Combining Classifier Guided by Semi-Supervision

Mohamad Mohamadi$^1$, Hamid Parvin$^{1,2}$, Es'hagh Faraji$^{1,2}$, Sajad Parvin$^1$

1) Department of Computer Engineering, Nourabad Manasani Branch, Islamic Azad University, Nourabad Manasani, Iran
2) Young Researchers and Elite Club, Nourabad Manasani Branch, Islamic Azad University, Nourabad Manasani, Iran

mohammadi.oty@gmail.com; parvinhamid@gmail.com; eshagh1363@gmail.com; sajad.parvin@gmail.com

Abstract

The article suggests an algorithm for regular classifier ensemble methodology. The proposed methodology is based on possibilistic aggregation to classify samples. The argued method optimizes an objective function that combines environment recognition, multi-criteria aggregation term and a learning term. The optimization aims at learning backgrounds as solid clusters in subspaces of the high-dimensional feature-space via an unsupervised learning including an attribute discrimination component. The unsupervised clustering component assigns degree of typicality to each data pattern in order to identify and reduce the effect of noisy or outlayd data patterns. Then, the suggested technique obtains the best combination parameters for each background. The experimentations on artificial datasets and standard SONAR dataset demonstrate that our classifier ensemble does better than individual classifiers in the ensemble.

Keywords: Semi-supervised Learning, Ensemble Learning, Classifier Ensemble

1. Introduction

Around five million new mines are yearly buried in the ground [1]. Most of these mines are anti-personal, and claim the lives of around 70 civilians on daily basis, in regions like Afghanistan, Angola, Bosnia, and Cambodia [2]. The tactical and psychological effectiveness of these land mines, along with their simple and low cost fabrication are the main reasons behind their proliferation. In other words, they emerged as an interesting alternative for country and armed organizations which cannot acquire sophisticated defense systems [4]. Thus, despite mine-clearing efforts around the world, million landmines are still deployed. The United Nations claim that conventional mine neutralization methods would require over 1000 years to clear out the mine fields around the world [1]. Hand probes and metal detectors which represent the classical detection methods of buried land mines cannot be used for large operations. Other techniques such as dogs trained to sniff out explosives, and ground-penetrating radar proved to be slow and dangerous because they should operate very close to mines to be able to detect them. During the last decade, infrared camera based detection of buried mines proved to be effective when the field conditions are appropriate. Also, novel mine neutralization technologies include energetic photon detection, thermal neutron activation, infrared emission, and ground-penetrating radar [1]. The latest researches
attempt to aggregate these technologies with conventional metal detectors in order to enhance their detection performance [4].

Recently, the SOund NAvigating and Ranging (SONAR) which is a sound propagation technology, has been exploited for underwater mine detection. It has yielded an impressive growth of research related to the detection and classification of SONAR signals [5]. Side-scan sonar imagery conveys high resolution shots of the sea floor scene. Despite the contribution of these images in promoting applications like Mine Counter Measure (MCM) where speed is a key factor, they have not facilitated that much the detection of objects of interest in these scenes. In fact, for underwater scenes, object detection and classification is even more challenging due to the acoustical medium and the wide variability and heterogeneity of this environment. Also, the objects located on the sea floor or buried under the sand, are difficult to detect because their appearance may vary considerably based on the neighboring scene. Despite these challenges, the ability of SONAR imaging to operate in poor visibility conditions without additional light sources requirements triggered tremendous efforts to develop under water mine detection systems based on this image modality [6, 7]. Additionally, the low fabrication cost along with the low power consumption represents other main advantages that motivated researchers to include SONAR module in Autonomous Underwater Vehicles (AUVs) for under water mine detection. Processing this high-dimensional data on board has become an urgent need for this solution. Also, embedding unsupervised decision-making features and reducing the expert involvement in the recognition process emerged as new active research field. Novel clearing operations of underwater mines would rely on AUVs equipped with SONAR capabilities, and adopting Computer Aided detection (CAD) solutions. The recognition task consists in classifying signatures of region of interest as mine or not. Choosing an appropriate supervised classification model for sonar data pattern recognition is a critical issue for objects of interest detection under the sea [8].

Recently, classifiers aggregation has emerged as heuristic approach relying on the assumption that experts based on various features and/or methodologies would inherit complementary information and yield more accurate decisions. In other words, if the experts collaborate, aggregated decisions would exploit the strengths of the individual experts, and reduce the impact of their weaknesses in order to enhance the final classification accuracy. A variety of schemes have been proposed for combining multiple classifiers. Most of these approaches rely on the assumption that classifier decisions are independent. However, this assumption rarely holds, and the outputs of multiple classifiers are usually highly correlated in practice. Therefore, assigning weights to subsets of classifiers in order to consider their mutual interaction, along with the fusion weights to the individual classifiers would overcome this issue. In [9], a fuzzy based optimization to partition the feature space has been adopted for classifiers fusion. The partitioning of the feature space is based on the standard sum of intra-cluster distances. However, for complex classification problems, the data is usually noisy which yields inaccurate partitions of the feature space. In fact, noise points cannot be identified and get assigned to the closest component. Moreover, since their fuzzy memberships can be high (close to 1), they can affect the feature space clustering.

Learning using clustering and feature discrimination algorithm may lead to sub-optimal solutions depending on the complexity of the data. One possible approach to achieve robust results is to use partial supervision to guide the clustering process and
narrow the space of possible solutions. This additional information can be under the form of labels, hints, or constraints. Supervision in the form of constraints is more practical. Typically, it consists of a set of pairs of points that should belong to the same cluster and another set of points that should belong to different clusters.

In this paper, we propose a possibilistic based local approach that adapts multi-classifier fusion to different regions of the feature space. The proposed approach starts by categorizing the training samples into different clusters based on the subset of features used by the single classifiers, and their confidences. This phase is a complex optimization problem which is prone to local minima. To alleviate this problem we include a semi-supervised learning term [10] in the proposed objective function. This categorization process associates a possibilistic membership, representing the degree of typicality, with each data sample in order to identify and reduce the influence of noise points and outliers [11]. Also, an expert is appointed for each obtained cluster. These experts represent the best classifiers for the corresponding cluster/context. Then, aggregation weights are estimated by the fusion component for each classifier. These weights reflect the performance of the classifiers within all contexts. Finally, for a given test sample, the fusion of the individual confidence values is obtained using the aggregation weights associated with the closest context.

