055
Vehicle (V)	4	18	946
Knowledge (K)	4	5	403

[7]Three classes (CYT, NUC and $ME3$) are selected in Yeast data set to the evaluate our method, since the objects in these three classes are hard to classify.

[8]The data sets can be freely downloaded in http://archive.ics.uci.edu/ml

The 10-fold cross validation is performed on these data sets by different classification methods. The data set is divided into 10 parts, and each part will be considered as test patterns and all the other parts are used as training patterns in each fold. The normal Euclidean distance is applied here. The tuning parameter ϵ in HCS is optimized using the training samples in each fold, and we have chosen[9] $K_o = K_{y_1} = 4$, $\rho = 3$ with 3×4 nodes in SOM. The classification results by K-NN, EK-NN and BK-NN with different values of K ranging from 5 to 15 are respectively shown on Table V. The average error rate R_e, imprecision rate R_{i_2} (for BK-NN and HCS), the unified value U expressed in eq. (15) and the execution time T in millisecond (ms) with the standard deviation std are given in Table V. Each data set is represented by the first letter of world (e.g. I=Iris) here.

The results of Table V show that HCS can provide lower error rate and higher utility value than EK-NN and K-NN methods thanks to the introduction of meta-class in the credal classification. HCS generally produces performances similar to BK-NN method according to the compromise between the error rate and imprecision rate, but the computational burden of HCS is much lower than BK-NN thanks to the use of hard and fuzzy classification based on SOM technique. Therefore, the execution time of HCS is quite shorter than BK-NN. So HCS produces a good compromise between classification performance and the computational burden. In HCS, when the object is committed to the meta-class, it indicates that this object is really difficult to correctly classify based on the used attributes, and other costly technique or new information source should be employed if one wants to get more precise classification results. So the HCS can provide more useful information in the classification for the user with a tractable computational complexity.

V. CONCLUSION AND PERSPECTIVES

The Hybrid Classification System (HCS) has been proposed in this work, and the proper method (i.e. hard classification, fuzzy classification or credal classification) is adaptively selected for the classification of object according to the K_o nearest weighting vectors (corresponding to the nodes) obtained by SOM technique. The hard classification is adopted when the object can be easily classified, and the hard classification result is usually considered very trustable. The fuzzy classification will be employed if the K selected nodes are from different classes but the nearest one is not in the border of classes, and the object is classified based on the fusion of K_o BBA's associated with the K_o nodes. If the nearest node is in the overlapped border of several classes and the K_o nodes come from various classes, it implies that this object very likely lies also in the border. Such object could be very difficult to classify because it cannot be clearly distinguished with respect to these classes close together.

[9]If the values of K_o and K_{y_1} are too big, it will increase drastically the computation burden. Whereas, the rather small value of K_o and K_{y_1} may cause high error rate. $K_o = K_{y_1} = 4$ is selected for a good compromise between computation burden and misclassification errors according to the cross validation on the training data set.

Then the K nearest training data of the object are found in the close neighborhood area of this nearest node for the credal classification, which can well capture the potential imprecision of classification using the meta-class, and also reduce the errors. The effectiveness of this new HCS has been demonstrated by three experiments using artificial and real data sets. One has shown that HCS can produce lower error rate at the price of partial imprecision with respect to K-NN and EK-NN methods, and the computational burden of HCS is much smaller than the classical K-NN based methods thanks to the use of SOM. As future research works, we will try to find an efficient method to automatically determine the number of nodes in SOM and the K value in K-NN classifier according to the training data set.

Acknowledgement

This work has been partially supported by National Natural Science Foundation of China (Nos.61135001, 61403310, 61672431).

REFERENCES

[1] A.K. Agrawala (Editor), *Machine Recognition of Patterns*, IEEE Press, New York, 1977.

[2] J.M. Keller, M.R. Gray, J.A. Givens, *A fuzzy K-nearest neighbor algorithm*, IEEE Trans. Systems, Man and Cybernetics. Vol.15, No. 4, pp. 580–585, 1985.

[3] J. Derrac, S. Garcia, F. Herrera, *Fuzzy Nearest Neighbor Algorithms: Taxonomy, Experimental analysis and Prospects*, Information Sciences, Vol. 260 98-119, 2014.

[4] C. Domeniconi, J. Peng, D. Gunopulos, *Locally adaptive metric nearest neighbor classification*, IEEE Trans. Pattern Anal. Mach. Intell., Vol.24(9): 1281–1285, 2002.

