Semi Supervised Weighted K-Means Clustering for Multi Class Data Classification

Vijaya Geeta Dharmavaram Associate Professor, Department of Operations,
GITAM Institute of Management
GITAM University

Shashi Mogalla
Professor, Department of Computer Science and Systems Engineering
College of Engineering
Andhra University

Abstract

Supervised Learning techniques require large number of labeled examples to train a classifier model. Research on Semi Supervised Learning is motivated by the availability of unlabeled examples in abundance even in domains with limited number of labeled examples. In such domains semi supervised classifier uses the results of clustering for classifier development since clustering does not rely only on labeled examples as it groups the objects based on their similarities. In this paper, the authors propose a new algorithm for semi supervised classification namely Semi Supervised Weighted K-Means (SSWKM). In this algorithm, the authors suggest the usage of weighted Euclidean distance metric designed as per the purpose of clustering for estimating the proximity between a pair of points and used it for building semi supervised classifier. The authors propose a new approach for estimating the weights of features by appropriately adopting the results of multiple discriminant analysis. The proposed method was then tested on benchmark datasets from UCI repository with varied percentage of labeled examples and found to be consistent and promising.

Keywords: *Classification, Semi Supervised Classification, Weighted Metrics, Discriminant Analysis, Potency Index, k-means*

Introduction

Data Mining Techniques are successfully used as prediction models that can predict future trends and behaviors. Some of them explore the dataset for extracting hidden and interesting pattern that provide a comprehensive impression on the dataset. Accordingly, these data mining techniques come under supervised learning techniques and unsupervised learning techniques respectively. Supervised techniques require a large set of labeled examples to build the prediction model. In unsupervised learning, a set of unlabeled examples are grouped into different clusters based on the group similarities which is referred to cluster cohesion. The proximity between every pair of points is estimated in terms of Euclidean distance giving equal importance to all features. Popular clustering algorithm like K-Means uses centroid of sthe cluster cohesion by minimizing sum of squared distances between every member of the cluster to its centroid. However, multiple clustering solutions are acceptable / desirable for a given dataset for varied contexts in which data has to be analyzed. For example, the census data that consists of data objects described in terms of socio, economical, educational, medical etc., types of features needs to be clustered into groups depending on the purpose of the data analysis. Accordingly relative importance of the features varies to provide pertinent clustering solutions; some of the features like economic status and social backwardness play major role for identifying the impact of welfare schemes whereas features like location and commutability etc play a major role while setting up the amenities or community centers. Hence the authors suggest the usage of weighted Euclidean distance metric for estimating the proximity for forming the clusters.

It is often expensive and difficult to get fully labeled dataset which is essential for supervised learning. Even though unsupervised learning doesn't require any training set, it was noted that a dataset can be partitioned in many different ways based on the features used in the clustering process. The clusters thus formed may not be appropriate for one's requirement. However if one can acquire partial labeled dataset, then one can use semi supervised classification which is a combination of supervised and unsupervised learning techniques. It can help in forming uni-class clusters with reduced labeled data.

In this paper, the authors proposes a new algorithm Semi Supervised Weighted K-Means (SSWKM) to define weights of various features in order to get uni-class clusters provided some of the objects has class labels. A uni-class cluster consists of data objects most of which if not all belongs to a single class and can be labeled based on majority class. Such a clustering solution can be used for classifying unknown objects also based on its proximity / membership in one of the labeled clusters.

The algorithm SSWKM discussed in this paper has adopted the K-Means algorithm from unsupervised learning to semi supervised learning that uses few labeled examples to build the classifier. Feature relevancy was identified by multiple discriminant analysis that was later adjusted by weight adjustment equation in the model. The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 shows the methodology adopted in the study. Section 4 discusses the results and in Section 5 gives conclusions.

Related Work

In recent past, much research has focused on semi supervised learning. Basu et al., (2002) used small subset of labeled data to aid the clustering task. These labeled data were used as initial seeding in the K-means algorithm. Two variants of the algorithm were proposed, one in which the initial seeds remain unchanged during the entire process of clustering and in another the seeds are allowed to change their initial clusters.

