A Rough Set Method for Co-training Algorithm

Donghai Guan[1*] and Weiwei Yuan[2]

[1]*College of Automation, Harbin Engineering Univ., Harbin, China*
[2]*College of Computer Science & Technology, Harbin Engineering Univ., Harbin, China*

dhguan@gmail.com, yuanweiwei@hrbeu.edu.cn

Abstract

In recent years, semi-supervised learning has been a hot research topic in machine learning area. Different from traditional supervised learning which learns only from labeled data; semi-supervised learning makes use of both labeled and unlabeled data for learning purpose. Co-training is a popular semi-supervised learning algorithm which assumes that each example is represented by two or more redundantly sufficient sets of features (views) and additionally these views are independent given the class. To improve the performance and applicability of co-training, ensemble learning, such as bagging and random subspace has been used along with co-training. In this work, we propose to use the rough set based ensemble learning method with co-training algorithm (RSCO). Inherited the inherent characteristics of rough set, ensemble learning is expected to meet both the diversity and accuracy requirement. Finally experimental results on the UCI data sets demonstrate the promising performance of RSCO.

Keywords*: Semi-supervised learning, Co-training, Ensemble learning*

1. Introduction

In many practical applications, labeled data are difficult (or costly) to obtain since labeling is a labor-intensive and time-consuming process. Conversely, unlabeled data are far easier to obtain because they do not need human labeling effort. Due to this fact, we need techniques which can learn from both labeled and unlabeled data.

In machine learning literature, mainly there are three paradigms can combine the labeled and unlabeled data for learning: semi-supervised learning [1-5], transductive learning [6-9] and active learning [10-12]. Semi-supervised learning attempts to use unlabeled data together with labeled data to train better classifiers by either modifying or reprioritizing hypotheses obtained from labeled data only.

Co-training is an important algorithm of semi-supervised learning which is devel-oped by Blum and Mitchell [13]. It is proposed for the problem in which the features of each example can be partitioned into two distinct views (two disjoint feature sets). Generally co-training works well only if the two feature sets are sufficient an indepen-dent.

There exists several works which extend standard co-training algorithm to improve its performance and applicability. The Co-EM algorithm [14] extends the original bootstrap approach of the co-training algorithm to operate simultaneously on all unlabeled samples in an iterative batch mode. Democratic Co-learning proposed by Goldman and Zhou [15] uses two

[*]Donghai Guan is the corresponding author of this paper.

different supervised learning algorithms to cooperate, and cross-validation is taken to label the unlabeled data. In recent years, people have proposed to use ensemble learning along with co-training to boost the performance of co-training. Tri-Training [16] uses bootstrap (bagging) approach to train three classifiers. RASCO [17] uses the random subspace of the feature space and trains ensemble classifiers based on each subspace.

In this paper, we propose a novel co-training algorithm. It combines the rough set theory with co-training (thus, called RSCO). RSCO mainly consists of three phases. The first phase is to obtain multiple subsets of features which have the same approxi-mation ability as the whole features based on the rough set theory. Such a subset of feature is called a reduct in rough set theory. Then the second phase is to train ensem-ble classifiers with these multiple reducts. The third phase is to combine the trained ensemble classifiers with co-training.

In essence, RSCO is a kind of algorithm which combines ensemble learning with co-training. In this sense, it is similar to RASCO and Tri-Training. However, they are different on the ways to generate the diversities in ensemble learning. Tri-Training adopts "bagging" method to generate different sets of training data to achieve the diversity among ensemble classifiers. RASCO is based on random subspace which promotes the diversity through feature set manipulation instead of training set manipulation. RSCO promotes the diversity by training classifiers with different reducts.

Reduct is the key concept in rough set. It is the subset of the whole feature space which has the same discernibility ability as the whole features. Generally one data set consists of multiple reducts referring to different subsets of features.

