



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Review of Nearest Neighbor-Support Vector Machines Hybrid Classification Models

¹Lam Hong, Lee, ¹Chin Heng, Wan, ¹Tien Fui, Yong and ²Hui Meian, Kok

¹Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman, Jalan Universiti,
Bandar Barat, 31900 Kampar, Perak, Malaysia

²Faculty of Arts and Social Science, Universiti Tunku Abdul Rahman,
Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia

Abstract: This study presents our investigation on different hybrid classification models which integrate support vector machines with nearest neighbor algorithm. We study the advantages and disadvantages of Support Vector Machines (SVM) classification and k-nearest neighbor (KNN) classification in performing their classification tasks. In our investigation, we found that the well-performing SVM classification approach may suffer from high time consumption, high CPU and physical memory usages, due to its convoluted training and classifying processes, especially when the dimensionality of data is high. On the other hand, KNN classification approach which implements NN algorithm is outstanding with its simplicity and low cost training process. However, it has been reported to be less accurate than the SVM classification. Many research works have been carried out in order to further improve the performance of the established SVM classifier by integrating Nearest Neighbor (NN) algorithm into the conventional SVM classification approach. The research works which have been reviewed and investigated in this paper emphasize in simplifying the convoluted training and classifying processes and further improving the classification algorithm of the conventional SVM, using NN algorithm. Overall, we concluded that while SVM classification approach has been reported as one of the best-performing classifiers since some decades now, it could be further improved by using NN algorithm in order to obtain more effective and efficient classification models.

Key words: Machine learning, pattern recognition, gene expression data classification, spam e-mail filtering, visual category recognition

INTRODUCTION

In recent years, the rapidly growing phenomenon of creating, editing, processing, transferring and storing data in digital form has made the automated machine learning classification systems become important due to the advent of large amount of electronic data. These classification systems are useful in discovering, organizing, analyzing and mining data in order to transform data into information. Since some decades now, an increasing number of machine learning approaches have been developed to perform the tasks of classifying data into groups for some specified purposes, including decision tree induction (Greiner and Schaffer, 2001; Quinlan, 1993), rule induction (Apte *et al.*, 1994),

self-organizing map (Isa *et al.*, 2008c, 2009), k-nearest neighbor classification (Han *et al.*, 1999; Shin *et al.*, 2006; Osuna, 2002; Tan, 2006), artificial neural network (Abd Alla, 2006; Soltanizadeh and Shakirar, 2008), support vector machines (Burgess, 1998; Haykin, 1999; Isa *et al.*, 2008b, c; Lee, 2008; Joachims, 1998, 1999; Lin, 1999; Sani *et al.*, 2009; Bayesian classification (Domingos and Pazzani, 1997; Eyheramendy *et al.*, 2003; Isa *et al.*, 2008a; Kim *et al.*, 2002a; Lee, 2008; Lee *et al.*, 2010; McCallum and Nigam, 2003; Rish, 2001).

Each of the classification schemes previously mentioned has its own unique properties and associated strengths and problems. The simple decision tree induction and rule induction approaches are easy to be understood and interpreted. As the trade-off, the low

Corresponding Author: Lam Hong, Lee, Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman, Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia
Tel: +605-4688888 Fax: +605-4661672

classification performance of these approaches has restricted them to be widely implemented in real world application, especially when the number of distinguishing features within the data is large (Greiner and Schaffer, 2001; Quinlan, 1993). k-nearest neighbor (KNN) is another approach which is easy to be implemented with high degree of effectiveness in many classification tasks of varying problem domains. However, when the amount of training data is large, k-nearest neighbor approach suffers from the problem of computational intensive and its classification accuracy could be drastically degraded when the number of attributes grows (Han *et al.*, 1999; Shin *et al.*, 2006; Osuna, 2002; Tan, 2006). Artificial neural networks outperform the other classification approaches with its ability in handling data with high-dimensional features and also noisy and contradictory data. However, the high computing cost which consumes high CPU and physical memory usage has become the main disadvantage of the artificial neural networks. Bayesian approach is outstanding with its simplicity and low computational cost in both the training and classifying stage and it has been widely implemented in various types of domains and applications. However, this generative method has been reported to be less accurate than the discriminative methods such as support vector machines (Brucher *et al.*, 2002; Chakrabarti *et al.*, 2003; Godbole, 2006; Joachims, 1998, 1999; Lin, 1999; Yang and Pederson, 1997; Yang and Liu, 1999).

As one of the discriminative classification methods, Support Vector Machines (SVM) classification has been shown to be more accurate than other classification approaches (Brucher *et al.*, 2002; Chakrabarti *et al.*, 2003; Godbole, 2006; Isa *et al.*, 2008b; Joachims, 1998, 1999; Lee, 2008; Lin, 1999; Yang and Pederson, 1997, Yang and Liu, 1999). The outstanding generalization ability of the SVM is contributed by the implementation of Structural Risk Minimization (SRM) principle which entails finding an optimal separating hyper-plane, thus guaranteeing the lowest classification error (Haykin, 1999). This unique characteristic of the SVM has contributed to its success in most of the classification application of varying problem domains, such as text classification (Joachims, 1998, 1999; Lee, 2008), spam e-mail filtering (Drucker *et al.*, 1999), gene expression data classification (Brown *et al.*, 1999), protein fold and remote homology detection (Rangwala and Karypis, 2005), content based image retrieval (Tao *et al.*, 2006), face recognition (Sani *et al.*, 2009), alternative exons identification (Dror *et al.*, 2005)

and texture classification (Kim *et al.*, 2002b; Li *et al.*, 2003).

There exist some problems associated with the implementation of SVM as a classifier. One of the major problems of the SVM classification approach is that efforts are needed to transform data into a representation suitable format, typically in numerical. Furthermore, the computation of inner products between support vectors and data points and the iterative binary classification processes in multi-class classification, lead to its convoluted training and categorizing algorithms, hence consume a high computing time and cost (Borges, 1998; Haykin, 1999; Joachims, 1998, 1999; Lin, 1999).

As for the review study presented in this paper, we configure the situations and reasons for SVM classification to fail in performing effective and efficient classification tasks. It is the nature and properties of the conventional SVM algorithm that restrict the SVM classifier to perform well in certain problem domains, such as gene expression data classification, visual category recognition and spam e-mail filtering. In order to overcome the problems of the conventional SVM as mentioned previously, many research works have been carried out with the same target to improve the classification performance of SVM by modifying the algorithm of conventional SVM. In this study, we carry out the review of the research works which modify conventional SVM by integrating Nearest Neighbor (NN) algorithm, in order to develop hybrid classification models which fulfill the problem solving requirements of specified domains, hence contribute to better classification effectiveness and efficiency. Although the investigated NN-SVM hybrid classification models presented the integration of NN algorithm with SVM approach, they have their own unique way and philosophy of integration, in order to counter their own classification domains with unique problem solving requirements.

BACKGROUND OF SUPPORT VECTOR MACHINES AND k-NEAREST NEIGHBOR CLASSIFICATION APPROACHES

Support vector machines classification approach: Support Vector Machines (SVM) is one of the discriminative classification approaches which is commonly recognized to be more accurate. SVM classification approach is based on Structural Risk Minimization (SRM) principle from statistical learning theory (Vapnik, 1995). SRM is an inductive principle for

model selection used for learning from finite training data and it provides a method for controlling the generalization ability of learning machines that uses a small size training data (Zhang, 2004). The idea of this principle is to find a hypothesis to guarantee the lowest true error. In addition to this, the derivation of SVM is mathematically rigorous and very open to theoretical understanding and analysis (Joachims, 1998).

SVM needs both positive and negative training datasets which are uncommon for other classification methods. These positive and negative training sets are needed for SVM to seek for the decision surface, also known as hyper plane, which best separate the positive from the negative data in the n -dimensional space (Brucher *et al.*, 2002). The document representatives which are closest to the decision surface are called support vectors. The performance of SVM classification remains unchanged if documents that do not belong to the support vectors are removed from the set of training data (Brucher *et al.*, 2002).

