Performance Analysis of *k*-NN on High Dimensional Datasets

Pradeep Mewada*
Research Scholar
SATI, Vidisha (M.P.) India

Jagdish Patil*
Research Scholar
SATI, Vidisha (M.P.) India

ABSTRACT
Research on classifying high dimensional datasets is an open direction in the pattern recognition yet. High dimensional feature spaces cause scalability problems for machine learning algorithms because the complexity of a high dimensional space increases exponentially with the number of features. Recently a number of ensemble techniques using different classifiers have proposed for classifying the high dimensional datasets. The task of these techniques is to detect and exploit relevant patterns in data for classification. The *k*-nearest neighbor (*k*-NN) algorithm is amongst the simplest of all machine learning algorithms. This paper discusses various ensemble *k*-NN techniques on high dimensional datasets. The techniques mainly include: Random Subspace Classifier (RSM), Divide & Conquer Classification and Optimization using GA (DCC-GA), Random Subsample ensemble (RSE), Improving Fusion of dimensionality reduction (IF-DR). All these approaches generates relevant subset of features from original set and the results is obtain from combined decision of ensemble classifiers. This paper presents an effective study of improvements on ensemble *k*-NN for the classification of high dimensional datasets. The experimental result shows that these approaches improve the classification accuracy of the *k*-NN classifier**.**

General Terms
Machine learning, Pattern Recognition

Keywords
k-Nearest Neighbor, Ensemble Classifiers, High Dimensional Feature Space.

1. INTRODUCTION
The goal of building systems that can adapt to their environments and learn from their experience has attracted researchers from many fields, including computer science, engineering, mathematics, physics, neuroscience, and cognitive science. Out of this research has come a wide variety of learning techniques that are transforming many industrial and scientific fields. With the recent advances in hardware and software, a variety of practical applications of the Machine Learning research is emerging. For pattern classification, multiple classifiers are combined to improve the classification accuracy. The combination of multiple classifiers has been viewed as a new direction for the development of highly reliable pattern recognition systems. Classifiers are combined to improve classification decision, but there are no general rules as to how a number of classifiers should be combined [3]. There are several reasons which describe the necessity of combining multiple classifiers. For any recognition problem, the retrieval time increases directly with the number of features and high dimensional feature spaces also cause scalability problems**.**

Noisy or irrelevant features can have the same influence on retrieval as predictive features so they will impact negatively on accuracy [4]. Recently, machine learning researchers begin paying attention on feature subset selection for multiple classifiers to deal with the problem of high dimensional feature spaces. The primary objective of using multiple features in data classification is to improve the classification performance and enhance the generalization ability of classifier by using different features extracted from original data. These features are also represented in much diversified forms and it is rather hard to lump them together for single classifier to make a decision [5]. In this paper we discuss four different ensemble techniques using *k*-NN classifier for classifying the dataset consists of multiple features. All these techniques randomly generate some feature subsets and evaluating the subsets for obtaining best feature subset using ensemble *k*-NN classifiers. We study the performance of all these methods in the form of average classification accuracy and error rate.

2. MACHINE LEARNING
The field of machine learning is driven by the idea that computer algorithms and systems can improve their own performance with time [2]. Classification is one of the important techniques used in the field of Machine Learning. Data classification is the categorization of data for its most effective and efficient use. Data describe the characteristics of a living species, depict the properties of a natural phenomenon, summarize the results of a scientific experiment, and record the dynamics of a running machinery system.

Classification plays an important and indispensable role in the long history of human development [10]. In order to learn a new object or understand a new phenomenon, people always try to identify descriptive features and further compare these features with those of known objects or phenomena, based on their similarity or dissimilarity, generalized as proximity, according to some certain standards or rules. Basically, classification systems are either supervised or unsupervised, depending on whether they assign new data objects to one of a finite number of discrete supervised classes or unsupervised categories, respectively. In supervised classification, we have a training set of data & for each record of this set, the respective class to which it belongs is also known. Using the training set, the classification process attempts to generate the descriptions of the classes & these descriptions help to classify the unknown records.