Compared with RASCO and Tri-Training, RSCO is more similar to RASCO since both of them generate the diversity among ensemble classifiers by feature set manipulation. In RASCO, the randomly subset of feature is used to train each classifier. Although diversity is achieved, we cannot ensure the quality (*i.e.* accuracy) of each classifier since it is trained on the randomly selected features. In this sense, RSCO is superior to RASCO. On one hand, the different reducts of rough set generate the diversity among multiple classifiers. On the other hand, the quality of each classifier is ensured because it is trained on the reduct which has the same approximation ability as the whole feature.

We use a set of benchmark data sets from the Machine Learning Database Repository to test the performance of RSCO. Experimental results demonstrate the good performance of RSCO.

The rest of the paper is organized as follows. Section 2 describes the related works on co-training. Then we present the knowledge related to rough set in Section 3. In Section 4 we present our proposed rough set based co-training method. Section 5 discusses the experiments by using benchmark data. Section 6 summarizes our contributions and future work.

2. Related Works

Co-training [13] (Table 1) is a well-known semi-supervised learning algorithm which exploits unlabeled in addition to labeled training data for classifier learning. It has been widely used in many domains with two independent views [18-19]. Although co-training shows the good performance for these applications, they cannot be directly applied on the applications where multiple independent views are not available.

There are some works which aim to extend co-training's applicability for the applications where multiple independent views are not available. In some works, co-training was applied by artificially splitting the available feature set into two views [19-20]. In addition to these

works, in recently years, it was found that co-training's performance and applicability could be effectively improved through combining ensemble learning.

Table 1. Standard Co-Training Algorithm

Algorithm 1: Standard Co-training Algorithm

Input:

L (set of m labeled data)

U (set of unlabeled data)

V1,*V2* (two views of data)

numIter (maximal number of co-training iterations)

baseAlg (base learning algorithm)

$\{\Pr_k\}_{k=1}^{K}$ (prior probability of class k)

1: $h_1^{(0)} \leftarrow baseAlg(V_1(L))$, $h_2^{(0)} \leftarrow baseAlg(V_2(L))$

2: for $i = 1$ to *numIter*

3: if U is empty

4: $numIter = i - 1$ and abort loop

5: end if

6: Apply $h_1^{(t-1)}$ on U

7: Select a subset S_1 as follows: for each class k, select the $n_k \propto \Pr_k$ most confident examples assigned to class k

8: Move S_1 from U to L

9: Apply $h_2^{(t-1)}$ on U

10: Select a subset S_2 as follows: for each class k, select the $n_k \propto \Pr_k$ most confident examples assigned to class k

11: Move S_2 from U to L

12: Retrain classifiers $h_1^{(t)}$ and $h_2^{(t)}$ using the new L

$h_1^{(t)} \leftarrow baseAlg(V_1(L))$ $h_2^{(t)} \leftarrow baseAlg(V_2(L))$

13: end for

14: return combination of the predictions of $h_1^{(T)}$ and $h_2^{(T)}$

In Ref [16], Tri-Training algorithm is proposed. In Tri-Training, initially three classifiers are trained on bootstrap subsamples generated from the original labeled training set. These classifiers are then refined during the Tri-Training process, and the final hypothesis is produced via majority voting. The construction of the initial classifiers is trained from the labeled data with Bagging. At each iteration, an unlabeled data is added to the training set of a classifier if the other two classifiers agree on their prediction under certain conditions. Tri-Training is more applicable than co-training because it neither requires different views nor does it depend on different supervised learning algorithms as in Ref [13].

Another similar work is RASCO [17] which combines random subspace with co-training. RASCO chooses the random subspace of the feature space to train each classifier. Each subspace can be seen as a view of the example. The main idea of RASCO is that different classifiers are sensitive to different features, and can complement each other. Then these classifiers are used for co-training to enlarge the training data set of the base classifiers.

An important factor that affects the performance of ensemble learning based co-training (Table 2) is how to measure the confidence of a given unlabeled example which deter-mines its probability of being selected. In this study, the confidence of labeling is estimated through consulting the number of classifiers which give the same label for an unlabeled data. For example, for a two-class classification problem, suppose there are five classifiers. For an unlabeled data, four classifiers label it as class 1, and one classifier label it as class 2. Then in Alg. 2, it is given the label of class 1 with confidence 4/5.

