KB-CB-N Classification: Towards Unsupervised Approach for Supervised Learning

Zahraa Said Abdallah
Centre for Distributed Systems and Software Engineering,
Monash University,
900 Dandenong Rd, Caulfield East, VIC3145
Australia
Email:zsabd1@student.monash.edu

Mohamed Medhat Gaber
School of Computing
University of Portsmouth
Portsmouth, Hampshire, England, PO1 3HE
UK
Email:mohamed.gaber@port.ac.uk

***Abstract*—Data classification has attracted considerable research attention in the field of computational statistics and data mining due to its wide range of applications. *K Best Cluster Based Neighbour (KB-CB-N)* is our novel classification technique based on the integration of three different similarity measures for cluster based classification. The basic principle is to apply unsupervised learning on the instances of each class in the dataset and then use the output as an input for the classification algorithm to find the *K* best neighbours of clusters from the density, gravity and distance perspectives. Clustering is applied as an initial step within each class to find the inherent in-class grouping in the dataset. Different data clustering techniques use different similarity measures. Each measure has its own strength and weakness. Thus, combining the three measures can benefit from the strength of each one and eliminate encountered problems of using an individual measure. Extensive experimental results using eight real datasets have evidenced that our new technique typically shows improved or equivalent performance over other existing state-of-the-art classification methods.**

I. Introduction

In recent times, there has been an explosive growth in the amount of data that is being collected in the business and scientific arena. Data mining techniques can be used to discover useful patterns that in turn can be used for classifying new instances of data [1]. Classification is an important problem for machine learning and data mining research communities. The basic idea of a classification algorithm is to construct a classifier according to a given training set. Once the classifier is constructed, it can predict the class value(s) of unknown test data sample(s).

Classification techniques have attracted the attention of researchers due to the significance of their applications [2], [3]. A variety of methods such as decision trees, rule based methods, and neural networks are used for the classification problems. *KNN* (K Nearest Neighbour) [4], [5] is a simple, but yet effective classification method. The main idea is finding *K* nearest instances in the training sample to classify any unlabelled data instance. *KNN* has been chosen by the data mining community among the top 10 data mining algorithms [6]. However, there are some problems that negatively affect the performance of *KNN*. One of these problems that has a clear negative impact on the classification performance of *KNN* is the use of standard Euclidean distance in finding the nearest neighbours [7]. In this paper, we claim that combining other similarity measures will help enhancing the performance of *KNN* classifier. Another shortcoming in *KNN* is being an instance based learner. Therefore, the decision of classifying an unlabelled instance relies on the individual training instances similar to the test sample. Hence, the classification accuracy is directly affected by noise and training samples individual accuracy.

We present in this paper a novel form of optimisation for *KNN* classifiers by two significant contributions. The first one is the idea of cluster-based classification. For each class label, unsupervised learning is applied for instances that belong to the same class and the primary sampled training instances are replaced by representative sub-clusters. The classification decision for unlabelled samples relies on sub-clusters instead of instances. That directly assists enhancing the classification performance and reducing the effect of noisy data. Moreover, that helps discover hidden patterns and categories within individual classes. The second contribution is based on the improvement of the similarity measure in *KNN*. While traditional *KNN* classification techniques typically employ Euclidean distance to assess pattern similarity and choose the nearest neighbour, other measures may also be utilised to improve the accuracy and find the best neighbour among sub-clusters in all labelled classes instead of nearest one. We coined our novel technique as *K* Best Cluster Based Neighbour *(KB-CB-N)*.

The remainder of this paper is organised as follows. Section 2 discusses basic concepts necessary to introduce the novel algorithm. Section 3 surveys the related work. In Section 4, the new method is introduced in details. Section 5 presents and discusses the performance of the novel *KB-CB-N* algorithm through extensive experimental evaluation. The paper is finally concluded in section 6.

