Information Sciences 249 (2013) 1–12

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Adaptive neighborhood granularity selection and combination based on margin distribution optimization

Pengfei Zhu [a,b], Qinghua Hu [a,c,d,*]

[a] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[c] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 30072, China
[d] Key Laboratory Systems Biotechnology, Minister of Education, Tianjin 300072, China

ARTICLE INFO

Article history:
Received 2 November 2012
Received in revised form 2 June 2013
Accepted 6 June 2013
Available online 13 June 2013

Keywords:
Granular computing
Granularity
Neighborhood rough set
Margin distribution
Ensemble learning

ABSTRACT

Granular computing aims to develop a granular view for interpreting and solving problems. The model of neighborhood rough sets is one of effective tools for granular computing. This model can deal with complex tasks of classification learning. Despite the success of the neighborhood model in attribute reduction and rule learning, it still suffers from the issue of granularity selection. Namely, it is an open problem to select a proper granularity of neighborhood for a specific task. In this work, we explore ensemble learning techniques for adaptively evaluating and combine the models derived from multiple granularity. In the proposed framework, base classifiers are trained in different granular spaces. The importance of base classifiers is then learned by optimizing the margin distribution of the combined system. Experimental analysis shows that the proposed method can adaptively select a proper granularity, and combining the models trained in multi-granularity spaces leads to competent performance.

1. Introduction

Granular computing utilizes information granules, drawn together by indistinguishability, similarity, proximity or functionality, to develop a granular view of the world and solve problems described with incomplete, uncertain, or vague information [16,37,38]. There are usually two basic issues with granular computing, including construction of information granules and computation with these granules [34,39]. The representative granular computing models include fuzzy sets, rough sets [11,15,18], fuzzy rough sets [19,33], neighborhood rough sets [6,7], covering rough set [42,43], and so on. Neighborhood rough set is one of the most effective granular computing models in mining heterogeneous data. It has been successfully applied in vibration diagnosis [40], cancer recognition [5] and tumor classification [29].

Neighborhood rough sets extract information granules by computing the neighborhoods of samples. Thus feature space is granulated into a family of neighborhood information granules. Hu et al. introduced neighborhood attribute reduction and a classification algorithm based on the neighborhood model [6]. For interpretation of neighborhood information granules, partition of universe is replaced by neighborhood covering and a neighborhood covering reduction based approach was derived to extract rules from numerical data [1].

How to select a proper granularity is a key problem in granular computing [32,17,13]. The sizes of granules, the relations between granules, and the operations with the granules provide the essential ingredients for developing a theory of granular computing [38]. The size of neighborhood has effect on consistency of neighborhood spaces and their approximation ability.

[*] Corresponding author. Address: School of Computer Science and Technology, Tianjin University, Tianjin 300072, China. Tel.: +86 22 27401839.
E-mail address: huqinghua@tju.edu.cn (Q. Hu).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.06.012

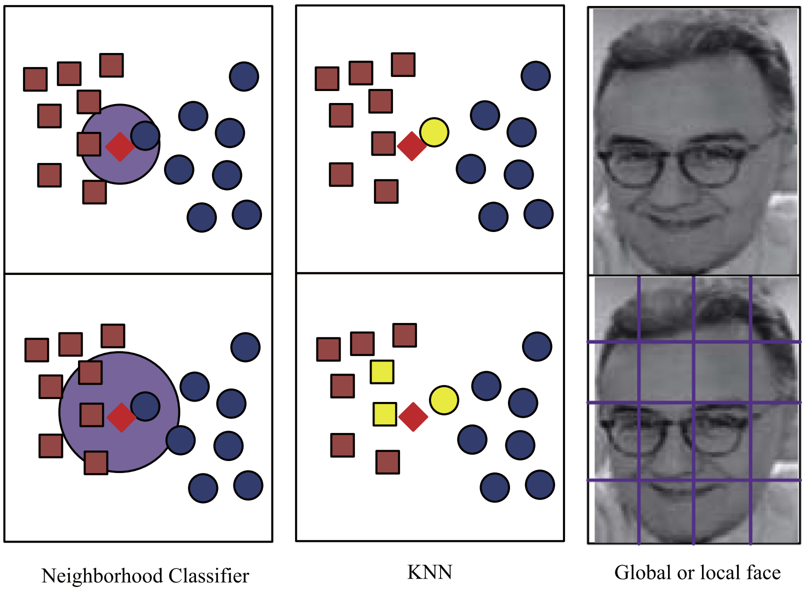

Fig. 1. Impact of granularity on classification.

If the neighborhood is small, the consistency of classification in the neighborhood space would be large. As shown in Fig. 1, the test sample may be misclassified if the granularity is not correctly set for neighborhood rough sets [6] and KNN classifier [12]. In [8], the impact of neighborhood size on attribute reduction based on neighborhood dependency was discussed. Even though the size of neighborhood of each sample varies according to their position in feature space for neighborhood covering reduction [1], the selection of neighborhood size is still up to empirical values and is still an open problem.

Given a learning task, we may obtain diverse results in different granular spaces. Hence, combining these patterns may lead to performance improvement. As illustrated in Fig. 1, we can recognize a person from the global face or local patch [4]. The combination of global and local information may lead to great improvement in recognition performance [27,41]. It is known that there are multiple attribute reducts that keep the discrimination ability of the original feature space. In different granular spaces, we can get a set of attribute reducts with complementary information. We can combine the outputs from different granular spaces.

Boosting [2] and AdaBoost [21] are the most typical and successful ensemble learning methods. They learn the weights of base classifies and the final output is a linear weighted combination of the individual outputs. Schapire [23] explained AdaBoost from margin distribution and gave the generalization bound. In [35], a bagging pruning technique was proposed based margin distribution optimization. In [41], by optimizing margin distribution, an ensemble face recognition method was proposed to combine multi-scale outputs.

In this paper, we propose a technique to select and combine different granularity based on margin distribution optimization. In different neighborhood granular spaces, we get a corresponding classification model. By optimizing margin distribution of the final decision function, we derive the weights of different granularity. The granularity with the largest weight is considered to be optimal. In addition, weights can be used to rank the granularity or combine recognition results of different granular spaces. Experimental results show that the proposed granularity selection and combination method can significantly improve the classification performance.

The structure of this paper is described as follows. In Section 2, neighborhood based granular models are introduced. Section 3 shows the granularity selection and combination method. In Section 4, experiment analysis is given to show the performance of the proposed method. Finally, conclusions and future work are presented in Section 5.

2. Neighborhood granular models

In this section, the neighborhood based granular computing model is introduced. The granularity sensitivity of the neighborhood granular models is discussed in Section 2.3.

2.1. Neighborhood rough set

As rough set theory proposed by Pawlak cannot deal with numerical data, Hu et al. introduced a rough set model based on neighborhood granulation [7].

Given an information system $\langle \boldsymbol{U},\boldsymbol{A},\boldsymbol{D}\rangle$, $\boldsymbol{U} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is a non-empty set of objects, $\boldsymbol{A} = \{a_1, \ldots, a_m\}$ is a set of attributes which describe samples, and $\boldsymbol{D}$ is the decision variable.

