Journal of Computer Science and Applications.
ISSN 2231-1270 Volume 6, Number 2 (2014), pp. 79-89

http://www.irphouse.com

A Novel Approach to Improve TNNC Using Leaders Algorithm

P.Venkatasaichandrakanth[1]
PG scholar, Dept. of CSE
Madanapalle Institute of Technology and Science
Madanapalle,India
pvsck.36@gmail.com

Y.C.A.Padmanabha Reddy[2]
Asst.Professor, Dept. of CSE
Madanapalle Institute of Technology and Science
Madanapalle,India
padmanabhareddyyca@mits.ac.in

S.Mohommad.ghouse[3]
Asst.Professor, Dept. of CSE
Madanapalle Institute of Technology and Science
Madanapalle,India
ghouse.smd@mits.ac.in

Abstract

During last few years we have gone across many Machine Learning algorithms, i.e. Supervised Learning ,Unsupervised Learning and Semi Supervised Learning algorithms, now we introduced and implemented a Leaders Algorithm to increase the Accuracy and reduce the Time Complexity , we have a Training dataset Consists of both positive and negative labels, in which the features of Training pattern having positive label will group into one cluster, and remaining features of Training pattern having negative label will grouped into another cluster . After that we have to find the mean for each cluster. such that proposing the centroid as a leader and applying Euclidean distance and label the remaining Unlabeled pattern and it is compare with the existing methods nearest neighbor classifier(NNC), Iterative Dichotomiser3(ID3),Graph min cut, TNNC of accuracy's are compared and these existing methods achieve low LOOCV error with respect to nearest neighbor based classifiers. In these paper shows that, through a Leader's algorithm, it is feasible to obtain different solutions having zero leave-one-out

cross-validation error with respect to nearest neighbor based classifiers.

Keywords— semi-supervisedlearning; graph-mincut; nearest neighbor classifier; leaders algorithm;

I. Introduction

Data Mining refers to Extracting the Knowledge from large amount of data [15]. The key concepts (or) functionalities of Data Mining are Classification, Clustering, Frequent Pattern- Mining, Characterization & Descritization, Outlier analysis [15]. Data Mining Process can be done by human on a Particular dataset. If the dataset is massive, complicated or may have some problems with the process of going or dataset. To avoid this machine is needed to perform that type of tasks.

In 1950, Alan Turing was Proposed a test to machine. The main aim of that test was to determine how much efficiency that the machine works as like as human. To improve the efficiency we need some learning strategies. These learning strategies are known as machine learning algorithms[20]. The main aim of artificial intelligence is to develop a machine that can do or think like a human at the particular time instance. Artificial Intelligent System will be build by considering machine learning algorithms only. Machine Learning means to improve the efficiency of the system or machine by learning or training with some complex, critical datasets.

Machine learning can use data mining Techniques to develop models or algorithms for system performance or improvement purpose. Machine learning algorithms face a problems due to lack of sufficient labeled data. In the market luckily we have large amount of Unlabeled data available, but we have small amount of labeled data available to do well with machine learning algorithms. In these paper we mainly concentrate on the increasing the accuracy and reducing the efficiency of the Leaders algorithm with the help of centroid, and electing the centroid as the leader, by use of these leader to decrease the process or efficiency, as we come up to SSL(semi supervised learning) only for the reason that of small amount of labeled data.

II. Definitions

A. Class labels: Class labels can be used for Identification purpose. Normally it consist of Positive and Negative labels.

B. Training set: Training set is also known as Labeled set. The Size of Training set is L, and It is a Collection of Both Positive and Negative Labels.

C. Test set: Test set is also known as Unlabeled set. The Size of Test set is U, and It is also a Collection of Both Positive and Negative Labels.

D. Classification Accuracy: This is Calculation Accuracy over the Unlabeled set. The Classification Accuracy on given Unlabeled set (test set) is the Percentage of Unlabeled set patterns or Tuples that are Correctly Classified by the Classifier.

