A Distributed Rough Evidential *K*-NN Classifier: Integrating Feature Reduction and Classification

Zhi-gang Su, Qinghua Hu, and Thierry Denœux

***Abstract*—The Evidential *K*-Nearest Neighbor (EK-NN) classification rule provides a global treatment of imperfect knowledge in class labels, but still suffers from the curse of dimensionality as well as runtime and memory restrictions when performing nearest neighbors search, in particular for large and high-dimensional data. To avoid the curse of dimensionality, this paper first proposes a rough evidential *K*-NN (REK-NN) classification rule in the framework of rough set theory. Based on a reformulated *K*-NN rough set model, REK-NN selects features and thus reduces complexity by minimizing a proposed neighborhood pignistic decision error rate, which considers both Bayes decision error and spatial information among samples in feature space. In contrast to existing rough set-based feature selection methods, REK-NN is a synchronized rule rather than a stepwise one, in the sense that feature selection and learning are performed simultaneously. In order to further handle data with large sample size, we derive a distributed REK-NN method and implement it in the Apache Spark. The theoretical analysis of the classifier generalization error bound is finally presented. It is shown that the distributed REK-NN achieves good performances while drastically reducing the number of features and consuming less runtime and memory. Numerical experiments conducted on real-world datasets validate our conclusions.**

***Index Terms*—Dempster-Shafer theory, feature selection, neighborhood rough set model, generalization error bound, nonparametric classification, Big Data, Apache Spark.**

I. Introduction

As a case-based learning method that does not need any prior assumptions [1], the voting K-NN classifier [2], assigning a sample into the class represented by a majority of *K* nearest neighbors in the training set, has been widely used in practice due to its efficiency and simplicity. To further enhance its performance, the evidential K-NN classifier (EK-NN) [3] was proposed in the conceptual framework of Dempster-Shafer theory of belief functions [4]–[10], a powerful tool for modeling and reasoning with uncertain and/or imprecise information.

In the EK-NN, each neighbor of a sample to be classified is considered as an item of evidence that supports certain hypotheses regarding the class membership of that sample. The degree of support/belief is defined as a function of the distance between two samples. The evidence of the *K* nearest neighbors is then pooled by means of Dempster's rule of combination [7]. In this way, the EK-NN classifier provides a global treatment of imperfect knowledge regarding the class membership of training samples, and thus became widely used in the Pattern Recognition community (see, for example, [6], [11], [12]).

The original EK-NN classifier has been improved in several ways. The first intuitive way is to optimize some parameters in the EK-NN by minimizing a certain error function. In [13], a gradient algorithm was proposed for that purpose, and more recently evolutionary algorithms were used in [14]. The second way is to apply different combination rules instead of Dempster's rule. One underlying motivation is that Dempster's rule assumes independence of the item of evidence, but this assumption seems hard to be guaranteed in practice. In [15], Pichon and Denoeux proposed a family of t-norm based combination rules (t-rules for short), including conjunctive and cautious rules as its two members [5], to deal with non-distinct or dependent items of evidence. It was demonstrated that better performance can be obtained by the EK-NN classifier using t-rules. Su and Denoeux proposed a class of parametric t-rules by introducing tunable parameters and functions [12]. The authors showed that better performances can be achieved for the EK-NN classifier by optimizing these parametric t-rules. Among other variants of the EK-NN, we can mention the hybrid classification rule [16], the ensemble enhanced EK-NN [17], [18] and the contextual discounting-based EK-NN [19]. Nevertheless, none of these methods address specifically the classification of data featuring high dimensionality and/or large sample size.

Dimensionality is known to be a crucial factor affecting the performance of a classifier, in particular of K-NN classifiers. As shown in [20], high dimensionality usually causes problems such as distance concentration and hubness when performing nearest neighbors search. Hence, applying EK-NN to high dimensional data is a major issue. Lian, Ruan and Denoeux [21] proposed to reduce dimensionality of the input space for the EK-NN classifier by extracting features from initial high-dimensional feature space. However, in many applications such as gene selection, users want to know which features play crucial roles in classification performance. Therefore, it is interesting to improve EK-NN through feature selection rather than feature extraction. In [22], the authors implemented feature selection in the EK-NN method by minimizing a $\{0,1\}$-binary weighted distance using a genetic algorithm; a feature is selected when its associated weight equals 1 after optimization. However, the complexity of this method does not allow it to be applied to very large data. How to implement the EK-NN on high-dimensional data with good performance

This work is supported in part by the National Natural Science Foundation of China under Grants 51676034, 51876035 and 51976032.

Z.-G Su is with the School of Energy and Environment, Southeast University, Nanjing, Jiangsu 210096, China (Corresponding e-mail: zhigang-su@seu.edu.cn).

Q. Hu is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: huqinghua@tju.edu.cn).

T. Denœux is with Université de technologie de Compiègne, CNRS, Heudiasyc, Compiègne, France, and Institut universitaire de France, Paris, France France (e-mail: thierry.denoeux@utc.fr).

through feature selection is still an open issue.

Rough set theory, proposed by Pawlak [23], has proved to be an effective tool for feature selection as a preprocessing step for pattern recognition, machine learning and data mining (see, e.g., [24]–[27] and references therein). In this paper, we propose a *rough EK-NN* rule (REK-NN for short) to deal with classification of data featuring high dimensionality in the framework of neighborhood rough set theory [24]. More precisely, a K-NN rough set model is first reformulated to partition data into positive and boundary regions in a currently selected feature space; then, to reduce misclassified samples in the boundary region, a neighborhood pignistic decision error rate is defined. This criterion evaluates the significance of inclusion of a new feature. Finally, a forward greedy search strategy is used to select a minimal feature subset with high classification performance. As will be shown soon, the REK-NN is a new rule that synchronizes feature selection and classification learning simultaneously rather than just a data preprocessing as most traditional rough set based methods do in the existing literature, and it also considers both Bayes decision error and spatial information (distances) among samples to avoid sensitive and confusing decisions when performing feature selection.

However, the REK-NN algorithm still explores the *K* nearest neighbors of each testing sample in the same way as K-NN and EK-NN. Hence, REK-NN is not feasible to classify data with a large number of observations due to runtime and memory restrictions. Fortunately, the recent MapReduce paradigm offers an ideal environment to handle this issue [28], [29]. As an improved MapReduce implementation, Apache Spark [30], [31] is one of the most flexible and powerful engines to perform faster distributed computing with big data using in-memory primitives. Spark-based K-NN rules have been proposed for big data with limited numbers of features, for example, [32]–[34]. Motivated by these contributions, we have implemented REK-NN in the Spark framework by deriving a distributed version of REK-NN, which makes it possible to search for *K* nearest neighbors in a distributed manner while relaxing the limitation on runtime and memory.

As will be shown in Section IV, both the REK-NN and the distributed REK-NN have good performances with reducing dimensionality significantly and outperforms some EK-NN classifiers as well as the traditional EK-NN taking feature selection as a data preprocessing in the majority of cases.

The rest of this paper is organized as follows. Some basic notions of the theory of belief functions and the EK-NN classifier are first briefly recalled in Section II. In Section III, the REK-NN and the distributed REK-NN are then introduced, and a theoretical analysis of their generalization error bounds is performed. In Section IV, some experiments are reported to validate the performances of the REK-NN and the distributed REK-NN classifiers using some real-world datasets. The last section concludes the paper.

II. Preliminaries

A. Dempster-Shafer theory

Given a *frame of discernment* $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$, a *mass function* is defined as a mapping from 2^Ω to $[0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{1}$$

Each number $m(A)$ denotes a degree of belief assigned to the hypothesis that "$\omega \in A$". The subsets A of Ω such that $m(A) > 0$ are called the *focal sets* of m. A mass function is said to be *Bayesian* if it all its focal sets are singletons. In this case, it is equivalent to a probability distribution. A mass function is *non-dogmatic* if Ω is a focal set; in particular, the *vacuous* mass function, verifying $m(\Omega) = 1$, corresponds to total ignorance. Finally, a mass function is *normalized* if the empty set is not a focal set; otherwise, it is said to be *unnormalized*.

There are other equivalent representations of a mass function such as the *belief* and *plausibility* functions defined, respectively, as

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \tag{2}$$

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B), \tag{3}$$

for all $A \subseteq \Omega$. $Bel(A)$ indicates the degree to which the evidence supports A, while $Pl(A)$ denotes the degrees to which the evidence is not contradictory to A. Functions Bel and Pl are linked by the relation $Pl(A) = 1 - Bel(\bar{A})$, where $\bar{A}$ is the complement of set A. They are in one-to-one correspondence with mass functions.

Let m_1 and m_2 be two mass functions. The *conjunctive combination* of m_1 and m_2 yields the unnormalized mass function

$$m_{1 \cap 2}(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \ \forall A \subseteq \Omega. \tag{4}$$

If necessary, the normality condition $m(\emptyset) = 0$ can be recovered by dividing each mass $m_{1 \cap 2}(A)$ by $1 - m_{1 \cap 2}(\emptyset)$. The resulting operation, denoted by $\oplus$, is called *Dempster's rule of combination*. It is defined by $m_{1 \oplus 2}(\emptyset) = 0$ and

$$m_{1 \oplus 2}(A) = \frac{m_{1 \cap 2}(A)}{1 - m_{1 \cap 2}(\emptyset)} \tag{5}$$

for all $A \subseteq \Omega$ such that $A \neq \emptyset$. Both rules are commutative, associative and admit the vacuous mass function as a unique neutral element.

After all pieces of evidence have been combined, the *pignistic probability distribution* associated to a mass function m can be defined by

$$BetP(\{\omega\}) = \sum_{\{A \subseteq \Omega | \omega \in A\}} \frac{m(A)}{|A|} \tag{6}$$

for all $\omega \in \Omega$.

B. EK-NN: evidential K-NN classifier

Let us consider a collection of n training samples $TR = \{(x_i, \omega(x_i)) \mid i = 1, 2, \cdots, n\}$, where $x_i \in R^p$ is a feature vector with class label $\omega(x_i) \in \Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$. Let x_t be a testing vector to be classified and $\mathcal{N}_K(x_t)$ be the set

of K nearest neighbors of x_t in TR. Each neighbor x_i with label $\omega(x_i) = \{\omega_q\}$ in $\mathcal{N}_K(x_t)$ constitutes a distinct item of evidence regarding the class membership of x_t. This item of evidence can be described using the following mass function

$$\begin{cases} m_t(\{\omega_q\}|x_i) & = \alpha \exp(-\beta_q \|x_i - x_t\|^2), \\ m_t(\Omega|x_i) & = 1 - \alpha \exp(-\beta_q \|x_i - x_t\|^2), \end{cases} \tag{7}$$

where α is a constant such that $0 < \alpha \leq 1$ and β_q ($q = 1, 2, \cdots, c$) are positive parameters associated to class $\{\omega_q\}$. Usually, $\alpha = 0.95$, and β_q are optimized [13]. In our method, we prefer fixing $\alpha = 0.95$ and $\beta_q = 1$ for all q for simplicity.

