Signal Processing 148 (2018) 68–77

Contents lists available at ScienceDirect

Signal Processing

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

A hierarchical classification method using belief functions

Daniel Alshamaa[a,*], Farah Mourad Chehade[a], Paul Honeine[b]

[a] Institut Charles Delaunay, ROSAS, Université de Technologie de Troyes, UMR 10300, CNRS, Troyes, France
[b] LITIS Lab, Normandie Université, Université de Rouen, Rouen, France

ARTICLE INFO

Article history:
Received 30 May 2017
Revised 1 February 2018
Accepted 13 February 2018
Available online 15 February 2018

Keywords:
Belief functions
Decision making
Error rate
Hierarchical clustering
Multi-class classification

ABSTRACT

Classification is one of the most important tasks carried out by intelligent systems. Recent works have proposed deep learning to solve the classification problem. While such techniques achieve a very good performance and reduce the complexity of feature engineering, they require a large amount of data and are extremely computationally expensive to train. This paper presents a new supervised confidence-based classification method for multi-class problems. The method is a hierarchical technique using the belief function theory and feature selection. The method predicts, for a new sample input, a confidence-level for each class. For this purpose, a hierarchical clustering approach is adopted to create a two-level classification problem. A feature selection technique is then carried out at each level to reduce the complexity of the algorithm and enhance the classification performance. The belief function theory is then used to combine all information and to give out decisions, by computing the confidence of the sample being in each class. The proposed method has been tested for indoor localization in a wireless sensors network and for facial image recognition using well-known databases. The obtained results prove the effectiveness of the proposed method and its competence as compared to state-of-the-art methods.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

The classification problem is widely tackled in data mining applications. It is stated as follows: given a set of labeled training observations relative to certain features, determine the class label of a new unlabeled data instance. Usually, classification methods include two phases. An offline phase where a model is constructed from training data, and an online phase where a new instance is labeled using the constructed model. The output of such methods is either a discrete label for the new instance, or a numerical score for each class label determining the relative tendency of an instance to belong to different classes. This issue is important in text categorization [1], multimedia applications [2], computer vision [3], medical imaging [4], mobile sensor networks [5], etc.

There are two main types of classification: a flat classification that refers to the standard binary or multi-class methods [6], or hierarchical classification where the classes are classified at each level of a defined dendrogram. Figs. 1 and 2 are examples of such classifications, where $\{a, b, c, d, e, f, g, h\}$ is a set of classes, $C1$, $C2$, and $C3$ are parent nodes, and R is a root node. The parent nodes might be either predefined as a taxonomy or created via hierarchical clustering. In hierarchical models, it is distinguished between local and global classifier approaches. In local classifiers, the hierarchy is taken into account by using a local information perspective. This can be represented by three standard ways: Local classifier per node that trains one binary classifier for each node; local classifier per parent node where, for each parent node, a multi-class classifier is trained to distinguish between its child nodes; and local classifier per level that consists of training one multi-class classifier for each level of the class hierarchy. Although the problem can be tackled using any of the previously described approaches, having a single complex model for all classes reduces the size of the global classification model. This is known as the global classifier approach where one single classification model is built taking into account the whole class hierarchy. The dendrogram is either predefined, or created by means of hierarchical clustering techniques according to similarity metrics.

Although no theoretical evidence or proof whether hierarchical or flat classification models are better [7], experiments throughout previous studies have shown that a better accuracy could be obtained by the former especially for a large number of classes [8,9]. However, a large number of levels in the dendrogram causes slowness in the classification procedure, in addition to the risk of propagating any error in a top level all along the hierarchy [7,9]. In both, flat and hierarchical approaches, classical classification methods such as naive Bayes, neural networks, support vector machines [10–12] can be applied either on the original classes or at each

* Corresponding author.
E-mail addresses: daniel.alshamaa@utt.fr (D. Alshamaa), Farah.chehade@utt.fr (F.M. Chehade), Paul.honeine@univ-rouen.fr (P. Honeine).

https://doi.org/10.1016/j.sigpro.2018.02.021
0165-1684/© 2018 Elsevier B.V. All rights reserved.

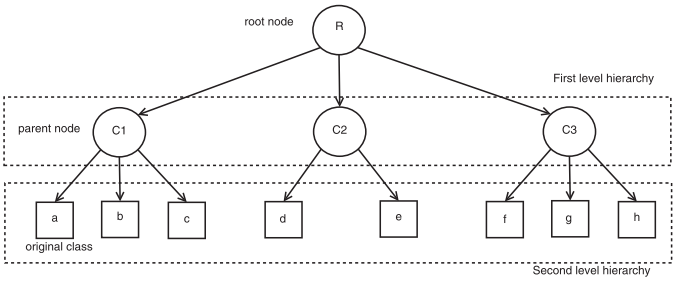

Fig. 1. Hierarchical classification.

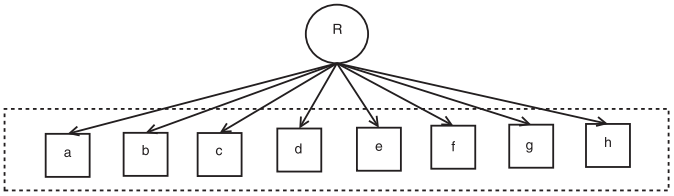

Fig. 2. Flat classification.

level of the hierarchy. Another concept related to the hierarchical approach is the deep learning that uses a cascade of layers for feature extraction and transformation, each layer taking as inputs the outputs of the previous one [13]. Though it has best-in-class performance and reduces the complexity of feature engineering as compared to other solutions in the domain, deep learning requires a large amount of data and is extremely computationally expensive to train. In difference with deep learning, this paper does not tackle feature extraction, rather the work starts once the features are derived. As a consequence, the proposed method can benefit from any feature extraction technique as a preliminary phase, including deep learning.

This paper proposes a confidence-based classification method for multi-class problems. The proposed method is a hierarchical technique, using belief functions and feature selection, described in the following. Given a database of labeled training observations, the classes are merged into clusters using an agglomerative hierarchical clustering method. An optimal level of clustering is selected from the obtained dendrogram, by optimizing the inter- and intra-clusters scatters. The hierarchy is reformed into two levels: the first consisting of the optimal selected clusters, and the second of the original classes in each cluster. Reducing the hierarchy to only two levels decreases considerably the complexity of the method, compared to classic hierarchical methods, with more robustness against error propagation. It also reduces the considered labels at a level, which makes it more efficient than flat techniques. Afterwards, the objective of classification becomes to determine the correct cluster and the correct class at the first and second levels respectively. At each stage, a feature selection technique is applied to choose the best features capable of discriminating between classes and clusters. This creates a framework for the belief function theory (BFT) that associates masses and combines evidence to determine a level of confidence of having the new instance belonging to each class.

The contribution of our work can be summarized as follows. First, the transformation of the problem from a classical flat classification to a two-level hierarchical classification. Second, the feature selection technique. The proposed approach maximizes the discriminative capacity of the ensemble of features at each level of the hierarchy and is consistent with the statistical distributions used to model the observations of the classes. Third, the belief functions framework where masses are assigned to supersets of clusters and classes taking advantage of all available evidence at each level. All assigned masses are then combined to attribute a level of confidence to each original class.

