Combining complementary information sources in the Dempster–Shafer framework for solving classification problems with imperfect labels

Mahdi Tabassian\textsuperscript{a,b,*}, Reza Ghaderi\textsuperscript{a}, Reza Ebrahimpour\textsuperscript{b,c}

\textsuperscript{a}Faculty of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran
\textsuperscript{b}School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5746, Naivaran Sq., Tehran, Iran
\textsuperscript{c}Brain and Intelligent Systems Research Lab, Department of Electrical and Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran

\textbf{Abstract}

This paper presents a novel supervised classification approach in the ensemble learning and Dempster–Shafer frameworks for handling data with imperfect labels. Through a re-labeling procedure and by utilizing the prototypes of the pre-defined classes, the possible uncertainty in the label of each learning sample is detected and based on the level of ambiguity concerning its class membership, it is assigned to only one class or a subset of the pre-defined classes. In order to properly estimate the class labels, complementary representations of the data are employed using a diversity-based feature space selection method. Multilayer perceptron neural network is used to learn characteristics of the data with new labels in each feature space. For a given test pattern the outputs of the neural networks, which are generated based on the evidences raised from the feature spaces, are considered as basic belief assignments (BBAs). The BBAs represent partial knowledge of a test sample’s class and are combined using Dempster’s rule of combination. Experiments on artificial and real data demonstrate that by considering the ambiguity in labels of the data, the proposed method can provide better results than single and ensemble classifiers that solve the classification problem using data with initial imperfect labels.

\textbf{1. Introduction}

Beside the ability of a classification technique in learning characteristics of the given data and making proper decisions on the classes of the previously unseen samples, the appropriateness of the class labels assigned to the data is another important factor that influence the accuracy of a supervised classification methodology. In the classical supervised learning framework, it is assumed that the class labels have been correctly assigned to the data and the focus is on the learning and decision-making capabilities of the classification scheme. However, in some applications, unambiguous label assignment may be difficult, imprecise or expensive. Such situations can occur when differentiating between two or more classes is not easy due to lack of information required for specifying certain labels to data or the difficulty of labeling complicated data of the problem at hand. In such cases, the classifier cannot suitably identify the nature of the classification problem and its performance could be degraded dramatically.

Combination of multiple classifiers’ decisions induced from different information sources is a promising solution to gain acceptable classification performance [1–3]. This strategy has also proved to be an appropriate approach for improving the performance of a classification system that deals with imprecise and/or uncertain information. Although there are different strategies for constructing an ensemble structure, utilizing a framework which is able to manipulate partial knowledge is of great importance when the goal is to solve a classification problem with uncertain data.

The Dempster–Shafer (D–S) theory of evidence [4] is a well-suited framework for representation of partial knowledge. Compared to the Bayesian approach, it provides a more flexible mathematical tool for dealing with imperfect information. It offers various tools for combining several items of evidence and, as understood in the transferable belief model (TBM) [5], allows to make decision about the class of a given pattern through transforming a belief function into a probability function.

Thanks to its flexible characteristics, the D–S theory can be used in combination with other mathematical frameworks such as fuzzy logic [6] and rough set theory [7,8] and it provides a suitable theoretical tool for combining classifiers especially those learned using imprecise and/or uncertain data. Rogova [9] introduced an approach for combining neural network classifiers based on the D–S theory and used the proximity of a given sample to some reference
vectors for defining mass functions. Each reference vector was considered as the mean of the neural network outputs for training samples belong to one of the main classes. By combining the mass functions using Dempster’s rule of combination and merging the evidences from all the neural networks with the same combination rule, final decision about the class of a test pattern was made. De-noeux [10] proposed an evidential neural network method for pattern classification problems and used the evidence $K$-nearest neighbor algorithm [11] for assigning a pattern to a class by computing distances to limited number of prototypes. It was shown that in the sensor fusion application, this approach can provide an optimal solution in case of sensor failures. The issue of uncertainty representation in pattern classification in the D–S framework was addressed in [12] and a variant of the bagging approach was proposed to improve the reliability of the outputs of the evidence $K$-nearest neighbor algorithm [11]. In [13] the authors proposed a framework based on the supervised learning for data fusion. They used the D–S theory and addressed the issues of constructing evidence structure and dependence among information sources by utilizing neural networks. It has been shown that this approach could improve the performance of the D–S evidence theory and the majority voting method and it brought satisfactory results considering the speed of learning convergence. A pairwise classifier combination based on the D–S theory was proposed in [14] and it was argued that the belief function framework is an appropriate tool to deal with partial information provided by each pairwise classifier about the class of the object under consideration.