Definition 1 [7]. Given $\boldsymbol{x}_i \in \boldsymbol{U}$ and $\boldsymbol{B} \subseteq \boldsymbol{A}$, the neighborhood $\delta_{\boldsymbol{B}}(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$ with respect to $\boldsymbol{B}$ is defined as

$$\delta_{\boldsymbol{B}}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j | \boldsymbol{x}_j \in \boldsymbol{U}, \Delta^{\boldsymbol{B}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leqslant \delta\}, \tag{1}$$

where Δ is a distance function defined in feature spaces.

Given a metric space $\langle \boldsymbol{U}, \Delta \rangle$, the family of neighborhood granules $\{\delta_{\boldsymbol{B}}(\boldsymbol{x}_i) | \boldsymbol{x}_i \in \boldsymbol{U}\}$ forms an elemental granule system that covers the universe. A neighborhood relation $\boldsymbol{N}$ on the universe can be written as a relation matrix $(r_{ij})_{n \times n}$, where

$$r_{ij} = \begin{cases} 1, & \Delta(\boldsymbol{x}_i, \boldsymbol{x}_j) \leqslant \delta; \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

Neighborhood relations are a kind of similarity relations, which satisfy the properties of reflexivity and symmetry. Neighborhood relations draw the objects together for similarity in terms of distances. The samples in the same neighborhood granule are close to each other, so they are difficult for distinguishing [6].

A neighborhood granule degrades to an equivalence class if $\delta = 0$. In this case, the samples in the same neighborhood granule are equivalent to each other and the neighborhood rough set model degenerates to Pawlak's one. Therefore, the model of neighborhood rough sets is a natural generalization of Pawlak's rough set model [6].

In real applications, $\boldsymbol{A}$ could consist of both numerical and categorial attributes. The form of Δ is up to the types of attributes. There are a number of distance functions for mixed numerical and categorical data [30]. For example, Heterogeneous Euclidean-Overlap Metric function (HEOM) is defined as:

$$\text{HEOM}(x, y) = \sqrt{\sum_i w_{a_i} \times d_{a_i}^2(x_{a_i}, y_{a_i})}. \tag{3}$$

where w_{a_i} is the weight of attribute a_i, $d_{a_i}(x, y)$ is the distance between sample x and y with respect to attribute a_i. It is defined as:

$$d_{a_i}(x, y) = \begin{cases} 1 & \text{if the attribute value of } x \text{ or } y \text{ is unknown}, \\ overlap_{a_i}(x, y), & \text{if } a_i \text{ is a nominal attribute}, \\ rn\_diff_{a_i}(x, y), & \text{if } a_i \text{ is a numerical attribute}. \end{cases} \tag{4}$$

Definition 2 [7]. Given $\boldsymbol{U}$ and a neighborhood relation $\boldsymbol{N}$ over $\boldsymbol{U}$, $\langle \boldsymbol{U}, \boldsymbol{N} \rangle$ is called a neighborhood approximation space. For any $\boldsymbol{X} \subseteq \boldsymbol{U}$, two subsets of objects, called lower and upper approximations of $\boldsymbol{X}$ in $\boldsymbol{X} \subseteq \boldsymbol{U}$, are defined as:

$$\begin{aligned} &\underline{\mathrm{N}}\boldsymbol{X} = \{\boldsymbol{x}_i | \delta(\boldsymbol{x}_i) \subseteq \boldsymbol{X}, \boldsymbol{x}_i \in \boldsymbol{U}\}; \\ &\overline{\mathrm{N}}\boldsymbol{X} = \{\boldsymbol{x}_i | \delta(\boldsymbol{x}_i) \cap \boldsymbol{X} \neq \emptyset, \boldsymbol{x}_i \in \boldsymbol{U}\}. \end{aligned} \tag{5}$$

Definition 3 [7]. Given $\langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D} \rangle$, if $\boldsymbol{A}$ generates a family of neighborhood relation on the universe, then $NDT = \langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D} \rangle$ is a neighborhood decision system.

Definition 4 [7]. Given a neighborhood decision system $NDT = \langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D} \rangle$, $\boldsymbol{D}$ divides $\boldsymbol{U}$ into N equivalence classes: $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$, $\boldsymbol{B} \subseteq \boldsymbol{A}$ generates the neighborhood relation $\mathrm{N}_{\boldsymbol{B}}$ on $\boldsymbol{U}$. $\delta_{\boldsymbol{B}}(\boldsymbol{x}_i)$ is neighborhood information granule generated on feature space $\boldsymbol{B}$. Then the lower and upper approximations of $\boldsymbol{D}$ with respect to attributes $\boldsymbol{B}$ are defined as

$$\begin{aligned} &\underline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{D} = \cup_{i=1}^{N} \underline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{X}_i; \\ &\overline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{D} = \cup_{i=1}^{N} \overline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{X}_i. \end{aligned} \tag{6}$$

where

$$\begin{aligned} &\underline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{X} = \{\boldsymbol{x}_i | \delta_{\boldsymbol{B}}(\boldsymbol{x}_i) \subseteq \boldsymbol{X}, \boldsymbol{x}_i \in \boldsymbol{U}\}; \\ &\overline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{X} = \{\boldsymbol{x}_i | \delta_{\boldsymbol{B}}(\boldsymbol{x}_i) \cap \boldsymbol{X} \neq \emptyset, \boldsymbol{x}_i \in \boldsymbol{U}\}. \end{aligned} \tag{7}$$

Definition 5 [7]. Given $NDT = \langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D} \rangle$, the dependency degree of $\boldsymbol{B} \subseteq \boldsymbol{A}$ with respect to $\boldsymbol{D}$ is defined as

$$\gamma_{\boldsymbol{B}}(\boldsymbol{D}) = Card(\underline{\mathrm{N}_{\boldsymbol{B}}}\boldsymbol{D}) / Card(\boldsymbol{U}). \tag{8}$$

Definition 6 [7]. Given $NDT = \langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D} \rangle$, $\boldsymbol{B} \subseteq \boldsymbol{A}$, $a \in \boldsymbol{B}$, if

$$\begin{aligned} &\gamma_{\boldsymbol{B}}(\boldsymbol{D}) = \gamma_{\boldsymbol{A}}(\boldsymbol{D}), \\ &\forall a \in \boldsymbol{B} : \gamma_{(\boldsymbol{B}-a)}(\boldsymbol{D}) < \gamma_{\boldsymbol{B}}(\boldsymbol{D}), \end{aligned} \tag{9}$$

then $\boldsymbol{B}$ is an attribute reduct and can also be called a δ neighborhood separable subspace.

Definition 7 [7]. Given $NDT = \langle \boldsymbol{U}, \boldsymbol{A}, \boldsymbol{D}\rangle$, $\{\boldsymbol{B}_j | j \leqslant r\}$ is a set of attribute reducts, $\boldsymbol{Core} = \cap_{j\leqslant r}\boldsymbol{B}_j$.

$$\begin{aligned} &\boldsymbol{K} = \cup_{j\leqslant r}\boldsymbol{B}_j - \boldsymbol{Core}. \\ &\boldsymbol{K}_j = \boldsymbol{B}_j - \boldsymbol{Core}. \end{aligned} \tag{10}$$

In essence, an attribute reduct is a subset of attributes which keeps the approximation ability of the original features. It is accepted that there might be multiple reducts. In different granular spaces, each reduct contains different information, and describes the original feature space from different perspectives.

