Learning-based text classifiers using the Mahalanobis distance for correlated datasets

Noopur Srivastava*

Schneider Electric India Pvt. Ltd.,
Beary's Global Research Triangle,
Bangalore 560037, India
Email: noopur.srivastava@iiitb.org
*Corresponding author

Shrisha Rao

International Institute of Information Technology – Bangalore,
Bangalore 560100, India
Email: shrao@ieee.org

Abstract: We present a novel approach to text categorisation with the aid of the Mahalanobis distance measure for classification. For correlated datasets, classification using the Euclidean distance is not very accurate. The use of the Mahalanobis distance exploits the correlation in data for the purpose of classification. For achieving this on large datasets, an unsupervised dimensionality reduction technique, principal component analysis (PCA) is used prior to classification using the k-nearest neighbours (kNN) classifier. As kNN does not work well for high-dimensional data, and moreover computing correlations for huge and sparse data is inefficient, we use PCA to obtain a reduced dataset for the training phase. Experimental results show improvement in classification accuracy and a significant reduction in error percentage by using the proposed algorithm on huge datasets, in comparison with classifiers using the Euclidean distance.

Keywords: Mahalanobis distance; k-nearest neighbours; kNN; text classification; precision; recall; dimensionality reduction; principal component analysis; PCA.

Reference to this paper should be made as follows: Srivastava, N. and Rao, S. (2016) 'Learning-based text classifiers using the Mahalanobis distance for correlated datasets', *Int. J. Big Data Intelligence*, Vol. 3, No. 1, pp.18–27.

Biographical notes: Noopur Srivastava received her MTech degree from International Institute of Information Technology-Bangalore (IIIT-B). She is currently working as Senior Software Developer at Schneider Electric India Pvt. Ltd. She is interested in theory of algorithms, statistics, machine learning and artificial intelligence.

Shrisha Rao received his PhD in Computer Science from the University of Iowa, and before that his MS in Logic and Computation from Carnegie Mellon University. He is an Associate Professor at IIIT-Bangalore. His research interests are in distributed computing, specifically algorithms and approaches for concurrent and distributed systems, and include solar energy and microgrids, cloud computing, energy-aware computing ('green IT'), and demand side resource management. He is a senior member of the IEEE and a member of the IEEE Computer Society, the ACM, the American Mathematical Society, and the Computer Society of India.

1 Introduction

As information on the web expands with huge numbers of articles and documents in diverse forms being added constantly, the task of classifying information becomes challenging because of the large number of classes (Sebastiani, 2005; Bayer et al., 1998). While research has mostly focused on classification of web resources into categories, work is also in progress to classify information based on the sentiment associated with it, or to classify a message or review as deceptive or not (Liu, 2010). As far as classification into topic categories is concerned, the process of being able to effectively learn from sparse training documents is both crucial and difficult for the success of text categorisation. This is measured with *classification accuracy* as a metric; thus there is a strong interest in improving the classification accuracy for huge datasets (Sebastiani and Ricerche, 2002; Alpaydin, 2014).

Our work in this paper is to propose an enhanced text document classification framework named Mahal-kNN that uses an unsupervised dimensionality reduction technique (Lee and Verleysen, 2010) on the dataset, followed by the application of the k-nearest neighbours (kNN) classifier utilising Mahalanobis distance (Vapnik, 1998) for training and classification. In our algorithm Mahal-kNN, we exploit the correlations between terms in a given distribution through the Mahalanobis distance measure (Maesschalck et al., 2000), after using the unsupervised dimensionality reduction technique named principal component analysis (PCA). In the case of text categorisation, the input is the set of documents and the output is their assignment to their respective categories. We find that by our algorithm, better classification accuracy is obtained in large text datasets. It is seen that the accuracy obtained in Mahal-kNN is better compared to classifiers like Euclidean-kNN and naïve Bayes (Lewis, 1998). The results of the proposed algorithm are significant because they enable us to address the multi-criteria optimisation problem of considering the divergent goals of *precision* and *recall* in text categorisation (Makhoul et al., 1999; Manning et al., 2008) (precision refers to the fraction of the retrieved documents that are relevant, and recall refers to the fraction of relevant documents that are retrieved.)

Text preprocessing involves several data retrieval and cleansing techniques like stopword removal, whitespace removal, word segmentation, stemming (Porter, 1980), feature extraction (Combarro et al., 2005), term weighting, etc., and then representing a document in the vector space model (VSM) (Katzagiannaki and Plexousakis, 2003). In feature extraction, weights of keywords are computed and some keywords with weights lower than a predefined threshold are eliminated. To compute the weight of each keyword, popular methods such as term frequency (TF), term frequency inverse document frequency (TFIDF), and information gain (IG) (Katzagiannaki and Plexousakis, 2003) are adopted. The second step involves dimensionality reduction—since text document vectors are usually of very high dimensions, the PCA linear unsupervised dimensionality reduction technique is used to transform it into a lower dimensional space (Van der Maaten et al., 2009). The next step involves training a classifier like kNN with the statistics of each category formed. The test documents are then fed to the classifier to automate the classification process.

We choose kNN as our basic classifier because kNN is seen to be good after applying an unsupervised dimensionality reduction technique (Davi and Luz, 2007). In our method we aim to improve the classification accuracy by transforming the document vectors to a different vector space by means of feature transformation before proceeding with training and classification. Some well-known choices exist for classification methods, like neural networks (Kwok, 1999), kNN (Arya et al., 1998), support vector machines (SVM) (Joachims, 1998; Cortes and Vapnik, 1995), naïve Bayes (Lewis, 1998), etc. kNN is known to be sensitive to the distance function used but performs well when dimensionality reduction is applied using unsupervised learning (Davi and Luz, 2007).

