ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Using Co-Training to Empower Active Learning

M.Sc. THESIS

Payam VAKIL ZADEH AZAD

Department of Computer Engineering

Computer Engineering Program

October 2017

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

Using Co-Training to Empower Active Learning

M.Sc. THESIS

Payam VAKIL ZADEH AZAD
(504111553)

Department of Computer Engineering

Computer Engineering Program

Thesis Advisor: Yar. Doç. Dr. Yusuf YASLAN

October 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Aktif Öğrenmeyi Güçlendirmek için Eş-öğrenme Kullanılması

YÜKSEK LİSANS TEZİ

Payam VAKIL ZADEH AZAD
(504111553)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yar. Doç. Dr. Yusuf YASLAN

Ekim 2017

Payam VAKIL ZADEH AZAD, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504111553 successfully defended the thesis entitled "Using Co-Training to Empower Active Learning", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Yar. Doç. Dr. Yusuf YASLAN** ....
Istanbul Technical University

Jury Members : **Associate Prof. Dr. Hatice Köse** ....
Istanbul Technical University

Assistant Prof. Dr. Serap Kırbız Şimşek ....
MEF University

....

Date of Submission : **6 September 2017**
Date of Defense : **6 October 2017**

FOREWORD

I’d like to thank my professor that have lead me in all steps of this research and always was a good friend to me.

October Payam VAKIL ZADEH AZAD

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANOVA	Analysis of Variances
CEAL	Co-training Enhanced Active Learning
DT	Decision Tree
EDF	European Data Format
EM	Expectation Maximization
EMD	Empirical Mode Decomposition
FFT	Fast Fourier Transform
GNB	Gaussian Naive Bayes
IG	Information Gain
KDE	Kernel Density Estimation
KNN	K-Nearest Neighbor
MLP	Multi Layer Perceptron
NB	Naive Bayes
NN	Neural Network
RF	Random Forest
SVM	Support Vector Machine
SSLCA	Semi-Supervised Learning combining Co-training with Active Learning
UCI	University of California, Irvine

LIST OF TABLES

Page

LIST OF FIGURES

Using Co-Training to Empower Active Learning

SUMMARY

A huge amount of data is currently available and it is crucial to extract knowledge using this huge amount of data. However, only small portion of this data is classified and labeled as knowledge. Machine learning algorithms are often used to extract knowledge from these datasets; a family of machine learning algorithms that exploit both labeled and unlabeled data to enhance the labeling procedure is called Semi-Supervised Learning. In semi-supervised methods we develop a model, based on scarce labeled instances then, try to expand and improve it using abundant unlabeled instances. Active Learning and Co-training are two prominent members of Semi-Supervised Learning algorithms on which there has been an extensive amount of research.

Co-training is the process of learning labels for unlabeled instances from multiple-view datasets. Co-training learns two different classifiers for two different feature views, then, unlabeled instances are labeled when two classifiers are asserted on the class label of an instance. Some of these labeled instances are selected and added to the training set to enhance the learning model. This process is repeated until termination criteria are reached or a classification accuracy is achieved.

Meanwhile, Active Learning is a procedure that uses limited human annotator (oracle) knowledge to improve the models. In these algorithms, a plain model is first trained using a small labeled dataset, then some informative unlabeled instances are iteratively selected and labeled by an oracle to improve the model. The most important challenge at this stage is to find the instances which will make optimal improvement over the whole model by knowing their labels.

The proposed algorithms in this thesis use Co-training techniques to detect the optimal queries for Active Learning. However, two challenges arise here. Since Co-training uses two independent feature views, first is a need to split feature sets into two different feature views. The second challenge is to select instances to query from oracle using Co-training results.

Co-training works based on assumption of having two or more independent and sufficient views. This means that each view of the dataset should be sufficient to train a model and they should be independent from each other. These assumptions are customarily inconceivable for typical single view datasets. Therefore, the best practice to achieve these assumptions is to split the single feature view into two sets of feature views that have the closest amount of information as much as possible. The amount of information is calculated using four methods that are primarily used for feature selection. Previously, the Semi-Supervised Learning combining Co-training with Active Learning (SSLCA) algorithm is only applied Information Gain to attain different feature views. In this thesis, Chi-Square, Analysis of Variance (ANOVA) and ReliefF algorithms are applied for this purpose in addition to Information Gain.

For selecting instances from unlabeled dataset to query labels, two approaches are used. First is to calculate the contribution of the instances based on each instance and its neighbors' uncertainty. Second is to use unsupervised learning and clustering uncertain instances to find the best candidate instances for querying. These two approaches form our proposed methods, namely Co-training Enhanced Active Learning (CEAL) and Co-Active Learning. Co-Active Learning was also recently proposed and applied to two feature views. We have extended that algorithm by applying feature splitting so that it can be used in single feature view datasets.

Experimental results are conducted over eight benchmark datasets provided by (University of California, Irvine) UCI Machine Learning Repository. These datasets are very common datasets at machine learning research which have been referred and used by hundreds of research. For training models five machine learning algorithms from different types of algorithms have been used. From statistical methods Gaussian Naive Bayes is used that is a generative algorithm. From tree based algorithms Decision Tree and Random Forest have been used. From Neural Networks, Multilayer Perceptrons and from neighborhood based methods K-Nearest Neighbor have been employed. Gaussian Naive Bayes gave us direct statistical certainties and is an iterative fast algorithm. Multilayer Perceptron also provide certainties at a layer before the last layer (softmax). However, other algorithms have been implemented in regression form and by calibrating regression results to be in the range of 0-1 certainty has been inferred. Several tests are performed to compare partitioning methods, query selection strategies and machine learning methods. Classification accuracies are compared with our implementation of SSLCA, Active Learning and Co-training as baseline algorithms. Experimental results show that in most cases CEAL surpass other methods and the best partitioning method was Information Gain with a very slight margin.

Aktif Öğrenmeyi Güçlendirmek için Eş-öğrenme Kullanılması

ÖZET

Son günlerde, yüksek miktarlarda veri erişilebilir hale gelmiştir, fakat bunların çok küçük bir kısmı sınıflandırılmış ve etiketlenmiştir. Ayrıca, bu yüksek miktardaki veriden bilginin çıkarılması ise oldukça önemlidir. Etiketli ve etiketsiz verinin her ikisinden de faydalanarak etiketleme prosedürünü geliştiren bir grup makine öğrenmesi algoritması Yarı-Denetimli Öğrenme olarak adlandırılmaktadır. Yarı-denetimli öğrenme yöntemlerinde, nadir bulunabilen etiketli örnekler üzerinden bir model eğitilir ve daha sonra bu model çokça bulunabilen etiketsiz verilerden faydalanarak genişletilir ve iyileştirilir. Aktif Öğrenme ve Eş Eğitim ise üzerine sayısız ve yoğun araştırmalar yapılan Denetimli Öğrenme algoritmalarından önde gelen iki yöntemdir.

Aktif Öğrenme ve Eş-öğrenme, bilinen yarı denetimli öğrenme yöntemlerindendir ve literatürde bu konular üzerinde çok sayıda çalışma yapılmıştır. Ayrıca her iki yöntemi birlikte kullanan farklı araştırmalar da literatürde yer almaktadır. Bu çalışmada, bu iki yöntem, iki farklı mimaride birleştirelek, Aktif Öğrenme ve Eş-öğrenmein en başarılı uygulamalarından biri olan SSLCA yöntemi ile karşılaştırılmıştır. Bu çalışma için yeniden geliştirilen SSLCA yöntemiyle pek çok farklı senaryoda test edilmiş, başarıları değerlendirilmiştir. Aktif Öğrenmeyi gerçekleştirirken Eş-öğrenmenin de dahil edilmesi ile, üç farklı şekilde yöntemler karşılaştırılmıştır: 1) mimariye dayalı 2) bölümleme yöntemine dayalı 3) algoritmaya dayalı.

Bu çalışmadaki temel fikir az sayıda etiketli veri ve çok sayıda etiketsiz verinin bulunduğu problemlerde, etiketsiz veri içerisinden başarımı arttırabilecek örneklerin seçimidir. Yarı-denetime tabi olan bu tür etiketleme sorununa Eş-öğrenme ve aktif öğrenmenin birleştirilmesiyle çözüm üretilmeye çalışılmıştır. Klasik Eş-öğrenme yaklaşımlarının aksine, tekli görünümlü veri kümeleri üzerinde çalışılmıştır. Önerilen yaklaşım Beklenen Hata Azaltma veya Varyans Azaltma gibi birçok aktif öğrenme yaklaşımlarının aksine, yoğun hesaplama ve işlem gücü gerektirmemektedir.