Based on neighborhood rough set model, neighborhood classifier (NEC) was proposed in [7] based on the general idea of estimating the class of a sample from its neighbors.

2.2. Neighborhood covering reduction

An algorithm for relative neighborhood covering reduction was designed in [1]. This algorithm can extract rules for classification.

Definition 8. Given $\boldsymbol{U} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{C} = \{\boldsymbol{F}_1, \ldots, \boldsymbol{F}_k\}$ is a family of non-empty subsets of objects and $\bigcup_{i=1}^{k}\boldsymbol{F}_i = \boldsymbol{U}$. We say $\boldsymbol{C}$ is a covering of $\boldsymbol{U}$, and $\boldsymbol{F}_i$ is a covering element.

Obviously, the neighborhoods of all samples form a covering of the universe. The neighborhood size of each object varies according to their spatial location. δ is set as classification margin of each object [1,3].

Definition 9 1. Given $\boldsymbol{U} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{x} \in \boldsymbol{U}$, $NH(\boldsymbol{x})$ is the nearest object of $\boldsymbol{x}$ from the same class, $NM(\boldsymbol{x})$ is the nearest object of $\boldsymbol{x}$ from other classes. Then the classification margin of $\boldsymbol{x}$ is computed as

$$M(\boldsymbol{x}) = \Delta(\boldsymbol{x}, NM(\boldsymbol{x})) - \Delta(\boldsymbol{x}, NH(\boldsymbol{x})). \tag{11}$$

If $M(\boldsymbol{x})$ is less than zero, $\boldsymbol{x}$ would be misclassified according to the nearest neighborhood rule. In this case, δ is set as zero. Hence, if there are no samples that have the same conditional attribute value while belong to the different classes, the neighborhood of $\boldsymbol{x}$ would consistently belong to the same class.

The family of neighborhood N = $\{\delta(\boldsymbol{x}_1), \delta(\boldsymbol{x}_2), \ldots, \delta(\boldsymbol{x}_n)\}$ forms a pointwise covering of the universe. $\langle \boldsymbol{U}, \boldsymbol{C}\rangle$ is a neighborhood covering space and $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ is a neighborhood covering decision system.

Definition 10 [1]. $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ is a neighborhood decision system, $\boldsymbol{X}_i$ is a decision class. If $\exists\delta(\boldsymbol{x}) \in \boldsymbol{C}$, such that $\delta(\boldsymbol{x}') \subseteq \delta(\boldsymbol{x}) \subseteq \boldsymbol{X}_i$, then $\delta(\boldsymbol{x}')$ is relatively consistent reducible with respect to $\boldsymbol{X}_i$; otherwise $\delta(\boldsymbol{x}')$ is relatively consistent irreducible.

Definition 11 [1]. Let $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ be a Type-1 consistent neighborhood covering decision system. If $\delta(\boldsymbol{x}) \in \boldsymbol{C}$, there does not exist $\delta(\boldsymbol{x}') \in \boldsymbol{C}$, such that $\delta(\boldsymbol{x}') \subseteq \delta(\boldsymbol{x}) \subseteq \boldsymbol{X}_i$, where $\boldsymbol{X}_i$ is an arbitrary decision class, then $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ is relatively irreducible; otherwise, $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ is relatively reducible.

Definition 12 [1]. Let $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ be a Type-1 consistent neighborhood covering decision system. $\boldsymbol{C}' \subseteq \boldsymbol{C}$ is a derived covering from $\boldsymbol{C}$ by removing the relatively reducible covering elements, and $\langle \boldsymbol{U}, \boldsymbol{C}', \boldsymbol{D}\rangle$ is relatively irreducible. Then we say that $\boldsymbol{C}'$ is a $\boldsymbol{D}$-relative reduct of $\boldsymbol{C}$, denoted by $\mathrm{reduct}_{\boldsymbol{D}}(\boldsymbol{C})$.

Theorem 1 [1]. *Let $\langle \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D}\rangle$ be a Type-1 consistent neighborhood covering decision system and $reduct_{\boldsymbol{D}}(\boldsymbol{C})$ be a $\boldsymbol{D}$-relative reduct of $\boldsymbol{C}$. Then $\langle \boldsymbol{U}, reduct_{\boldsymbol{D}}(\boldsymbol{C}), \boldsymbol{D}\rangle$ is also a Type-1 consistent covering decision system, and $\forall\delta(\boldsymbol{x}) \in \boldsymbol{C}$, $\exists\delta(\boldsymbol{x}') \in reduct_{\boldsymbol{D}}(\boldsymbol{C})$, such that $\delta(\boldsymbol{x}) \subseteq \delta(\boldsymbol{x}')$.*

Neighborhood covering reduction provides us with a simple and intelligent way for classification. Although δ for each sample is adaptively selected and redundant covering elements are removed, there are still redundant features that may degrade classification performance.

2.3. Granularity sensitivity

Hu et al. developed a classification algorithm (NEC) based on neighborhood rough sets. The disadvantage of NEC is that it is sensitive to the neighborhood size δ. To show the sensitivity of NEC to the granularity, we test the performance of neighborhood classifier with different δ on four data sets [14]. As shown in Fig. 2, the accuracy of NEC varies greatly with granularity. The optimal performance occurs in different size of neighborhood. It is hard to find an optimal neighborhood size for different tasks.

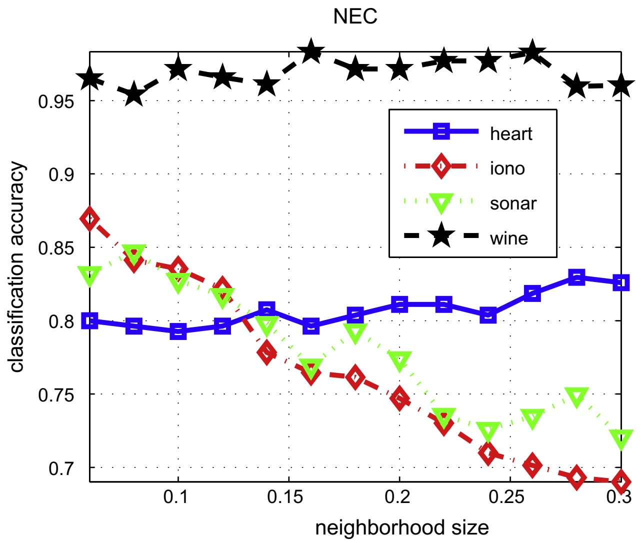

Fig. 2. Accuracy of neighborhood classifier with neighborhood size.

Neighborhood attribute reduction can remove the redundant features and keep the discrimination ability. However, different neighborhood sizes lead to different δ neighborhood separable subspaces. If we use NCR for classification, the performance is still affected by δ. As shown in Fig. 3, the performance of NCR is greatly affected by δ.