This paper addresses the supervised learning in which the class memberships of the training data are subject to ambiguity. In order to properly estimate the class labels, complementary feature spaces are constructed from the data and an approach is suggested, based on the supervised information, for detecting inconsistencies in the labels of the learning data and assigning crisp and soft labels to them. MLP neural network is used as base classifier and its outputs are interpreted as BBA. Final decision about the class of a test pattern is made by combining the BBAs produced by the base classifiers using Dempster’s rule of combination.

The rest of the paper is organized as follows. In Section 2 basic concepts of the evidence theory are reviewed. Details of our proposed method are described in Section 3. The employed feature space selection methodology is presented in Section 4. It is followed by the experimental results on artificial and real data in Section 5 and finally, Section 6 draws conclusion and summarizes the paper.

2. Dempster–Shafer theory of evidence

The Dempster–Shafer (D–S) theory of evidence [4], also referred to as evidence theory, is a theoretical framework for reasoning with uncertain and partial information and it can be considered as a generalization of Bayesian probability with flexibility. Several models for uncertain reasoning has been proposed based on the D–S theory. An example is the transferable belief model (TBM) [5].

2.1. Basic concepts

Let $\Omega = \{\omega_1, \ldots, \omega_m\}$ be a finite set of mutually exclusive and exhaustive hypotheses called the frame of discernment. A basic belief assignment (BBA) or mass function is a function $m: 2^\Omega \rightarrow [0, 1]$, and it satisfies the two following conditions:

\[ m(\emptyset) = 0, \]
\[ \sum_{A \subseteq \Omega} m(A) = 1. \]

where $\emptyset$ is the empty set and a BBA that satisfies the condition $m(\emptyset) = 0$ is called normal. The subsets $A$ of $\Omega$ with nonzero masses are called the focal elements of $m$ and $m(A)$ indicates the degree of belief that is assigned to the exact set of $A$ and not to any of its subsets. There are also two other definitions in the theory of evidence. They are belief and plausibility functions associated with a BBA and are defined respectively, as follow:

\[ \text{Bel}(A) = \sum_{B \subseteq A} m(B), \]
\[ \text{Pl}(A) = \sum_{A \subseteq \Theta} m(B). \]

Bel$(A)$ represents the total amount of probability that is allocated to $A$, while Pl$(A)$ can be interpreted as the maximum amount of support that could be given to $A$. Note that the three functions $m$, Bel and Pl are in one-to-one correspondence and by knowing one of them, the other two functions could be derived.

2.2. Combination of BBAs

Let $m_1$ and $m_2$ be two BBAs induced by two independent items of evidence. These pieces of evidence can be combined using Dempster’s rule of combination (also called the orthogonal sum) which is defined as:

\[ m(C) = \frac{\sum_{A \cap B = \emptyset} m_1(A) \times m_2(B)}{1 - \sum_{A \cap B \neq \emptyset} m_1(A) \times m_2(B)}. \]

Combining BBAs using Dempster’s rule of combination is possible only if the sources of belief are not totally contradictory which means that there exist two subsets $A \subseteq \Omega$ and $B \subseteq \Omega$ with $A \cap B \neq \emptyset$ such that $m_1(A) > 0$ and $m_2(B) > 0$.

2.3. Decision-making

After combining all pieces of evidence, a decision has to be made using the final belief structure. Different strategies can be adopted in this stage in which three main choices are (i) selecting the hypothesis with the highest degree of belief, (ii) utilizing the plausibility transformation for translating belief functions into probability functions [15], and as proposed by Smets [16], (iii) employing the pignistic transformation to derive probability functions from the belief functions. In this paper the pignistic transformation is used because it was shown that just this strategy is able to convince basic rationality necessities [16].

In [5], the TBM has been presented as a two-level mental model: the credal level where beliefs are represented and merged using belief functions, and the pignistic level where decision-making is performed. Pignistic probabilities are computed in the second level of the TBM. By uniformly distributing the mass of belief $m(A)$ among its elements for all $A \subseteq \Omega$, a pignistic probability distribution is defined as:

\[ \text{BetPr}(\omega) = \sum_{A \subseteq \Omega, \omega \in A} \frac{1}{|A|} \frac{m(A)}{1 - m(\emptyset)}, \quad \forall \omega \in \Omega, \]

where $|A|$ is the cardinality of subset $A$ and for normal BBAs (with $m(\emptyset) = 0$), $m(A)/(1 - m(\emptyset))$ would be $m(A)$.

3. The proposed ensemble classification scheme

In this paper the issue of ambiguity in labels of the learning data is taken into account. Although in the classification problems studied in this research each train pattern has been assigned to only one class, this assignment is subject to uncertainty due to lack of information required for precise label allocation or the difficulty of categorizing similar patterns into distinct classes. In the proposed method, the procedure of label assignment is reconsidered
and based on the level of uncertainty concerning the class membership of each train pattern, it is allowed to have a crisp class label or a soft label comprising any possible subset of the pre-defined classes. In our method, the accepted uncertainties are reduced by making use of several complementary representations of the data and merging the evidences raised from these representations through the D–S framework. Note that the use of the proposed method is advantageous when it is applied to classification problems with imperfect class labels and/or heavily overlapping class distributions. Otherwise, utilizing complementary sources of information in the ensemble learning framework can provide satisfactory results.