SVM classification approach is outstanding from the others with its better classification performance and its ability in handling documents with high-dimensional input space and culls out most of the irrelevant features. The good generalization characteristic of SVM is due to the implementation of SRM which entails finding an optimal hyper-plane, thus guaranteeing the lowest classification error. Besides, a capacity which is independent of the dimensionality of the feature space makes SVM a highly accurate classifier in most applications (Joachims, 1998). However, the major drawback of SVM is its relatively complex training and categorizing algorithms and also the high time and memory consumptions during the training stage and classifying stage due to its convoluted training and categorizing algorithms (Chakrabarti *et al.*, 2003). Besides, confusions occur during the classification tasks because the documents could be annotated to several categories because of similarities are typically calculated individually for each category (Brucher *et al.*, 2002).

SVM maps input vectors to a higher dimensional vector space where an optimal separating hyper plane is constructed. The discussion here starts with the simplest case with linear SVM on two categories of separable data. In a linear classifier, two groups of separable data can be divided by a hyper plane. In Fig. 1, white dots represent data points from one category while black dots represent data points from another category. As illustrated in Fig. 1, there are many possible hyper planes that can separate different types of data. However, there is only one hyper plane that maximizes the distance between itself and the nearest data vectors of each category. This hyper plane

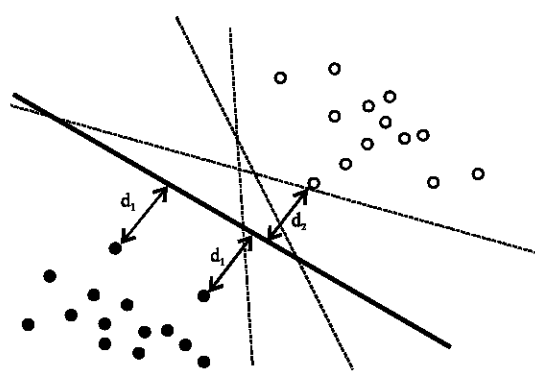


Fig. 1: Optimal separating hyper plane in a linear SVM (white dots and black dots represent data points from two different categories, respectively)

which maximizes the margin is called the optimal separating hyper plane and the margin is defined as the sum of distances of the hyper plane to the closest training vectors of each category.

The hyper planes that separate two groups of data can be expressed by Eq. 1.

$$w \cdot x + b = 0 \tag{1}$$

In the equation above, x represents a set of training vectors while w represents vectors perpendicular to the separating hyper plane and b represents the offset parameter which allows the increase of the margin. There are actually an infinite number of hyper planes that could separate data in a vector space into two groups, as represented by dashed lines illustrated in Fig. 1. According to SVM principle, there will just be one optimal hyper plane which leans half-way in between the maximal margin. The solid line illustrated in Fig. 1 represents this optimal separating hyper plane and the margin in this case is $d_1 + d_2$ (Gutschoven and Verlinde, 2000).

If the training data vectors are linearly separable, we can select the optimal separating hyper plane from the infinite number of hyper planes so that there are no vectors between them and maximize the margin. The vectors that constrain the width of the margin are called the support vectors. According to SVM principle, only support vectors are considered in determining the optimal separating hyper plane. Therefore, there is a way to represent the support vectors for a given set of training vectors. The illustration of the optimal separating hyper plane, two parallel hyper planes closest to the support vectors and the support vectors in the vector space are shown in Fig. 2. In Fig. 2, o symbols represent data points from one category while x symbols represent data points from another category.

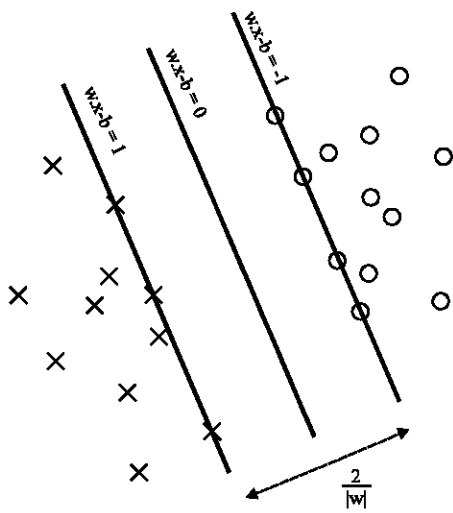


Fig. 2: Illustration of optimal separating hyper plane, hyper planes and support vectors (o symbols and x symbols represent data points from two different categories, respectively)

By using geometry, the distance between support vectors for each category is found as $2/|w|$. As to maximize the margin, $\{1/2 \|w\|^2\}$ need to be minimized first (Haykin, 1999). Thus, the optimal separating hyper plane can be configured by minimizing $\{1/2 \|w\|^2\}$, under the constraint that the training data is correctly separated, as illustrated in Eq. 2:

$$\min \{1/2 \|w\|^2\}, \text{ subject to } y_i \cdot (w \cdot x_i + b) \geq 1, \forall i \quad (2)$$

The discussion above describes that linear SVM on two categories of separable data is a general and the simplest algorithm of SVM classification approach. SVM classification approach may be more complex with cases such as multi-dimensional classification, non-linear separable data vectors and non-linear classifications. For n-dimensional classification by implementing SVM principle, as additional attributes are added, data vectors can be represented in n-dimensional space by using n-1 dimensional hyper planes to separate the data vectors into n categories (Haykin, 1999). In the case of non-linear separable data vectors in a SVM classification task, in order to allow some flexibility in separating the categories, SVM models introduce positive slack variables ξ_i in the constraints as illustrate in Eq. 3:

$$y_i \cdot (w \cdot x_i + b) \geq 1 - \xi_i, \forall i \quad (3)$$

this then becomes Eq. 4:

$$y_i \cdot (w \cdot x_i + b) \geq 1 - \xi_i, \forall i \quad (4)$$

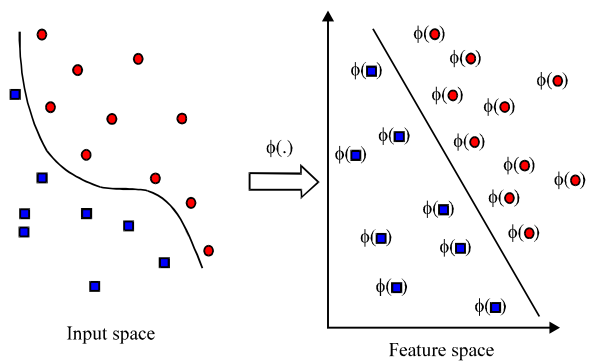


Fig. 3: Mapping non linear input space onto high dimensional space (squares and dots represent data points from two different categories, respectively)

The positive slack variables ξ_i controls the trade off between allowing training errors and forcing rigid margins. It creates a soft margin that permits some misclassifications. The increase the value of ξ_i raises the cost of misclassifying vectors and forces the creation of a more accurate model that may not generalize well.

According to SVM principle, non-linear classification can be handled by introducing kernel functions to map the data vectors into a different space where a hyper plane can be used to do the separation. Kernel function is used when decision function is not a linear function of the data and the data will be mapped from the input space into a high dimensional space through a non-linear transformation rather than fitting non-linear curves to the vector space to separate the data. As the result, a hyper plane can be used to do the separation in the high dimensional space as illustrated in Fig. 3. In Fig. 3, blue squares represent data points from one category while red dots represent data points from another category.

The algorithm of the non-linear classification is formally similar to the linear classification, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum margin hyper plane in the transformed high dimensional space. Therefore, although the classifier is a hyper plane in the high dimensional space, it may be non-linear in the original input space, as illustrated in Fig. 3.