Table 2. Co-Training with Ensemble Learning

Algorithm 2: Co-training with ensemble learning

Input:

L (set of m labeled data)

U (set of unlabeled data)

numIter (maximal number of co-training iterations)

ensembleAlg (ensemble learning algorithm)

numClassifier (number of classifiers in ensemble learning)

baseAlg (base learning algorithm)

$\{\Pr_k\}_{k=1}^{K}$ (prior probability of class k)

1: Construct multiple classifiers $H^{(0)} = ensembleAlg(L, baseAlg, numClassifer)$

2: Create a set U' of examples chosen randomly from U without replacement

3: for $i = 1$ to *numIter*

4: if U' is empty

5: $numIter = i - 1$ and abort loop

6: end if

7: $\forall x_j \in U'$, measure $Confidence(x_j, H^{(t-1)}, K)$

8: Select a subset S as follows: for each class k , select the $n_k \propto \Pr_k$ with the highest confidence assigned to class k

9: Set $U' = U' - S$ and $L = L \bigcup S$

10: Replenish U' with $|S|$ training examples chose at random from U

11: $H^{(t)} = ensembleAlg(L, baseAlg, numClassifier)$

12: end for

13: return multiple classifiers $H^{(T)}$

One important reason of the success of combining ensemble learning with co-training is the creation of diversity among a set of classifiers by exploiting different techniques: training set manipulation by Bagging [21] or feature set manipulation by Random Subspace [22]. An ensemble consists of a set of individual classifiers whose predictions are combined when classifying a given sample. In Ref [23], Dietterich has pointed out two essential conditions for an effective ensemble: error diversity and the accuracy of its member classifiers. Two classifiers are diverse if they produce different errors for a given set of instances. Bagging relies on the available training data for encouraging diversity. So if the size of the training set is small, then the diversity among the ensemble members will be limited. Consequently, the ensemble error reduction will be small. On the other hand, the diversity of Random Subspace is enough for the training set with small size since different subsets of features are used to train the classifiers. The work in [17] has proved the better performance of RASCO com-pared with Bagging based co-training.

Is random subspace algorithm perfect for co-training? Intuitively RSCSO meets the requirement of diversity. However, the accuracy of each classifier is hard to ensure since they are trained on the randomly selected features. On the whole, the "random feature selection" of random subspace can ensure the diversity, but not the accuracy.

To struggle with both diversity and accuracy, we propose to use rough set-based ensemble learning in co-training. The background knowledge of rough set theory and our proposed method will be presented in the next section.

3. Ensemble Learning based on Rough Set

Recent researches showed that a well-devised feature selection algorithm would significantly improve the efficiency and accuracies of subspace ensembles because attribute reduction lessens the impact of the "curse of dimensionality" and speeds up the training and test process [24]. The key problem of this ensemble method is how to get a set of attribute subset with good predicting power. Rough set theory, which was introduced by Pawlak [25-27], has attracted much attention from AI society. In the rough set framework, reducts are minimal attributes subsets which keep the discernibility of the original data and have no redundant attributes. There generally exist multiple reducts for a given data set. All the reducts can be employed for constructing multiple classifiers [28]. In this paper, we will use such multiple classifiers for co-training.

3.1. Preliminary Knowledge on Rough Set

Rough set theory [25] is a new mathematical approach to imprecision, vagueness and uncertainty. It approximates a given concept below and from above, using lower and upper approximations.