Fig. 1 shows the architecture of our proposed method. There are two main phases involved in the implementation of the method including re-labeling the learning data and classifying an input test sample by combining decisions of neural networks trained on the learning data with the new labels.

3.1. Re-labeling

Let \( \Omega = \{\omega_1, \ldots, \omega_M\} \) be a set of \( M \) classes and \( x \) be a data point described by \( n \) features in the training set which is associated to one class in \( \Omega \) with certainty. The goals of this stage are (i) to detect inconsistencies in the labels of the training data and, (ii) to reassign each train sample to just one main class or any subset of the main classes based on the level of ambiguity concerning the class membership of that sample.

Let \( P \) be a \( M \times n \) matrix containing prototype vectors of the main classes and let \( D_i = [d_{i1}, \ldots, d_{im}] \) be the set of distances between train sample \( x_i \) and the \( M \) prototypes according to some distance measure (e.g. the Euclidian one). Initial label of \( x_i \) is ignored and by utilizing the information provided by the vector \( D_i \), uncertainty detection and class reassignment for this sample are performed in a three-step procedure:

Step 1: The minimum distance between \( x_i \) and the class prototypes is taken from the vector \( D_i \),

\[
d_i = \min(d_{il}) \quad l = 1, \ldots, M
\]

and \( d_i \) is called \( d_{\min} \).

Step 2: A value \( 0 < \mu_i < 1 \) is calculated for each of the \( M \) classes using the following function:

\[
\mu_i(x_i) = \frac{d_{\min} + \beta}{d_i + \beta} \quad l = 1, \ldots, M
\]

in which \( 0 < \beta < 1 \) is a small constant value and ensures that the utilized function allocates a value greater than zero to each of the \( M \) classes even if \( d_{\min} = 0 \). \( \mu_i \) is a decreasing function of the difference between the \( d_{\min} \) and \( d_i \) and has values close to 1 for small differences.

Step 3: A threshold value \( 0 < \tau < 1 \) is defined and based on the level of ambiguity regarding the class membership of train sample \( x_i \), this sample can be assigned to (i) a set of classes if the corresponding values of \( \mu_i \) for these classes are greater than or equal to \( \tau \) (soft labeling) or (ii) just one main class, which has the closest prototype to \( x_i \) and \( \mu_i = 1 \), where \( x_i \) is far away from the other main classes’ prototypes (crisp labeling).

Close distances between a train pattern and some of the class prototypes can be interpreted as an indication of ambiguity in the pattern’s label and in such cases a soft label is assigned to that train pattern. The above procedure is repeated for all training samples and all feature spaces.

Since several representations of the data are employed to provide complementary information, it can be expected that if a soft label has been assigned to a train sample in one of the feature spaces, this sample could belong to one class or a subset of the main classes with less uncertainty in the other feature spaces. In this way, the negative effects of crisp but imperfect labels of some learning samples on the classification performance can be reduced and a satisfactory classification result will be achieved by combining evidences from the multiple feature spaces.

Example. Let for a 3-class problem, the set of distances between train samples \( x_1 \) and \( x_2 \) and the class prototypes are as follows:

\[
\begin{align*}
D_1 &= [1.134, 4.16], \\
D_2 &= [4.16].
\end{align*}
\]

By considering \( \beta = 0.001 \), the values of \( \mu \) for these two samples are calculated using Eq. (8),

\[
\begin{align*}
\mu(x_1) &= [1 0.77 0.25] \\
\mu(x_2) &= [0.25 0.1 0.167].
\end{align*}
\]

By defining a threshold value \( \tau = 0.6 \) and comparing the values of \( \mu \) with \( \tau \), label assignments for \( x_1 \) and \( x_2 \) are performed:

\[
\begin{align*}
&\{ \mu(x_1) = 1 < \tau \Rightarrow \text{assign } x_1 \text{ to } \{\omega_1\} \}, \\
&\{ \mu(x_1) = 0.25 < \tau \Rightarrow \text{assign } x_1 \text{ to } \{\omega_2\} \}, \\
&\{ \mu(x_2) = 0.167 < \tau \Rightarrow \text{assign } x_2 \text{ to } \{\omega_3\} \}.
\end{align*}
\]

Since there is not a large distance between train sample \( x_1 \) and prototype vectors of classes 1 and 2, a soft label comprising both classes is assigned to \( x_1 \). On the other hand, for train sample \( x_2 \) only class 2, with the closest prototype to \( x_2 \), has a value of \( \mu \) greater than \( \tau \) and consequently the crisp label \( \{\omega_2\} \) is assigned to \( x_2 \).