Eq. (7) means that vectors x_t and x_i are believed to belong to the same class if x_i is "close" to x_t; otherwise, x_i leaves us in a situation of almost complete ignorance concerning the class of x_t. With Dempster's rule (5), all the K items of evidence in $\mathcal{N}_K(x_t)$ can be combined as

$$m_t := \mathcal{F}(x_t) = \bigoplus_{x_i \in \mathcal{N}_K(x_t)} m_t(\cdot \mid x_i). \tag{8}$$

From the above final mass function, the lower and upper bounds for the belief of any specific hypothesis are then quantified by the belief (2) and plausibility (3) values, respectively. In the case of $0-1$ losses, the final decision on the class label of x_t can be made, alternatively, through maximizing the belief, the plausibility, or the pignistic probability [35], [36]. When maximizing pignistic probability in case of complete ignorance, i.e., $m_t(\Omega) \approx 1$, we have $BetP(\{\omega_q\}) \approx 1/c$ for all q. In this case, the class of sample x_t is unknown and it may be assigned to Ω.

III. REK-NN AND DISTRIBUTED REK-NN

In this section, we first propose a rough EK-NN classifier with attribute reduction in the framework of rough set theory, referred to as REK-NN, and then we scale up it in the Apache Spark framework in order to handle very large datasets.

Both REK-NN and its distributed version can be formulated as an optimization problem that consists in evaluating features and searching for optimal neighborhood size K^* and minimal feature subset $\mathcal{B}^*$. Formally, we want to solve the problem

$$\max_{K,\mathcal{B}} \ \mathcal{J}(K, \mathcal{B}), \tag{9}$$

where $\mathcal{J}(\cdot,\cdot)$ is an objective function to be determined.

The form of objective function in (9) as well as the feature evaluation function based on it will be discussed and defined in Section III-B after reformulating the K-NN rough set model in Section III-A. In Section III-C, the procedure of REK-NN is first presented, and the distributed REK-NN is consequently discussed in Section III-D. In Section III-E, the generalization error bound of REK-NN and the distributed one is discussed.

A. Reformulation of the K-NN rough set model

The training set can be represented as a decision table, denoted by $DT = \langle U, \mathcal{A} \cup \mathcal{D}, V \times \Omega, f \rangle$, where $U = \{x_1, x_2, \cdots, x_n\}$ is a set of samples called the *universe* or sample space; $\mathcal{A}$ is the set of conditional attributes (features); V is the value domain of attributes $\mathcal{A}$; $\mathcal{D}$ is the decision attribute (the class) and f is an information function: $f : U \times \mathcal{A} \times \mathcal{D} \to V \times \Omega$.

A *neighborhood* of sample x_i is a subset of samples close to x_i. From a concept of neighborhood, we define the *K-NN granule* as follows.

Definition 1: Let $\Delta_{\mathcal{B}}$ be a metric (e.g., the Euclidean metric) with features $\mathcal{B} \subseteq \mathcal{A}$, and let $\mathcal{N}_K^{\mathcal{B}}(x_i)$ denote the K-NN of sample x_i in U. Then, we call $\{\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i \mid x_i \in U, i = 1, 2, \cdots, n\}$ the *K-NN granules* on the universe.

The family of granules $\{\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i \mid x_i \in U\}$ forms a covering of U, as we have

1) $\forall x_i \in U, \ \ \mathcal{N}_K^{\mathcal{B}}(x_i) \neq \emptyset$;
2) $\bigcup_{i=1}^{n} (\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i) = U$.

Meanwhile, the operator $\mathcal{N}_K^{\mathcal{B}}(x_i)$ generates a binary *K*-nearest-neighborhood relation over the universe, denoted by $\mathcal{K}_{\mathcal{B}} = (\kappa_{ij})$ and defined as

$$\kappa_{ij} = \begin{cases} 1, & x_j \in \{\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i\}, \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

Definition 2: Given arbitrary subset X of the sample space and a family of K-NN granules $\{\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i \mid x_i \in U\}$, we define the *lower and upper approximations* of X with respect to relation $\mathcal{K}_{\mathcal{B}}$ as

$$\underline{\mathcal{K}_{\mathcal{B}}} X = \Big\{ x_i \in U \mid (\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i) \subseteq X \Big\},$$
$$\overline{\mathcal{K}_{\mathcal{B}}} X = \Big\{ x_i \in U \mid (\mathcal{N}_K^{\mathcal{B}}(x_i) \cup x_i) \cap X \neq \emptyset \Big\}.$$

Definition 3: Given $DT = \langle U, \mathcal{A} \cup \mathcal{D}, V \times \Omega, f \rangle$, let X_1, $X_2, \cdots, X_c$ be subsets of samples with classes $\{\omega_1\}$ to $\{\omega_c\}$. The lower and upper approximations of decision $\mathcal{D}$ with respect to attributes $\mathcal{B}$ are defined, respectively, as

$$\underline{\mathcal{K}_{\mathcal{B}}} \mathcal{D} = \bigcup_{q=1}^{c} \underline{\mathcal{K}_{\mathcal{B}}} X_q, \quad \overline{\mathcal{K}_{\mathcal{B}}} \mathcal{D} = \bigcup_{q=1}^{c} \overline{\mathcal{K}_{\mathcal{B}}} X_q, \tag{11}$$

where the lower and upper approximations $\underline{\mathcal{K}_{\mathcal{B}}} X_q$ and $\overline{\mathcal{K}_{\mathcal{B}}} X_q$ of X_q are defined according to Definition 2.

If $\underline{\mathcal{K}_{\mathcal{B}}} \mathcal{D} = \overline{\mathcal{K}_{\mathcal{B}}} \mathcal{D}$, we say that decision $\mathcal{D}$ is $\mathcal{K}_{\mathcal{B}}-$definable; otherwise, it is said to be $\mathcal{K}_{\mathcal{B}}-$rough. In the $\mathcal{K}_{\mathcal{B}}-$rough case, the difference between $\overline{\mathcal{K}_{\mathcal{B}}} \mathcal{D}$ and $\underline{\mathcal{K}_{\mathcal{B}}} \mathcal{D}$ is called the boundary of decision $\mathcal{D}$: $BN(\mathcal{D}) = \overline{\mathcal{K}_{\mathcal{B}}} \mathcal{D} \backslash \underline{\mathcal{K}_{\mathcal{B}}} \mathcal{D}$. The decision boundary is composed of the K-NN granules whose samples belong to more than one class. Hence, the samples in the boundary region are inconsistent. In contrast, the lower approximation of decision, also called positive region of decision, denoted by $POS_{\mathcal{B}}(\mathcal{D})$, is the union of granules whose samples consistently belong to one decision class. It is defined as

$$POS_{\mathcal{B}}(\mathcal{D}) = \cup_{q=1}^{c} POS_{\mathcal{B}}(\{\omega_q\}) = \cup_{q=1}^{c} \underline{\mathcal{K}_{\mathcal{B}}} X_q.$$

The *neighborhood dependency degree* (NDD) of $\mathcal{D}$ to $\mathcal{B}$ is defined as the ratio of consistent samples

$$\gamma_{\mathcal{B}}(\mathcal{D}) = \frac{|POS_{\mathcal{B}}(\mathcal{D})|}{|U|}. \tag{12}$$

The NDD defined in (12) reflects the description capability of attributes $\mathcal{B}$ to approximate decision $\mathcal{D}$. If samples are separable or consistent, the dependency degree equals one;

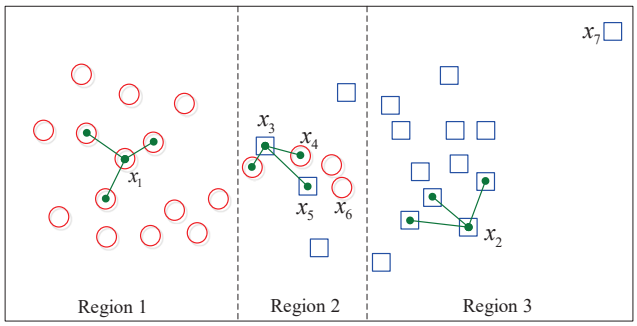

Fig. 1. Illustration of the *K*-nearest neighbor rough set model in a two-dimensional numerical feature space, circles: class $\{\omega_1\}$, squares: class $\{\omega_2\}$.

otherwise we have $\gamma_{\mathcal{B}}(\mathcal{D}) < 1$. Using the NDD, one can define the minimal feature subset $\mathcal{B}$ of the whole feature set $\mathcal{A}$ by the following two conditions:

1) $\gamma_{\mathcal{B}}(\mathcal{D}) = \gamma_{\mathcal{A}}(\mathcal{D})$, and
2) $\forall a \in \mathcal{B}, \quad \gamma_{\mathcal{B}}(\mathcal{D}) > \gamma_{\mathcal{B}-a}(\mathcal{D})$.

In the rest of this section, we present an example to illustrate the reformulated K-NN rough set model described above, and to explain why we reformulate it.

Example 1: Fig. 1 illustrates an example of binary classification in a two-dimensional numerical feature space $\mathcal{B}$, where circles and squares indicate, respectively, sets X_1 and X_2 of samples with decisions $\{\omega_1\}$ and $\{\omega_2\}$. Taking samples x_1, x_2 and x_3 as examples with $K = 3$, we have $(\mathcal{N}_3^{\mathcal{B}}(x_1) \cup x_1) \subset X_1$ and $(\mathcal{N}_3^{\mathcal{B}}(x_2) \cup x_2) \subset X_2$, while $(\mathcal{N}_3^{\mathcal{B}}(x_3) \cup x_3) \cap X_1 \neq \emptyset$ and $(\mathcal{N}_3^{\mathcal{B}}(x_3) \cup x_3) \cap X_2 \neq \emptyset$. According to above definitions, $x_1 \in \underline{\mathcal{K}_{\mathcal{B}}} X_1$, $x_2 \in \underline{\mathcal{K}_{\mathcal{B}}} X_2$ and $x_3 \in BN(\mathcal{D})$. All samples are partitioned into three regions: Regions 1 and 3 are decision positive, whereas Region 2 is the decision boundary. □

Remark 1: There are several ways to determine the neighborhood of a sample. Some authors define it by fixing a radius from the prototype sample. This approach has some nice properties [24]. One of them is monotonicity, which can guarantee the convergence of a greedy search algorithm. However, this definition cannot guarantee the existence of neighbors inside the neighborhood of the prototype sample, which is very important for the REK-NN algorithm. Furthermore, REK-NN requires the K-NN granules of a prototype sample to contain itself, while the traditional *K*-NN rough set model does not. This is why the K-NN rough set model is considered and needs to be reformulated in this paper.

B. Feature evaluation using neighborhood pignistic decision

Based on the reformulated K-NN rough set model, an evaluation function should be defined before solving problem (9) to measure the significance of the inclusion of a feature. In this section, we define such an evaluation function by first defining a neighborhood pignistic decision error rate.