Knowledge representation in rough sets is done via information systems. Let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects (the universe) and A is a non-empty finite set of attributes so that $a : U \to V_a$ for every $a \in A$. V_a is the set of values that a can take. For any $p \subseteq A$, there exists an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}$$

The partition generated by $IND(P)$ is denoted as $U / IND(P)$ or abbreviated to U / P and is calculated as follows:

$$U / IND(P) = \otimes\{a \in P \mid U / IND(\{a\})\} \quad U / IND(\{a\}) = \{\{x \mid a(x) = b, x \in U\} \mid b \in V_a\} \text{ and}$$
$$A \otimes B = \{X \bigcap Y \mid \forall X \in A, \forall Y \in B, X \bigcap Y \neq \varnothing\}, \text{ Where A and B are families of sets.}$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted by $[x]_p$. Equivalence classes generated by P are also called P-elemental granules, P-information granules. The set of elemental granules forms a concept system, which is used to characterize arbitrary subsets in the information system. Given an arbitrary subset X in the information system, two unions of elemental granules are associated with $\underline{P}X = \{x \mid [x]_p \subseteq X\}$, $\bar{P}X = \{x \mid [x]_p \bigcap X \neq \varnothing\}$

The concept X is approximated by the two sets of elemental granules. $\underline{P}X$ and $\bar{P}X$ are called lower and upper approximations of X in terms of attributes P. $\underline{P}X$ is also called positive region. X is a definable set if $\underline{P}X = \bar{P}X$. This means the concept X can be perfectly characterized with the knowledge P, otherwise, X is indefinable. An indefinable set is called

a rough set. $BND(X) = \bar{P}X - \underline{P}X$ is called the boundary of the approximations, and as a definable set, the boundary is empty.

Let P and Q be attribute sets that induce equivalence relations over U, then the positive, negative and boundary regions can be defined as $POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X$, $NEG_P(Q) = U - \bigcup_{X \in U/Q} \bar{P}X$, $BND_P(Q) = \bigcup_{X \in U/Q} \bar{P}X - \bigcup_{X \in U/Q} \underline{P}X$

The boundary region is the set of elemental granules which cannot be perfectly described by the knowledge P, and the positive region is the set of P-elemental granules which completely belong to one of the decision concepts. The size of positive or boundary regions reflects the approximation power of the condition attributes. By employing the definition of the positive region it is possible to calculate the rough set degree of dependency of a set of attributes Q on a set of attributes P. This can be achieved as follows: for $P, Q \subseteq A$, it can be said that Q depends on P in a degree k ($0 \le k \le 1$), this is denoted as ($(P \Rightarrow_k Q)$) if s $k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|}$

The dependency coefficient k measures the approximation power of a set of condition attributes. Given a decision table $DT = (U, C \bigcup D)$, C is the conditional feature set and D is the decision feature set. if $P \subseteq Q \subseteq C$, we have $\gamma_Q(D) \ge \gamma_P(D)$

Let $B \subseteq C$, $a \in B$, we say attribute a is indispensable in B if $\gamma_{(B-a)}(D) < \gamma_B(D)$; otherwise, we say a is redundant. We say $B \subseteq C$ is independent if any a in B is indispensable.

Attribute subset B is a reduct of the decision table if

(1) $\gamma_B(D) = \gamma_C(D)$ (2) $\forall a \in B : \gamma_B(D) > \gamma_{B-a}(D)$

A reduct of a decision table is a subset of condition attributes, which keeps the approximation power of the whole condition attributes, and has no redundant attributes. Usually there exists a number of reducts for a given decision table. Let $DT = (U, C \bigcup D)$ be a decision table, and $\{B_j \mid j \le r\}$ be the set of reducts, we denote the following attribute subset:

$Core = \bigcap_{j \le r} B_j$, $K = \bigcup_{j \le r} B_j - Core$, $K_j = B_j - Core$, $I = A - \bigcup_{j \le r} B_j$

Core is the attribute subset which cannot be deleted from any reduct, otherwise the discernibility of the system will decrease. Therefore, the core attributes will be in all of the reducts. I is called the completely irrelevant attribute set. The attribute in I will not be included by any reduct, which means I is completely useless in the system. K_j is a weak relevant attribute set. The union of Core and K_j forms a reduct of the information system.

Rough set theory discloses the fact that there exist multiple subsets of attributes which can keep the distinguish ability of the original data. They characterize the recognition problem in distinct subspaces and, therefore, capture different information of classification tasks. They are complementary to each other. The generalization power may be improved via combining a set of rough-set-based reducts.

