

Article

# EEkNN: $k$ -Nearest Neighbor Classifier with an Evidential Editing Procedure for Training Samples <sup>†</sup>

Lianmeng Jiao \* , Xiaojiao Geng and Quan Pan

School of Automation, Northwestern Polytechnical University, Xi'an 710072, China;  
xiaojiaogeng@mail.nwpu.edu.cn (X.G.); panquannpu@gmail.com (Q.P.)

\* Correspondence: jiaolianmeng@nwpu.edu.cn; Tel.: +86-29-8843-1306

<sup>†</sup> This paper is an extended version of our paper published in the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty.

Received: 23 April 2019; Accepted: 22 May 2019; Published: 27 May 2019



**Abstract:** The  $k$ -nearest neighbor ( $k$ NN) rule is one of the most popular classification algorithms applied in many fields because it is very simple to understand and easy to design. However, one of the major problems encountered in using the  $k$ NN rule is that all of the training samples are considered equally important in the assignment of the class label to the query pattern. In this paper, an evidential editing version of the  $k$ NN rule is developed within the framework of belief function theory. The proposal is composed of two procedures. An evidential editing procedure is first proposed to reassign the original training samples with new labels represented by an evidential membership structure, which provides a general representation model regarding the class membership of the training samples. After editing, a classification procedure specifically designed for evidently edited training samples is developed in the belief function framework to handle the more general situation in which the edited training samples are assigned dependent evidential labels. Three synthetic datasets and six real datasets collected from various fields were used to evaluate the performance of the proposed method. The reported results show that the proposal achieves better performance than other considered  $k$ NN-based methods, especially for datasets with high imprecision ratios.

**Keywords:** pattern classification;  $k$ -nearest-neighbor classifier; fuzzy editing; evidential editing; belief function theory

## 1. Introduction

Classification of patterns is an important area of research and practical applications in a variety of fields including biology [1], psychology [2], medicine [3], electronics [4], marketing [5], military affairs [6], etc. In the past several decades, a wide variety of approaches has been developed towards this task [7]. As a type of lazy learning algorithm, the  $k$ -nearest neighbor ( $k$ NN) rule introduced by Fix and Hodges [8] has been one of the most popular and successful pattern classification techniques due to its simplicity and validity. The basic idea of the  $k$ NN rule is that patterns close in feature space are likely to belong to the same class. Though the  $k$ NN rule is suboptimal, it has been shown that as  $k$  increases, its error rate approaches the optimal Bayes error rate asymptotically in the infinite sample situation [9].

However, in the practical cases of a finite number of samples, the classical  $k$ -NN rule is not always the optimal way of utilizing the information contained in the neighborhood of query patterns, and therefore, a large number of research works focused on the improvement of this rule in the past 60 years [10–15]. One of the major concerns when using the  $k$ NN rule is that all of the training samples are considered equally important for assigning the class label of the query pattern. This limitation will result in great difficulty for classification in regions where the samples from different classes overlap.

Atypical samples in overlapping regions may be assigned as much weight as those that are truly representative of the clusters. Furthermore, it may be argued that training samples with great noise should not be given equal weight. In order to overcome this difficulty, many editing procedures have been proposed to preprocess the original training samples and then to make classification based on the edited training set [16–29].

Based on different structures of the edited labels, the editing procedures can be divided into two groups: crisp editing and soft editing. The editing procedure was firstly developed by Wilson [17] to preprocess the training samples. In this procedure, a training sample  $x_i$  is classified using the  $k$ NN rule with the remainder of the training set and is then deleted from the original training set if its original label does not agree with the classification result. After that, many others followed Wilson's work and proposed some variants [18–22]. One of the representatives is the generalized editing procedure developed by Koplowitz and Brown [19], aiming to overcome the limitations of large amounts of samples being removed from the training set. In their work, instead of deleting all the conflicting samples as Wilson's work, if a particular class (excluding the original class) has at least  $k'$  ( $(k + 1)/2 \leq k' \leq k$ ) representatives among these  $k$  nearest neighbors, then  $x_i$  is labeled according to that majority class. Essentially, both Wilson's editing and its variants are crisp editing procedures, in which each edited sample is either removed or assigned to a single class. In order to overcome the weakness of the crisp editing methods, a fuzzy editing procedure was then proposed to reassign fuzzy membership to each training sample  $x_i$  based on its  $k$  nearest neighbors [25]. Several different realizations of this fuzzy editing procedure have been also developed [26–28]. As a type of soft editing procedure, fuzzy editing makes it possible for each edited sample to be assigned to several classes with different fuzzy memberships, which provides more detailed information about the samples' membership than the crisp editing procedures.

In real-world classification problems, different types of uncertainty may coexist due to the environments or other interference factors, e.g., fuzziness may coexist with imprecision. The fuzzy editing procedure, developed based on fuzzy set theory [30], cannot address imprecise or partial information effectively in the modeling and reasoning processes. In contrast, the belief function theory [31–33], also known as Dempster–Shafer theory or evidence theory, offers a well-founded and effective framework to represent and combine a variety of uncertain information. This theory has already been used in  $k$ NN-based classification [34–39]. In [34], an evidential version of  $k$ NN, called  $E_k$ NN, has been proposed by the introduction of the ignorance class to model the uncertainty. Then, this classification method was further extended to deal with uncertainty using the rejection class and meta-classes in [37]. In [38], Dempster's rule of combination used in  $E_k$ NN was replaced by a class of parametric combination rules. However, neither the  $E_k$ NN method nor its extensions consider any editing procedure in the classification process. Recently, an editing procedure for multi-label classification was developed in [29] based on the belief function theory, but it is essentially a crisp editing procedure, as each edited sample is just assigned a single set of classes without considering the membership degrees.

In this paper, an evidential editing version of the  $k$ NN classifier ( $EE_k$ NN) is proposed based on the belief function theory (A preliminary version of some of the ideas introduced here was presented in [40,41]. The present paper is a deeply revised and extended version of this work, with several new results.). The proposed  $EE_k$ NN classifier is composed of two procedures: evidential editing for the original training samples and classification based on the evidently edited training samples. First, an evidential editing procedure is developed to reassign the original training samples with new labels represented by an evidential membership structure. Compared with the crisp label or the fuzzy membership, the evidential membership provides more expressiveness to represent the imprecision and uncertainty for those samples in overlapping regions or with great noise. After the editing procedure, a  $k$ NN classification procedure specifically designed for evidently edited training samples is developed in the belief function framework. This classification procedure can well handle the more general situation where the edited training samples are assigned dependent evidential labels.

The rest of this paper is organized as follows. In Section 2, the basics of the belief function theory are recalled. Then, the evidential editing procedure is developed in Section 3. After that, the classification procedure is designed and realized based on the edited training samples in the belief function framework in Section 4. Section 5 provides several experiments to test the proposed method. Finally, Section 6 concludes the paper. To facilitate reading, Table 1 gives a list of the symbols used and their definitions.

**Table 1.** List of symbols and definitions.

Symbol	Definitions
$k$ NN	$k$ nearest neighbor
$E$ $k$ NN	evidential $k$ nearest neighbor
$EE$ $k$ NN	evidential editing $k$ nearest neighbor
$FE$ $k$ NN	fuzzy editing $k$ nearest neighbor
$GE$ $k$ NN	generalized editing $k$ nearest neighbor
$k$	number of nearest neighbors in the classification process
$k_{edit}$	number of nearest neighbors in the editing process
$m$	mass function
$s$	Frank $t$ -norms parameter
$\mathcal{T}$	original training set
$\mathcal{T}'$	edited training set
$\mathcal{T}_i$	training set with $x_i$ excluded
$\mathbf{x}$	input feature vector
$\mathbf{y}$	query pattern
$\omega$	class label
$\Omega$	frame of discernment

## 2. Basics of the Belief Function Theory

In belief function theory [31–33], a problem domain is represented by a finite set  $\Omega = \{\omega_1, \omega_2, \dots, \omega_M\}$  of mutually exclusive and exhaustive hypotheses called the *frame of discernment*. A *mass function* expressing the belief committed to the elements of  $2^\Omega$  by a given source of evidence is a mapping function  $m: 2^\Omega \rightarrow [0, 1]$ , such that:

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

Elements  $A \subseteq \Omega$  having  $m(A) > 0$  are called the *focal sets* of the mass function  $m$ . The mass function has several special cases to encode different types of information. A mass function is said to be:

- *Bayesian*, if all of its focal sets are singletons. In this case, the mass function just reduces to the classical probability distribution.
- *categorical*, if the whole mass is allocated to one focal set  $A$ . This indicates that the truth lies in  $A$  with certainty.
- *certain*, if the whole mass is allocated to a unique singleton. This indicates that we have complete knowledge about the truth.
- *vacuous*, if the whole mass is allocated to  $\Omega$ . This situation corresponds to complete ignorance.
- *simple*, if it has at most two focal sets and one of them is  $\Omega$  if it has two. It is usually denoted as  $A^w$ , where  $A$  is the focal set different from  $\Omega$  and  $1 - w$  is the confidence that the truth lies in  $A$ .