With SVM it is challenging to determine the appropriate kernel functions, and moreover the accuracy of classification heavily relies on the *soft margin parameter*. This accounts for the classification errors seen with SVM while dealing with certain non-separable points. A more detailed discussion of the soft margin parameter is given by Gunn (1998).

An early attempt at text categorisation was made by Katzagiannaki and Plexousakis (2003). Their work requires the use of a boolean model and a VSM. The boolean approach consists of comparing terms of the documents by means of boolean operators like AND or OR. The VSM involves creation of document vectors in the form of a document matrix. To measure similarity, the cosine measure is used.

Zhang and Pan (2011) present a classification algorithm based on the Mahalanobis distance and kNN classifiers, which they call MDKNN. There, they propose that in place of using an appropriate value of k in finding the k nearest neighbours, we can make use of the Mahalanobis distance to decide which category the text data belongs to. However, their method is effective only when dealing with smaller numbers of documents, and it hence supports fewer features. Until recently, kNN classifiers were popularly used for classification, but with 'Big Data' it is realised that they cannot be used without dimensionality reduction. Therefore, their method cannot be employed readily on huge datasets, as finding the covariance of a sparse matrix is highly inefficient.

Reshef et al. (2011) propose a nonlinear R-squared measure for computing correlations on large datasets. The idea behind their MIC algorithm is that if a relationship exists between two variables, then a grid is drawn on a scatterplot representing the relationship between them. The MIC algorithm discretises the data onto many different two-dimensional grids, and calculates a normalised score that represents the mutual information of the two variables on each grid. In this algorithm a separate grid is created for depicting a relationship between each pair of variables; therefore for multivariate correlation, this approach is computationally expensive and becomes infeasible for large datasets. Our approach avoids computing such pairwise correlations in large datasets having multivariate dependency. We rather use dimensionality reduction to get a reduced dataset which captures most of the information, so that computing correlations is also not as expensive.

Davi and Luz (2007) present the application of active learning on text categorisation problems using unsupervised dimensionality reduction. They suggest that utilising active learning is beneficial in the scenario where vast amounts of data are unlabelled. Because of this reason, supervised dimensionality reduction techniques are not commonly used for active learning text categorisation tasks. Therefore, we have chosen unsupervised dimensionality reduction techniques.

Paralic and Kostial (2003) propose the idea of using domain knowledge for the purpose of information retrieval.

The ontology-based retrieval mechanism is compared with traditional full-text search based on the vector IR model as well as with the latent semantic indexing method, and is shown to be beneficial. Van der Maaten et al. (2009) compare different linear and nonlinear dimensionality reduction techniques, and show that traditional PCA outperforms other dimensionality reduction techniques.

Most of the recent literature (Davi and Luz, 2007) improves the learning algorithm by using active learning (Wu and Ostendorf, 2013) or semi-supervised learning depending on the nature of training data (Sebastiani and Ricerche, 2002; Jirina and Jirina, 2014). Improvements in preprocessing tasks, feature extraction (Mishra et al., 2011) and dimensional reduction techniques (Van der Maaten et al., 2009; Xu et al., 2013) have also been proposed. These techniques are usually based on some specific assumptions. For example, an assumption may be made on the distribution of the input data sample (Fisch et al., 2014), or for considering the nature of training data being labelled or unlabelled (Reshef et al., 2011), or on the basis of semantic relations (Alpaydin, 2014). If the assumption is not met, the performance of the algorithm may get worse.

Here we have changed the distance function to compute the proximity of the data points. Our aim is to consider correlation in datasets, which presently is not exploited to any significant extent. For correlated datasets, computing the Mahalanobis distance for classification generates better results as compared to the Euclidean distance.

In this work, we make use of unsupervised dimensionality reduction techniques for text classification on the basis of likely correlations in 'Big Data'. We find that when the data are correlated, Euclidean distance classification can degrade in accuracy. So for such kinds of correlated data, our Mahal-kNN algorithm uses the Mahalanobis distance measure for classifying test data. The Mahalanobis distance measure utilises covariance matrix information which becomes difficult to compute in case the data are huge and sparse. So our approach utilises an unsupervised dimensionality reduction technique to retain the benefits of using the Mahalanobis distance on correlated data, without complicating the process of computation. In this paper, we also give a mathematical analysis supporting the claim that Euclidean distance in the transformed space is actually equivalent to Mahalanobis distance in the original space.

A detailed description on text preprocessing, dimensionality reduction, classification system, Mahalanobis distance and algorithm are given in Section 2. Our experiments conducted on the standard Reuters dataset and their results are discussed in Section 3. Finally, conclusions are covered in Section 4.

2 Concept

2.1 Text preprocessing

The text categorisation process involves four major steps: text preprocessing, dimensionality reduction, the training phase, and the classification phase. Text preprocessing involves data preprocessing techniques like removing frequently occurring words (stopword removal), eliminating whitespace (whitespace removal), segmentation of a long sentence to several shorter terms (word segmentation), extracting root keyword from word (stemming) (Porter, 1980), eliminating insignificant keywords (feature extraction), computing the weights of keywords (term weighting), etc., and then representing a document in the VSM (Katzagiannaki and Plexousakis, 2003). The stopword removal technique removes words such as *a, an, the*, etc., which occur frequently without adding any distinct feature to a particular document. After this, the whitespace removal technique trims extra whitespace in the document (which is often used for esthetic or readability purposes but does not itself carry any semantic significance). Then, word segmentation is done on the document. Word segmentation is the problem of dividing a string of written language into its component words. In English and many other languages, space is a good approximation of a word delimiter. This process returns all the individual words in a particular document except stopwords. Since the word count may still be huge, we need techniques to select only those keywords from the set which could add on as a distinctive feature for a particular document.

