Selection of Relevant and Non-Redundant Feature Subspaces for Co-training

Yusuf Yaslan and Zehra Cataltepe

Istanbul Technical University Computer Engineering Department
34469 Maslak, Istanbul/Turkey
yyaslan@itu.edu.tr , cataltepe@itu.edu.tr

Abstract. On high dimensional data sets choosing subspaces randomly, as in RASCO (Random Subspace Method for Co-training, Wang et al. 2008) algorithm, may produce diverse but inaccurate classifiers for Co-training. In order to remedy this problem, we introduce two algorithms for selecting relevant and non-redundant feature subspaces for Co-training. First algorithm relevant random subspaces (Rel-RASCO) produces subspaces by means of drawing features proportional to their relevances measured by the mutual information between features and class labels. We also modify a successful feature selection algorithm, Minimum Redundancy Maximum Relevance (MRMR), to be used for feature subset selection and introduced Prob-MRMR feature subset selection scheme. Experiments on 5 datasets show that proposed algorithms outperform both RASCO and Co-training in terms of the accuracy achieved at the end of Co-training. Theoretical analysis of the proposed algorithms is also provided.

Key words: Co-training, Random Subspace Methods, RASCO

1 Introduction

Unlabeled data have become abundant in many different fields ranging from bioinformatics to web mining and therefore semi-supervised learning methods have gained great importance. The unlabeled data become available where obtaining the inputs for data points is cheap, however labeling them is difficult. For example, in speech recognition, recording huge amount of audio doesn't cost a lot. However, labeling it requires someone to listen and type. Similar situations are valid for remote sensing, face recognition, medical imaging and etc [1].

Co-training algorithm [2] is a semi-supervised iterative algorithm, proposed to train classifiers on different feature splits and it aims to achieve better classification error by producing classifiers that compensate for each others' classification error. Recently, a multi-view Co-training algorithm, RASCO [3], which obtains different feature splits using random subspace method was proposed and shown to result in smaller errors than the traditional Co-training and Tri-training algorithm. RASCO uses random feature splits in order to train different classifiers. The unlabeled data samples are labeled and added

to the training set based on the combination of decisions of the classifiers trained on different feature splits. However, if there are many irrelevant features, RASCO may often end up choosing subspaces of features not suitable for good classification. Recently Zhou and Li proposed an ensemble method, Co-Forest, that uses random forests in Co-training paradigm [4]. Co-Forest uses bootstrap sample data from training set and trains random trees. At each iteration each random tree is reconstructed by newly selected examples for its concomitant ensemble. Similarly, in [5] a Co-training algorithm is evaluated by multiple classifiers on bootstraped training examples. Each classifier is trained on whole feature space and unlabeled data are exploited using multiple classifier systems. Another similar application, Co-training by Committee, is given by Hady and Schwenker in [6]. It should be noted that all extensions of Co-training that requires bootstrapping may need a lot of labeled samples in order to be successful.

In this paper, instead of totally random feature subspaces, we propose two algorithms to create subspaces for Co-training. Initial algorithm, Rel-RASCO, produces relevant random subspaces which are obtained by means of relevance scores of features. Mutual information between features and class labels gives the relevance scores. In order to also maintain randomness, each feature for a subspace is selected based on probabilities proportional to relevance scores of features. The second algorithm, Probabilistic Minimum Redundancy Maximum Relevance (Prob-MRMR) feature subset selection, uses MRMR feature selection algorithm probabilisticly. Experimental results on 5 different datasets show that proposed algorithms outperform RASCO and traditional Co-training.

2 Relevant Random Subspace Method and Probabilistic MRMR for Co-training

We assume that we are given a classification problem with C classes. Inputs are d dimensional real vectors $x \in R^d$. The labels are represented using 1-of-C coding $l(x) \in \{0,1\}^C = [l_1(x), \ldots, l_C(x)]$. There is a labeled dataset L which consists of N samples. There is also an unlabeled data set U which consists of inputs only.

Rel-RASCO selects each feature based on it's relevance score which is obtained using the mutual information between the feature and the class labels. Let F_j denote feature vector where $j = \{1, 2, ..., d\}$. The mutual information, $I(F_j, l)$, between a feature F_j and the target classes $l = l_1, l_2, ..., l_C$ can be written as:

$$I(F_j, l) = \sum_{i,c} p(F_{i,j}, l_{i,c}) log \frac{p(F_{i,j}, l_{i,c})}{p(F_{i,j})p(l_{i,c})} \quad (1)$$

where $F_{i,j}$ denotes the jth feature and $l_{i,c}$ denotes the cth class label for the ith training sample.