Belirtilen problemlerdeki başamın incelenmesi için, bu çalışmada önerilen Eş-öğrenmeyle Zenginleştirilmiş Aktif Öğrenme (Co-training Enhanced Active Learning - CEAL), daha önceki çalışmalarda başarılı olduğu gösterilen CoActive ve SSLCA olmak üzere üç farklı yöntem kullanılmıştır. SSLCA yakın zamanda önerilmiş yarı denetimli öğrenme problemleri için başarılı bir algoritma olduğu için bu çalışma referans algoritma olarak kullanılmıştır. CoActive yöntemi çalışmada hedeflenen tekli görünümlü veri kümeleri üzerinde uygulanamadığı için, bu yöntem değiştirilerek probleme uygulanmıştır.

Eş-öğrenme doğası gereği çok görünümlü veri kümeleri ile çalışmaya uygundur. Çalışmada kullanılan veri kümeleri ise tekli görünümlü olduğundan dolayı bunların iki görünüme ayırılması gerekmektedir. Bu nedenle öncelikle, tekli görünümlü veri kümesi (etiketsiz ve etiketli kümeler), öğrenmeye katkılarına dayanarak iki bağımsız öznitelik kümesine ayrılmıştır. Öğrenme kümesindeki her bir özelliğin sahip olduğu bilgi, Bilgi Kazancı (Information Gain), Chi-square, ANOVA ve ReliefF olmak üzere dört farklı metrik ile ölçülmüştür. Bilgi kazançları bilgi teorisine dayalı bir yöntemdir. Esas olarak, karar ağaçlarının (decision tree) özelliklerini inceleyerek ağaç oluşumu sırasında dallarda bulunacak öznitelik seçimi için kullanılır. Her bir özelliğin taşıdığı bilgi miktarı, sistemin entropisinde neden olacağı değişim oranı ile ilgilidir. Diğer yöntemler, Chi-Square (χ^2) ve ANOVA, öznitelik seçimi için kullanılan yöntemlerdir ve her bir öznitelik ile etiketin arasındaki ilişkiye dayanarak çalışmaktadır. Chi-Square, öznitelik ve etiketler arasındaki bağımlılığı hesaplar, ANOVA ise öznitelikler ve etiketler arasındaki kovaryansı hesaplar. Veri kümelerini bölmek için kullandığımız son yöntem ise diğer bir öznitelik seçimi yöntemi olan Relief algoritmasının güncellenmiş bir versiyonu olan ReliefF'tir. Bu algoritma yinelemeli bir algoritma olup, özniteliklere ağırlık vermekte ve her adımda rastgele seçilen bir örneğe en yakın aynı etiketli ve en yakın farklı etiketli örnekleri kullanarak ağırlık vektörünü güncellemektedir. Bu metriklerle yapılan ölçümler sonucunda, öğrenmeye etkilerine göre öznitelikler her iki kümeye de eş oranda dağıtılmaya çalışılmıştır. Bir diğer deyişle, her iki kümenin de eşit bir şekilde bilgiye ve güce sahip olması amaçlanmıştır. Daha sonra, Gaussian Naive Bayes, K-Nearest Neighbor, Karar Ağacı, Random Forest ve Çok Katmanlı Algılayıcı (Sinir Ağı) olmak üzere, bu etiketli görünümler üzerinde çeşitli öğrenme algoritmaları eğitilmiştir. Bu algoritmalar tarafından eğitilen modeller kullanılarak, etiketlenmemiş örneklerin her bir sınıf için o sınıfına ait olma ihtimali bulunmuştur.

Bu noktadan itibaren üç yöntem birbirlerinden ayrılmaktadır. SSLCA tekrarlayan bir algoritma olup; öncelikli olarak etiketli veri kümesi üzerinde sınıflandırıcı eğiterek etiketsiz veri üzerindeki en belirsiz ve en çok emin olunan örnekleri bulmaya çalışır. En belirsiz örneği aktif öğrenme prosedürüne gönderir ve en emin olunan örneği ise birlikte eğitim prosedürüne gönderir. Bu iki algoritmadan gelen yeni verileri etiketli veri kümesine katarak sınıflandırıcıyı günceller. Diğer bir yöntem ise CoActive olup; etiketsiz veri kümesi içerisinden örnekler sınıf sayısı kadar demet oluşturularak bulunur. Demetler iki farklı sınıflandırıcının kararlarının uyuşmadığı örnekler üzerinde oluşturulur. Her bir demetin medyanına en yakın üye seçilerek etiketlenir ve eğitim kümesine katılır. Bu çalışmada önerilen CEAL algoritması ise, etiketsiz veri kümesi içerisinde en çok bilgi içeren örnekleri bulmak için bir katkı değeri kullanır ve en yüksek katkı değerine sahip etiketsiz veriyi aktif öğrenme için etiketleyiciye gönderir.

Test sonuçları farklı makine öğrenme algoritmaları kullanılarak elde edilmiştir. Çalışmaya referans olan algoritma, olasılıkları doğrudan veren, üretken istatistiksel bir model olan Gaussian Naive Bayes'tir. Gaussian Naive Bayes, makine öğrenmesi algoritmaları arasında güçlü bir algoritma olarak anılmamaktadır. Bu yüzden Karar Ağacı, Random Forest ve Çok Katmanlı Perceptron gibi daha güçlü algoritmalar da test edilmiştir. Karar Ağacı ve Random Forest algoritmalarının çıktırlarında bir sınıfa dahil olma olasılık değerlerini elde edebilmek için bu algoritmalar regresyon yapılarak sonuçlar olasılık değerine çevrilmiştir. Sinir ağlarında ise, her sınıfa dahil olma olasılıklarını elde edebilmek için benzer bir kalibrasyon kullanılmıştır.

Yöntemler üzerindeki ilk karşılaştırma, mimari farkı üzerinde yapılmıştır. Çalışmada referans alınan mimari, Aktif Öğrenme ve Eş-öğrenme işlevlerini birbirine paralel olarak uygulayan SSLCA'dır ve her yinelemede her ikisini birden kullanmaktadır. Önerilen mimariler ise CEAL ve modifiye edilen CoActive adlı yönteme ait mimarilerdir. CEAL'de sıralı Aktif Öğrenme ve Eş-öğrenme kullanılmıştır. Fakat SSLCA'da olduğu gibi eş-öğrenme öğrenme sürecinde değil, en iyi sorguyu bulmak için kullanılmıştır. CoActive'de ise, etiketlenmemiş veri kümeleri arasında en iyi sorguyu bulmak için benzer yöntemler kullanılmış, ancak kümeleme yapılmıştır.

Diğer karşılaştırma, bölümleme algoritmalarına dayanarak yapılmıştır. Eş-öğrenme, veri kümelerinin özünde iki bağımsız ve kendine yeterli görünüme ayrıldığı varsayımına dayanılarak yürütülmektedir. Dolayısıyla özellikleri iki alt kümeye bölmek amacıyla, en bilgilendirici örnekleri elde etmek ve özniteliklerin alt bölümlere oldukça adil dağılmasını sağlamak için Bilgi Kazancı, Chi-Square, ANOVA ve ReliefF özellik seçme yöntemleri kullanılmıştır ve karşılaştırılmıştır.

Deney sonuçları göstermektedir ki, çoğu test koşulunda CEAL diğer yöntemlerden üstün gelmektedir ve en iyi bölümleme yöntemi ise az bir fark ile Bilgi Kazancı yöntemidir.

1. INTRODUCTION

In many machine learning problems, there exists huge amount of unlabeled data to exploit in addition to available labeled data. In most cases in real world problems, collecting raw data is far cheaper and easier than obtaining their labels. Labels are generally assigned through efforts from a human agent and this human agent typically has to be an expert. This problem can be seen in many areas such as image processing, speech recognition, and bioinformatics. As an example, [1] states that identifying methylated regions in DNA needs long investigation by a highly trained professional. Finding an expert human agent is very difficult and expensive.

However, developing proper machine learning models require a huge amount of labeled data. Therefore, the idea of training a complicated model with the fewest labeled instances is tempting. Many Semi-Supervised learning algorithms have been proposed that can utilize unlabeled data; two of these ideas are Active Learning [2] and Co-training [3]. In Active Learning we try to find the minimum number of instances that represent the whole data in the best way, and eliminate the labeling of redundant instances; then, simply asking the oracle (i.e. a human agent expert) to just label these instances.