The text classification process is performed after representing the documents in the VSM in the form of a document matrix. In the VSM (Katzagiannaki and Plexousakis, 2003), a document v_i is represented by an l-dimensional vector as $v_i = \langle w_{i1} w_{i2} \dots w_{il} \rangle$, where w_{ij} is a weight which we associate with the j^{th} keyword in the i^{th} document. Here, the dimensions of this vector space are mapped to an ordered set of keywords $K = k_1, k_2 \dots k_l$. The weights are determined by the frequency with which the keyword occurs within the document. For instance, if the keyword set is K = {finance, ball, bat, umpire, break} and we say document $v = \langle 4, 1, 0, 3, 2 \rangle$, it means that *finance, ball, bat, umpire, break* have occurred four, one, zero, three and two times respectively in that particular document. It is also important to note that in our model, these weights 'w_i' serve as features which we use to classify documents into categories. This l-dimensional document vector is associated with each of the n documents available in the pool. This forms a term document matrix (TMI) where rows correspond to documents in the collection, columns correspond to keywords, and entries correspond to weights.

The TFIDF measure is a refinement of the above stated weight measure (Salton and Buckley, 1988) which gives us a sense of the relative importance of that keyword in the document. Here the weight W_{ij} associated with the keyword is the relative frequency of the j^{th} keyword in the i^{th} document. We define this measure as

$$W_{ij} = TF_{ij} \times \log_2(N/n_j).$$

Here TF_{ij} is the term frequency of the j^{th} keyword in the i^{th} document, and the log-term is called as the inverse document frequency, where N is the total number

of documents in the collection and n_j is the number of documents containing the j^{th} keyword.

Porter (1980) gives a process for reducing derived words to their stem or root forms. Willett (2006) gives a stemming algorithm is a tool for morphological analysis that tries to associate variants of the same term with a root form. This process is essential for the operation of classifiers and index builders or searchers. It is because the operation is less dependent on particular word forms and therefore reduces the potential sizes of vocabularies which might otherwise contain several possible forms for a given stem.

2.2 Dimensionality reduction and classification

Each object in an abstract mathematical space can be described with certain number of dimensions, and *dimensionality reduction* refers to the study of methods for reducing this number to get a smaller abstraction of the object that retains its essential properties. In the process, high-dimensional data are represented in lower-dimensional space such that the meaning remains intact (Dash and Liu, 2007). The high-dimensional data cannot be reduced beyond some intrinsic dimensionality value, where the intrinsic dimensionality of data is the minimum number of parameters needed to capture the observed properties of data. Earlier machine learning and pattern recognition algorithms were often susceptible to the well-known problem of "the curse of dimensionality" (Indyk and Motwani, 1998), which refers to the degradation in performance of data processing algorithms as the number of dimensions of the data increases. For addressing this, dimensionality reduction techniques are used before classification. These techniques are applied as data preprocessing step so as to reduce the complexity of data model.

In mathematical terms, the process of dimensionality reduction can be stated as follows:

1 given an l-dimensional random variable $x = (x_1, \ldots, x_l)^T$

2 find a lower d-dimensional representation of it, $s = (s_1, \ldots, s_d)^T$ with $d \leq l$ which captures the content in the original data according to some criteria.

2.2.1 Principal component analysis

PCA (Davi and Luz, 2007) is a method of reducing higher-dimensional data to a new lower-dimensional space with minimum loss of information. The idea behind the method is that the direction of the largest variance carries most of the important information regarding dataset. Therefore, this unsupervised technique extracts features in the direction of maximal variance in the data. Here, the n-dimensional data space is rotated so that the direction of the maximum variance becomes the coordinate axis in the lower dimensional space. On the other hand, it can be seen as a linear projection that minimises the mean squared distance between the data points and the corresponding projected points. The new coordinates obtained after orthogonal transformation of the original coordinate system are called the *principal components*.

Consider the input high dimensional dataset A to be an $n \times l$ matrix where n corresponds to the number of documents and l corresponds to the dimensions to be reduced. The procedure of PCA can be described as follows.

Before proceeding to calculation, the input data are to be normalised by subtracting the empirical mean. Hence we calculate the empirical mean along each dimension $j = 1, \ldots, l$.

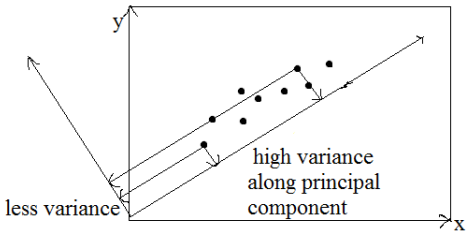
$$\mu[j] = \frac{1}{n} \sum_{i=1}^{n} A[i, j]. \tag{1}$$

where μ is the empirical mean vector of dimension $1 \times l$. The deviation from the mean has to be calculated to minimise the mean square error in approximating the data. The mean vector μ is subtracted from each row of matrix A.

$$B = A - h.\mu \tag{2}$$

where h is a $n \times 1$ column vector of all 1 s, and B is a normalised matrix of size $n \times l$.

Figure 1 Variance along principal component



The covariance matrix C should be calculated which represents how much the l dimensions vary from the mean with respect to one another. The principal components which maximise the variance in the lower-dimensional space are calculated using this covariance matrix C.

$$C = \frac{1}{n} \Sigma B.B^T \tag{3}$$

Now the required principal components are the eigenvectors v of the covariance matrix C that satisfy the equation

$$C.v = \lambda.v \tag{4}$$

The set of all eigenvectors V satisfy

$$V^{-1}CV = D, \tag{5}$$

where D corresponds to the diagonal matrix of all the eigenvalues, where each eigenvalue represents the variance contributed by the corresponding principal component (eigenvector).