Xing et al (2003) derived constraints from the classified examples such that the points in the same class will have minimal distance and the points from different classes have larger distance. K-means algorithm was used in conjunction with the modified distance function to form the clusters.

Harbi and Smith (2006) proposed supervised clustering model where K-means algorithm was used in conjunction with simulated annealing to derive weights for the features. Weighted Euclidean distance metric was then used in forming the cluster and thus the classifier. This algorithm is computationally heavy with simulated annealing needing much iteration to arrive at the desired weights.

Yang and Yuan (2009) proposed structured semi-supervised discriminant analysis that exploits the data structures hidden in the class to calculate the intra class differences within the same class. The authors claimed this method is an effective dimensionality reduction method.

Eick et al (2006) proposed weighted k-means algorithm that uses the distance between the majority class examples in a cluster to modify the weights and perform clustering. The authors claimed that their model has given improved accuracy in some of the datasets.

Dharmadhikari et al (2012) proposed model with preprocessing stage that exploits relationship between labeled and unlabeled and perform classification through graph. Semi supervised methods are used in the training stage to propagate labels of labeled documents to unlabeled documents by using energy function.

Soars et al., (2012) proposed cluster-based regularization (ClusterReg) for Semi Supervised Classification that takes the clusters formed by clustering algorithm as a regularization term in the loss function of the classifier. It predicts based on the cluster structure and labeled data.

SSWKM which is proposed in this paper is different from the previous work as it considers labeled examples to get relevant features that best partition the data into classes and uses the weighted features to form the clusters. The algorithm still retains the basic structure of K-means algorithm and thus has the advantage of finding natural groupings in the dataset and also helps in finding intra class differences, since a class can be defined by more than one cluster.

Methodology

When it comes to natural clustering, K-means is a popular algorithm that partitions a dataset into k clusters, locally minimizing the average squared distance between the data points and cluster centers. According to Tan (2006), it is defined as follows:

Consider

$X=\{x_1,x_2,\ldots,x_N\}$, $x_i \in R^d$ - set of data points

$\Phi = \sum_{l=1}^{K} \sum_{x_{i \in X_l}} \|x_i - \mu_l\|^2$ - Objective function

Given the set of data points and objective function, K-means algorithm creates K partitioning $\{X_l\}_{l=1}^{K}$ of X and set of centroids $C = \{\mu_1, \ldots, \mu_K\}$ that minimizes the objective function.

It uses Euclidean distance metric to find the distance between two data points. The equation is given in 1.

$$d(i,j) = \sqrt{\sum_{m=1}^{p}(x_{im} - x_{jm})^2} \quad \text{--- 1}$$

where i= $(x_{i1},x_{12},\ldots,x_{ip})$ and j=$(x_{j1},x_{j2},\ldots,x_{jp})$

For unsupervised learning, Equation 1 is an efficient distance metric where all the features are considered equally important in forming the clusters. However a dataset can contain multiple cluster solutions depending on the purpose of its usage and the features used for clustering will decide how the clusters are formed. Hence to have a optimal cluster solution which is applicable for some chosen purpose in this case clusters based on class labels, only relevant features need to be considered. Some weights can be given to the features to show their significance. The relative significance of different features will contribute to the d istance function. This is termed as weighted distance metric. A weighted Euclidean distance metric is given in equation 2.

$$\delta w\,(x_i, x_j) = \sqrt{\sum_{m=1}^{p} w_m(x_{im} - x_{jm})^2} \quad \text{---- 2}$$

where w_m indicates the weight of the feature. If the significance of the feature is more, its weight will be more.

In the literature many different approaches were proposed for measuring weights such as using weights based on what is judged as important by researchers' understanding of the data or using information gain to give weights the features (Ayan NF, 1999). Most of the research studies show that the researcher starts with some initial value for weight and then iteratively modifies the weight in accordance with the fitness function defined for the cluster. Eick et al (2006)used average distance between majority class examples and overall examples as the measure for weight determination and fitness function. Harbi and Smith (2006) suggest simulated annealing to determine the appropriate set of weights.

All these approaches initially start with a guess/random weights and proceeds further to determine the more acceptable weights. Instead of starting with some arbitrary values, authors have used the multiple discriminant functions to derive the initial weights of the features.