3. *k*- NEAREST NEIGHBOR

The *k*-nearest neighbor (*k*-NN) is a supervised learning algorithm where the result of a new instance query is classified based on the majority of *k*-nearest neighbors. *k*-NN assumes that each instance relates to a point in an *n*-dimensional space and can be described as a sequence of attributes, i.e. $a_1(x)$, $a_2(x)$, ... $a_n(x)$, where *n* is the number of attributes [2]. The distance between instance x_i and instance x_j is calculated by formula (1).

$$D(x_i, x_j)^n = [\sum_{k=1}^{n}(x_{ik} - x_{jk})^2]^{1/2}$$

To classify a new pattern x, the *k*-NN classifiers find *k* nearest patterns in the training set *D* and uses the *k* pattern to determine the class of a test object $x = (x', y')$. The algorithm computes the distance between test object (z) and the entire training object *(*x, *y)* which is belongs to *D* to determine its nearest-neighbor list, *Dz*. (x is the data of training object, while *y* is its class. Likewise, x' is the data of the test object and *y*' is its class). Once the nearest-neighbor list is obtained, the test object is classified based on the majority class of its nearest neighbors [5].

Training algorithm:

Input: - D, the set of *k* training objects and test object $z= (x', y')$.

Classification algorithm:

Process: - 1. Compute $d(x', x)$, the distance between z and every object(x, y) belongs to D.
2. Select Dz (subset of D, the set of *k* closest training objects to z).

Output: - $y'= \text{argmax} \sum_{(xi,yi)} I(v=y_i)$ (3)

There are several key issues that affect the performance of *k*-NN [11]. One is the choice of *k*. If *k* is too small, then the result can be sensitive to noise points. On the other hand, if *k* is too large, then the neighborhood may include too many points from other classes.

Another issue is the approach to combining the class labels. The simplest method is to take a majority vote, but this can be a problem if the nearest neighbors vary widely in their distance and the closer neighbors more reliably indicate the class of the object. A more sophisticated approach, which is usually much less sensitive to the choice of *k*, weights each object's vote by its distance, where the weight factor is often taken to be the reciprocal of the squared distance:

$$w_i = 1/d(x', x_i)^2 \quad (4)$$

This amounts to replacing the last step of the *k*-NN algorithm with the following:

Distance-Weighted Voting:

$$y'= \text{argmax} \sum_{(xi,yi)} w_i * I(v=y_i) \quad (5)$$

The choice of the distance measure is another important consideration. Although various measures can be used to compute the distance between two points [1], the most desirable distance measure is one for which a smaller distance between two objects implies a greater likelihood of having the same class. Thus, for example, if *k*-NN is being applied to classify documents, then it may be better to use the cosine measure rather than Euclidean distance. Some distance measure can also be affected by the high dimensionality of the data. A number of techniques have been developed for efficient computation of *k*- nearest neighbor distance that make use of the structure in the data to avoid having to compute distance to all the objects in the training set [11].

4. VARIOUS ENSEMBLE TECHNIQUES

The techniques in machine learning mainly include Random Subspace Classifier (RSM), Divide & Conquer Classification and Optimization using GA (DCC-GA), Random Subsample ensemble (RSE), Improving Fusion of dimensionality reduction (IF-DR) is discussed below:

4.1 Evolved Feature Weighting For Random Subspace Classifier (RSC*)*

In several classification problems, in particular, when data are high dimensional and the number of training samples is small compared to the data dimensionality, it may be difficult to construct a good classification rule. In fact, a classifier constructed on small training sets has usually a poor performance [14]. In the literature, it is shown that a good approach to improve a weak classifier is to construct a strong multi- classifier that combines many weak classifiers with a powerful decision rule [6]. The most studied approaches for creating a multi- classifier are bagging, boosting, and random subspace. The problem addressed in this letter concerns the multi-classifier generation by a random subspace method (RSM). In the RSM, the classifiers are constructed in random subspaces of the data feature space.