Rel-RASCO algorithm works as follows: We first discretize the features in the labeled data set and obtain the relevance scores Q_{Score} for all the features.

Next we normalize the scores and use them as a probability distribution Q, on all d features. We create K subspaces, $S_1, \ldots, S_K$ using Q_j as the probability of selection of a feature F_j. Similar to RASCO, In Rel-RASCO also, a classifier is trained on each one of the feature subspaces $S_1, ..., S_K$ and the final classifier is obtained by majority voting. At each iteration of co-training, one most surely classified example from U for each class is added to L. The goal of Rel-RASCO's selection scheme is to select random feature subspaces which are as relevant as possible to the class labels. Using probability of selection proportional to relevance scores ensures that informative features are selected. Randomly selecting features enables classifier diversity and also ability to produce as many subspaces as needed and possible.

We also propose Probabilistic Minimum Redundancy and Maximum Relevance feature subset selection scheme. MRMR is a feature selection method which tries to find an ordering of features based on their relevance to the class label [7]. MRMR also aims at selecting the next feature as uncorrelated as possible with the current subspace of selected features. MRMR uses mutual information as a measure of feature-feature or feature-label similarity.

Let S be the feature subspace that MRMR seeks, the redundancy of S can be described using the within mutual information, W, of S:

$$W = \frac{1}{|S|^2} \sum_{F_i, F_j \in S} I(F_i, F_j) \tag{2}$$

Feature selection tries to choose an S with as small W as possible. In order to measure the relevance of features to the target class, again mutual information is used. Let $I(l, F_i)$ denote the mutual information between feature F_i and the target classes l. V, the relevance of S, is computed as:

$$V = \frac{1}{|S|} \sum_{F_i \in S} I(l, F_i) \tag{3}$$

Feature selection should come up with a feature set S which is as relevant and as nonredundant as possible.The MRMR method achieves both goals maximizing either $(V - W)$ which is called MID(Mutual Information Distance) or V/W which is called MIQ (Mutual Information Quotient). We use MID in our computations. Probabilistic MRMR, selects the first feature by using V as a probability distribution. Then by using redundancy scores W, MID scores are calculated and they are used as a probility distribution for selecting the next features in the subset. By adding randomness we are able to create diverse enough and accurate classifiers for Co-training.

3 Analysis of Rel-RASCO and Prob-MRMR

The accuracy analysis of the proposed algorithms will be obtained by using the RM (Recursively More) characteristic of feature spaces [7]. Let S^1 and S^2 be two subspaces with n features. S^1 is more *characteristic*, if the classification

Algorithm 1 Rel-RASCO and Prob-MRMR Algorithm

Select random subspaces $S_1 ... S_k$ by using Rel-RASCO or Prob-MRMR
for $i = 1$ to I **do**
  for $k = 1$ to K **do**
    Project L to L_k using S_k
    Train classifier C_k using L_k
  end for
  Label examples on U by using $C = (1/K) \sum_{k=1}^{K} C_k$
  Select one most surely classified example from U for each class, add them to L.
end for

error, e^1 on S^1 obtained by classifier C is less than the classification error, e^2 on S^2. Let a series of subsets of S^1 obtained by a feature selection algorithm be $S_1^1 \subset S_2^1 \subset ... \subset S_k^1 \subset ... \subset S_{n-1}^1 \subset S_n^1 = S^1$ and similarly subsets of S^2 be $S_1^2 \subset S_2^2 \subset ... \subset S_k^2 \subset ... \subset S_{n-1}^2 \subset S_n^2 = S^2$. S^1 is *Recursively More characteristic* (RM characteristic) than S^2, if $\forall k$ $(1 \leq k \leq n)$ the classification error $e_k^1 < e_k^2$. However in most cases it is difficult to obtain $e_k^1 < e_k^2$ $\forall k$. Let ρ $(0 \leq \rho \leq 1)$ be a confidence score that gives the percentage of k values that satisfy $e_k^1 < e_k^2$. When $\rho = 0.9$, S^1 is said to be *approximately* RM-characteristic [7]. For the case of Rel-RASCO, Prob-MRMR and RASCO, we experimentally show that $e_k^1 < e_k^2$, i.e. mean of the initial classification accuracies for Rel-RASCO and Prob-MRMR are smaller than that of RASCO for different subset sizes.