Finally, the columns of D and correspondingly the columns of V are sorted in the decreasing order of the eigenvalues, and the first d eigenvalues are selected to reduce the given l-dimensional dataset to a d-dimensional dataset. This d-dimensional dataset is the required reduced dataset which contains most of the important information from the original dataset.

2.2.2 k-nearest neighbours

The kNN algorithm classifies an object by measuring the distance between the query object scenario and a set of predetermined scenarios in the dataset. This classification is performed on the basis of closest training examples and achieved by the majority vote of its kNN (k is a positive integer, typically small). If k = 1, the object is simply assigned to the class of its nearest neighbour where neighbours are taken from a set of objects for which the correct classification is known.

Our version of the kNN algorithm functions as follows:

1 given a query instance x_q to be classified, we compute the Mahalanobis distance between the sample point and the k instances from the training data

2 let $x_1, x_2, \ldots, x_k$ denote the k instances from training examples that are nearest to x_q

3 return the class that has a majority of the k instances.

2.3 The Mahalanobis distance measure

In statistics, the Mahalanobis distance is a measure which is based on the correlations between variables (weights in our case) by which different patterns can be identified and analysed. It gauges the similarity of an unknown sample vector to a known set (Haasdonk and Pekalska, 2008). It differs from the traditionally used Euclidean measure by taking into account the correlations of the dataset and its distribution, and is also scale-invariant. Formally, the Mahalanobis distance of a multivariate vector x from mean vector μ

$$x = (x_1, x_2, x_3 \ldots x_d)^T \tag{6}$$

$$\mu = (\mu_1, \mu_2, \mu_3 \ldots \mu_d)^T \tag{7}$$

is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)} \tag{8}$$

Here S is the covariance matrix of a particular category. We define

$$S = E[(X-\mu)(X-\mu)^T] \tag{9}$$

The difference between the Euclidean distance and the Mahalanobis distance is an important one. Figure 2 shows simulated bivariate normal data that are overlaid with prediction ellipses on x and y axes. A prediction ellipse is a region where a new observation is predicted among the population. The ellipses in the graph are the 10% (innermost), 20%, …, and 90% (outermost) prediction ellipses for the bivariate normal distribution that generated the data. The probability density is high for ellipses near the origin, such as the 10% prediction ellipse. The density is low for ellipses which are further away, such as the 90% prediction ellipse. Usually, the distance between two observations is specified by measuring how many standard deviations apart they are.

Figure 2 Ellipsoids showing correlated data

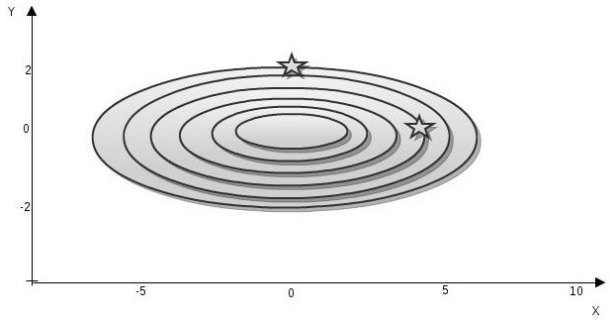


Figure 2 shows two observations displayed by stars. The first observation is at coordinates (5, 0), and the second at (0, 1). The origin is the multivariate center of this distribution. The Euclidean distances are 5 and 1 respectively, suggesting that the point at (0, 1) is closer to the origin. However, for this distribution, the variance in the y direction is less than the variance in the x direction, so in some sense the point (0, 1) is more standard deviations away from the origin than (5, 0) is. The point (0, 1) is located at the 90% prediction ellipse, whereas the point at (5, 0) is located at about the 75% prediction ellipse. This suggests that the point at (5, 0) is closer to the origin in the sense that it is more likely to observe an observation near (5, 0) than to observe one near (0, 1). The probability density is higher near (5, 0) than it is near (0, 1).

We will now describe how this measure better addresses the optimisation problem between precision and recall compared to the traditional Euclidean measure. In the Euclidean approach, we define an n-ball (or circle in 2-D) around a centroid with a defined threshold-radius, within which we say every point belongs to the cluster. Increasing this radius may help solve the recall problem while however degrading precision (Makhoul et al., 1999). In the Mahalanobis approach, the contour of a point with constant Mahalanobis distance from the centroid is an n-dimensional ellipsoid. For instance, in the 2-D case it is an ellipse. We see that the major axis is oriented with the direction along which there is maximum variance in the cluster-distribution, and the minor axis is in the direction of minimum variance. We can thus see that the possibility of vectors being erroneously classified is reduced, thus increasing precision. Also, more vectors in the direction of maximum variance are included, thus enhancing recall. In the n-D case, the axes of this ellipsoid are along the eigenvectors of the covariance matrix S (Ihrke, 2011).