The rest of this manuscript is organized as follows. State-of-the-art of underwater object detection, and fusion techniques are outlined in Section 2. Our local fusion approach is presented in Section 3. Section 4 describes the application of the proposed fusion to underwater mine detection using sonar data. Finally, we conclude in Section 5.

2. Research Method

2.1 Underwater Target Detection

The earliest researches applying supervised machine learning techniques on side-scan sonar imagery have been conducted at the US Naval Surface Warfare Center Coastal Systems Station (CSS), and Colorado State University [12]-[13]. In [12], the authors proposed an automatic technique for discriminating between mine-like target and clutter returns in sonar imagery. More specifically, they designed and applied an adaptive clutter suppression linear Finite Impulse Response (FIR) filtering technique to side scan sonar imagery data. The overall mine detection processing string includes automatic gain control, data decimation, Adaptive Clutter Filtering (ACF), 2D normalization, thresholding, exceedance clustering, limiting the number of exceedances and secondary thresholding processing blocks. The authors in [18] combined a Detection Density ACF, and Attracted-Based Neural Network. Their Approach consists of the following phases: image normalization, ACF, selecting the largest ACF output pixels, convolving the selected pixels with a mine-size rectangular window, applying a Bayesian decision rule to detect mine-like pixels, grouping the mine-like pixels into objects, extracting object features, and classifying objects as either a mine or a non-mine with a neural network. The obtained results show that the detection density ACF approach reduces significantly the false alarm rate, and improves the overall accuracy. An advanced mine detection and classification (AMDAC) algorithm has been outlined in [14]. It consists of an improved detection density algorithm, a stepwise feature selection strategy, a k-nearest neighbor attractor-based neural network classifier, and an optimal discriminatory filter classifier. The detection phase uses a nonlinear matched filter to recognize mine-size
regions in the sonar image which closely match the signature of a mine. In [15], a non-parametric algorithm is described for implementing the optimal multi-dimensional Bayesian classifier, whereby a given feature vector of a sonar image is assigned to either mine or mine-like class on the basis of a log-likelihood ratio test (LLRT). In [16], the authors presented an automatic image recognizer that can assist human operator. It relies on rules drawn from operator experience to recognize candidate mines in imagery. The authors in [17] described a sub band-based learning scheme for underwater mines classification. It consists of a feature extraction component, a feature selection stage, and a back-propagation neural network classifier. An adaptive underwater target classification system able to handle underwater environmental conditions in acoustic back-scattered data is outlined in [18]. The main component of this system consists in the adaptive feature mapping that optimizes the classification accuracy. The mapping feature vector aims to make it invariant to the environmental changes. On the other hand, a K-Nearest Neighbors (K-NN) classifier is adopted to assign the closest instance to unknown pattern in the feature space. The final classification decision is obtained using a back-propagation neural network (BPNN). The approach proved to be promising, especially when the environment conditions are varying. In [19], the researchers proposed a target detection system in side-scan sonar imagery. The images are categorized using Principal Component Analysis (PCA) into mine and mine-like classes. The researchers in [20] outlined an automated underwater mine detection method where the considered features are inherited from facial recognition application, and extracted based on shadow/highlight segmentation. These features have been coupled with typical supervised learning algorithms to categorize mine and mine-like objects. A statistical classification method using deformable template model has been proposed in [13] to discriminate between man-made and natural objects in high resolution sonar images. The classification task is formulated as a twofold process. First, the detection of a region of interest is achieved through the optimization of a cost function. Then, the convergence value of this function yields whether the expected object is present in the scene or not. The energy minimization problem is tackled using relaxation technique (hybrid genetic algorithm). This approach achieved promising results.

More recent underwater mine detection works aim to aggregate classifiers in order to improve the system accuracy. In [21], the authors proposed an image fusion approach for sonar systems that collect multiple images in a single pass. First, each image is processed using one classification algorithm. Then, a fuzzy-logic algorithm is adopted to generate the final decision by fusing the detections that are common to multiple images. The authors claim that the fusion process reduces considerably false alarms. The authors in [22, 23] described a sea mine Computer Aided Classification (CAC) system which the main component is the log-likelihood-ratio-test fusion algorithm which proved to be yielding significant improvements over single classifier based systems. A method for combining the outputs of three different CAC algorithms based on side-scan sonar imagery is described in [25]. The fusion consists in considering different planar image coordinates, assigning a confidence factor for each individual classifier, and fusing their outputs to yield the final decision. This algorithm relies on a constrained optimization approach aiming to minimize false alarms. In [25], the authors propose a fusion algorithm to aggregate the outputs of multiple classifiers in order to detect and classify sea mines in mine hunting operations. The researchers in [26] outline
an improved sea mine classification approach consisting in pre-processing, adaptive clutter filtering (ACF), normalization, detection, feature extraction, optimal subset feature selection, feature orthogonalization, classification and fusion processing blocks. The fusion components adopt a Volterra feature Log-Likelihood-Ratio-Test (LLRT) algorithm. The authors in [27] describe a score-based fusion algorithm to aggregate multiple detection and classification algorithms where only the scores of the individual algorithms are considered for the final decision on whether an object is a mine or not. As one can notice existing fusion based underwater mine detection systems rely on simple fusion approaches. On the other hand, advanced fusion techniques have been formulated to overcome various classification challenges for various applications. Therefore, in the following we outline state-of-the-art fusion techniques that are related to our proposed fusion approach.