When classifiers are independent, the increase in individual classifier accuracies translates to increase in ensemble accuracy. Let each classifier in RASCO have an accuracy p and K be odd. Then the accuracy of the ensemble P_{RASCO} is [8]:

$$P_{RASCO} = \sum_{m=\lfloor K/2 \rfloor + 1}^{K} \binom{K}{m} p^m (1-p)^{K-m} \tag{4}$$

If the feature spaces obtained by relevance scores are RM-characteristics than the features selected randomly, then each classifier in the RM-characteristic algorithm (Rel-RASCO, Prob-MRMR) will have an accuracy $p + 1/\epsilon$ $(\epsilon > 0)$. The accuracy of the ensemble P_{RM} is:

$$P_{RM} = \sum_{m=\lfloor K/2 \rfloor + 1}^{K} \binom{K}{m} (p + 1/\epsilon)^m (1 - (p + 1/\epsilon))^{K-m} \tag{5}$$

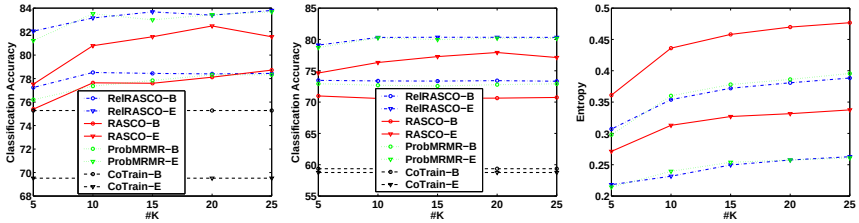
When $p > 0.5$ and $\epsilon > 0$, P_{RASCO} and P_{RM} are monotonically increasing, we can state that $P_{RM} \geq P_{RASCO}$. Note that $K \to \infty, P_{RASCO} \to 1, P_{RM} \to 1$.

4 Experimental Results

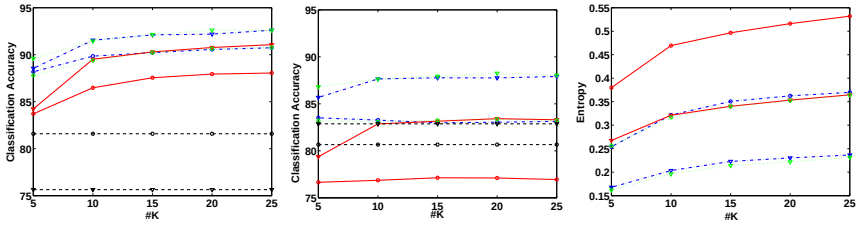
Experimental results are obtained on 5 different datasets: 'OptDigits' (Optical Recognition of Handwritten Digits), 'MFeat' (Multiple Features) and 'Isolet' (Isolated Letter Speech) datasets from the UCI machine learning repository [9], 'Classic-3' text dataset from [10] and the 'Audio Genre' dataset of [11]. Audio Genre data set has 50 features 500 instances and 5 classes. OptDigits data set has 64 features, 5620 instances and 10 classes. Classic-3 data set has 273 features, 3000 instances and 3 classes. Isolet data set has 617 features, 480 instances and 2 classes. Mfeat data set has 649 features 2000 instances and 10 classes.

For each dataset, experimental results for Prob-MRMR, Rel-RASCO and RASCO are obtained on 10 different random runs. At each random run, the whole dataset is splitted equally into a training partition and a test partition. Training set is splitted into unlabeled training set and μ % of the rest of the training data is used as labeled training set. PRTools [12] implementation of knn-3 classifier is used as the base classifier. In the experiments μ is selected as 10 or 20. $m = 25$ features are selected by both RASCO, Rel-RASCO and Prob-MRMR for each feature subset. Experiments are reported for different number of subsets, K= 5, 10, 15, 20 and 25. Note that, there isn't any natural split in the datasets except the audio genre dataset. Therefore Co-training algorithm is evaluated on 10 random feature partitions, each of them with 10 random runs and their mean accuracies are given. Co-training results don't change with respect to m (the dimensionality of subspaces) parameter. However, in order to be able to compare results, Co-training results are also given in figures as lines and they are named as CoTrain-B (B:at the beginning) and CoTrain-E (E:at the end). Similarly in figures, RelRASCO-B, RASCO-B, ProbMRMR-B and RelRASCO-E, RASCO-E, ProbMRMR-E represent the Rel-RASCO, RASCO and Prob-MRMR results at the beginning and end of the algorithms. In the figures each row of plots correspond to a particular data set. In each row, we report the averages of the ensemble accuracies, averages of the individual classifier accuracies and averages of the ensemble diversities.