The remainder of the paper is organized as follows. Section 2 is a state-of-the-art that states the problem and defines the concepts needed in the rest of the paper. Section 3 describes the proposed classification approach. Section 4 shows the results of applying the proposed method for facial image recognition and for localization in a wireless sensors network, compared to other well-known state-of-the-art classification techniques. Finally, Section 5 concludes this paper.

2. State-of-the-art

In this section, the classification problem is firstly stated and formulated. Afterwards, the concepts needed in the proposed approach to solve the problem are then introduced.

The proposed classification problem can be formulated as follows. Let

- $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a training dataset of n observations, with $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, p being the number of observations features;
- m be the number of classes and y_i^{cl} denote the class i;
- $L = \{\ell_1, \ldots, \ell_n\}$ be the labels set associated to the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and whose values are taken within $\{y_1^{cl}, \ldots, y_m^{cl}\}$.

The aim of the algorithm is to find a function $\mathbf{h} : \mathbb{R}^p \rightarrow [0, 1]^m$ such that $\mathbf{h}(\mathbf{x}) = (\mathcal{C}f(y_1^{cl}), \ldots, \mathcal{C}f(y_m^{cl}))$, where $\mathcal{C}f(y_i^{cl})$ is the level of confidence of the statement: "$\mathbf{x}$ *belongs to class* y_i^{cl}".

The first step of the proposed approach is merging the classes, once the distributions are defined, using clustering. Clustering aims to organize a set of data into groups called clusters, according to some criteria [14]. Hierarchical clustering builds a hierarchy of clusters or dendrogram driving two strategies: agglomerative or divisive approaches. In the agglomerative or the *bottom up* approach, each observation starts as an independent cluster, and pairs of clusters are merged upon moving up in the hierarchy; whereas in the divisive or *top down* approach, all observations start as one single cluster, and are split upon moving down in the hierarchy [15].

At the end of the developed clustering phase, a two-level hierarchy is obtained. At each level, a feature selection technique is applied to choose the best features. The observations have p components, each one being related to a certain feature, of the set $F = \{f_1, f_2, \ldots, f_p\}$. Feature selection aims at searching for the best subset of the competing $2^p - 1$ candidate subsets of F according to some evaluation function. This can be solved using filter or wrapper method [16]. The filter approach selects feature subsets based on the general characteristics of the data without considering the learning algorithm. Alternatively, the wrapper approach searches for the best subset of features according to an evaluation criterion based on the same learning algorithm. Although the wrapper approach performs better than the filter approach in general, however it is more computationally complex which makes it impractical in many cases [17].

The belief theory, which is also called the DempsterShafer theory or the evidence theory, is a variant of the probability theory where elements are not single points but rather sets or intervals [18]. It is a branch of mathematics that provides an original framework for data fusion based on evidence [19]. In general, the belief function based decision fusion framework mainly includes two phases, mass construction and basic belief assignment (BBA) combination. In the proposed classification method, this is the last step where masses are associated and translated into confidence levels. By taking into account information uncertainty, the proposed method yields several possibilities of classes with different levels of confidence of covering the new observation.

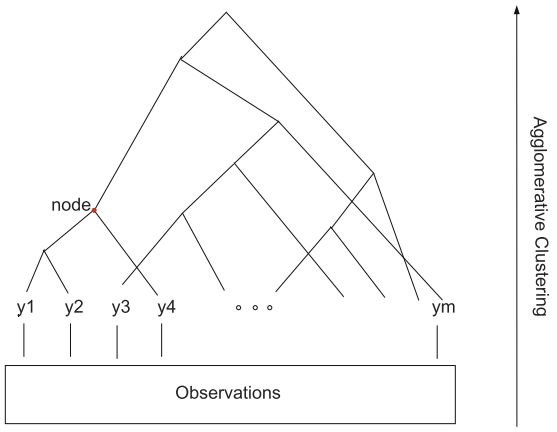

Fig. 3. Hierarchical clustering.

3. Proposed classification method

The proposed classification method consists of the four phases; distribution fitting, clustering, feature selection, and belief functions, described in the following.

3.1. Distribution fitting

Consider a set of o_i observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_{o_i}\}$ labeled at the class y_i^{cl}. The aim of this section is to fit these observations to a distribution Q_i having a probability density function $q_i(\mathbf{x})$ defined over a set of parameters to be estimated with the available data. First, choose the types of distributions to be tested. Then, estimate their parameters using the observations. And finally, apply a statistical goodness of fit test to evaluate their fitting error. The problem is in the form of hypothesis testing where the null and alternative hypotheses are:

H_0: Sample data come from the stated distribution.

H_a: Sample data do not come from the stated distribution.

The Kolmogorov–Smirnov (K-S) test [20] is used to test the hypotheses. An extension to this test in multivariate case is presented in [21]. For each considered distribution, the hypothesis H_0 is rejected at a significance level α if the test statistic is greater than a critical value obtained from the K-S table [22]. The significance level is chosen by convention, and could be set to 0.01, 0.02, or up to 0.05 if available distributions failed to fit with smaller levels. All the considered distributions are tested, and the accepted ones are ranked according to their statistics, the best fitting one being selected. It is noteworthy that the observations of each class could be fitted to a separate distribution.

3.2. Clustering algorithm

3.2.1. Building the dendrogram

To build the dendrogram, an agglomerative strategy is adopted here since it is less complex than the divisive case [23]. This strategy is shown in Fig. 3. To avoid having observations of the same classes in different clusters, the proposed method considers the classes as units. Indeed, a statistical distribution is assigned to each class, by fitting its corresponding observations to one of the existing multivariate distributions. Let $Q_1, \ldots, Q_m$ be the fitted distributions, defined over a set of parameters, of the classes $y_1^{cl}, \ldots, y_m^{cl}$ respectively. Each class is then considered as an independent cluster at the beginning of the algorithm. To merge clusters according to a criterion, the agglomerative hierarchical clustering technique measures the dissimilarity between the clusters. Of these criteria are single-linkage, complete linkage, Ward's minimum variance, etc [24]. Since distributions are being clustered here, statistical measures could be applied like Kullback–Leibler divergence, Hellinger distance, total variation distance, etc [25]. The Kullback–Leibler divergence [26] or relative entropy of two distributions Q_i and Q_j with density functions q_i and q_j of input $\mathbf{x}$ is defined as

$$D_{KL}(Q_i||Q_j) = \int_{\mathbf{x}} \log\left(\frac{q_i(\mathbf{x})}{q_j(\mathbf{x})}\right) q_i(\mathbf{x}) d\mathbf{x}. \tag{1}$$

The relative entropy is asymmetric, always positive and equal to zero when the two distributions are identical. The J-divergence [27] symmetrizes the Kullback–Leibler divergence as follows

$$D_J(Q_i||Q_j) = D_{KL}(Q_i||Q_j) + D_{KL}(Q_j||Q_i). \tag{2}$$

This divergence computes the level of discrepancy or lack of similarity between probability distributions. It is a measure of how different two probability distributions, over the same event space, are [28]. The proposed clustering method employs the J-divergence as the dissimilarity measure to construct the dendrogram. At each iteration, it merges the two clusters whose distributions have the maximal divergence. Merging two clusters means here a merge of all the observations of the infant clusters and a computation of a new distribution according to the new set of observations. By maximizing the divergence, the infant clusters would be dissimilar, which helps the classification process within a cluster. The algorithm is iterated until all the classes are merged into one cluster [29].