Mapping the data vectors into a high dimensional space by using kernel functions contributes to a superb performing classification task since it allows SVM models to perform data vectors separation even with very complex boundaries. Kernel functions can be in an infinite of number but only certain kernel functions have been found to perform well classification tasks in a wide variety of applications. Some common well performing kernel functions in most cases are:

- Polynomial (homogeneous):

$$k(x, x') = (x \cdot x')^d \quad (5)$$

- Polynomial (inhomogeneous):

$$k(x, x') = (x \cdot x' + 1)^d \quad (6)$$

- Radial basis function:

$$k(x, x') = \exp(-\gamma \|x - x'\|^2), \text{ for } \gamma > 0 \quad (7)$$

- Gaussian Radial basis function:

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (8)$$

- Sigmoid:

$$k(x, x') = \tanh(kx \cdot x' + c) \quad (9)$$

for some (not every) $k > 0$ and $c < 0$

Each of the kernel function listed above has its own properties and unique responds in handling different kinds of data. SVM model using a sigmoid kernel function is equivalent to a two-layer perceptron neural network (Burges, 1998). SVM model with Radial Basis Function kernel is closely related to radial basis function neural networks and the feature space is in an infinite dimension. Therefore, in order to perform optimal classification tasks with SVM model, a good kernel function is required by selecting proper and optimal kernel function based on the classification task's requirements. With an optimal kernel function implemented in SVM model, the classification task is able to scale high dimensional data relatively well, tradeoff between classifier complexity and classification error can be controlled explicitly.

k-nearest neighbor classification approach: k-nearest neighbor (KNN) classification method is an instant-based learning algorithm that categorized objects based on closest feature space in the training set (Han *et al.*, 1999; Osuna, 2002). The training data is mapped into multi-dimensional feature space. The feature space is partitioned into regions based on the category of the training set. A point in the feature space is assigned to a particular category if it is the most frequent category among the k nearest training data. During the classifying stage, KNN classification approach finds the k closest labeled training samples for an unlabeled input sample and assigns the input sample to the category that appears most frequently within the k subset. As KNN outperforms

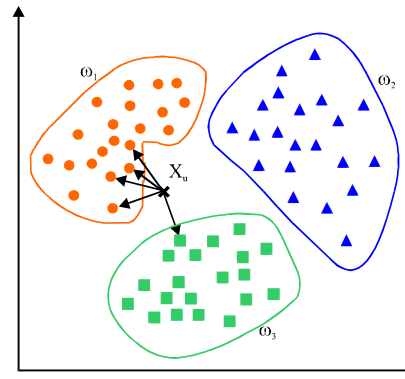


Fig. 4: Feature space of a 3-dimensional KNN classifier (dots, blue triangles and squares represent data points from ω_1 , ω_2 and ω_3 , respectively; X_u represents an unlabeled input sample) (Osuna, 2002)

the other classification approaches by its simplicity, it only requires a small training set with small number of training samples, an integer which specifies the variable of k and a metric to measure closeness (Osuna, 2002). Figure 4 shows an example of the feature space of a KNN classifier with three categories, where ω_1 , ω_2 and ω_3 represent three different categories with associated training samples and dots, blue triangles and squares represent data points from ω_1 , ω_2 and ω_3 , respectively. X_u represents an unlabeled input sample to be classified (Osuna, 2002).

Euclidean distance is typically used in computing the distance between the vectors. The key element of this method is the availability of a similarity measure for identifying neighbors of a particular document (Han *et al.*, 1999). The training phase consists only of storing the feature vectors and categories of the training set. In the classifying phase, distances between the input sample and all stored vectors are computed and k closest samples are selected. The annotated category of an input sample is predicted based on the nearest point which has been assigned to a particular category. In the case of N-dimensions, the Euclidean distance between two points, p and q is computed by using the formula as illustrated in Eq. 10.

$$\sqrt{(\sum_{i=1}^N (p_i - q_i)^2)} \quad (10)$$

where, p_i (or q_i) is the coordinate of p (or q) in dimension i.

Based on the example as shown in Fig. 4, the classification task is to annotate an input sample, X_u to its right category. In this case, the value of k has been

assigned as 5 and Euclidean distance is used to compute the distance between X_u to the training samples in the feature space. Of the 5 closest neighbors, 4 belong to ω_1 and 1 belongs to ω_3 . As the result of KNN classification, X_u is annotated to the category of ω_1 , which is the predominant category (Osuna, 2002).

KNN classification approach is outstanding with its simplicity. It performs well even in handling the classification tasks with multi-categorized documents. The major drawback of KNN is that it uses all features in distance computation and causes the method computationally intensive, especially when the size of training set grows. Besides this, the accuracy of k-nearest neighbor classification is severely degraded by the presence of noisy or irrelevant features, especially when the number of attributes grows.

As KNN classification is considered as an instance-based learning, or lazy learning approach, the algorithm suspends the processes of training data until it receives a command to classify an input sample. When the classifier receives an input sample to be classified, it compiles all the training samples in order to reply to the classification request, then abandons the constructed answer and any immediate result (Osuna, 2002). On the other hand, the eager learning algorithms, such as artificial neural networks, build the classification model during training stage by compiling the training data and discard the training data after the classification model has been built. The actuated model is then used to classify the input sample and it is retained for the classification requests in the future (Osuna, 2002). As the nature for an instance-based learning approach, the KNN classification approach consumes less processing time than the eager learning algorithms during training stage. However, KNN classification approach requires greater processor and physical memory usages and higher time consumption during the classifying stage.

There exist two classical enhancing algorithms for KNN classification approach to reduce the time consumption and the processor and physical memory usages during the stage of classifying, the bucketing algorithm and the k -dimensional trees algorithm (Osuna, 2002). The bucketing algorithm was introduced by Welch in 1971 and the k -dimensional trees algorithm was introduced by Osuna (2002).

In the bucketing algorithm, in order to improve the efficiency of the conventional KNN classification approach, the feature space is segmented into identical cells, according to their category. Each individual cell contains training data points and the training data points are recorded and stored in a list (Osuna, 2002). During the

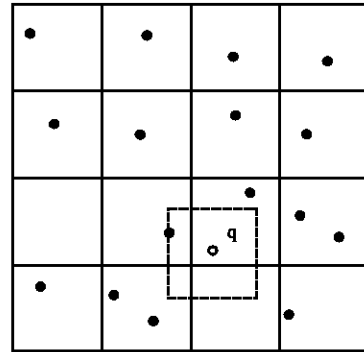


Fig. 5: Bucketing algorithm for KNN classification approach (Osuna, 2002)

classifying stage, the distance between the internal data points of each cell and the input data point is computed. The classifying process terminates when the distance between the input data point and the cell which is currently examined, is larger than the distance between the input data point to the cell which has been examined (Osuna, 2002). The classification result is obtained based on the cell which has the shortest distance to the input data point (Osuna, 2002). Figure 5 shows an illustration for an example of the bucketing algorithm.

The k -dimensional trees algorithm for KNN classification algorithm is a binary search tree which has been generalized in high dimensions (Osuna, 2002). It is a space-partitioning data structure to organize data points into a k -dimensional feature space. The training stage starts with a partitioning process to partition the nodes in a k -dimensional tree in order to segment the multi-dimensional feature space according to the underlying distribution of the training data (Osuna, 2002). Each of the nodes of a k -dimensional tree is associated with a hyper-rectangle and a hyper-plane which are orthogonal to one of the coordinate axis (Osuna, 2002). The hyper-rectangle of each node is divided by the hyper-plane into two parts and these two parts of the hyper-rectangle are associated with the successors of the node. The training stage terminates when the number of training data points in the hyper-rectangle is less than a given threshold (Osuna, 2002). During the classifying stage, an input data point is mapped into one of the nodes in the k -dimensional tree. The algorithm searches the tree in descending to find all the training data points which included in the node that contains the input data point (Osuna, 2002). Afterwards the algorithm examines the surrounding nodes to determine whether they overlap the ball centered at the input data point. The classification result is obtained based on the label of the training data point which has the shortest distance to the input data point (Osuna, 2002).