3. Granularity selection and combination

Both neighborhood classifiers and neighborhood attribute reduction are sensitive to the granularity δ. The granularity selection is a non-trivial task. The information of different granularity may be different and complementary to each other. Assume that three features {13, 1, 10} of wine are selected by neighborhood feature selection. Then rules are learned separately in feature subspace {13, 10} and feature subspace {1, 10}, as shown in Fig. 4. The learned rules are different and they may be complementary to each other. Similarly, for different granularity, we can get different attribute reducts. Although they all keep the discrimination ability of the original feature subspace, they describe original feature space from different perspectives. Multiple granular information can be combined to improve the classification performance.

Fig. 5 shows the flow chart of the proposed method. In different granularity, we can get different classification outputs on neighborhood classifiers or neighborhood attribute reducts. Then, weights of granularity are learned by optimizing margin

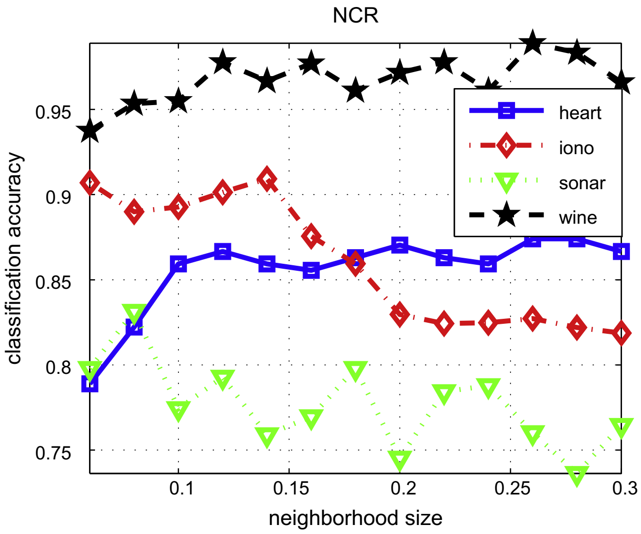

Fig. 3. Accuracy of NCR in different δ neighborhood separable subspaces.

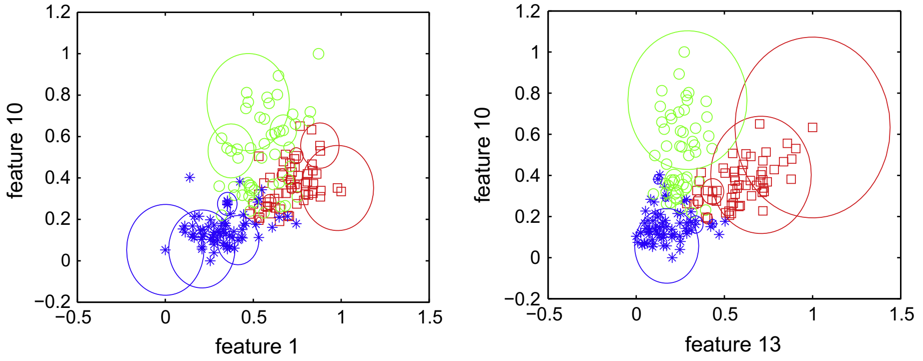

Fig. 4. Learned rules in different feature subspaces.

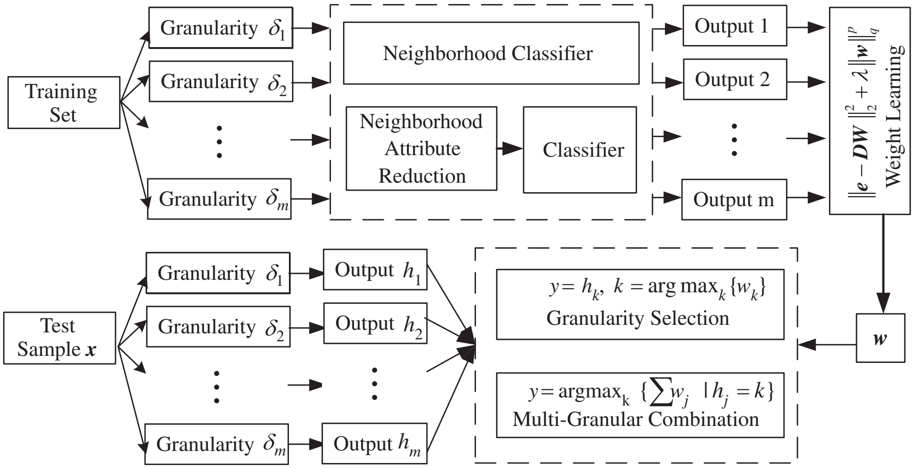

Fig. 5. Flow chart of granularity selection and combination.

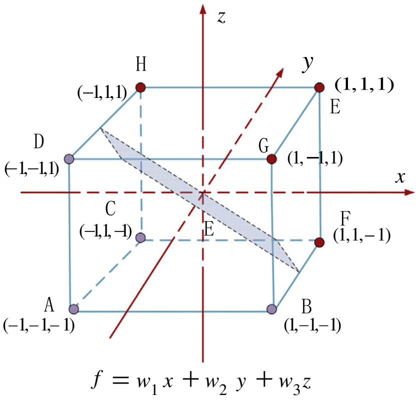

Fig. 6. Demo of multi-granular combination.

distribution. The optimal granularity can be selected according to the weights. Additionally, we can use the weights to combine the outputs from different granularity.

3.1. Ensemble margin

Granularity evaluation and combination can be considered as a special classification task. As shown in Fig. 6, we consider a binary classification task $\{1, -1\}$ in three different granular spaces $\{x, y, z\}$. The outputs from these granular models belong to one of the eight vertexes. We expect to find a classification plane $f = sgn(w_1x + w_2y + w_3z)$, which crosses the origin of coordinates, to correct classify the samples. For the combination task in Fig. 6, if samples on vertexes $\{A,B,C,D\}$ belong to the first class and samples on vertexes $\{E,F,G,H\}$ belong to the second class, there are a set of planes which can correctly classify the samples. So there is a question, i.e., which plane is optimal. More specially, if samples on vertexes $\{A,B,C,F\}$ belong to the first class and samples on vertexes $\{E,D,G,H\}$ belong to the second class, we can correctly classify the samples only using granularity z. Inspired by feature selection [3] and classifier pruning [35], we can learn the weights of different granularity to evaluate the granularity importance.

Given $\boldsymbol{S} = \{(\boldsymbol{x}_i, y_i)\}$, $i = 1, 2, \ldots, n$, $y_i \in \{+1, -1\}$ and m granularity, the classification results in m different granularity spaces are $\boldsymbol{H} \in \Re^{n\times m}$, where $\boldsymbol{w} = \langle w_1, w_2, \ldots, w_m\rangle$ is the weight vector of different granularity.

Definition 13. For sample $\boldsymbol{x}_i \in \boldsymbol{S}$, the classification outputs in m different granularity spaces are $\{h_{ij}\}$, $j = 1, 2, \ldots, m$. The discriminant function is $f = \text{sgn}\left(\sum_{j=1}^{m} w_j h_{ij}\right)$. The margin of sample $\boldsymbol{x}_i$ is defined as

$$\rho(\boldsymbol{x}_i) = y_i \sum_{j=1}^{m} w_j h_{ij} \tag{12}$$

Obviously, if $\rho(\boldsymbol{x}_i) > 0$, $\boldsymbol{x}_i \in \boldsymbol{S}$ is correctly classified; otherwise, it is misclassified.