Loris Nanni and Alessandra Lumini [6] proposed an evolved feature weighting approach in which, the features are multiplied by a weight factor in each subspace for minimizing the error rate in the training set. The method was based on particle swarm optimization (PSO), which was used for finding a set of weights for each feature in each subspace.

The basic idea for applying PSO to random subspace is the following: for each feature in each subspace, we initially assign a random weight between 0 and 1, in this way, the dimension of each particle of the PSO problem is (number of features) * (number of subspaces). Then, the weights of these features are adjusted using the PSO1 so that the classification error rate of the ensemble on the training set is minimized. The classifiers trained using these weighted random subspaces are combined by majority vote rule.

The experiments were conducted on various data sets from the University of California at Irvine (UCI) Repository [9]. All the data sets were normalized between 0 and 1. To minimize the possible misleading caused by the training data, the results on each data set have been averaged over ten experiments. The performance is compared in the form of error rate. The average classification error rate of this method was 12.9%. The results were compared with the standard *k*-NN method on selected data set. The result indicates that this method reduces the error rate of the *k*-NN classifier from 16.67% to 12.9%.

4.2 Divide & Conquer Classification Using Genetic Algorithm (DCC-GA)

Improving the recognition performance is one of the most challenging tasks in pattern classification. Commonly, to

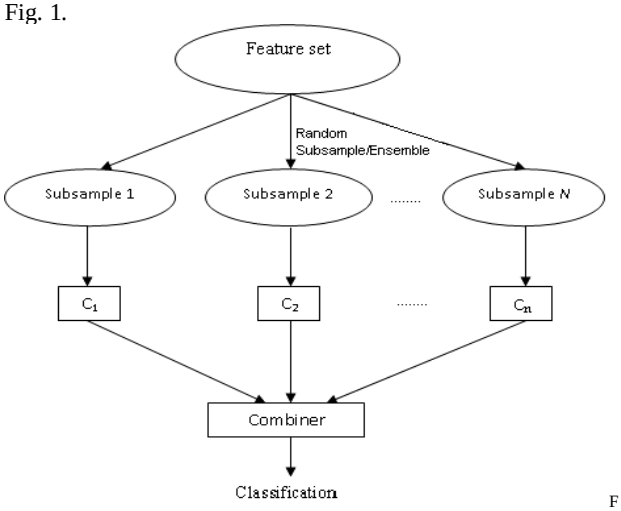
improve recognition ratio we use ensemble classifiers. It has been shown that the performance of ensemble classifier systems is usually better than single classifier system. Hamid Parvin, Hosein Alizadeh, Mohsen Moshki, Behrouz Minaei-Bidgoli and Naser Mozayani [13] proposed a new method for multiclass classification. The main idea of this method is to divide the classification problem into smaller problems. Suppose that a metaclass is a subset of classes. Also, suppose that all classes are in a one large metaclass. So, in each level, there is a classifier to divide a metaclass into two smaller metaclasses. Indeed, this method is like divide-and-conquer method. For this method, the dataset is partitioned into three sets: training, evaluation and test sets. This approach contains several steps. In the first step, by using a base classifier, we do a primary classification and extract the confusion matrix from the evaluation data set. In this step a multiclass classifier is trained on training dataset. Then, the confusion matrix is made by using the results of this classifier on evaluation data. This matrix contained important information about functionality of classifiers. Also, close and error prone classes are recognized using this matrix. In fact, the confusion matrix determines error distribution on different classes. In the second step, they were used a classifier ensemble more or less like decision tree, than trained one classifier correspond to each node that divides the data into two metaclasses. Each of metaclasses can contain several classes. This categorization is done based on error rate of confusion matrix.