As already stated in the previous section, the NDD reflects the size of the overlapping region between classes. In particular, in the strong overlap case, the neighborhood dependency degree tends to zero. In fact, the samples in the boundary region are better categorized into two groups: samples from minority classes and samples from majority classes. According to the Bayes rule, only the samples with minority classes could be misclassified. In this way, the *neighborhood decision* for sample x_i can be defined as [37]:

$$ND(x_i) = \arg \max_{1 \le q \le c} P(\{\omega_q\} \mid \mathcal{N}_K^{\mathcal{B}}(x_i)) \tag{13}$$

with neighborhood probability $P(\{\omega_q\} \mid \mathcal{N}_K^{\mathcal{B}}(x_i)) = n_{iq}/K$, where n_{iq} is the number of samples from class $\{\omega_q\}$ in $\mathcal{N}_K^{\mathcal{B}}(x_i)$. Note that, when making either the neighborhood decision or the neighborhood pignistic decision that will be introduced in this section, it is not reasonable to take into account the decision label of sample x_i when evaluating itself.

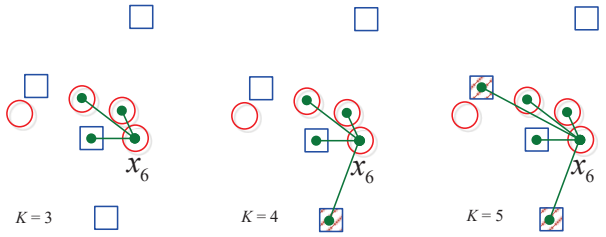

Fig. 2. Illustration of neighborhood decision in boundary with different K

In the example shown in Fig. 1, we have $P(\{\omega_1\} \mid \mathcal{N}_3^{\mathcal{B}}(x_3)) = P(\{\omega_2\} \mid \mathcal{N}_3^{\mathcal{B}}(x_4)) = P(\{\omega_1\} \mid \mathcal{N}_3^{\mathcal{B}}(x_5)) = 2/3$, which implies that $ND(x_3) = ND(x_5) = \{\omega_1\} \neq \{\omega_2\}, ND(x_4) = \{\omega_2\} \neq \{\omega_1\}$. Hence, according to the neighborhood decision, only samples x_3, x_4 and x_5 are misclassified. This example shows the ability of the neighborhood decision rule to reflect the classification complexity in complex decision boundary. However, it is still insufficient to be directly used to derive REK-NN. The reasons are presented below.

Firstly, the neighborhood decision rule (13) does not take into account spatial information among samples in the neighborhood of a prototype sample. This may lead to a confusing decision or a sensitive decision. For instance, as shown in Fig. 2, sample x_6 will be difficult to assign to a class if $K = 4$, because the number of nearest neighbors in $\mathcal{N}_K^{\mathcal{B}}(x_6)$ belonging to class $\{\omega_1\}$ and $\{\omega_2\}$ are equal. If $K = 5$, x_6 will be misclassified in class $\{\omega_2\}$. In fact, the samples (e.g., dashed squares in Fig. 2) situated far away from the prototype sample do not provide as useful information as those located nearby.

Secondly, it is important to distinguish outliers from regular samples when performing feature selection for classification. A decision rule should be able to identify outliers. As illustrated in Fig. 1, sample x_7 can be seen as an outlier because it is located far away from other samples. In this case, a good decision rule should not make precise decision on outliers and then ignore them when evaluating the significance of features.

Finally, and most importantly, it is difficult to perform feature selection and EK-NN classification simultaneously using rule (13), because this rule corresponds to the K-NN classifier rather than the EK-NN classifier.

Motivated by above statements, we define the neighborhood pignistic decision rule as follow.

Definition 4: Given decision table $DT = \langle U, \mathcal{A} \cup \mathcal{D}, V \times \Omega, f \rangle$, let m_i be the final mass function obtained by combining the evidence provided by the neighbors in $\mathcal{N}_K^{\mathcal{B}}(x_i)$. The neighborhood pignistic decision of sample x_i in feature space $\mathcal{B} \subseteq \mathcal{A}$ is defined as

$$bet_{(K,\mathcal{B})}(x_i) = \begin{cases} \Omega & \text{if } \ m_i(\Omega) \approx 1, \\ \arg\max\limits_{1 \le q \le c} BetP(\{\omega_q\} \mid \mathcal{N}_K^{\mathcal{B}}(x_i)) & \text{otherwise,} \end{cases} \tag{14}$$

where $BetP(\{\omega_q\} \mid \mathcal{N}_K^{\mathcal{B}}(x_i))$ is the pignistic probability (6) associated to the final combination m_i. In practice, when implementing (14), we set $m_i(\Omega) > \eta = 0.9$ instead of $m_i(\Omega) \approx 1$. As will be seen, $\eta = 0.9$ can achieve appropriate performance in our experiments.

In Definition 4, each item of evidence provided by each neighbor in $\mathcal{N}_K^{\mathcal{B}}(x_i)$ can be directly established according to (7) and then be pooled to get a final mass function according to (8). When $bet_{(K,\mathcal{B})}(x_i) = \omega(x_i) = \{\omega_q\}$, we say that sample x_i can be correctly recognized with zero loss even if $\mathcal{N}_K^{\mathcal{B}}(x_i) \nsubseteq X_q$, the subset of samples with decision class $\{\omega_q\}$. If $bet_{(K,\mathcal{B})}(x_i) \neq \omega(x_i)$ or $bet_{(K,\mathcal{B})}(x_i) = \Omega$, sample x_i will be misclassified with unit loss. Define a $0-1$ loss function $\lambda(\cdot \mid \cdot)$ such that $\lambda(\omega(x_i) \mid bet_{(K,\mathcal{B})}(x_i)) = 0$ if $bet_{(K,\mathcal{B})}(x_i) = \omega(x_i)$ and $\lambda(\omega(x_i) \mid bet_{(K,\mathcal{B})}(x_i)) = 1$ otherwise. We can now define the following *neighborhood pignistic decision error rate*:

$$\mathcal{L}_{(K,\mathcal{B})} = \frac{1}{n} \sum\nolimits_{i=1}^{n} \lambda(\omega(x_i) \mid bet_{(K,\mathcal{B})}(x_i)). \tag{15}$$

The neighborhood pignistic decision error rate depends on the size and complexity of overlap among classes, and therefore it is related to the NDD. We have the following proposition.

Proposition 1: Given decision table $DT = \langle U, \mathcal{A} \cup \mathcal{D}, V \times \Omega, f \rangle$, the neighborhood pignistic decision error rate satisfies:

1) $\mathcal{L}_{(K,\mathcal{B})} = \frac{1}{n} \sum_{x_k \in U - POS_{\mathcal{B}}(\Omega)} \lambda(\omega(x_k) \mid bet_{(K,\mathcal{B})}(x_k))$;
2) $\mathcal{L}_{(K,\mathcal{B})} \le 1 - \gamma_{\mathcal{B}}(\mathcal{D})$.

Proof: The whole proof consists of following two steps:

1) For any $x_k \in POS_{\mathcal{B}}(\mathcal{D})$, we have zero loss, i.e., $\lambda(\omega(x_k) \mid bet_{(K,\mathcal{B})}(x_k)) = 0$. Hence, only samples x_k in $U - POS_{\mathcal{B}}(\mathcal{D})$ may be misclassified. Consequently, $\mathcal{L}_{(K,\mathcal{B})} = \frac{1}{n} \sum_{x_k \in U - POS_{\mathcal{B}}(\mathcal{D})} \lambda(\omega(x_k) \mid bet_{(K,\mathcal{B})}(x_k))$.
2) Let us dividing the universe U into subsets X_1, X_2, $\cdots$, and X_c according to decision $\mathcal{D}$. For all x_k in $POS_{\mathcal{B}}(\mathcal{D})$, there is some X_q such that $(\mathcal{N}_K^{\mathcal{B}}(x_k) \cup x_k) \subseteq X_q$. Hence, we have $BetP(\{\omega_q\} \mid \mathcal{N}_K^{\mathcal{B}}(x_k)) = 1$ and $bet_{(K,\mathcal{B})}(x_k) = \omega(x_k) = \{\omega_q\}$, which implies that all the samples in decision positive region have zero decision error. Therefore, $\gamma_{\mathcal{B}}(\mathcal{D})$ is not greater than $1 - \mathcal{L}_{(K,\mathcal{B})}$, i.e., $\mathcal{L}_{(K,\mathcal{B})} \le 1 - \gamma_{\mathcal{B}}(\mathcal{D})$.

■

Proposition 1 indicates that, on the one hand, the decision positive region can simplify the computation of the neighborhood pignistic decision error rate (as well as the evaluation of a new testing sample as will be remarked later). On the other hand, the neighborhood pignistic decision error rate is not greater than the loss induced by the NDD (i.e., $1-\gamma_{\mathcal{B}}(\mathcal{D})$). Furthermore, $\mathcal{L}_{(K,\mathcal{B})}$ can be viewed as an estimation of the Bayes decision error with consideration of spatial information among samples. Hence, it is suitable for feature selection by minimizing $\mathcal{L}_{(K,\mathcal{B})}$ or maximizing $1-\mathcal{L}_{(K,\mathcal{B})}$. In the consistent case, i.e., when $POS_{\mathcal{B}}(\mathcal{D}) = U$, zero error rate can be achieved, i.e., $\gamma_{\mathcal{B}}(\mathcal{D}) = 1$ and $\mathcal{L}_{(K,\mathcal{B})} = 0$.

Using neighborhood pignistic decision error rate, the objective function $\mathcal{J}(K, \mathcal{B})$ in (9) for the REK-NN classifier can be defined as

$$\mathcal{J}(K, \mathcal{B}) := 1 - \mathcal{L}_{(K,\mathcal{B})}. \tag{16}$$

Traditionally, the significance of a feature a relative to feature subset $\mathcal{B}$ can be defined as a feature evaluation function using the NDD, i.e., $SIG(a, \mathcal{B}, \mathcal{D}) = \gamma_{\mathcal{B} \cup a}(\mathcal{D}) - \gamma_{\mathcal{B}}(\mathcal{D})$. However, as remarked above, $SIG(a, \mathcal{B}, \mathcal{D})$ cannot reflect classification complexity well in the decision boundary. In contrast, the neighborhood pignistic decision error rate is better with consistent to classification complexity. Hence, we define the significance of a feature a relative to $\mathcal{B}$ according to

$$SIG(a, \mathcal{B}, K) = \mathcal{J}(K, \mathcal{B} \cup a) - \mathcal{J}(K, \mathcal{B}). \tag{17}$$

The feature evaluation function (17) indicates that significance increases when adding an informative feature. A minimal feature subset is achieved if SIG cannot be improved. Here, the $SIG(a, \mathcal{B}, K)$ cannot be guaranteed to be monotonous with respect to the order of selected features. Therefore, the explored minimal feature subset according to (17) may be suboptimal. Fortunately, it will be shown that good performance can be achieved with the selected minimal feature subset.