3.2.2. Two-level hierarchy

After the dendrogram is created, it should be cut based on the desired number of clusters. However, since there is no prior knowledge regarding this parameter, it is calculated by solving an optimization problem that takes into account both inter- and intra-clusters scatters. Several indices have been proposed to solve this problem [30–32]. A method developed by Fischer [33] finds the optimal number of clusters that maximizes the following quantity:

$$\rho(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right| \tag{3}$$

such that

$$DIFF(k) = (k-1)^{\frac{2}{p}} W(k-1) - (k)^{\frac{2}{p}} W(k), \tag{4}$$

$DIFF(k)$ being defined as the difference between a clustering of the data in k and a clustering in $k-1$ clusters. $W(k)$, the sum of squares function that corresponds to the clustering in k clusters, is equal to:

$$W(k) = \sum_{j=1}^{k} \sum_{\substack{n' \\ \ell_{n'} \in y_j^C}} ||\mathbf{x}_{n'} - \mu_j||^2, \tag{5}$$

μ_j being the mean of the distribution of the cluster j.

Let K^C be the number of clusters that maximizes Eq. (3), namely $K^C = \text{argmax}_k\, \rho(k)$. The quantity $\rho(k)$ represents the ratio between $DIFF(k)$ and $DIFF(k+1)$. In fact, it is shown that if there exists an optimal clustering solution in K^C groups, the value of $DIFF(K^C)$ should be comparably large and positive. In contrast, all values of $DIFF(k)$ for $k > K^C$ will have small values, while values for $k < K^C$ will be rather large and positive [34].

The dendrogram is cut at a certain level where K^C clusters are obtained, denoted by y_j^C with $j \in \{1, \ldots, K^C\}$. All infant clusters of each selected cluster are merged yielding a set of classes for each cluster. This is shown in Fig. 4. In the following, I_j denotes the set of indices of the classes included in the cluster j, that is, $y_i^{cl} \in y_j^C \quad \forall i \in I_j$. The clustering technique is described in Algorithm 1 , where inputs are the observations set S, their labels L

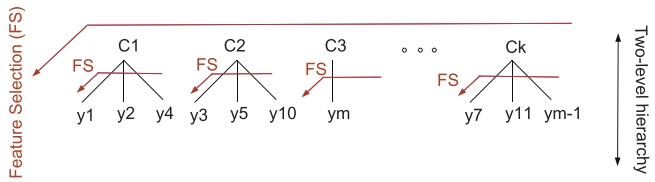

Fig. 4. The two-level hierarchy and feature selection at each level.

Algorithm 1: Clustering technique.

Input : $S, L, \{Q_1, \ldots, Q_m\}$
Output: $K^C, I_1, \ldots, I_{K^C}$

$k = m$;
$\mathbf{D} = \mathbf{0}_{m\times m}$;
for $u \in \{1, \ldots, m\}$ **do**
  for $v \in \{1, \ldots, u-1\}$ **do**
    $\mathbf{D}(u, v) = D_J(Q_u || Q_v)$;
  end
  $Q'_u = Q_u$;
end
$U = \{1, \ldots, m\}$;
while $k > 1$ **do**
  $(u_{\max}, v_{\max}) = \text{argmaxA}_{u,v}\mathbf{D}(u, v)$;
  $Q'_{v_{\max}} = Q'_{u_{\max}} \cup Q'_{v_{\max}}$, delete $u_{\max}$ from U;
  if $|U| > 1$ **then**
    for $\{v \in U, v < u_{\max}\}$ **do**
      $\mathbf{D}(u_{\max}, v) = \mathbf{0}$;
    end
    for $\{u \in U, u > v_{\max}\}$ **do**
      $\mathbf{D}(u, v_{\max}) = D_J(Q'_u || Q'_{v_{\max}})$;
    end
    for $\{v \in U, v < v_{\max}\}$ **do**
      $\mathbf{D}(v_{\max}, v) = D_J(Q'_{v_{\max}} || Q'_v)$;
    end
  end
  Create parent node with child nodes $y^{cl}_{u_{\max}}$ and $y^{cl}_{v_{\max}}$;
  Compute $B(k)$ using equations 3, 4, and 5 ;
  $k = k - 1$;
end
$K^C = \text{argmaxA}B(k)$;
Cut dendrogram at the K^C clusters;
Create $I_1, \ldots, I_{K^C}$ by associating to each cluster all its infant classes without intermediary clusters.

and the distributions $Q_1, \ldots, Q_m$ of the original classes, the outputs are the number of clusters K^C and the sets of indices of the classes in the clusters I_j, $\mathbf{0}_{m\times m}$ denotes the $m \times m$ null matrix, $Q_{u_1} \cup Q_{u_2}$ means the merge of the observations of both distributions and fitting them to a new one, and $|U|$ denotes the cardinal of U.

3.3. Feature selection technique

The feature selection algorithm is applied equivalently at classes of each cluster and between clusters, as shown in Fig. 4. For the sake of simplicity, unique notations for clusters y^C and classes y^{cl} are considered in the following, that is, let y denote either a cluster or a class within a cluster, and let K denote their numbers. A greedy filter feature selection method is adopted to maximize the discriminative capacity of the selected features. Though it might not reach global optima, rather fall in a local one, the greedy search algorithm is simple and easy to compute, especially that it will be applied at the clusters level, and at the classes level of each cluster. Gibbs sampling [35] could be used instead of the greedy search and aim to converge to a stationary distribution representing the best subset of features. It indirectly samples from the posterior distribution on the set of possible subset choices. Those subsets with higher probability can then be identified by their more frequent appearance in the Gibbs sample. A similar approach has been presented in [17] for naive Bayes classification, where each feature is given a score and ranked in a descent order. However, features cannot be treated independently, since a feature that might be useless by itself can provide a significance improvement in the performance when taken with others [36]. Hence, it is important to select the subset of features that, when taken all together, maximizes the discrimination between the sets to be classified. To do this, all the nonempty subsets of F are considered. Let $F' \subseteq F$ denote one considered subset. All the observations at the features of F' belonging to each entity y_j are thus taken, and they are fitted to a distribution denoted $Q_{F',j}$, $j \in \{1, \ldots, K\}$. The distribution $Q_{F',j}$ is either univariate or multivariate depending on the cardinal of F'. The farther the distributions $Q_{F',1}, \ldots, Q_{F',K}$ are one from the other, the more discriminative the feature subset F' is. Indeed, this reduces the overlapping between the distributions and thus decreases the ambiguity in discriminating between them. The Kullback–Leibler divergence is a metric that could be used to measure such a quantity. The discriminative capacity of a subset of features $F' \subseteq F$ is then defined as follows,

$$DisC(F') = \sum_{u=1}^{K}\sum_{v=1}^{K} D_{KL}(Q_{F',u} || Q_{F',v}), \tag{6}$$

$D_{KL}(Q_{F',u} || Q_{F',v})$ being the Kullback–Leibler divergence measured between the distributions of the observations belonging to entities y_u and y_v, while considering only the features of F'. The objective of the feature selection technique is to find the subset F_s such that $DisC(F_s)$ is maximum. A greedy search algorithm with backward elimination strategy is applied to choose this subset. One starts with the whole set of features and progressively eliminates the least promising feature, whose elimination maximizes the increase of the discriminative capacity of the set. This process is iterated until the discriminative capacity of the set is no more improved. Applying this technique at each level of the two-level hierarchy leads on one hand, the optimal subset of features that is best to distinguish between the clusters and, on the other hand, sets of features that should be used for classification between the classes within each cluster. The feature selection technique is described in Algorithm 2.