II. Background

The proposed algorithm is an optimisation and enhancement approach of the *K*- nearest neighbour classifier. In the classical *KNN* algorithm, the training samples are described by n dimensional attributes. Each sample represents a point in an n dimensional space. When given an unlabelled sample, a k-nearest neighbour classifier searches the pattern space for the k training samples that are closest to the unlabelled sample. ”Closeness” is defined in terms of Euclidean distance, where the Euclidean distance between two samples with n dimensions, X=(x_1,x_1,.,x_n) and Y=(y_1,y_2,.,y_n) is

$$D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \ . \quad (1)$$

An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours. *k* is a positive integer, typically small. If *k* = 1, then the object is simply assigned to the class of its nearest neighbour. Our novel *KB-CB-N* algorithm is a cluster-based classifier which is based on applying a clustering technique within each class while training before the classification step. Traditionally clustering techniques are broadly divided into hierarchical and partitioning [7]. The EM algorithm [8] is a general technique for finding maximum likelihood estimates for parametric models when the data are not fully observed. EM has been well studied for unsupervised learning and has been shown to be superior to other alternatives for statistical modelling purposes [7]. The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely [9].

Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximised under the assumption that the missing data are known. The estimate of the missing data from the E-step is used in lieu of the actual missing data. The process continues until log-likelihood convergence is achieved. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration [10].

III. Related Work

Many techniques have been applied for classification, including decision trees [11], neural network (*NN*) [12], support vector machine (*SVM*) [13], K nearest neighbour (*KNN*) [4] and many other techniques. *K* nearest neighbour has been widely used as an efficient classification model; however it has many shortcomings [14]. Many methods have been developed to improve the *KNN* performance, including Weight Adjusted K-Nearest-Neighbor (*WAKNN*) [15], Dynamic K-Nearest-Neighbor (*DKNN*) [14], K-Nearest-Neighbour with Distance Weighted (*KNNDW*) [9], and K-Nearest-Neighbour Nave Bayes (*KNNNB*) [14]. The main contributions of the above techniques are how to improve the distance similarity measure function, select neighbour size, and enhance voting system [14]. However, combining various similarity measures for classification purpose has not been addressed. Other similarity measures have been applied individually in the decision rule for classification purpose. For example density similarity measure used in density based classification [16] and data gravitation based classification [17].The classification method proposed in [18] with varying similarity measures (Euclidean distance, cosine similarity, and Pearson correlation) represents the first attempt. However, combining between other similarity measures like density and gravity has the potential to present a better view of the data distribution and hidden patterns within the training samples.

KNN is considered an Instance Based Learning method *IBL* which predicts the label of any test samples by voting among *K* individual training instances. Alternatively, some techniques have been developed to replace the individual training samples by set of clusters as in [19] in order to improve the classification performance. However, the similarity measure used in this class-based clustering algorithm is only distance but not considering merging other similarity measures like density and gravity. The combination of density, gravity and distance similarity measures have been first introduced in our earlier work in [20], but for clustering purposes.

IV. KB-CB-N Classification

In this section we introduce our novel *KB-CB-N* classification algorithm. In Section 4.1, the outline of the proposed algorithm is described. The various similarity measurements applied are introduced in Section 4.2. In Section 4.3, we devise the new classification algorithm

A. Algorithm outline

KB-CB-N classification algorithm is composed of two principle phases: clustering and prediction. The contributions of the proposed algorithm can be summarised in the following:

1) The idea of cluster-based classification improves the accuracy of prediction decision as the classification decision is based on a set of instances rather than individual samples.
2) The clustering technique applied within each class makes a significant effect in finding hidden patterns and related features among the class objects and this consequently leads to higher classification performance.
3) Applying various similarity measures in the prediction phase gives more accurate anticipation by considering not only the distance metric, but also the distribution and size of the candidate class.
4) Voting among different candidates is done by weighted voting system which considers the rank of candidates by various similarity measures in addition to the count and size of candidate sub-clusters.

The two phases of the proposed algorithm will be described separately as follow:

1) Clustering phase: In the first phase, EM clustering technique is applied for the members of each class. The number of sub-clusters produced in each class and the size of each are totally dependent on how distinguished are the data objects within each labelled class. The cluster-based classification method is considered to be very efficient in detecting hidden patterns especially when there are definite distinguished features into each class as shown in Figure 1. One practical example is classification of animals in a zoo. It is expected to have a class representing the bird category. Therefore, clustering inside the bird class will produce different species of birds as sub-clusters inside the class and that directly will enhance the overall cluster-based classification performance through detection of more specific categories inside each class.