C. REK-NN classification procedure and time complexity

To solve problem (9), we still need a search strategy. There exist a number of candidate search strategies to find a minimal feature subset, e.g., the greedy search strategy such as sequentially forward selection (SFS), sequentially backward selection (SBS) [24], branch-and-bound search strategy [38], and genetic algorithm-based feature selection [39]. In this paper, we mainly focus on synchronizing EK-NN classification with feature reduction for high dimensional data rather than exploring an efficient search strategy for selecting features. For the sake of simplicity, the SFS procedure is adopted. As will be seen, good performance can be achieved using SFS.

Based on SFS, the REK-NN classifier can be realized by Algorithm 1. As can be seen, REK-NN consists of two parts: learning and evaluation. When feature subset $\mathcal{B}^*$ has been selected, the computational time in learning part is

$$O\Big(\sum_{j=0}^{|\mathcal{B}^*|} \{\frac{n(n+1)}{2}(p-j) + nK(c+1) + (p-j)\}\Big),$$

in which the first term is time complexity used to explore the K nearest neighbors, while the second and third terms are times consumed by pooling evidence of the K nearest neighbors and sorting SIG, respectively. To evaluate samples x_t, the computational time is

$$O\Big(nn_t|\mathcal{B}^*| + (n_t - n_{POS})K^*(c+1)\Big),$$

Algorithm 1: REK-NN classifier

Input: $DT = \langle U, \mathcal{A} \cup \mathcal{D} \rangle$, termination ϵ, bound $[\underline{K}, \overline{K}]$ for K and testing samples $x_t, t = 1, 2, \cdots, n_t$.

Output: Optimal K^*, minimal feature subset $\mathcal{B}^*$ ($|\mathcal{B}^*| \ll p$) and estimations $\hat{\omega}(x_t)$ of x_t.

```
% Learning part
K* = K̲, B* ← ∅, J* = 0
for K = K̲ to K̄ do
    B ← ∅,
    while A − B ≠ ∅ do
        for each a_i ∈ A − B do
            Determine positive region POS_{B∪a_i}(D)
            Compute
              SIG(a_i, B, K) = J(K, B ∪ a_i) − J(K, B)
        Select the feature a_k such that
              SIG(a_k, B, K) = max_i{SIG(a_i, B, K)}
        if SIG(a_k, B, K) > ε then
            B ← B ∪ a_k
        else
            break
    if J(K, B) > J* then
        B* ← B, K* ← K, J* ← J(K, B)
% Evaluation part
Find the K* nearest neighbors for each x_t in space B*
if N^{B*}_{K*}(x_t) ⊆ POS_{B*}({ω_q}) then
    ω̂(x_t) = {ω_q}
else
    Derive evidence of K*-NN in N^{B*}_{K*}(x_t) using (7)
    Combine K* items of evidence using rule (8)
    Estimate ω̂(x_t) from the final combination using (14)
```

where n_{POS} is the number of testing samples x_t such that $\mathcal{N}^{\mathcal{B}^*}_{K^*}(x_t) \subseteq POS_{\mathcal{B}^*}(\mathcal{D})$.

Obviously, the learning part consumes more computational time than the evaluation. Fortunately, the REK-NN can be learnt off-line and the evaluation is more computationally efficient than EK-NN because $|\mathcal{B}^*| \ll p$.

Remark 2: Algorithm 1 shows that the REK-NN goes beyond traditional feature selection methods based on rough set theory: it is not just a data preprocessing method such as, e.g., the method described in [40], but it synchronizes feature selection and Leave-One-Out (LOO) classification learning in a single procedure. Besides, REK-NN can also perform sample selection that can be used to simplify evaluation. More precisely, if all K nearest neighbors of a testing sample are located in a decision positive region with class $\{\omega_q\}$, such testing sample can be assigned directly to class $\{\omega_q\}$ without pooling evidence of its K nearest neighbors. Finally, as a mass function can be viewed as a generalized random set, a probabilistic or fuzzy partition can be obtained from a credal partition [41], [42]. With this viewpoint, a probabilistic or fuzzy K-NN classification can be achieved in a similar way from the credal classification by the REK-NN.

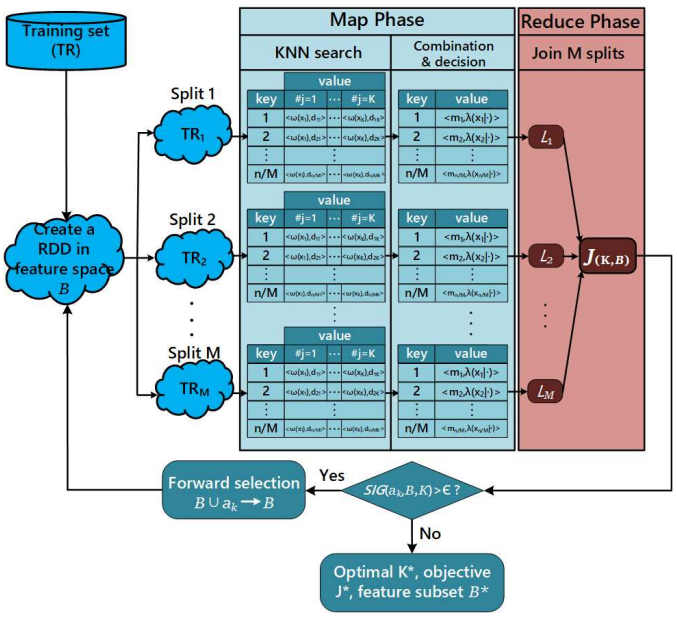

Fig. 3. The workflow of distributed REK-NN (learning part) in Apache Spark

D. Scaling up REK-NN in the Apache Spark

As discussed above, the greatest part of the runtime consumed by REK-NN is spent searching for the K nearest neighbors of each sample, i.e., $O(\sum_{j=0}^{|\mathcal{B}^*|} \frac{n(n+1)}{2}(p-j))$. When processing a dataset with large sample size n, the runtime and memory requirements become excessively demanding. To alleviate the bottleneck of computation and storage, we now present a distributed REK-NN method implemented in Apache Spark [31] in this section.

Spark parallelizes the calculation transparently through a distributed data structure, called *Resilient Distributed Datasets* (RDD) [31]. RDD allows data structures stored in main memory to persist and be reused. The workflow of the distributed REK-NN in Spark is illustrated in Fig.3. By comparing Fig. 3 and Algorithm 1, we can see that the main difference between the distributed REK-NN and REK-NN lies in the search for the K nearest neighbors for each sample. More specifically, after creating a RDD object in feature space $\mathcal{B}$, the distributed REK-NN performs the following two phases:

- *Map phase*: all n samples in RDD are partitioned into M splits, $TR_1, TR_2, \cdots, TR_M$, with approximately the same number n/M of samples; the exploration of K nearest neighbors, evidence combination and decision making are performed in each split in a distributed manner;
- *Reduce phase*: all losses $\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_m$ are joined from all splits to calculate the final objective function $\mathcal{J}(K, \mathcal{B})$.

As can be seen from Fig. 3, the distributed REK-NN procedure applies a divide-and-conquer approach, where each local split/map does not know the samples in the other splits. From this viewpoint, the distributed REK-NN is an approximate nearest neighbor technique with parallel computation when learning the classifier. The computational time in the learning

part for the distributed algorithm is now decreased to

$$O\Big(\sum_{j=0}^{|\mathcal{B}^*|}\{\frac{n(n+M)}{2M^2}(p-j)+nK(c+1)+(p-j)\}\Big),$$

which suggests that much runtime can be reduced once more splits have been partitioned.

In contrast to learning, the distributed REK-NN performs evaluation similarly as the REK-NN: it first explores the exact K^* nearest neighbors for each testing sample in each split, and then pools the evidence of the final K^* nearest ones selected from the MK^* nearest neighbors that have been explored.

E. Generalization error bound of the (distributed) REK-NN

In this section, we determine the *generalization error bound* (GEB for short) of the REK-NN and the distributed REK-NN classifiers, as similarly do in [43].

Suppose $p(x)$ is the density function of the input variable x, and $\mathcal{F}_{(K,\mathcal{B})}(x)$ is an approximation of the truth $\mathcal{F}(x)$ in operation (8). The expected risk or loss of the approximation $\mathcal{F}_{(K,\mathcal{B})}(x)$ with parameters $(K,\mathcal{B})$ can be defined as

$$R(\mathcal{F})=\int \lambda\Big(\omega(x)\mid bet_{(K,\mathcal{B})}(x)\Big)p(x)dx. \tag{18}$$

Because the density $p(x)$ is unknown in practice, the empirical risk corresponding to (18) can be calculated as

$$\hat{R}(\mathcal{F})=\frac{1}{n}\sum_{i=1}^{n}\lambda\Big(\omega(x)\mid bet_{(K,\mathcal{B})}(x)\Big). \tag{19}$$

Minimizing the empirical risk means finding $(K^*,\mathcal{B}^*)$ such that $\mathcal{F}_{(K^*,\mathcal{B}^*)}=\arg\min_{\mathcal{F}}\hat{R}(\mathcal{F})$.

In both the REK-NN and the distributed REK-NN methods, we are interested in finding an upper bound of the expected risk when selecting an arbitrary pair $(K,\mathcal{B})$. We have the following proposition.

Proposition 2: The expected risk of the REK-NN and its distributed variant satisfies the following inequality with probability $1-\frac{\delta}{2}$:

$$R(\mathcal{F})\le\mathcal{L}_{(K^*,\mathcal{B}^*)}+\frac{t_{\frac{\delta}{2}}}{\sqrt{n-1}}\sqrt{\mathcal{L}_{(K^*,\mathcal{B}^*)}-\mathcal{L}^2_{(K^*,\mathcal{B}^*)}}, \tag{20}$$

where δ is the level of significance, $t_{\frac{\delta}{2}}$ is the critical value of Student distribution $\mathcal{T}$, and $\mathcal{L}_{(K^*,\mathcal{B}^*)}$ is the minimal total loss by maximizing problem (9) based on training set.