In there study [13], a two hidden layer perceptron is used as a base classifier. Also, the *k*-NN with *k*=3 is another base classifier. The confusion matrix is obtained from these classifiers. After that, GA is used to determine the optimal tree. Gaussian and Scattered operators were used respectively for mutation and crossover. The Scattered crossover function creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. The Gaussian mutation on each entry of the parent vector follows a Gaussian distribution. For GA optimization, 200 individuals were used in the population, running the GA over 500 generations. Fitness function for GA is the error ratio obtained from confusion matrix. The experiments of this method were conducted using Farsi digits from OCR databases and the results were compared in the form of prediction accuracy. The average prediction accuracy of this method was 96.86%.

4.3 Classification By Random Subsample Ensemble (RSE)

Gursel Serpen and Santhosh Pathical [12] was proposed random subsample ensemble for classifying high dimensional datasets. Classifying high dimensional datasets is a challenging task due to several reasons. High dimensional feature spaces cause scalability problems for machine learning algorithms because the complexity of a high dimensional space increases exponentially with the number of features. This effect is called as the curse of dimensionality. The concept of the Random subsample Method (RSE) is that, the divide-and-conquer methodology can be used to accomplish this goal while dealing with the adverse effects of the curse of dimensionality within a machine learner ensemble context. The aim of the ensemble is to break down a complex high dimensional problem into several lower dimensional sub-problems, and thus conquer the computational complexity aspects of the original problem. High dimensional feature space is projected onto a set of lower dimensions by selecting random feature subsamples from the original set [12].

The architecture of random subsample ensemble is depicted in Fig. 1.



F

igure1: Conceptual architecture of Random Subsample Ensemble

As an example, consider the original high-dimensional feature space to be represented by F={x1, x2… xk}. The feature space F is randomly projected onto N d-dimensional subspaces, fi = {x1, x2 … xd} for, i=1 to N with d<<k. The values for d (cardinality of subsample set) and N (number of subsamples) are likely to be problem-dependent and require empirical search and/or theoretical analysis for optimality. In one scheme, if the d-dimensional feature sets are generated by randomly selecting d features from the original feature set without replacement, then the following holds:

$$\bigcup_{i=1}^{N} f_i \subseteq \{x_1, x_2 \ldots x_k\} \qquad (6)$$

where any two subsets fi and fj for i, j=1, 2,…, N, might have some of their elements (features) the same. The random numbers used for the sampling may be generated using a uniform probability distribution. Within the context of random sampling without replacement, a larger value for the product of d and N can help in the inclusion of a higher number, if not all, of original features into the ensemble. There are also other approaches to sub sampling the original high-dimensional feature space, where one such technique is the mutually exclusive partitioning. It results in each every feature to be included by exactly one subsample. Each lower dimensional subspace of the feature space is used to train a base learner of the ensemble, which will be called as random subsample ensemble (RSE). Once the base learners are trained, their predictions are combined to get the final ensemble prediction.

RSE makes the learning task scalable by using lower dimensional projections of the original high dimensional feature space. Smaller scale projections of the original high-dimensional feature set also help in faster induction of the individual classifiers. In cases where the learning task is

difficult or may not be accomplished using a standalone machine learning algorithm, RSE can be employed to divide the original task into smaller sub-tasks, each of which can be learned with much less effort, and thus overcoming the computational intractability.

RSE has two parameters, values of which need to be determined through empirical and/or theoretical means. These two parameters are: (1) cardinality of the feature subset and (2) the number of such subsets to be generated. In the absence of any theoretical insight, empirical evaluation for different values of these two parameters needs to be carried out to identify good value ranges based on a set of performance measures like the cross-validation accuracy and the computational cost.

A simulation study was performed to assess the feasibility of RSE on datasets with up to 20K features from the UCI machine learning repository. The average classification accuracy of RSE method on various multiple feature data sets is 81.12%, and the classification accuracy of k-NN on these selected dataset is 79.7%. Results provide strong evidence that RSE is well positioned to address high dimensional datasets with practically none or little loss in prediction accuracy or any substantial increase in the computational cost.