After representing the available pieces of evidence as mass functions, the next step is to combine these mass functions into a single one for decision making. Many combination rules have been developed. The differences among them mainly depend on two issues: the dependence and the conflict among the available pieces of evidence.

Dempster’s rule is the most popular choice to combine several distinct pieces of evidence [31]. Its combination of two mass functions  $m_1$  and  $m_2$  defined on the same frame of discernment  $\Omega$  is:

$$m_1 \oplus m_2(A) = \begin{cases} 0, & A = \emptyset \\ \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}, & A \in 2^\Omega \setminus \emptyset. \end{cases} \tag{2}$$

To combine mass functions induced by nondistinct pieces of evidence, a cautious rule and, more generally, a family of parameterized  $t$ -norm based rules were proposed in [42]:

$$m_1 \otimes_s m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A) \top_s w_2(A)}, \tag{3}$$

where  $m_1$  and  $m_2$  are separable mass functions, such that  $m_1 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A)}$  and  $m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_2(A)}$ .

The operator  $\top_s$  denotes Frank’s parameterized family of t-norms:

$$a \top_s b = \begin{cases} a \wedge b, & \text{if } s = 0 \\ ab, & \text{if } s = 1 \\ \log_s \left( 1 + \frac{(s^a - 1)(s^b - 1)}{s - 1} \right), & \text{otherwise,} \end{cases} \tag{4}$$

for all  $a, b \in [0, 1]$ , with  $s$  being a positive parameter. When  $s = 0$ , the t-norm-based rule reduces to the cautious rule, and when  $s = 1$ , it reduces to Dempster’s rule.

For the above combination rules, it is assumed that the pieces of evidence to be combined are fully reliable. However, when this assumption fails, there may exist large conflicts among the pieces of evidence, in which case the performance of the above combination rules degrades greatly. Dubois and Prade [43] proposed an alternative rule to the combination of pieces of conflicting evidence as:

$$m_1 \odot m_2(A) = \begin{cases} 0, & A = \emptyset \\ \sum_{B \cap C = A} m_1(B)m_2(C) + \sum_{\substack{B \cap C = \emptyset, \\ B \cup C = A}} m_1(B)m_2(C), & A \in 2^\Omega \setminus \emptyset. \end{cases} \tag{5}$$

This rule boils down to Dempster’s rule when there is no conflict between the two combined pieces of evidence.

For decision making, Smets [33] proposed the pignistic transformation to transform a mass function into a probability function as:

$$BetP(A) = \sum_{B \cap A \neq \emptyset} \frac{|A \cap B|}{|B|} m(B), \quad A \in 2^\Omega, \tag{6}$$

where  $|X|$  is the cardinality of set  $X$ .

### 3. Evidential Editing Procedure for Training Samples

Let us consider an  $M$ -class classification problem in a predefined category  $\Omega = \{\omega_1, \dots, \omega_M\}$ . Assuming that a set of  $N$  labeled training samples  $\mathcal{T} = \{(\mathbf{x}_1, \omega^{(1)}), \dots, (\mathbf{x}_N, \omega^{(N)})\}$  with input vectors  $\mathbf{x}_i \in \mathbb{R}^P$  and class labels  $\omega^{(i)} \in \Omega$  is available, the editing procedure aims to generate a new edited training set  $\mathcal{T}'$ , which is more powerful than the original one for classification. In this section, we develop an evidential editing procedure for training samples in the belief function framework. First, in Section 3.1, an evidential membership structure is introduced as a general representation model for class membership. Then, in Section 3.2, an evidential editing algorithm is proposed to edit the training samples based on the evidential membership structure.

### 3.1. Evidential Membership Structure

The purpose of the evidential editing procedure is to assign to each sample in the training set  $\mathcal{T}$  a new soft label represented by an evidential membership structure as:

$$\mathcal{T}' = \{(\mathbf{x}_1, m_1), (\mathbf{x}_2, m_2), \dots, (\mathbf{x}_N, m_N)\}, \tag{7}$$

where  $m_i, i = 1, 2, \dots, N$ , are mass functions defined on the frame of discernment  $\Omega$ .

The above evidential membership modeled by mass function  $m_i$  provides a general representation model regarding the class membership of sample  $\mathbf{x}_i$ :

- when  $m_i$  is a *Bayesian* mass function, the evidential membership reduces to the fuzzy membership as a special case.
- when  $m_i$  is a *categorical* mass function, the evidential membership reduces to the crisp set of labels as defined in [29].
- when  $m_i$  is a *certain* mass function, the evidential membership reduces to the crisp label.
- when  $m_i$  is a *vacuous* mass function, the sample  $\mathbf{x}_i$  is useless for classification and can be considered as an outlier.

**Example 1.** Let us consider a set of  $N = 5$  samples  $\mathcal{T}' = \{(\mathbf{x}_1, m_1), (\mathbf{x}_2, m_2), (\mathbf{x}_3, m_3), (\mathbf{x}_4, m_4), (\mathbf{x}_5, m_5)\}$  with evidential membership regarding a set of  $M = 3$  classes  $\Omega = \{\omega_1, \omega_2, \omega_3\}$ . Mass functions for each sample are given in Table 2. They illustrate various situations: the case of sample  $\mathbf{x}_1$  corresponds to the situation of probabilistic uncertainty ( $m_1$  is Bayesian), whereas the case of sample  $\mathbf{x}_2$  corresponds to the situation of imprecision ( $m_2$  is categorical); the class of sample  $\mathbf{x}_3$  is known with precision and certainty ( $m_3$  is certain), whereas the class of sample  $\mathbf{x}_4$  is completely unknown ( $m_4$  is vacuous); finally, the mass function  $m_5$  models the general situation where the class of sample  $\mathbf{x}_5$  is both imprecise and uncertain.

**Table 2.** Example of the evidential membership.

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$	$m_5(A)$
$\emptyset$	0	0	0	0	0
$\{\omega_1\}$	0.2	0	0	0	0
$\{\omega_2\}$	0.3	0	0	0	0.1
$\{\omega_1, \omega_2\}$	0	0	0	0	0
$\{\omega_3\}$	0.5	0	1	0	0.2
$\{\omega_1, \omega_3\}$	0	0	0	0	0
$\{\omega_2, \omega_3\}$	0	1	0	0	0.4
$\Omega$	0	0	0	1	0.3

As illustrated in the above example, the evidential membership is a powerful model to represent the imprecise and uncertain information existing in the training samples. In the following part, we will study how to edit each training sample with the evidential membership.

### 3.2. Evidential Editing Algorithm

For each training sample  $\mathbf{x}_i, i = 1, 2, \dots, N$ , we denote the leave-it-out training set as  $\mathcal{T}_i = \mathcal{T} \setminus \{(\mathbf{x}_i, \omega^{(i)})\}, i = 1, 2, \dots, N$ . Now, we will show how the evidential editing procedure works for one training sample  $\mathbf{x}_i$  based on the other samples contained in  $\mathcal{T}_i$ . The evidence modeling

method developed in [34] is used here to generate a mass function for each neighbor  $\mathbf{x}_j$  regarding the class membership of  $\mathbf{x}_i$  as:

$$\begin{cases} m_i(\{\omega_q\} | \mathbf{x}_j) &= \alpha \phi_q(d_{ij}) \\ m_i(\Omega | \mathbf{x}_j) &= 1 - \alpha \phi_q(d_{ij}) \\ m_i(A | \mathbf{x}_j) &= 0, \forall A \in 2^\Omega \setminus \{\{\omega_q\}, \Omega\}, \end{cases} \tag{8}$$

where  $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\omega_q$  is the class label of  $\mathbf{x}_j$  (i.e.,  $\omega^{(j)} = \omega_q$ ), and  $\alpha$  is a parameter such that  $0 < \alpha < 1$ . A recommended value of  $\alpha = 0.95$  can be used to obtain good results on average, and a good choice for  $\phi_q$  is:

$$\phi_q(d) = \exp(-\gamma_q d^2), \tag{9}$$

where  $\gamma_q$  is a positive parameter associated with class  $\omega_q$ , and it can be set to the inverse of the mean squared distance between training samples belonging to class  $\omega_q$  heuristically.