In the current study two approaches for calculating prototype vectors of the main classes have been adopted. The first approach computes the mean vectors of the main classes as a general estimation of class prototypes and uses these vectors for re-labeling all train samples while the second one aimed at using a local approach for computing the prototype vectors. Here, we propose to use the local mean-based (LMB) method introduced in [17] for
estimating class prototypes. This approach can give better estimation of the main classes’ prototypes and as will be shown in the next experiments is able to provide robustness against labeling errors. The LMB method is reviewed below.

### 3.1.1. The local-mean based method for prototype estimation

Let \( \{x_i\}_{i=1}^{N_i} \) denotes the set of training samples from class \( \omega_i \) where \( N_i \) is the number of training samples belong to this class. For each main class \( \omega_i \), \( K \)-nearest training samples of a test pattern \( x \), \( \{x_1, x_2, \ldots, x_K\} \), are selected using the Euclidean distance. The local mean vector of class \( \omega_i \) is computed by utilizing the \( K \)-nearest neighbor (K-NN) set from this class:

\[
p_i = \frac{1}{K} \sum_{j=1}^{K} x_j.
\]

In the LMB method a test sample is assigned to the class with the closest mean vector.

In this paper this strategy is employed for computing prototypes of the main classes. For each train sample \( x \), the re-labeling procedure is performed by employing \( p_i \) as the prototype of class \( \omega_i \) and the set of distances between this train sample and all class prototypes \( \{D\} \) is incorporated in Eq. (8).

It has been shown that the LMB classification method is robust to outliers and has favorable performance in classification problems with high dimensionality and small training sample size [17]. Beside utilizing the LMB approach for computing class prototypes, which is unique for each train sample, our proposed method takes advantage of robustness to outliers property of the LMB method.

### 3.2. Training and decision-making

MLP neural network, which is capable of making a global approximation to the classification boundary, is used as base classifier. The learning set of each feature space, which consists data with new crisp or soft labels, is employed to train a base classifier. Since different types of features are extracted from the data, the training data in each feature space have their own level of uncertainty in class labels. As a result, after the re-labeling procedure the number and type of classes in each feature space could be different from the other one and base classifiers with different models (different number of output nodes) are trained on these feature spaces. In this fashion, diversity among the base classifiers can be achieved.

In the test phase, same types of features as the training stage are extracted from the test data and they are applied to their corresponding trained base classifiers. The output values of each base classifier can be interpreted as measures of confidence associated with different decisions. Consider a test sample that is located at one of the ambiguous areas in the feature space. It is expected that after the re-labeling stage, the train samples belong to this area will have soft labels comprising their initial crisp class labels. So, the base classifiers can detect similarities of the given test sample to characteristics of the train samples belong to this soft class and they will assign great values of probability to this soft class in their output results.

For combining the evidences induced by the feature spaces using Dempster’s rule of combination, the decisions of the base classifiers should be converted in the form of BBAs. This can be done through normalizing the outputs of the base classifiers by a softmax operator as:

\[
m_i(\omega_j) = \frac{\exp(O_{ij})}{\sum_{j=1}^{C} \exp(O_{ij})}, \quad j = 1, \ldots, C,
\]

where \( O_{ij} \) is the \( j \)-th output value of the \( i \)-th base classifier, \( C \) is the number of classes in the \( i \)-th feature space after the re-labeling stage and \( m_i(\omega_j) \) is the mass of belief given to class \( \omega_j \).

Note that, although our method allows to compute a BBA \( m_i \) that could have any focal set over the set of the main classes, the BBA contains only a set of states which have corresponding classes in the \( i \)-th feature space after the re-labeling stage. In other words, the frame of discernment is composed of the initial classes and it is similar for all classifiers but each base classifier makes a decision about a subset of this frame. Since after the re-labeling stage each train sample will have a crisp or soft class label, the quantity \( m_i(\phi) \) is zero and BBAs are normal.

For making decision on the class of a given test sample, the opinions of the base classifiers are merged using Dempster’s rule of combination. By applying the pignistic transformation on the resulting BBA, the pignistic probability will be achieved and the test sample will be assigned to the class with the largest pignistic probability.

Note that, although the main contribution of our method is to classify data with ambiguous labels, its application can be extended to classification problems which involve heavily overlapping class distributions and nonlinear class boundaries.