The other approach is Co-training; in which we exploit a huge number of unlabeled instances. Co-training is an iterative algorithm and learns two different classifiers on two different feature views. It essentially classifies data samples on an unlabeled set and adds the most certain examples into the training set. There are a vast number of extensions of these two algorithms in the literature.

This thesis, proposes an Active Learning algorithm which uses Co-training to select the most promising unlabeled instance for obtaining its label from oracle. The proposed algorithm has been tested on eight datasets that will be investigated thoroughly in the Experimental Results section. It is also compared with its counterparts, Co-training, Active Learning, Co-Active and Semi-Supervised Learning combining Co-training and Active Learning (SSLCA) algorithms. The next section discusses technical details of Co-training and Active Learning algorithms will be discussed.

1.1 Literature Review on Co-training and Active Learning

This section summarizes previous works on Co-training and Active Learning algorithms and previous combinations of these methods are discussed in the literature.

1.1.1 Co-training Algorithm

Co-training is an approach suggested by Blum and Mitchell in 1998 [3]. Blum et al. discussed how gathering data can be done automatically, but labeling it requires expensive human effort and collecting a large amount of unlabeled data is far less expensive than labeling them. In most cases, a substantial amount of expert effort is required which leads us to having a huge set of unlabeled data and a small set of labeled data. The principle idea of Co-training is to advance a way to exploit unlabeled data to enhance models trained with labeled data.

The original Co-training algorithm implementation was performed over webpage classification problem. While exploring grabbed webpages, two sets of information was available; first set was features extracted from each website's words and the second set was features extracted from words in websites that have a hyperlink to that website. Blum et al. had a small number of labeled webpages L in contrast, to the large number of unlabeled webpages, U, that had been grabbed automatically. They used features (Bag of Words) obtained from these two sets of labeled instances L_1 and L_2 to train two distinct Naive Bayes models h_1 and h_2. Then, they used these models to predict unlabeled instances U_1 and U_2. Therefore, two predictions t_1 and t_2 and the probability of these predictions were retained for each unlabeled instance.

Instances that have conflicts in the prediction of two models were selected as conflicting instances. Conflicting instances were those that had the highest difference between the probabilities of predictions; the models could be bootstrapped by taking the prediction from the more confident classifier, setting the label of the other feature set of that instance, and adding this instance to the labeled instances. Two classifiers were retained in the next iteration using the newly added instances and Co-training had the same idea of Expectation Maximization (EM), except with two distinct datasets [4].

Naive Bayes was used as classifier in the original Co-training algorithm; it is a statistical model that inherently extracts the probability of each instance belonging to any class and a final prediction is obtained using this probability [5]. Naive Bayes is an incremental algorithm that helps Co-training run faster and incremental algorithms are algorithms where adding one instance does not mean that we have to retrain the whole model, we just need to consider the new instances' effects [6]. Blum et al. illustrated that having two feature sets is necessary for running Co-training and that they should satisfy the following two conditions:

1) Each feature set needs to be sufficient, which means that each feature set is sufficient to train a properly representative model alone and

2) Each data set has to be independent from the other.

Note that these preconditions are satisfied in the website classification problem. In this problem, there are two datasets where each trains a model sufficiently and they are quite independent of each other. There are other works that apply Co-training using different algorithms similar to Naive Bayes, such as the Support Vector Machine (SVM) and decision trees [7]. They have shown that instead of generative models, one can also use discriminative models that are more powerful and can replace generative models in recent machine learning research [7].

Since then, there has been much work in this field. Pierce et al. [8] have discussed Co-training use in natural language processing and showed that Co-training is a more discriminative model than the generative model. Generative models are models that are able to describe the model that produces the problem dataset. Example generative models are Naive Bayes and EM [9]. In these models, a statistical model (combination of distributions) is generated that is able to describe the original data in the best way. In contrast, discriminative models have nothing to do with where the data has come from. They try to find labels without any idea of how data has been created. Example discriminative models are SVM, linear regression, and Neural Networks [9]. Although Co-training and EM algorithms exploit unlabeled data, Co-training outperforms EM in most cases because EM tries to match the model to its assumption of data. In cases when these assumptions are slightly wrong, it tries to learn wrong models and in each iteration drifts from reality. Although this problem exists in Co-training, it is less severe because instead of using this set of assumptions like EM, Co-training uses the classifier outputs to retrain the model [4].

Another problem in using the Co-training algorithm is the condition of having two independent feature sets for each problem. This is a quite difficult precondition. Previously, for solving this problem, two different classifiers were used on one dataset [10]. Therefore, different weak trainers are used to train and predict instances, and in this case, we are not limited to the number of independent distinctive datasets. Therefore, by working similarly to AdaBoost or any other bootstrapping algorithm, the number of predictions can be increased and a final decision can be obtained by their ensembles.

1.1.2 Active Learning

Active Learning or query learning is a semi-supervised machine learning algorithm that has been employed in a variety of applications. In Active Learning, in addition to a set of labeled and unlabeled instances, we have access to some sort of annotator that can give us the label of each instance. These labels can be used to expand the set of labeled instances and, therefore, empower the model trained from this set. This means that we can query for the labels of unlabeled instances from an oracle. This third-party source can be a human operator or any other form of labeling system. The challenge is that obtaining labels from an oracle is expensive in a variety of formats (e.g. it can be vastly time-consuming, or we could have access to just a limited number of queries).

The main challenge of Active Learning is finding the optimum query. The optimum query is the process of finding an instance where having its label would decrease the Generalization Error the most. In other words, we try to find the most informative instances to query in favor of gaining the most improvement in our model with the least number of queries. This improvement could be minimizing the general error, maximizing the general F-score, or any other quality metric of the model that is important to us.

By using the Vapnik–Chervonenkis dimension (VC-dimension) concept, Cohn et al. [11] showed that for achieving the accuracy of ε in Active Learning we just need $O(log(N))$ instances to obtain the label where N is the number of instances. The worst-case scenario, Active Learning will be equally as good as passive learning (a normal machine learning method), but the objective of Active Learning is to find approaches that are empirically far better than the worst-case scenario.

From a theoretical perspective, Active Learning can be divided into two cases. Either we assume that we have a target function that represents the labeled and unlabeled data perfectly (realized case), or we do not have such an assumption and we try to improve our classifier in each step to obtain a better performance (agnostic case) [12]. From a practical perspective, we have different types of Active Learning scenarios. The two most common scenarios are 1) stream-based, where unlabeled instances arrive in a stream and there is a need to decide online whether to query and 2) pool-based, where we have access to all unlabeled instances at any time; thus, we can decide the best instances to query, being able to audit all instances. In this research, pool-based scenarios are investigated because they are the most common form of problem that occurs in the real world [2].

1.1.2.1 Query Selection Strategies

Different approaches for selecting the best unlabeled instance to query are discussed in the literature. The seven most important approaches are reviewed below [2, 12, 13].

In the following approaches, the selected instance will be shown as x^*.

Uncertainty Based

The most common and basic methods of querying in Active Learning are uncertainty-based methods. The easiest implementation of such a method is to implement it by using statistically generative models, because they give the likelihood of each class inherently for every instance in the form of a probability. Other algorithms are also widely used for this purpose by performing normalization and calibration of the regression results of algorithms [2].

In uncertainty methods the most uncertain unlabeled instances are detected and queried. In binary classification problems, the uncertainty of a model is high when the posterior probabilities of the two classes are too close. This can be represented using the following formula:

$$x^* = argmin_x |P(y = +|x) - P(y = -|x)| \tag{1.1}$$

In this equation, $P(y = +|x)$ is the probability of instance x belonging to the class + and $P(y = -|x)$ is the probability of it belonging to the class -.

But the case when we have more than two classes the previous formula must be updated. In this case, the selection would be the instance with minimum of maximum posterior probabilities. It means this instance has the lowest probability for predicted label. This informative instance is selected by the following equations.

$$x^* = argmin_x P(\hat{y}|x) \tag{1.2}$$

$$\hat{y} = argmax_y P(y|x) \tag{1.3}$$

where y is the class that has the maximum probability. Therefore, x^* will be the member who has the minimum of maximum posterior probability. In this case we lose the probability information of classes that are not maximum posterior. For overcoming this problem margin sampling has been suggested [2] with the formulation given in

Eq.1.4.

$$x^* = argmin_x P(\hat{y}_1|x) - P(\hat{y}_2|x) \tag{1.4}$$

where $\hat{y}_1$ and $\hat{y}_2$ are the most probable class and the second most probable class respectively. In this approach, we aim to find the instance that has the largest difference between the maximum posterior probable class with the second maximum posterior probable class. In other words, this is the instance that our model is most skeptical about choosing between classes. Though it is better than the first method, it still does not consider the probability of third and lower classes. Thus, an entropy based approach (see the Eq.1.5) has been proposed by the same researchers.

$$x^* = argmax_x - \sum_i P(y_i|x) logP(y_i|x) \tag{1.5}$$

$P(y_i|x)logP(y_i|x)$ is the entropy of each class. In this method, the members that have the highest entropy (variance) have been chosen based on current models, and by querying these instances, we are reducing entropy among instances. [2].