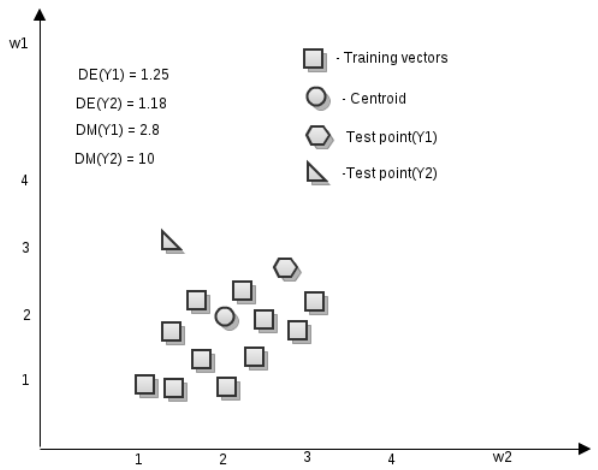
The TMI is comprised of training vectors $X = X_1 \ldots X_n$. From this, we first compute the mean vector or centroid μ. We then estimate the covariance matrix of the distribution as:

$$S = \sum_{i=1}^{n} \frac{(X_i - \mu)(X_i - \mu)^T}{(n-1)} \tag{10}$$

With this estimate of S, we compute the Mahalanobis distance using (8).

In Figure 3, DE refers to the Euclidean distance and DM to the Mahalanobis distance. We can see from Figure 3 that the test vector y_1 is more probably a part of the cluster than y_2. We also observe that the Euclidean distances (DE) of y_1 and y_2 are comparable, indicating that they are almost equally likely to be a part of the cluster, which is counter-intuitive. The Mahalanobis distance (DM) indicates that the point y_1 is far more likely to be a part of the cluster than y_2, in line with our intuition.

Figure 3 Comparison of the Mahalanobis distance with the Euclidean distance



In feature transformation we construct a transformed space in which the Mahalanobis distance between any two points is the same as the Euclidean distance. This enables us to use the kNN classifiers in the transformed space with the Euclidean distance, though we are actually computing with respect to the Mahalanobis distance in the original feature space. This allows us to use the kNN classifiers with the Mahalanobis distance. Hence, we propose an enhanced text document classification framework wherein the document vectors are subjected to feature transformation before training and classification by the kNNs. The preprocessed document vectors are transformed into another vector space, then reduced to a meaningful lower-dimension space, and then fed into the classifier for training and classification using the kNN classifier. We mathematically prove that in the transformed space the distance between two points is the Mahalanobis distance between the pre-images of the same two points in the original space. The Mahalanobis distance of a point $x = (x_1, x_2, x_3 \ldots x_d)^T$ from a set of data points with mean $\mu = (\mu_1, \mu_2, \mu_3 \ldots \mu_d)^T$ is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{11}$$

The covariance matrix S is diagonalisable as:

$$S = QDQ^T \tag{12}$$

D is the diagonal containing the eigenvalues of S as diagonal elements. Q is a matrix with corresponding eigenvectors. Now, rewriting S we have:

$$S^{-1} = QD^{-1}Q^T \tag{13}$$

Here D^{-1} is the inverse of D. Rewriting the Mahalanobis equation (8) we have:

$$D_M(x) = \sqrt{(x-\mu)^T QD^{-1}Q^T(x-\mu)} \tag{14}$$

$$D_M(x) = \sqrt{(Q^T(x-\mu))^T D^{-1}Q^T(x-\mu)} \tag{15}$$

Defining $x' = Q^T(x-\mu)$ (as the feature transformation) we have:

$$D_M(x) = \sqrt{x'^T D^{-1} x'} \tag{16}$$

Since D is diagonal, we write $D = \sqrt{D}\sqrt{D}$

$$\hat{x} = \sqrt{D^{-1}}x' \tag{17}$$

$$D_M(x) = \sqrt{(\hat{x})^T(\hat{x})} \tag{18}$$

where $\hat{x}$ is the transformation of x.

We see that the Mahalanobis distance in the original feature space is the Euclidean distance in the transformed space ($\hat{x}$ is the transformation of x).

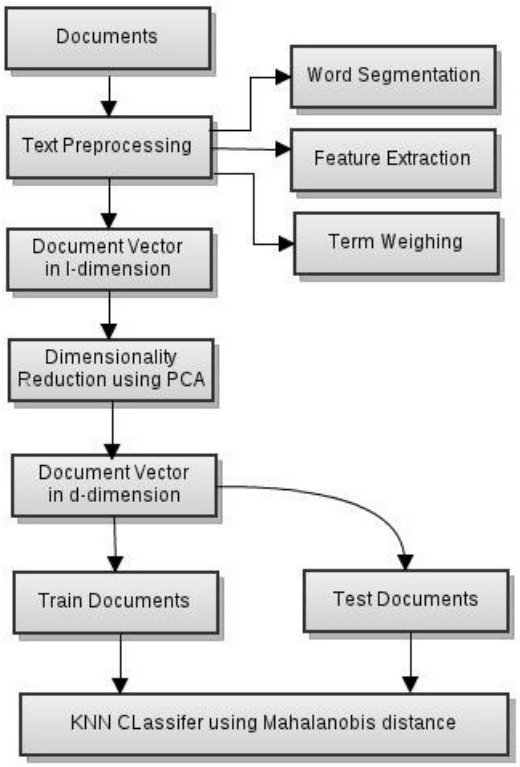
2.4 Algorithm

The proposed algorithm Mahal-kNN is divided into four phases: the preprocessing phase, the dimensionality reduction phase, the training phase, and the classification phase. Figure 4 shows the entire process of text classification for the proposed Mahal-kNN algorithm. The Mahal-kNN classification algorithm is divided into these phases as the output of one phase becomes input to the next.