4.4 Improving Fusion Of Dimensionality Reduction Methods (IF-DR)

Classification accuracy of *k*-NN can be improved using dimensionality reduction and further improved by using different methods of feature and classifier fusion. It has been demonstrated that the fusion of dimensionality reduction methods, either by fusing classifiers obtained from each set of reduced features, or by fusing all reduced features are better than using any single dimensionality reduction method. However, none of the fusion methods consistently outperform the use of a single dimensionality reduction method. Sampath Deegalla and Henrik Bostrom [7] proposed a new way of fusing features and classifiers, which is based on searching for the optimal number of dimensions for each considered dimensionality reduction method.

An empirical evaluation on microarray classification is presented by him, comparing classifier and feature fusion with and without the proposed method, in conjunction with three dimensionality reduction methods; Principal Component Analysis (PCA), Partial Least Squares (PLS) [8] and Information Gain (IG).

PCA is a classical dimensionality reduction method that has been applied in many different contexts, including face recognition, image compression, cancer classification and applications related to high-dimensional datasets. This method is well known for allowing the original dimensionality to be reduced a much smaller, uncorrelated feature set with minimum information loss [15]. Partial Least Squares (PLS) was originally developed within the social sciences and has later been used extensively in chemo metrics as a regression method [8]. It seeks for a linear combination of features whose correlation with the output variable is maximum.

In PLS regression, the task is to build a linear model, $Y = BX + E$, where B is the matrix of regression coefficients and E is the matrix of error coefficients. In PLS, this is done via the factor score matrix $Y = WX$ with an appropriate weight matrix W. Then it considers the linear model, $Y = QY + E$, where Q is the matrix of regression coefficients for Y. First, raw features are transformed into a lower number of dimensions using all reduction methods.

PCA and PLS transformations are applied to the training set and the generated weight matrix is used to transform the test set. In IG, features based on the information content are ranked in decreasing manner in the training set and the same rankings are used when classifying the test set. For *k*-NN, k =1 is considered, i.e., a single nearest neighbor is chosen. To find the optimal number of features, cross-validation is performed using the training set with the nearest neighbor classifier. Then the optimal number of features for the training set is selected for the final classification. *k*-NN classification is performed on the reduced space generated from the training set with optimal number of features by PCA, PLS and IG.

The performance of this method was compared using eight different microarray datasets. The results were comparing in the form of average classification accuracy. The FF2 perform average classification accuracy as 83.1% on the selected data set, while CF2 perform average classification accuracy as 84.1%. It was observed that the novel methods perform particularly well when all the dimensionality reduction methods outperform using the original feature set.

5. RESULT AND DISCUSSION

In this paper we discuss four different ensemble techniques using *k*-NN classifier for classifying the data set consists of multiple features. The Table below shows the performance of various ensemble techniques which was performed on different datasets by different authors and compare with the standard *k*-NN.

Author	Dataset	Algorithm/ Technique	Accuracy In %
Loris, Alessadra [6]	UCI-Data	*k*-NN	83.3
		RSM	**87.1**
Hamid, Hosein [13]	Farsi-Digits	*k*-NN	96.6
		DC-GA	**96.8**
Gursel, Santhosh [3]	Multiple-Feature	*k*-NN	79.7
		RSE	**81.1**
Sampath, Henrik [7]	Micro-Array	*k*-NN	79.8
		IF-DR	**83.6**

k-NN- *k*-nearest neighbor, RSM- Random subspace method, DC-GA- Divide and Conquer classification using genetic algorithm, RSE- Random subsample ensemble, IF-DR- Improving fusion of dimension reduction