Based on the distance  $d(\mathbf{x}_i, \mathbf{x}_j)$ , we first select  $k_{edit}$  nearest neighbors of  $\mathbf{x}_i$  in training set  $\mathcal{T}_i$  and construct the corresponding  $k_{edit}$  mass functions according to the above way. These  $k_{edit}$  mass functions are then combined to form a resulting mass function  $m_i$ , synthesizing the final evidential membership regarding the class of  $\mathbf{x}_i$ . Considering the different degrees of conflict among the constructed mass functions, we developed a hierarchical combination process that is carried out at two levels: intra-class combination and inter-class combination.

At the first level, we consider the combination of mass functions derived from the neighbors with the same class label. As all the mass functions to be combined support the same class, there is no conflict among them. Besides, as the training samples are usually collected independently, the items of evidence from different neighbors are independent. In this case, Dempster’s rule is a good choice for its effectiveness and simplicity. If we denote by  $\Psi_i^q$  the set of the  $k$  nearest neighbors of  $\mathbf{x}_i$  belonging to class  $\omega_q$  and assuming that  $\Psi_i^q$  is not empty, the intra-class combination for mass functions derived from the neighbors with class label  $\omega_q$  is given by:

$$m_i(\cdot | \Psi_i^q) = \bigoplus_{\mathbf{x}_j \in \Psi_i^q} m_i(\cdot | \mathbf{x}_j). \tag{10}$$

As shown in Equation (8), all the mass functions to be combined are *simple*. Thanks to this particular structure, the computational burden of Dempster’s rule can be greatly reduced, and the above intra-class combination can be further formulated analytically as:

$$\begin{cases} m_i(\{\omega_q\} | \Psi_i^q) &= 1 - \prod_{\mathbf{x}_j \in \Psi_i^q} m_i(\Omega | \mathbf{x}_j) \\ m_i(\Omega | \Psi_i^q) &= \prod_{\mathbf{x}_j \in \Psi_i^q} m_i(\Omega | \mathbf{x}_j) \\ m_i(A | \Psi_i^q) &= 0, \forall A \in 2^\Omega \setminus \{\{\omega_q\}, \Omega\}. \end{cases} \tag{11}$$

If  $\Psi_i^q$  is an empty set, then  $m_i(\cdot | \Psi_i^q)$  is simply a *vacuous* mass function satisfying  $m_i(\Omega | \Psi_i^q) = 1$ .

After the intra-class combination for mass functions derived from the neighbors belonging to each class, at the second level, we combine these sub-combination results to get a global combination result as the final evidential membership regarding the class of  $\mathbf{x}_i$ . As these sub-combination results support different classes, large conflicts may exist among them. In this case, Dubois–Prade’s rule is a good alternative combination method. However, when the number of classes is large, Dubois–Prade’s rule of combination for all the sub-combination results will generate a great number of focal sets (as many as  $2^M - 1$ ), which results in overmuch imprecision for the edited label. Therefore, at the inter-class combination level, if there is more than one mass function having non-zero mass for the support class, we only combine those two having largest mass as:

$$m_i = m_i(\cdot | \Psi_i^{q_1}) \odot m_i(\cdot | \Psi_i^{q_2}), \tag{12}$$

where  $m_i(\{\omega_{q_1}\} | \Psi_i^{q_1}) \geq m_i(\{\omega_{q_2}\} | \Psi_i^{q_2}) \geq m_i(\{\omega_q\} | \Psi_i^q)$ ,  $q = 1, 2, \dots, M, q \neq q_1, q \neq q_2$ . Noting that the sub-combination results shown in Equation (11) are also *simple* mass functions, the above inter-class combination can be further formulated analytically as:

$$\begin{cases} m_i(\{\omega_{q_1}\}) &= m_i(\{\omega_{q_1}\} | \Psi_i^{q_1})m_i(\Omega | \Psi_i^{q_2}) \\ m_i(\{\omega_{q_2}\}) &= m_i(\{\omega_{q_2}\} | \Psi_i^{q_2})m_i(\Omega | \Psi_i^{q_1}) \\ m_i(\{\omega_{q_1}, \omega_{q_2}\}) &= m_i(\{\omega_{q_1}\} | \Psi_i^{q_1})m_i(\{\omega_{q_2}\} | \Psi_i^{q_2}) \\ m_i(\Omega) &= m_i(\Omega | \Psi_i^{q_1})m_i(\Omega | \Psi_i^{q_2}) \\ m_i(A) = 0, \quad \forall A \in 2^\Omega \setminus \{\{\omega_{q_1}\}, \{\omega_{q_2}\}, \{\omega_{q_1}, \omega_{q_2}\}, \Omega\}. \end{cases} \tag{13}$$

If there is only one mass function having non-zero mass for the support class, then  $m_i$  is simply the same as  $m_i(\cdot | \Psi_i^{q_1})$ . Algorithm 1 shows the pseudocode of the evidential editing algorithm.

---

**Algorithm 1** Evidential editing algorithm.

---

**Require:** the original training set  $\mathcal{T} = \{(\mathbf{x}_1, \omega^{(1)}), \dots, (\mathbf{x}_N, \omega^{(N)})\}$  with  $\mathbf{x}_i \in \mathbb{R}^P$  and  $\omega^{(i)} \in \{\omega_1, \dots, \omega_M\}$ , the number of nearest neighbors  $k_{edit}$

- 1: Initialize  $\mathcal{T}' \leftarrow \emptyset$ ;
- 2: **for**  $i = 1-N$  **do**
- 3:   Find  $k_{edit}$  nearest neighbors of  $\mathbf{x}_i$  in  $\mathcal{T} \setminus \{(\mathbf{x}_i, \omega^{(i)})\}$ ;
- 4:   Generate a mass function  $m_i(\cdot | \mathbf{x}_j)$  for each neighbor  $\mathbf{x}_j$  using Equations (8)–(9);
- 5:   **for**  $q = 1$  to  $M$  **do**
- 6:     Combine mass functions derived from the neighbors belonging to class  $\omega_q$  to get a sub-combination result  $m_i(\cdot | \Psi_i^q)$  using Equation (11);
- 7:   **end for**
- 8:   Combine all the sub-combination results to get a global combination result  $m_i$  using Equation (13);
- 9:    $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{(\mathbf{x}_i, m_i)\}$ ;
- 10: **end for**
- 11: **return** the edited training set  $\mathcal{T}'$

---

**Example 2.** Figure 1 illustrates a simplified three-class classification example in the two-dimensional plane. A total number of thirteen training samples was collected with  $\mathbf{x}_1$ – $\mathbf{x}_5$  belonging to class  $\omega_1$ ,  $\mathbf{x}_6$ – $\mathbf{x}_9$  belonging to class  $\omega_2$ , and  $\mathbf{x}_{10}$ – $\mathbf{x}_{13}$  belonging to class  $\omega_3$ . We consider the evidential editing process for sample  $\mathbf{x}_1$  based on the information from the other samples. In this example, the number of nearest neighbors  $k_{edit}$  was set to five. Based on the Euclidean distance, five samples  $\mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_8, \mathbf{x}_{12}$  were selected, and the corresponding five mass functions were constructed using Equations (8) and (9) regarding the class membership of  $\mathbf{x}_1$  as:

$$\begin{aligned} m_1(\{\omega_1\} | \mathbf{x}_3) &= 0.751, & m_1(\Omega | \mathbf{x}_3) &= 0.249 \\ m_1(\{\omega_1\} | \mathbf{x}_5) &= 0.751, & m_1(\Omega | \mathbf{x}_5) &= 0.249 \\ m_1(\{\omega_2\} | \mathbf{x}_6) &= 0.751, & m_1(\Omega | \mathbf{x}_6) &= 0.249 \\ m_1(\{\omega_2\} | \mathbf{x}_8) &= 0.751, & m_1(\Omega | \mathbf{x}_8) &= 0.249 \\ m_1(\{\omega_3\} | \mathbf{x}_{12}) &= 0.428, & m_1(\Omega | \mathbf{x}_{12}) &= 0.572. \end{aligned}$$

