Using Co-Training to Empower Active Learning
Aktif Öğrenmeyi Güçlendirmek için Eş-öğrenme Kullanılması

Payam V. AZAD
Computer Engineering Departmenet
Istanbul Technichal Univ ersity
Email: vakil@itu.edu.tr

Yusuf Yaslan
Computer Engineering Departmenet
Istanbul Technichal University
Email: yyaslan@itu.edu.tr

***Abstract*—Active Learning and co-training are cases of semi-supervised learning both are used when labeled data is scarce. Active learning attempts to improve learning model by querying over unlabeled data and the main challenge there, is to find the optimum instance query. And co-training tries to exploit two different feature sets to enlarge number of labeled data without any need to get external information. Several researches tried to couple these two methods and get best out of them and they achieve noteworthy results. But we have witnessed that using co-training and active learning in sequence architecture outperforms when they are working in parallel. Using them in sequence means we have used co-training techniques to just find the best queries for active learning, and not in learning process itself. We will demonstrate that it has better results than plain active learning and co-training and even current parallel architectures. For this work we have used different techniques to split data into two distinct datasets; we will also discuss about it alongside our query selection method.**

***Özetçe* —Aktif öğrenme (active learning) ve eş-öğrenme (co-training), az sayıda etiketli veriye sahip olduğumuzda etiketsiz olan veriden en iyi sonucu ortaya çıkaran yöntemlerdir. Aktif öğrenme etiketsiz olan veriyi tahmin etmeye çalışarak öğrenme modelini iyileştirmeye çalışır. Buradaki en önemli zorluk en etkili sorgulamanın bulunmasıdır. Eş-öğrenme ise herhangi bir ek kaynağa ihtiyaç duymadan etiketli veriyi çoğaltmak amacıyla birbirinden bağımsız ve kendi başına yeterli özellik kümelerinden faydalanır. Bu iki yöntemi aynı anda kullanan pek çok başarılı araştırma yapılmıştır. Fakat biz çalışmamızda gördük ki; eş-öğrenme ve aktif öğrenme birbiri ardına kullanıldığında aynı anda kullanılmalarından daha başarılı sonuçlar vermektedir. Çalışmamızda eş-öğrenme yöntem ve yapısını, doğrudan öğrenme aşamasında kullanmak yerine, aktif öğrenme için en iyi sorgulamanın bulunmasında kullanıldık ve bu önerdiğimiz yöntemin sadece aktif öğrenme ve eş-öğrenme kullanıldığında ve hatta paralel mimarilerde kullanıldığında elde edilen başarıdan daha iyi sonuçlar verdiğini gösterdik. Bu çalışmada, veriyi birbirinden farklı iki veri kümesine bölmek için farklı ölçütler kullandık ve uyguladığımız her sorgu seçme yöntemi için bu ölçütleri değerlendirdik.**

Keywords*—*Active Learning, co-training, machine learning, semi-supervised learning.

Anahtar Kelimeler*—*aktif öğrenme, eş-öğrenme, makine öğrenmesi, yarı-denetimli öğrenme.

978-1-5090-6494-6/17/$31.00 ©2017 IEEE

I. Introduction

Semi-supervised learning algorithms are set of algorithms that are in between supervised learning algorithms that uses just labeled data and unsupervised learning algorithms that just use unlabeled data. Semi-supervised algorithms are using both labeled and unlabeled data in learning procedure. Active learning as a member of this family, tries to enrich labeled dataset by querying unlabeled instances from an outsider resource. In active learning biggest challenge is to find the best queries. It means to choose the best unlabeled instances to query its label and add them to labeled set, in a way that with the minimum queries get the largest improvement [1].

In original co-training paper [2] Blum et al. have proposed a method to enlarge labeled dataset and thus improving learning algorithm, exploiting two distinct datasets. In this work they have classified internet webpages, that they had few of labeled pages and a huge number of unlabeled pages in hand. They have attempted to use two different and almost independent set of features for each page; first set of features extracted from content of the page and second set of features gathered from pages that have linked to this page.

Several works done exploiting these algorithms like [3], [4] where they both have worked over splitted unrelated datasets; but Zhang et al. [5] introduced a new algorithm they have called SSCLA that they have tried to run both co-training and active learning in parallel and by partiting one single dataset into two separate ones and also calculating a measure to find the most informative instances, improve both of them together.

In this paper we tried to use this idea; but instead of assembling these algorithms in parallel we have used them in sequence. The idea is that first partitioning single dataset into two feature sets using some metrics, then by co-training try to find the certainty and contribution degree of each predicted instance, then using this values we choose the best instance to query. In the experiments we have used different methods to divide our dataset, including information gain, chi-square and ANOVA.