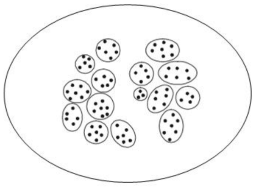


Figure 1: An illustration of the sub-clusters inside each class

Another example, if there are *N* definite classes of diseases, each disease is known by specific symptoms. However, symptoms related to one disease could be different from one patient to another. Applying a clustering technique, as an initial phase, for each class (disease) is very significant in specifying different sub-clusters that represent different sets of symptoms into each class. Consequently, each class is represented by a set of sub-clusters corresponding to different features inside this class.

2) Prediction Phase: The second phase of our novel technique is the classification of data objects based on the different sets of sub-clusters produced from the first phase and presenting different classes. In this step, the classification process is applied by using combination of three different similarity measures (distance, density and gravity). Applying different measures for the classification purpose instead of using only one is highly significant for producing more efficient classification results. As shown in Figure 2, we may consider point A as a member of the small class because it is closer. However, taking into consideration other similarity measures such as the density of the small and big classes and the size of both will lead us to the right classification decision. The three measures used in the proposed algorithm are explained in details in the following section.

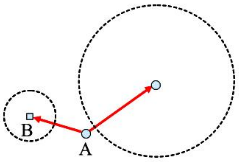


Figure 2: illustration of mis-classification using one similarity measure

B. Similarity measurements among sub-clusters

KB-CB-N uses a combination of three different similarity measures for the classification purpose. For each measure, a set of candidate sub-clusters have been chosen and ranked according to each measure. Then the ranked sub-clusters from each measure are merged together and re-ranked according to their standing among different measures. The vote is done among the finally ranked sub-clusters to choose the class with the majority votes of sub-clusters. The first similarity measure is the distance between each sub-cluster centre and data object; sub-cluster is highly ranked candidate if distance between the test data and sub-cluster centre is short. The density of each sub-cluster is among the measures used for *KB-CB-N* classification. The sub-cluster is chosen as a density high ranked candidate if the density gain increased in the whole class when the test data joined sub-cluster belongs to this class. The last measure is the gravity effect of each cluster. The test data might be attracted by sub-cluster gravity which is dependent on the sub-cluster size and therefore it will be highly ranked from the gravity point of view.

The three measures used in the *KB-CB-N* Classification are as follow:

1) Euclidean distance: The Euclidean distance between two data objects with n features P = (p_1, p_2, ...,p_n) and Q = (q_1, q_2,......,q_n) is defined as:

$$D(P,Q) = \sqrt{\sum_{i=1}^{n}(P_i - Q_i)^2}\ . \tag{2}$$

The distance is calculated between the test sample and each sub-cluster centre in the data object space. Then the sub-clusters are ranked as the one with the closest distance is highly ranked as described in Algorithm 1.

Algorithm 1 GetDistance(DataObj[], SubClusters)

```
for $i = 1$ to $n$ SubClusters do
    Calculate Euclidean distance $ED$ between $DataObj[]$ and $SubClusters_i$
    Assign $ED$ to $DistanceArray[i]$
end for
```

2) Sub-cluster density: The second measure used in *KB-CB-N* Classification technique is density. Each sub-cluster in the data space has its own density .The density of a sub-cluster is simply considered as the distribution of the data points into the cluster as the following formula:

$$ClusterDensity = \frac{SizeOfCluster}{AvgDist}\ . \tag{3}$$

$$AvgDist = \frac{\sum_{i=1}^{m}|(P_i - C)|}{m}\ . \tag{4}$$

Where (*m*) is the number of points in the cluster, (*p*) is the data point and (*C*) is the sub-cluster centre. The effect on the sub-cluster's density if the test sample joined is calculated for each sub-cluster.As shown in Algorithm 2, a data object may either cause a density gain or loss when joining a sub-cluster. Accordingly, the sub-cluster that is ranked as the highest rank sub-cluster is the one which attains the most density gain among all sub-clusters when the data sample joined it.