2.2 Fusion of Classifier Decisions

Classifier fusion has yielded in promising finding, and has outperformed single classifier systems both theoretically and experimentally [28]. The classifiers complementarily is the main characteristic which allows an ensemble of classifiers to outperform individual learners by inheriting their strengths and limiting their weaknesses. Nonetheless, two critical conditions for fusion algorithms to outperform individual classifiers are diversity and accuracy [29]. Classifier fusion relying on effective aggregation of classifiers outputs considers all learners equally trained and competitive over the feature space. For testing, single experts classifications are performed simultaneously, and the resulting outputs are aggregated to yield a final fusion decision. Typical fusion approaches include Borda count [30], average [31], majority vote [32, 33], Bayesian [34], probabilistic [35], and weighted average [36, 37].

Approaches for combining diverse learners can be categorized into two main classes: local approaches and global approaches. Unlike global approaches which assign an average relevance degree for each learner with respect to the whole training set, local approaches consider a relevance degree to the different training set subspaces. This relies on the assumption that higher classification accuracy can be reached using appropriate data-dependent weights. Clustering of the input data samples into homogeneous categories during the training phase is required for local fusion approach. This clustering may be achieved on the space of individual learner classification [38], based on which classifiers behave similarly [39], or using attributes of the input space [40]. Next, in each space region, the most accurate classifier is appointed as expert [41]. For classification, unknown instances get assigned to regions, and the corresponding expert learner of this region generates the final decision. A dynamic data partitioning and classification during the testing phase is proposed in [42, 43]. The authors estimate the classifier accuracies using sample vicinity in the local regions of the feature space, and the most accurate one is used to predict the class of the test sample. The local fusion approach called Context-Dependent Fusion (CDF) in [43] starts by clustering the training instances into homogeneous categories of contexts. This clustering phase and the selection of a local expert learner are two sequentially independent stages of CDF. The researchers in [44] described a generic framework for context-dependent fusion which simultaneously clusters the feature space, and aggregates the outputs of the expert learners. This fusion approach uses a simple linear aggregation to generate fusion weights for individual learners. However, these weights may be inefficient to capture the classifiers mutual interaction.
The authors in [44] outlined an approach that assesses the performance of each expert in local regions of the feature space. For each local region, the most accurate classifier is exploited to predict the final decision. However, the performance evaluation for test instances is timely complex, and affects the practicality of the approach with large data. In [45], the clustering and selection phase determines the statically most accurate classifier. First, the clustering of the training instances discovers the decision regions. Then, the most accurate learner on this local region is chosen. The main drawback of this solution is that it does not handle more than one classifier per region. In [46], the researchers extended the clustering and selection algorithm so it divides the training dataset into correctly and incorrectly categorized instances. The feature space is then grouped by clustering the training instances. For testing, the most effective classifier in the vicinity of the test instance is selected in order to generate the final decision. In other words, each learner maintains its corresponding cluster. This approach reduces the computational effectiveness of the solution in [45]. Recently, in [9, 47], the authors outlined a local fusion approach that categorizes the feature space into homogeneous clusters based on their features, and takes into consideration the obtained clusters when aggregation individual learners outputs. The fusion stage consists in assigning an aggregation weight to each individual learner with respect to each context based on its relative performance within it. Notice that the used fuzzy approach increases the sensitivity of the fusion component to outliers which decrease the overall classification performance. In [11], the authors proposed a possibilistic based local approach that adapts the fusion method to different regions of the feature space. It categorizes the training samples into different clusters based on the subset of features used by the single classifiers, and their confidences. This clustering phase generates possibilistic memberships representing the degree of typicality of each data sample in order to identify and discard noise points. Also, an expert classifier is associated with each obtained cluster. More specifically, aggregation weights are simultaneously learned for each classifier. Finally, for a given test sample, the fusion of the individual confidence values is obtained by using the aggregation weights associated with the closest context/cluster. Although this approach yielded promising results, the adopted optimization approach is prone to local minima.

3. Local Fusion Based on Possibilistic Context Extraction

Let \( T = \{t_j \mid j = 1, \ldots, N\} \) be the desired output values of \( N \) training observations. These output values have been obtained using \( K \) classifiers. Each classifier \( k \) uses its own feature set \( X_k = \{x^k_j \mid j = 1, \ldots, N\} \) and generates the confidence values \( Y_k = \{y^k_j \mid j = 1, \ldots, N\} \). The \( K \) feature sets are then concatenated to generate one global descriptor, \( \chi = \bigcup_{k=1}^{K} x^k = \{x_j = [x^1_j, \ldots, x^K_j] \mid j = 1, \ldots, N\} \). The possibilistic-based context extraction for local fusion algorithm in [45] has been formulated as partitioning the data into \( C \) clusters minimizing one objective function. However, this clustering approach requires the estimation of various parameters using complex optimization and is prone to fall in several local minima. To overcome this potential drawback, we propose a semi-supervised version of the algorithm in [11]. The supervision information relies on two sets of pairwise constraints. The first one is Should-link constraints which specify that two data instances are expected to belong to the same cluster. The second
set of constraints is the ShouldNot-link ones which specify that two data instances are expected to belong to different clusters.

Let SL be the set of Should-link pairs of instances. If the pair \((X_i, X_j)\) belongs to SL, then \(X_i\) and \(X_j\) are expected to be assigned to the same cluster. Similarly, let NL be the set of Should-not-link pairs. If the pair \((X_i, X_j)\) belongs to NL, then \(X_i\) and \(X_j\) are expected to be assigned to different clusters. In this work, we reformulate the problem of identifying the \(C\) components/clusters as a constrained optimization problem. More specifically, we modify the objective function in [11] as follows:

\[
J = \sum_{j=1}^{N} \sum_{i=1}^{C} u_{ij}^N \sum_{k=1}^{K} v_{ik}^q d_{ijk}^2 + \sum_{j=1}^{N} \sum_{i=1}^{C} \beta_i u_{ij}^N \left( \sum_{k=1}^{K} \omega_{ik} y_{kj} - t_j \right)^2 + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - \sum_{i=0}^{C} u_{ij} + \mu(X_i, X_k \in NL) i=1Cutimukim + (X_i, X_k \in SL) i=1Cp=1, p\neq iCutimukpm
\]