The above mass functions were then combined at two levels sequentially. At the intra-class combination level, we combined those mass functions derived from the neighbors with the same class label using Equation (11) and obtained the sub-combination results as:

$$\begin{aligned} m_1(\{\omega_1\} | \{\mathbf{x}_3, \mathbf{x}_5\}) &= 0.938, & m_1(\Omega | \{\mathbf{x}_3, \mathbf{x}_5\}) &= 0.062 \\ m_1(\{\omega_2\} | \{\mathbf{x}_6, \mathbf{x}_8\}) &= 0.938, & m_1(\Omega | \{\mathbf{x}_6, \mathbf{x}_8\}) &= 0.062 \\ m_1(\{\omega_3\} | \{\mathbf{x}_{12}\}) &= 0.428, & m_1(\Omega | \{\mathbf{x}_{12}\}) &= 0.572. \end{aligned}$$

Next, at the second level, we combined the above sub-combination results to get a global one. In this step, only the two mass functions having largest mass for the support class, i.e.,  $m_1(\cdot | \{x_3, x_5\})$  and  $m_1(\cdot | \{x_6, x_8\})$ , were combined using Equation (13) to get the final evidential membership regarding the class of  $x_i$  as:

$$m_1(\{\omega_1\}) = 0.058, \quad m_1(\{\omega_2\}) = 0.058, \\ m_1(\{\omega_1, \omega_2\}) = 0.880, \quad m_1(\Omega) = 0.004.$$

It can be seen that the focal set  $\{\omega_1, \omega_2\}$  obtained the largest mass. This indicates that the sample  $x_1$  had a great chance of being in the overlapping region of class  $\omega_1$  and class  $\omega_2$ , which is consistent with the actual situation.

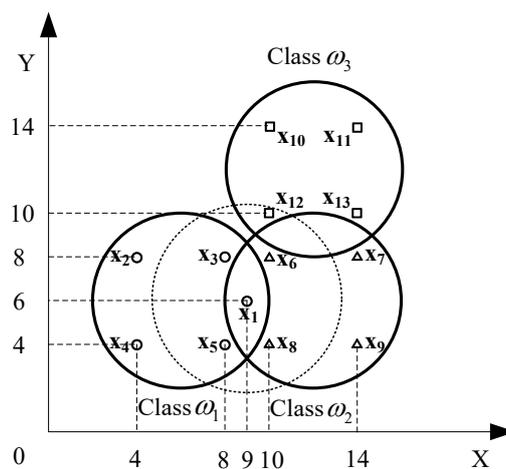


Figure 1. A simplified three-class classification example.

#### 4. kNN Classification with Evidently Edited Training Samples

After the evidential editing procedure developed in Section 3, the problem now turns into classifying a query pattern  $y \in \mathbb{R}^P$  based on the evidently edited training set  $\mathcal{T}'$ . In this section, a classification procedure specifically designed for evidently edited training samples is developed in the belief function framework. This classification procedure is composed of the following two steps: evidence representation for the edited training samples and evidence combination for decision making.

##### 4.1. Evidence Representation for the Edited Training Samples

Assume that the  $k$  nearest neighbors of the query pattern  $y$  have been selected from the edited training set. Generally, one training sample  $x_i$  is a very reliable piece of evidence for the classification of  $y$  if it is very close to  $y$ . In contrast, if  $x_i$  is far from  $y$ , then it is not reliable evidence. In the belief function society, the discounting operation proposed by Shafer [32] is a common tool to address the partially reliable evidence.

Denote as  $m_i$  the evidential label of the training sample  $x_i$  and  $\beta_i$  the confidence degree of the class membership of  $y$  with respect to the training sample  $x_i$ . The evidence provided by  $x_i$  for the class membership of  $y$  is represented with a discounted mass function  $\beta_i m_i$  by discounting  $m_i$  with a rate  $1 - \beta_i$  as:

$$\begin{cases} \beta_i m_i(A) = \beta_i m_i(A), \quad \forall A \in 2^\Omega \setminus \Omega \\ \beta_i m_i(\Omega) = \beta_i m_i(\Omega) + (1 - \beta_i). \end{cases} \quad (14)$$

The confidence degree  $\beta_i$  is determined based on the distance  $d_i$  between  $x_i$  and  $y$ . Generally, a larger distance results in a smaller confidence degree, and therefore,  $\beta_i$  should be a decreasing function of  $d_i$ . A similar decreasing function with Equation (9) is used here to define the confidence degree  $\beta_i \in (0, 1]$  as:

$$\beta_i = \exp(-\lambda_i d_i^2), \tag{15}$$

where  $\lambda_i$  is a positive parameter associated with the training sample  $\mathbf{x}_i$  and is defined as:

$$\lambda_i = \left[ \sum_{A \in 2^\Omega \setminus \Omega} m_i(A) \bar{d}_A + m_i(\Omega) \bar{d} \right]^{-2}, \tag{16}$$

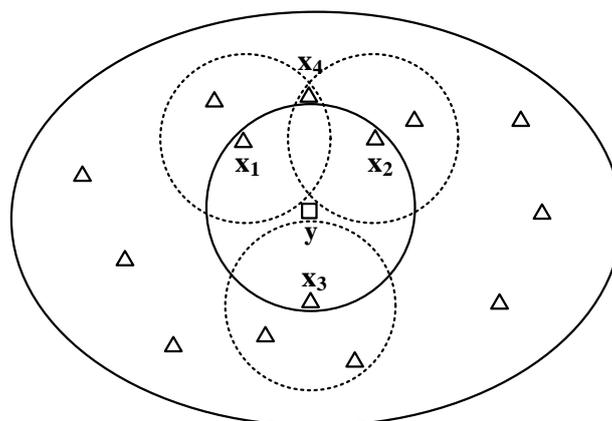
where  $\bar{d}$  is the mean distance among all training samples and  $\bar{d}_A$  is the mean distance among training samples belonging to class set  $A, \forall A \in 2^\Omega \setminus \Omega$ .

**Remark 1.** In calculating the confidence degree, parameter  $\lambda_i$  is designed by extending the parameter  $\gamma_q$  in Equation (9) to the cases of evidential labels. In Equation (16), if the label of the training sample  $\mathbf{x}_i$  is crisp with  $\omega_q$ , i.e.,  $m_i(\{\omega_q\}) = 1, m_i(A) = 0, \forall A \in 2^\Omega \setminus \{\omega_q\}$ , then the parameter  $\lambda_i$  just reduces to  $\gamma_q$  as a special case.

#### 4.2. Evidence Combination for Decision Making

In this section, we will combine the above generated  $k$  mass functions into a single one in order to make a decision about the class of the query pattern  $\mathbf{y}$ . The popular Dempster’s rule of combination relies on the assumption that the items of evidence to be combined are independent. However, as illustrated in the following example, the  $k$  mass functions derived from different edited samples cannot be regarded as fully independent any longer.

**Example 3.** Figure 2 illustrates the dependence among different edited training samples, where the training samples are denoted by “ $\Delta$ ” and the query pattern is denoted by “ $\square$ ”. In the evidential editing process,  $k_{edit} = 2$  was assumed to search for the nearest neighbors, and in the classification process, the number of nearest neighbors  $k = 3$  was assumed. We can see that  $\mathbf{x}_1, \mathbf{x}_2,$  and  $\mathbf{x}_3$  were the three nearest neighbors used for the classification of the query pattern  $\mathbf{y}$ . In the evidential editing process, as the training sample  $\mathbf{x}_4$  was used to calculate both the class membership of  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , the edited training samples  $\mathbf{x}_1$  and  $\mathbf{x}_2$  were no longer independent. In contrast, the edited training sample  $\mathbf{x}_3$  was still independent of both  $\mathbf{x}_1$  and  $\mathbf{x}_2$  as they did not use common training samples in the evidential editing process. Therefore, the items of evidence from different edited training samples may have partial dependence.



**Figure 2.** Illustration of dependence among edited training samples.

To account for this partial dependence, we used the parameterized t-norm-based rule shown in Equation (3) to combine the generated  $k$  mass functions to get the final result for query pattern  $\mathbf{y}$  as:

$$m = \beta_{i_1} m_{i_1} \otimes_s \beta_{i_2} m_{i_2} \otimes_s \dots \otimes_s \beta_{i_k} m_{i_k}, \tag{17}$$

where  $k$  is the number of nearest neighbors with  $i_1, i_2, \dots, i_k$  being the indices of the  $k$  nearest neighbors of  $\mathbf{y}$  in  $\mathcal{T}'$  and  $s$  is the Frank t-norms parameter defined in Equation (4). Different values of parameter  $s$  result in a series of combination rules ranging from the cautious rule ( $s = 0$ ) to the Dempster's rule ( $s = 1$ ). The selection of parameter  $s$  depends on the potential dependence of the edited training samples. A smaller value should be assigned to  $s$  for the case of larger dependence. In practice, we can use cross-validation to search for the optimal t-norm-based rule.