Algorithm 2 GetDensity(DataObj[], SubClusters)

```
for $i = 1$ to $n$ SubClusters do
    Calculate $CurrDens$ of $SubClusters_i$
    Add $DataObj[]$ to $SubClusters_i$
    Calculate $ExpDens$ of $SubClusters_i$
    Calculate $DensityGain$ = $ExpDens$ -$CurrDens$
    Assign $DensityGain$ to $DensityArray[i]$
    Remove $DataObj[]$ from $SubClusters_i$
end for
```

3) Sub-cluster gravity: The last perspective is to examine the sub-clusters according to their gravitational force. There exists a natural attraction force between any two objects in the universe and this force is called gravitation force. According to Newton universal law of gravity, the strength of gravitation between two objects is in direct ratio to the product of the masses of the two objects, but in inverse ratio to the square of distance between them. The law can be described as follows:

$$F_g = G\frac{m_1 m_2}{r^2} \ . \tag{5}$$

Where *F* is the gravitation between two objects; *G* is the constant of universal gravitation; m_1 is the mass of object 1; m_2 is the mass of object 2; *r* the distance between the two objects. Each sub-cluster generates its own gravitational force created from its weight. The bigger the weight of the cluster the stronger the gravitational force produced from it. And therefore, the probability that the sub-cluster will attract the data object could be increased. If the data object location is within the gravitational field of a sub-cluster, then the data point will be attracted by the sub-cluster's gravitational force. The gravitational force is calculated between each sub-cluster and the test data object then stored in descending order in the ranked gravitational array as described in Algorithm 3.

Algorithm 3 GetGravity(DataObj[], SubClusters)

```
for $i = 1$ to $n$ SubClusters do
    Calculate Distance $Dis$ between $DataObj[]$ and $SubClusters_i$ centre
    Calculate GravitationalForce $F_g$ of $SubClusters_i$
    Add $F_g$ to $GravityArr$
end for
```

After applying the three measures, the outcome will be different rankings of all sub-clusters from each point of view. The algorithm then merges the rankings of the three approaches and combines them in one set of ranked sub-clusters with respect to the sub-cluster's standing in each measure. *K*-best sub-clusters are chosen from the combined ranking set and the test object is being assigned to the most significant weighted class among its *k* best sub-clusters.

The concept of weighted ranking for each class is applied after combining the three measures. That means, among the *k*-best sub-clusters distributed in all classes, each class has a specific number of sub-clusters assigned to it and each sub-cluster has its rank. The class weighted rank is considered as the the average rank of sub-clusters chosen among the *k*-candidates divided by their count. The class with the lowest weight is the class chosen by the voting procedure for the classification result. This weighted approach showed more accurate performance as it gives high weight for the class contains the largest number of best sub-cluster and the highest rank among all measures.

C. KB-CB-N Classification algorithm

In the clustering phase , as shown in Algorithm 4, we applied EM clustering technique for each labelled class as the following pseudo code:

Algorithm 4 Clustering Phase (Set of Classes $ClassSet$)

```
for $i = 1$ to $n$ Classes do
    Apply EM Clustering technique to $ClassSet_i$
    Store sub-clusters for each $ClassSet_i$ for further calculation
end for
```

The output of the clustering step is then used as an input for the next prediction phase as per Algorithm 5.

Algorithm 5 Predicting Phase (SubClusters)

```
for $i = 1$ to $n$ TestDataObject $DataObj[]$ do
    for $j = 1$ to $m$ SubClusters do
        getDistance ($DataObj[i]$ , $SubClusters[j]$)
        getDensity ($DataObj[i]$, $SubClusters[j]$)
        getGravity($DataObj[i]$, $SubClusters[j]$)
    end for
    Sort the Distance Array $DisArr[]()$ Ascending
    Sort the Density Array $DensArr[]$ Descending
    Sort the Gravity Array$GravArr[]$ Descending
    for $K = 1$ to $m$ SubClusters do
        Calculate $GlobalRank$= $DisRank$ + $DensRank$ + $GravRank$
        Add $GlobalRank$ to $GlobalRankArr$
    end for
    Sort $GlobalRankArr$ Ascending
    Get $K$ best SubClusters in $GlobalRankArr$
    Vote among $K$ subClusters to choose the best class $BC$
    Assign $DataObj[i]$ to $BC$
end for
```