Definition 14. For multi-class classification, the classification outputs in m different granular spaces are $\{h_{ij}\}$, $j = 1, 2, \ldots, m$. The matrix $\boldsymbol{D} = \{d_{ij}\}_{n\times m}$ is defined as:

$$d_{ij} = g(y_i, h_{ij}) = \begin{cases} +1, & \text{if } y_i = h_{ij}, \\ -1, & \text{if } y_i \neq h_{ij}. \end{cases} \tag{13}$$

$d_{ij} = +1$ means that $\boldsymbol{x}_i$ is correctly classified in the jth granular space; otherwise, it is misclassified. Obviously this definition is fit for binary classification.

Definition 15. For $\boldsymbol{x}_i$, the classification outputs in m different granular spaces are $\{h_{ij}\}$, $j = 1, 2, \ldots, m$. The ensemble margin of $\boldsymbol{x}_i$ is defined as

$$\rho(\boldsymbol{x}_i) = \sum_{j=1}^{m} w_j d_{ij}. \tag{14}$$

Ensemble margin reflects the misclassification degree in classifier fusion. We should make the ensemble margin as large as possible by weight learning. Margin maximization is usually converted into a loss minimization problem [9,25,26].

Definition 16. For each sample $\boldsymbol{x}_i \in \boldsymbol{S}$, ensemble margin of $\boldsymbol{x}_i$ is $\rho(\boldsymbol{x}_i)$. Then the ensemble loss of $\boldsymbol{x}_i$ is

$$l_{\boldsymbol{x}_i} = l(\rho(\boldsymbol{x}_i)) = l\left(\sum_{j=1}^{m} w_j d_{ij}\right) \tag{15}$$

where l is a loss function.

Squared loss is widely used in support vector machine [28], spare coding [31] and least square regression [20]. If we use squared loss, then weights of granularity should satisfy $\sum_{j=1}^{m} w_j = 1$. If $\boldsymbol{x}_i$ is correctly classified in all the granular spaces, ensemble margin is 1; if $\boldsymbol{x}_i$ is misclassified in all of the granular spaces, ensemble margin is -1.

For a sample set $\boldsymbol{S}$, the ensemble square loss is

$$l(\boldsymbol{S}) = \sum_{i=1}^{n} l_{\boldsymbol{x}_i} = \sum_{i=1}^{n} (1 - \rho(\boldsymbol{x}_i))^2 = \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{m} w_j d_{ij}\right)^2 = \|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \tag{16}$$

where $\boldsymbol{e}$ is a vector whose elements are 1 and the length is n.

3.2. Margin distribution optimization

To learn the optimal granularity weights, we should minimize the ensemble loss in Eq. (16). Whereas, there may be many solutions that can minimize the loss for a given task, as illustrated in Fig. 6. In [22], Saharon et al. showed that AdaBoost approximately minimizes its loss function with l_1-regularization imposed on the weights. The work in [26] showed that AdaBoost optimizes margin distribution rather than minimum margin. Additionally, Shawe–Taylor gave the bound on generalization error based on margin distribution for linear classifiers ($f = \boldsymbol{wx} + b$) and showed that both the square loss (when $\sum_{j=1}^{s} w_j = 1$ and $\boldsymbol{x} \in \{+1, -1\}$) and the norm of $\boldsymbol{w}$ should be minimized to improve the generalization ability [24].

To get better margin distribution, we should minimize the ensemble square loss with l_p-norm regularization imposed on the weight vector to get a stable solution. Hence, we can construct the optimization objective as follows.

Table 1
The algorithm of granularity weight learning.

```
Input: A set of samples $\boldsymbol{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
Output: Granularity weights $\boldsymbol{w}$
Step 1: Choose $m$ granularity $\delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$
Step 2: Get classification outputs $\{h_{ij}\}$ for $m$ granularity
Step 3: Get the decision matrix
    $d_{ij} = g(y_i, h_{ij}) = \begin{cases} +1, & \text{if } y_i = h_{ij} \\ -1, & \text{if } y_i \neq h_{ij} \end{cases}$
Step 4: Learn granularity weights
    $\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \|\hat{\boldsymbol{e}} - \widehat{\boldsymbol{D}}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_{l_p}$
```

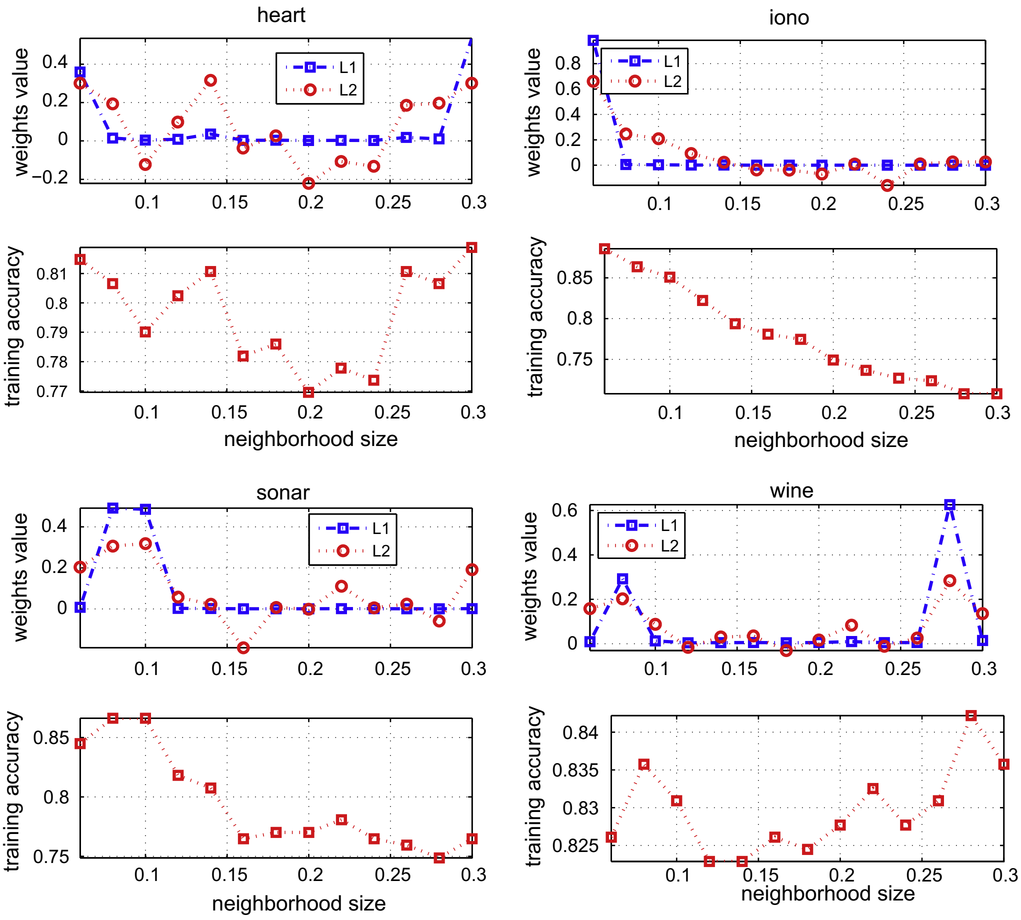

Fig. 7. Weights of different granularity in NEC.