Algorithm 2: Feature selection technique.

Input : $F = \{f_1, \ldots, f_p\}$
Output: F_s

$F' = F$;
while $F' \neq \phi$ **do**
  for $j \in \{1, \ldots, |F'|\}$ **do**
    $X = F'$;
    $T_j = F' \setminus \{f_j\}$;
    $DisC(T_j) = \sum_{u=1}^{K}\sum_{v=1}^{K} D_{KL}(Q_{T_j,u}, Q_{T_j,v})$;
  end
  $F' = \text{argmaxA}_{T_j} DisC(T_j)$;
  if $DisC(F') < DisC(X)$ **then**
    $F_s = X$;
    return F_s and quit algorithm;
  end
  $F_s = F'$;
end

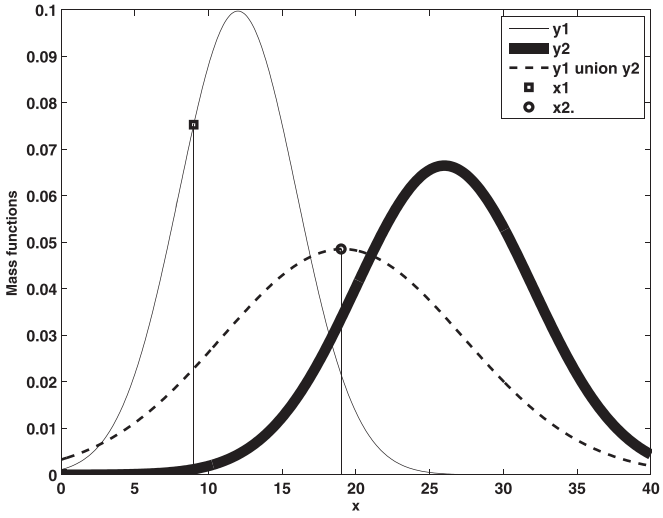

Fig. 5. Mass assignments of an observation.

3.4. Belief functions framework

3.4.1. Mass assignments

Let y be a discrete variable taking values in $Y = \{y_1, \ldots, y_K\}$ and let 2^Y be the set of all the supersets of Y, i.e, $2^Y = \{\emptyset, \{y_1\}, \ldots, Y\}$. The cardinal of 2^Y is equal to $2^{|Y|} = 2^K$, where $|Y|$ denotes the cardinal of Y. One fundamental function of the BFT is the mass function, also called the basic belief assignment (BBA). A mass function $m_f(\cdot)$ is a mapping from 2^Y to the interval [0, 1], defined according to a certain information source f. It satisfies:

$$\sum_{A \in 2^Y} m_f(A) = 1. \tag{7}$$

The mass $m_f(A)$ given to $A \in 2^Y$ stands for the proportion of evidence, brought by the source f, saying that the observed variable belongs to A. In the following, y denotes a cluster y^C or a class y^{cl} within a cluster depending on the level of the hierarchy where the computations are made, F_s denotes the set of selected features either at the classes or the clusters levels and the information source f denotes one selected feature of F_s.

In order to define the features BBAs, all observations related to each selected feature belonging to a set $A \in 2^Y$ are fitted to a distribution $Q_{f,A}$. Then, having an observation x_f related to a feature $f \in F_s$, the mass $m_f(A)$ is calculated as follows,

$$m_f(A) = \frac{Q_{f,A}(x_f)}{\sum_{A' \in 2^Y, A' \neq \emptyset} Q_{f,A'}(x_f)}, \quad A \in 2^Y, A \neq \emptyset. \tag{8}$$

The quantity $m_f(A)$ represents the amount of evidence brought by the feature f saying that the observation x_f belongs to the set A, A being a singleton, a pair, or more. By taking all the supersets of Y and not only the singletons, the proposed algorithm uses all available pieces of evidence, even if they are uncertain about a single element. Note that $m_f(A)$ is not the probability of having x_f in A, but only an interpretation of the information brought by the feature f by means of observation x_f, that is, $m_f(A)$ could be higher than $m_f(B)$ even if $A \subset B$. Indeed, consider the example of Fig. 5, where $Y = \{y_1, y_2\}$. The set Y has three non-empty supersets $\{y_1\}$, $\{y_2\}$ and $\{y_1, y_2\}$, represented by their three distributions in Fig. 5. An observation $x^{(1)}$ for instance is more likely to be of the entity y_1 and indeed the mass of $\{y_1\}$ is higher than those of $\{y_2\}$ and $\{y_1, y_2\}$ using Eq. (8). However, for observation $x^{(2)}$, the distributions of y_1 and y_2 are too close. By taking the superset $\{y_1, y_2\}$ within the possibilities, a higher evidence is associated to $\{y_1, y_2\}$, instead of taking a risk in setting more evidence to $\{y_1\}$ or $\{y_2\}$. The evidence assigned to $\{y_1, y_2\}$ is higher than those of singletons only for observations where the distributions of singletons are two close, to avoid erroneous assignments, and to take advantage even of uncertain data. This example illustrates clearly the effectiveness of the use of other supersets than singletons and motivates the use of the belief function theory. Note that instead of taking all the supersets, one can only take singletons and pairs for example, in order to reduce the complexity of the algorithm. One can also consider the supersets whose distributions are higher than the others at some observations. A set whose distribution is flat with a high standard deviation is non-informative and it could be eliminated from the considered set, to reduce the complexity.

3.4.2. Discounting operation

The features selected at the features selection phase are not completely reliable. Indeed, each feature could yield an erroneous attribution of evidence for some observations. In order to correct this, one can discount the BBAs of Eq. (8) by taking into account the error rate of the feature. The discounted BBA ${}^{\alpha}m_f$ of a feature f having an error rate α_f is deduced from the BBA m_f as follows [37],

$${}^{\alpha}m_f(A) = \begin{cases} (1-\alpha_f)m_f(A), & \text{if } A \in 2^Y,\ A \neq Y; \\ \alpha_f + (1-\alpha_f)m_f(A), & \text{if } A = Y; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

By doing this, the amounts of evidence given to the supersets of Y are reduced, and the remaining evidence is given to the whole set Y.