Subject to:

\[
\forall j: 0 < \sum_{i=1}^{C} u_{ij} < N,
\forall i,j: u_{ij} \in [0,1],
\forall i: \sum_{k=1}^{K} v_{ik} = 1,
\forall i, k: v_{ik} \in [0,1],
\forall i, k: \omega_{ik} \in [0,1],
\forall i: \sum_{k=1}^{K} \omega_{ik} = 1.
\]

In (1), \(u_{ij}\) represents the possibilistic membership of \(X_i\) in cluster \(i\) [11]. The \(C \times N\) matrix, \(U = [u_{ij}]\) is called a possibilistic partition if it satisfies:

\[
\left\{ \begin{array}{l}
\forall i,j: u_{ij} \in [0,1] \\
\forall j: 0 < \sum_{i=1}^{C} u_{ij} < N \\
\end{array} \right.
\]

On the other hand the \(C \times K\) matrix of feature subset weight, \(V = [v_{ik}]\) satisfies

\[
\left\{ \begin{array}{l}
\forall i, k: v_{ik} \in [0,1] \\
\forall i: \sum_{k=1}^{K} v_{ik} = 1 \\
\end{array} \right.
\]

Finally the \(C \times K\) matrix of classifier weight, \(\Omega = [\omega_{ik}]\) satisfies

\[
\left\{ \begin{array}{l}
\forall i, k: \omega_{ik} \in [0,1] \\
\forall i: \sum_{k=1}^{K} \omega_{ik} = 1 \\
\end{array} \right.
\]

The first term in (1) corresponds to the objective function of the SCAD algorithm [48]. It aims to categorize the \(N\) points into \(C\) clusters centered in \(c_i\) such that each sample \(x_j\) is assigned to all clusters with fuzzy membership degrees. Also, it is intended to simultaneously optimize the feature relevance weights with respect to each cluster. SCAD term is minimized when a partition of \(C\) compact clusters with minimum sum of intra-cluster distances is discovered. The second term in (1) intends to learn cluster-dependent aggregation weights of the \(K\) algorithm outputs. \(\omega_{ik}\) is the aggregation weight assigned to classifier \(k\) within cluster \(i\). This term is minimized when the aggregated partial output values match the desired output ones. The third term in (1)
results in the generation of the possibilistic membership \( u_{ij} \) which represents the degree of typicality of each data point within every cluster, and reduces the effect of outliers on the learning process. In (1), \( m \in [1, \infty) \) is called the fuzzier, and \( \eta_i \) is a positive constant that controls the importance of the third term with respect to the first and second ones. This term is minimized when \( u_{ij} \) are close to 1, thus, avoiding the trivial solution of the first term (where \( u_{ij} = 0 \)). Note that the term \( \sum_{i=1}^{C} u_{ij} \) is not constrained to be equal to 1. In fact, points that are not representative of any cluster will have \( \sum_{i=1}^{C} u_{ij} \) close to zero and will be considered as noise. This constraint relaxation overcomes the disadvantage of the constrained fuzzy membership approach which is the high sensitivity to noise and outliers. The parameter \( \eta_i \) is related to the resolution parameter in the potential function and the deterministic annealing approaches. It is also related to the idea of "scale" in robust statistics. In any case, the value of 0.7 determines the distance at which the membership becomes 0.5. The value of \( \eta_i \) determines the "zone of influence" of a point. A point \( X_j \) will have little influence on the estimates of the model parameters of a cluster if \( \sum_{k=1}^{K} v_{ik}^q (d_{ijk}^s)^2 \) is large when compared with \( \eta_i \). On the other hand, the "fuzzier" \( m \) determines the rate of decay of the membership value. When \( m = 1 \), the memberships are crisp. When \( m \to \infty \), the membership function does not decay to zero at all. In this possibilistic approach, increasing values of \( m \) represent increased possibility of all points in the data set completely belonging to a given cluster. The last term in (1) represents the cost of violating the pairwise Should-link, and ShouldNot-link constraints. These penalty terms are weighted by the membership values of the instances that violate the constraints. In other words, typical instances of the cluster which have high memberships yield larger penalty term. The value of \( \mu \) controls the importance of the supervision information compared to the other terms.

The performance of this algorithm relies on the value of \( \beta \). Over estimating it results in the domination of the multi-algorithm fusion criteria which yields non-compact clusters. Also, sub-estimating \( \beta \) decreases the impact of the multi-algorithm fusion criteria and increases the effect on the distances in the feature space. When appropriate \( \beta \) is chosen, the algorithm yields compact and homogeneous clusters and optimal aggregation weights for each algorithm within each cluster.

Minimizing \( J \) with respect to \( U \) is equivalent to minimizing the following individual objective functions with respect to each column of \( U \):

\[
J^{(i)}(U_i) = \sum_{j=1}^{N} u_{ij}^m \sum_{k=1}^{K} v_{ik}^q d_{ijk}^2 + \sum_{j=1}^{N} \beta_i u_{ij}^m \left( \sum_{k=1}^{K} \omega_{ik} y_{kj} - t_j \right)^2 + \eta_i \sum_{j=1}^{N} (1 - u_{ij})^m \\
+ \mu \left( \sum_{(X_i, X_k \in NL)} u_{it}^m u_{ik}^m + \sum_{(X_i, X_k \in SL)} \sum_{p=1, p \neq l}^{C} u_{it}^m u_{pk}^m \right) + \Psi_i \left( 1 - \sum_{q=1}^{K} v_{iq} \right) \\
+ \delta_i \left( 1 - \sum_{k=1}^{K} \omega_{ik} \right)
\]

For \( i = 1, \ldots, C \). By setting the gradient of \( J^{(i)} \) with respect to the possibilistic memberships \( u_{ij} \) to zero, we obtain
This can result in the following necessary condition to update $u_{ij}$:

$$u_{ij} = \left[ 1 - \left( \frac{v_{ij}^q}{u_{ij}} \right)^{\frac{1}{m-1}} \right]^{-1}$$  \hspace{1cm} (6)

where

$$D_{ij} = \sum_{k=1}^{K} v_{ik}^q d_{ijk}^2 + \beta_i \left( \sum_{k=1}^{K} \omega_{ik} y_{kj} - t_j \right)^2 + \mu \left( \sum_{x_j, x_k \in NL} u_{ik}^m + \sum_{x_j, x_k \in SL} \sum_{p=1, p \neq i}^{C} u_{ipk}^m \right)$$  \hspace{1cm} (7)