Discriminant analysis is a dimensional reduction technique which can be used to find the predictors that can discriminate between groups. Discriminant analysis linear equation:

$$\mathbf{D=V_1X_1 + V_2X_2 + V_3X_3 + \ldots.. + V_iX_i + a} \quad \text{---- 3}$$

Where

D=Discriminant Function

V= the discriminant coefficient or weight of that variable

X= respondent's score for that variable

a = a constant

i = the number of predictor variable

Good predictors will have large coefficients. Hence discriminant analysis not only determines the relevant features but its coefficients reflect how relevant the feature is.

The values of the features are normalized with Z-scores to perform discriminant analysis and standardized coefficients thus obtained were considered as the discriminant coefficients.

Depending upon the number of classes in the dataset, either binary classifier or multi-class classifier has to be developed.

Case 1: Binary Classifier:

In binary class datasets, we can derive only one discriminant function. This function will provide a linear combination of features that best discriminants the two classes in the dataset. The coefficients thus obtained are used as the initial weights of the features.

Case 2: Multi-Class Classifier:

In multiple class datasets, we get more than one discriminant function where each function represents the separation between two groups. Multiple discriminant functions will give different coefficients to different features. To arrive to a single weight that can best describe the relevancy of the features in discriminating the objects of the groups, we use the notion of potency index.

Potency Index is defined as the *"Composite Measure of the discriminatory power of an independent variable when more than one discriminant function is estimated. Based on the discriminant loadings it is relative measure used for comparing the overall discrimination provided by each step independent variable across all significant discriminant loadings".* (Joseph F Hair Jr et al., 2010)

Potency value includes both the contribution of a variable to a discriminant function and the relative contribution of the function to the overall solution which is based on eigenvalues. Potency index is the sum of the individual potency value across all significant discriminant functions. It shows the relative position or rank of the variable.

Calculation of Potency Index:

Potency index is calculated by a two step process:

Step 1: Calculate a potency value (PV) of each variable for each significant function which is represented by the following equation:

PV of variable i on function j = (Discriminant loading$_{ij}$)^2 X Relative eigenvalue of function j ---- 4

where Relative eigenvalue is calculated as follows: Relative eigenvalue of function j

$$= \frac{Eigenvalue\ of\ discriminant\ function\ j}{Sum\ of\ eigenvalues\ across\ all\ significant\ functions} \quad \text{--- 5}$$

Step 2: Calculate a composite potency index across all significant functions: Composite potency index is calculated by using the following equation:

Composite potency of variable i = Sum of potency values of variable i across all significant discriminant functions --- 6

Calculation of Feature Weights:

Potency index provides the rank or the order of importance of the features that best separates different groups in the dataset. For a dataset with 'n' features placed in the increasing order of relevance, the weight of an feature 'i' is calculated as follows:

$$W_i = \frac{i}{\left(\frac{n(n+1)}{2}\right)} \quad \text{---7}$$

Proposed Semi Supervised Weighted K-Means algorithm (SSWKM)

Dataset will contain both labeled and unlabeled examples. Labeled examples are used to find the discriminant functions and thus the initial weights of the features. The labeled examples along with unlabeled examples are clustered using K-means algorithm with weighted Euclidean metric as distance function. Each cluster is given a class label based on the majority labeled examples found in the cluster. Here number of clusters could be more than number of classes. This ensures that we can understand the intra class differences and the natural grouping or profiling within the same class examples.

Cluster purity is defined as the percentage of examples that are correctly classified to the respective cluster as indicated in equation 8.

$$\text{Cluster Purity} = \frac{\#\ Correctly\ classified\ instances}{\#\ Total\ Instances} \text{ x } 100 \quad \text{--- 8}$$

This is equivalent to the definition of accuracy given in classifiers. The cluster purity is calculated based on the labeled examples in the dataset. For each feature, cluster purity without the feature is measured. This is done to see how much the feature is contributing to the cluster purity. If the new cluster purity is less than the initial cluster purity, it indicates that the absence of the feature has worsened the cluster purity and hence the feature is more significant, and thus its weight is increased. If the new cluster purity is more than the initial cluster purity, it indicates that the feature is not that relevant and thus is removed. The modified weights are normalized such that the sum of weights is 1. Once again, we run k-means with the modified weights and cluster purity is calculated. If there is improvement in the cluster purity, the new weights are accepted. The process continues until there is no change in the cluster purity. It thus follows a step wise refinement in weights. In brief the algorithm is as follows:

Algorithm

Step 1: Perform Discriminant Analysis on the dataset and get the relevant features list with coefficients as weights. A = (W_1x_1,W_2x_2,….W_nx_n)

Step 2: Perform K – means with weighted features to get K-clusters.