3.2. Ensemble Learning with Reducts

We propose to train the multiple classifiers of ensemble learning with rough-set-based reducts (Figure 1). Given a decision table, there are a set of attributes $\{a_1, a_2, ..., a_n\}$. These attributes are grouped into a number of reducts $\{R_1, R_2, ..., R_N\}$ with a reduction algorithm. Then each reduct is used to train a base classifier with some learning algorithm.

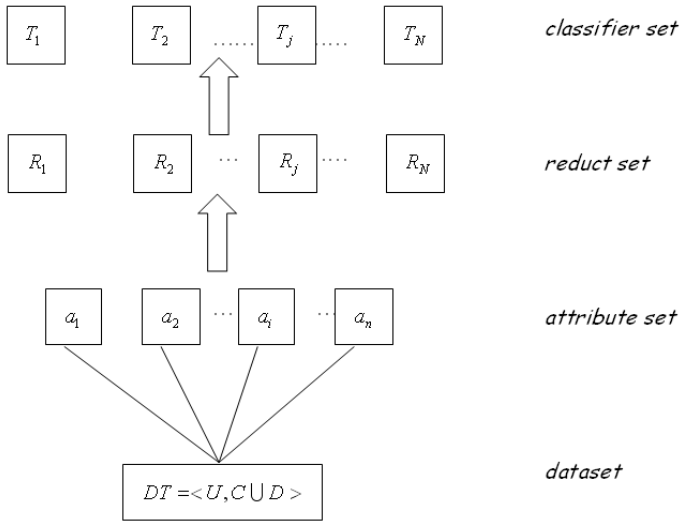

Figure 1. Ensemble Learning with Multiple Reducts

Several algorithms for finding a good reduct have been proposed based on heuristic strategies, such as discernibility matrix [29], dependency function [30], and the information entropy [31]. Here we give the dependency-based algorithm.

Let C and D be the condition attribute set and decision attribute, respectively. $B \subseteq C$, $\forall a \in B$, we define a coefficient

$$SIG(a, B, D) = \frac{|POS_B(D)|}{|U|} - \frac{|POS_{B-a}(D)|}{|U|}$$

as the significance of attribute a in B relative to decision D. Because the core attributes belong to any reducts, the reduction process can be started with the core. The core can be defined as

$$Core = \{a \mid |POS_{C-a}(D)|/|U| < |POS_C(D)|/|U|, a \in C\}$$

The algorithm for searching the core attributes is shown in Alg. 3.

Table 3. Searching Core in Decision Table

Algorithm 3: Searching core in decision table

Input: $DT =< U, C \bigcup D >$

Output: core

1: $core \leftarrow \varnothing$

2 for i=1 to n //n is the number of attributes

3 $SIG(a_i, C, D) = \left|POS_C(D)\right| / \left|U\right| - \left|POS_{C-a_i}(D)\right| / \left|U\right|$

4 if $SIG(a_i, C, D) > 0$

5 $Core \leftarrow Core \bigcup a_i$

6 end if

7 end for

8 return Core

Started with core, there exist several methods to search the reducts. In this work, we consider the WADF (worst-attribute-drop-first) algorithm proposed in [28]. WADF consists of two phases. The first phase is to search for the best reduct. The second phase is to search for multiple reducts. The two phases are shown in Alg. 4 and 5 respectively.

Table 4. Searching for the Best Reduct

Algorithm 4: Searching for the best reduct

Input: $DT =< U, C \bigcup D >$, core

Output: reduct

1: $B \leftarrow C - core$

2: $B^* \leftarrow sorted$ B in the order of ascending g(b),

g(b)=$\left|POS_A(U/D)\right| / \left|U\right| + \left|U/\{a\}\right| / \left|U\right|$

3: $C^* \leftarrow C + core$

4: for i=1 to $\left|B^*\right|$

5: $SIG(b_i, C, D) = \left|POS_{C^*}(D)\right| / \left|U\right| - \left|POS_{C^*-b_i}(D)\right| / \left|U\right|$