$D_{ij}$ represents the total cost when considering point $X_j$ in cluster $i$. As it can be seen, this cost depends on the distance between point $X_j$ and the cluster's centroid $c_i$, the cost of violating the pairwise *Should-link*, and *ShouldNot-link* constraints (weighted by $\mu$), and the deviation of the combined algorithms' decision from the desired output (weighted by $\beta$). More specifically, points to be assigned to the same cluster: (i) are close to each other in the feature space, and (ii) their confidence values could be combined linearly with the same coefficients to match the desired output.

As it is provided in Appendix A, minimizing $J$ with respect to the feature weights

$$v_{lk} = \frac{1}{\sum_{r=1}^{K} \left( \frac{D_{ij}}{v_{ir}^q} \right)^{\frac{1}{m-1}}}$$  \hspace{1cm} (8)

where $D_{ij} = \sum_{j=1}^{N} u_{ij}^m d_{ij}^2$.

Minimization of $J$ with respect to the prototype parameters, and the aggregation weights yields

$$c_{lk} = \frac{\sum_{j=1}^{N} u_{ij}^m x_{lk}}{\sum_{j=1}^{N} u_{ij}^m}$$  \hspace{1cm} (9)

And according to Appendix A

$$\omega_{lk} = \frac{1 - \sum_{r=1}^{K} \sum_{j=1}^{N} u_{ij}^m y_{kj}^2 (y_{kj} - y_{lj})^2}{\sum_{r=1}^{K} \sum_{j=1}^{N} u_{ij}^m y_{kj}^2 (y_{kj} - y_{lj})^2} \cdot \frac{\sum_{j=1}^{N} u_{ij}^m y_{kj}}{\sum_{j=1}^{N} u_{ij}^m y_{kj}^2}$$  \hspace{1cm} (10)
Algorithm 1: The proposed semi-supervised possibilistic clustering, feature weighting and classifier aggregation.

**Inputs:**
- \( X \): The data instances.
- \( Y \): The confidences obtained using the different classifiers.
- \( NL \): The set of ShouldNot-Link constraints.
- \( SL \): The set of Should-Link constraints.
- \( T \): The labels of the data instances.
- \( C \): The number of clusters.
- \( m \): The fuzzyfier.
- \( q \): The exponent of the feature weights.
- \( \beta \): The weight assigned to second term at the objective function (1).
- \( \eta \): The weight assigned to third term at the objective function (1).

**Outputs:**
- \( U \): The possibilistic membership matrix of the data instances.
- \( c_i \): The Clusters centers.
- \( V \): The feature weights.
- \( W \): The aggregation weights.

**Begin**

Initialize the centers;
Initialize the possibilistic partition matrix \( U \);
Initialize the relevance weights;

**Repeat**

Compute \( d_{ik}^2 \), for \( 1 \leq i \leq C \) and \( 1 \leq j \leq N \) and \( 1 \leq k \leq K \);
Update the relevance weights \( v_{ik} \) using equation (8);
Update the partition matrix \( U \) using equation (6);
Update the aggregation weights matrix \( W \) using equations (10);
Update the feature weights matrix \( V \) using equations (8);
Update the centers using equation (9);

**Until** (centers stabilize)

The obtained iterative algorithm starts with an initial partition and alternates between the update equations of \( u_{ij} \), \( v_{ik} \), \( w_{ik} \) and \( c_{ik} \) as shown in Algorithm 1.

The time complexity of one iteration of this first component is \( O(N \times d \times K \times C) \), where \( N \) is the number of data points, \( C \) is the number of clusters, \( d \) is the dimensionality of the feature space, and \( K \) is the number of feature subsets. The computational complexity of one iteration of other typical clustering algorithms (e.g. FCM [32], PCM [44]) is \( O(N \times d \times C) \). Since we use small number of feature subsets \((K = 3)\), one iteration of our algorithm has a comparable time complexity to other similar algorithms. However, we should note that since we optimize for more parameters, it may require a larger number of iterations to converge.

After training the algorithm described above, the proposed local fusion approach adopts the steps below in order to generate the final decision for test samples:

- Run the different classifiers on the test sample within the corresponding feature subset space, and obtain the decision values, \( Y^j = \{ y_{kj} | k = 1, ..., K \} \).
The unlabeled test sample inherits the class label of the nearest training sample.

- Assign the membership degrees $u_{ij}$ to the test sample $j$ in each cluster $i$ using eq. (7).
- Aggregate the output of the different classifiers within each cluster using $\hat{y}_{ij} = \sum_{k=1}^{K} w_{ik} y_{kj}$.
- The final decision confidence is estimated using $\hat{y} = \sum_{i=1}^{C} u_{ij} \hat{y}_{ij}$.

The flowchart in Figure 1 describes the proposed local fusion approach based on the semi-supervised possibilistic context extraction. As one can notice, the training phase consists in two main interactive components. Namely, the semi-supervised possibilistic clustering and feature weighting component, and the decision fusion component. The first one categorizes the training samples into homogeneous contexts based on the features extracted by the different classifiers, their confidences, and the supervision information. The classifier aggregation component uses the confidence values generated by the individual learners, and generates weights for each classifier within each cluster. These weights represent how accurate is the learner within each context. As shown in the lower part of Figure 1, each test sample is first assigned to the closest context. Then, the fusion decision is obtained by the combination of the confidence values, obtained using the individual learners, with the context aggregation weights.