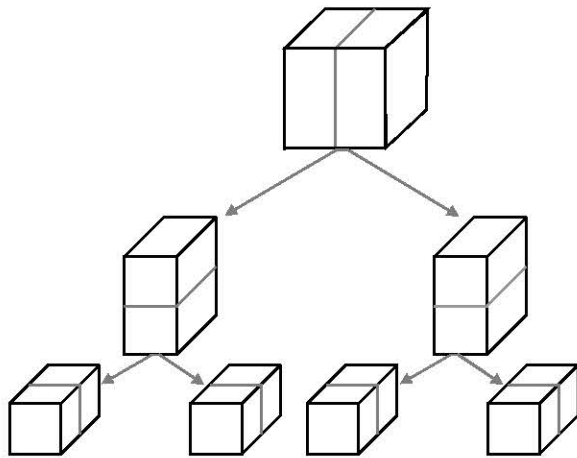


Fig. 6: Example of the data structure of a k-dimensional tree (3-D view) (Osuna, 2002)

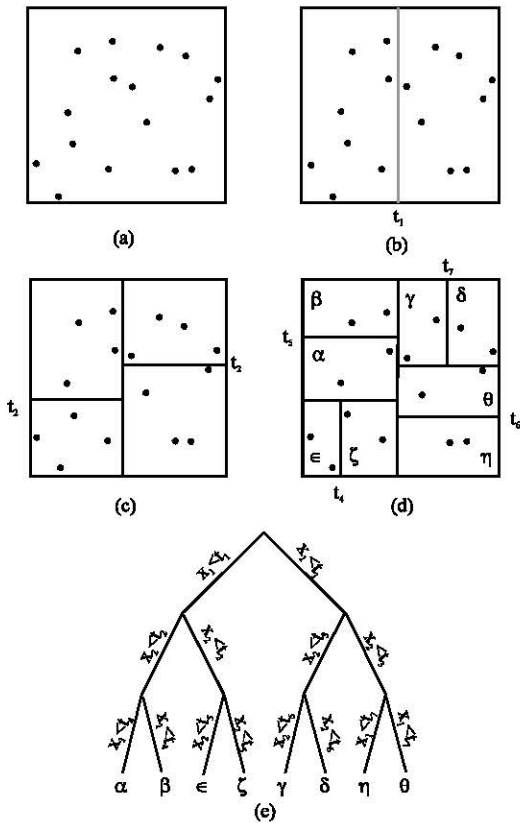


Fig. 7: (a-e) Example of a k-dimensional tree decomposition (2-D view) (Osuna, 2002)

Figure 6 shows an example of the data structure of k-dimensional tree in 3-dimensional view, while Fig. 7a-e show an example of k-dimensional tree decomposition after the partitioning process in 2-dimensional view (Osuna, 2002).

NEAREST NEIGHBOR-SUPPORT VECTOR MACHINES HYBRID CLASSIFICATION MODELS

SVM-KNN: Discriminative nearest neighbor classification for visual category recognition by Zhang *et al.* (2006): Zhang *et al.* (2006) proposed an integration of nearest neighbor classifier and support vector machine to classify visual objects, coined as SVM-KNN. The proposed SVM-KNN hybrid classification model can be used for large and multiclass datasets with reasonably lower computational complexity both in training and classifying stage. This property of lower computational complexity is inherited from KNN classification approach which does not need construction of a feature space and it is able to handle and manage multiclass datasets with high dimensionality. Due to this fact, the proposed SVM-KNN model is able to provide a more flexible framework from a computer vision perspective as the emphasis is on similarity of objects (Zhang *et al.*, 2006).

NN algorithm has been used by SVM-KNN hybrid classification approach as the initial process of its classification task. In the situation that there is no conflict in the classification of a given input data, where all k neighbors are from a same class, a straightforward classification task is performed by the simple NN algorithm without needing the involvement of SVM approach. On the other hand, if the k neighbors are not with the same class, the hybrid classification approach is used to perform the classification task (Zhang *et al.*, 2006). In the proposed SVM-KNN hybrid classification approach, Nearest Neighbor (NN) algorithm is utilized to prune the original training set, so that SVM which acts as a classification engine of this hybrid model does not need to be train with the original training set, but with a pruned training set which insufficient training samples have been eliminated. This is to ensure a faster training process for SVM with a smaller training set, without sacrificing the sufficiency of classification performance. For the pruned training set, pair-wise distances between the training samples are computed and the distance matrix will be converted to kernel matrix as the training data for SVM (Zhang *et al.*, 2006). The general idea of this work is to train the SVM until it preserves the distance function on the set of neighbors and to find close neighbors to an input sample. Algorithm of the proposed SVM-KNN has been summarized in Table 1. While performing the task of classifying visual objects, SVM-KNN focuses on two major elements of visual object recognition, shape and texture. Several successful distance functions have been used in this study, such as X^2 distance, marginal distance,

Table 1: Steps of KNN-SVM algorithm proposed by Zhang *et al.* (2006)

1	Calculate the distance of an input data to all the training samples and select the k nearest neighbors
2	If all k neighbors of the input data are under a same category, the input data is assigned with that particular category
3	If not, calculate the pair-wise distances between the k neighbors
4	Distance matrix is converted into kernel matrix as the input for SVM
5	The input document is assigned with the category resulted from the SVM classification

Table 2: Summary of experimental results of SVM-KNN hybrid classification model on benchmark datasets error rate for each distance function

Dataset	Error rate					
	MNIST		USPS		CURET	Caltech-101
Distance function	L_2	SC	L_2	Tangent Distance	X^2	Algo. A
SVM-KNN	1.66	1.67	4.285	2.59	1.73	40.92
NN	2.81	2.2	5.53	2.89	2.53	59.02
DAGSVM	Not tested	Not tested	4.4	Intractable	1.75	43.60
HKNN	Not tested	Not tested	3.93	N/A	Not tested	Not tested

tangent distance, shape context based distance, geometric blur based distance (L_2) and kernelizing the distance (Zhang *et al.*, 2006).

According to the authors, their research is very similar to local learning by Bottou and Vapnik which using the same idea but difference distance formula (Zhang *et al.*, 2006). The experiments and evaluations on the proposed SVM-KNN hybrid classification model using some standard benchmark datasets indicated that SVM-KNN hybrid model performed relatively well on MNIST, USPS, CURET and Caltech-101 datasets, as compared to SVM and KNN classifiers alone, as well as DAGSVM and HKNN classifier (Zhang *et al.*, 2006). In MNIST dataset, it contains of 60,000 training samples and 10,000 testing samples. During the testing stage, Geometric blur based distance, L_2 and shape context distance have been used to compare SVM-KNN with NN alone. USPS dataset consists of 7291 training samples and 2007 testing samples. Two types of distance, L_2 and tangent distance are used in comparison between SVM-KNN, NN, DAGSVM and HKNN. For CURET dataset, training data and testing data are chosen randomly from 61 real world textures. Based on the experiments conducted on the datasets mentioned above, the results show that SVM-KNN has obtained the higher classification accuracy as compared to NN and DAGSVM.

The Caltech-101 is the most challenging dataset among all the tested datasets because it contains many poses, lighting and variations of colors (Zhang *et al.*, 2006). Formula for distance calculation used in Caltech-101 is defined as Eq. 11 and 12:

$$D^A(I^L \rightarrow I^R) = \frac{1}{m} \sum_{i=1}^m \min_{j=1-n} \|F_i^L - F_j^R\|^2 \quad (11)$$

$$D^A(I^L, I^R) = D^A(I_L \rightarrow I_R) + D^A(I_R \rightarrow I_L) + \lambda \sum_{i=1}^{min} \|h_k^L - h_k^R\| L1 \quad (12)$$

With refer to Eq. 10 and 11, $D^A(I^L, I^R)$ is the distance between left and right image. F_i^L represents the i 'th feature of the left image and F_i^R is the i 'th feature of the right image. Both F_i^L and F_i^R are computed from geometric blur features. h_k^L stands for left image output for k 's filter, respectively for h_k^R (Zhang *et al.*, 2006). From the experimental results, the authors found out that their proposed SVM-KNN hybrid classification model has higher classification accuracy as compared to NN and DAGSVM.

The experimental results illustrated in Table 2 show that SVM-KNN hybrid model has the lowest error rates compared to the classification methods mentioned above. Figure 8 shows that the proposed SVM-KNN hybrid classification model has a trade-off between time consumption and classification error rate, where the classification error rate is high when the time consumption is low and vice versa. According to the researchers of this work, SVM-KNN hybrid model can overcome the drawback of conventional SVM which is less efficient especially when the problem domain becomes intractable (Zhang *et al.*, 2006).

Gene expression data classification using SVM-KNN classifier by Shen and Lin (2004): Shen and Lin (2004) presented a new binary classifier that combines Support Vector Machines (SVM) with k-nearest neighbor (KNN) algorithm for gene expression data classification in order to improve disease diagnosis. Conventional SVM classification produces lower classification accuracy if the sample data of the two classes in a binary classification are all close to the separating hyper plane. When conventional KNN classification has been applied to perform gene expression data classification, KNN distance functions are anticipated to have poor classification performance as the dimensionality of the noisy data increases (Shen and Lin, 2004).