Step 3: Calculate the initial cluster purity C_{init}

Step 4: For each feature i in A

Perform K-means without the feature i.

Calculate cluster purity C_i

If $C_i < C_{init}$

$Wi_{new} = Wi\left(1 + \left(\frac{C_{init} - C_i}{C_{init}}\right)\right)$

Else

remove the feature

[this step ensures feature reduction]

End If

Normalize the remaining weights. Perform K-means with the modified weights and calculate cluster purity C_{final}

If $C_{final} > C_{init}$ then

Accept the new weights

Set $C_{init} = C_{final}$

Else

Keep the old weights

End If

Step 5: Perform step 4 till there is no improvement in the cluster purity.

It can be observed that weight modification is done by considering the amount of improvement obtained in the cluster purity with and without the feature.

Experiments and Results

The model was developed in Intel Pentium dual core processor with 3GB of DDR2 667 Mhz memory. Discriminant analysis was performed using SPSS statistics to generate discriminant coefficients, loadings and eigenvalues to calculate the initial weights of the features. The algorithm was coded in VB.Net.

To demonstrate the effectiveness of the algorithm ten benchmark datasets i.e., Breast Cancer, Credit, Ionosphere, PimaIndia Diabetes, Ecoli, Glass, Iris, Wine, Yeast and Zoo from UCI repository and Bankloan dataset from Spss Inc. were taken. The description of the datasets are given in table 1

Table 1 : Description of Datasets

S.No.	Dataset	No of Instances	No of Features	No of classes
Binary Classes				
1.	Breast Cancer	683	9	2
2.	Credit	690	15	2
3.	Ionosphere	351	34	2
4.	Pimaindia Diabetes	768	8	2
5.	Bankloan	700	8	2
Multi classes				
6.	Ecoli	336	7	8
7.	Glass	214	9	10
8.	Iris	150	4	3
9.	Wine	178	13	3
10.	Yeast	1484	8	10
11.	Zoo	101	7	7

The dataset was split into training and testing dataset. To understand how the model works with varied percentage of labeled examples, dataset were split in the ratio of 75:25, 50:50, 25:75 and 15:85. To make it more authentic, the set were split in random fashion using SPSS random select cases option. The training examples were used to derive the weights and for assigning labels to cluster. Once the final clusters are formed, cluster purity was calculated for the test examples.

The final test results are compared with the other classifier models. The other models considered are Weka implementation of Bagging, Multiboost, Random Forests and C4.5. The classifiers were run by taking the specified percentage of labeled examples as training set and rest as test set. The results are tabulated in table 2 and comparison graphs for a few datasets are shown in fig 1.