4. Experiments

We illustrate the performance of the proposed semi-supervised local fusion algorithm using synthetic data sets. For these data sets, we compare our approach to individual classifiers, and the method in [11]. Both, our approach and the method in [11] rely on possibilistic based local multi-classifier fusion. The comparison is intended to illustrate the contribution of the supervision information in enhancing the fusion accuracy. For this experiment, we use data sets with varying supervision information. In the second part of the experiments section, we use real SONAR dataset to assess the proposed work.

4.1 Synthetic Dataset

In this experiment, we illustrate the need for semi-supervised possibilistic local fusion. We use our semi-supervised local fusion approach to classify the synthetic 2-dimensional dataset in Figure 2. Let each sample be processed by two single algorithms ($K$-Nearest Neighbors ($K$-NN) with $K = 3$). Each algorithm, $k$, considers one feature $X_k$; and assigns one output value $y_k$. As shown in Figure 2, samples from Class 0 are represented using blue dots and samples from Class 1 are displayed in red. Black samples represent noise samples. The dataset consists of four clusters. Each one of them is a set of instances from the two classes. In Figure 3, we show the clustering result of this dataset using possibilistic clustering and feature weighting algorithm [49]. As it can be seen, points assigned low memberships ($<0.1$) with respect to all clusters (i.e. noise points) are shown in magenta. Figure 4 shows the classification results obtained using individual learners. As one can notice, both learners fail to optimally categorize the data, and their accuracies are 69% and 81%, for learner 1 and learner 2, respectively. The accuracy of each learner depends on the region of the feature space. More specifically, Figure 4(a) illustrates how learner 1 classifies correctly most of the
instances in the top right ellipsoidal cluster. However, it achieves 48% accuracy on the three spherical clusters. Also, learner 1 performs better for the top right ellipsoidal cluster as shown in Figure 4(b). This example proves the need for local fusion approach combine strengths of the classifiers in different regions of the feature space.

In order to construct the set pairwise constraints, we randomly select samples that are at the boundary of each cluster. We consider 7% of the total number of instances as Should-link and ShouldNot-link sets. Pairs of instances belonging to the same cluster (based the ground truth) form the Should-link set. Similarly, pairs that belong to different clusters form the ShouldNot-link set.

![Fig. 1. The proposed fusion algorithm based on the semi-supervised possibilistic context extraction](image-url)
The results obtained using the unsupervised local fusion approach (with a threshold = 0.1) in [11] and the proposed approach are shown in Figure 5. As it can be seen in Figure 5(b), the supervision information improved the overall fusion performance compared to the method in [11]. The corresponding aggregation weights assigned to the individual learners with respect to the different clusters are shown in Table 1. As one can notice, the method in [11] and the proposed method consider classifier 1 more reliable for clusters 2 (top right cluster). On the other hand, both methods find out that classifier 2 is more reliable for the other clusters. In order to assess the contribution of the supervision information included in the proposed approach, we increase the supervision rate progressively from 0 to 11%.

**Table 1. Learned weights for each classifier with respect to the different clusters obtained using the method in [45] and the proposed semi-supervised method**

<table>
<thead>
<tr>
<th></th>
<th>Cluster#</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method in [50]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifier 1</td>
<td></td>
<td>0.3112</td>
<td>0.7301</td>
<td>0.0266</td>
<td>0.299</td>
</tr>
<tr>
<td>Classifier 2</td>
<td></td>
<td>0.6888</td>
<td>0.2699</td>
<td>0.9734</td>
<td>0.9771</td>
</tr>
<tr>
<td><strong>Proposed semi-supervised method</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 1</td>
<td></td>
<td>0.2946</td>
<td>0.7366</td>
<td>0.0290</td>
<td>0.0247</td>
</tr>
<tr>
<td>Cluster 2</td>
<td></td>
<td>0.7054</td>
<td>0.2634</td>
<td>0.9710</td>
<td>0.9753</td>
</tr>
</tbody>
</table>
Fig. 3. Clustering results of the synthetic data in Figure 1 using the method in [26]. Magenta points correspond to the identified noise samples (threshold =0.1)

Fig. 4. Classification result of (a) the first algorithm (based on feature x1). (b) the second algorithm (based on feature x2)
Fig. 5. Local possibilistic fusion results (threshold = 0.1) using (a) method in [11] and (b) the proposed semi-supervised approach with 7% pairwise constraints

Fig. 6. Accuracy of the proposed method on the dataset in Figure 3, and when the supervision rate is varied from 0 to 11%

We use the accuracy as performance measure to evaluate the performance of our semi-supervised method. The overall accuracy of the partition is computed as the average of the individual class rates weighted by the class cardinality. To take into account the sensitivity of the algorithm to the initial parameters, we run the algorithm 10 times using different random initializations. Then, we compute the average accuracy values for each supervision rate. As it can be seen in Figure 6, the accuracy increased at a much lower rate with supervision rate larger than 7%. Thus, for the rest of the experiments we set the supervision rate used to guide our clustering algorithm to 7%.
4.2 Standard Dataset

In this section, we use our approach to classify standard dataset frequently used by researchers from the machine learning community. Namely, we consider the SONAR dataset [50] which consists of 208 instances and 60 attributes. 97 instances were obtained by bouncing sonar signals off a metal cylinder under various condition and at various angles. A variety of different aspect angles, spanning 90 degree for the cylinder and 180 degrees for the rock were considered to contain the dataset signal. Each attribute represents the energy within a particular frequency band, integrated over a given period of time. SOANR dataset [50] is summarized in Table 2.

Table 2. Standard dataset from [50] used to compare the performance of individual classifiers, method in [11], and the proposed approach

<table>
<thead>
<tr>
<th>Number of samples</th>
<th>208</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensionality</td>
<td>60</td>
</tr>
<tr>
<td>Number of classes</td>
<td>2</td>
</tr>
</tbody>
</table>

In our experiments, for individual learners and local fusion approaches we adopt a 5-fold cross-validation in which each fold is treated as a test set with the rest of the folds used for training. Also, we use the following performance measures to assess single learners and fusion techniques:

Table 3. Confusion matrix obtained by the individual classifiers and the fusion approaches

<table>
<thead>
<tr>
<th>Peredicted as Class 1</th>
<th>Peredicted as Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>False Positive (FP)</td>
<td>True Negative (FN)</td>
</tr>
</tbody>
</table>

- **Accuracy**: The fraction of decisions that is correct for both classes. We define the classifier accuracy as the ratio of the number of correctly classified samples to the total number of instances as follows:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)
\]

where TP, TN, FP and FN are defined in Table 3.