Expected Error Reduction

Another very powerful approach is finding the instance if its label is being assigned, which will make the largest change in the Generalization Error. This is done by the one-by-one assigning of labels to unlabeled instances and training the whole model. As it is obvious this is a very effective method, but too resource-consumptive, this method can be used when queries are much more expensive than the computational power. In this case, we consider all labels for all unlabeled instances, find the difference that the labels make, and choose the instance whose label will make the largest modification. [12].

Variance Reduction

The Generalization Error (E) of a machine learning model is a combination of variance and bias of that model as $E = Bias^2 + Variance$ [5]. By knowing this fact, it is possible to reduce Generalization Error by reducing the variance. Previously in [11] showed that finding members to reduce the variance is far less consumptive than finding the member that reduces the error because in variance reduction we just need to calculate the variance of the dataset for each instance and label instead of training the whole model like error reduction approach [12].

Expected Model Change

In the models that are working based on Gradient Descent or familiar optimization algorithms like Neural Networks and Gradient Boosting we can rely on the amount of change each instance will make over gradients if its label is given. It is working like Expected Error Reduction unless there is no need to train the whole model from scratch. E.g. in Neural Networks, we need to just do one forward pass step instead of several forward-backward passes. So if we choose instances with highest expected model change we will reach significant quality with very few computation [2].

Density Based

In the uncertainty based methods the fact that each instance can be an outlier is never being considered. It means that all instances are evaluated equally. But in density based approaches, densest regions are targeted and the query is selected from this region. This can be done by calculating the Kernel Density Estimation (KDE) [5] or any similar density calculation method. Then, the unlabeled instance which is closest to the highest density region is chosen [2].

Cluster Based

Very similar approach to density based method is been proposed by Hsu et al. [14]. In this approach unlabeled instance set is clustered into *n* clusters that *n* is the number of labels. After that, members that are close to the center of clusters are selected. This approach is highly dependent on the quality of clustering and also spatial distribution of instances.

Query by Committee or Disagreement

There exist another two ways to find the optimum query in literature: first is to train several models and try to find the instance that all models are uncertain about it (Query by Committee), and second is to select instances that models have the highest disagreement over them (Query by Disagreement). After selection of this critical instance, its label is queried by oracle. This approach has been proposed by Seug et al. [15] and has been adapted for different problems. Our proposed model Co-training Enhanced Active Learning (CEAL) will be based on this approach.

1.1.3 Active Learning and Co-training Combination

As stated in this text, there has been a vast amount of work conducted on Active Learning and Co-training separately. These two methods have also been used together in several studies, such as [16] [17] [18] [19]. The special references for this thesis will be two recent studies [20] [21]. The first one proposes a method called SSLCA to find the high-density regions and query instances selected from these regions. Zhang et. al. [20] have used Co-training and Active Learning in parallel. Their method involves obtaining a single-view dataset and splitting it into two different views, and each step performs both Co-training and Active Learning. In Co-training, they add instances with high certainty in one view and the lowest certainty in another view to the labeled set with the label obtained from a certain view. At the same time, the Active Learning step also takes place, which tries to find the instance with the lowest overall certainty and its label by querying the oracle and adding it to the labeled dataset.

Yuce et. al. [21] have proposed a disagreement based method named Co-Active Learning. Co-Active Learning also works with naturally separated datasets, like Co-training. It operates based on clustering uncertain instances and then sending the cluster center to a query from the oracle. The assumption is that the data will be spread in *n* spatial clusters, where *n* is the number of classes. They assume that the most informative instance exists in the middle of the cluster and this instance will represent the largest number of instances; so, by finding the label of this instance, a larger number of instances will be affected. This cluster center is the closest member to the median of clusters. The drawback of this method is that it just works with naturally separated datasets and is heavily dependent on the quality of clustering and data distribution.

2. PROBLEM AND METHODOLOGY

2.1 Problem Definition

In most cases in real-world problems, collecting raw data is far cheaper and easier than obtaining their labels. In many cases, labels are the results of a human agent's work. This human agent needs to be an expert (e.g. flawed chromosomes [1] can be detected after a long investigation by a highly trained professional, or earthquake-damage-vulnerable-building detection is the result of a long investigation by engineers). Finding an expert human agent is difficult and expensive. However, raw data is becoming cheaper and more abundant every day. Terabytes of human genome data and geographical information is gathered by satellites.

To develop a proper machine learning algorithm, a huge amount of labeled data is needed. The idea of training a complicated model with the fewest labeled instances is tempting, and different methods have been proposed to accomplish this goal. Two of these ideas are Active Learning [2] and Co-training [3] , which both belong to semi-supervised machine learning algorithms. In Active Learning, we try to find the minimum number of instances that represent the whole data in the best way and avoid labeling redundant instances; then, we ask the oracle (i.e. a human expert agent) to label just these minimum instances.

In Co-training, a huge number of unlabeled instances are exploited. This is mainly possible in datasets that have two or more views inherently, where all of these views are self-sufficient and independent. This means there is a need to have more than one type of data for each instance so that it is possible to detect the real label of these instances with just one of these types (views).

The main purpose of this research is to design a method that could use the power of both Active Learning and Co-training when there is just one view available. This leads us to make a model can produce optimum-quality models with the minimum number of labels.

2.2 Methods

In this thesis, two methods have been proposed to solve this problem, CEAL and a partitioned version of Co-Active [21] and the results will be compared with our implementation of SSLCA [20].

2.2.1 Co-training Enhanced Active Learning (CEAL)

CEAL is composed of four steps:

1. Partitioning single-view data into two separate, independent views. We attempt to partition these two views fairly from information perspective. Thus, two sets of L_1 and L_2 labeled data and U_1 and U_2 unlabeled data would be gained, that $||U|| >> ||L||$ ($||U||$ and $||L||$ are the number of instances in unlabeled set and labeled set).

2. Using an algorithm from our set of algorithms to train a model for labeled data and obtain models m_1 and m_2 (each model is trained from one view).

3. Implementing obtained models in the previous step in unlabeled data and obtaining the probabilities of each instance belonging to any of classes $P(C_i|U_j)$. C_i is the class i representer where U_j is the j^{th} unlabeled instance.

4. Using calculated probabilities from previous step, to find the most uncertain instance and send it to oracle to find its label and adding the queried instance with its label to labeled sets.

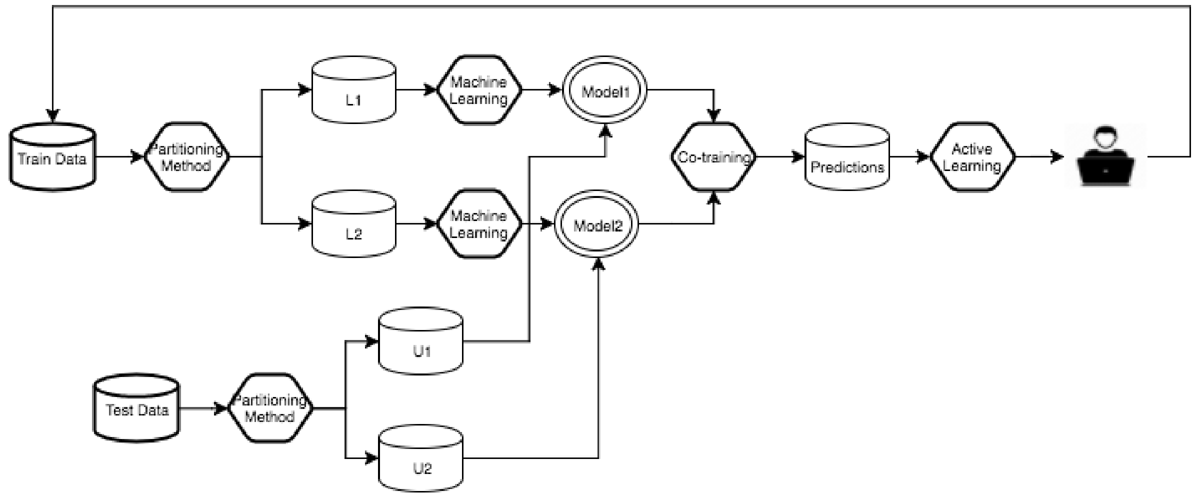

Figure 2.1 : CEAL algorithm block diagram.