The Voting procedure among classes, represented in Algorithm 6, is based on two parameters. First, the number of candidate sub-clusters in the class. Second, the average rank of candidate sub-clusters in the class. This will give different ranking weights for class based on the two parameters. The average rank for m sub-clusters is computed as:

$$AvgRank = \frac{\sum_{i=1}^{m} globalrank}{m} . \quad (6)$$

Algorithm 6 Vote(K Candidate Sub-Clusters)

```
for $i = 1$ to $n$ Classes  do
    Count $J$ SubClusters among $k$ candidates assigned to $class_i$
    Calculate averageRank $avgRank_j$ Of $J$ subclusters
    Calculate weightedRank $WRank$= $avgeRank_j$/ $J$
    if $WRank \leq tempWeight$ then
        BestCandidate $BCand$= $i$
        $tempWeight$= $WRank$
    end if
end for
Return BestCandidate $BCand$
```

V. Experimental Results

We have run our experiments under the framework of *Weka* [21] to study the efficiency of the above proposed method. We have used *Weka* implemented techniques for clustering phase and comparison purposes. The *Weka* clustering techniques used for the clustering step within each class are: *EM* ,*K-means*, *DBScan* and *Density-based clustering*. The objectives of our experimental study are stated as follows:

- Evaluation of the different state-of-the-art clustering techniques to be used in the first step of our technique.
- Evaluation of the performance of our *KB-CB-N* when compared with high performance classification techniques using real datasets.
- Assessment of the sensitivity of the algorithm to the value of K using real datasets.

A. Datasets

We have used different datasets to assess our proposed technique from *UCI* database [22] in addition to the colon cancer [23] and grass grubs datasets [24]. These datasets represent a wide range of domains and data characteristics. A description of each dataset is given in the following:

- *Waveform* dataset which represents 3 classes of waves with 21 attributes. All of the 21 attributes include noise. The dataset has 5000 instances.
- *Iris* dataset which consists of 150 instances, 4 attributes, and 3 classes, each class being composed of 50 instances where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other.
- *Balance Scale* data set is composed of 625 instances, 4 attributes and 3 classes. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced.
- *Ionosphere* consists of 351 instances, 34 continuous attributes represented in 2 classes either ”Good” or ”Bad”. This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas. The targets were free electrons in the ionosphere. ”Good” radar returns are those showing evidence of some type of structure in the ionosphere. ”Bad” returns are those that do not; their signals pass through the ionosphere.
- *Breast Cancer* dataset includes 268 instances of two classes. The instances are described by 9 attributes, some of which are linear and some are nominal.
- *Diabetes* dataset consists of 768 instances, each with 8 attributes over 2 classes representing either tested positive or negative for diabetes.
- *Grass Grubs Agriculture* dataset contains measured grass grub population density from a variety of locations and the level of pasture damage present. It composed of 155 instances, 8 attributes and 4 classes.
- *Colon Cancer* dataset contains expression levels of 2000 genes taken in 62 different samples. For each sample, it is indicated whether it came from a tumor biopsy or not. It has two classes where one is cancer and the other is normal. This dataset is used in many different research papers on gene expression data.

Table I shows a brief summary of the characteristics of the above described datasets.

Table I: Datasets summarized characteristics

Dataset	Number of instances	Number of attributes	Number of classes
Waveform	5000	21	3
Iris	150	4	3
Balance Scale	625	5	3
Ionosphere	351	34	2
Breast Cancer	268	9	2
Diabetes	768	8	2
Grass Grubs	155	8	4
Colon Cancer	62	2000	2

B. Results analysis

Our new proposed *KB-CB-N* algorithm clustering phase applies a clustering technique within each class. Figure 3 and Figure 4 illustrate the algorithm performance with different clustering techniques on *Balance Scale* and *Ionosphere* datasets respectively. Clustering techniques applied are *EM* using both unspecified and previously specified number of clusters, *K-means* with different values of *K*, *DBScan* and *Density-based clustering*. All used techniques have been implemented in *Weka 3.5.8*.