In order to make a decision based on the above combined mass function  $m$ , the pignistic probability  $BetP$  shown in Equation (6) was calculated. Finally, the query pattern  $\mathbf{y}$  was assigned to the class with the maximum pignistic probability.

## 5. Experiments

The performance of the proposed  $k$ NN classifier with evidential editing procedure (EEkNN) was evaluated using four different experiments. In the first experiment, the combination rules used in the classification process were evaluated under different dependence degrees of the edited samples. In the second experiment, the effects of the two main parameters  $k_{edit}$  and  $k$  in the editing and classification processes were analyzed. In the last two experiments, the performance of the EEkNN classifier was compared with those of other  $k$ NN-based methods, including the  $k$ NN classifier with generalized editing procedure (GEkNN) [19], the  $k$ NN classifier with fuzzy editing procedure (FEkNN) [25], and the evidential  $k$ NN classifier (EkNN) [34], using synthetic datasets and real datasets, respectively.

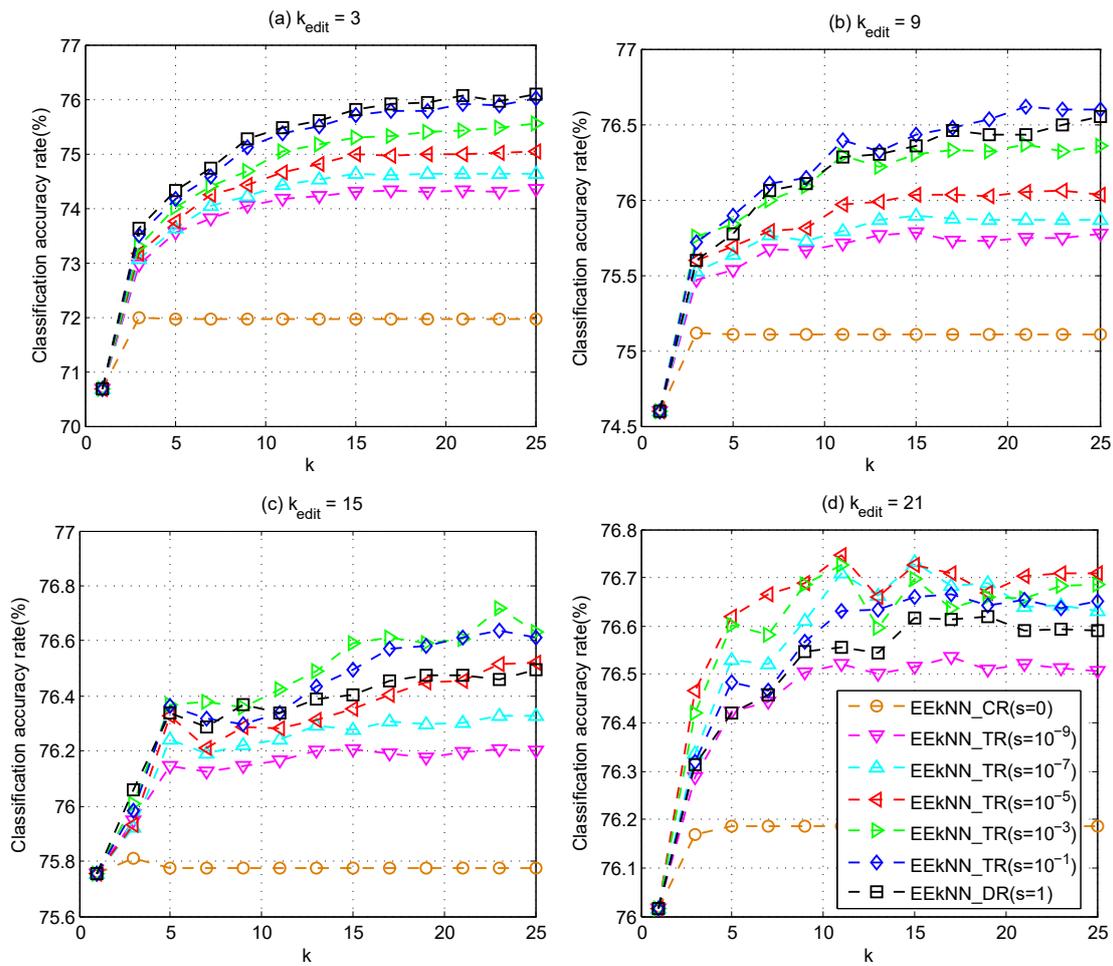
### 5.1. Evaluation of the Combination Rules

This experiment was designed to evaluate the combination rules used in the classification process of the EEkNN classifier. A two-dimensional three-class classification problem was considered. The following normal class-conditional distributions were assumed:

$$\begin{aligned} \text{Class A: } & \mu_A = (6, 6)^T, \Sigma_A = 4\mathbf{I}; \\ \text{Class B: } & \mu_B = (14, 6)^T, \Sigma_B = 4\mathbf{I}; \\ \text{Class C: } & \mu_C = (14, 14)^T, \Sigma_C = 4\mathbf{I}. \end{aligned}$$

A set of 150 training samples and a set of 3000 test samples were generated from the above distributions using equal prior probabilities. The average test classification rate over 30 independent trials was calculated. In the evidential editing process,  $k_{edit} = 3, 9, 15, 21$  were selected, and in the classification process, values of  $k$  ranging from 1–25 have been investigated. The t-norm-based rules (TR) with parameter  $s$  ranging from 0–1 have been evaluated (the cautious rule (CR) was retrieved when  $s = 0$ , and Dempster's rule (DR) was retrieved when  $s = 1$ ).

Figure 3 shows the classification accuracy for different combination rules. We note that the best combination rule varied with changes of the value of  $k_{edit}$ . In other words, the  $k_{edit}$  value had great influence on the dependence of the edited samples, and a larger  $k_{edit}$  value tended to result in larger dependence. For one specific classification problem, the selection of the best combination rule depends on the potential dependence of the edited samples, which further depends on the utilized  $k_{edit}$  value. Therefore, for the EEkNN classifier, the optimal t-norm-based rule should be searched for each specific  $k_{edit}$  value.



**Figure 3.** Classification results for different combination rules under different  $k_{edit}$  values with values of  $k$  ranging from 1–25.

### 5.2. Parameter Analysis

This experiment was designed to analyze the effect of parameters  $k_{edit}$  and  $k$  for the proposed EEKNN classifier. The same training and test samples with the previous experiment were used. The difference was that in the evidential editing process,  $k_{edit} = 3, 6, 9, 12, 15, 18, 21, 24$  were selected, and the optimal t-norm-based rule for each specific  $k_{edit}$  value was used to make the classification. Average classification accuracy over the 30 trials with values of  $k$  ranging from 1–25 has been investigated.

From Figure 4, we can see that the classification performance can improve clearly as the parameter  $k_{edit}$  increases within an interval ( $[3, 12]$  in this example). However, when  $k_{edit}$  exceeded an upper boundary ( $\overline{k_{edit}} = 12$  in this example), the classification performance no longer improved ideally. In addition, when  $k_{edit}$  took small values, the classification performance could improve as the parameter  $k$  increased. However, when  $k_{edit}$  exceeded the upper boundary, the parameter  $k$  had little effect on the classification performance.

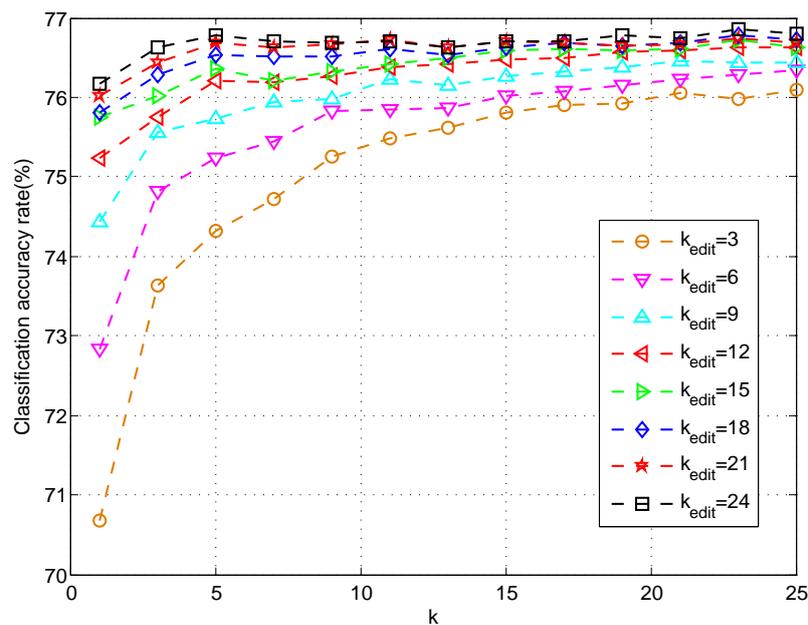


Figure 4. Classification results of the EEkNN classifier for different values of  $k_{edit}$  and  $k$ .