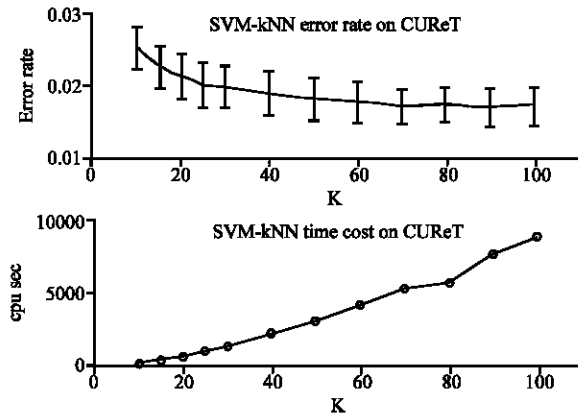


Fig. 8: Trade-off between speed and accuracy of SVM-KNN in the case of texture classification (Zhang *et al.*, 2006)

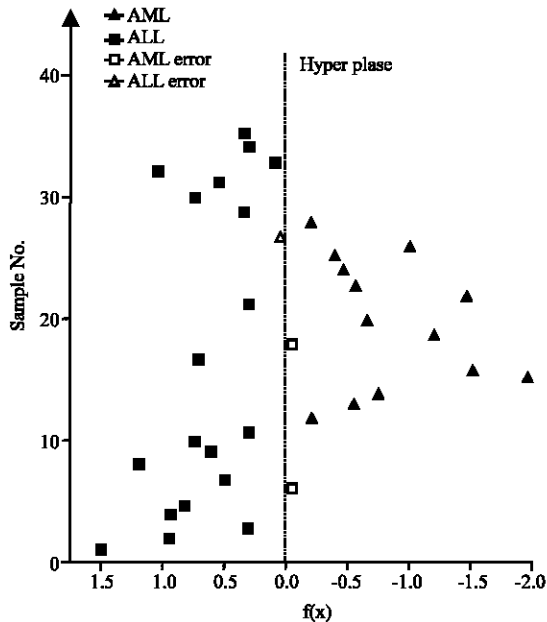


Fig. 9: Feature space with training samples of SVM with three misclassified samples (two white rectangles and one white triangle) (Shen and Lin, 2004)

The KNN and SVM hybrid classification system proposed in this work, coined as KSVM, is developed due to the fact that conventional SVM is not able to handle the situation when the training samples are close to the optimal separating hyper plane. In such a situation, misplacing of training samples into another class could happen, as illustrated in Fig. 9. In Fig. 9, black triangles represent data points from category AML while black rectangles represent data points from category ALL and the two white rectangles and the white triangle are the data points which have been misclassified.

Table 3: Steps for KSVM algorithm proposed by Shen and Lin (2004)

- 1 SVM is considered as a 1NN classifier in which only one representative point is selected for each class and is used to get support vectors and its factor, constant b
- 2 In the classification phase, the algorithm calculates the distance from the input document (sample) to the optimal separating hyper-plane of SVM in the feature space
- 3 If the distance is greater that a given threshold, the input document is assigned with the category resulted from the SVM; otherwise, the KNN algorithm is used

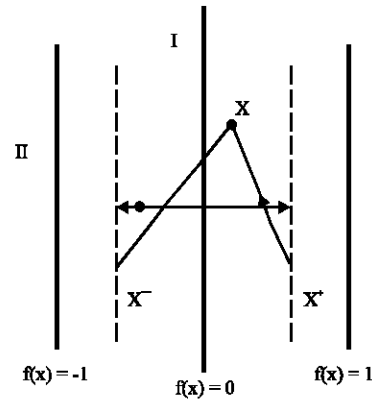


Fig. 10: Distance from sample x to hyper plane determines which algorithm should be used to classify x. (Shen and Lin, 2004)

During the classification phase, the KSVM algorithm calculates the distance from the testing sample to the optimal separating hyper plane in feature space. NN classification is a distance based approach in which Euclidean distance is typically used in order to calculate the distance between sample data. However, in this proposed KSVM method, distance between sample data is computed using Eq. 13 as illustrated below:

$$d(x, x_i) \| \phi(x) - \phi(x_i) \|^2 = k(x, x) - 2k(x, x_i) + k(x_i, x_i) \quad (13)$$

According to the algorithm of KSVM approach, should the distance be greater than a given threshold x , the input sample data will be classified using SVM, otherwise classification is performed by using KNN. The classification algorithm of the proposed KSVM approach is as shown in Table 3 and Fig. 10.

Shen and Lin (2004) have compared the performance of KSVM with KNN and SVM individually in the experiments conducted in this research work. Leukemia dataset and Colon dataset have been used to conduct the experiments in order to evaluate the performance of their proposed KSVM classification approach. Leukemia dataset consists of 38 training data and 35 testing data. On the other hand, Colon data set has 31 samples, for both training set and testing set. According to Shen and Lin (2004), there exist two feature selection methods for

Table 4: Absolute number of errors of the various methods (Shen and Lin, 2004)

Data sets	No. of samples	Method	Errors		
			Class+1	Class-1	Total
Data set (A) Leukemia	49	KNN	1	3	4
		SVM	1	2	3
		KSVM	0	1	1
	999	KNN	1	1	2
		SVM	0	1	1
		KSVM	0	0	0
7129	KNN	1	1	2	
	SVM	1	1	2	
	KSVM	0	0	0	
Data set (B) Colon	49	KNN	3	2	5
		SVM	2	2	4
		KSVM	1	1	2
	999	KNN	3	1	4
		SVM	1	1	2
		KSVM	0	0	0
	2000	KNN	2	1	3
		SVM	2	2	4
		KSVM	1	0	1

gene selection problem, which are signal-to-noise (S2N) and Recursive Feature Elimination (RFE) (Shen and Lin, 2004). In the experiments conducted in this work, they have selected S2N feature selection technique and have computed the following statistics as shown in Eq. 14:

$$S(j) = \frac{\mu_+(j) - \mu_-(j)}{\sigma_+(j) - \sigma_-(j)} \quad (14)$$

The symbol $\sigma_+(j)$ and $\sigma_-(j)$ represent the standard deviation for the two classes for the j th gene. j th gene of classes +1 and -1 are represented by $\mu_+(j)$ and $\mu_-(j)$, respectively. The experimental results obtained in this research work are illustrated in Table 4.

According to the experimental results obtained by Shen and Lin (2004), KSVM classifier has produced a higher accuracy as compared to conventional SVM and conventional KNN. Based on their findings, the better classification performance of KSVM classifier then conventional SVM classifier is due to the fact that more information is carried after the training process is performed, as compared to SVM. Besides this, the number of genes used in the training process will not have high effect to KSVM classifier than the two latter classifiers (Shen and Lin, 2004).

Hybridized KNN and SVM for gene expression data classification by Mei et al. (2009): Mei et al. (2009) proposed a hybridized algorithm that uses both k-nearest neighbor (KNN) and Support Vector Machine (SVM) classifiers, coined as HKNNNSVM, to analyze gene expression for cancer classification, in both binary and multi-class categorization. They mentioned that precise diagnosis and classification is the ultimate goal for

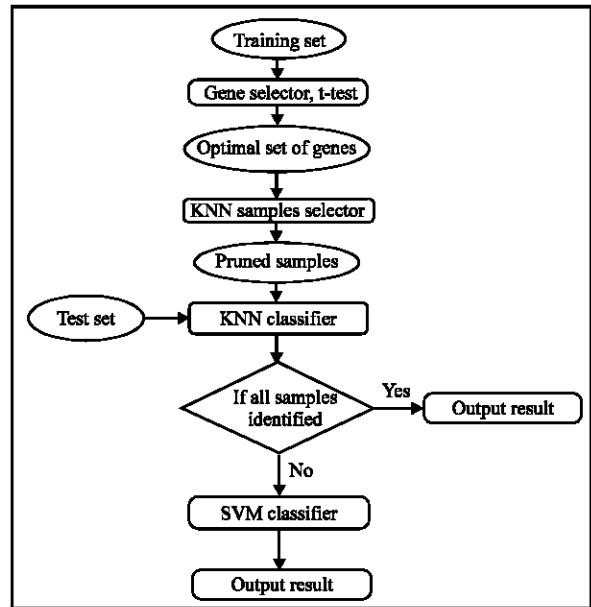


Fig. 11: The chart of HKNNNSVM scheme (Mei et al., 2009)

Table 5: Steps of HKNNNSVM hybrid algorithm in training stage

- 1 Choose the train data set using gene ranking method, t-test
- 2 First KNN used to filter and prune training sample. If label of data set is not same as majority of KNN; pass to step 3 else; labeled as same as KNN label
- 3 Second KNN used to classify. If still unlabeled, pass to step 4
- 4 SVM used to classify unlabeled sample prune from step 1-3

Table 6: Steps of HKNNNSVM hybrid algorithm in testing stage

- 1 KNN used to classify test data. If still unlabeled, pass to step 2
- 2 SVM used to classify unlabeled sample

successful treatment of illness. The proposed HKNNNSVM is developed in order to improve the conventional SVM performance and to solve the problem of overlapped boundary region (Mei et al., 2009).