III. K-means

k-means is the one of the most important Partitioned based method [15]. In the k-means partitions the number of objects into grouping of k (k is the no of clusters) clusters. The k-means partition algorithm mainly focused on the centroid of the cluster. By using this k-means partition algorithm first to find the centroid of Training dataset in which features having positive and negative labels. k-means idea is mainly help us to increase the accuracy of Leaders algorithm. The centroid of the clusters help us to decrease the labeling data sets which help to build a improved algorithm [15].

The procedure we follow to perform the k-means as follows.

1. Randomly choose k objects from unknown data which are considered as centers for clusters.
2. Assign each object to the center which is similar to it by considering the mean value.
3. Calculate the means once again after adding the objects.
4. Repeat step 2 &step 3 until the mean values does not change.

By performing the k-means, the resulting inter cluster similarity is low but the intra cluster similarity is high. Among the all Unsupervised learning types k-means is one of the most easiest method.

We get two centroids if the data sets are binary .

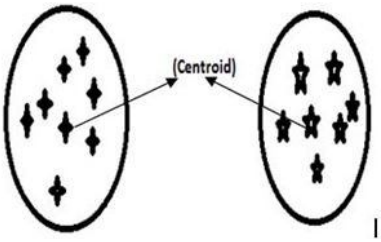

Fig. 1. K-Means

IV. Minimizing LOOCV(leave-one-out cross-validation) error

Why may the mincut methodology be a sensible one to effort? In this part, we inspire this methodology by considering the objective of assigning the label to the unlabeled information with a specific end goal to increase the "enjoyment" of some given learning algorithms [7].

A. we will demonstrate two related yet actually various types of consequences:

1. For certain learning algorithms A , we can characterize edge weights so that mincut calculation creates a naming(labeling) of the unlabeled information that (out of all conceivable such marking) brings about A having the minimum LOOCV error when connected to the whole data set L□U.
2. For certain (other) learning algorithms A, we can characterize edges weights so that the mincut algorithm's marking(labeling) consequences in A having Zero LOOCV error when just illustration in U are held out.

The sort of learning algorithms we will have the capacity to handle the majority of the closest neighbor style. We start with a basic aftereffect of sort (1) for the fundamental 1-closest neighbor algorithms[7].

A. Theorem1: The edge weights between the example nodes have to be defined as mentioned below:

Here define $nn_{xy}=0$ for each and every pair of node x and node y otherwise let $_{xy}+nn_{yx}$. After that for several binary labeling of respective examples $x \in U$ the cost of associated cut should be equal to the number of leave one out cross validation mistakes made by 1-nearest neighbor on $L \cup U$.

The above theorem generally minimizes values of the cut with respect to minimizing LOOCV error.

Proof: The Binary labeling *f(x)* of unlabeled examples have to be fixed $x \in U$. The error rate of LOOCV for particular 1-nearest neighbor is defined as the number of $x \in L \cup U$. The rate should be calculated with the main intention as the label of *x* is different from the label of *x*'s nearest neighbor. The obtained result is the sum, then the total ordered pairs (*x,y*) where as *x* has a different label and *y* has different label with respective of nn_{xy}. After that the cut value that was produced have to be classified by placing the positive examples in to V_+ and after placing negative examples in to V.

After that it is necessary to extend the result to the value of k-nearest neighbor algorithm. There is one problem in that respective which is nothing but, unfortunately the majority-vote operation of kNN causes a problem. The solution to that is to substitute the majority-vote operation with average values. The averaging *k*NN is defined as the algorithm which examines the *k* nearest neighbors for given test examples *x* and after that it predicts the fraction *t/k*, where *t* is the number of positive examples in the set. For a given set of some labeled examples S and for a test example *x* here a locally- weighted averaging have to be defined as an algorithm which predicts the label of *x* based on average weight of the example labels in S. We can use the *k* nearest examples to *x* as averaging of *k*NN, or, for instance, we could weight examples as some function of this distance from *x* [7].