TABLO I: DataSets

	instance number	feature number	class number
cancer	698	10	2
chess	3195	37	2
sonar	207	61	2
credit	389	16	2
ionosphere	350	35	2

II. METHOD

Our algorithm is an iterative algorithm proposed of four steps, 1) divide labeled and unlabeled data into two different sets L1, L2 and U1, U2 using a splitting method, e.g. information gain or chi-square. 2) train these two sets of labeled data with an algorithm l and create two classifiers $h1$ and $h2$. 3) Use these classifiers to predict the label of unlabeled data. 4) Based on some criteria select most informative unlabeled instance, and query them accuse the label of this instance and add it to labeled set. In next iteration of the algorithm we will have newly labeled data in our labeled set and it will lead into more powerful model and better predictions.

A. Feature Partitioning

Co-training algorithm is working based on a logic that we have two different independent and self-sufficient set of labeled and unlabeled data. At the original co-training study [2] data was inherently of two different independent sources, links to a website and website's content. But while in our datasets as Zhang et al. proposed [5] we are trying to partition our data into two different sets, and naively assume that they are independent of each other. We have used three different methods: Information Gain, Chi-Square statistics and ANOVA (Analysis of variance) to divide our dataset into two sub-sets.

1) Information Gain: By using information gain one dataset have been divided into two subsets that have pretty same amount of information. We have first calculated the information gain of each feature and sort them by their information gain, then assign first feature, third, fifth and so on to first set and assigned second feature, fourth, sixth and so on to second dataset. This way we will have 2 different set of features that carry almost same amount of information. Largest disadvantage of this method (or the other partitioning methods) is the possibility of losing the information exists in inter-connected features, i.e. feature that have valuable data while they are bound together [6].

2) Chi-Square: Chi-Square (X^2) statistics calculate the independence of two stochastic variables [7]. This measure has been widely used for feature selection in a way that features that are most independent from class label are the least valuable ones. We have used this idea to split our data and in find the most valuable features.

3) ANOVA: ANOVA (Analysis of variance) [8] is the calculating of the variation of dataset and analysing how much this variations are similar in two datasets. It leads to find how much two datasets carry same meaning. While using this value we between each feature and the class label, the features that have the most contribution to class label can be derived.

B. Classifier Training

At this step we have used Naive Bias as base classifier. Being a probabilistic method is the most specific property of Naive Bayes for us. It gives empirical confidence of results that help us computing the certainty of predictions straight-forward. Using this algorithm separately over two feature sets we create classifiers to predict unlabeled data with certainties.

Having two different datasets, offers two different prediction (p_1, p_2) for each instances and their confidences (probability at our case: $prob_1, prob_2$). We have used some sort of voting to get the definite prediction for each instance in a way that the probability or certainty of each prediction is also taken into account. If both classifiers are predicting the same class the final choice is obvious but in the case that they have different predictions we accept the decision of a classifier that predict higher probability for each instance.

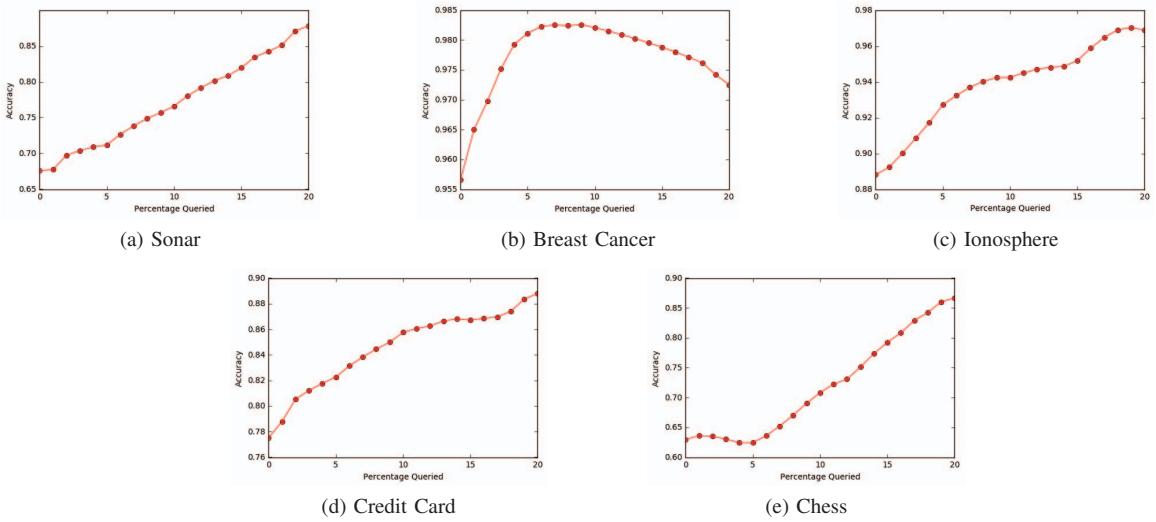
For training iterative learning method has been used; it means we have partially fit the model to training set because it will give us the power to add new instances iteratively without any need to re-train whole model from the beginning. It trains the model in batches, the first batch in our case is whole labeled dataset and next batches are instances chosen from unlabeled set by active learning algorithm.