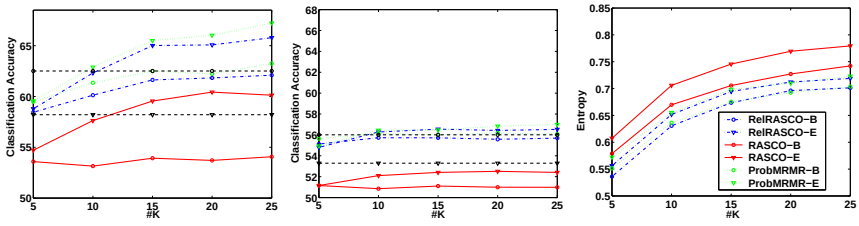
Audio genre dataset: The 5 least confused genres of Tzanetakis dataset [11], Classical, Hiphop, Jazz, Pop and Reggae, each with 100 samples, are used. Two different sets of audio features are computed. First 30 features are extracted using the Marsyas Toolbox [11]. Next 20 features covering temporal and spectral properties are extracted using the Databionic Music Miner framework [13]. Parameter μ is selected as 20. Ensemble accuracies at the beginning and end of Co-training with respect to different values of K are given in figure 1(a). CoTrain-B and CoTrain-E ensemble accuracies are 75.28 and 69.52 respectively, which means that Co-training does not benefit from the unlabeled data. Proposed algorithms outperform both RASCO and Co-training. Increasing the number of classifiers (K) increases both Rel-RASCO, Prob-MRMR and RASCO's accuracies, however the increase after $K = 10$ is not as significant as the increase when K increases from 5 to 10.



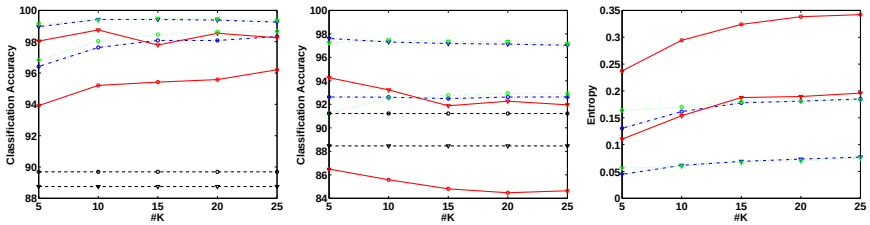
(a) Audio Genre, Ensemble (b) Audio Genre, Individual (c) Audio Genre, Diversity



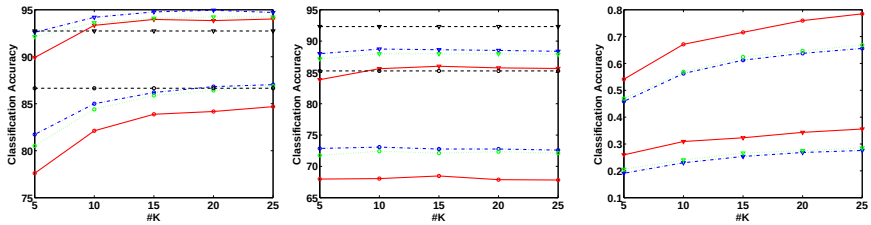
(d) Optdigits, Ensemble (e) Optdigits, Individual (f) Optdigits, Diversity



(g) Classic-3, Ensemble (h) Classic-3, Individual (i) Classic-3, Diversity



(j) Isolet, Ensemble (k) Isolet, Individual (l) Isolet, Diversity



(m) Mfeat, Ensemble (n) Mfeat, Individual (o) Mfeat, Diversity

Fig. 1. Mean ensemble and individual test accuracies and diversities on different datasets obtained by different algorithms with respect to K, m=25.

UCI Optdigits dataset: Parameter μ is selected as 10 for this experiment. Ensemble classification accuracies are given in figure 1(d). Accuracies of CoTrain-B and CoTrain-E are 81.59 and 75.64 respectively.

Classic-3 dataset: Term Frequencies of words are used as features and they are obtained using Term-to-Matrix generator (TMG) Matlab Toolbox [1].Parameter μ is selected as 20. Average ensemble accuracies are given in figure 1(g). CoTrain-B and CoTrain-E are 62.51 and 58.21 respectively. Co-training accuracy at the beginning is better than Rel-RASCO and RASCO when K=5. However increasing the classifiers increse the performance of Rel-RASCO, Prob-MRMR and RASCO. Note that, proposed algorithms significantly outperforms RASCO and Co-training when $K \geq 15$.