V. Transductive Nearest neighbor Classifier

If we consider a set of fruits as training set and one fruit apple as test set. Here we provide a label for apple. the fruit apple having a some characteristics that are similar to characteristics of fruits of training set. By taking this idea the NNC evolved. NNC stands for Nearest Neighbor Classifier. It is one of the Lazy Learner. In this the training set acts as model or classifier. In this we can classify test or unknown pattern by measuring the distance between the training pattern and test pattern. In NNC There is no predefined model because it is an Lazy Learner. First we can plot the all training data and test data into a n-dimensional space then we compute the distance among the one test pattern to all training patterns. Among those distances we consider which is the minimum value take the label of that training pattern and it has assigned to the test

pattern [15]. The closest neighbor classifier (NNC) groups given test example as per its closest neighbor in the preparation set. This is possible in the accompanying way additionally, in each one class of preparing examples closest neighbors are discovered independently, and the closest neighbor is found afterwards.For example 'x' be the test, let its closest neighbor's separate in L+ be d+(x), and in L− be d−(x). At that point, class-name relegated to x is = +1 if d+(x) < d−(x), = −1 generally. The best way to measure this assignment of goodness (γ) is

$$\gamma(x, y') = \begin{cases} \left(d_-(x) - d_+(x)\right) & if \;\; y' = +1 \\ \left(d_+(x) - d_-(x)\right) & if \;\; y' = -1 \end{cases} \quad (4)$$

This above (4) equation can be simplified and modified as

$$\gamma(x, y') = y'\left(d_-(x) - d_+(x)\right) \quad (5)$$

In this Scenario, Hereditary Algorithms could be utilized which takes over the top measures of time. Rather, a paper proposes a heuristic based inexact method to take care of this issue. This is an iterative incremental marking system. For every one example x U a score s(x) = |d+(x) − d−(x)| is given. Here let us consider just L to discover closest neighbors of x in both classes. To its closest neighbor's name the most noteworthy score is added to L alongside. Till all examples in the test set, alongside their closest neighbor's mark, are added to L this methodology will be rehashed.

VI. Leader Algorithm with *K-Means*

we can find the centroids by using k-means whatever we explained the above process. we go away with this k-means process[15] only as to elect a leader. we go away with the methodology to increasing the labeled datasets as 15,25,35,40 and each and every time we elect the leader as whatever we will get the centroid, these leader vector is taken as the labeled data to the leaders algorithms, by these process we increase the accuracy and the consequences are plotted.

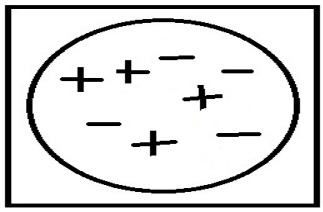

Fig. 2. Training Data Set

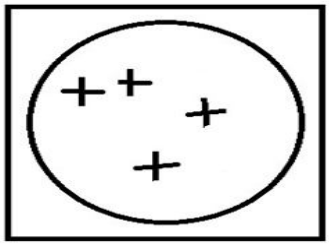

Fig. 3. Separating the positive label from training dataset kept it in cluster

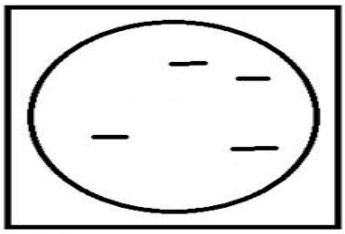

Fig. 4. Separating the negative label from training dataset kept it in cluster

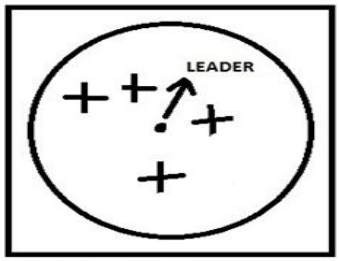

Fig. 5. finding the mean to the cluster & electing the centroid as leader

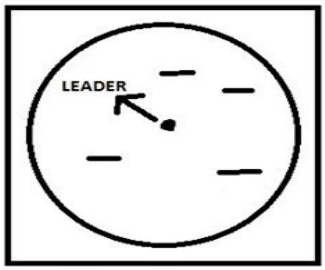

Fig. 6. finding the mean to the cluster & electing the centroid as leader

ALGOTITHM:
INPUT: Training data with labels
OUTPUT: Test data with labels
Procedure:

1.In Train data we already have a labels i.e. we can consider either 0(positive) or 1(negative) labels.
2.In the next step we will separate the positive labels and negative labels.
3.After that we will kept the positive labels and negative labels in separate clusters.
4.After separating the both labels we can find the mean for the each cluster whatever the labels present in the cluster.
5.After finding the mean we will get centroid in each cluster.
6.In next step find the Euclidean distance from each cluster in centroid to the each pattern in the test dataset.
7. Among that the minimum distance is retrieved . Find the label which will gives the minimum distance. i.e. either positive or negative label.
8.Assign the label for the test pattern.
9.Repeat the above process for all test patterns.

VII. EXPERIMENTAL RESULTS

The five standard data sets with testing experiments on algorithms are organized . The five data sets are present in UCI Machine Learning Repository [13]. That are listed below.

TABLE I. Data-sets with no. of features, labeled and unlabeled

Data-set	Number Of Features	|L|	|U|	Distance Function
VOTING	16	45	390	Jaccard Coefficient
MUSH	22	20	1000	Simple Matching
IONO	34	50	300	Euclidean
BUPA	6	45	300	Euclidean
PIMA	8	50	718	Euclidean

Note: same data-sets are used in [1], [7] and [8]. The classifiers used for the comparison purpose are, (i) graph mincut-±opt [7] (a transductive classifier), (ii) randomized graph mincut [8] (a transductive classifier), (iii) spectral graph partitioning [9] (a transductive classifier), (iv) ID3 (a decision tree based classifier, an inductive classifier) [14][15], (v) 3-NNC (3-nearest neighbor classifier, an inductive classifier) [16], (vi) SITNNC[1] vii)SITNNC with Centroid (the proposed method of

this paper, a transductive classifier). Classifiers for comparison are chosen so as to compare with other transductive methods which are similar to the proposed method of the paper.

TABLE II. CA (%) FOR VARIOUS CLASSIFIERS

Data-set	Mincut -? opt	Rand. Graph Mincut	Spectral Graph Prtit.	ID3	3-NNC	Leaders with K-Means
VOTING	90.4	90.3	87.9	89.3	88.7	**93.6**
MUSH	**96.8**	94.3	91.7	93.2	91.0	97.9
IONO	81.7	82.9	79.8	**88.5**	69.6	86.68
BUPA	59.2	63.6	61.7	55.4	52.5	**69.68**
PIMA	72.4	67.6	68.3	69.8	68.2	**74.8**

By using the table II the below graph is plotted. where the values of X axis indicates the datasets[13] whatever present in the above table II. Y axis indicates the classification accuracy retrieved by each algorithm.

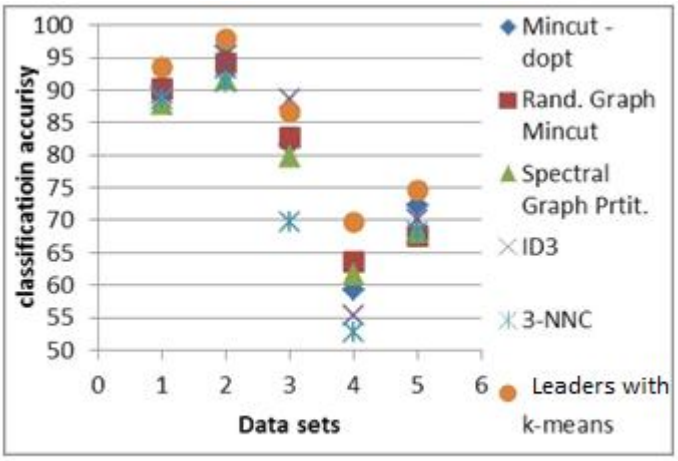

Fig. 7. Classification Accuracy of Data sets

TABLE III. Time Complexity of Data ets

Data-set	E-TNNC (T.C)	TNNC with K-Means (T.C)
VOTING	0.092	0.072
MUSH	0.424	0.080
IONO	0.027	0.020
BUPA	0.030	0.022
PIMA	0.127	0.070

By using the table III the below graph is plotted. where the values of X axis indicates the datasets whatever present in the above table III. Y axis indicates the time complexity retrieved by each algorithm.