Now, to compute the error rate of a selected feature f, consider an observation x_f being truly in A. The feature f is assumed not reliable if, according to x_f, it associates more evidence to any set other than A, that is, the mass associated to A is less than the mass of another set of 2^Y. Since the BBAs are defined using the probability distributions related to each set, then a feature is erroneous for all observations of A where $Q_{f,A}(x_f)$ is less than any $Q_{f,A'}(x_f)$, for any $A' \neq A$. Let $\epsilon_f(A)$ be the error rate related to the set A with respect to the feature f. Then,

$$\epsilon_f(A) = \int_{\mathbb{D}_{f,A}} Q_{f,A}(x)dx, \tag{10}$$

such that $\mathbb{D}_{f,A}$ is the domain of error of set A according to f, defined as follows,

$$\mathbb{D}_{f,A} = \{x \mid Q_{f,A}(x) \leq \max_{A' \in 2^Y, A' \neq A} (Q_{f,A'}(x))\}. \tag{11}$$

The error rate α_f of a feature f is then the average error of all sets according to this feature, namely

$$\alpha_f = \frac{\sum_{A \in 2^Y} \epsilon_f(A)}{2^{|Y|}}. \tag{12}$$

3.4.3. Combining evidence

According to the information retrieved from the features, mass functions ${}^{\alpha}m_f(\cdot)$ are defined at the clusters levels and at the classes levels within each cluster. Combining the evidence consists of aggregating the information coming from all the features at a given level [38]. By using the discounted mass functions, the information is now reliable. The mass functions can then be combined using the conjunctive rule of combination as follows,

$$m_{\cap}(A) = \sum_{\substack{A^{(f_c)} \in 2^Y \\ \cap_{f_c \in F_s} A^{(f_c)} = A}} {}^{\alpha}m_{f_1}(A^{(f_1)}) \times \ldots \times {}^{\alpha}m_{f_{|F_s|}}(A^{(f_{|F_s|})}), \tag{13}$$

for all the sets $A \in 2^Y$, with $A^{(f)}$ is the set A with respect to feature f, $|F_s|$ being the cardinal of F_s. This combination rule leads to a more informative and specialized mass function [39]. The mass function is then normalized, leading to the Dempster rule of combination. These computations are applied at the two levels of the hierarchy,

which yield a final normalized mass function $m^C_\cap(\cdot)$ that works on the sets of $\{y^C_1, \ldots, y^C_{K^C}\}$ and also other functions $m^{cl,j}_\cap(\cdot)$ that work on the sets of $\{y^{cl}_i, i \in I_j\}$, with $j \in \{1, \ldots, K^C\}$.

3.4.4. Decision making using the belief function theory

An adequate notion of the BFT to make the decision is the pignistic level [40]. It is defined as follows,

$$BetP(A) = \sum_{A \subseteq A'} \frac{m_\cap(A')}{|A'|}, \tag{14}$$

where A is a singleton of 2^Y. The pignistic level is equivalent to the probability of having the observation belonging to the considered set. One could also compute the pignistic level of higher-cardinal supersets. However, only the singleton sets are taken into consideration, as we are interested in determining a level of confidence for the original classes only. Eq. (14) is applied at the clusters level and the classes level within each cluster, leading respectively to $BetP^C(\{y^C_j\})$, $j \in \{1, \ldots, K^C\}$, and $BetP^{cl,j}(\{y^{cl}_i\})$, $i \in I_j$. Finally, to make a decision between the original classes, pignistic levels of classes and clusters are combined leading to a confidence of each original class as follows,

$$\mathcal{C}f(y^{cl}_i) = BetP^C(\{y^C_j\}) \times BetP^{cl,j}(\{y^{cl}_i\}), \; i \in I_j, \; j \in \{1, \ldots, K^C\}. \tag{15}$$

The class having the highest confidence is then selected. It is worth noting that this method yields ranked results. As a consequence, the resulting ranked classes can be used whenever required by the task under scrutiny.

To show the importance of the belief based approach, we consider the following example. Suppose we have three classes A, B, and C. If we had, for a certain new observation belonging in reality to class B, evidence as follows, $m(A) = 0.25$, $m(B) = 0.2$, $m(C) = 0.02$, $m(A, B) = 0.2$, $m(A, C) = 0.02$, $m(B, C) = 0.3$, and $m(A, B, C) = 0.01$. If we depend on the masses only, we choose class A because $m(A) > m(B)$, while considering all associated evidence and using the pignistic transformation, the following is obtained, $\mathcal{C}f(A) = 0.363$, $\mathcal{C}f(B) = 0.453$, and $\mathcal{C}f(C) = 0.183$. As we can see, the confidence level of class B is now greater, and hence class B will be decided. Hence, using a pignistic transformation to pass from the attributed masses to the confidence level through considering all evidence assigned to all supersets of classes in the belief functions framework enhances the classification.

4. Experiments

4.1. Facial image recognition

Facial image recognition has gained a great attention in the recent years due to its wide applications in video surveillance, database image matching, security measurements, etc. The Extended Yale B [41], the ORL [42], and the AR [43] face databases were used to evaluate the proposed classification method. Many state-of-the-art methods have been targeting these databases to solve the facial recognition problem. In order to study the influence of the classification (recognition) method only, the same data partition and feature extraction technique of each state-of-the-art method were taken.

The Extended Yale B database consists of 2414 frontal-face images for 38 individuals taken under various lighting conditions. Yang et al. [44] investigated the use of Gabor features for sparse representation based classification (SRC) with a learned Gabor occlusion dictionary. The authors randomly selected half of the images for training (32 images per subject) and used the other half for testing. All images were normalized to 192 × 168. They demonstrate the results of the method versus the feature dimension while comparing with well-known classification techniques SVM, nearest neighbor (NN), and linear regression classification (LRC). Khorsandi and Abdel-Mottaleb [45] presented a classification method based on a weighted sparse representation. They also cropped and normalized the images to 192 × 168, and used half of the images for training and the others for testing. The method used the mutual information between the query sample and the training samples to give a weight for the latter in each class in the dictionary. We compared the results of these methods with those given by our proposed method using the same data portion and the same feature extraction technique. The results are presented in Tables 1 and 2. The numbers in bold refer to the best classification method for a feature extraction method and a feature dimension.

Table 1
Average face recognition accuracy (%) on the extended Yale B database based on the Gabor feature robust representation. Number of classes = 38, number of training data = 32, number of test data = 32.

Method	Feature dimension			
	56	**120**	**300**	**504**
NN	81.4	89.2	91.9	92.0
SRC	92.6	95.6	97.4	97.9
Yang and Zhang	92.7	95.6	97.9	**99.0**
SVM	92.6	95.3	96.3	96.4
HSVM	92.9	96.1	95.7	97.2
LRC	**94.1**	94.7	95.4	95.7
Random forests	90.4	90.6	92.6	93.5
Proposed method	93.9	**96.2**	**98.3**	98.7

Table 2
Average face recognition accuracy (%) on the extended Yale B database based on weighted sparse representation. Number of classes = 38, number of training data = 32, number of test data = 32.

Method	Feature dimension			
	30	**56**	**120**	**504**
NN	69.3	72.8	78.5	79.5
NS	79.6	84.1	88.7	90.8
SRC	75.7	84.8	93.9	96.8
HSVM	73.5	78.2	81.8	84.8
Random forests	76.7	80.3	84.1	87.6
Khorsandi et al.	78.5	86.7	95.3	**97.9**
Proposed method	**83.1**	**87.8**	**95.8**	96.9

Table 3
Average face recognition accuracy (%) on the AR database based on grayscale features.