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_{l_p} \ s.t. \sum_{j=1}^{m} w_j = 1, \tag{17}$$

where λ is the regularization parameter.

For the constraint condition $\sum_{j=1}^{m} w_j = 1$, it is equal to $1 = \boldsymbol{e}\boldsymbol{w}$, where $\boldsymbol{e} = [1; 1; \ldots; 1]$ is a column vector, then

$$\|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 = \|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w} + 1 - \boldsymbol{e}\boldsymbol{w}\|_2^2 = \|[\boldsymbol{e}; 1] - [\boldsymbol{D}; \boldsymbol{e}]\boldsymbol{w}\|_2^2. \tag{18}$$

Let $\hat{\boldsymbol{e}} = [\boldsymbol{e}; 1]$, $\widehat{\boldsymbol{D}} = [\boldsymbol{D}; \boldsymbol{e}]$, then we can get

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \{\|\hat{\boldsymbol{e}} - \widehat{\boldsymbol{D}}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_{l_p}\}. \tag{19}$$

Table 2
Accuracy of KNN, NEC and LSVM.

Data	NEC (0.1)	NEC (0.15)	KNN (1)	KNN (3)	LSVM
Heart	79.3 ± 4.7	79.6 ± 6.6	75.6 ± 10.1	79.3 ± 8.2	84.1 ± 4.6
Iono	83.5 ± 4.7	76.7 ± 5.5	86.4 ± 4.9	85.8 ± 7.1	87.6 ± 6.5
Sonar	82.7 ± 5.5	78.8 ± 7.6	87.1 ± 7.6	81.3 ± 7.8	77.9 ± 7.1
Wine	97.2 ± 3.0	97.8 ± 3.9	94.9 ± 5.1	96.6 ± 2.9	98.9 ± 2.3
Average	85.6	83.3	85.9	85.8	87.1

Table 3
Accuracy of GSC_MD for neighborhood classifier.

Data	l_1WV	l_1MV	l_1BG	l_2WV	l_2MV	l_2BG	Sum
Heart	81.1 ± 8.1	84.4 ± 5.2	80.7 ± 7.8	81.1 ± 5.4	84.8 ± 5.1	80.7 ± 6.9	81.5 ± 5.7
Iono	87.0 ± 3.7	87.0 ± 3.7	87.0 ± 3.6	87.3 ± 3.2	87.0 ± 3.6	87.0 ± 3.6	75.8 ± 6.8
Sonar	85.2 ± 6.5	87.0 ± 5.5	84.7 ± 7.6	84.1 ± 6.7	85.2 ± 6.9	81.2 ± 11.2	79.3 ± 6.0
Wine	98.3 ± 2.8	98.3 ± 2.8	97.2 ± 3.0	97.7 ± 3.0	98.3 ± 2.8	97.7 ± 3.0	97.7 ± 2.7
Average	87.9	89.2	87.4	87.6	88.8	86.6	83.6

When l_1-norm regularization is imposed on $\boldsymbol{w}$, the objective function can be written as

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}}\{\|\hat{\boldsymbol{e}} - \widehat{\boldsymbol{D}}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1\}. \tag{20}$$

There are several fast l_1-minimization approaches: Gradient Projection, Homotopy, Iterative Shrinkage-Thresholding, Proximal Gradient, and Augmented Lagrange Multiplier (ALM) [36]. We use l_1_ls to solve this problem [10]. In this case, learned weights are sparse and it is suitable for granularity selection.

When l_2-norm is used, we can directly get $\boldsymbol{w} = \left(\widehat{\boldsymbol{D}}^T\widehat{\boldsymbol{D}} + \lambda\boldsymbol{I}\right)^{-1}\widehat{\boldsymbol{D}}^T\hat{\boldsymbol{e}}$, where $\boldsymbol{I}$ is an identity matrix.

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}}\left\{\|\hat{\boldsymbol{e}} - \widehat{\boldsymbol{D}}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2\right\}. \tag{21}$$

After learning the weights, we use the weights to evaluate and combine different granular models. As to the combination, we can use linear weighted combination or select the granular models with large weights. We denote this method by GSC_MD. The algorithm is formulated in Table 1.

The time complexity of the proposed method consists of two parts. We get the decision matrix and learn the granularity weights. The first step varies for different classification algorithms. As to the second step, it is known that sparse coding with an $m \times n$ -sized dictionary has a computational complexity of $O(m^2n^\varepsilon)$, where $\varepsilon \geqslant 1.2$, m is the dimensionality of signal feature, and n is the number of dictionary atoms. For l_2-norm, the time complexity is $O(mn)$.

4. Experiment analysis

To show the effectiveness of the proposed method, we firstly show the granularity weights in Section 4.1. Then for granularity-sensitive classifiers, we use NEC and KNN as an example to show the superiority of the proposed GSC_MD in Section 4.2. Finally, the experiment for multi-granularity subspaces based classification is conducted.

4.1. Granularity weights

To show the process of granularity weight learning, we give the training accuracy and weights corresponding to each granularity. In Fig. 7, granularity weights for neighborhood classifier are given on four datasets. L1 and L2 represent l_1-norm and l_2-norm regularization respectively. From Fig. 7 we can see that greater weights are assigned to granularities with higher training accuracy. Additionally, weights with l_1-norm regularization are more sparse than l_2-norm regularization.

4.2. Multi-granularity classifier

In this section, we test the proposed granularity selection and combination method for granularity-sensitive classifiers (e.g., NEC and KNN). After weights learning, we have several ways to utilize the weights. Firstly, we can select the granularity with the greatest weight. In this way, we can adaptively select the optimal neighborhood size and K. Secondly, we can combine the classification results of different granularity using the weights directly. Thirdly, similar to feature selection, we can use the top k granularity making the classification accuracy the greatest. BG (Best Granularity), WV (Weighted Voting) and MV (Majority Voting) represent the three cases respectively.

Table 4
Accuracy of GSC_MD for KNN.

Data	l_1WV	l_1MV	l_1BG	l_2WV	l_2MV	l_2BG	Sum
Heart	81.1 ± 5.9	82.6 ± 5.3	81.1 ± 5.9	81.1 ± 5.6	82.2 ± 4.6	80.0 ± 6.1	81.5 ± 5.5
Iono	85.8 ± 6.2	87.5 ± 5.7	86.4 ± 4.9	85.8 ± 6.5	87.5 ± 5.7	86.4 ± 4.9	83.8 ± 4.7
Sonar	87.1 ± 7.6	88.5 ± 7.2	87.0 ± 7.6	86.6 ± 7.8	88.5 ± 7.2	87.0 ± 7.6	77.9 ± 6.1
Wine	97.7 ± 3.0	97.7 ± 3.0	97.1 ± 4.3	97.2 ± 3.0	97.7 ± 3.0	97.7 ± 3.0	97.0 ± 3.1
Average	88.0	89.3	87.9	87.7	89.0	87.8	85.0

Table 5
Accuracy of neighborhood attribute reducts, Bagging and Adaboost.