C. Query Selection

Active learners in general use the most uncertain instance to query in quest for gaining the most information. This uncertainty is straight-forward for probabilistic learning models. In these models at binary classification those instances that the probability of them belonging to both classes are close to 0.5 are the most uncertain ones.
But we have used the approach that have been proposed by SSLCA. We are trying to find the most informative instance using K-Nearest Neighbors. In this approach those instances carry the most information that they themselves are certain ones but their neighbors are of very low certainty; that is against clustering and neighborhood hypothesis. It means that it can contribute huge information to dataset if the label has been guessed. At this point we are using co-training approach to calculate the contribution value as follows:

$$Contribution(Conf, x_i) = \frac{1}{M * Conf(x_i, c))} + \alpha \left| \sum_{x \in N(x_i)} [Conf(x_i, c) - Conf(x, c)] \right| \quad (1)$$

Where, M denotes the number of classifiers, that is in our case is two, $Conf(x_i, c)$ denotes the confidence of instance x_i belong to class c, a denotes the weight of contribution degree, $N(x_i)$ denotes the nearest neighbor of instance x_i. It is based on the idea that if the instance's confidence is high, but the confidence of its neighbor is low, which does



(a) Sonar (b) Breast Cancer (c) Ionosphere

(d) Credit Card (e) Chess

Şekil 1: Proposed algorithm results using unlabeled data

not meet the clustering hypothesis; otherwise, those have higher possibility to be informative instances because of high contribution degree. these instances are the most valuable instances to query from operator [5].

Then in each iteration a portion of least reliable predictions or the instances with highest contribution rate would be queried from oracle. For now we have considered this portion to be just in the size of 1, it means we have queried just one instance and add it to train in each iteration. And then update our model by calling the partially-fit (iterative learning) function to add the new instances to our model without need to learn the previously learned instances.

TABLO II: Results of different algorithms with 70% training data in comparison with our algorithm with 50% + 20% training data

	cancer	chess	sonar	credit	ionosphere
AdaBoost	0.957	0.964	0.738	0.778	0.869
Decision Tree	0.926	0.942	0.710	0.832	0.861
Gaussian Process	0.955	**0.984**	0.752	0.534	0.857
Linear SVM	0.967	0.916	0.676	0.828	0.849
Naive Bayes	0.959	0.609	0.621	0.810	0.865
Nearest Neighbors	0.959	0.894	0.676	0.677	0.800
Neural Net	0.941	0.966	0.752	0.685	0.894
QDA	0.955	0.667	0.593	0.429	0.849
RBF SVM	0.896	0.515	0.759	0.567	0.739
Random Forest	0.951	0.842	0.669	0.770	0.902
AL + NB	**0.972**	0.867	**0.878**	**0.888**	**0.969**

III. Implementation

Tests are done over 5 different dataset from UCI. UCI repository currently maintains different data sets as a service to the machine learning community. We chose five datasets which all are binary classification problems. These datasets are: 1)Breast Cancer Wisconsin [9], 2)Credit Approval [10], 3)Ionosphere [11], 4)Chess (King-Rook vs. King-Pawn) [12] and 5)Connectionist Bench(sonar) [13]. For each dataset, we kept the label and put them into the Labeled data pool L, remove the label and put them into the unlabeled data pool U, the ratio of labeled data and unlabeled data can be set in the next experiments (Table. **??**).

For all datasets we have implemented 10-fold class validation and divide 10% for test at each fold. We have divided remaining 90% train section into two set of 50% train and 40% validation. Then start active learning iteratively for 20% of data. That leads to 20% reduce from validation and after been queried, add it into training set therefore at the end of iteration we would have 70% train, 20% validation and 10% test in hand.

IV. Results

A. Reference Results

As discussed datasets are trained with 50% of data and then increase it into 70% of data. We have done all process using a very simple Naive Bayes classifier (later we will improve trainer itself) for sake of simplicity and having a uniform results independent of power of the classifier.