6. CONCLUSION

k-NN is stable to the change of the training datasets while sensitive to the variation of the feature sets. Since *k*-NN is sensitive to the change of feature sets, the combination of *k*-NN based on different feature subsets may lead to a better performance. This paper presents a complete review of *k*-NN method and various ensemble methods for the classification of high dimensional datasets. RSM [6] uses random subspace &

feature weighting where weights are refined using PSO algorithm. This method improves the accuracy of *k*-NN classifier from 83.3% to 87.1%. The divide and Conquer method (DC-GA) [13] based on divide the classification problem into smaller problems. The average classification result of this method on Farsi digit dataset is 96.86%. Gursel, Pathical [12] was proposed random subsample ensemble (RSE) for classifying high dimensional data set. The RSE break down a complex high dimensional problem into several lower dimensional sub-problems, and thus conquer the computational complexity aspects of the original problem. The average classification accuracy of RSE method on various multiple feature datasets is 81.12%, and the classification accuracy of *k*-NN on these selected dataset is 79.7%. The experimental results show that the RSE method outperforms the standard *k*-NN. Sampath and Henrik [7] proposed a new way of fusing features and classifiers, which is based on searching for the optimal number of dimensions for each considered dimensionality reduction method. This method performs classification in conjunction with three dimensionality reduction methods; Principal Component Analysis (PCA), Partial Least Squares (PLS) and Information Gain (IG). The result shows that, this method not only improves the average classification accuracy of *k*-NN from 79.9% to 83.6%, also it reduces the error rate. Our Future research will be concerned with applying the meta-heuristic search technique for selection of feature subsets to deal with the problems of classifying high dimensional datasets.

7 REFERENCES

[1]. D. R. Wilson and T. R. Martinez "Improved heterogeneous distance functions" Journal of Artificial Intelligence Research, 6(1):1–34, 1997.

[2]. Tom M. Mitchell "Machine Learning" Mcgraw-Hill Science/ Engineering/ Math March, 1997.

[3]. Stephen D. Bay "Combining nearest neighbor classifiers through multiple feature subsets" Proceeding 17th Intl. Conf. on Machine Learning-1998.

[4]. M. L. Raymer et al "Dimensionality Reduction using Genetic algorithms" IEEE Transactions on Evolutionary Computation, 4(2), 164– 171, 2000.

[5]. Pradeep Mewada, Shailendra K. Shrivastava "Review of combining multiple *k*-nearest neighbor classifiers" International Journal of Computational Intelligence Research & Applications(IJCIRA), July-December 2010, pp. 187-191.

[6]. Loris Nanni and Alessandra Lumini "Evolved feature weighting for random subspace classifier" IEEE - transactions on neural networks, vol.19, no.2 February 2008.

[7]. Sampath Deegalla and Henrik Bostrom, "Improving Fusion of Dimensionality Reduction Methods for Nearest Neighbor Classification", IEEE International Conference on Machine Learning and Applications, 978-0-7695-3926, 2009.

[8]. H. Abdi, "Partial Least Squares regression (PLS-regression)".Thousand Oaks (CA): Sage, pp. 792–795 2003.

[9]. C. Blake, E. Keogh, And C. J .Merz "UCI Repository of Machine Learning Databases" University Of California, Irvine.

[10]. A. K. Pujari "Data mining techniques" University Press February 2001.

[11]. X. Wu et al. "Top 10 algorithms in data mining" Knowledge information Springer-Verlag London Limited 2007.

[12]. Gursel Serpen and Santhosh Pathical "Classification in High-Dimensional Feature Spaces: Random Subsample Ensemble" IEEE -International Conference on Machine Learning and Applications 2009.

[13]. Hamid Parvin, Hosein Alizadeh, Mohsen Moshki, Behrouz Minaei-Bidgoli and Naser Mozayani "Divide & Conquer Classification and Optimization by Genetic Algorithm" third International Conference on Convergence and Hybrid Information Technology, IEEE-978-0-7695-3407-2008.

[14]. R. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," Expert Systems with Applications, vol. 33, 2007, pp. 49-60.

[15]. Oleg Okun, Helen Priisalu "Ensembles of K-Nearest Neighbors and Dimensionality Reduction", IEEE - International Joint Conference on Neural Networks (IJCNN), 978-1-4244-1821-2008.