Data	NN (0.1)	NN (0.15)	Bagging	Ada-Boost
Heart	74.4 ± 11.2	77.4 ± 7.5	82.2 ± 7.8	78.9 ± 6.5
Iono	89.8 ± 5.0	90.1 ± 5.5	92.1 ± 5.9	94.3 ± 4.3
Sonar	68.7 ± 7.6	79.8 ± 4.6	79.8 ± 7.0	86.4 ± 8.2
Wine	95.5 ± 2.4	93.2 ± 4.1	95.4 ± 3.8	97.2 ± 3.1
Average	82.1	85.1	87.4	89.2

Table 6
Accuracy of GSC_MD on 1-NN classifier.

Data	l_1WV	l_1MV	l_2WV	l_2MV	Sum
Heart	78.5 ± 12.9	84.4 ± 6.9	80.7 ± 11.3	84.9 ± 7.1	81.1 ± 9.1
Iono	91.3 ± 5.4	92.9 ± 5.3	91.9 ± 5.4	93.3 ± 5.6	91.5 ± 5.2
Sonar	85.0 ± 5.9	90.9 ± 4.8	83.6 ± 6.1	90.9 ± 4.8	81.2 ± 7.8
Wine	95.4 ± 3.8	98.9 ± 2.3	97.8 ± 2.8	98.9 ± 2.3	98.3 ± 2.6
Average	87.6	91.8	88.5	92	88

Table 7
Accuracy of neighborhood covering reduction.

Data	NCRST	NCRSS	NCRRT	NCRRS
Heart	77.8 ± 10.8	81.5 ± 10.9	81.9 ± 12.2	85.9 ± 9.7
Iono	83.8 ± 9.3	84.7 ± 9.7	86.4 ± 3.8	89.3 ± 5.5
Sonar	69.8 ± 7.6	75.0 ± 7.7	70.6 ± 14.4	77.4 ± 8.8
Wine	96.5 ± 4.2	97.2 ± 3.0	90.3 ± 5.6	95.5 ± 2.4
Average	82.0	84.6	82.3	87.0

The classification accuracy of neighborhood classifier, KNN and LSVM is shown in Table 2. For neighborhood classifier, the recommended neighborhood sizes are 0.1 and 0.15 [6]. In KNN classification, K is usually set as 1 and 3. Hence, we give the classification accuracy of the two usual granularity. As shown in Table 2, for neighborhood classifier, if we could not choose the proper neighborhood size, the classification accuracy would vary greatly.

Tables 3 and 4 show the classification performances of the best granularity and multiple granularity combination for neighborhood classifier and KNN classifier. Compared to the recommended granularity, classification accuracy of neighborhood classifier and KNN is greatly improved by granularity selection and combination. For granularity selection, its performance is similar to weighted combination, which validates that granularity with the greatest weight plays an important role in combination. For combination technique, sum (i.e., combine all the outputs by majority voting) is much worse than l_1WV and l_2WV, which proves the effectiveness of the proposed granularity combination method. Besides, the classification performance can be further improved by l_1MV and l_2MV. Although classification accuracy is obtained by adding the ranked granularity one by one on the test set, it indicates that if we can properly choose the number of the ranked granularity, we can get much better classification performance.

4.3. Multi-granularity subspaces based classification

Neighborhood size has a great impact on the approximation and discrimination ability of neighborhood information granules. The approximation ability of neighborhood granules affects neighborhood dependency. Hence, neighborhood attribute reduction is affected by δ. Information of different attribute reducts is complementary to each other. In this section, we test the performance of GSC_MD for multiple granularity subspaces combination.

Table 8
Accuracy of GSC_MD for NCR.

Data	Classifier	l_1WV	l_1MV	l_2WV	l_2MV	Sum
Heart	NCRST	81.5 ± 7.0	85.2 ± 6.0	81.1 ± 5.6	84.4 ± 5.5	79.6 ± 5.0
	NCRSS	81.1 ± 4.4	86.7 ± 4.7	83.0 ± 4.3	87.4 ± 4.3	81.9 ± 5.1
	NCRRT	84.4 ± 6.2	86.3 ± 6.5	84.1 ± 6.5	86.7 ± 6.6	83.0 ± 6.1
	NCRRS	89.3 ± 4.1	91.9 ± 4.2	90.4 ± 3.6	91.9 ± 4.2	90.0 ± 4.6
Iono	NCRST	89.5 ± 6.7	91.2 ± 6.0	89.5 ± 6.3	91.2 ± 6.0	88.1 ± 8.9
	NCRSS	88.7 ± 7.8	90.4 ± 7.3	88.7 ± 6.6	90.7 ± 7.4	88.1 ± 8.5
	NCRRT	88.7 ± 7.4	91.0 ± 4.8	89.9 ± 6.1	90.7 ± 4.9	85.0 ± 5.4
	NCRRS	92.4 ± 5.7	93.8 ± 4.1	93.5 ± 5.4	94.4 ± 4.2	88.4 ± 5.9
Sonar	NCRST	75.0 ± 9.1	83.2 ± 6.8	76.9 ± 8.0	83.6 ± 6.1	75.0 ± 8.4
	NCRSS	83.7 ± 6.4	90.8 ± 4.3	84.6 ± 3.8	91.3 ± 3.9	87.5 ± 5.2
	NCRRT	75.5 ± 8.8	82.6 ± 6.7	74.9 ± 9.0	82.6 ± 6.7	76.9 ± 7.6
	NCRRS	85.0 ± 4.9	89.9 ± 4.3	86.5 ± 5.2	89.9 ± 4.3	86.5 ± 3.2
Wine	NCRST	96.6 ± 2.9	98.9 ± 2.3	96.0 ± 2.7	98.9 ± 2.3	94.9 ± 3.2
	NCRSS	96.5 ± 5.0	98.8 ± 2.5	97.2 ± 3.0	98.8 ± 2.5	96.6 ± 2.9
	NCRRT	97.8 ± 2.9	98.9 ± 2.3	97.2 ± 2.9	98.9 ± 2.3	97.8 ± 2.9
	NCRRS	98.3 ± 2.7	99.4 ± 1.8	98.9 ± 2.3	99.4 ± 1.8	98.9 ± 2.3
Average	NCRST	85.6	89.6	85.9	89.5	84.4
	NCRSS	87.5	91.7	88.4	92.1	88.5
	NCRRT	86.6	89.7	86.5	89.7	85.7
	NCRRS	91.3	93.7	92.3	93.9	91.0

The neighborhood sizes are set as 0.1 and 0.15 to get neighborhood separable subspaces. Then the classification of nearest neighbor classifier in δ neighborhood separable subspaces is tested, as illustrated in Table 5. The ensemble learning methods including Bagging and AdaBoost are also listed for comparison. From the Table 5, we can see that neighborhood size greatly affects the classification performance of neighborhood reducts. Besides, the performance of Bagging and Adaboost is much better than that on single reduct.

The classification accuracy of GSC_MD for nearest neighbor classifier is shown in Table 6. The proposed method is much better than classification in single neighborhood separable subspace and is comparable to Adaboost.