UCI Isolated Letter Speech dataset: A high dimensional dataset with 617 features and 480 instances from B and C letters are used in this experiment. Analysis are performed for $\mu = 10\%$. Ensemble accuracies are given in figure 1(j).CoTrain-B and CoTrain-E are 89.69 and 88.75 respectively. Proposed algorithms significantly outperform RASCO and Co-training for all cases of K.

MFeat dataset: Mfeat dataset is also a high dimensional dataset with 649 features. Analysis are performed for $\mu = 10\%$. Ensemble accuracies are given in figure 1(m). CoTrain-B and CoTrain-E are 86.64 and 92.74 respectively. Although Co-training is the best method for $K = 5$, for larger values of K proposed algorithms outperform both RASCO and Co-training

The second and third columns of Figure 1 show the average classification accuracies of individual classifiers and diversities of ensembles. Classifier diversities are measured using the entropy measure [8]. Figures 1(b), 1(e), 1(h), 1(k) and 1(n) show the average classification accuracies of individual classifiers obtained for Audio genre, optdigits, classic-3, isolet and Mfeat datasets respectively. Ensemble diversities are given in figures 1(c), 1(f), 1(i), 1(l) and 1(o) respectively. The average individual classification accuracies of the proposed algorithms are better than that of RASCO at the beginning and at the end of the algorithms. On the other hand, classifier diversities decrease when unlabeled data is used, except for the classic-3 dataset. Even though the diversity of RASCO is better than proposed algorithms, ensemble accuracies of Rel-RASCO and Prob-MRMR are better, which may be due to the fact that the individual classifier accuracies are better. Note that, although there are have been efforts to explain the relationship between classifier diversity and accuracy [8], [14] states that in problems with large number of features diversity is not a problem. Our experimental results support this statement and show that Rel-RASCO and Prob-MRMR produces more relevant and diverse enough classifiers that perform better than RASCO. The classification accuracies of Prob-MRMR and Rel-RASCO are generally similar. When d is large enough probability that correlated features will be in the same ensemble is small. This is the reason why Rel-RASCO performs almost as good as Prob-MRMR.

[1] $http://scgroup6.ceid.upatras.gr:8000/wiki/index.php/Main_Page$

5 Conclusion

In this paper, we introduced the Rel-RASCO and Prob-MRMR algorithms which are extensions of the Random Subspace method for Co-Training, RASCO [3]. Our purpose is to be able to select more relevant and non-redundant random subspaces and hence increase the performance of each classifier in the ensemble. We see that this increase translates into better Co-Training also. Rel-RASCO and Prob-MRMR classifiers are less diverse than RASCO classifiers, but diverse enough so that the ensemble accuracies are still more than that of RASCO. Experimental results on 5 different datasets show that, especially for high dimensional datasets, proposed methods outperform Co-Training and RASCO.

References

1. Roli, F.: Semi-supervised multiple classifier systems: Background and research directions. In Oza, N.C., Polikar, R., Kittler, J., Roli, F., eds.: Proc. of 6th Int. Workshop on Multiple Classifier Systems, Heidelberg, Springer-Verlag (2005)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 11th Annual Conference on Computational Learning Theory (COLT '98). (1998) 92–100
3. Wang, J., Luo, S.W., Zeng, X.H.: A random subspace method for co-training. In: International Joint Conference on Neural Networks(IJCNN 2008). (2008) 195–200
4. Li, M., Zhou, Z.H.: Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. IEEE Transactions on Systems, Man and Cybernetics **6** (2007) 1088–1098
5. Didaci, L., Roli, F.: Using co-training and self-training in semi-supervised multiple classifier systems. In: Lecture Notes in Computer Science. Volume 4109. (2006) 522–530
6. Hady, M.F.A., Schwenker, F.: Co-training by committee: A new semi-supervised learning framework. In: IEEE International Conference on Data Mining Workshops. (2008) 563–572
7. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancys. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1226 – 1238
8. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
9. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
10. Boley, D., Gini, M., Gross, R., Han, E., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore., J.: Partitioning-based clustering for web document categorization. Decision Support Systems **27** (1999) 329–341
11. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing **10**(5) (2002) 293–302
12. Duin, R.: PRTOOLS A Matlab Toolbox for Pattern Recognition. (2004)
13. Moerchen, F., Ultsch, A., Thies, M., Loehken, I.: Modelling timbre distance with temporal statistics from polyphonic music. IEEE Transactions on Speech and Audio Processingg **14** (2006) 81–90
14. Cunningham, P., Carney, J.: Diversity versus quality in classification ensembles based on feature selection. In: 11th European Conference on Machine Learning. (2000) 109–116