- **Recall**: The ratio of True Positive comments over the sum of True Positive and False Negative comments:

\[
\text{Recall} = \frac{TN}{TN + FN} \quad (13)
\]

- **Precision**: The ratio of True Positive comments over the sum of True Positive and False Positive comments:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (14)
\]

- **F-measure**: The harmonic mean of precision and recall:
We divide the SONAR features into three subsets, and we dedicate one learner for each one of them. We run simple K-NN learner to generate confidence values for each instance. We categorize the training samples using 3 K-NN classifiers (K=3) within their corresponding feature subspaces. Then, the proposed semi-supervised local fusion is used to categorize the training instances into 3 homogeneous clusters, and learn the optimal aggregation weights. Then, test instances are classified using the three individual learners, and assigned to the closest cluster. Finally, the fusion decision is generated by combining the partial confidences with the aggregation weights of the closest cluster. Notice that Should-link and ShouldNot-link constraints are generated using a clustering algorithm. More specifically, we cluster the training dataset using the possibilistic-based algorithm in [58], and we include pairs of typical instances (with high possibilistic membership), belonging to the same cluster, in the Should-link set. On the other hand, pairs of typical instances (with high possibilistic membership), belonging to different clusters, are included in the ShouldNot-link set. We limit the number of pairwise constraints to 7% of the total number of instances.

In Figure 7, we report the mine detection accuracies, precision, recall and F-measure obtained using K-NN classifier with different values of the parameter K. As it can be seen, K = 5 yields the best overall performance measures. Thus, for the rest of the experiments, we set this K to 5.

We compare the obtained average accuracy, precision, recall, and F-measure values obtained using individual K-NN learners, the method in [11], and the proposed method with the SONAR dataset in Table 4. Our semi-supervised approach outperforms the other classifiers on this dataset based on the four performance measures. This proves that the association of supervision information with local fusion technique yields better clustering results and let individual learners cooperate more efficiently to generate more accurate final decision. This confirms the results obtained with synthetic datasets in the previous experiment.

\[
F\text{- measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]  

\[ (15) \]

![Figure 7](image-url)  

**Fig. 7.** Performance measures obtained using K-NN classifier on SONAR data [50] when the parameter K varies from 1 to 7

Table 5 shows the learners aggregation weights with respect to the different clusters generated by our algorithm. These weights reflect the impact of each individual learner

43
within each cluster. For instance, the second individual $K$-NN is perceived by our approach as the most accurate classifier for instances from cluster 1. Similarly, the highest aggregation weight is assigned to the first individual $K$-NN within cluster 3.

To demonstrate that the semi-supervised local fusion exploits the strengths of the individual learners within local regions of the features space, we report the accuracy of the three individual learners ($K$-NN) within the 3 clusters. These performance measures shown in Table 6 are calculated based on the classification of test samples belonging to each cluster separately (given the membership degrees generated by the proposed semi-supervised clustering algorithm). As one can notice, the local performances of the individual $K$-NN depends on the cluster. $K$-NN classifier 2 performs better than the other learners for samples from cluster 1. Consequently, $K$-NN classifier 2 is the most relevant classifier with respect to cluster 1. Thus, the highest aggregation weight is assigned to this classifier as reported in Table 5. Similarly, in cluster 3, the most accurate individual classifier is $K$-NN classifier 2.

Table 4. Performance comparison of the individual learners, the method in [11], and the proposed method for SOANR data set [50]

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$-NN1</td>
<td>0.8269</td>
<td>0.7680</td>
<td>0.8803</td>
<td>0.8203</td>
</tr>
<tr>
<td>$K$-NN2</td>
<td>0.8416</td>
<td>0.7972</td>
<td>0.8842</td>
<td>0.8384</td>
</tr>
<tr>
<td>$K$-NN3</td>
<td>0.8511</td>
<td>0.8142</td>
<td>0.8808</td>
<td>0.8461</td>
</tr>
<tr>
<td>Method in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[11]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>0.9024</td>
<td>0.9090</td>
<td>0.8947</td>
<td>0.9017</td>
</tr>
<tr>
<td>method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Learned weights for each classifier in each cluster obtained using the proposed semi-supervised local fusion with SONAR data set [50]

<table>
<thead>
<tr>
<th></th>
<th>Cluster #</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$-NN1</td>
<td>0.2080</td>
<td>0.2758</td>
<td>0.8003</td>
<td></td>
</tr>
<tr>
<td>$K$-NN2</td>
<td>0.6142</td>
<td>0.5309</td>
<td>0.0138</td>
<td></td>
</tr>
<tr>
<td>$K$-NN3</td>
<td>0.1177</td>
<td>0.1933</td>
<td>0.0959</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Per-cluster accuracy of the three $K$-NN classifiers with SONAR data [50]

<table>
<thead>
<tr>
<th></th>
<th>Cluster #</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$-NN1</td>
<td>0.6947</td>
<td>0.7523</td>
<td>0.8846</td>
<td></td>
</tr>
<tr>
<td>$K$-NN2</td>
<td>0.8713</td>
<td>0.8600</td>
<td>0.5992</td>
<td></td>
</tr>
<tr>
<td>$K$-NN3</td>
<td>0.6401</td>
<td>0.6765</td>
<td>0.5889</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Learned weights for each classifier in each cluster with SONAR data [50]

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.3775</td>
<td>0.3311</td>
<td>0.4589</td>
</tr>
<tr>
<td>K-NN</td>
<td>0.3488</td>
<td>0.3298</td>
<td>0.4702</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.2737</td>
<td>0.3391</td>
<td>0.0709</td>
</tr>
</tbody>
</table>