In the preprocessing phase, the dataset (the Reuters-21578 dataset in XML format, in our analysis) is taken as input. Cleansing techniques like stopword removal and whitespace removal are used, after which the feature extraction techniques are used as discussed previously. Here the features are the sets of keywords specific to each of the documents, which helps in determining its class. Word segmentation, the stemming algorithm (Porter, 1980) and TFIDF reduces the Reuters-21578 XML (or other) documents into an l-dimensional matrix. This becomes input for the next phase, the dimensionality reduction

phase; the term information matrix obtained in the first phase is a high-dimensional matrix. This phase reduces the l-dimensional matrix to d-dimensional matrix in the way also mentioned previously. The training phase takes as input the above obtained matrix along with the category detail of each training document, which is fed to the kNN classifier. As a result of this phase, we obtain a trained classifier which is to be used for the purpose of classification of the set of test documents. Therefore, this classifier along with the set of testing documents constitutes an input for the classification phase. kNN classifies the document to a category with the smallest Mahalanobis distance. For each document in the set of test documents, the output is the category label for that document. The four phases are discussed as follows.

Figure 4 Mahal-kNN classification system



2.4.1 Preprocessing phase

- *Input:* Reuters-21578 documents in XML format
  1. read Reuters-21578 corpus
  2. perform word segmentation
  3. perform stemming and term weighing
  4. perform TFIDF for eliminating insignificant words.
- *Output:* TMI of size $n \times l$.

2.4.2 Dimensionality reduction phase

- *Input:* matrix of size $n \times l$.
  1. Compute empirical mean for each dimension as given in (1) on matrix A.
  2. Normalise matrix A to obtain matrix B.
  3. The covariance matrix C is computed and principal components are obtained, which are the eigenvectors of the solution of (4).
  4. The diagonal matrix D is computed as from (5). Eigenvalues are arranged in decreasing order and the first d-eigenvalues are selected.
- *Output:* matrix of size $n \times d$.

2.4.3 Training phase

- *Input:* pool of labelled examples (training dataset T=(X_1,X_2, …, X_N), X_N is training set of a category and N is the category number).
  1. using the training dataset, generate a kNN classifier ϕ_f
  2. based on (9), calculate each category's covariance matrix S = (S_1, S_2, …, S_N), where S_N corresponds to the N^{th} category.
- *Output:* a classifier ϕ_f.

2.4.4 Classification phase

- *Input:* kNN-Classifier and testing documents
  1. based on (11) and covariance matrix S from the training stage, calculate the Mahalanobis distance between each category and test datapoint Y
  2. find the smallest distance which is nearest to Y.
- *Output:* assign the category with the smallest distance.

3 Results

Experiments were conducted to examine the effect of the proposed Mahal-kNN algorithm on the performance of k-NN classifier. One of the standard benchmark corpora generally used in text categorisation research, the Reuters-21578 corpus, was used.

The original feature set was obtained by pre-processing the corpus to remove stopwords and punctuation. Stemming was performed using the standard Porter stemming algorithm (Willett, 2006). Reduced feature sets were constructed using the unsupervised dimensionality reduction techniques performed on the unlabelled and seed data. PCA transformed the original data onto a d-dimensional space where d was chosen as the number of principal components

which accounted for 90% of the variance in the data. Both the training and test sets were reexpressed in the reduced feature representation.

The kNN is a high performance classifier for text categorisation; however, it is sensitive to high-dimensional data. While it is not commonly used for learning text categorisation tasks, we chose it since it benefits greatly from dimensionality reduction. The output of the kNN was transformed into a class membership probability estimate where the distribution is based on the Mahalanobis distance of the query example to the k nearest neighbours. The estimate was then used as a measure of uncertainty. The experiment was performed with different values of k (k initially being from 3 to 20, and then choosing the best value of k as classification accuracy). kNN efficiency increases when PCA was deployed on the Reuters-21578 dataset which reduced high dimensional data to low dimensional data. The statistics toolbox of MATLAB was used to perform these experiments.

As mentioned, for experimental purposes we used the standard Reuters-21578 (Lewis, 2004) dataset. Historically, the classic Reuters-21578 collection was the main benchmark for text classification evaluation. This is a collection of 21578 newswire articles, originally collected and labelled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text classification system. The articles are assigned classes from a set of 118 topic categories.

Recent work almost invariably uses the ModApte split, which includes only documents that were viewed and assessed by a human indexer, and comprises 9,603 training documents and 3,299 test documents. The distribution of documents in classes is very uneven, and some work evaluates systems on only documents in the ten largest classes. The list of ten largest classes is as given in Table 1.

Table 1 Reuters-21578's ten largest classes

Classes	*Training docs*	*Testing docs*
Earn	2,877	1,087
Acquisitions	1,650	179
Money-fx	538	179
Grain	433	149
Crude	389	189
Trade	369	119
Interest	347	131
Ship	197	89
Wheat	212	71
Corn	182	56

The aim of classification is to minimise the classification error on test data using (19), where CE stands for classification error and CA stands for classification accuracy. This measure is good in those cases where percentage of documents in a class is high, but in case where this percentage is small, accuracy is not considered to be a good measure.

$$CE = 1 - CA \tag{19}$$

For small classes, precision, recall and F1 are better measures. The F1 score is a measure of accuracy of a test. It considers both the precision (π) and the recall (ρ) of the test to compute the score: π is the number of correct results divided by the number of all returned results, and ρ is the number of correct results divided by the number of results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The F1 value of precision (π) and recall (ρ) was chosen as the performance metric (where $F1 = 2.\pi.\rho/(\pi+\rho)$).

We used both the micro-averaged F1 score and the macro-averaged F1 score as evaluation measures. Micro-averaged values are calculated by constructing a global contingency table for a class and then calculating precision and recall using these sums. In contrast, macro-averaged scores are calculated by first calculating precision and recall for each category and then taking the average of these. The notable difference between these two calculations is that micro-averaging gives equal weight to every document (it is called a document-pivoted measure) while macro-averaging gives equal weight to every category (category-pivoted measure).