Then this process continues for a given number of iterations. This process is illustrated at Figure 2.1. At each step learned models are being tested over completely separate test dataset and using aggregation of probabilities for two views suggesting the argmax of probabilities as the prediction class.

$$\hat{y} = argmax_j \sum_{j=\{1,2\}} P(C_j|X) \tag{2.1}$$

where j is the class indicator and $P(C_j|X)$ is the probability of X belonging to class j.

2.2.1.1 Feature Partitioning

Original Co-training paper [3] makes the assumption of working with naturally separated, independent, and self-sufficient data. However, in our test sets, we just have single-view data. For implementing CEAL, Co-Active, or SSLCA, there is a need to partition our datasets as in Co-training. Therefore, we try to partition datasets over features in such a way that each view (partition) has a fair (close to equal) amount of information and naively assume that these features are independent and self-sufficient. To achieve this fairness Information Gain, Analysis of Variance (ANOVA), Chi-Square, and ReliefF have been used for feature selection. These algorithms work based on an estimation of the information each feature may hold.

Certainly, some valuable information that exists in the combination of features will be missed while dividing the features into two subsets, but it will be demonstrably compensated for with the other aspects of our algorithm.

Information Gain

Information Gain is the metric to show how much uncertainty would be dissolved in the case one is aware of a condition of a feature [5]. Its formulation for a feature set D is given below:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \tag{2.2}$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{D_j}{D} * Info(D_j) \tag{2.3}$$

where m is the number of possible labels and p_i is the probability of belonging to ith label for an instance. p_i is calculated by finding the ratio of occurrences of that class in whole data set. $p_i log_2(p_i)$ is the entropy of when giving ith label to feature. j is the indicator of each feature that Information Gain is trying to find its effect on general entropy of model. So, $info_A(D)$ is the amount of entropy decreased by assertion of its each feature's label and this value is called Information Gain. As can be seen, it is based on Information Theory and been successfully employed in Decision Trees in forming branches. In this thesis, Information Gain is used for partitioning single view datasets into two distinct, and fairly distributed views. First, Information Gain for each feature is calculated and features are sorted according to their gain values. Then, assigning first, third, fifth, and so on features into view number one and assign second, fourth, sixth, and so on to view number two. By this way, it is assumed that two distinct views that have Information Gains close to each other is obtained.

Chi-Square

Chi-Square (χ^2) statistics calculate the independence of two stochastic variables [22]. While calculating between a feature and an instance label, Chi-Square showed us the amount of correlation of each feature with the label. We could consider the feature that has the highest correlation with the label as the most important. This method is generally used for feature selection. Therefore, features are sorted as in Information Gain, but this time based on Chi-Square statistics, and assigned into two views. The formula of the Chi-Square statistics is as follows.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - N_i)^2}{E_i} \tag{2.4}$$

where E_i is the variance, O_i is the observance came from each instance and N_i the label of instance. This is calculated for each feature, and we can obtain a list of χ^2 that we can sort based on the level of influence they have on the label.

ANOVA

ANOVA (Analysis of variance) [23] is a method used for feature selection. ANOVA calculates the impact of each feature over the label, and any feature that has the most impact on the label is the most informative feature. Therefore, as in previous methods, we sort features based on the ANOVA value and assign them to two views. The value calculated for ANOVA is the covariance of each feature with its label. This means when a feature varies, how much it makes a difference to the label leads to knowledge of how we can predict the label using just this feature.

ReliefF

ReliefF is an iterative method for feature selection [24]. It works based on the nearest neighbor algorithm. In the first step, a weight vector is created by the length equal to the number of features. Then, at each iteration, a random instance is chosen and instances are sorted based on the Manhattan distance (L-1 norm) in two groups. The first group contains the instances of the opposite class (misses), and the second group contains the instances that belong to the same class (hits). Then, the nearest instance of each group to a randomly selected instance called, respectively, the near-miss or near-hit, is chosen. Then, based on the difference between the value of features in these two instances and our randomly selected instance, the weight vector based on the below equation is updated.

$$\omega_i^j = \omega_i^{j-1} + |x_i - nearhit_i| - |x_i - nearmiss_i| \tag{2.5}$$

where ω_i^j shows the weight corresponds to feature i at iteration j and x_i is the value of randomly selected instances i_{th} feature as well as $nearhit_i$ and $nearmiss_i$.

In the original version of Relief the Euclidean distance (L-2 norm) instead of the Manhattan distance is used therefore, the update formula was as follows:

$$\omega_i^j = \omega_i^{j-1} + (x_i - nearhit_i)^2 - (x_i - nearmiss_i)^2 \tag{2.6}$$

At the end of the *m*th iteration, the weight vector would represent the influence of each feature on the label. It can be considered as the amount of information each feature sustain. We divide features into two sets as we did in previous methods.

2.2.1.2 Base Classifier Training

In this thesis, several classifiers have been used as the base classifier. Gaussian Naive Bayes is used as representative of statistical generative models, which produce the probability of predictions inherently. This probability is crucial for calculating the confidence in the query selection phase. Additionally, K-NN and decision tree classifiers are used, which are both simple algorithms and fast to train. Furthermore, a random forest classifier is used as a powerful model based on boosting. The last algorithm that has been used is the Multilayer Perceptron (MLP) classifier belonging to the neural network family, which is also a powerful algorithm. For training the MLP, back-propagation and Adaptive Moment Estimation (Adam) optimizer [25] have been used. We will discuss the detail of the algorithms in the results section.

Naive Bayes and MLP models have been trained in an iterative manner, as in online learning. This means at each iteration where a new instance comes, the model is not trained from scratch, but the old model just updates the weights and parameters considering the new instance. This results in a huge computational improvement in using these two algorithms. However, the characteristics of other algorithms do not allow the use of this trick and the whole model train is needed at each iteration.

Our algorithms are working based on the probability of each instance belonging to different classes $P_j(C_i|X)$; where C_i is the i^{th} class and $j \in \{1,2\}$ is the indicator of model trained based on view. By using these probabilities, both the final label prediction and next instance are determined. As has been mentioned, statistical models naturally gave us the probability of each prediction (the prediction is performed using the probability), but non-statistical models such as random forests just give us a value for regression, and by calibrating this regression result (normalizing the output of regression to constrain it between 0 and 1), a value that could be considered the probability for the model would be obtained with a good approximation.

2.2.1.3 Query Selection

Query selection is the most important part of our algorithm. The best query is the query over an instance where if it is added to the dataset, it will make the biggest improvement. Therefore, by this means, a model can be trained with the fewest queries.

Different methods have been implemented for Active Learning, which has been discussed in the previous chapter. The most common method for binary classifiers is selecting instances whose certainty (probability of prediction) is low (close to 0.5 with binary classifiers). In these cases, the trained model is unable to detect the class with high confidence.

In this research, neighbors' certainty is also considered in finding the less confident members. We call this the contribution degree, which is a measure to calculate the high certainty of a member and the low certainty of its nearest neighbors. In a case where an instance has high certainty, but its closest neighbors have low certainty, this means that we have a very vague idea of the region. In this case, querying it would not only find an instance's label, but would also enable one to grasp what is happening in its neighbors and may be able to help label a large group of instances correctly. The contribution value is calculated as follows:

$$Contribution(Conf, x_i) = \frac{1}{K * Conf(x_i, c))} * \left| \sum_{x \in N(x_i)} [Conf(x_i, c) - Conf(x, c)] \right| \quad (2.7)$$

$Conf(x_i, c)$ is the probability of instance x_i having the label of c and K is the number of nearest neighbors that we have chosen to evaluate. We have set K to five during this research.

2.2.2 Co-Active Learning Algorithm

Yuce et al. [21] have proposed a method called Co-Active Learning. They worked with the Sleep European Data Format (EDF) dataset and tried to predict sleeping stages. They used the Fast Fourier Transform (FFT) and Empirical Mode Decomposition (EMD) to extract two separate views from a dataset. In this thesis, a slightly different approach based on Co-Active has been proposed for this problem. The main difference is that we have a dataset with just one view.

In our implementation of the Co-Active Learning algorithm datasets have been partitioned into two views, as with CEAL. Then training step took place and probabilities are extracted. In the third step, uncertain members are clustered into k clusters using k-means where k is the number of classes. Then, the closest instance to the median of each cluster is considered the most informative unlabeled instance. The Euclidean distance is used to calculate the distances of instances. In this step, the assumption that instances are distributed spatially fairly is taken. In the next step, the label of this instance is queried. By this means, the best representation of the model with the fewest members can be achieved Figure 2.2.