---

**Fig. 2.** Generic representation of the re-labeling procedure using the LMB method for computing class prototypes. (a) Learning samples before the re-labeling, (b) computed class prototypes for samples \( x_1 \) and \( x_2 \) and their corresponding distances to the classes and (c) learning samples after the re-labeling procedure.
4. Feature space/classifier selection

Assuming that by employing different feature extraction techniques or by sampling several subsets of features from the original feature space different representations of the data are available. By training a classifier on each of these feature spaces a pool of classifiers will be obtained. One of the necessary conditions for the success of the proposed method, as well as other ensemble networks, is the appropriate choice of the base classifiers. In order to choose an optimal subset of the base classifiers from the classifier pool, a classifier selection methodology should be adopted which necessitates the incorporation of a search algorithm and a selection criterion. Fig. 3 shows the overall implementation procedure of a typical classifier selection algorithm.

4.1. The random subspace method

The Random Subspace Method (RSM) [18] is a popular sampling approach for building an ensemble structure. In this method, different subsets of features are randomly sampled from the original feature space and multiple classifiers are constructed in the low dimensional feature spaces. Finally, the decision of the ensemble network on a given test pattern is made by majority vote.

In the feature space/classifier selection stage of our proposed method the RSM is used for creating a pool of classifiers. Let for a training set \( \{x_i\}_{i=1}^N \), each sample is described by a \( n \)-dimensional vector. In the RSM, \( g < n \) features are randomly selected from the original \( n \)-dimensional feature space and in this way, a \( g \)-dimensional random subspace is obtained. This process is repeated \( S \) times and by building a classifier in each of the random subspaces, a classifier pool of size \( S \) is achieved.

4.2. Search algorithm and selection criterion

In this paper the forward search algorithm which is known as the most intuitive greedy approach is used to explore among the classifier pool. The optimality of this search method has been shown in [19] based on extensive experiments and it has been demonstrated that the forward search method can outperform the exhaustive search algorithm which suffers from the overfitting problem. The forward search algorithm starts with the most accurate classifier and in a sequential manner, other classifiers are added to the initial one. By examining the performance of the ensemble networks made from the first classifier and any of the remaining classifiers, a classifier that leads to the best value of an evaluation criterion is selected as the second classifier. This process is repeated and finally a set of best classifiers is chosen.

Because in our proposed method the ensemble network has a fixed size, the first \( R \) selected classifiers are incorporated into our model.

In order to assess the quality of the selected classifiers, a diversity-based selection criterion is adopted in the classifier selection stage. The interrater agreement [20], \( k \), which is a non-pairwise diversity measure is used. This classifier selection criterion measures the level of agreement (or disagreement) between classifiers and picks the most diverse subset of classifiers. The \( k \) is briefly reviewed in the following.

Let \( B = \{B_1, \ldots, B_k\} \) be a system of \( S \) classifiers and let \( \{x_i\}_{i=1}^N \) be the data set containing \( N \) labeled data. Let \( y_i = [y_{i1}, \ldots, y_{ik}] \) be the joint output of a system for the \( i \)th input sample \( x_i \) where \( y_{ij} \) denotes the output of the \( j \)th classifier and it is defined to be 1 if \( B_j \) recognizes \( x_i \) correctly and 0 otherwise. Let \( s(x_i) \) denote the number of classifiers from \( B \) that correctly recognize the input sample \( x_i \). It can be expressed by

\[
s(x_i) = \sum_{j=1}^{S} y_{ij}.
\]

Denote by \( a_j = \frac{1}{N} \sum_{i=1}^{N} y_{ij} \) the classification rate of the \( j \)th classifier, the average accuracy of the ensemble is defined by

\[
a = \frac{1}{S} \sum_{j=1}^{S} a_j.
\]

Using the notation presented above, the \( k \) measure of diversity can be expressed as

\[
k = 1 - \frac{\sum_{i=1}^{N} s(x_i) (S - s(x_i))}{NS(S - 1)(1-a)^2}
\]

where \( k = 0 \) indicates that the classifiers are independent. Small values of \( k \) leads to better diversity and a negative \( k \) shows negative dependency (high diversity) among the classifiers [20].

5. Experimental results

In this section, we report experimental results on artificial and real datasets to highlight the main aspects of our proposed method. We focus on supervised classification problems in which the labels of the learning data are only partially known or those consist data with erroneous labels. Since in Eq. (8) \( \beta \) is used to prevent possible unwanted exceptions and it has not a considerable influence on the re-labeling process, a value of \( \beta = 0.001 \) was adopted in the re-labeling stage.

![Fig. 3. Overall stages involved in a typical feature space/classifier selection procedure.](image-url)
Fig. 4. Generic representation of the artificial data. For generating new feature spaces, the classes are transferred to their near corners in clockwise direction. To demonstrate how different parts of a class overlap with other classes in different feature spaces, Class 1 is divided to four parts and positions of these parts in the three feature spaces are shown. (a) First feature space, (b) second feature space and (c) third feature space.