The flow chart of the proposed HKNNNSVM algorithm is depicted in Fig. 11. KNN is used as the preliminary classifier to prune the training samples and eliminate those insufficient samples effectively in order to reduce the training size for better classification. The second KNN classifier, coupled with SVM, is used to classify the input samples based on the remaining training samples which are not pruned away from the first KNN. Both KNN classifiers use Euclidean distance formula to calculate the distances between each pair of samples (Mei et al., 2009). The steps for training process of HKNNNSVM are as listed in Table 5 and the steps for classifying process are shown in Table 6.

KNN algorithm is used to reduce the number of training samples by computing the distance matrix

(symmetrical matrix containing the Euclidean distance between each pair of samples). Then, the k nearest neighbors for each sample is sought. If the category of a training sample is same as the label of the majority of its k nearest neighbors, the training sample is reserved, else it will be eliminated. As the outcome of the pruning process, a reduced training set is obtained and it is then used to train the KNN classifier for further classification. During the classifying stage performed by the KNN classifier, if the k neighbors of the input sample are all in the same category, then the input sample is assigned to that particular category. On the other hand, if all the k neighbors of the input sample are not from the same category, the hybrid classification approach, HKNNNSVM will calculate the pair-wise distances between the k neighbors and use the computed distance matrix as the input to the SVM. After the classification process performed by SVM, the input sample is assigned with the category resulted from the SVM classification.

Three benchmark datasets for gene expression classification, which are Colon dataset, Estrogen dataset and Acute Lymphoblastic Leukemia dataset, have been used by the authors in order to conduct the experiments for evaluating their proposed HKNNNSVM classification approach. Colon dataset consists of 50 training samples and 12 testing samples, meanwhile Estrogen dataset has 40 training samples and 20 testing samples. For Acute Lymphoblastic Leukemia dataset, the authors have selected 208 training samples and 40 testing samples. Based on their experimental results, the authors claimed that HKNNNSVM has obtained the lowest classification error rate as compared to SVM, KNN and NN-SVM when they are used independently from one another (Mei *et al.*, 2009). Table 7 summarized all the experimental results of each dataset that contained mislabeled samples. Misclassification rate for the testing set was found to be lower with samples pruning used than that of KNN and SVM, which indicated the stability of the classification performance of HKNNNSVM algorithm. From the experimental results, the authors conclude that, HKNNNSVM can perform better than SVM, KNN and

NN-SVM, by having the prune function which can eliminate the mislabeled training samples effectively.

Instance-based spam filtering using SVM nearest neighbor classifier by Blanzieri and Bryl (2007a): Enrico Blanzieri and Anton Bryl had presented a hybrid classification model, coined as SVM-NN, which combines Support Vector Machines (SVM) classification approach and Nearest Neighbor (NN) classification approach and had conducted an evaluation on the performance of SVM-NN in handling the task of spam filtering. The proposed SVM-NN classifier is the combination of SVM-based decision rule and k-nearest neighbor (KNN) classifiers. The aim of this research work is to evaluate the performance of SVM-NN algorithm in performing unsolicited bulk email filtering, as spam has become one of the serious problems of Internet since some decades now and spam has no proper uniformity but contains messages that differ greatly in terms of topic, format and content (Blanzieri and Bryl, 2007a).

The proposed SVM-NN hybrid classification model leads to a bigger margin, hence results a lower generalization error bound. The dataset is first classified using KNN classification approach. If the dataset cannot be categorized, dataset will then be classified using SVM. In the experiment, Euclidean metric is used to determine the nearest neighbors and the linear kernel for SVM. The steps for this hybrid algorithm are shown in Table 8.

In order to classify the sample, SVM-NN hybrid classifier finds out the k samples in the training set and SVM classifier is trained based on these k samples. During the classifying stage, the trained SVM classifier is used to classify the unlabeled input sample. The combination of SVM and KNN is able to obtain a lower generalization error bound, produced by the combination of a greater margin and smaller ball containing the points (Blanzieri and Bryl, 2007a). A comparison experiment is conducted in order to evaluate the performance of the proposed SVM-NN classifier, the conventional KNN classifier and also the conventional SVM classifier. The experiments are carried out in the way of ten-fold

Table 7: Results of misclassification rates for dataset contained mislabeled samples (Mei *et al.*, 2009)

Datasets		Classification models			
		HKNNNSVM	SVM	KNN	NN-SVM
Colon	No. of genes	200	500	50	500
	Training set	3.74%	28.7%	19.4%	26.7%
	Test Set	10.78%	20.61%	12.86%	18.92%
Estrogen	No. of genes	50	100	50	100
	Training set	5.42%	23.62%	18.5%	17.13%
	Test set	9.74%	19.0%	15.41%	18.26%
ALL	No. of genes	200	500	100	500
	Training set	0.56%	2.48%	4.45%	4.16%
	Test set	2.03%	3.07%	3.53%	4.08%

Table 8: Steps of hybrid SVM-KNN algorithms (Blanzieri and Bryl, 2007a)

Algorithm: SVM nearest neighbor
Inputs: sample x to classify; training set $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$;
 number of nearest neighbors k ; threshold t .
Output: decision $y \in \{-1, 1\}$

Find k samples (x_i, y_i) with minimal values of $K(x_i, x) - 2 * K(x_i, x)$
 Train SVM model on the k selected samples.
 Classify x using SVM, get the result in the form of a real number.
 Make the decision using the threshold t

cross-validation on the SpamAssassin corpus, which is a reproduction of one of the experiments by another research group (Blanzieri and Bryl, 2007a). The SpamAssassin corpus contains 4150 legitimate and 1897 spam messages. Two variants of the proposed hybrid classification model were involved in the comparison, which are SVM-NN (Full scale) and SVM-NN (Partial scale). Total Cost Ratio (TCR) is used for parameters optimizations and it is defined as Eq. 15:

$$TCR = nS / (\lambda \cdot nL \rightarrow S + nS \rightarrow L) \quad (15)$$

where, nS represents the total number of spam messages in the corpus. $nL \rightarrow S$ is the number of spam messages but classified as legitimate message and $nS \rightarrow L$ is the reverse way. Relative cost of misclassification is represented by λ .

From the experiments conducted by the authors in this work, we can find that SVM-NN generally performs better than conventional SVM. SVM-NN outperforms SVM in terms of error rates by having lower rate on false negatives rate (FN), as well as false positives rate (FP). However, SVM-NN only outperforms SVM significantly on low dimensional features space, but the outperformance is not significant on higher dimensional features space. The relation of FN and the dimensionality of feature space are shown in Fig. 12 while Fig. 13 illustrated the relation of FP and the dimensionality of feature space. Based on the experimental results, the authors conclude that the major drawback of SVM-NN is its high resources consumption, thus results low processing speed and costly classification process, especially when the numbers of k increase. In high dimensional features space, the outperformance of SVM-NN compared to SVM is less significant, but it persists. This might due to the fact that the SVM-NN approach has higher sensitivity to unrelated features, as compared to SVM (Blanzieri and Bryl, 2007a).