2.2.2.1 Uncertainty Measure

For selecting uncertain members, the probability of each prediction model from each views is calculated for all instances. These selected members are used for forming clusters. Then a threshold, θ is used with this formula:

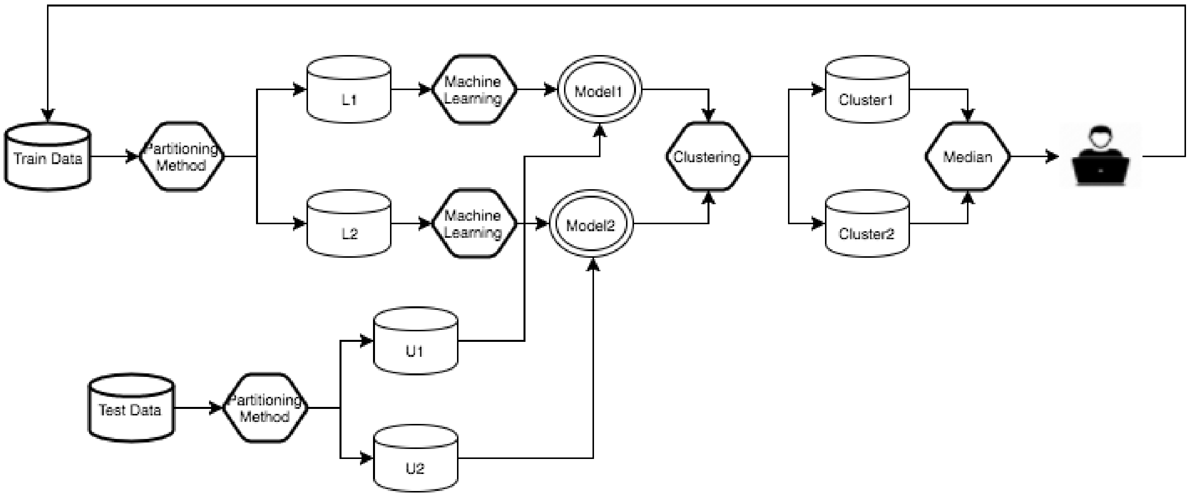

Figure 2.2 : Co-Active Learning algorithm block diagram.

$$|P(y=+|x)-P(y=-|x)|<\theta \tag{2.8}$$

where $|P(y=+|x)-P(y=-|x)|$ is the difference between probability of instance x belonging to classes $+$ and $-$. θ is constant that is used as lower limit for difference between probabilities to be able to selected for clustering.

2.2.2.2 Threshold Setting

In this method, the θ value is a parameter e.g. setting θ to be 0 means to choose instances that our classifiers disagree over it without considering each classifier's confidence. Choosing θ to be close to zero means to take instances that both classifiers have lack of confidence in predicting it. And also having θ to be close to zero, but small than zero mean those instances that both classifiers are less confident about them and also disagree over their label.

2.2.2.3 Co-Active Algorithm Drawbacks

The main problem of Co-Active Learning is that it assumes data points are spread spatially coherent. It means those instances that belong to the same class are closest by euclidean measure and can be clustered easily using k-means. And in the case, that same class members would be in different spatial clusters it would not work fine.

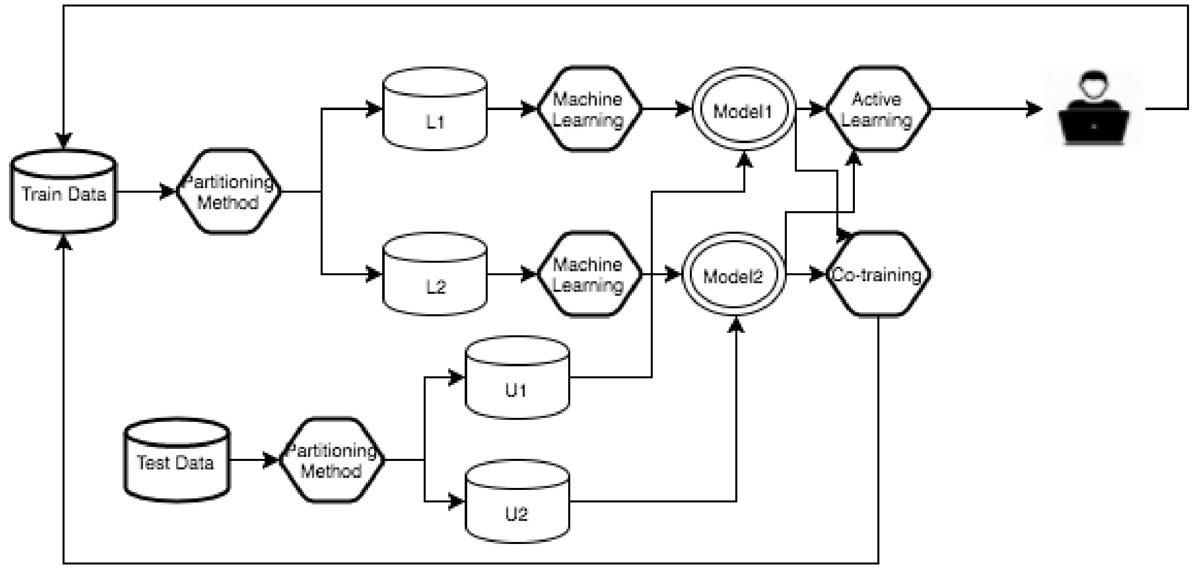

Figure 2.3 : SSLCA algorithm block diagram.

2.2.3 Semi-supervised learning combining Co-training with Active Learning (SSLCA)

As a third method, the exact implementation of SSLCA proposed by Zhang et al. has been implemented [20]. This method has been implemented to ensure a reference point. Unfortunately, we could not obtain the author's code or empirical results and in their paper, they include only a group of figures. Furthermore, they did not give the details of their tests, so the setting that produces the closest results has been implemented. A block diagram of the SSLCA algorithm is given in Figure 2.3.

The largest difference of SSLCA with our method is that in SSLCA they were using Co-training and Active Learning in a parallel architecture. It means at each iteration two Active Learning and Co-training steps take place. In the Active Learning step, most uncertain instance selected to send to oracle to obtain the label. In the Co-training step, most conflicted instance that one of models is certain about the label and the other is uncertain about its label is selected. The certain label is assigned as instance's label. Then these two instances with their assigned label is added to unlabeled dataset. In CEAL algorithm we are getting values from Co-training but do not use them in labeling instances.

Another innovation in our research was the use of Chi-Square, ANOVA, and ReliefF to partition features into two views, in addition to Information Gain.

YK

3. Experimental Results

In this section, we will first give the details of the datasets. Next, we will mention the experimental setup and then conclude with the classification results.

3.1 Datasets

Eight different datasets were used to test the proposed hypothesis. These datasets are published in the University of California, Irvine (UCI) Machine Learning Repository [26]. The UCI Machine Learning Repository is a repository of datasets gathered by University of California, Irvine. Datasets that have been used are in Table 3.1.

3.2 Experimental Setup

Tests on all eight datasets were done in a sort of 10-fold cross-validation. At each fold, 10% of the dataset was set aside for the test, and the remaining 90% was used for training and validation. All of the results are reported using this 10% test and take the average of 10 folds.

From the 90% dedicated to training, 20% of instances were used for training as labeled instances and the remaining 80% was an unlabeled set. The model obtained from training was tested on the unlabeled set, and based on the prediction results and probabilities at each iteration, using three different algorithms, one instance was selected to query. These algorithms' results will be discussed in the next section in detail. This newly labeled instance was added to the training set and its classifiers were retrained.

This setting was done for eight datasets, three methods (CEAL, Co-Active, SSLCA), four partitioning methods (Information Gain, ANOVA, Chi-Square and ReliefF)

Table 3.1 : Properties of the datasets used in the experimental results.

	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
AD	3278	1558	2
voter	434	17	2
vertebral	309	7	2

and using five different machine learning algorithms (Naive Bayes, decision tree, K-nearest neighbor, random forest and MLPs).

3.2.1 Comparison of CEAL with Co-training and Active Learning Algorithms

The results of CEAL were first compared with basic algorithms for Co-training and Active Learning. Naive Bayes and Information Gain were used for the default machine learning algorithm and partitioning method, respectively. For Active Learning, a basic disagreement-based algorithm was implemented, and Co-training was implemented based on [10]. All of the algorithms ran for 100 iterations.