Method	Accuracy
SVM	68.10
LRC	68.75
SRC	63.87
CRC	68.25
Huange et al.	77.14
HSVM	65.23
Random forests	67.76
Proposed method	**82.64**

The AR face dataset consists of more than 4000 images of 126 distinct subjects. Following the recent work of Huang et al. [46], a subset of 1680 images for 120 subjects was constructed, where each image is 50×40 pixels. The authors proposed a class specific sparse representation-based classifier which incorporates the class information in the learning process. The method defined classes as groups that compete to represent the test sample. It considered L_1 and L_2 norm constraints to the classes and samples and was solved by convex optimization. Table 3 shows the results of different classification approaches.

Table 4
Average face recognition accuracy (%) on the ORL database.

Method	Number of training images			
	2	**4**	**6**	**8**
HSVM	90.8	92.6	93.6	94.7
Random forests	90.2	91.3	94.8	95.3
Wang et al.	**95.6**	96.7	97.3	98.4
Proposed method	95.2	**96.8**	**97.8**	**99.3**

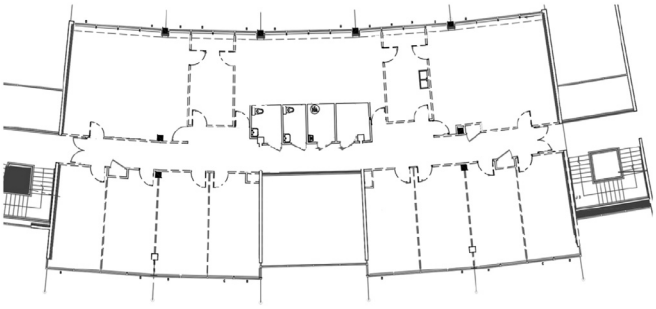

Fig. 6. The first floor of the statistical and operational research department of the University of Technology of Troyes.

The ORL database consists of 400 images for 40 subjects. Wang and Sun [47] proposed a multiple kernel local Fisher discriminant analysis for face recognition. The authors presented a method that searches for maximum discrimination between inter- and intra-classes scatters, producing nonlinear discriminant features with multiple base kernels. They selected different numbers of images per individual to form the training set, and the rest for test. The experiments were repeated 50 times and the average recognition accuracy was computed. All images were cropped and resized to 32×32 pixels, with 256 Gy levels per pixel. Each image is represented by a 1024-dimensional vector in the image space. The comparison results are shown in Table 4.

The results in the aforementioned tables show the competence of the proposed method as compared to other well-known classification techniques in the domain of facial image recognition, outperforming them in almost all the cases.

4.2. Wireless sensors network

To test the efficiency of the developed method, it was applied in the domain of wireless sensor networks [48]. The objective is to track, in real time, any mobile sensor in indoor environments. The WiFi technology was adopted since it is ubiquitous process. However, it suffers from high variations in the power of its signals which magnifies the complexity of the algorithm to obtain high localization accuracy. While most schemes have been focusing on determining the exact position of these sensors [49], a zoning-based localization technique was applied here. The objective is to determine the zone where the mobile sensor resides which orients the issue to a multi-class classification problem.

Real experiments are realized in a WLAN environment at the first floor of the statistical and operational research department of the University of Technology of Troyes, France. As shown in Fig. 6, the considered floor of approximated area of 500 m^2 is partitioned into fifteen zones from both sides of a corridor, that we divided into three zones according to its architecture leading to eighteen zones in total. A personal computer,[1] with a WiFi scanner software,[2] can distinguish Access Points (APs) of the network throughout their MAC addresses. It measures then the Received Signal Strength Indicators (RSSIs) of their transmitted signals. Note that several APs could be detected at the considered area. Only the APs located in the ground, first and second floors of the building to which belongs the studied area are considered. This leads to six of the total APs installed in the network. Sets of 30 measurements are taken in each zone in random positions and orientations of the personal computer and used to construct the databases and train the classifiers. The code is implemented in Matlab environment. A new set of 20 measurements in each zone were taken after a month to test the proposed method, as measurements from the same day are strongly dependent.

Table 5
Parameters of the experiment as obtained by the method.

Cluster number	1	2	3	4	5	6	7
Number of classes per cluster	2	4	2	3	2	2	3
Number of features per cluster	6	6	5	5	5	6	5

The RSSIs of the database are statistically modeled to be used in the clustering phase. For each zone, the normal function was ranked first according to a significance level of 0.02. A dendrogram of the original eighteen classes is created and cut at seven clusters level by optimizing the intra-inter cluster distances. The observations of each cluster are fitted to a generalized extreme value function with the same significance level. The seven clusters with the initial classes constituting them are taken, forming a two-level classification problem. At each level, the feature selection technique is applied to determine the most discriminating features. Five APs were chosen at the first level, while five or six APs were chosen at the second level depending on each cluster. Table 5 shows the details of the experiment. The method generates then a set of confidence levels of the classes, that are the confidence levels of having the mobile sensor residing in each zone.

Table 6 shows the results of applying the developed method to 360 test points and the influence of each phase over the percentage of accuracy and the processing time. The offline training time was considered as computationally complex training algorithms are not preferred. An estimation is said to be correct if the algorithm assigns the highest confidence level to the right zone. As this table clearly shows, just modeling the data with associating and combining masses lead to an accuracy of 78.61%. This low accuracy is due to the wide overlapping of the various functions representing the distributions of the data in the different zones. However, when the two-level hierarchical clustering and classification phase was carried out, a significant enhancement in the percentage of accuracy was noted (83.88%). This amelioration is at the expense of the processing time. It is clear that the training and test times were approximately doubled. In addition, the feature selection phase had a significant impact on the overall process. An accuracy of 86.38% could be obtained with a slight gain in the online test time, yet with an increase in the training time. Moreover, discounting the BBAs of the sources of information based on the features' error rates raised the overall accuracy to 87.77% without a huge impact on the processing time. One advantage of the confidence-based approach is that it allows a second choice in case of an erroneous estimation, by choosing the zone having the second highest level of confidence. In our case, 7.22% of the erroneous estimations were recovered by this second choice. One more advantage can be examined when combining another confidence-based method related to the problem itself to make the final decision. It is to note that the software scans the network and mea-

[1] 4 CPUs, 2.10 GHz.

[2] Wi-Fi Scanner by Lizard Systems.

Table 6
Influence of each phase of the developed method on the accuracy and the processing time.

Applying the proposed method	Accuracy (%)	training time (s)	test time (s)
As it is	87.77	182	0.2508
Without discounting of sources' reliability	86.38	176	0.2184
Without feature selection and discounting	83.88	126	0.2417
Without clustering, feature selection, and discounting	78.61	67	0.1250

Table 7
Comparison between classification methods in terms of accuracy and processing time, over a uniformly and non-uniformly distributed data.

Method	Uniformly distributed data			Non-uniformly distributed data		
	Accuracy (%)	**Training time (s)**	**Test time (s)**	**Accuracy (%)**	**Training time (s)**	**Test time (s)**
K-nearest neighbors	83.33	**16**	0.1289	83.88	**17**	0.1308
Naive Bayes	81.66	91	**0.1018**	80.55	89	**0.1126**
MLR	82.78	122	0.1498	82.50	124	0.1514
Neural networks	84.72	135	0.1866	85.27	131	0.1962
SVM	85.55	143	0.1859	85.83	146	0.1884
Random forests	86.66	224	0.4077	87.22	219	0.4222
HSVM	86.38	291	0.4667	86.66	286	0.4394
Proposed method	**87.77**	182	0.2508	**87.50**	188	0.2637

sures the RSSIs at a 0.75 second intervals, which is enough for the mobile sensor to execute the proposed method.