We also test the method of multi-granularity subspaces for rule learning. Multiple rule sets are learned in different neighborhood separable subspaces. As shown in Table 7, neighborhood size for attribute reduction is set as 0.15 and the classification accuracy of NCR is given. In Table 8, multiple granularity combination results are given. Obviously, GSC_MD is much better than rule learning in neighborhood separable subspace for fixed neighborhood size.

5. Conclusions and future work

Neighborhood rough set is a granularity sensitive granular computing model. We can train multiple models from different granularity, which leads to diverse granular views of a learning task. As base classifiers trained in different granular spaces are complementary, in this paper we explore ensemble learning techniques to solve the granularity selection and combination problem. By optimizing margin distribution, we learn the weights of different granularity. Then weights are used for granularity selection and combination. Experimental analysis shows the proposed methods are effective and the derived models produce competent performances compared with other classical techniques.

In this work, squared loss is used for training weights of different granularity. In fact, there are several other loss functions used in classification and regression, such as logistic loss and exponential loss. They can also be tried in this task. Moreover, although we just discuss the issue of granularity selection for neighborhood rough sets, the idea can also be used in fuzzy rough sets.

Acknowledgments

This work is partly supported by National Program on Key Basic Research Project under Grant 2013CB329304, National Natural Science Foundation of China under Grants 61222210 and 61105054 and New Century Excellent Talents in University under Grant NCET-12-0399.

References

[1] Y. Du, Q. Hu, P. Zhu, P. Ma, Rule learning for classification based on neighborhood covering reduction, Information Sciences 181 (2011) 5457–5467.

[2] Y. Freund, R. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: Computational Learning Theory, Springer, 1995, pp. 23–37.

[3] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection-theory and algorithms, in: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 43.

[4] B. Heisele, P. Ho, T. Poggio, Face recognition with support vector machines: global versus component-based approach, in: ICCV 2001, vol. 2, IEEE, 2001, pp. 688–694.

[5] Q. Hu, W. Pan, S. An, P. Ma, J. Wei, An efficient gene selection technique for cancer recognition based on neighborhood mutual information, International Journal of Machine Learning and Cybernetics 1 (2010) 63–74.
[6] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (2008) 3577–3594.
[7] Q. Hu, D. Yu, Z. Xie, Neighborhood classifiers, Expert Systems with Applications 34 (2008) 866–876.
[8] Q. Hu, D. Yu, Z. Xie, Numerical attribute reduction based on neighborhood granulation and rough approximation, Journal of Software 19 (2008) 640–649.
[9] Q. Hu, P. Zhu, Y. Yang, D. Yu, Large-margin nearest neighbor classifiers via sample weight learning, Neurocomputing 74 (2011) 656–660.
[10] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale l1-regularized least squares, IEEE Journal of Selected Topics in Signal Processing 1 (2007) 606–617.
[11] M. Kryszkiewicz, Rough set approach to incomplete information systems, Information Sciences 112 (1998) 39–49.
[12] Y. Liao, V. Vemuri, Use of k-nearest neighbor classifier for intrusion detection1, Computers & Security 21 (2002) 439–448.
[13] G. Lin, J. Liang, Y. Qian, Multigranulation rough sets: from partition to covering, Information Sciences (2013).
[14] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, University of California, Department of Information and Computer Science, Irvine, CA, 1998. <http://www.ics.uci.edu/ mlearn/MLRepository.html>.
[15] Z. Pawlak, Rough set approach to knowledge-based decision support, European Journal of Operational Research 99 (1997) 48–57.
[16] W. Pedrycz, Granular Computing: An Emerging Paradigm, vol. 70, Physica Verlag, 2001.
[17] W. Pedrycz, Granular Computing: Analysis and Design of Intelligent Systems, vol. 13, CRC Press, 2013.
[18] Y. Qian, J. Liang, Y. Yao, C. Dang, Mgrs: a multi-granulation rough set, Information Sciences 180 (2010) 949–970.
[19] A. Radzikowska, E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126 (2002) 137–155.
[20] J. Ramsey, Tests for specification errors in classical linear least-squares regression analysis, Journal of the Royal Statistical Society, Series B (Methodological) 31 (1969) 350–371.
[21] G. Ratsch, T. Onoda, K. Muller, Soft margins for adaboost, Machine Learning 42 (2001) 287–320.
[22] S. Rosset, J. Zhu, T. Hastie, Boosting as a regularized path to a maximum margin classifier, The Journal of Machine Learning Research 5 (2004) 941–973.
[23] R. Schapire, Y. Freund, P. Bartlett, W. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, The Annals of Statistics 26 (1998) 1651–1686.
[24] J. Shawe-Taylor, N. Cristianini, Robust bounds on generalization from the margin distribution, in: 4th European Conference on Computational Learning Theory, Citeseer, 1998.
[25] C. Shen, H. Li, Boosting through optimization of margin distributions, IEEE Transactions on Neural Networks 21 (2010) 659–666.
[26] C. Shen, H. Li, On the dual formulation of boosting algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence (2010) 2216–2231.
[27] X. Tan, S. Chen, Z. Zhou, F. Zhang, Face recognition from a single image per person: a survey, Pattern Recognition 39 (2006) 1725–1745.
[28] T. Van Gestel, J. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, Machine Learning 54 (2004) 5–32.
[29] S. Wang, X. Li, S. Zhang, J. Gui, D. Huang, Tumor classification by combining pnn classifier ensemble with neighborhood rough set based gene reduction, Computers in Biology and Medicine 40 (2010) 179–189.
[30] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.
[31] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 210–227.
[32] W. Wu, Y. Leung, Theory and applications of granular labelled partitions in multi-scale decision tables, Information Sciences 181 (2011) 3878–3897.
[33] W. Wu, J. Mi, W. Zhang, Generalized fuzzy rough sets, Information Sciences 151 (2003) 263–282.
[34] W. Wu, W. Zhang, Neighborhood operator systems and approximations, Information Sciences 144 (2002) 201–217.
[35] Z. Xie, Y. Xu, Q. Hu, P. Zhu, Margin distribution based bagging pruning, Neurocomputing 85 (2012) 11–19.
[36] A. Yang, S. Sastry, A. Ganesh, Y. Ma, Fast l1-minimization algorithms and an application in robust face recognition: a review, in: ICIP 2010, IEEE, 1849. pp. 1849–1852.
[37] Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Information Sciences 111 (1998) 239–259.
[38] Y. Yao, Granular computing: basic issues and possible solutions, in: Proceedings of the 5th Joint Conference on Information Sciences, vol. 1, Citeseer, 2000, pp. 186–189.
[39] Y.Yao, Granular computing, Computer Science 31 (2004) 4–10.
[40] X. Zhao, Q. Hu, Y. Lei, M. Zuo, Vibration-based fault diagnosis of slurry pump impellers using neighbourhood rough set models, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 224 (2010) 995–1006.
[41] P. Zhu, L. Zhang, Q. Hu, S. Shiu, Multi-scale patch based collaborative representation for face recognition with margin distribution optimization, in: ECCV 2012, vol. 7572, 2012, pp. 822–835.
[42] W. Zhu, Topological approaches to covering rough sets, Information Sciences 177 (2007) 1499–1508.
[43] W. Zhu, F. Wang, Reduction and axiomization of covering generalized rough sets, Information Sciences 152 (2003) 217–230.