6: if $SIG(b_i, C, D) == 0$

7: $C^* \leftarrow C^* - \{b_i\}$

8: end if

9: end for

10: return C^*

Table 5. Searching for Multiple Reducts

Algorithm 5: Searching for multiple reducts

Input: $DT =< U, C \bigcup D >$, core

Output: multiple reducts

1: $B \leftarrow C - core$

2: $B^* \leftarrow sorted$ B in the order of ascending g(b), where

g(b)=$\left|POS_A(U/D)\right|/\left|U\right| + \left|U/\{a\}\right|/\left|U\right|$

3: $C^* \leftarrow C + core$

4: Find the best reduct, $RED(C)$, basd on Alg. 4

5: $K \leftarrow RED(C) - core$

6: for i=1 to $\left|K\right|$

7: $C^* \leftarrow C^* - \{k_i\}$

8: $RED_i =$ Find a new reduct

9: If $RED_i \notin RED$ then $RED = RED + RED_i$

10: $C^* \leftarrow C^* + \{k_i\}$

11: End for

12: Output: Reducts $RED = \{RED_1, RED_2 ... RED_N\}$

13: Output: Number of reducts= $\left|RED\right|$

4. Experiments

Five data sets from the University of California at Irvine (UCI) Machine Learning Repository [32] are used in the study. The information about the data is shown in Table 6.

Table 6. Data Description

	Data	Samples	Features	Classes
1	vote	435	16	2
2	t3	958	9	2
3	breast	449	9	2
4	diabetes	768	9	2
5	ionosphere	351	35	2

In the experiment, we compare our method (RSCO) with RASCO due to their similarity, *i.e.* manipulating feature space to generate diversities among multiple classifiers. Each data set is divided into training set and test set. RSCO and RASCO works on the training set and generates the augmented training set. Then, the test set is classified by the augmented training set with the decision tree algorithm. Classification accuracy is the measure to evaluate the performance of RSCO and RASCO, where

$$\text{classification accuracy} = \frac{\text{No. of correct classifications on testing instances}}{\text{No. of testing instances}}$$

When two semi-supervised learning methods are applied on the same data set with the same decision tree algorithm, higher classification accuracy means that the semi-supervised learning performance is better. To obtain the classification accuracy, each data set D is processed as follows:

(1) Data set D is randomly partitioned into two parts: labeled set L and unlabeled set U.

(2) Ten trials derived from ten-fold cross-validation on L are used to evaluate the performance of RSCO and RASCO. At each trial, 90% of L, that is T is used as training set. T is processed by RSCO and RASCO. The remaining 10% of L is used as test set to evaluate the performance of RFCO & RASCO.

(3) The average classification accuracy is obtained by averaging ten trials' accuracies.

(4) Considering that the partition of data set could influence this average classification accuracy, we execute the partition five times and get five classification accuracies (execute step 1-3 five times).

(5) Finally the reported accuracy is the further averaged value of these five values.

In the experiment, ensemble learning (by multiple reducts and random subspace) is configured as follows. The number of classifiers equals to the number of reducts which are generated by Alg. 3. In random subspace, the number of randomly selected features in each subspace equals to [n/2], where n is the dimensionality of the feature space. In the ensemble-based-co-training process, *iterNum* is set to 20, $\min_{1\le k\le K} n_k$ is set to 3.

When the setting of ensemble learning is fixed, there is only one parameter which can affect the experiment. It is the parameter determining data partitioning, i.e. the ratio between labeled data to whole data, referred to labeled ratio. In the experiments, we consider various labeled ratios including 10%, 20%, 30%, 40%, and 50%.

The experimental results are shown in Table 6. Table 6 consists of the classification accuracies of RASCO and RSCO for each dataset under different labeled ratios. For each comparison, the higher accuracy between RASCO and RSCO is shown in bold. Table 5 clearly shows the outstanding performance of RSCO: among five datasets with four different labeled ratios (thus, totally 20 cases), RSCO wins 18 cases.