### 5.3. Synthetic Data Test

This experiment was designed to compare the proposed EEkNN classifier with other  $k$ NN-based classifiers using synthetic datasets with different class imprecision ratios, defined as the number of imprecise samples divided by the total number of training samples. A training sample  $x_i$  is considered to be imprecise if a non-singleton set gets the largest mass after the evidential editing procedure. A two-dimensional four-class classification problem was considered. The following normal class-conditional distributions were assumed. For comparisons, we changed the variance of each distribution to control the class imprecision ratio.

- Case 1** Class A:  $\mu_A = (0, 0)^T, \Sigma_A = \mathbf{I}$ ; Class B:  $\mu_B = (5, 0)^T, \Sigma_B = \mathbf{I}$ ;  
 Class C:  $\mu_C = (0, 5)^T, \Sigma_C = \mathbf{I}$ ; Class D:  $\mu_D = (5, 5)^T, \Sigma_D = \mathbf{I}$ . Imprecision ratio  $\rho = 33\%$
- Case 2** Class A:  $\mu_A = (0, 0)^T, \Sigma_A = 2\mathbf{I}$ ; Class B:  $\mu_B = (5, 0)^T, \Sigma_B = 2\mathbf{I}$ ;  
 Class C:  $\mu_C = (0, 5)^T, \Sigma_C = 2\mathbf{I}$ ; Class D:  $\mu_D = (5, 5)^T, \Sigma_D = 2\mathbf{I}$ . Imprecision ratio  $\rho = 60\%$
- Case 3** Class A:  $\mu_A = (0, 0)^T, \Sigma_A = 3\mathbf{I}$ ; Class B:  $\mu_B = (5, 0)^T, \Sigma_B = 3\mathbf{I}$ ;  
 Class C:  $\mu_C = (0, 5)^T, \Sigma_C = 3\mathbf{I}$ ; Class D:  $\mu_D = (5, 5)^T, \Sigma_D = 3\mathbf{I}$ . Imprecision ratio  $\rho = 79\%$

A training set of 200 samples and a test set of 4000 samples were generated from the above distributions using equal prior probabilities. For each case, 30 trials were performed with 30 independent training sets. The average classification accuracy and the corresponding 95% confidence interval were calculated. For each trial, the best values for the parameters  $k_{edit}$  and  $s$  in the EEkNN classifier were determined in the sets  $\{3, 6, 9, 12, 15, 18, 21, 24\}$  and  $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0\}$ , respectively, by cross-validation. For all of the considered method, values of  $k$  ranging from 1–25 have been investigated.

Figures 5–7 show the training set and the classification results for cases with different imprecision ratios. From the left three subfigures, we can see that the three cases corresponded to slight, moderate, and severe class overlapping, respectively. The average classification accuracy rates of different methods, as well as the corresponding 95% confidence intervals of the proposed one are shown in the right three subfigures. It can be seen that for all the considered three cases, the proposed EEkNN classifier provided better classification accuracy than other kNN-based ones, because in our proposed EEkNN classifier, the uncertainty of samples in overlapping regions can be well characterized with the introduction of the evidential editing procedure. We also notice that the performance improvement was more significant for Case 3, where the samples from different classes overlapped severely. Furthermore, different from other kNN-based classifiers, the proposed one was less sensitive to the value of  $k$ , and it performed well even with a small value of  $k$ .

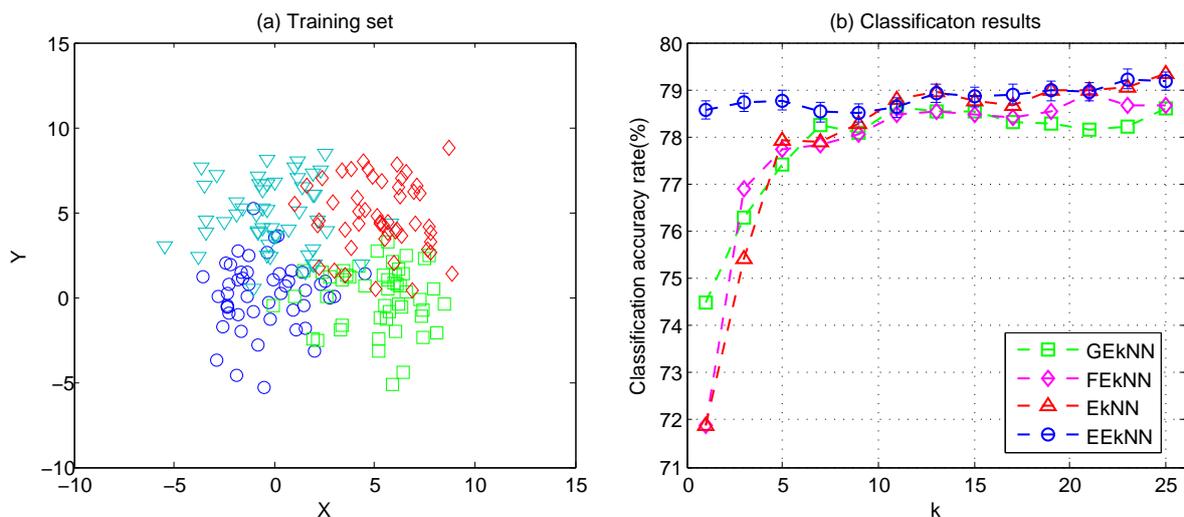


Figure 5. Training set and classification results for Case 1 with imprecision ratio  $\rho = 33\%$ .

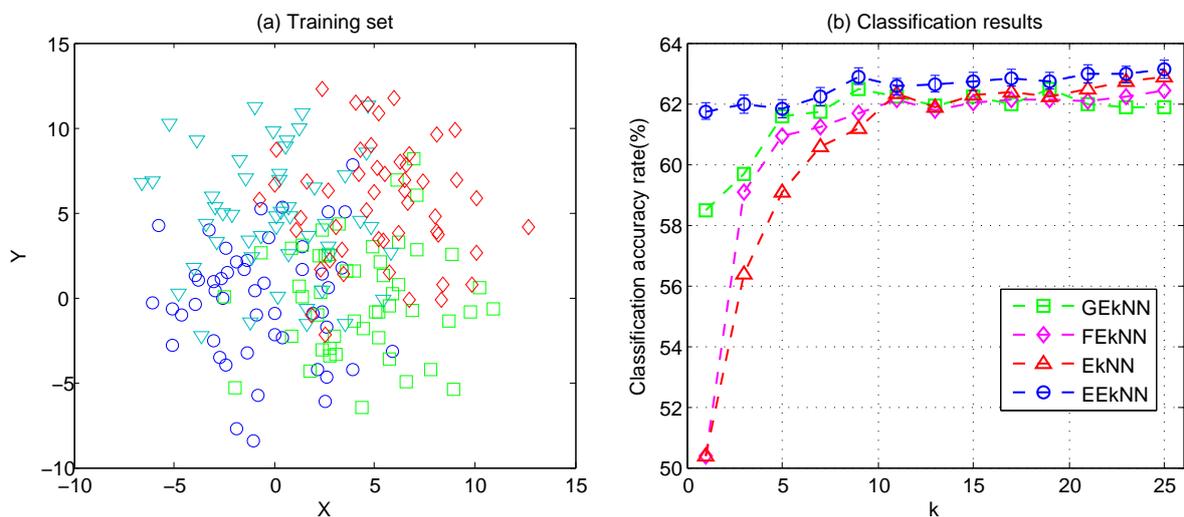


Figure 6. Training set and classification results for Case 2 with imprecision ratio  $\rho = 60\%$ .