Proof: Denote $Y=\lambda(\omega(x)\mid bet_{(K,\mathcal{B})}(x))$. Hence, $Y_i=\lambda(\omega(x_i)\mid bet_{(K,\mathcal{B})}(x_i))$ can be viewed as a sample drawn from the random variable Y. According to (19), we have $\hat{R}(\mathcal{F})=\frac{1}{n}\sum_{i=1}^{n}Y_i=\bar{Y}$. From the Central Limit Theorem, the empirical risk follows asymptotically a normal distribution when $n\to\infty$, i.e., $\hat{R}(\mathcal{F})=\bar{Y}\sim\mathcal{N}(R(\mathcal{F}),\frac{\sigma^2}{n})$, which results in

$$\frac{\hat{R}(\mathcal{F})-R(\mathcal{F})}{\sigma/\sqrt{n}}\sim\mathcal{N}(0,1).$$

Hence, with a critical value $\mu_{\frac{\delta}{2}}$, we have $p(R(\mathcal{F})\le\hat{R}(\mathcal{F})+\mu_{\frac{\delta}{2}}\frac{\sigma}{\sqrt{n}})=1-\frac{\delta}{2}$. Namely, the following inequality holds with probability $1-\frac{\delta}{2}$:

$$R(\mathcal{F})\le\hat{R}(\mathcal{F})+\mu_{\frac{\delta}{2}}\frac{\sigma}{\sqrt{n}}.$$

As the true variance σ is unknown, we replace it by the sample standard derivation $S=\sqrt{\frac{1}{n-1}\sum_{n=1}^{n}(Y_i-\bar{Y})^2}$ and we get

$$\frac{\hat{R}(\mathcal{F})-R(\mathcal{F})}{S/\sqrt{n}}\sim\mathcal{T}(n-1),$$

where $\mathcal{T}(n-1)$ denotes the Student distribution with $n-1$ degrees of freedom. Using (19), we get

$$\begin{aligned}R(\mathcal{F})&\le\hat{R}(\mathcal{F})+t_{\frac{\delta}{2}}\frac{S}{\sqrt{n}}\\&\le\hat{R}(\mathcal{F})+\frac{t_{\frac{\delta}{2}}}{\sqrt{n-1}}\times\\&\quad\sqrt{\frac{1}{n}\sum_{i=1}^{n}\lambda^2(\omega(x_i)\mid bet_{(K,\mathcal{B})}(x_i))-\hat{R}^2(\mathcal{F})}.\end{aligned} \tag{21}$$

In the case of 0-1 losses, $\frac{1}{n}\sum_{i=1}^{n}\lambda^2(\omega(x)\mid bet_{(K,\mathcal{B})}(x))=\mathcal{L}_{(K,\mathcal{B})}$. When finding the minimum empirical risk based on training, we have $\hat{R}(\mathcal{F})=\mathcal{L}_{(K^*,\mathcal{B}^*)}$. In this way, (21) can be transformed into (20), which completes the proof. ■

Remark 3: Proposition 2 states that the generalization error bound is a function of training loss and sample size with a certain probability. It will become small if one can obtain sufficiently enough samples and/or a small training loss. In fact, the error bound in (20) is derived in a general way from the probabilistic point of view, and thus it could be adequate for any classifiers rather than just the (distributed) REK-NN.

IV. Experimental results

In this section, some numerical experiments are reported to validate the (distributed) REK-NN by comparing with some other well-known methods based on some real-world datasets, as described in Table I. These datasets were selected from the UCI Machine Learning Repository [44], the Keng Ridge Biomedical Data set Repository [45], and the KEEL dataset repository [46].

All numerical attributes of samples in Table I were normalized into the interval $[0,1]$. The server used to implement the distributed REK-NN had the following configurations and software set-ups:

- Processor: Intel(R) Xeon(R) CPU E5-2630 v3 @2.4GHz;
- Cores: 12 cores (24 threads);
- RAM: 24 GB;
- Network: Gigabit Ethernet (1Gbps);
- Cache: 15MB;
- Operative System: Windows Server 2016;
- Apache Spark version: 2.4.3;
- Scala version: 2.11.8.

A. An illustrative example

In this section, we use the *seeds* dataset to intuitively illustrate the results of REK-NN classifier, including feature selection, sample selection, global treatment of imperfect knowledge in labels and an insight of classification.

REK-NN was implemented on the seeds dataset with increasing K from 3 to 40. For a given K, there are two stopping criteria for the REK-NN. The search stops if all candidate

TABLE I
DATASET DESCRIPTION

	Data	Samples	# Attributes	# classes
1	Seeds	210	7	3
2	Wine	178	13	3
3	Wdbc	569	30	2
4	Wpbc	198	33	2
5	Iono	351	34	2
6	Soybean	47	35	4
7	Sonar	208	60	2
8	LSVT	126	309	2
9	DLBCL	77	5469	2
10	Leukemia	72	11225	3
11	MLL	72	12582	3
12	Prostate	136	12600	2
13	Tumors	327	12558	7
14	APS	60000	171	2
15	Covtype	581012	54	7
16	Kddcup	494020	41	2
17	Poker	1025010	10	10

features have been selected or the inclusion of any new feature into the current feature subset does not improve the SIG. In either case, we set $\epsilon = 0$, which will be assumed in what follows unless when otherwise specified.

Fig. 4 shows that the contour surface of objective function $\mathcal{J}(K, \mathcal{B})$ increases rapidly with the order of selected features and achieves appropriate performance in most cases when selecting two or three features. The global best performance $\mathcal{J}(K, \mathcal{B}) = 0.9619$ is achieved with $K = 8$ and $\mathcal{B} = \{2, 7\}$ (i.e., the 2nd and 7th features). Correspondingly, the contour surface of classification accuracy using REK-NN is shown in Fig. 5, from which we can see that the REK-NN gives some samples a class membership with uncertainty rather than a crisp one. According to the maximum pignistic probability rule, all samples have been partitioned into two groups: the misclassified and correctly classified samples. Some of these correctly classified samples belong to decision positive regions. These decision positive regions are shown by closed bold lines.

Finally, we can see from Fig. 5 that there are eight misclassified samples, outlined by plus signs with numbers. To have an insight of these misclassified samples, we plotted the distribution of eight nearest neighbors for each misclassified sample in Fig. 6. We can see that the samples 20, 62, 198 and 202 are misclassified, while the samples 9, 24, 125 and 136 can be viewed as noisy samples because their eight nearest neighbors belong to another one class.

For the misclassified sample 202 in the bottom right subplot, we can see that there are four neighbors in each of two classes. In this case, the four nearest neighbors belonging to class $\{\omega_3\}$ play a less important role because they are located far away from sample 202. Note that this does not mean that

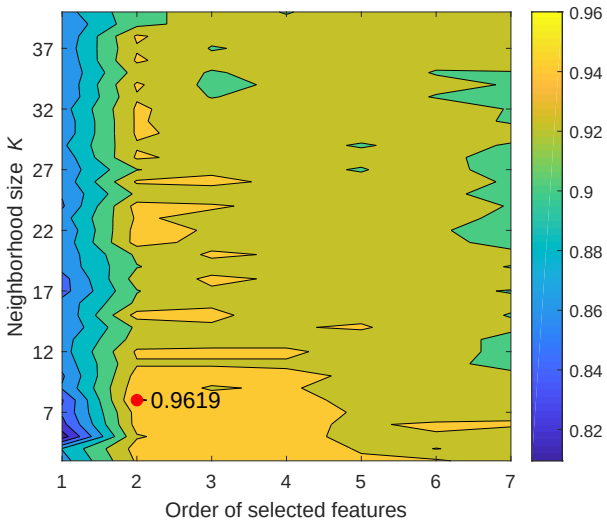

Fig. 4. Contour surface of the objective function $\mathcal{J}_{(K,\mathcal{B})}$ for the seeds dataset.

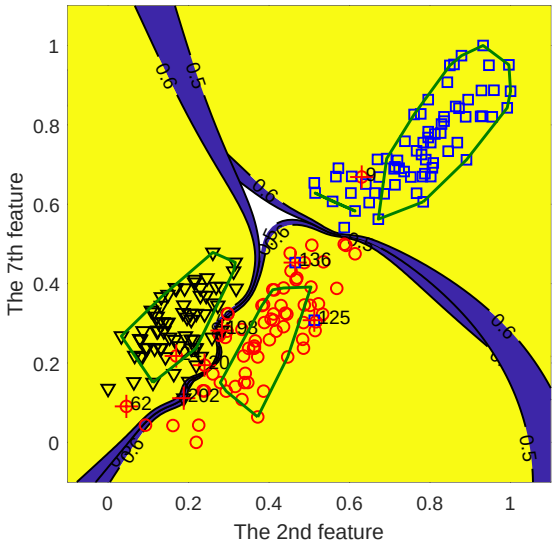

Fig. 5. Contour surface of class membership of REK-NN for seeds dataset where circles: class $\{\omega_1\}$, squares: class $\{\omega_2\}$, triangles: class $\{\omega_3\}$, bold line: boundary of decision positive regions, plus signs: misclassified samples.

this misclassified sample can be correctly classified using the neighborhood decision rule, because it is "left out" when evaluating itself. Hence, it is difficult for the neighborhood decision rule to make a decision in this confusing case.

B. Performance evaluation

a) Performance of feature selection on the entire dataset: This case study aims to show the selected features as well as the generalization error bound with REK-NN on each entire dataset. For each $K \in [3, 40]$, the feature subset was selected according to the search strategy such that the inclusion of any new feature into the current subset does not improve the $SIG(a, \mathcal{B}, K)$. Then, we obtained 38 values of the objective function $\mathcal{J}$ for each dataset and the largest objective value indicates the best performance. Note that, $\mathcal{J}$ is in fact the

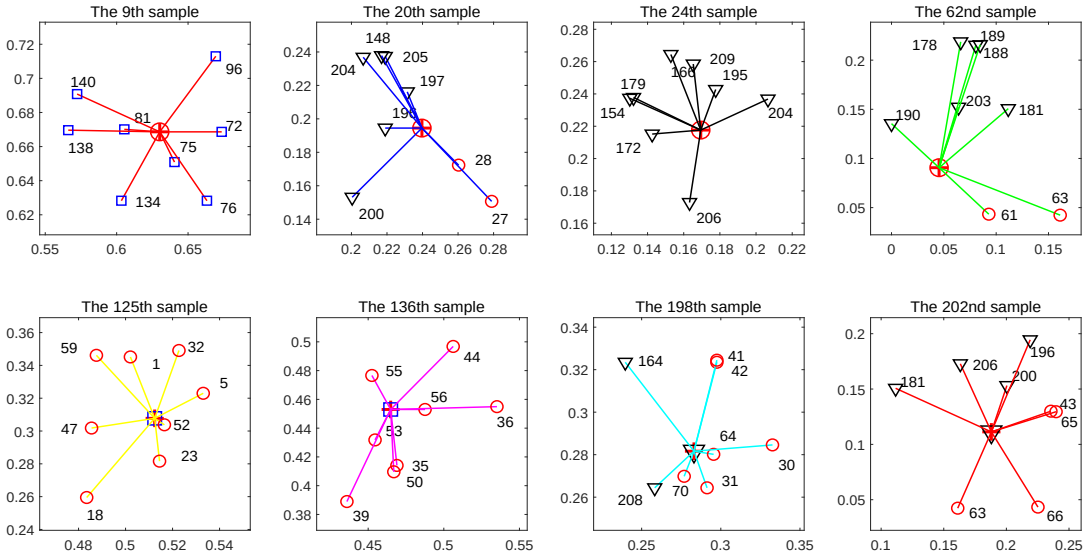

Fig. 6. Distributions of eight nearest neighbors of misclassified samples for seeds dataset. The meanings of all symbols are the same as in Fig. 5.