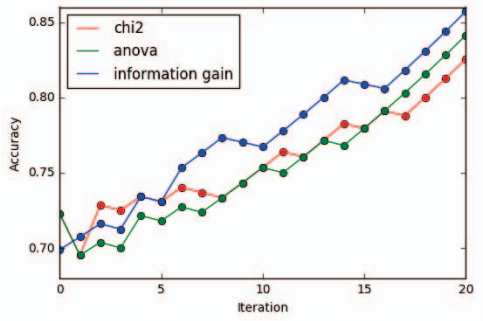
As a reference we are keeping this portion and attempt to test our algorithm against other well established algorithms. So we used 70% of our datasets to train using some of most powerful algorithms like Random Forrest, SVN, Gaussian Process and Neural Networks for this purpose (Table **??**). For sure we are not aiming to compare our algorithm to these algorithms because they are of different characteristic and we have custom queries that is not available in other algorithms. But it just gave us a glimpse of if 70% of data been chosen as training randomly, how much comparable to choosing 50% of data, train it with simple algorithm and then custom-choose 20% percent more of data and reach 70%.

As you can see we also include the result of Naive Bayes without Active Learning that is in most of cases so feeble

than other algorithms but when it is coupled with our version of active learning it passes other powerful algorithms easily in all datasets except for chess. The reason for under-expected result in chess dataset is that the features of this dataset is highly correlated and trying to separate its feature into two subsets we are losing this information laying in relations. We can see that at this dataset Gaussian Process [14] acts very well because it tries to consider all of relations between features in a huge matrix or there is neural network that with its complex structure tries to extract inter-relation inside features.

B. Feature Partitioning Results

As mentioned, Information Gain, Chi-Square Statistics and ANOVA are used for partitioning the dataset into two subsets. In almost all cases Information Gain achieves better results. So in the rest of report we will use just Information Gain. For a reference we just showed differences at Fig. 2 for sonar dataset obtained with three different partitioning method.



Şekil 2: Comparison of partitioning methods

C. Proposed Algorithm Results

The average results for 10-fold cross validation for each dataset is available in figure 1. Results show how much improvement gained at test while each percent of data added from validation set into training set.

Also we tried to compare our results to SSLCA results in Table III. Though they did not report their results in quantitative values; We could only roughly extract their results into numbers from their figures. From numbers it can be assumed that except chess dataset we have always improved the results over SSLCA. Note that unlabeled data partition size and cross validation type is not available for SSLCA results in [5].

V. Discussion

From the initial experiments it is shown that simple Naive Bayes after just 20% queries from oracle act far better than most of the other complicated algorithms having the same number of instances to train. This shows the effect and importance of query strategies. In future we can work on better confidential calculation and query algorithms and also we will try this with other learning algorithms. We are planning to expand this research for more complicated datasets with not only binary classes but also for multi-label datasets.

TABLO III: Comparison of our results with SSLCA.

	SSLCA		Our algorithm	
	start	end	start	end
cancer	0.958	0.97	0.957	**0.972**
chess	0.75	**0.95**	0.62	0.87
sonar	0.53	0.83	0.68	**0.88**
credit	0.745	0.88	0.80	**0.89**
ionosphere	0.6	0.875	0.88	**0.97**

There is also the possibility to use regression algorithms and then using regression results to derive both classification and confidence; just there is a need for some calibration (normalizing output to range 0-1) to make it work reasonably and properly [15]. In future works we are also planning to get use of this method also.

Kaynakça

[1] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[2] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[3] I. Muslea, S. Minton, and C. A. Knoblock, "Active+ semi-supervised learning= robust multi-view learning," in *ICML*, vol. 2, 2002, pp. 435–442.

[4] C.-H. Mao, H.-M. Lee, D. Parikh, T. Chen, and S.-Y. Huang, "Semi-supervised co-training and active learning based approach for multi-view intrusion detection," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 2042–2048.

[5] Y. Zhang, J. Wen, X. Wang, and Z. Jiang, "Semi-supervised learning combining co-training with active learning," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2372–2378, 2014.

[6] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, pp. 1–58, 2006.

[7] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 80–89, 2004.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[9] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology." *Proceedings of the national academy of sciences*, vol. 87, no. 23, pp. 9193–9196, 1990.

[10] J. R. Quinlan, "Simplifying decision trees," *International journal of man-machine studies*, vol. 27, no. 3, pp. 221–234, 1987.

[11] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Technical Digest*, vol. 10, no. 3, pp. 262–266, 1989.

[12] A. D. Shapiro, *Structured induction in expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[13] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural networks*, vol. 1, no. 1, pp. 75–89, 1988.

[14] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[15] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 161–168.