Fig. 5. Representation of partitioning the first learning set into crisp and soft subsets by the proposed re-labeling approach with $\tau = 0.8$. (a) First feature space, (b) second feature space and (c) third feature space.

Fig. 6. Classification error rates as a function of number of hidden nodes for single MLPs trained on one of the feature spaces, ensemble networks constructed by combining the outputs of the single MLPs using fixed combining rules and our proposed method. (a) First training set ($\tau = 0.8$ for the proposed method), (b) second training set ($\tau = 0.75$ for the proposed method) and (c) third feature space ($\tau = 0.7$ for the proposed method).
5.1. Artificial data

We used a two-dimensional data so that the results could be easily represented and interpreted. The dataset was made of three classes with equal sample size, Gaussian distribution and common identity covariance matrix. We generate 150, 300 and 1500 independent samples for training, validation and testing, respectively. The center of each class was located at one of the vertices of an equilateral triangle. In order to use different representations of the data, the classes were transferred to their near vertices in clockwise direction and in this fashion two other feature spaces were generated. In each feature space, a unique subset of each class overlapped with the data of the other classes. It means that the high ambiguity pertaining to class membership of a pattern in one of the feature spaces can be reduced because this pattern may be located in the non-overlapping or less ambiguous areas in the other feature spaces. To evaluate the performance of the proposed method in classification tasks with different levels of ambiguity in class labels, the length of the equilateral triangle (the distance between the neighborhood classes) was varied in \{1, 2, 3\} and in this way, three cases from strongly overlapping to almost separated classes were studied. Fig. 4 represents graphi-
cally the explained above procedure for generating the artificial data.

Since a controlled dataset with known and similar class characteristics was used in this experiment, a general estimation of the class prototypes was adopted. The centers of the main classes were taken by averaging the learning data of each class and the resulting vectors have been considered as the main classes’ prototypes. The optimality of the employed local approach for computing the prototype vectors will be investigated later using the real data.

5.1.1. Soft and crisp label generation

In order to qualitatively examine how training samples with different levels of ambiguity in their labels are treated in the re-labeling stage, the results of partitioning the first training set with \( \tau = 0.8 \) is demonstrated in Fig. 5. Each partition is represented by a convex hull and its class label indicates that the samples of that partition were assigned to which subset of the main classes. It can be seen that soft labels were assigned to samples situated in the boundaries of the main classes or those located at the ambiguous regions.

5.1.2. Performance comparison

The performance of the proposed method was compared to the following classification schemes:

- Three MLP neural networks trained separately on one of the feature spaces.
- Ensemble networks constructed by merging the decisions of the single MLPs using three fixed combining rules (average, product and maximum).

The above classification approaches discarded the possible uncertainties in class labels and employed data with initial certain labels. The MLPs were trained by the Levenberg–Marquardt algorithm with default parameters and 80 epochs of training and had one hidden layer. Each neural network was trained 50 times with random initializations. To evaluate the performance of our proposed method based on different threshold values, we generated 19 training sets from each original training data by varying \( \tau \) from 0.05 to 0.95 with a step size of 0.05.

The average test error rates of the employed classification models for the three datasets and for different number of hidden nodes are represented in Fig. 6. Note that, the classification results of our proposed method with the best \( \tau \) for each dataset are shown. Obviously the improvements that the ensemble networks have caused over the single MLPs are very significant which indicates that these networks were constructed by a set of base classifiers trained on complementary feature spaces. As can be seen, our method yields considerably better classification results than the other classifiers when the first dataset, as the most difficult set with the highest level of uncertainty in class labels, was used (Fig. 6(a)). As shown in Fig. 6(b) and (c), the differences between the test performances of our proposed scheme and the three ensemble networks on the second and third datasets are small. So, as mentioned before, there is not much benefit to be gained from considering ambiguity in perfectly labeled data or in a classification problem with well separated classes.

5.2. Real data

In order to specify if the results obtained on the artificial data hold for higher dimensional real data, the proposed method was applied to four datasets from the well-known UCI repository [21] and subsets of UIUCTex [22] and Hoda Farsi handwritten digit [23] databases.

5.2.1. Experimental settings

In the experimental study presented in this section the sizes of the classifier pools for all datasets were fixed to 50. The utilized datasets were randomly partitioned into train, validation and test sets. The experiments were repeated five times for each randomly partitioned data.

In the classifier selection stage, a subset of 10 classifiers were selected for each dataset from the pool of classifiers generated by the RSM and the feature spaces correspond to the selected classifiers were used in the re-labeling phase of the proposed method.