In this paragraph, the proposed method is compared to some of the well-known classification techniques. Flat classification methods such as k-nearest neighbors, naive Bayes, multinomial logistic regression (MLR), neural networks, and support vector machines (SVM) were considered. A 10-folded cross validation was used on the training database to train the classifiers and tune their parameters. The parameters which minimized the average error of all folds were considered. This aims to enhance the ability of the classifiers to generalize, with a better classification accuracy on the new data. For k-nearest neighbors, the optimal number of neighbors used to estimate the class membership was found to be 23. For naive Bayes and MLR, the maximum likelihood estimate was used to evaluate the probability of having the data instance belong to each class. As for neural networks, radial basis functions were used as activation functions for a one single hidden layer. A Gaussian kernel was used for SVM.

Moreover, the proposed method is compared to hierarchical methods such hierarchical support vector machines (HSVM) and random forests. Chen et al. [50] developed an HSVM technique that solves a series of max-cut problems to recursively partition the classes into two-subsets, till pure leaf nodes that have only one class are obtained. Then, the classical SVM is applied to solve the binary two-subsets classification problem at each internal node. In addition, a random forest model has been proposed in [51] to localize sensors in indoor networks. Random forests [52] is an ensemble of trees, obtained both by bootstrap sampling, and by randomly changing the feature set during learning. More precisely, at each node in the decision tree, a random subset of the input attributes is taken, and the best feature is selected from this subset instead of the set of all attributes. The authors propose a straightforward random forest model and another modified one by defining a localization model for each access point that predicts the localization only when a signal is detected from that access point.

Two configurations are studied, where in the first, the measurements are collected in a uniformly distributed way over the targeted area, while in the second, training instances were acquired on half of the zone and duplicated to create a skewness in the fitting distribution. Table 7 shows the percentage of accuracy and the processing time of the proposed technique compared to these described methods. As this table shows, the proposed method outperforms all the other ones in terms of classification accuracy. On the other hand, its processing time is considered to be competitive to the others, yet with a clear advantage of naive Bayes for instance, and k-nearest neighbors that has no training phase where the indicated time is just for storing the training data and calculating the optimal k by a ten-fold folded cross validation.

In addition, since levels of confidence are being assigned to classes, it is interesting to consider a measure to evaluate the proposed method in these terms and compare it to other techniques. A utility function $L^{\gamma} : [0, 1]^m \to \mathbb{R}$ is used to measure the distribution of levels of confidence on classes for a new observation γ, whose true class is y_i^{cl}. It is defined as follows,

$$L^{\gamma} = \sum_{j=1}^{m} p_j \times Cf(y_j^{cl}), \tag{16}$$

where p_j is a weight assigned to each class, $Cf(y_j^{cl})$ is the confidence level assigned by the method to class y_j^{cl} for a new measurement γ. This utility function is computed for a set of Γ observations and the average is recorded to determine the utility of the applied method. Since there is no difference between an erroneous estimation between a class and another, all weights p_j, $j \neq i$ are set to zero, and hence the utility function becomes

$$\mathcal{L} = \frac{\sum_{\gamma=1}^{\Gamma} L^{\gamma}}{\Gamma}. \tag{17}$$

A set of 10 measurements in each class were taken, and evaluated by all the classification methods indicated above. A probabilistic version of the originally categorical methods was considered to be able to calculate the utility function. The confidence function was replaced by the output probability. As the utility function increases for a certain method, this means that the latter is assigning the highest confidence or probability to correct class. The utility of the k-nearest neighbors was found to be $L_{KNN} = 0.6177$, of naive Bayes $L_{NB} = 0.5781$, of multinomial logistic regression $L_{MLR} = 0.6593$, of neural networks $L_{NN} = 0.7668$, of SVM $L_{SVM} = 0.6891$, of random forest $L_{RF} = 0.7103$, of HSVM $L_{HSVM} = 0.7245$, and of the proposed method $L_{prop} = 0.7258$. This indicates that the proposed method is assigning, in average, a confidence of 0.7258 for correct estimations.

5. Conclusion and future work

This paper presented a new confidence-based classification technique for multi-class problems. The main contributions of the

paper can be described by the following points. First, a two-level hierarchy was created having on the advantages of hierarchical clustering using only two levels, and hence with no need to huge dendrograms and complex computations. Second, the method performed feature selection that not only serves to reduce the complexity of the algorithm, but also ameliorated the execution time and the classification accuracy by maximizing the discriminative capacity of the ensemble of features. At last, the problem was solved in the belief functions framework created to discount the reliability of each feature according to its error rate, and then combining evidence and associating levels of confidence of having the new instance belongs to each one of the different classes. The proposed method was applied for facial image recognition using the extended Yale B, the ORL, and the AR databases. It was also applied in the domain of wireless sensor networks to localize, in real time, a mobile sensor in indoor environments. The results showed that the proposed method outperforms a set of well-known classification algorithms in terms of classification accuracy, yet with a competitive offline and online processing time. One drawback of the proposed method is that it requires a parametric distribution of the data. This assumption might not be true in certain cases where the observations cannot be considered to be from any parametric distribution. Although this can be solved by increasing the number of observations, adopting an approximation, or using a non-parametric distribution, yet this complicates the algorithms and might not be practical and applicable in certain cases. As a future work, focus will be on observation modeling to overcome the distribution problem. On the other hand, research must be carried on to optimize the number of levels to be considered in the hierarchical clustering. The use of non-parametric distributions such as kernel density estimation will also be studied.

Acknowledgment

The authors would like to thank the FEDER (Fonds européen de développement économique et régional) and the Grand Est Region in France for funding this work.