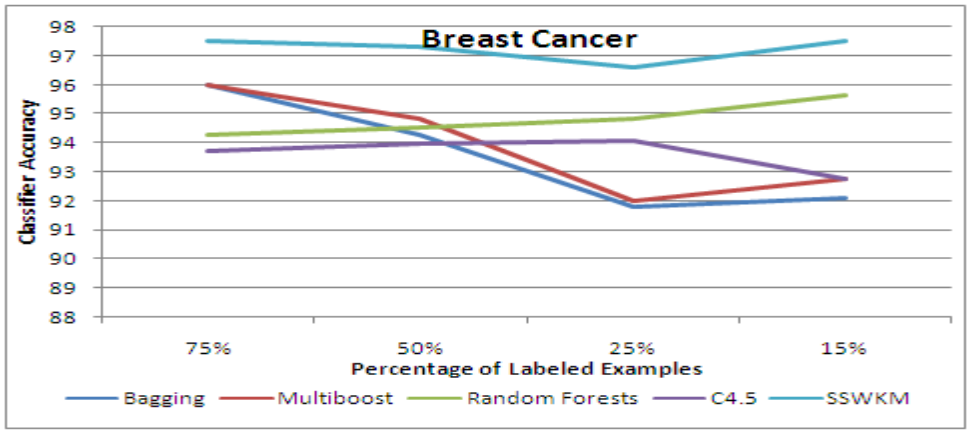
Table 2: Results Summary

S.no		Dataset	Bagging				MultBoost				Random Forests				C4.5				SSWKM			
			75	50	25	15	75	50	25	15	75	50	25	15	75	50	25	15	75	50	25	15
1	BINARY	Bcancer	96	94.26	91.79	92.1	96	94.84	91.98	92.8	94.28	94.55	94.84	95.6	93.7	93.98	94.08	92.8	97.5	97.3	96.6	97.5
2		Credit	58.72	55.07	50.46	54.6	87.2	84.63	83.36	84.3	80.23	74.49	78.91	74	87.7	84.63	83.9	84.1	86.6	85.9	85.6	85.7
3		Ion	88.63	83.42	92.01	87.6	80.79	77.71	84.79	82.2	87.5	92.57	87.07	82.2	80.68	86.28	87.83	82.2	86.8	90.5	88.5	86.5
4		pima indian	79.16	74.47	72.76	70.9	78.12	76.82	70.65	73.2	75	74.21	68.92	64.9	77.08	74.21	69.79	66.2	76.5	75.8	76.9	77.1
5		Bankloan	80.57	80.85	76.19	76.3	78.85	80	76.19	76.5	80	78.28	76.76	77.5	81.14	76.85	74.09	77.5	79.11	77.3	77.9	77.4
6	MULTIPLE	ecoli	92.85	82.14	83.33	82.2	75	67.85	67.06	66.1	90.47	85.11	81.74	74.1	82.14	81.54	81.74	68.9	87.1	84.4	84.7	87.4
7		glass	73.5	71.96	66.87	59.9	43.39	31.7	34.3	48.9	77.35	70.09	69.37	63.2	54.71	65.42	60.62	61	71.8	69.9	67.6	70.9
8		Iris	89.18	94.66	94.64	92.1	89.18	94.66	94.64	92.1	91.89	96	94.64	92.1	89.18	94.66	94.64	93.7	97.9	97.9	97.9	97.3
9		wine	97.2	91.01	88.72	84.8	95.45	91.01	87.21	86.1	97.72	94.38	90.22	87.4	95.45	94.38	86.46	84.1	96.6	97.7	94.3	94.9
10		yeast	61.45	60.1	56.78	51.8	49.97	60.1	40.07	40.5	58.22	55.66	53.27	50.8	57.95	55.25	51.66	50.8	56.6	58.5	57.2	56.6
11		zoo	88	82	78.94	79.1	57	82	57.89	50	92	82	86.84	80.2	92	82	78.94	80.2	93.9	92	87	89

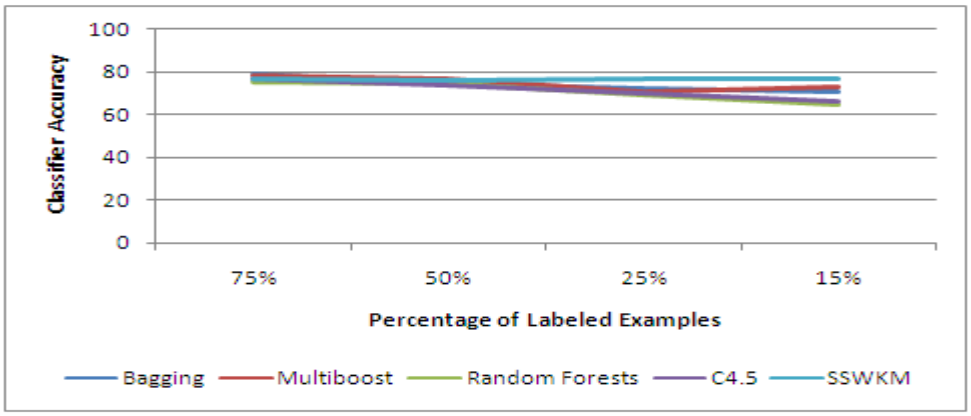
Fig 1: Comparison of performance of few sample datasets:

Binary Classes:

1. Breast Cancer

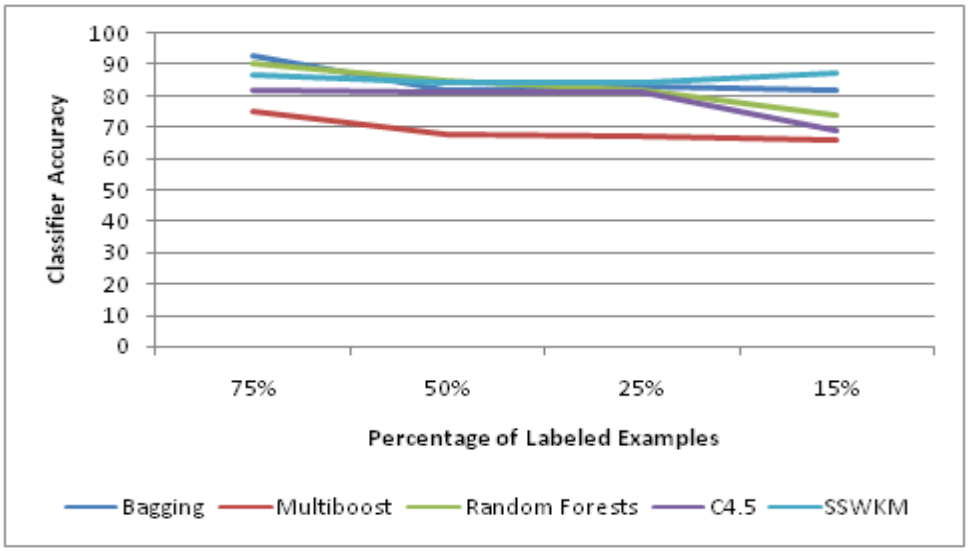


2. Pima Indian Diabetics

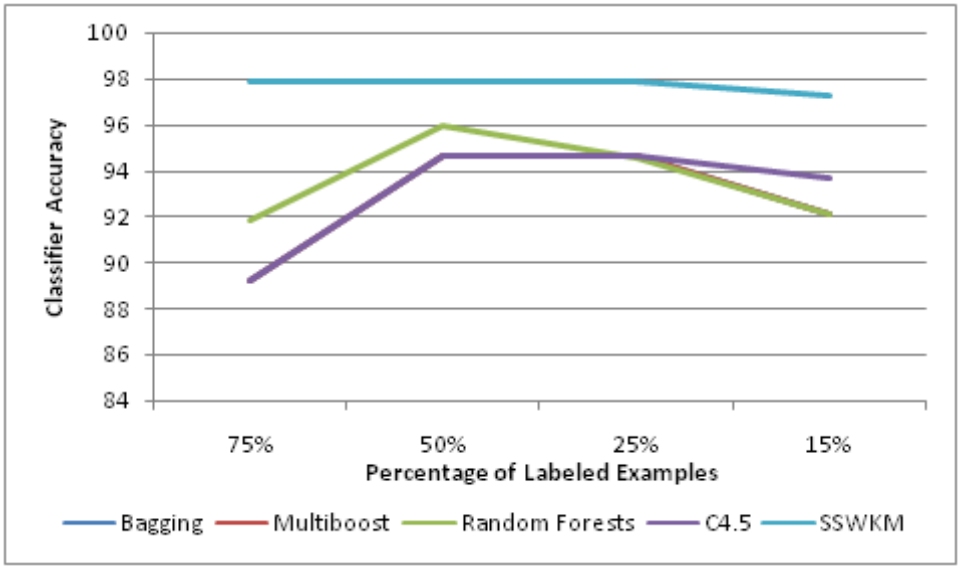


Multiple Datasets:

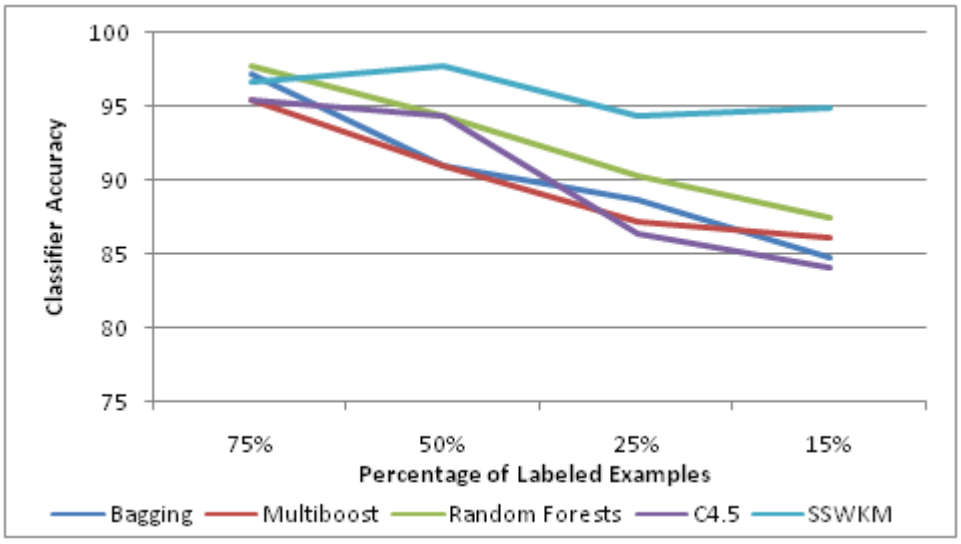
1. Ecoli



2. Iris



3. Wine



The results indicate that SSWKM has given consistent performance. With minimal number of labeled examples, it has achieved better results in most of the datasets when compared with other classification models. It was able to sustain its performance even with less percentage of labeled examples thus making it suitable as a semi supervised classifier.

Conclusion

In this paper, K-means algorithm was adopted for building semi supervised classifier. The Euclidean distance metric was modified as weighted Euclidean distance metric with weights based on the relevance of features for identification of class labels obtained by discriminant functions. Feature reduction was made a part of the model where the irrelevant features are eliminated based on their contribution to the overall cluster purity.

The performance of SSWKM algorithm in terms of accuracy was compared with the other promising classifier techniques. It is observed that SSWKM has given better results when compared with other classifiers even with less number of labeled examples and was found to be most suitable for semi supervised learning as its performance was consistent on a wide ranging percentage of labeled examples.

References

1. Ayan NF (1999) Using information gain as feature weight. In: Proceedings of the 8th Turkish symposium on artificial intelligence and neural networks (TAINN'99), Turkey

2. Basu S, Banerjee A, Mooney R (2002) Semi-supervised clustering by seeding. In: Proceedings of the 19th international

conference on machine learning (ICML-2002), Sydney, Australia.

3. Dharmadhikari Shweta C., Ingle Maya, Kulkarni Parag (2012), A Novel Multi label Text Classification Model using Semi supervised learning, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.4, July 2012, pp. 11-20

4. Eick , Christoph F, Rouhana Alain, Bagherjeiran A, Vilata R (2006), Using Clustering to learn distance functions for supervised similarity assessment, Engineering Applications for Artificial Intelligence, 19, pp. 395 – 401.

5. Harbi Al, S.H., Smith Raymond V.J., Adapting K-Means for Supervised Clustering, Applied Intelligence, 24, pp: 219-226

6. Joseph F. Hair Jr, William C. Black, Barry J. Babin, Rolph E. Anderson (2010), Multivariate Data Analysis, Prentice Hall, 7th Edition

7. Soars Rodrigo G. F., and Chen Huanhuan, Semi Supervised Classification with cluster regularization, IEEE Transactions on Neural Networks and Learning Systems,

8. Tan Pang-Ning, Steinbach Michael, Kumar Vipin (2006), Introduction to Data Mining, Pearson Education, New Delhi

9. Xing, E.P., Ng A., Jordan, M., Russell, S. Distance Metric Learning with Applications to clustering with side information, Advances in Neural Information Processing 15, MIT Press, 2003

10. YangMing, Yuan Xing-Mei, (2009), Structured Semi-Supervised Discriminant Analysis, Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, 12 -15 July, 2009