Table 6. Experimental Results

Dataset	Labeled ratio	Classification accuracy	
		RASCO	RSCO
vote	20%	**0.917**	**0.917**
	30%	0.921	**0.923**
	40%	0.920	**0.937**
	50%	0.918	**0.923**
tic-tac-toe	20%	0.692	**0.721**
	30%	0.757	**0.767**
	40%	0.805	**0.822**
	50%	0.812	**0.854**
breast	20%	**0.827**	0.824
	30%	0.836	**0.841**
	40%	0.852	**0.857**
	50%	0.873	**0.895**
diabetes	20%	0.844	**0.860**
	30%	0.858	**0.879**
	40%	0.885	**0.896**
	50%	0.885	**0.899**
ionosphere	20%	0.878	**0.890**
	30%	0.886	**0.893**
	40%	0.907	**0.914**
	50%	0.910	**0.918**

5. Conclusions and Future Works

Co-training is an important semi-supervised learning technique which has been widely used in many applications. Standard co-training suffers from the applicability limita-tion. It can only be applied to the applications which can be represented by multiple independent views.

Ensemble learning has been proposed to use along with co-training to extend the performance and applicability of co-training. Under this methodology, we propose to combine co-training with rough set-based ensemble learning method. This ensemble learning method exploits rough-set-based attribute reduction algorithm to get a set of reducts of the original data and train base classifiers with reducts. Theoretically speaking, reducts are the optimal attribute subsets of the original data because they do not lose any indistinguishing information and have the least redundancy. Therefore, the base classifiers trained with reducts will get good generalization. At the same time, the base classifiers are constructed in different subspaces; there is a great opportunity for them to get diversity. Experimental results show that our proposed method is better than random subspace-based co-training in most of the cases in terms of classification accuracy.

In this work, we proposed to combine rough-set theory with co-training algorithm. In the future, we will explore to utilize rough-set theory with other semi-supervised classification or regression techniques.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 61100007, 61100081), and Fundamental Research Funds for the Central Universities (HEUCF100605, HEUCF041204).

References

[1] U. Brefeld and T. Scheffer, "Semi-supervised learning for structured output variables", Proceedings of the 23rd international conference on Machine learning, **(2006)**, pp. 145-152.
[2] R. K. Ando and T. Zhang, "A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data", J. Machine Learning, vol. 6, **(2005)**, pp. 1817-1853.
[3] M. Balcan and A. Blum, "A PAC-Style Model for Learning from Labeled and Unlabeled Data", Learning Theory, **(2005)**, pp. 111-126.
[4] K. Nigam, A. K. McCallum, S. Thrun and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM", Machine Learning, vol. 39, **(2000)**, pp. 103-134.
[5] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods", Proceedings of the 33rd annual meeting on Association for Computational Linguistics, **(1995)**, pp. 189-196.
[6] T. De Bie and N. Cristianini, "Convex Methods for Transduction", Advances in Neural Information Processing Systems, vol. 16, **(2003)**, pp. 73-80.
[7] T. Joachims, "Transductive Learning via Spectral Graph Partitioning", Proceedings of the 20th International Conference on Machine Learning, **(2003)**, pp. 290-297.
[8] Y. S. Chen, G. P. Wang and S. H. Dong, "Learning with progressive transductive Support Vector Machine", Proceedings of 2002 IEEE International Conference on Data Mining, **(2002)**, pp. 67-74.
[9] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines", Proceedings of the 16th International Conference on Machine Learning, **(1999)**, pp. 200-209.
[10] A. McCallum and K. Nigam, "Employing EM and Pool-Based Active Learning for Text Classification", Proceedings of the 15th International Conference on Machine Learning, **(1998)**, pp. 350-358.
[11] G. Schohn and D. Cohn, "Less is More: Active Learning with Support Vector Machines", Proceedings of the 17th International Conference on Machine Learning, **(2000)**, pp. 839-846.
[12] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification", J. Machine Learning, vol. 2, **(2002)**, pp. 45-66.