In the procedure of re-labeling the real data, the LMB method was employed for computing class prototypes. The re-labeling process was performed by examining 10 \( K \)-NNs (\( K = 1, \ldots, 10 \)) and 14 \( \tau \)s (\( \tau = 0.3, \ldots, 0.95 \) with a step size of 0.05) which results in 140 new training set with crisp and soft labels. The generated training sets were used in the proposed ensemble structure and based on the evaluated performances on the validation data, a small set of suitable training data were selected to assess the performance of the proposed method in the test phase.

In order to study the robustness of the proposed method against erroneous labels, the initial labels of the employed databases were artificially corrupted. By making use of the Bernoulli distribution, the label of each train sample \( x \) was changed to another class in which the probability of mislabeling takes values from \( \{0.1, 0.15, 0.2, \ldots, 0.5\} \).

5.2.1.1. UCI data. Table 1 gives the characteristics of the UCI datasets employed in this research. Here, all databases with their original sample sizes were used except the Waveform data where a subset of 1050 randomly selected samples from 5000 original samples was employed.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Selected features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brick</td>
<td>20</td>
</tr>
<tr>
<td>Glass</td>
<td>5</td>
</tr>
<tr>
<td>Carpet</td>
<td>20</td>
</tr>
<tr>
<td>Stone</td>
<td>5</td>
</tr>
<tr>
<td>Fabric</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 8. Samples of the Hoda Farsi handwritten digit images.
Using the validation data and by varying the sizes of the feature spaces within a range given in Table 2, the appropriate number of dimension for each dataset was determined.

Note that for the Wine and Waveform data, only a pair of classes was used for artificial label corruption.

5.2.1.2. UIUCTex data. The database is composed of 25 classes and 14 types of texture surfaces. In our experiments five types of textures including Brick, Glass, Carpet, Stone and Fabric were selected from the original database and each one was considered as a separate classification problem. The first four problems have two classes of texture images while there are four classes of textures for the Fabric data. Fig. 7 shows samples of the employed texture datasets. There are 40 images of size $640 \times 480$ for each class. The original images were rescaled to $84 \times 70$ and 24, 6 and 10 samples of each class were used for training, validation and testing sets, respectively.

Principal Component Analysis (PCA) [24] was used for reducing the sizes of the original feature spaces to the maximum allowed dimension (number of training patterns − 1) and the RSM was applied to the obtained subspaces. In order to determine the appropriate cardinalities of the feature spaces, subspaces with six different dimensions (5, 10, 15, 20, 25 and 30) were examined and the best dimensions were selected based on the results on the validation data. Table 3 lists the feature space cardinality of each dataset employed in the feature space/classifier selection stage.

5.2.1.3. Hoda Farsi handwritten digit data. A subset of the Hoda Farsi handwritten digit dataset is used in our experiments. Only the samples of digits 2, 3 and 4 that have analogous shapes have been used. All classes had the same number of samples and 75, 225 and 300 samples were used for training, validation and testing sets, respectively. Digits were represented as $34 \times 29$ black-and-white images. Fig. 8 illustrates some samples from the three classes.

Similar to the UIUCTex data, the PCA method was used for reducing the size of the original feature space. The appropriate cardinality of the final subspace was determined by examining six different dimensions (10, 15, 20, 25, 30 and 35) and based on the performance on the validation data, 35-dimensional subspaces were employed in the feature space/classifier selection stage.

5.2.2. Performance comparison

Similar to the experiments carried out using the artificial data, we compared the performance of our proposed method on the real data with those of ensemble neural networks constructed by pooling together the results of the single MLPs with the fixed combining methods. The results of our method were also compared with those of the evidence-based neural network ensemble proposed by Rogova [9]. Note that, similar feature spaces as those used in our proposed model have been incorporated into these ensemble networks. All the MLPs in the proposed classification scheme and the other ensemble structures were trained by the resilient back-propagation algorithm with default parameters and 50 epochs of training. They had one hidden layer and were trained 10 times with random initial weights. The number of hidden neurons was selected based on the results obtained on the validation set.

Average classification error rates (as a function of the label corruption level) of the employed ensemble structures on the UCI, UIUCTex and Hoda Farsi digit databases are illustrated in Figs. 9–11, respectively.

Using the validation data and by varying the sizes of the feature spaces within a range given in Table 2, the appropriate number of dimension for each dataset was determined.

Note that for the Wine and Waveform data, only a pair of classes was used for artificial label corruption.
isfactory performance of our method in dealing with the texture data can be interpreted by considering the similar characteristics of the classes in which the extracted features from the data make overlapping areas in the feature space. In this case, classification task is difficult and the assumption of ambiguity in the labels of the data allows the proposed method to provide better classification.