Results show that the highest performance is attained when applying *EM* clustering algorithm with unspecified number of clusters. That makes the number of clusters generated may vary from one class to another within the same dataset. This is a reasonable assumption as the number of intraclass groups are likely to vary from one class to the other, and from one dataset to the other. Hence we have chosen to use *EM* clustering in the first step of our method.

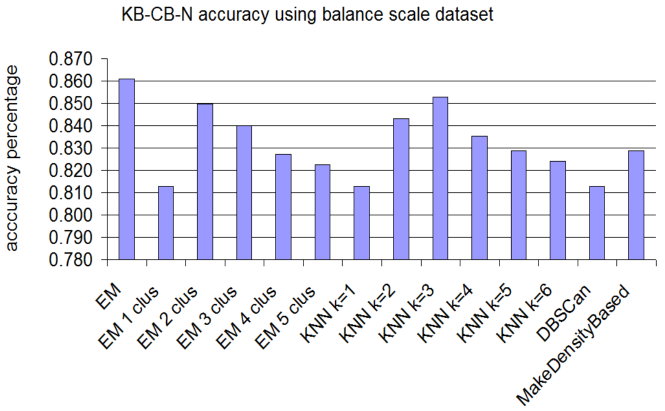


Figure 3: KB-CB-N performance with different clustering algorithms on Balance Scale dataset

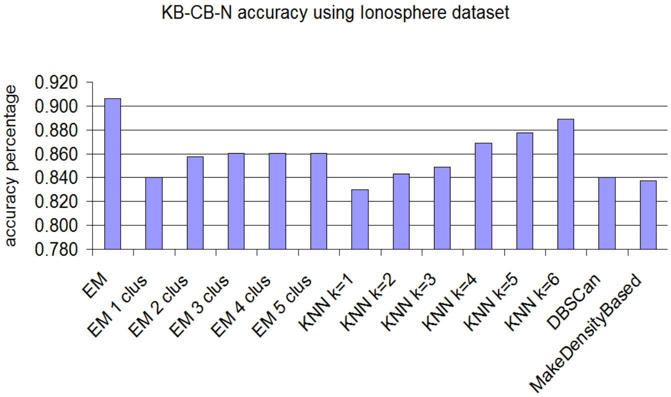


Figure 4: KB-CB-N performance with different clustering algorithms on Ionosphere dataset

As mentioned earlier, traditional *K-NN* is sensitive to noise. Thus, changing the value of K may have a great negative impact on the performance. A comparison of the accuracy level achieved with different values of K have been conducted. Figure 5 shows how the performance of *KB-CB-N* changes with different values of K. The results show that the sensitivity to the value of K is low. This represents an important achievement of our technique. It is the outcome of using a cluster instead of individual instances in classifying unlabelled instances.

We have adopted two steps for the proposed algorithm. The first one called K- cluster based nearest neighbour *K-CB-NN* which explains the effect of cluster based classification algorithm with only the Euclidean distance similarity measure. The second is our novel cluster based classification method using the combination of the three different similarity measures *KB-CB-N*. The aim of this is to distinguish the effect of each individual contribution on the experimental results.

We have conducted extensive empirical comparison for *KB-CB-N* (both steps) and other classifiers including *Best First Decision tree*, *Fast decision tree learner*, *Locally-weighted learning LWL* and *K- Nearest Neighbor* algorithms in terms of classification accuracy. The classification accuracy of each classifier on each data set was obtained via 10-fold cross validation, with the various algorithms applied on the same training sets and evaluated on the same test sets. The clustering algorithm used in the clustering phase is *EM* with unspecified number of clusters. K is set to 3 in all algorithms with a K parameter. The results are summarized in Table II.