Evaluation of the highest probability SVM nearest neighbor classifier with variable relative error cost by Blanzieri and Bryl (2007b): Blanzieri and Bryl (2007b) had conducted research work to investigate the performance of combining two classifiers, which are

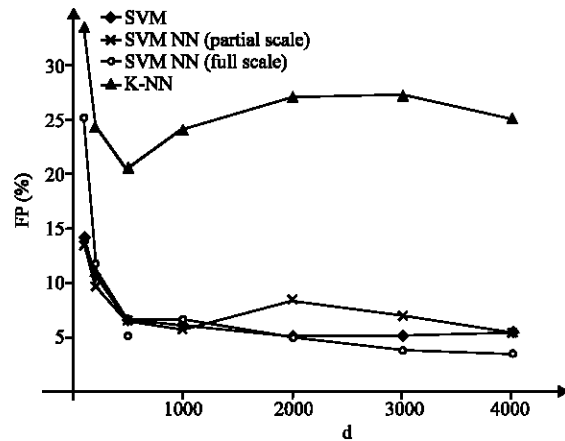


Fig. 12: Accuracy of false negative rate, FN versus number of feature, d (Blanzieri and Bryl, 2007a)

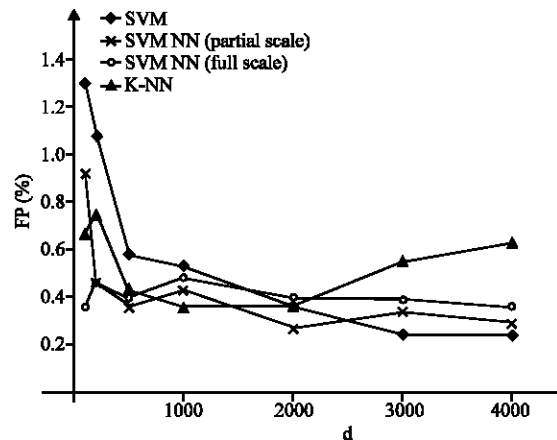


Fig. 13: Accuracy of false positive rate, FP versus number of feature, d (Blanzieri and Bryl, 2007a)

Support Vector Machines (SVM) and k -nearest neighbor (KNN) to perform the task of spam filtering. Unsolicited bulk email or E-mail spam has causing huge financial losses due to misuse of time, storage and resources to promote online scam or fraud and as well as advertised illegal goods. Hence, spam filtering is widely applied to overcome the issue to email spamming. According to Enrico Blanzieri and Anton Bryl, there exist some methods

Table 9: The pseudo code of Highest Probability-SVM-Nearest Neighbor classifier (Blanzieri and Bryl, 2007b)

Algorithm: The Highest Probability SVM Nearest Neighbor Classifier
Require: sample x to classify; training set $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
 set of possible values for the number of nearest neighbors $\{k_1, k_2, \dots, k_n\}$;
 parameter C for adjusting the balance between the two types of errors
Ensure: decision $y_p \in \{-1, 1\}$
 1: Order the training samples by the value of $K(x_i, x_i) - 2 * K(x_i, x)$ in ascending order.
 2: $MinErr = 1000$; $Value = 0$
 3: **if** the first k_1 values are all from the same class c **then**
 4: **return** c
 5: **end if**
 6: **for all** k **do**
 7: Train SVM model on the first k training samples in the ordered list.
 8: Classify x using the SVM model with equal error costs, get the result y'_p
 9: Classify the same training samples using this model.
 10: Fit the parameters A and B for the estimation of $P(y = 1 | y'_p)$
 11: $ErrorNegative = P(y = 1 | y'_p)$; $ErrorPositive = 1 - P(y = 1 | y'_p)$
 12: **if** $ErrorPositive < MinErr$ **then**
 13: $MinErr = ErrorPositive$; $Value = 1$
 14: **end if**
 15: **if** $ErrorNegative * C < MinErr$ **then**
 16: $MinErr = ErrorNegative * C$; $Value = -1$
 17: **end if**
 18: **end for**
 19: **return** $Value$

of spam filtering, such as SMTP path analysis, human language technologies, Naïve Bayes, SVM and KNN.

The authors have proposed the integration of highest probabilities-support vector machine-nearest neighbor, coined as HP-SVM-NN, as a classification method for spam filtering. This proposed classification approach is able to outperform the conventional SVM classifier and has achieved a lower generalization error bound as compared to the conventional SVM (Blanzieri and Bryl, 2007b). The authors combined the algorithms of SVM and KNN by first selecting the parameter k for each sample x from a pre-defined set in the neighborhood of the samples which must be classified. Then, the hybrid algorithm trains the SVM model on k nearest labeled samples and uses the model to perform classification on some given samples. This hybrid model classifies the input data using the following probabilities, $P(\text{legitimates}|x)$ and $P(\text{spam}|x) = 1 - p(\text{legitimate}|x)$. The output from SVM model will be used to estimate and calculate the probabilities. The highest probability value will be used to calculate error for the negative answer and error for the positive answer. The pseudo codes of HP-SVM-NN algorithm are as listed in Table 9.

In order to evaluate the performance of HP-SVM-NN hybrid classification approach, authors had used SpamAssasin corpus which consists of 4150 legitimate messages and 1897 spam messages. Messages are broken into message header and message body by using a feature extraction method (Blanzieri and Bryl, 2007b). Each of the messages will be represented by a vector of d -binary features. The highest information gain for

d -binary features will be selected by using formula as illustrated in Eq. 16:

$$\text{Information gain, IG}(fk) = \sum_{c \in \{c_1, c_2\}} \sum_{f \in \{fk, \bar{fk}\}} P(f, c) \times \log \frac{P(f, c)}{P(f) \times P(c)} \quad (16)$$

where, two classes of samples represented by c_1 and c_2 , meanwhile fk represents a binary feature.

From the experiments conducted by authors, the performance of HP-SVM-NN has been compared to the performance of SVM in terms of Receiver Operating Characteristic (ROC) curves. The comparison shows that HP-SVM-NN outperforms the conventional SVM, as illustrated in Fig. 14. However, the major drawback of the HP-SVM-NN classifier is its low processing speed due to the fact that it needs to calculate distances for each of the sample data, thus increasing the accuracy of the classification at the price of computational intensive (Blanzieri and Bryl, 2007b). The authors suggested building a filter by cascading the model with a faster algorithm such as Naïve Bayes to classify a message and HP-SVM-NN method is only used for final checking of those spam message labeled by Naïve Bayes (Blanzieri and Bryl, 2007b).

LDA/SVM driven nearest neighbor classification by Peng et al. (2003): Peng et al. (2003) proposed a locally adaptive neighborhood morphing classification method to reduce bias caused by Nearest Neighbor's (NN) finite samples due to the curse of high dimensionality. These severe partialities can happen in