[5] P. Jing, D.R. Heisterkamp, H.K. Dai, *Adaptive quasiconformal kernel nearest neighbor classification*, IEEE Trans. Pattern Anal. Mach. Intell., Vol.26(5):656–661, 2004.

[6] B. Li, Y.W. Chen, Y.Q. Chen, *The nearest neighbor algorithm of local probability centers*, IEEE Trans. Systems, Man, and Cybernetics B, Vol. 38(1):141–154, 2008.

[7] G. Shafer, *A mathematical theory of evidence*, Princeton Univ. Press, 1976.

[8] F. Smarandache, J. Dezert (Editors), *Advances and applications of DSmT for information fusion*, American Research Press, Rehoboth, Vol. 1-4, 2004-2015. http://fs.gallup.unm.edu/DSmT.htm

[9] P. Smets, *The combination of evidence in the transferable belief model*, IEEE Trans. Pattern Anal. and Mach. Intell., Vol. 12(5):447–458, 1990.

[10] A.L. Jousselme, C. Liu, D. Grenier, E. Bossé, *Measuring ambiguity in the evidence theory*, IEEE Trans. Systems, Man and Cybernetics-Part A: Vol.36(5):890–903, 2006.

[11] T. Denœux, *A k-nearest neighbor classification rule based on Dempster-Shafer Theory*, IEEE Trans. on Systems, Man and Cybernetics, Vol. 25, No. 5, pp. 804–813, 1995.

[12] T. Denœux, *A neural network classifier based on Dempster-Shafer theory*, IEEE Trans. on Systems, Man and Cybernetics A, Vol. 30, No. 2, pp. 131–150, 2000.

[13] Z.g. Liu, Q. Pan , J. Dezert, A. Martin, *Adaptive imputation of missing values for incomplete pattern classification*, Pattern Recognition, Vol.52:85-95, 2016.

[14] S. Trabelsi, Z. Elouedi, P. Lingras, *Classification systems based on rough sets under the belief function framework*, International Journal of Approximate Reasoning, Vol.52(9):1409-1432, 2011.

[15] C. Lian, S. Ruan, T. Denœux, *An evidential classifier based on feature selection and two-step classification strategy*, Pattern Recognition, Vol. 48:2318-2327, 2015.

[16] Z.g. Liu, Q. Pan, J. Dezert, *A new belief-based K-nearest neighbor classification method*, Pattern Recognition, Vol. 46(3):834–844, 2013.

[17] L. Jiao, T. Denœux, Q. Pan, *A hybrid belief rule-based classification system based on uncertain training data and expert knowledge*, IEEE Trans. Systems, Man and Cybernetics: Systems, doi:10.1109/TSMC.2015.2503381, 2016.

[18] L.M. Zouhal, T. Denœux, *An evidence-theoretic k-NN rule with parameter optimization*, IEEE Trans. Systems, Man and Cybernetics - Part C, Vol. 28, No. 2, pp. 263–271, 1998.

[19] Z.g. Liu, Q. Pan, J. Dezert, G. Mercier, *Credal classification rule for uncertain data based on belief functions*, Pattern Recognition, Vol. 47(7): 2532–2541,2014.

[20] Z.g. Liu, Q. Pan, G. Mercier, J. Dezert, *A new incomplete pattern classification method based on evidential reasoning*, IEEE Trans. Cybernetics, Vol.45(4): 635-646, 2015.

[21] T. Denœux, P. Smets, *Classification using belief functions: relationship between case-based and model-based approaches*, IEEE Trans. Systems, Man and Cybernetics, Part B: Vol. 36(6):1395–1406, 2006.

[22] T. Denœux, M.-H. Masson, *EVCLUS: EVidential CLUStering of proximity data*, IEEE Trans. on Systems, Man and Cybernetics Part B, Vol. 34(1):95–109, 2004.

[23] Z.-g. Liu, Q. Pan, J. Dezert, G. Mercier, *Credal c-means clustering method based on belief functions*, Knowledge-based systems, Vol.74:119-132,2015.

[24] T. Denœux, *Maximum likelihood estimation from uncertain data in the belief function framework*, IEEE Transactions on Knowledge and Data Engineering, Vol. 25(1):119–130, 2013.