Table 8. Per-cluster accuracy obtained using SVM, K-NN and Naive Bayes classifiers on SONAR data [50]

<table>
<thead>
<tr>
<th>Cluster #</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.8980</td>
<td>0.8430</td>
<td>0.8799</td>
</tr>
<tr>
<td>K-NN</td>
<td>0.8577</td>
<td>0.8366</td>
<td>0.8831</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.6389</td>
<td>0.8466</td>
<td>0.5984</td>
</tr>
</tbody>
</table>

Table 9. Performance measures of the individual learners, the method in [11], and the proposed method with SONAR dataset

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-NN1</td>
<td>0.8259</td>
<td>0.7589</td>
<td>0.8788</td>
<td>0.8144</td>
</tr>
<tr>
<td>SVM</td>
<td>0.8581</td>
<td>0.7995</td>
<td>0.8865</td>
<td>0.8407</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.8337</td>
<td>0.7987</td>
<td>0.8623</td>
<td>0.8292</td>
</tr>
<tr>
<td>Method in [11]</td>
<td>0.8659</td>
<td>0.8641</td>
<td>0.8503</td>
<td>0.8571</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.9087</td>
<td>0.9126</td>
<td>0.9010</td>
<td>0.9067</td>
</tr>
</tbody>
</table>

In the following experiment we use the same feature subsets defined in the previous experiment. However, we use them with different classifiers. Namely, we classify SONAR instances in each features subset using K-NN, Naive Bayes and SVM classifiers. Then, our semi-supervised local fusion algorithm clusters the training data, generates 3 categories, and learns optimal aggregation weights. This experiment is intended to show that our approach does not require specific classifiers, and can deal with various supervised learning algorithms.

In Table 7, we report the aggregation weights learned by our semi-supervised local fusion approach for each classifier with respect to the different clusters. The achievements of the different supervised learning techniques vary drastically depending on the context/cluster. More specifically, SVM is the most important learner with respect to cluster 1. This can be explained by the highest weight assigned for SVM classifier within cluster 1. Similarly, K-NN is the most relevant classifier for cluster 3.

In Table 8 shows the per-cluster accuracy values obtained within the different clusters generated by our semi-supervised algorithm. As one can notice, the reported values are consistent with the relevance weights in Table 7. For instance, SVM which obtained the highest aggregation weight with respect to cluster 1, yields the highest accuracy with respect to this cluster. Similarly, Naive Bayes and K-NN are the most accurate classifiers in cluster 2 and cluster 3, respectively.
Table 9 displays four performance measures obtained by the different individual learners, the method in [11], and our semi-supervised local fusion approach. Namely, accuracy, precision, recall and F-measure are reported for SONAR data [50]. Our approach outperforms the other methods with respect to all the performance measures.

5. Conclusion

In this paper, we have proposed a novel approach of automatic mine detection in SONAR dataset. This approach consists in a semi-supervised local fusion algorithm which categorizes the feature space into homogeneous clusters, learns optimal aggregation weights for individual classifiers and optimal fusion parameters for each context in a semi supervised manner. The experiments have shown that the semi-supervised fusion approach yields more accurate classification than the unsupervised version in [11] and the individual classifiers on synthetic and real datasets.

Although the proposed approach yields promising results, there is still room for improvement. Future work may consist in extending the proposed approach so it handles multiple class (more than two classes) categorization problems. Finally, in order to overcome the need to specify the number of clusters/context apriori, we can investigate the ability of the possibilistic logic to generate duplicated clusters in order to find the optimal number of clusters.

References


Appendix A

Equation (8)

\[
\frac{\partial f_j^{(i)}(U_i)}{\partial v_{ik}} = \sum_{j=1}^{N} q u_{ij}^{m} v_{ik}^{q-1} d_{ijk}^{q} - \Psi_i = 0
\]

\[
\frac{\partial f_j^{(i)}(U_i)}{\partial \Psi_i} = \left(1 - \sum_{q=1}^{K} v_{iq}\right) = 0
\]

\[
v_{ik} = \left(\frac{\Psi_i}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}}
\]

\[
\sum_{k=1}^{K} \left(\frac{\Psi_i}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}} = 1
\]

\[
\Psi_i = \left(\frac{1}{\sum_{k=1}^{K} \left(\frac{1}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}}}\right)^{q-1}
\]

\[
v_{ik} = \left(\frac{\sum_{j=1}^{N} \left(\frac{1}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}}}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}}
\]

\[
\frac{1}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}} = \frac{1}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}
\]

\[
\Psi_i = \left(\frac{1}{\sum_{k=1}^{K} \left(\frac{1}{\sum_{j=1}^{N} q u_{ij}^{m} d_{ijk}^{q}}\right)^{\frac{1}{q-1}}}\right)^{q-1}
\]
\[ v_{ik} = \frac{1}{\sum_{r=1}^{K} \left( \frac{D_{ik}}{D_r} \right)^{\frac{1}{q-1}}} \]

Equation (10)

\[ \frac{\partial f^{(i)}(U_i)}{\partial \omega_{ik}} = \sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}(\omega_{ik} y_{kj} - t_j) - \theta_i = 0 \]

\[ \sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} \omega_{ik} = \sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j + \theta_i \]

\[ \omega_{ik} = \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j + \theta_i}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2} \]

\[ \sum_{k=1}^{K} \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j + \theta_i}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2} = 1 \]

\[ \theta_i \sum_{k=1}^{K} \frac{1}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2} = 1 - \sum_{k=1}^{K} \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2} \]

\[ \theta_i = \frac{1 - \sum_{k=1}^{K} \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}}{\sum_{k=1}^{K} \frac{1}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}} \]

\[ \omega_{ik} = \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2} + \frac{1 - \sum_{k=1}^{K} \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}}{\sum_{k=1}^{K} \frac{1}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}} \]

\[ \omega_{ik} = \frac{1 - \sum_{k=1}^{K} \frac{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj} t_j}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}}{\sum_{k=1}^{K} \frac{1}{\sum_{j=1}^{N} 2\beta_j u_{ij}^m y_{kj}^2}} \]