Table II: The accuracy rate (%) comparison of BFTree, REPTree, LWL, KNN with the two steps of the proposed method

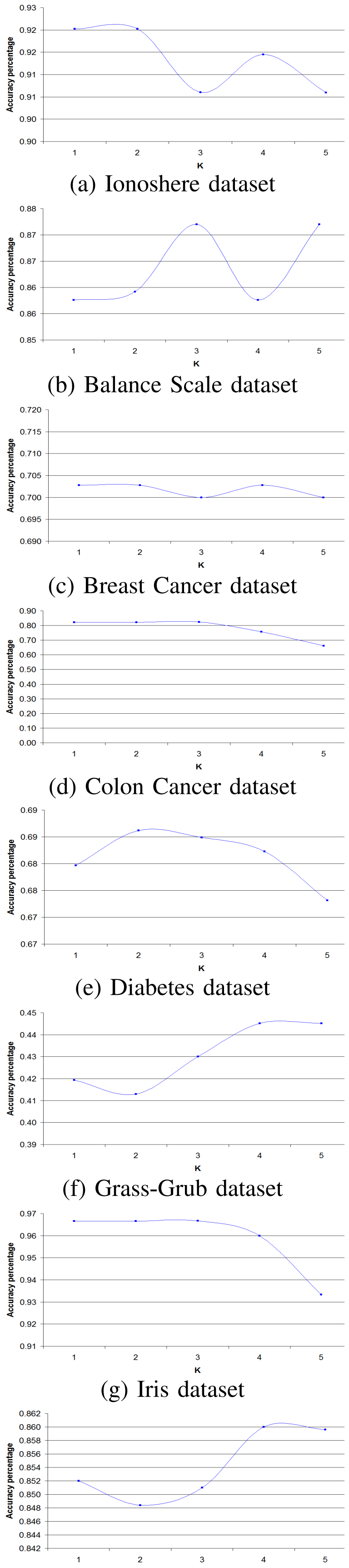
Dataset	*KB-CB-N*	*K-CB-NN*	*KNN*	*BFTree*	*REPTree*	*LWL*
Waeform	**0.85**	**0.85**	0.79	0.76	0.77	0.57
Balance-Scale	**0.87**	0.85	0.82	0.79	0.77	0.55
Ionosphere	**0.91**	0.74	0.84	0.90	0.90	0.82
Colon Cancer	**0.82**	0.52	0.76	0.77	0.68	0.74
Diabetes	0.68	0.64	0.66	0.74	**0.75**	0.71
Iris	**0.97**	0.92	**0.97**	0.95	0.94	0.93
Grass Grubs	0.43	0.36	0.41	**0.49**	0.42	0.32
Breast Cancer	0.70	0.70	0.68	0.68	0.71	**0.72**

As shown in Table II, our proposed *KB-CB-N* algorithm shows high accuracy level among the majority of the datasets. The superior performance of the novel algorithm is attained especially when the dataset has noise like *Waveform* and/or the dataset has a set of distinguished features inside each labeled class like *Colon Cancer* dataset. Comparing KB-CB-N with other algorithms, the proposed algorithm attains 5% - 7% higher classification accuracy than other techniques for most of the examined datasets. This is a significant increase in the classification performance. The *K-CB-NN* which uses the Euclidean as the only similarity measure for cluster based classification attains high accuracy level on some datasets like *Balance Scale* and *Waveform*, however using combination of different similarity measures produces better and consistent accuracy among various datasets.

VI. Conclusion

In this paper, we have proposed, developed and evaluated our novel cluster-based K best neighbor classification method based on three different similarity measures, namely, distance, density and gravity. Using unsupervised learning method for labeled classes before applying supervised learning improves detection of hidden features inside each class. Then applying the aforementioned similarity measures for classification purpose shows superiority over the use of individual ones, and enhances the classification accuracy. Empirical results show that *KB-CB-N* achieved better classification accuracy than several efficient classification methods for a wide range of different real datasets.

Having obtained a high classification performance, our plan for future work includes using the techniques in a streaming settings. The use of a traditional *K-NN* in such a setting is infeasible. However, our *KB-CB-N* uses the clustering results to perform classification. This makes it a potential streaming technique.