[25] Z.g. Liu, J. Dezert, Q. Pan, G. Mercier, *Combination of sources of evidence with different discounting factors based on a new dissimilarity measure*, Decision Support Systems, Vol. 52:133–141, 2011.

[26] X. Deng, D. Han, J. Dezert, Y. Deng, *Evidence Combination From an Evolutionary Game Theory Perspective*, IEEE Trans. Cybernetics, Vol.46(9): 2070–2082, 2016.

[27] J.b. Yang, D.l. Xu, *Evidential reasoning rule for evidence combination*, Artif. Intell., Vol. 205, pp.1–29, 2013.

[28] Z.J. Zhou, L.L. Chang, C.H. Hu, X.X. Han, Z.G. Zhou, *A new BRB-ER based model for assessing the life of products using data under various environments* IEEE Trans. Systems, Man and Cybernetics: Systems, Vol.46(11): 1529–1543, 2016.

[29] D.q. Han, J. Dezert, Z.s. Duan, *Evaluation of Probability Transformations of Belief Functions for Decision Making*, IEEE Trans. Systems, Man, and Cybernetics: Systems, Vol.46(1): 93–108, 2016.

[30] S.y. Huang, X.y. Su, Y. Hu, S. Mahadevan, Y. Deng, *A new decision-making method by incomplete preferences based on evidence distance*, Knowledge-Based Systems, Vol.56:264–272, 2014.

[31] S. Garcia, J. Derrac, J.R. Cano, F. Herrera, *Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34(3):417–435,2012.

[32] T. Kohonen, *The Self-Organizing Map. Proceedings of the IEEE*, Vol.78(9):1464–1480, 1990.

[33] I.J. Li, J.C. Chen, J.L. Wu, *A fast prototype reduction method based on template reduction and visualization-induced Self-Organizing Map for nearest neighbor algorithm*, Applied Intelligence, Vol.39(3):564–58, 2013.

[34] L.A. Silva, E. Del-Moral-Hernandez, *A SOM combined with KNN for Classification Task*, Proceedings of the International Joint Conference on Neural Networks, Califianor, USA, 2011.

[35] I. Hammami, G. Mercier, A. Hamouda, J. Dezert, *Kohonen's map approach for the belief mass modeling*, IEEE Trans. Neural Network and Learning System, Vol. 27(10):2060–2071, 2016.

[36] J. Dezert, A. Tchamova, *On the validity of Dempster's fusion rule and its interpretation as a generalization of Bayesian fusion rule, International Journal of Intelligent Systems*, Vol. 29(3):223–252, 2014.

[37] M. Zaffalon, G. Corani, D. Maua, *Evaluating credal classifiers by utility-discounted predictive accuracy*, International Journal of Approximate Reasoning, Vol. 53:1282–1301, 2012.

Table III
CLASSIFICATION RESULTS OF THE 3-CLASS DATA SET BY DIFFERENT METHODS (IN %).

Training samples		N=1500	N=3000	N=6000
K-NN	R_e	6.45±0.43	6.25±0.39	6.21±0.32
	U	92.55±0.43	93.75±0.39	93.79±0.32
	T	40.63±5.42	258.59±15.75	904.69±18.90
FK-NN	R_e	6.63±0.46	6.33±0.40	6.28±0.38
	U	93.37±0.46	93.67±0.40	93.72±0.38
	T	49.21±4.152	287.5±20.51	975.78±26.73
EK-NN	R_e	6.28±0.48	6.12±0.29	6.16±0.35
	U	93.72±0.48	93.88±0.29	93.84±0.35
	T	47.09±4.78	279.76±20.77	931.96±32.85
SOM-KNN	R_e	8.27±0.41	8.23±0.59	8.15±0.61
	U	91.73±0.41	91.77±0.59	91.85±0.61
	T	32.81±5.30	109.38±6.15	445.62±2.80
HCS	R_e	5.07±0.33	4.83±0.45	4.81±0.47
	R_{i_2}	3.2±1.23	3.03±0.49	3.08±0.73
	U	93.81±0.1	94.11±0.27	94.11±0.22
	T	40.63±4.42	95.31±19.72	267.19±28.09

Table V
CLASSIFICATION RESULTS OF DIFFERENT METHODS (IN%) WITH REAL DATA SETS.