TABLE II
OPTIMAL K, $\mathcal{J}$, MINIMAL FEATURES SUBSETS AND GENERALIZATION ERROR BOUND WITH $\delta = 0.1$ FOR THE REK-NN ON THE WHOLE DATASETS.

Data	K^*	Selected features $\mathcal{B}^*$	$\mathcal{J}^*$	GEB
Wine	34	7, 1, 3, 4, 10, 13, 11	0.9944	0.0149
Wdbc	23	24, 25, 21, 1, 2, 27, 3, 23	0.9789	0.0310
Wpbc	23	13, 2, 14, 29, 10, 19	0.8384	0.2049
Iono	3	6, 5, 16	0.9373	0.0841
Soybean	3	21, 22	1.00	0
Sonar	8	11, 18, 37, 27, 49, 51, 45, 3, 2	0.9183	0.1132
LSVT	18	126, 1, 86, 153, 112, 91, 19	0.9206	0.1195
DLBCL	3	409, 2840, 773	1.00	0
Leukemia	3	1939, 2903, 59, 3912	1.00	0
MLL	3	2592, 3712, 7081, 693	1.00	0
Prostate	3	8965, 8306, 11858, 8636, 10234	0.9853	0.0319
Tumors	6	8642,11368,3264,9833,2721,5909, 3324,10402,4497,9920,295,7087, 5737,1925,1235,8625,10958	0.9541	0.0650

approximate accuracy of REK-NN based on LOO learning. The optimal K^*, selected feature subsets and generalization error bound for each dataset are shown in Table II. The values of the objective function $\mathcal{J}$ for various number of (partial) selected features are shown in Fig. 7 for the optimal K^*.

From Table II and Fig. 7, we can see that the approximate accuracy $\mathcal{J}$ increases rapidly in most cases when several features have been included into the feature subset. Note that the selected feature subsets are minimal and optimal in most cases. In some cases, such as with the LSVT and Sonar datasets, the selected feature subsets are minimal but suboptimal because much higher approximate accuracies can be obtained with the inclusion of much more new features into the feature subset, as indicated by the curves of objective $\mathcal{J}$ of these two datasets shown in Fig 7.

More interestingly, zero bounds of generalization error have been achieved for the Soybean, DLBCL, Leukemia and MLL. In other words, all samples can be correctly classified by the REK-NN in this case. Nevertheless, it will be seen that training losses will be inevitable when applying other validation strategy such as ten-fold cross validation.

b) Comparing REK-NN to EK-NN with feature selection: We compared REK-NN with EK-NN by considering feature selection as a data preprocessing. Namely, we first selected features for the EK-NN, and then we implemented EK-NN in the selected feature space for each dataset. Here, the neighborhood rough set (NRS) model [24] was used to select feature subsets with the following three feature evaluation methods:

1) Neighborhood dependency degree method (NDD) [24];
2) Neighborhood mutual information method (NMI) [47];
3) Fuzzy information entropy method (FINEN) [48].

Note that these NRS model-based feature selection methods are considered mainly for the sake of fairness, because they are based on similar principles as REK-NN. Comparisons between NRS model-based feature selection methods and some other popular feature selection methods can be found, for example, in [24], [37].

To implement the above three NRS model-based feature evaluation methods, the neighborhood size, i.e., the radius from the prototype sample to its neighbors, should be predetermined. As suggested in [24], we varied the neighborhood size from 0.02 to 0.4 with step 0.02 in order to get different feature subsets, and then we evaluated the selected features according to the performances of EK-NN. To terminate NRS-NDD and NRS-NMI, a termination threshold as ϵ in Algorithm 1 was preset to 0.001. To implement EK-NN, we selected $\alpha = 0.95$

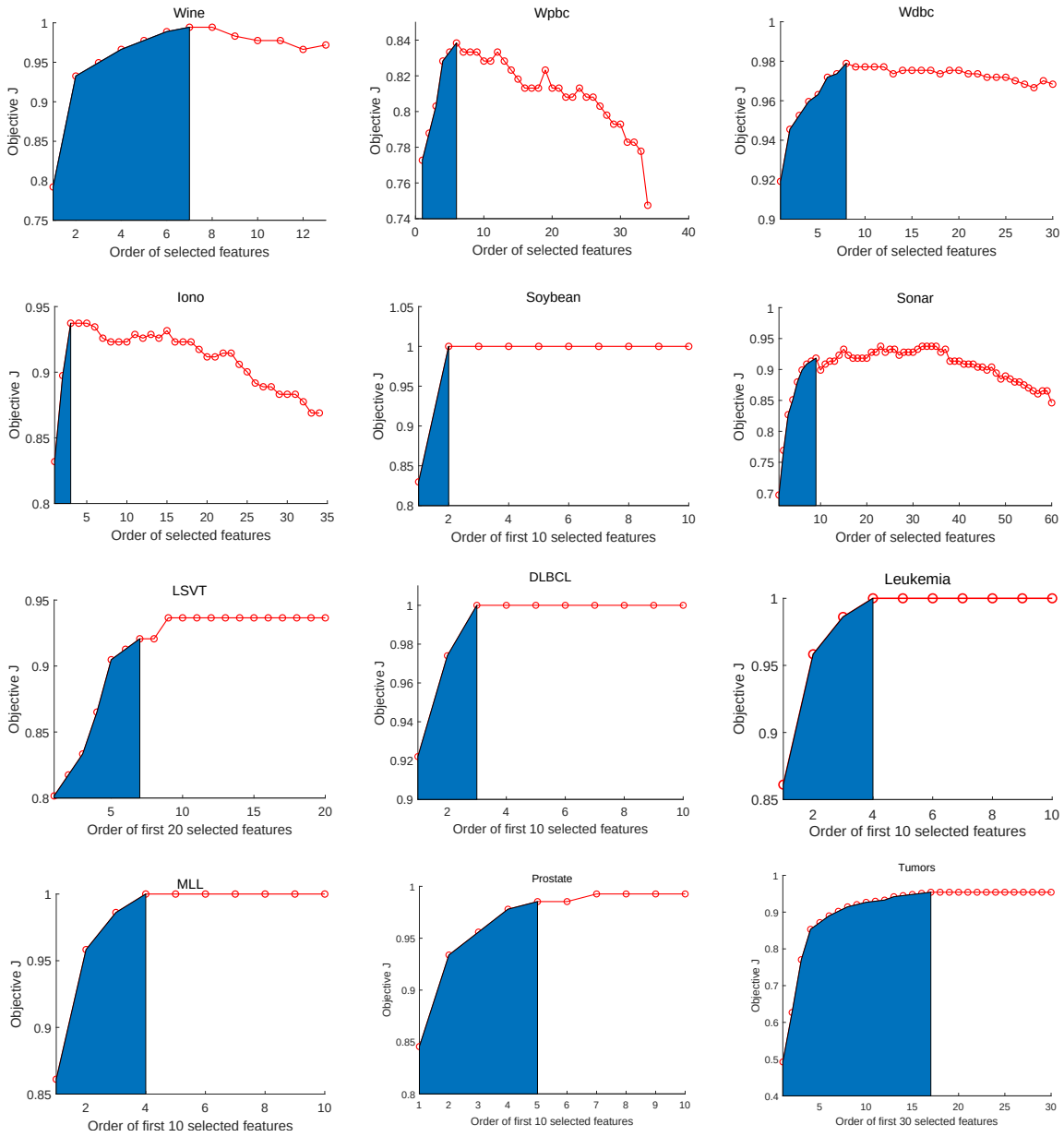

Fig. 7. Objectives $\mathcal{J}_{(K,\mathcal{B})}$ vs. the order of (partial) selected features for each dataset in the case of optimal K^* as shown in Table II. The shaded areas are used to indicate the numbers of selected features as well as the searching step after which the REK-NN will be terminated.

and we initialized β_q to the inverse of the mean distances among all training samples with decision class $\{\omega_q\}$. The value of K for the EK-NN method was the same as the value found by REK-NN for each dataset.

The experimental comparison was conducted using ten-fold cross-validation. For each dataset ,we randomly divided the samples in the selected feature space (in Table II or III) into ten subsets, and used nine of them as training set and the rest one as the test set. After ten rounds, we computed the average value as the final performance for each method. The ten-fold cross-validation was repeated ten times for each dataset.

To gain insight into the selected features based on NRS with NDD, NMI and FIENE, we listed the selected feature subsets for all datasets in Table III for EK-NN. The selected feature subsets in Table III, together with those in Table II, show that, on the one hand, the selected feature subsets using NRS-NDD, NRS-NMI and NRS-FINEN are different from those obtained by the REK-NN; on the other hand, REK-NN selected fewer features in the majority of cases. In particular, the significance of each feature evaluated by REK-NN is also different from NRS-NDD, NRS-NMI and NRS-FINEN, as indicated by the order of selected features for each dataset. The main reason is that the neighborhood pignistic decision error rate reflects well the classification complexity and considers spatial information among samples.

Table IV presents the performances of EK-NN in the feature space selected by NRS with NDD, NMI and FINEN. We can conclude from Table IV that the performances of EK-NN can be enhanced through feature selection as a preprocessing step. However, it was still outperformed by the REK-NN