The results are reported in Table 3.2. As we can see, CEAL surpass both Active Learning and Co-training in most cases. Co-training is the weakest method among the three algorithms, and in two cases, it worsened the results after 100 iterations. Active Learning was behind CEAL in every instance except the chess dataset. Each instance in the chess dataset was a set of moves in a chess game. Since the moves were highly related to each other, dividing them into two views caused CEAL perform worse than the Active Learning with a single view did. Co-training worked without any supervision or help from any human/machine agent; therefore, its lower quality is normal, and even impressive.

Table 3.2 : Classification accuracy improvements for the algorithms after unlabeled data utilization. (Base Classifier: Naive Bayes, Feature Splitting: Information Gain).

	Co-training	Active Learning	CEAL
cancer	-0.8	2.2	**3.26**
chess	12.1	**37.7**	30.87
sonar	7.8	28.4	**36.78**
credit	6.5	14.1	**25.47**
ionosphere	-1.7	4.01	**9.16**
AD	10.4	33.3	**38.42**
voter	9.2	23	**45.31**
vertebral	6.9	11.1	**19.22**

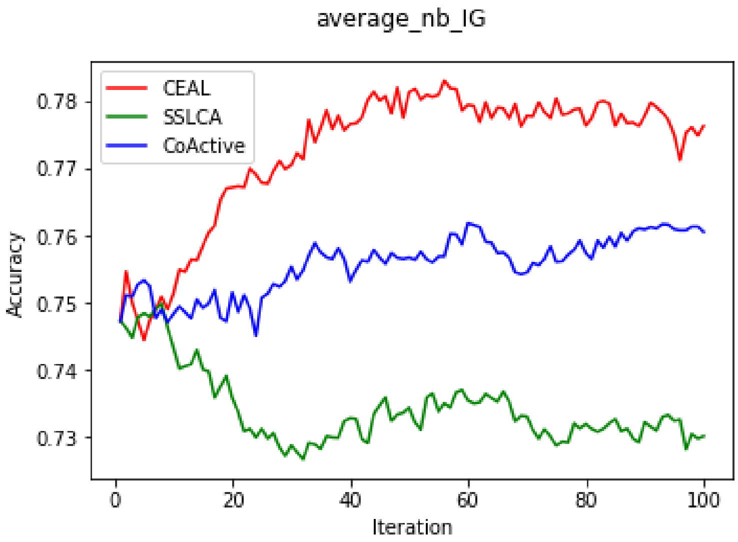

Figure 3.1 : Average classification accuracies over the datasets using Naive Bayes Classifier and Information Gain Splitting.

3.3 Comparison of CEAL, SSLCA, and Co-Active Algorithms

The proposed CEAL algorithm was compared with Co-Active and SSLCA algorithms. Co-Active was proposed in [21] and has been modified for our problem sets, and SSLCA was introduced in [20] and has been implemented as it was reported.

Naive Bayes was considered as the default algorithm and Information Gain as the default partitioning method. The average classification accuracies for the eight datasets are reported in Figure 3.1. The reason for choosing Naive Bayes and Information Gain as default algorithm is to be able to compare with [20] that was working just with these algorithms.

The results show that in most cases, SSLCA drops the quality of the model in its first iteration. This is mainly because of the Co-training step that it has. As we can see, the best result is gained from CEAL, and the best average is from Co-Active.

In Table 3.3 we can see the improvement each algorithm made in 100 iterations. These results are reported in percentages. Of the eight datasets, CEAL outperformed other methods in six datasets and SSLCA in two of them.

3.4 Feature Partitioning Results

Four different methods have been used to divide the datasets into two separate views. These methods are Information Gain, Chi-Square, ANOVA, and ReliefF and are mainly used for feature selection and sharing features fairly between two datasets.

Table 3.3 : Classification accuracy improvements for the algorithms after unlabeled data utilization using different feature splitting algorithms. (Base Classifier: Naive Bayes).

Dataset	SSLCA				Co-Active				CEAL			
	ANOVA	Chi	IG	ReliefF	ANOVA	Chi	IG	ReliefF	ANOVA	Chi	IG	ReliefF
cancer	2.97	2.97	2.97	2.97	2.54	2.54	2.69	2.54	3.12	3.12	**3.26**	3.12
chess	31.22	31.16	30.66	30.93	30.91	30.85	30.45	30.62	**31.32**	31.26	30.87	31.03
sonar	36.19	**37.2**	36.59	35.7	35.9	36.92	36.4	35.41	36.29	37.3	36.78	35.79
credit	21.88	25.81	23.89	**39.94**	21.53	25.48	23.66	34.24	22.0	25.92	25.47	34.63
ionosphere	9.32	9.03	8.89	9.03	8.92	8.62	8.62	8.62	**9.46**	9.16	9.16	9.16
AD	37.95	37.76	38.24	37.1	37.68	37.48	38.06	36.82	38.05	37.86	**38.42**	37.19
voter	45.23	45.23	45.15	45.23	44.99	44.99	44.99	44.99	45.31	45.31	45.31	**45.31**
vertebral	19.1	19.1	18.98	19.1	18.74	18.74	18.74	18.74	19.22	19.22	19.22	**19.22**

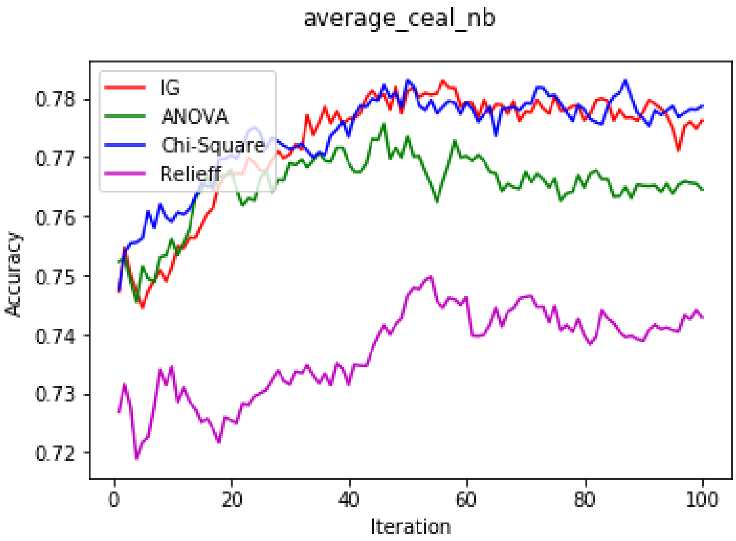

Figure 3.2 : Average Classification accuracies of the datasets using CEAL algorithm and NB as classifier.

As shown in the previous subsection, the CEAL algorithm outperforms its counterparts in terms of classification accuracy. Therefore, in this section, CEAL is set as the default method because it was the best method in the previous section, and Naive Bayes was set as the default machine learning algorithm. The average results for the eight datasets are illustrated in Figure 3.2. Table 3.3 illustrates that the partitioning methods do not create tangible differences, except in a single case, the Credit dataset ReliefF, where they exceeded other methods considerably.Information Gain and Chi-Square are the two methods that generated the best results, and acted very similar to each other.

Information Gain and Chi-Square are the two methods that have the best results, they are acting very close to each other.

3.5 Comparison of Basic Classifiers

The final comparisons were obtained for the base classifiers. In this experiment, the base classifiers of the models were changed and the final classification accuracies were compared.

For this purpose, Naive Bayes, KNN, neural network, decision tree, and random forest classifiers were compared. *K* in KNN algorithm is set to 5. In neural network algorithm the overall optimum architecture was 2 hidden layers and 200 nodes for each layer. For

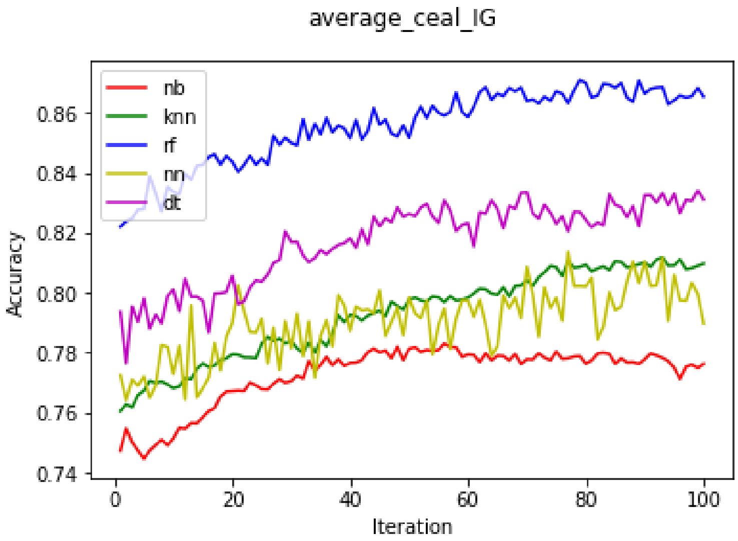

Figure 3.3 : Average Classification accuracies of the datasets using CEAL algorithm and IG as feature splitting algorithm.

random forest 100 decision tree is trained. As for the original SSLCA algorithm, we have used Information Gain as the partitioning method for all algorithms. The final average classification accuracies of the eight datasets are given in Figure 3.3. As shown by the figure, increasing the number of unlabeled data samples increases the average classification accuracies. On the other hand, random forests have the best classification accuracy. The best result is derived from random forest, and the worst result was produced by Naive Bayes.