	ENN [12]	K-NN [1]	FK-NN [2]	EK-NN [11]	SOM-KNN [34]	BK-NN [16]	HCS
H R_e	42.47±0.86	35.42±1.05	36.20±0.68	35.35±0.87	40.86±0.57	35.13± 1.25	33.52±0.26
H R_{i_2}	NA	NA	NA	NA	NA	2.35±1.81	3.89±0.76
H U	57.53±0.86	64.58±1.05	63.80±0.68	64.65±0.87	59.14±0.57	64.05±0.51	65.12±0.54
H T	11.35±0.24	2.20±0.14	2.82±0.54	3.14±0.73	3.26±0.39	2.33±0.35	2.05±0.33
B R_e	3.09±0.01	3.03±0.16	2.91±0.18	2.86±0.18	4.07±0.16	2.85±0.23	2.51±0.20
B R_{i_2}	NA	NA	NA	NA	NA	1.43±0.23	1.53±0.12
B U	96.91±0.01	96.97±0.16	97.09±0.18	97.14±0.18	95.93±0.16	96.55±0.31	96.95±0.18
B T	24.84±2.33	5.32±2.5	6.43±1.05	6.97±3.19	4.06±1.88	9.53±1.6	4.53±1.25
I R_e	5.50±1.67	4.03±0.69	3.89±0.47	3.67±0.47	4.13±0.28	2.10±1.13	2.67±0.47
I R_{i_2}	NA	NA	NA	NA	NA	5.78±1.35	2.67±0.01
I U	94.50±1.67	95.97±0.69	96.11±0.47	96.33±0.47	93.27±1.15	95.88±0.86	96.40±0.16
I T	8.28±0.66	0.31±0.11	0.44±0.15	0.47±0.14	0.56±0.19	0.50±0.16	0.38±0.13
S R_e	11.33±0.20	10.69±0.86	10.69±0.33	10.43±0.84	8.81±0.46	7.68±1.10	5.48±0.34
S R_{i_2}	NA	NA	NA	NA	NA	2.16±1.85	3.62±0.78
S U	88.67±0.20	89.31±0.86	89.31±0.33	89.57±0.84	91.19±0.46	91.56±0.69	93.25±0.41
S T	15.03±1.25	0.66±0.15	2.40±0.45	1.01 ±0.43	0.50±0.13	1.21±0.19	0.56±0.13
W R_e	29.83±2.16	31.97±1.54	25.76±0.76	28.65±2.43	29.89±0.23	21.01±1.89	18.06±1.18
W R_{i_2}	NA	NA	NA	NA	NA	11.55±3.16	16.67±1.57
W U	70.17±2.16	68.03±1.54	74.24±0.76	71.35±2.43	69.08±1.5	74.95±1.57	76.11±1.21
W T	20.44±1.24	0.18±0.17	0.89±0.26	0.57±0.15	0.33±0.21	10.32±1.20	0.19±0.13
Y R_e	43.41±0.76	35.11±0.93	33.09±1.10	34.66±0.90	37.07±0.43	28.23±2.68	26.85±0.93
Y R_{i_2}	NA	NA	NA	NA	NA	14.12±2.33	14.76±0.20
Y U	56.59±0.76	64.89±0.93	66.91±1.10	65.34±0.90	62.93±0.43	66.83±1.38	67.94±0.86
Y T	76.08±10.10	13.71±0.21	19.79±1.61	15.54±0.82	9.30±8.11	23.63±2.10	5.63±1.09
V R_e	56.62±0.87	36.88±1.83	37.59±1.03	36.93±1.75	37.88±0.52	33.76±0.72	33.71±0.01
V R_{i_2}	NA	NA	NA	NA	NA	5.72±1.50	6.06±0.01
V U	43.38±0.87	63.12±1.83	62.41±1.03	63.07±1.75	62.12±0.52	64.24±0.83	64.17±0.11
V T	149.06±5.11	11.44±0.33	13.41±0.52	12.57±3.66	10.16±5.81	35.26±4.07	11.06±7.43
K R_e	22.59±0.91	13.09±1.01	12.79±0.53	12.73±0.43	16.20±0.71	11.93±1.12	10.24±1.90
K R_{i_2}	NA	NA	NA	NA	NA	2.12±1.02	2.93±1.55
K U	77.41±0.91	86.91±1.01	87.21±0.53	87.27±0.43	83.80±0.71	87.33±1.03	88.73±1.81
T T	17.89±0.76	3.40±0.30	3.59±0.67	3.42±1.63	3.22±1.67	9.22±1.72	3.25±0.56