TABLE III
SELECTED FEATURES FOR EK-NN BASED ON NEIGHBORHOOD ROUGH SET MODEL WITH NDD, NMI AND FINEN METHODS

Data	NRS-NDD	NRS-NMI	NRS-FINEN
Wine	13,10, 11, 1, 7, 5, 2, 12, 3	7, 10, 13, 11, 1	7,1,10,13, 5, 2, 12, 8,11, 3, 9, 4, 6
Wdbc	23, 28, 12, 22, 25, 19, 21, 10, 9, 7, 30, 2, 16, 18, 29, 15, 1, 3, 4, 5	28, 21, 22, 23, 8, 29, 11, 5, 16, 27, 12, 24, 3, 13, 30, 19, 25, 2, 10, 26, 9, 18,4	23, 28, 22, 11, 21, 7, 25, 12, 9, 26, 19, 2, 27, 30, 8, 5, 16, 29, 10, 15
Wpbc	2, 13, 33, 4	2, 33, 7, 9	2, 13, 29, 4
Iono	1, 5, 13, 18, 34, 24, 3, 16	5, 6, 8, 9, 3	5, 6, 8, 28, 33, 24, 25, 10, 20, 21, 3, 17, 34, 12, 30, 23, 22, 19, 15, 31, 1, 32, 29
Soybean	22, 4	22, 4	22, 21
Sonar	1, 17, 11, 37, 31, 23, 34, 26, 12, 22, 2	11, 17, 37, 48, 27, 22, 24, 40, 1	12, 27, 21, 37, 32, 30, 54, 15, 24, 39, 22, 34, 11, 57, 16, 10, 46, 6, 36, 48, 33
LSVT	85, 82, 84, 53	80, 85, 84, 42, 53, 113, 96, 108, 246, 19	86,153,87,4,84,42,80,94,53,93,68,100, 110,113,103,65,107,122,91,76,108
DLBCL	3127, 5452, 10	3127, 3988, 3942, 59	3127, 5452, 1259, 534
Leukemia	10038, 1285, 5555, 10712, 6998, 731, 2	2833, 6720, 6322, 4583, 4223	11071,9682,788, 2295,6839,3839,9192
MLL	11297, 6565, 12026, 11, 11234, 318	3634, 11643, 5265, 12391, 3249	12418, 7347, 2776, 10274, 7106, 1228
Prostate	5920, 1792, 6181, 9850, 2358, 1196, 7121, 11640, 6390, 55	6185, 12067, 8458, 11529, 6615, 6367, 9850, 5314, 6390, 9626, 2580, 2358, 4855, 7121, 8073, 10968, 2862, 2896, 4761, 10143, 3415, 2799	6185, 6615, 9850, 1087,11529,6367,316, 2580,3542,6261,7121,8073,231,2576, 11372,11789,3419,2823,207,9058,5032, 6993, 9068,6390,363,6359,351,5648,749, 12021,7961,2799,2291,55,2097,2162, 3651, 10096,8594,10745,7705,5314,1819, 11776,6493,1466,7843,6275,327,45,4483
Tumors	6320, 7648, 5811, 8904, 11169, 6149, 3264,12148,5718,12319,6639,6547,18	5411, 6320,7648, 3264,3324, 7121, 12319, 9668, 2450, 5234, 7252, 364	2543, 3264, 7648, 6320,5411,6079,6671, 2943, 10126, 12101, 4178, 8337, 9046, 10540, 6311, 9220, 8748, 6506, 8281

TABLE IV
AVERAGE ACCURACIES WITH 95% CONFIDENCE INTERVALS BASED ON NRS MODEL WITH NDD, NMI AND FINEN

Data	EK-NN	NRS-NDD based EK-NN	NRS-NMI based EK-NN	NRS-FINEN based EK-NN	REK-NN
Wine	0.9794 ± 0.0078	0.9814 ± 0.0045	0.9843 ± 0.0073	0.9793 ± 0.0106	$\underline{0.9896 \pm 0.0064}$
Wdbc	0.9655 ± 0.0041	0.9680 ± 0.0026	0.9682 ± 0.0056	0.9645 ± 0.0048	$\underline{0.9721 \pm 0.0043}$
Wpbc	0.7409 ± 0.0180	0.7912 ± 0.0106	0.7794 ± 0.0193	0.7789 ± 0.0129	$\underline{0.8193 \pm 0.0083}$
Iono	0.8954 ± 0.0079	0.9211 ± 0.0058	0.9182 ± 0.0045	0.9059 ± 0.0102	0.9331 ± 0.0079
Soybean	0.9967 ± 0.0128	$\underline{1.0000 \pm 0.0000}$	$\underline{1.0000 \pm 0.0000}$	$\underline{1.0000 \pm 0.0000}$	$\underline{1.0000 \pm 0.0000}$
Sonar	0.8024 ± 0.0253	0.8207 ± 0.0215	0.8396 ± 0.0231	0.8247 ± 0.0241	$\underline{0.8988 \pm 0.0185}$
LSVT	0.8120 ± 0.0209	0.8769 ± 0.0159	0.9077 ± 0.0139	0.8792 ± 0.0174	$\underline{0.9085 \pm 0.0166}$
DLBCL	0.8603 ± 0.0308	0.9896 ± 0.0092	$\underline{0.9948 \pm 0.0112}$	0.9921 ± 0.0150	0.9827 ± 0.0316
Leukemia	0.8654 ± 0.0290	0.9621 ± 0.0200	0.9727 ± 0.0028	0.9666 ± 0.0162	$\underline{0.9950 \pm 0.0150}$
MLL	0.8454 ± 0.0222	0.9716 ± 0.0130	0.9641 ± 0.0235	0.9691 ± 0.0098	$\underline{0.9914 \pm 0.0192}$
Prostate	0.7888 ± 0.0258	0.8727 ± 0.0157	0.9332 ± 0.0141	0.8812 ± 0.0212	$\underline{0.9768 \pm 0.0141}$
Tumors	0.8249 ± 0.0148	0.8193 ± 0.0111	0.8645 ± 0.0059	0.8295 ± 0.0130	$\underline{0.9360 \pm 0.0117}$

in the majority of cases. This indicates that a synchronized rule can usually achieve better performances than a stepwise one. Furthermore, it is easy to obtain the error bounds corresponding to the 95% confidence intervals of the average accuracies, and they are within the generalization error bounds given in Table II in the majority of cases except for three datasets: DLBCL, Leukemia and MLL. This indicates that over-fitting may occur and/or when performing the ten-fold cross validation the training samples in the nine folds may not contain sufficient numbers of samples to distinguish some samples in the remaining fold.

Finally, we computed the ratio of evaluation times consumed respectively by the EK-NN and REK-NN, i.e., $time_{\mathrm{EK-NN}}/time_{\mathrm{REK-NN}}$, for each dataset in each experiment. We collected a set of ten ratios of evaluation times for each dataset. We show the mean and the associated standard deviation of the ten ratios of evaluation times for each dataset in an ascending way according to a proportion of the size of that dataset (i.e., $\log(np)$) in Fig. 8. We can see that, on the one hand, the evaluation times consumed by REK-NN on each dataset is smaller than that consumed by the EK-NN; on the other hand, the larger the size of a dataset, the bigger is the

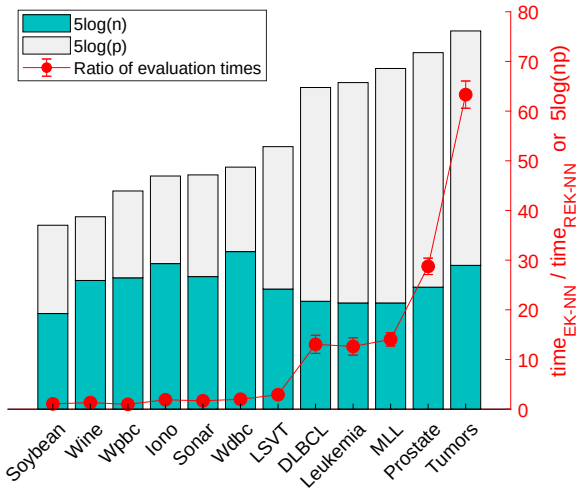

Fig. 8. Ratio of evaluation times consumed by EK-NN and REK-NN.

ratio of evaluation times.

c) Comparing distributed REK-NN with other Spark-based K-NN classifiers: In this experiment, we compared the distributed REK-NN with following two Spark-based K-NN classifiers: Spark-based K-NN and Spark-based EK-NN. The latter was derived directly from the distributed REK-NN by implementing the distributed REK-NN in the whole feature space without feature selection, whereas the former was obtained from the Spark-based EK-NN by replacing the the EK-NN rule by the voting K-NN rule.

The last four datasets with large sample size in Table I were considered in this experiment. To maximize the parallelism and reduce communication overhead simultaneously, three kind of maps were considered, i.e., $M \in \{12, 18, 24\}$, and the interval $[\underline{K}, \overline{K}]$ was set to $[8, 12]$. The parameter K in the Spark-based K-NN and Spark-based EK-NN was set to the optimal K^* found by the distributed REK-NN. When performing evaluation, we used LOO cross-validation instead of ten-fold cross validation because of the large sample size.

Table V shows the optimal K^*, $\mathcal{J}^*$, the selected feature subsets and the generalization error bounds of the distributed REK-NN for different numbers of maps. Table VI reports the classification accuracies among the Spark-based K-NN, Spark-based EK-NN and the distributed REK-NN classifiers.

We can conclude from Tables V and VI that the distributed REK-NN achieves better performance with less features in the majority of cases, and better performances are obtained with larger number of maps. This means that irrelevant features can be removed from large high-dimensional data to improve classification performances. In particular, we can achieve a comparable performance for the Poker dataset by selecting only one feature with 12 and 24 splits. Furthermore, we find that the selected feature subsets are usually different when taking different number of maps. This suggests that the partition of sample space sometimes destroys the data structure when applying approximate nearest neighbor strategy, and thus poor results may be obtained on some datasets. To avoid the influence of partitions on classification, the distributed REK-NN can be an exact nearest neighbor method by sacrificing runtime, as suggested and done in [33].

Finally, we can remark that we did not get the 95% confidence intervals of classification accuracies due to application of the LOO cross-validation, but we believe they could be bounded by the generalization error bounds presented in Table V with a certain probability in the majority of cases. Furthermore, we did not compare the runtime between the distributed REK-NN and Spark-based EK-NN, as done in Fig. 8, because they are approximately equal when the sample volume is large enough.

TABLE V
OPTIMAL K, $\mathcal{J}$, MINIMAL FEATURES SUBSETS AND GENERALIZATION ERROR BOUND WITH $\delta = 0.1$ FOR THE DISTRIBUTED REK-NN

Data	M	K^*	Selected features $\mathcal{B}^*$	$\mathcal{J}^*$	GEB
APS	12	8	113, 123, 100, 8, 42	0.9813	0.0196
	18	12	113, 36, 33, 35, 90, 34	0.9804	0.0205
	24	9	7, 113, 33, 35, 90	0.9827	0.0182
Covtype	12	8	21, 49, 14, 50	0.8890	0.1117
	18	8	21, 49, 14, 50	0.8890	0.1117
	24	8	21, 47, 14, 50	0.8889	0.1118
Kddcup	12	10	21, 31, 8, 2, 30, 34, 6,10	0.9976	0.0025
	18	9	21, 31, 8, 2, 30, 34, 33, 6, 10, 36	0.9977	0.0024
	24	12	21, 31, 8, 2, 30, 34, 6,10	0.9976	0.0025
Poker	12	8	8	0.4464	0.5544
	18	10	8	0.4448	0.5560
	24	10	8, 2, 4, 6	0.5098	0.4910

TABLE VI
AVERAGE ACCURACIES OF SPARK-BASED K-NN, SPARK-BASED EK-NN AND THE DISTRIBUTED REK-NN

Data	M	Spark-based K-NN	Spark-based EK-NN	The distributed REK-NN
APS	12	0.9615	0.9708	0.9807
	18	0.9603	0.9723	0.9814
	24	0.9612	0.9722	0.9817
Covtype	12	0.8765	0.8865	0.8903
	18	0.8743	0.9043	0.8896
	24	0.8657	0.8857	0.8885
Kddcup	12	0.9958	0.9962	0.9979
	18	0.9957	0.9965	0.9981
	24	0.9961	0.9966	0.9973
Poker	12	0.4463	0.4478	0.4467
	18	0.4466	0.4475	0.4449
	24	0.4467	0.4472	0.5099

V. CONCLUSIONS

In this paper, we introduced a new rough evidential K-nearest neighbor classifier for large sample size and/or high-

dimensional data, which performs feature selection and classification simultaneously.