3.5.1 Classification accuracy improvements of the algorithms after unlabeled data utilization

In Table 3.3, the progress for each combination of methodologies is reported based on Naive Bayes. Each number in the table shows the corresponding combination of methodologies and improvement over 100 iterations in percentages.

As we can see in most datasets, CEAL was out-performing other algorithms. In only two datasets, SSLCA was slightly better than CEAL, and we can see that this is primarily a result of the feature separation algorithm.

4. CONCLUSIONS

Recently, in the big data era, the number of data samples has increased enormously. However, it is difficult to obtain labeled examples for most problems. Therefore, in most cases, it is easy to obtain a few labeled samples and a huge amount of unlabeled examples. In this thesis, the CEAL algorithm was proposed to utilize unlabeled data samples. The proposed algorithm is an iterative algorithm that computes an uncertainty value for each unlabeled instance for query selection. The selected instances are then labeled by an oracle and added to the training set.

The proposed algorithm was compared with the recently proposed SSLCA and Co-Active methods, and the results are used for comparing partitioning methods, query selection strategies and machine learning methods.The algorithms were trained on the same datasets using the same number of unlabeled instance iterations. ll of the different test sets illustrate the superiority of CEAL over the SSLCA and Co-Active methods, as reported in [27]. CEAL uses the power of Co-training to enhance Active Learning and is superior to SSLCA, which uses Co-training and Active Learning in parallel. There is a Co-training factor in SSLCA that may reduce the performance of the overall algorithm when models with a very small subset of labeled data are trained. If the classifiers are trained with few examples and have very low classification accuracies, then the Co-training algorithm may not improve the performance of SSLCA. Zheng et al. trained their model with a far larger portion of training data, though they did not report their exact settings, but we obtained the same results in our tests when SSLCA was trained with 60% labeled data, which is far more than what we are using in our research (5%). On the other hand, the Active Learning algorithm is not dependent on the number of labeled examples and can also work without any labeled datasets. In the experimental results, different feature splitting methods were compared.

Using different partitioning algorithms did not make a huge difference in any datasets except the Credit dataset. In the Credit dataset, ReliefF outperformed the other three algorithms by a high margin. In other cases, however, all of the results were very close to each other, though Information Gain and Chi-Square slightly outperformed the others. Chi-Square was a very computationally efficient algorithm in comparison to ReliefF or Information Gain.

Finding the best machine learning algorithm was not the goal of this study, but the tests were implemented using different algorithms and it was found that, unsurprisingly, random forest was the best algorithm in most cases. Due to their boosting characteristics, random forests are easy to tune and easy to avoid over-fitting, so it obtained reasonable results in most cases. Gaussian Naive Bayes was better than expected, and except in the case of the Ad dataset, it had prominent results. By contrast, neural network was not as good as expected, which may be due to the usage of the same architecture, including layer counts and nodes, in the layers of all datasets. Decision tree also had reasonably good results given its complexity compared to random forest. The use of different algorithms showed us that 100% accuracy can be reached in some cases even with less than 100 iterations.

REFERENCES

[1] **Haque, M.M.**, **Holder, L.B.**, **Skinner, M.K. and Cook, D.J.** (2013). Generalized query-based active learning to identify differentially methylated regions in DNA, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *10*(3), 632–644.

[2] **Settles, B.** (2010). Active learning literature survey, *University of Wisconsin, Madison*, *52*(55-66), 11.

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

[4] **Nigam, K. and Ghani, R.** (2000). Understanding the behavior of co-training, *Proceedings of KDD-2000 workshop on text mining*, pp.15–17.

[5] **Alpaydin, E.** (2014). *Introduction to machine learning*, MIT press.

[6] **Nigam, K. and Ghani, R.** (2000). Analyzing the effectiveness and applicability of co-training, *Proceedings of the ninth international conference on Information and knowledge management*, ACM, pp.86–93.

[7] **Kiritchenko, S. and Matwin, S.** (2011). Email classification with co-training, *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, IBM Corp., pp.301–312.

[8] **Pierce, D. and Cardie, C.** (2001). Limitations of co-training for natural language learning from large datasets, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp.1–9.

[9] **Bishop, C.M.** (2006). Pattern recognition, *Machine Learning*, *128*, 1–58.

[10] **Goldman, S. and Zhou, Y.** (2000). Enhancing supervised learning with unlabeled data, *ICML*, pp.327–334.

[11] **Cohn, D.**, **Atlas, L. and Ladner, R.** (1994). Improving generalization with active learning, *Machine learning*, *15*(2), 201–221.

[12] **Balcan, M.F. and Urner, R.** (2016). Active learning–modern learning theory, *Encyclopedia of Algorithms*, 8–13.

[13] **Settles, B.** (2012). Active learning, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*(1), 1–114.

[14] **Dasgupta, S. and Hsu, D.** (2008). Hierarchical sampling for active learning, *Proceedings of the 25th international conference on Machine learning*, ACM, pp.208–215.

[15] **Seung, H.S.**, **Opper, M. and Sompolinsky, H.** (1992). Query by committee, *Proceedings of the fifth annual workshop on Computational learning theory*, ACM, pp.287–294.

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

[17] **Muslea, I.**, **Minton, S. and Knoblock, C.A.** (2000). Selective sampling with redundant views, *AAAI/IAAI*, pp.621–626.

[18] **Muslea, I.**, **Minton, S. and Knoblock, C.A.** (2002). Active+ semi-supervised learning= robust multi-view learning, *ICML*, volume 2, pp.435–442.

. A disagreement based co-active learning method for sleep stage classification, Systems, Signals and Image Processing (IWSSIP), 2016 International Conference on, IEEE, pp.1–4.

[19] **Cheng, J. and Wang, K.** (2007). Active learning for image retrieval with Co-SVM, *Pattern recognition*, *40*(1), 330–334.

[20] **Zhang, Y.**, **Wen, J.**, **Wang, X. and Jiang, Z.** (2014). Semi-supervised learning combining co-training with active learning, *Expert Systems with Applications*, *41*(5), 2372–2378.

[21] **Yüce, A.B. and Yaslan, Y.** (2016). A disagreement based co-active learning method for sleep stage classification, *Systems, Signals and Image Processing (IWSSIP), 2016 International Conference on*, IEEE, pp.1–4.

[22] **Zheng, Z.**, **Wu, X. and Srihari, R.** (2004). Feature selection for text categorization on imbalanced data, *ACM Sigkdd Explorations Newsletter*, *6*(1), 80–89.

[23] **Guyon, I. and Elisseeff, A.** (2003). An introduction to variable and feature selection, *Journal of machine learning research*, *3*(Mar), 1157–1182.

[24] **Kononenko, I.** (1994). Estimating attributes: analysis and extensions of RELIEF, *European conference on machine learning*, Springer, pp.171–182.

[25] **Kingma, D. and Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980.*

[26] **Repository, U.M.L.**, `https://archive.ics.uci.edu/ml/datasets.html`.

[27] **Azad, P.V. and Yaslan, Y.** (2017). Using co-training to empower active learning, *Signal Processing and Communications Applications Conference (SIU), 2017 25th*, IEEE, pp.1–4.

PHOTO

CURRICULUM VITAE

Name Surname: Payam VAKIL ZADEH AZAD

Place and Date of Birth: 06.08.85 Tabriz, Iran

E-Mail: vakil@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2009, Tabriz University, Computer and Electrical Engineering Faculty

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Azad, Payam V., and Yusuf Yaslan. "Using co-training to empower active learning." Signal Processing and Communications Applications Conference (SIU), 2017 25th. IEEE, 2017.