In Table 2, the micro-averaged and macro-averaged scores for F1 are given for naïve Bayes (Chen and Fu, 2005), Euclidean-kNN and Mahal-kNN. As shown in Table 2, the micro-averaged F1 score for Mahal-kNN is 91%, which is 5% more than Euclidean-kNN (86%). Similarly, the macro-averaged F1 score for Mahal-kNN is 64% which is greater than other classifiers. The results show that the Mahal-kNN returns better F1 scores as compared to conventional classifiers.

Table 2 Macro F1 and Micro F1 scores of different classifiers

Classifier	*Macro F1*	*Micro F1*
Naive Bayes	47	80
Rocchio	59	85
Euclidean-kNN	60	86
Mahal-kNN	64	91

Figure 5 Comparison of the Mahalanobis distance with the Euclidean distance on Reuters-21578

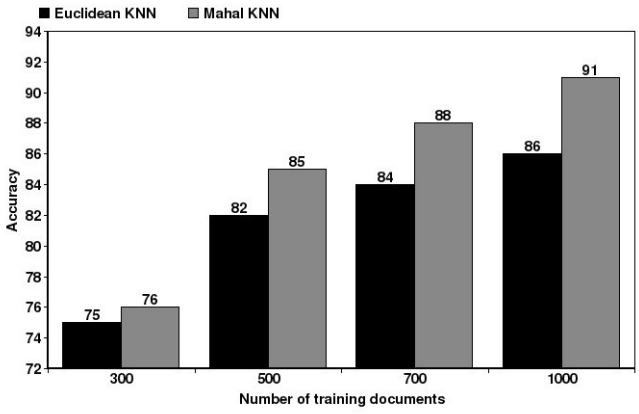
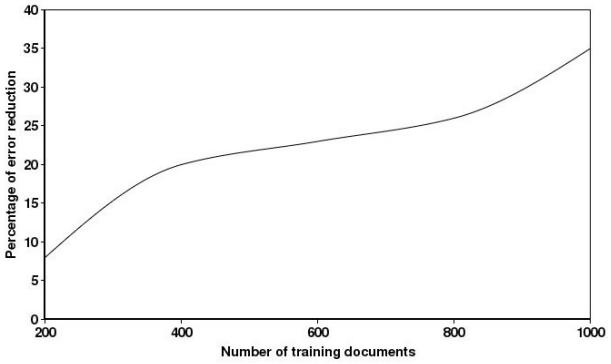


Figure 5 shows the relation between accuracy and number of training documents. As the number of training documents increases, the accuracy increases gradually. In each case, the accuracy obtained through the proposed method was higher as compared to the traditional kNN classification technique.

Figure 6 shows the percentage reduction in classification error for Mahal-kNN relative to the classification error for Euclidean-kNN in the process of text classification. As the number of training documents increases, Mahal-kNN shows increased error reduction percentage, with a highest of 35% on Euclidean-kNN.

Figure 6 Percentage of error reduction relative to Euclidean kNN



4 Conclusions

We have seen that the Mahalanobis measure improves classification accuracy. We made a comparative study between the Euclidean and the Mahalanobis distance measures to clearly justify how the accuracy is better with the latter. This work utilises the benefits of the Mahalanobis distance for large datasets by using an unsupervised dimensionality reduction techique, since for huge datasets the Mahalanobis distance cannot be readily used without dimensionality reduction (because computing covariance matrix for high-dimensional data is time-consuming and inefficient). The use of PCA in this respect reduces sparse high-dimensional data to low dimensional data while retaining most of the information. The results are particularly promising because they are accurate and as described, the use of Mahalanobis distance provides the right balance between precision and recall. Classification has many applications in customer segmentation, business modelling, marketing, credit analysis, and biomedical and drug response modelling, etc., so this work will help solve many practical problems in such diverse domains.

References

Alpaydin, E. (2014) *Introduction to Machine Learning*, 3rd ed., MIT Press.

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. and Wu, A.Y. (1998) 'An optimal algorithm for approximate nearest neighbor searching in fixed dimensions', *Journal of the ACM* Vol. 45, No. 6, pp.891–923, November, doi: 10.1145/293347.293348.

Bayer, T., Kressel, U., Mogg-Schneider, H. and Renz, I. (1998) 'Categorizing paper documents: a generic system for domain and language independent text categorization', *Computer Vision and Image Understanding*, Vol. 70, No. 3, pp.299–306.

Chen, Z. and Fu, B. (2005) 'On the complexity of Rocchio's similarity-based relevance feedback algorithm', *Journal of the American Society for Information Science and Technology*, Vol. 58, pp.1392–1400.

Combarro, E.F., Montanes, E., Diaz, I., Ranilla, J. and Mones, R. (2005) 'Introducing a family of linear measures for feature selection in text categorization', *IEEE Trans. Knowl. Data Eng.*, Vol. 17, pp.1223–1232, September, doi: 10.1109/TKDE.2005.149.

Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, Vol. 20, pp.273–297, doi: 10.1007/BF00994018.

Dash, M. and Liu, H. (2007) 'Dimensionality reduction', *Wiley Encyclopedia of Computer Science and Engineering*, December, doi: 10.1002/9780470050118.ecse112.

Davi, M. and Luz, S. (2007) 'Dimensionality reduction for active learning with nearest neighbour classifier in text categorisation problems', in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, Cincinnati, OH, December, pp.292–297, doi: 10.1109/ICMLA.2007.9.

Fisch, D., Kalkowski, E. and Sick, B. (2014) 'Knowledge fusion for probabilistic generative classifiers with data mining applications', *IEEE Trans. Knowl. Data Eng.*, March, Vol. 26, No. 3, pp.652–666.