![Graphs showing classification error rates as a function of label corruption level for different datasets.](image)

**Fig. 10.** Average classification error rates (%) as a function of the label corruption level for the proposed method, ensemble networks constructed by the fixed combination rules and the Rogova’s method on the UIUCTex datasets. (a) Brick, (b) Glass, (c) Carpet, (d) Stone and (e) Fabric.

![Graphs showing average classification error rates as a function of label corruption level for different datasets.](image)

**Fig. 11.** Average classification error rates (%) as a function of the label corruption level for the proposed method, ensemble networks constructed by the fixed combination rules and the Rogova’s method on the Hoda Farsi handwritten digit dataset.

### Table 4

<table>
<thead>
<tr>
<th>Correlation level (%)</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td>-13.54</td>
<td>11.05</td>
<td>43.65</td>
<td>32.65</td>
<td>10.25</td>
</tr>
<tr>
<td>Waveform</td>
<td>15.78</td>
<td>13.61</td>
<td>20.09</td>
<td>15.31</td>
<td>11.89</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>1.1</td>
<td>-2.09</td>
<td>-2.68</td>
<td>-2.85</td>
<td>13.95</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin</td>
<td>6.68</td>
<td>29.79</td>
<td>35.67</td>
<td>13.01</td>
<td>22.20</td>
</tr>
<tr>
<td>Brick</td>
<td>16.50</td>
<td>30.05</td>
<td>37.30</td>
<td>11.66</td>
<td>7.22</td>
</tr>
<tr>
<td>Glass</td>
<td>23.59</td>
<td>34.60</td>
<td>23.25</td>
<td>17.23</td>
<td>15.41</td>
</tr>
<tr>
<td>Carpet</td>
<td>76.19</td>
<td>90.32</td>
<td>80.91</td>
<td>63.69</td>
<td>14.03</td>
</tr>
<tr>
<td>Stone</td>
<td>54.04</td>
<td>63.89</td>
<td>63.29</td>
<td>57.14</td>
<td>29.94</td>
</tr>
<tr>
<td>Fabric</td>
<td>-1.32</td>
<td>12.27</td>
<td>14.87</td>
<td>37.30</td>
<td>15.29</td>
</tr>
<tr>
<td>Farsi handwritten digit</td>
<td>1.32</td>
<td>9.89</td>
<td>10.30</td>
<td>3.25</td>
<td>7.43</td>
</tr>
</tbody>
</table>

The error reduction rates (%) as a function of the label corruption level caused by the proposed method over the Rogova’s method for the real data.

The satisfactory performance of our method in dealing with the texture data can be interpreted by considering the similar characteristics of the classes in which the extracted features from the data make overlapping areas in the feature space. In this case, classification task is difficult and the assumption of ambiguity in the labels of the data allows the proposed method to provide better classification.
tion results than the other ensemble schemes which try to handle the classification problems using the initial labels.

From a different viewpoint, the results of the proposed method can be compared to those of the evidence-based ensemble structure proposed by Rogova. Both classification schemes make use of similar information sources and merge the decisions of their base classifiers by utilizing Dempster’s rule of combination. In Table 4, the improvements caused by the proposed scheme over the Rogova’s method are quantitatively investigated in terms of the error reduction rate. It can be seen that in most cases the proposed method has better classification performance than the Rogova’s method which can be interpreted by considering the role of the proposed re-labeling algorithm in our method. Using this approach, the ambiguity in the supervised information is considered in the learning phase and by allowing each train data to have a soft label comprising any subset of the main classes, more sophisticated information can be encoded by the BBA. On the other hand, the Rogova’s method tries to deal with the ambiguities in the labels of the data only in the decision level and uses BBAs which only composed of the main classes and Ω.

We can conclude the experimental results by focusing on the strength of the proposed approach in handling classification problems with noisy labels in which it is able to outperform the other ensemble schemes especially for classification problems with high levels of label noise. Again, it should be noticed that our method would be helpful if it is applied to a supervised classification problem with ambiguous/noisy labels or when class overlap is considerable.

6. Conclusion

In this paper, a method for handling imperfect labels in the evidence theory and ensemble learning frameworks has been presented. By extracting different types of features from the data, the proposed method takes advantage of information redundancy and complementariness between sources. The initial label of each training sample is ignored and based on its closeness to prototypes of the main classes, it is then reassigned to a class with crisp or soft label. MLP neural network is used as base classifier and its outputs are interpreted as BBA and in this way, partial knowledge about the label. MLP neural network is used as base classifier and its outputs of the main classes, it is then reassigned to a class with crisp or soft label. By extracting different types of features from the data, the ensemble method is constructed by a set of complementary sources of information, a feature space/classifier selection stage was adopted in our method. Experiments carried out on controlled simulated data and 10 real datasets. It has been shown that by considering the uncertainty in the labels of the data, our method can outperform classifiers that rely on the initial imperfect labels and it is more robust against erroneous labels.

References