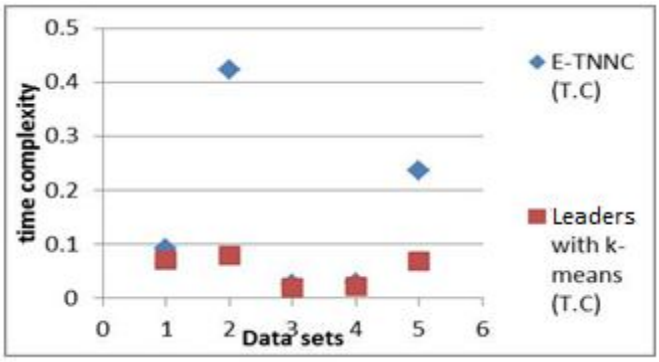

Fig. 8. Time Complexity of Data sets

VIII. Conclusion

The same as on top of indicated methodology will give better performances when the information sets are nearly bounded and if any not in layer vectors come into boundary the centroid which we are picking will as a test information's accuracy will decrease fundamentally.

References

[1] O. Chapelle, B. Scholkopf, and A. Zein, *Semi- Supervised Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[2] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. S. lkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. S. lkopf, Eds., vol. 16. Cambridge, MA: The MIT Press, 2004, pp. 321–328.

[3] Vapnik, *Statistical Learning Theory*. John Wiley & Sons: A Wileyinterscience Publication, New York, 1998.

[4] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, 2nd ed. New York: Springer Series in Statistics, Springer-Verlag, 2006.

[5] K. Bennett, "Combining support vector and mathematical programming methods for classification," in *Advances in kernel methods – support vector learning*, B. Scholkopf et al., Ed. MIT-Press, 1999.

[6] T. Joachims, "Transductive inference for text classification using support vector machines," in *Sixteenth International Conference on Machine Learning*. Bled Slovenia: Morgan Kaufmann, 1999, pp. 200–209.

[7] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincut," in *Eighteenth International Conference on Machine Learning*. Morgan Kaufmann, 2001, pp. 19–26.

[8] A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy,"Semi-supervised learning using randomized mincuts," in *International Conference on Machine Learning*. Morgan Kaufmann, 2004.

[9] P.S. Bradley and Usama M. Fayyad: Refining initial points for k-means clustering. In *Proceedings Fifteenth International Conference on Machine Learning*, pages 91-99, San Francisco, CA, 1998, Morgan Kaufmann.

[10] X. Zhu, Z. Gharahmani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *20th International Conference on Machine Learning*, 2003, pp. 912–919.

[11] A. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264– 323, 1999.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University Press, 2000.

[13] P.M.Murphy, *UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/mlearn/MLRepository.html]*, Department of Information and Computer Science, University of California, Irvine, CA, 2000.

[14] R. O. Duda, P. E.Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons: A Wiley-interscience Publication, 2000.

[15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Academic Press, 2001.

[16] B. V. Dasarathy, "Data mining tasks and methods: Classification Nearest-neighbor approaches," in *Hand*

[17] *book of data mining and knowledge discovery*. New York: Oxford University Press, 2002, pp. 288–298.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA, U.S.A: The MIT Press, 1990.

[19] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge UK: Cambridge university Press, 1995.

[20] http://en.wikipedia.org/wiki/Machine_learning Machine Learning.

[21] A book titled "*Enhancement to Selective Incremental Approach for Transductive Nearest Neighbour Classification*", ISBN 978-3-656-342502, published by GRIN Verlag GnbH, Norderstedt Germany.

[22] S.Md Ghouse, Y.C.A Padmanabha Reddy, E. Madhusudhana Reddy, (2012), "An Enhancement for Tranaductive Nearest Neighbor Classification" International journal of Advanced Computing (ISSN: 2233-2433), 2012, Volume 35, Special Issue 2, PP 361-366.