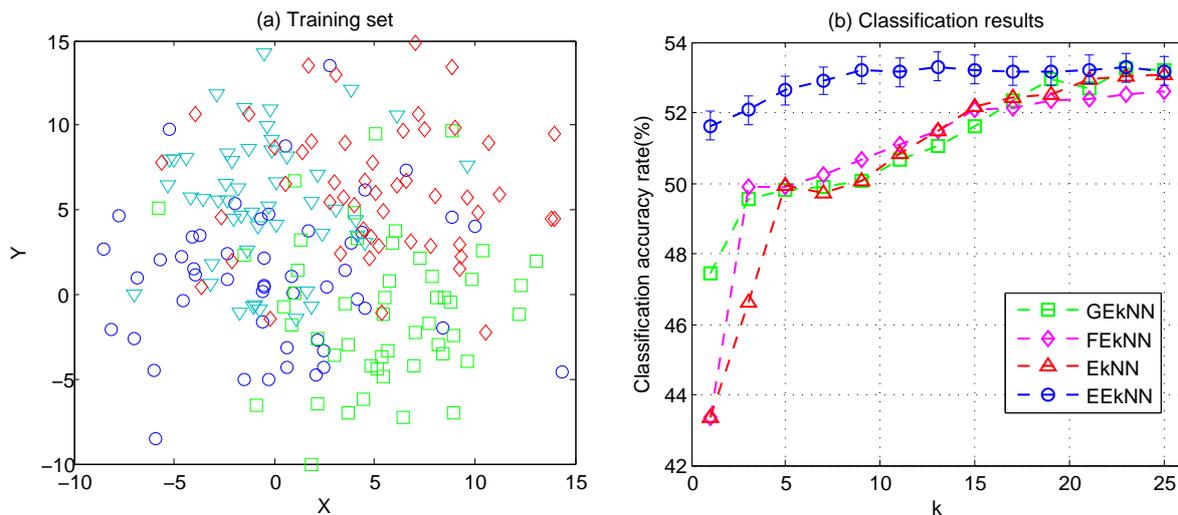


Figure 7. Training set and classification results for Case 3 with imprecision ratio  $\rho = 79\%$ .

#### 5.4. Real Data Test

This experiment was designed to compare the proposed EEkNN classifier with other kNN-based classifiers using some real-world classification problems from the well-known UCI Machine Learning Repository [44]. These datasets covered a variety of applications in many fields, i.e., biology, medicine, phytology, and astronomy. The main characteristics of the six real datasets used in this experiment are summarized in Table 3, where “# Samples” is the number of samples in the dataset, “# Features” is the number of features, and “# Classes” is the number of classes. To assess the results, we considered the resampled paired test. A series of 30 trials was conducted. In each trial, the available samples were randomly divided into a training set and a test set (with equal sizes). For each dataset, we calculated the average classification rate of the 30 trials and the corresponding 95% confidence interval. For the proposed EEkNN classifier, the best values for the parameters  $k_{edit}$  and  $s$  were determined with the same procedure used in the previous experiment. For all of the considered methods, values of  $k$  ranging from 1–25 have been investigated.

Table 3. Description of the real datasets employed in the study.

Dataset	# Samples	# Features	# Classes
Diabetes	393	8	2
Glass	214	9	6
Ionosphere	214	9	6
Seeds	210	7	3
Transfusion	748	4	2
Yeast	1484	8	10

Figure 8 shows the classification results of different methods for real datasets. It can be seen that, for most datasets, the EEkNN classifier provided better classification performance than other kNN-based ones. The reason is that in our proposed EEkNN classifier, the uncertainty of samples in overlapping regions or noisy patterns can be well characterized with the introduction of the evidential editing procedure. In the GEkNN classifier, however, each uncertain sample was either removed or assigned to a single class with great risk. Though in the FEkNN classifier, the fuzzy membership was reassigned to each uncertain sample, it could not address the involved imprecise information effectively. For the original EkNN classifier developed based on the belief function theory, the original training set was just used to make classification without considering any editing procedure. However, for dataset *Glass*, the classification performances of different methods were quite similar. The reason is that, for this

dataset, the best classification performance was obtained when  $k$  took a small value, and under this circumstance, the evidential editing procedure could not improve the classification performance.

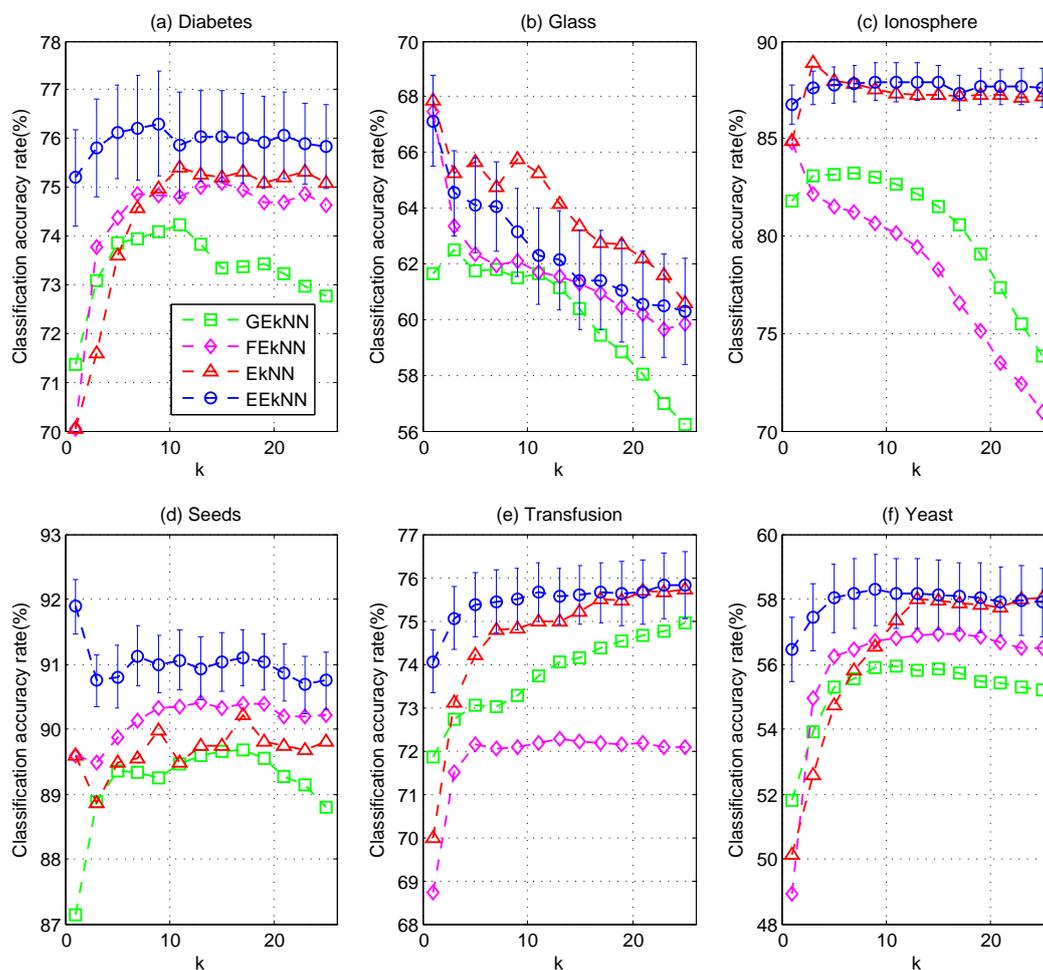


Figure 8. Classification results of different methods for real datasets.

## 6. Conclusions

An evidential editing version of the  $k$ NN classifier (EEkNN) has been developed based on an evidential editing procedure that reassigns the original training samples with new labels represented by an evidential membership structure. Thanks to this procedure, noisy patterns or those situated in overlapping regions had less influence on the decisions. In addition, in the subsequent classification procedure, the parameterized t-norm-based rule was optimized to combine the  $k$  nearest neighbors of one query pattern by taking into account the potential dependence among them. Experiments based on both synthetic and real datasets have been carried out to evaluate the performance of the proposal. From the results reported in the last section, we can conclude that the proposed EEkNN classifier can achieve higher classification accuracy than other considered  $k$ NN-based methods, especially for datasets with high imprecision ratios. Moreover, the proposed EEkNN classifier was not too sensitive to the value of  $k$ , and it could gain a quite good performance even with  $k = 1$ . This is an advantage in time- or space-critical applications, in which only a small value of  $k$  is permitted in the classification process.

The proposal can be potentially used in many classification applications where the available data are imperfect. For example, in brain–computer interface (BCI) systems [45], the electroencephalogram (EEG) signals may contain great uncertainties due to the varying brain dynamics and the presence

of noise. The proposed EEkNN classifier can minimize the effect of these uncertainties with the introduction of the evidential editing procedure for the raw data.