References

[1] S. Kumar, X. Gao, I. Welch, M. Mansoori, A machine learning based web spam filtering approach, in: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2016, pp. 973–980.
[2] Y. Liu, X. Feng, Z. Zhou, Multimodal video classification with stacked contractive autoencoders, Signal Process. 120 (2016) 761–766.
[3] F.L.C. dos Santos, M. Paci, L. Nanni, S. Brahnam, J. Hyttinen, Computer vision for virus image classification, Biosyst. Eng. 138 (2015) 11–22.
[4] I. Dimitrovski, D. Kocev, I. Kitanovski, S. Loskovska, S. Džeroski, Improved medical image modality classification using a combination of visual and textual features, Computerized Medical Imag. Graphics 39 (2015) 14–26.
[5] D.A. Tran, C. Pham, Fast and accurate indoor localization based on spatially hierarchical classification, in: 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, IEEE, 2014, pp. 118–126.
[6] P. Honeine, Z. Noumir, C. Richard, Multiclass classification machines with the complexity of a single binary classifier, Signal Process. 93 (5) (2013) pp.1013–1026.
[7] R. Babbar, I. Partalas, E. Gaussier, M.-R. Amini, On flat versus hierarchical classification in large-scale taxonomies, in: Advances in Neural Information Processing Systems, 2013, pp. 1824–1832.
[8] C.N. Silla Jr, A.A. Freitas, A survey of hierarchical classification across different application domains, Data Min. Knowl. Discov. 22 (1–2) (2011) pp.31–72.
[9] S. Dumais, H. Chen, Hierarchical classification of web content, in: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2000, pp. 256–263.
[10] M.-H. Cheng, K.-S. Hwang, J.-H. Jeng, N.-W. Lin, Classification-based video super-resolution using artificial neural networks, Signal Process. 93 (9) (2013) pp.2612–2625.
[11] S. Beniwal, J. Arora, Classification and feature selection techniques in data mining, International J. Eng. Res.Technol. (IJERT) 1 (6) (2012).
[12] L. Liu, L. Shao, P. Rockett, Human action recognition based on boosted feature selection and naive bayes nearest-neighbor classification, Signal Process. 93 (6) (2013) pp.1521–1530.
[13] I. Chaturvedi, Y.-S. Ong, R.V. Arumugam, Deep transfer learning for classification of time-delayed gaussian networks, Signal Process. 110 (2015) pp.250–262.
[14] M. Shahbaba, S. Beheshti, Mace-means clustering, Signal Process. 105 (2014) pp.216–225.
[15] K. Sasirekha, P. Baby, Agglomerative hierarchical clustering algorithm-a, Int. J. Sci. Res.Publications (2013) 83.
[16] S. Wang, X. Chang, X. Li, Q.Z. Sheng, W. Chen, Multi-task support vector machines for feature selection with shared knowledge discovery, Signal Process. 120 (2016) pp.746–753.
[17] B. Tang, S. Kay, H. He, Toward optimal feature selection in naive bayes for text categorization, IEEE Trans. Knowl. Data Eng. 28 (9) (2016) pp.2508–2521.
[18] L. Liu, R. Yager, Classic works of the Dempster–Shafer theory of belief functions: an introduction, in: Classic works of the Dempster–Shafer theory of belief functions, 2008, pp. 1–34.
[19] W. Zhang, Z. Zhang, Belief function based decision fusion for decentralized target classification in wireless sensor networks, Sensors 15 (8) (2015) pp.20524–20540.
[20] F.J. Massey Jr, The Kolmogorov–Smirnov test for goodness of fit, J. Am. Stat. Assoc. 46 (253) (1951) pp.68–78.
[21] A. Justel, D. Peña, R. Zamar, A multivariate Kolmogorov–Smirnov test of goodness of fit, Stat. Probability Lett. 35 (3) (1997) pp.251–259.
[22] S. Facchinetti, A procedure to find exact critical values of Kolmogorov–Smirnov test, Statistica Applicata (21) (2009) pp.337–359.
[23] C.D. Manning, P. Raghavan, H. Schütze, et al., Introduction to information retrieval, Cambridge university press Cambridge, 2008.
[24] F. Murtagh, P. Contreras, Algorithms for hierarchical clustering: an overview, Wiley Interdiscip. Rev. 2 (1) (2012) pp.86–97.
[25] M. Basseville, Divergence measures for statistical data processing, an annotated bibliography, Signal Process. 93 (4) (2013) pp.621–633.
[26] J. Harmouche, C. Delpha, D. Diallo, Incipient fault amplitude estimation using KL divergence with a probabilistic approach, Signal Process. 120 (2016) pp.1–7.
[27] F. Nielsen, A family of statistical symmetric divergences based on Jensen's inequality, arXiv preprint arXiv:1009.4004 (2010).
[28] C. Röver, T. Friede, Discrete approximation of a mixture distribution via restricted divergence, Jo. Comput. Graphi. Stat. 26 (1) (2017) pp.217–222.
[29] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, 20, Siam, 2007.
[30] L. Rokach, O. Maimon, Clustering Methods, in: Data mining and knowledge discovery handbook, Springer, 2005, pp. 321–352.
[31] M.A. Islam, B.Z. Alizadeh, E.R. van den Heuvel, R. Bruggeman, W. Cahn, L. de Haan, R.S. Kahn, C. Meijer, I. Myin-Germeys, J. van Os, et al., A comparison of indices for identifying the number of clusters in hierarchical clustering: a study on cognition in schizophrenia patients, Commun. Stat. 1 (2) (2015) pp.98–113.
[32] L. Galluccio, O. Michel, P. Comon, A.O. Hero, Graph based k-means clustering, Signal Process. 92 (9) (2012) pp.1970–1984.
[33] A. Fischer, On the number of groups in clustering, Stat. Probability Lett. 81 (12) (2011) pp.1771–1781.
[34] W.J. Krzanowski, Y. Lai, A criterion for determining the number of groups in a data set using sum-of-squares clustering, Biometrics (1988) 23–34.
[35] S.H. Cheung, S. Bansal, A new gibbs sampling based algorithm for bayesian model updating with incomplete complex modal data, Mech. Syst. Signal Process. 92 (2017) 156–172.
[36] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) pp.1157–1182.
[37] D. Mercier, É. Lefèvre, F. Delmotte, Belief functions contextual discounting and canonical decompositions, Int. J. Approx. Reason. 53 (2) (2012) pp.146–158.
[38] M. Kurdej, V. Cherfaoui, Conservative, proportional and optimistic contextual discounting in the belief functions theory, in: Information Fusion (FUSION), 2013 16th International Conference on, IEEE, 2013, pp. 2012–2018.
[39] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.
[40] P. Smets, Belief functions: the disjunctive rule of combination and the generalized bayesian theorem, Int. J. Approx. Reason. 9 (1) (1993) pp.1–35.
[41] K.-C. Lee, J. Ho, D.J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) pp.684–698.
[42] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on, IEEE, 1994, pp. 138–142.
[43] A.M. Martinez, The ar face database, CVC technical report, 1998.
[44] M. Yang, L. Zhang, S.C. Shiu, D. Zhang, Gabor feature based robust representation and classification for face recognition with gabor occlusion dictionary, Pattern Recognit. 46 (7) (2013) pp.1865–1878.
[45] R. Khorsandi, M. Abdel-Mottaleb, Classification based on weighted sparse representation using smoothed L^0 norm with non-negative coefficients, in: Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015, pp. 3131–3135.
[46] S. Huang, Y. Yang, D. Yang, L. Huangfu, X. Zhang, Class specific sparse representation for classification, Signal Process. 116 (2015) pp.38–42.
[47] Z. Wang, X. Sun, Multiple kernel local fisher discriminant analysis for face recognition, Signal Process. 93 (6) (2013) pp.1496–1509.
[48] T. Wang, Z. Peng, J. Liang, S. Wen, M.Z.A. Bhuiyan, Y. Cai, J. Cao, Following targets for mobile tracking in wireless sensor networks, ACM Trans. Sensor Netw. 12 (4) (2016) 31.
[49] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, H. Snoussi, Kernel-based machine learning using radio-fingerprints for localization in wsns, IEEE Trans. Aerosp. Electron Syst. 51 (2) (2015) pp.1324–1336.

[50] Y. Chen, M.M. Crawford, J. Ghosh, Integrating support vector machines in a hierarchical output space decomposition framework, in: Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International, 2, IEEE, 2004, pp. 949–952.

[51] R. Górak, M. Luckner, Modified random forest algorithm for wi–fi indoor localization system, in: International Conference on Computational Collective Intelligence, Springer, 2016, pp. 147–157.

[52] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.