(a) Ionoshere dataset

(b) Balance Scale dataset

(c) Breast Cancer dataset

(d) Colon Cancer dataset

(e) Diabetes dataset

(f) Grass-Grub dataset

(g) Iris dataset

(h) Waveform dataset

Figure 5: KB-CB-N performance with different values of K

References

[1] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Wiley-Interscience, 1998.

[2] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA: MIT Press, 2001.

[3] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2005, vol. 27, no. 2. [Online]. Available: http://scholar.google.de/scholar.bib?q=info:roqIsr0iT4UJ:scholar.google.com/&output=citation&hl=de&as_sdt=2000&ct=citation&cd=0

[4] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern Recognition*, vol. 10, no. 2, pp. 105 – 112, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/B6V14-48MPGX5-2V/2/2fc3a1a811e0741a9b89d03ea4cd56b1

[5] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification." *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, 1967.

[6] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008. [Online]. Available: http://scholar.google.com.au/scholar.bib?q=info:CLKitXYAt4sJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&ct=citation&cd=0

[7] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Survey*, vol. 31, no. 3, pp. 264–323, 1999. [Online]. Available: http://explorer.csse.uwa.edu.au/reference/browse_paper.php?pid=233281708

[8] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm." *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[10] S. Borman, "The expectation maximization algorithm: A short tutorial. unpublished paper available at http://www.seanborman.com/publications," Tech. Rep., 2004.

[11] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 5, no. 1, pp. 71–100, 1986.

[12] H. Lu, R. Setiono, and H. Liu, "Effective data mining using neural networks." *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 957–961, 1996. [Online]. Available: http://dblp.uni-trier.de/db/journals/tkde/tkde8.html#LuSL96

[13] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Computational Learing Theory*, 1992, pp. 144–152. [Online]. Available: http://www.svms.org/training/BOGV92.pdf

[14] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving k-nearest-neighbor for classification." in *FSKD (1)*, J. Lei, Ed. IEEE Computer Society, 2007, pp. 679–683. [Online]. Available: http://dblp.uni-trier.de/db/conf/fskd/fskd2007-1.html#JiangCWJ07

[15] E.-H. Han, G. Karypis, and V. Kumar, "Text categorization using weight adjusted k-nearest neighbor classification." in *PAKDD*, ser. Lecture Notes in Computer Science, D. W.-L. Cheung, G. J. Williams, and Q. Li, Eds., vol. 2035. Springer, 2001, pp. 53–65. [Online]. Available: http://dblp.uni-trier.de/db/conf/pakdd/pakdd2001.html#HanKK01

[16] H. Wang, D. A. Bell, and I. Dntsch, "A density based approach to classification." in *SAC*. ACM, 2003, pp. 470–474. [Online]. Available: http://dblp.uni-trier.de/db/conf/sac/sac2003.html#WangBD03

[17] L. Peng, B. Yang, Y. Chen, and A. Abraham, "Data gravitation based classification." *Inf. Sci.*, vol. 179, no. 6, pp. 809–819, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/isci/isci179.html#PengYCA09

[18] M. R. Peterson, T. E. Doom, and M. L. Raymer, "Ga-facilitated knn classifier optimization with varying similarity measures." in *Congress on Evolutionary Computation*. IEEE, 2005, pp. 2514–2521. [Online]. Available: http://dblp.uni-trier.de/db/conf/cec/cec2005.html#PetersonDR05

[19] B. Zhang and S. N. Srihari, "Fast k-nearest neighbor classification using cluster-based trees." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 525–528, 2004. [Online]. Available: http://dblp.uni-trier.de/db/journals/pami/pami26.html#ZhangS04

[20] Z.S.Ammar and M. .Gaber, "Ddg-clustering : A novel technique for highly accurate results," in *IADIS European Conference on Data Mining*, 2009.

[21] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[22] C. Blake, E. Keogh, and C. Merz, "UCI repository of machine learning databases," 1998, (http://www.ics.uci.edu/mlearn/MLRepository.html).

[23] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[24] R.J.Townsend, T.A.Jackson, J.F.Pearson, and R.A.French, "Grass grub population fluctuations and damage in canterbury," in *6th Australian Grasslands Invert. Ecol. Conference*, 1993.