[13] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training", Proceedings of the 11th annual conference on Computational learning theory, **(1998)**, pp. 92-100.
[14] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training", Proceedings of the ninth international conference on Information and knowledge management, **(2000)**, pp. 86-93.
[15] S. A. Goldman and Y. Zhou, "Enhancing Supervised Learning with Unlabeled Data", Proceedings of the 17th International Conference on Machine Learning, **(2000)**, pp. 327-334.
[16] Z. H. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers", IEEE Transactions on Knowledge and Data Engineering, vol. 17, **(2005)**, pp. 1529-1541.
[17] J. Wang, S. W. Luo and X. H. Zeng, "A random subspace method for co-training", Proceedings of IEEE International Joint Conference on Neural Networks, **(2008)**, pp. 195-200.
[18] S. Kiritchenko and S. Matwin, "Email classification with co-training", Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research, **(2001)**, pp. 8-15.
[19] M. F. A. Hady, F. Schwenker and G. Palm, "Semi-supervised Learning of Tree-Structured RBF Networks Using Co-training", Proceedings of the 18th international conference on Artificial Neural Networks, **(2008)**, pp. 79-88.
[20] F. Feger and I. Koprinska, "Co-training using RBF Nets and Different Feature Splits", Proceedings of IEEE International Joint Conference on Neural Networks, **(2006)**, pp. 1878-1885.
[21] L. Breiman, "Bagging Predictors", J. Machine Learning, vol. 24, **(1996)**, pp. 123-140.
[22] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests", IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, **(1998)**, pp. 832-844.
[23] T. G. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization", J. Machine Learning, vol. 40, **(2000)**, pp. 139-157.
[24] R. Bryll, R. Gutierrez-Osuna and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets", Pattern Recognition, vol. 36, **(2003)**, pp. 1291-1302.
[25] Z. Pawlak, "Rough Sets: Theoretical Aspects of Reasoning about Data", Kluwer Academic Publishers, **(1992)**.
[26] L. Polkowski, S. Tsumoto and T.Y. Lin, "Rough set methods and applications: new developments in knowledge discovery in information systems", Physica-Verlag GmbH, **(2000)**.
[27] R. W. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition", Pattern Recogn. Lett., vol. 24, **(2003)**, pp. 833-849.
[28] Q. Wu, D. Bell and M. McGinnity, "Multiknowledge for decision making", Knowledge and Information Systems, vol. 7, **(2005)**, pp. 246-266.
[29] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems", Intelligent Decision Support: Handbook of Applications and Advances of Rough Set Theory, **(1991)**, pp. 331-362.
[30] J. Jelonek, K. Krawiec and R. Slowinski, "Rough set reduction of attributes and their domains for neural networks", Comput. Intell., vol. 11, **(1995)**, pp. 339-347.
[31] G. Y. Wang, "Rough reduction in algebra view and information view", Int. J. Intell. Systems, vol. 18, **(2003)**, pp. 679-688.
[32] UCI KDD Archive, <http://kdd.ics.uci.edu>.

Authors

Donghai Guan received his B.S. from Harbin Engineering University, Harbin, China. He got his M.S. degree in Computer Science from Kumoh National Institute of Technology (KIT), Gumi, South Korea in 2004. He got his Ph.D. degree in Computer Science from Kyung Hee University, South Korea in 2009. From 2009, he was a Post Doctoral Fellow at Computer Science Department, Kyung Hee University. Since February 2011, he has been an assistant professor in Harbin Engineering University, China. Since March 2012, he has been an assistant professor in Kyung Hee University, South Korea. His research interests are Machine Learning, Data Mining, Activity Recognition, and Social Computing. **E-mail:** dhguan@gmail.com

Weiwei Yuan received Bachelor's degree and Master's degree in Automation and Computer Engineering in 2002 and 2005 respectively from Harbin Engineering University, China. In 2010, she got her Ph.D. degree in the Department of Computer Engineering, Kyung Hee University, South Korea. Since September 2010, she has been an assistant professor in Harbin Engineering University, China. Her research interests are social computing, information security and machine learning. **E-mail:** yuanweiwei00@gmail.com