Besides making it possible to represent imperfect knowledge on class membership in the form of mass function, the rough EK-NN can reduce redundant input features and thus the classification complexity of decision boundary. In contrast to the evidential K-NN classifier with feature selection as a preprocessing step, the new rough EK-NN has better performance for some real-world datasets, in particular with high dimensionality.

In order to further handle data with large sample size, we implemented the rough EK-NN in the Spark framework and derived a distributed rough EK-NN, which inherits all the merits of the rough EK-NN but is an approximate nearest neighbor method. Based on the distributed rough EK-NN, we have also derived the Spark-based EK-NN and Spark-based K-NN rules in an intuitive way. Compared to the Spark-based EK-NN and Spark-based K-NN rules, the distributed rough EK-NN has better performances with fewer features for some large datasets.

Several avenues for further research are currently considered, such as extending the classifier to deal with data with uncertain decision labels, and improving feature selection using more sophisticated search strategies.

ACKNOWLEDGEMENT

We are grateful to the significant contributions of editors and anonymous referees. We also thank Dr. Linhao Li (Hebei University of Technology) and Chaoyu Gong (Southeast University) for their contributions.

REFERENCES

[1] T. Denœux and P. Smets, "Classification using belief functions: relationship between case-based and model-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1395–1406, 2006.

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

[3] T. Denœux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 804–813, 1995.

[4] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1987.

[5] T. Denœux, "Conjunctive and disjunctive combination of belief functions induced by non-distinct bodies of evidence," *Artificial Intelligence*, vol. 172, no. 2-3, pp. 234–264, 2008.

[6] ——, "40 years of Dempster-Shafer theory," *International Journal of Approximate Reasoning*, vol. 79, pp. 1–6, 2016.

[7] G. Shafer, *A mathematical theory of evidence*. Princeton, N.J.: Princeton University Press, 1976.

[8] ——, "A mathematical theory of evidence turns 40," *International Journal of Approximate Reasoning*, vol. 79, pp. 7–25, 2016.

[9] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, pp. 191–243, 1994.

[10] T. Denœux, D. Dubois, and H. Prade, "Representations of uncertainty in artificial intelligence: Beyond probability and possibility," in *A Guided Tour of Artificial Intelligence Research*, P. Marquis, O. Papini, and H. Prade, Eds. Springer Verlag, 2020, ch. 4.

[11] Z.-G. Liu, Q. Pan, and J. Dezert, "A new belief-based K-nearest neighbor classification method," *Pattern Recognition*, vol. 46, pp. 834–844, 2013.

[12] Z.-G. Su, T. Denoeux, Y.-S. Hao, and M. Zhao, "Evidential K-NN classification with enhanced performance via optimizing a class of parametric t-rules," *Knowledge-Based Systems*, vol. 142, pp. 7–16, 2018.

[13] L. Zouhal and T. Denoeux, "An evidence-theoretic K-NN rule with parameter optimization," *IEEE Transactions on Systems, Man and Cybernetics-Part C*, vol. 28, no. 2, pp. 263–271, 1998.

[14] Z.-G. Su, P.-H. Wang, and X.-J. Yu, "Immune genetic algorithm-based adaptive evidential model for estimating unmeasured parameter estimating levels of coal powder filling in ball mill," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5246–5258, 2010.

[15] F. Pichon and T. Denoeux, "T-norm and Uninorm-based combination of belief functions," in *In Proceedings of NAFIPS'08*, New York, May 19-22 2008.

[16] Z.-G. Liu, Q. Pan, J. Dezert, and G. Mercier, "Hybrid classification system for uncertain data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2783–2790, 2017.

[17] A. Trabelsi, Z. Elouedi, and E. Lefevre, "Ensemble enhanced evidential k-NN classifier through random subspaces." Lugano, Switzerland: In Proceedings of 14th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), July 10-14 2017, pp. 212–221.

[18] ——, "Ensemble enhanced evidential k-NN classifier through rough set reducts," in *Proceeding of the 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2018*, Springer, Cham, June 11-15 2018, pp. 383–394.

[19] T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta, "A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning," *International Journal of Approximate Reasoning*, vol. 113, pp. 287–302, 2019.

[20] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Hubs in space: Popular nearest neighbors in high-dimensional data," *Journal of Machine Learning Rearch*, vol. 11, pp. 2487–2531, 2010.

[21] C. Lian, S. Ruan, and T. Denoeux, "Dissimilarity metric learning in the belief function framework," *IEEE Transactions on Fuzzy Systems*, vol. 46, no. 12, pp. 1711–1723, 2016.

[22] ——, "An evidential classifier based on feature selection and two-step classification strategy," *Pattern Recognition*, vol. 48, pp. 2318–2327, 2015.

[23] Z. Pawlak, *Rough sets, Theoretical Aspects of Reasoning About Data*. Dordrecht: Kluwer Academic Publishers, 1991.

[24] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Information Sciences*, vol. 178, pp. 3577–3594, 2008.

[25] Y. Yang, D. Chen, and H. Wang, "Active sample selection based incremental algorithm for attribute reduction with rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 825–838, 2017.

[26] Q. Hu, L. Zhang, Y. Zhou, and W. Pedrycz, "Large-scale multimodality attribute reduction with multi-kernel fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 226–238, 2018.

[27] J. Dai, Q. Hu, H. Hu, and D. Huang, "Neighbor inconsistent pair selection for attribute reduction by rough set approach," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 937–950, 2018.

[28] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[29] ——, "Mapreduce: A flexible data processing toll," *Communications of the ACM*, vol. 53, no. 1, pp. 72–77, 2010.

[30] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning spark: lightning-fast big data analysis*. Sebastopol, CA, USA: OReilly Media Inc., 2015.

[31] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012, pp. 2–2.

[32] G. Chatzimilioudis, C. Costa, D. Zeinalipour-Yazti, W. Lee, and E. Pitoura, "Distributed in-memory processing of all k nearest neighbor queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 925–938, 2016.

[33] J. Maillo, S. Ramirez, I. Triguero, and F. Herrera, "KNN-IS: An iterative Spark-based desition of the k-nearest neighbors classifier for big data," *Knowledge-Based Systems*, vol. 117, pp. 3–15, 2017.

[34] J. Maillo, S. Garcia, and J. Luengo, "Fast and scalable approaches to accelerate the fuzzy k nearest neighbors classifier for big data," *IEEE Transactions on Fuzzy Systems*, 2019, dOI: 10.1109/TFUZZ.2019.2936356.

[35] T. Denœux, "Analysis of evidence-theoretic decision rules for pattern classification," *Pattern Recognition*, vol. 30, no. 7, pp. 1095–1107, 1997.

[36] ——, "Decision-making with belief functions: a review," *International Journal of Approximate Reasoning*, vol. 109, pp. 87–110, 2019.

[37] Q. Hu, W. Pedrycz, D. Yu, and J. Lang, "Selecting discrete and continuous features based on neighborhood decision error minimization," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 40, no. 1, pp. 137–150, 2010.
[38] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, no. 9, pp. 917–922, 1977.
[39] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424–1437, 2004.
[40] Z.-G. Su and P.-H. Wang, "Minimizing neighborhood evidential decision error for feature evaluation and selection based on evidence theory," *Expert Systems with Applications*, vol. 39, no. 1, pp. 527–540, 2012.
[41] T. Denœux and O. Kanjanatarakul, "Evidential clustering: a review," in *5th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM)*, Da Nang, Dec. 2016, pp. 24–35.
[42] Z.-G. Su and T. Denoeux, "BPEC: Belief-Peaks Evidential Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 1, pp. 111–123, 2019.
[43] J. Li and X. Wang, "A new generalization error bound (in Chinese)," Hebei University, Tech. Rep., 2012.
[44] C. L. Blake and C. J. Merz, "UCI Repository of Machine Learning." Available: http://www.ics.uci.edu/ mlearn/MLRepository, 1998.
[45] "Kent Ridge Bio-medical Dataset." Available: http://datam.i2r.a-tar.edu.sg/datasets/krbd/index.html, 2015.
[46] J. AlcalaFdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "KEEL DATA-Mining software tool data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.
[47] Q. Hu, L. Zhang, D. Zhang, W. Pan, S. An, and W. Pedrycz, "Measuring relevance between discrete and continuous features based on neighborhood utual information," *Expert Systems with Applications*, vol. 38, pp. 10 737–10 750, 2011.
[48] Q. Hu, D. Yu, Z. Xie, and J. Liu, "Fuzzy probabilistic approximation spaces and their information measures," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 191–201, 2006.

Zhi-gang Su received his M.S. and Ph.D from Southeast University (SEU), China, in 2006 and 2010 respectively, and then became an assistant professor with the Dept. of Power Engineering and Automation, School of Energy and Environment at the SEU. In 2013, he became an associate professor. From 2014 to 2015, he worked as a visiting scholar with Dept. of Electrical and Computer Engineering at The University of Texas at San Antonio in USA. His research interests concern artificial intelligence and theory of belief function with applications to pattern recognition, data mining and in particular to similar practical issues in thermal power engineering. He is also interested in nonlinear control theory with applications to thermal processes, and he was selected as one outstanding reviewer by the journal *Automatica* in 2016-2017. He is the first author of more than 20 journal papers in areas of machine learning, reasoning with uncertainty, pattern recognition and automation. He is an Associate Editor of the journal *Array* (Elsevier).

Qinghua Hu (SM'13) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Post-Doctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology with Tianjin University, Tianjin, China. He has authored over 200 journal and conference papers in the areas of granular computing-based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His research interests include rough sets, granular computing, and data mining for classification and regression.

Prof. Hu was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology, the International Conference on Machine Learning and Cybernetics in 2014,and the General Co-Chair of International Joint Conference on Rough Sets 2015. He is currently the PC-Co-Chairs of China Conference on Machine Learning 2017 and Chinese Conference on Computer Vision 2017. He is currently an Associate Editor for the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Thierry Denœux is a Full Professor (Exceptional Class) with the Department of Information Processing Engineering at the Compiègne University of Technology (UTC), France. He is the director of the Laboratory of Excellence on "Technological Systems of Systems" and the president of the Belief Functions and Applications Society. In 2019, he was appointed as a senior member of Institut Universitaire de France. His research interests concern reasoning and decision-making under uncertainty and, more generally, the management of uncertainty in intelligent systems. His main contributions are in the theory of belief functions with applications to statistical inference, pattern recognition, machine learning and information fusion. He is the author of more than 300 papers in journals and conference proceedings and he has supervised more than 30 PhD theses. He is the Editor-in-Chief of the *International Journal of Approximate Reasoning* and *Array* (Elsevier), and an Associate Editor of several journals including *Fuzzy Sets and Systems* and *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*.