Gunn, S.R. (1998) *Support Vector Machines for Classification and Regression*, Image Speech and Intelligent Systems Group, University of Southampton, Southampton, UK, Tech. Rep., May [online] http://users.ecs.soton.ac.uk/srg/publications/pdf/SVM.pdf.

Haasdonk, B. and Pekalska, E. (2008) 'Classification with kernel Mahalanobis distance classifiers', in *32nd Annual Conference of the Gesellschaft fur Klassifikation e.V.*, July, pp.351–361, doi: 10.1007/978-3-642- 01044-6_32.

Ihrke, I. (2011) 'Some notes on ellipses', February [online] http://manao.inria.fr/perso/ ihrke/software/ellipse.pdf.

Indyk, P. and Motwani, R. (1998) 'Approximate nearest neighbors: towards removing the curse of dimensionality', in *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC '98)*, pp.604–613, doi: 10.1145/276698.276876.

Jirina, M. and Jirina, (2014) 'Correlation dimension-based classifier', *IEEE Trans. Cybern.*, December, Vol. 44, No. 12, pp.2253–2263.

Joachims, T. (1998) 'Text categorization with support vector machines: learning with many relevant features', in *10th European Conference on Machine Learning (ECML-98)*, Ser. LNCS 1398, pp.137–142, doi: 10.1007/BFb0026683.

Katzagiannaki, I-E. and Plexousakis, D. (2003) 'Information dissemination based on semantic relations', in J. Eder and T. Welzer (Eds.): *The 15th Conference on Advanced Information Systems Engineering (CAiSE '03), Ser. CEUR Workshop Proceedings*, June, Vol. 74.

Kwok, J-Y. (1999) 'Moderating the outputs of support vector machine classifiers', *IEEE Trans. Neural Netw.*, Vol. 10, No. 5, pp.1018–1031, September, doi: 10.1109/72.788642.

Lee, J.A. and Verleysen, M. (2010) 'Unsupervised dimensionality reduction: overview and recent advances', in *The International Joint Conference on Neural Networks (IJCNN 2010)*, Barcelona, Spain, July, doi: 10.1109/IJCNN.2010.5596721.

Lewis, D.D. (1998) 'Naive (Bayes) at forty: the independence assumption in information retrieval', in *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, Springer-Verlag, London, UK, pp.4–15.

Lewis, D.D. (2004) 'Reuters-21578 text categorization test collection' [online] http://www.daviddlewis.com/resources/testcollections/reuters21578.

Liu, B. (2010) 'Sentiment analysis and subjectivity', in N. Indurkhya and F.J. Demerau (Eds.): *Handbook of Natural Language Processing*, 2nd ed., CRC Press.

Maesschalck, R.D., Jouan-Rimbaud, D. and Massart, D. (2000) 'The Mahalanobis distance', *Chemometrics and Intelligent Laboratory Systems*, Vol. 50, No. 1, pp.1–18, doi: 10.1016/S0169-7439(99)00047-7.

Makhoul, J., Kubala, F., Schwartz, R. and Weischedel, R. (1999) 'Performance measures for information extraction', in *Proceedings of the DARPA Broadcast News Workshop*, pp.249–252.

Manning, C.D., Raghavan, P. and Schutze, H. (2008) *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA.

Mishra, S., Mishra, D., Das, S. and Rath, A. (2011) 'Feature reduction using principal component analysis for agricultural data set', in *3rd International Conference on Electronics Computer Technology (ICECT 2011)*, April, Vol. 2, pp.209–213.

Paralic, J. and Kostial, I. (2003) 'Ontology-based information retrieval', in *14th International Conference on Information and Intelligent systems (IIS 2003)*, pp.1024–1025.

Porter, M.F. (1980) 'An algorithm for suffix stripping', *Program*, Vol. 14, No. 3, pp.130–137.

Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M. and Sabeti, P.C. (2011) 'Detecting novel associations in large data sets', *Science*, December, Vol. 334, No. 6062, pp.1518–1524, doi: 10.1126/science.1205438.

Salton, G. and Buckley, C. (1988) 'Term-weighting approaches in automatic text retrieval', *Information Processing and Management*, Vol. 24, No. 5, pp.513–523.

Sebastiani, F. and Ricerche, C.N.D. (2002) 'Machine learning in automated text categorization', *ACM Computing Surveys*, Vol. 34, pp.1–47.

Sebastiani, F. (2005) 'Text categorization', in *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pp.109–129, WIT Press.

Van der Maaten, L., Postma, E. and van den Herik, H. (2009) *Dimensionality Reduction: A Comparative Review*, Tilburg University, Tilburg, The Netherlands, Tech. Rep. TiCC TR 2009005, October [online] https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf.

Vapnik, V.N. (1998) *Statistical Learning Theory*, Wiley, September.

Willett, P. (2006) 'The Porter stemming algorithm: then and now', *Program: Electronic Library and Information Systems*, Vol. 40, No. 3, pp.219–223.

Wu, W. and Ostendorf, M. (2013) 'Graph-based query strategies for active learning', *IEEE Trans. Audio, Speech, Lang. Proc.*, February, Vol. 21, No. 2, pp.260–269.

Xu, M., Chen, H. and Varshney, P.K. (2013) 'Dimensionality reduction for registration of high-dimensional data sets', *IEEE Trans. Image Process.*, August, Vol. 22, No. 8, pp.3041–3049.

Zhang, S. and Pan, X. (2011) 'A novel text classification based on Mahalanobis distance', in *3rd International Conference on Computer Research and Development (ICCRD 2011)*, March, Vol. 3, pp.156–158, doi: 10.1109/ICCRD.2011.5764268.