**Author Contributions:** L.J. conceived of the idea and designed the methodology. X.G. wrote the paper. Q.P. provided the laboratory support and improved the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (Grant Nos. 61790552 and 61801386) and the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2018JQ6043), the China Postdoctoral Science Foundation (Grant No. 2019M653743), and the Aerospace Science and Technology Foundation of China.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tran, T.T.; Choi, J.W.; Le, T.H.; Kim, J.W. A Comparative Study of Deep CNN in Forecasting and Classifying the Macronutrient Deficiencies on Development of Tomato Plant. *Appl. Sci.* **2019**, *9*, 1601. [[CrossRef](#)]
2. Seo, Y.S.; Huh, J.H. Automatic emotion-based music classification for supporting intelligent IoT applications. *Electronics* **2019**, *8*, 164. [[CrossRef](#)]
3. Iqbal, U.; Ying Wah, T.; Habib Ur Rehman, M.; Mastoi, Q. Usage of model driven environment for the classification of ECG features: A systematic review. *IEEE Access* **2018**, *6*, 23120–23136. [[CrossRef](#)]
4. Wu, C.; Yue, J.; Wang, L.; Lyu, F. Detection and classification of recessive weakness in superbuck converter based on WPD-PCA and probabilistic neural network. *Electronics* **2019**, *8*, 290. [[CrossRef](#)]
5. Donati, L.; Iotti, E.; Mordonini, G.; Prati, A. Fashion Product Classification through Deep Learning and Computer Vision. *Appl. Sci.* **2019**, *9*, 1385. [[CrossRef](#)]
6. Jiao, L.; Dencœux, T.; Pan, Q. A hybrid belief rule-based classification system based on uncertain training data and expert knowledge. *IEEE Trans. Syst. Man Cybern.* **2016**, *46*, 1711–1723. [[CrossRef](#)]
7. Jain, A.K.; Duin, R.P.W.; Mao, J. Statistical pattern recognition: A review *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 4–37. [[CrossRef](#)]
8. Fix, E.; Hodges, J. *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*; Technical Report 4; USAF School of Aviation Medicine: Randolph Field, TX, USA, 1951.
9. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
10. Dudani, S.A. The distance-weighted  $k$ -nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern.* **1976**, *4*, 325–327. [[CrossRef](#)]
11. Jiao, L.; Pan, Q.; Feng, X. Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric. *Neurocomputing* **2015**, *151*, 1468–1476. [[CrossRef](#)]
12. Tang, B.; He, H. ENN: Extended nearest neighbor method for pattern recognition. *IEEE Comput. Intell. Mag.* **2015**, *10*, 52–60. [[CrossRef](#)]
13. Yu, Z.; Chen, H.; Liu, J.; You, J.; Leung, H.; Han, G. Hybrid  $k$ -nearest neighbor classifier. *IEEE Trans. Cybern.* **2016**, *46*, 1263–1275. [[CrossRef](#)] [[PubMed](#)]
14. Ma, H.; Gou, J.; Wang, X.; Ke, J.; Zeng, S. Sparse coefficient-based  $k$ -nearest neighbor classification. *IEEE Access* **2017**, *5*, 16618–16634. [[CrossRef](#)]
15. Chatzigeorgakidis, G.; Karagiorgou, S.; Athanasiou, S.; Skiadopoulos, S. FML- $k$ NN: Scalable machine learning on Big Data using  $k$ -nearest neighbor joins. *J. Big Data* **2018**, *5*, 1–27. [[CrossRef](#)]
16. Devijver, P.; Kittler, J. *Pattern Recognition: A Statistical Approach*; Prentice Hall: Englewood Cliffs, NJ, USA, 1982.
17. Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data sets. *IEEE Trans. Syst. Man Cybern.* **1972**, *2*, 408–421. [[CrossRef](#)]
18. Tomek, I. An experiment with the edited nearest neighbor rule. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 121–126. [[CrossRef](#)]
19. Koplowitz, J.; Brown, T.A. On the relation of performance to editing in nearest neighbor rules. *Pattern Recognit.* **1981**, *13*, 251–255. [[CrossRef](#)]
20. Kuncheva, L. Editing for the  $k$ -nearest neighbors rule by a genetic algorithm. *Pattern Recognit. Lett.* **1995**, *16*, 809–814. [[CrossRef](#)]
21. Jiang, Y.; Zhou, Z. Editing training data for  $k$ NN classifiers with neural network ensemble. In *Advances in Neural Networks*; Yin, F., Wang, J., Guo, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 356–361.

22. Chang, R.; Pei, Z.; Zhang, C. A modified editing  $k$ -nearest neighbor rule. *J. Comput.* **2011**, *6*, 1493–1500. [[CrossRef](#)]
23. Triguero, I.; Derrac, J.; Garcia, S.; Herrera, F. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 86–100. [[CrossRef](#)]
24. Garcia, S.; Derrac, J.; Cano, J.; Herrera, F. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 417–435. [[CrossRef](#)]
25. Keller, J.; Gray, M.; Givens, J. A fuzzy  $k$ -nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 580–585. [[CrossRef](#)]
26. Yang, M.; Chen, C. On the edited fuzzy  $k$ -nearest neighbor rule. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1998**, *28*, 461–466. [[CrossRef](#)]
27. Zhang, C.; Cheng, J.; Yi, L. A method based on the edited FKNN by the threshold value. *J. Comput.* **2013**, *8*, 1821–1825. [[CrossRef](#)]
28. Liu, Z.; Pan, Q.; Dezert, J.; Mercier, G.; Liu, Y. Fuzzy-belief  $k$ -nearest neighbor classifier for uncertain data. In Proceedings of the 17th International Conference on Information Fusion, Salamanca, Spain, 7–10 July 2014; pp. 1–8.
29. Kanj, S.; Abdallah, F.; Dencœux, T.; Tout, K. Editing training data for multi-label classification with the  $k$ -nearest neighbor rule. *Pattern Anal. Appl.* **2015**, *19*, 145–161. [[CrossRef](#)]
30. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
31. Dempster, A. Upper and lower probabilities induced by multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339. [[CrossRef](#)]
32. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
33. Smets, P. Decision making in the TBM: The necessity of the pignistic transformation. *Int. J. Approx. Reason.* **2005**, *38*, 133–147. [[CrossRef](#)]
34. Dencœux, T. A  $k$ -nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. Syst. Man Cybern.* **1995**, *25*, 804–813. [[CrossRef](#)]
35. Dencœux, T.; Smets, P. Classification using belief functions relationship between case-based and model-based approaches. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2006**, *36*, 1395–1406. [[CrossRef](#)]
36. Jiao, L.; Pan, Q.; Feng, X.; Yang, F. An evidential  $k$ -nearest neighbor classification method with weighted attributes. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July 2013; pp. 145–150.
37. Liu, Z.; Pan, Q.; Dezert, J. A new belief-based  $k$ -nearest neighbor classification method. *Pattern Recognit.* **2013**, *46*, 834–844. [[CrossRef](#)]
38. Su, Z.; Dencœux, T.; Hao, Y.; Zhao, M. Evidential  $k$ -NN classification with enhanced performance via optimizing a class of parametric conjunctive  $t$ -rules. *Knowl. Based Syst.* **2018**, *142*, 7–16. [[CrossRef](#)]
39. Jiao, L.; Geng, X.; Pan, Q. BPkNN:  $k$ -nearest neighbor classifier with pairwise distance metrics and belief function theory. *IEEE Access* **2019**, *7*, 48935–48947. [[CrossRef](#)]
40. Jiao, L.; Dencœux, T.; Pan, Q. Evidential editing  $k$ -nearest neighbor classifier. In Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Compiègne, France, 15–17 July 2015; pp. 461–471.
41. Jiao, L. Classification of Uncertain Data in the Framework of Belief Functions: Nearest-Neighbor-Based and Rule-Based Approaches. Ph.D. Thesis, Université de Technologie de Compiègne, Compiègne, France, 2015.
42. Dencœux, T. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artif. Intell.* **2008**, *172*, 234–264. [[CrossRef](#)]
43. Dubois, D.; Prade, H. Representation and combination of uncertainty with belief functions and possibility measures. *Comput. Intell.* **1988**, *4*, 244–264. [[CrossRef](#)]
44. Dua, D.; Karra Taniskidou, E. UCI Machine Learning Repository, 2017. Available online: <http://archive.ics.uci.edu/ml> (accessed on 1 December 2017).
45. Katona, J.; Kovari, A. Examining the learning efficiency by a brain computer interface system. *Acta Polytech. Hung.* **2018**, *15*, 251–280.