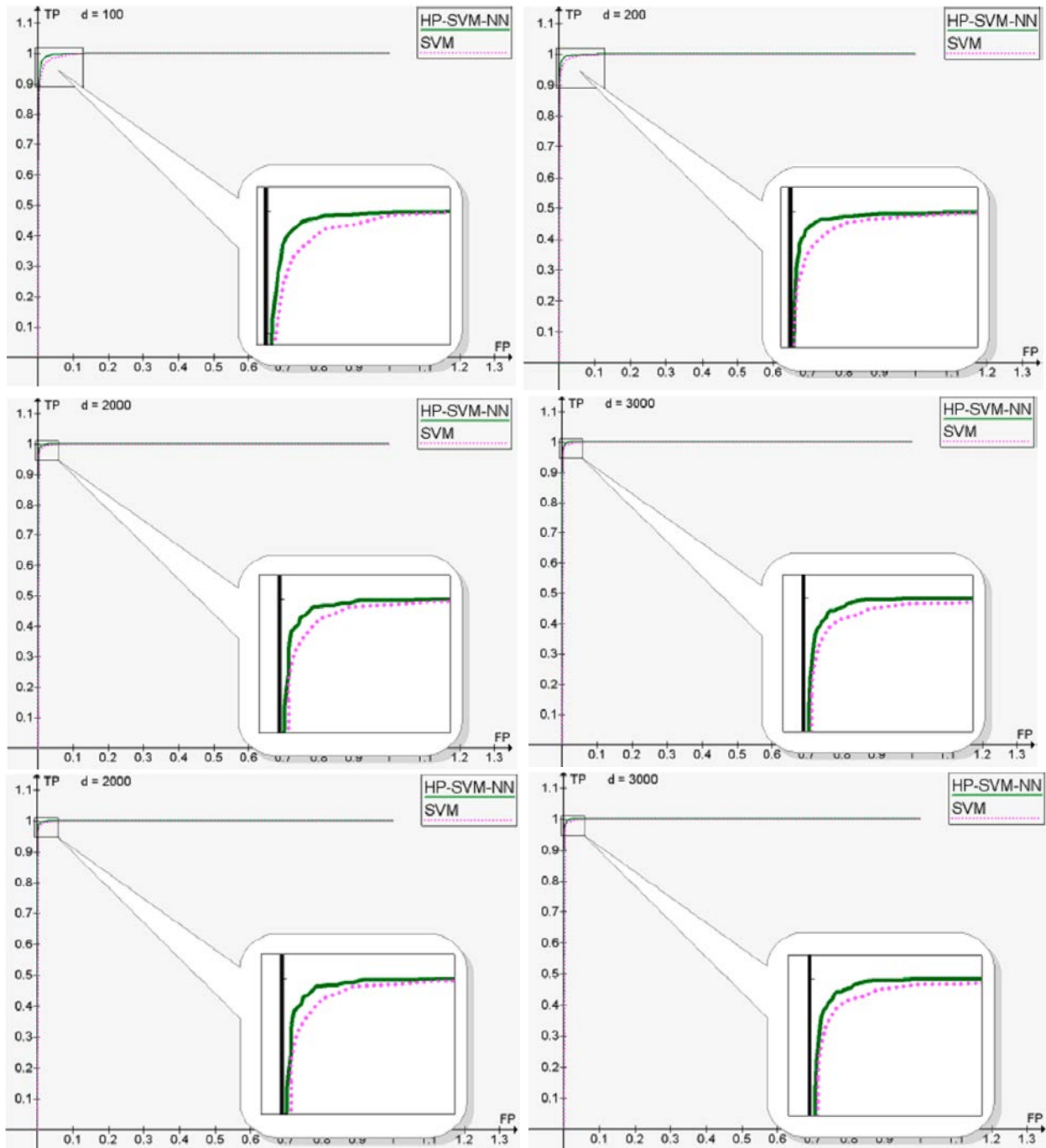


Fig. 14: Comparison of SVM and HP-SVM-NN in terms of ROC curves (Blanzieri and Bryl, 2007b)

the NN rule, which if not properly dealt with can strongly affect classification performance. Local Support Vector Machines (SVM) learning was applied to produce neighborhoods that are extended along less relevant feature spaces, particularly those along the most influential ones (Peng *et al.*, 2003). As the result, the

modification of the neighborhood will produce efficacy in classification performance, while still maintaining the stability of class conditional probabilities.

The neighborhood morphing NN algorithm, named MORF, has three stages, as described in Table 10. Firstly, it uses Euclidean distance function to compute a nearest

Table 10: Steps of neighborhood morphing NN algorithm (MORF)

- 1 Use NN points out the query x_0 and calculated the distance using Euclidean distance function
- 2 Linear SVM is built and to obtain exponential feature weighting scheme, w_i
- 3 To classify x_0 , formula $D(x_0, x) = \sqrt{\sum_{i=1}^n w_i (x_{0i} - x_i)^2}$ is used to compute K Nns

Table 11: Average classification error rates of competing classification schemes on different datasets (Peng *et al.*, 2003)

	SVM-R		MORF		LDAW		KNN		DANN		MACHETE		SCYTHER		C4.5	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Iris	4.9	3.11	4.6	2.88	5.4	2.88	4.9	3.58	6.4	3.83	6.0	4.21	4.8	2.95	8.3	3.54
Vote	3.7	1.78	3.5	1.96	7.6	3.79	8.4	2.35	3.9	1.90	5.4	2.12	5.4	1.65	3.5	1.60
Sonar	14.4	5.06	13.4	4.24	16.0	3.42	16.0	3.44	12.9	4.02	21.0	3.81	18.0	3.87	30.1	4.69
Ion	5.4	0.96	7.2	1.98	11.4	2.82	12.59	2.24	10.4	2.48	11.5	2.29	13.5	2.54	10.7	2.60
Liver	28.0	2.46	30.3	2.63	36.3	4.46	36.4	33.96	32.6	4.36	36.1	3.02	36.8	4.53	37.6	3.12
Hep	15.2	3.93	14.5	3.95	14.4	4.13	14.8	4.80	13.6	3.90	17.4	4.35	16.9	4.13	19.6	4.21
Cancer	2.98	0.62	2.9	0.89	3.2	0.89	3.1	0.91	2.8	0.78	3.6	1.04	3.2	0.95	4.0	1.31
Pima	23.5	2.41	24.6	2.57	26.6	2.91	27.1	3.35	26.4	2.47	25.7	3.00	25.7	3.00	19.1	1.33
OQ	3.1	1.35	4.3	2.11	6.1	2.16	6.4	2.04	4.5	1.88	8.0	1.69	6.3	2.01	3.5	0.81
Unstruct	29.6	2.70	7.0	5.92	26.1	11.78	34.0	3.56	30.0	3.39	9.0	2.25	13.6	3.07	10.1	7.40

neighborhood of points around the query. Then, a local linear SVM is developed to obtain the exponential feature weighting scheme. Finally, the resulting weighting scheme is used to compute nearest neighbors for classification (Peng *et al.*, 2003).

The authors derive a neighborhood algorithm morphing with SVM to produce a neighborhood algorithm that has a better performance than conventional SVM. The authors proposed using WSVM, where W denotes the within sum-of squares matrix, to compute feature relevance, r_i , with the formula as illustrated in Eq. 17:

$$r_i(x_0) = |w_{SVM} \cdot \Theta| \tag{17}$$

where, Θ is the direction whose inner product with WSVM.

The effectiveness of their proposed algorithm was compared against other classification techniques using ten different datasets, namely Iris, Vote, Sonar, Ion, Liver, Hep, Cancer, Pima, OQ and Unstruct. The competing classification methods, such as MORF, LDAW, SVM-R, KNN, C4.5, MACHETE, SCYTHER and DANN, are carefully chosen and compared to determine the usefulness of the proposed method. The experimental results are shown in Table 11 and Fig. 15 for performance distribution.

Based on the experimental results as illustrated in Table 11 and Fig. 15, the authors conclude that MORF classification model outperforms the other classifiers. The experimental results showed that KNN had the worst classification performance and MORF, a hybridized algorithm proposed in this work, had obtained the best result. MORF can minimize the bias of NN which assumption for class conditional probabilities are locally statics and constants (Peng *et al.*, 2003). While it is usually costly in building local linear SVMs for classification, the increase in classification performance

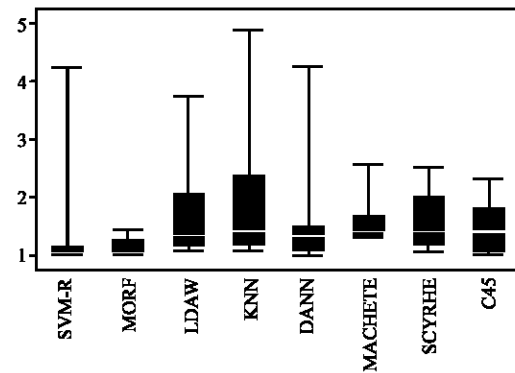


Fig. 15: Performance distribution of competing classification schemes on different datasets (Peng *et al.*, 2003)

over KNN methods outweighs this additional rate. The experimental results showed evidently that the MORF algorithm can potentially improve the performance of classification over KNN and SVM-R, when used independently from each other.

DISCUSSION AND CONCLUSION

A comprehensive review of Support Vector Machine (SVM), k-nearest neighbor (KNN) and several SVM-KNN hybrid classification models have been presented in this study. In our early stage of investigation on different machine learning classification schemes, we found that both KNN classification approach and SVM classification approach have been widely implemented in many real world applications due to their good performance which guarantee high classification effectiveness and efficiency. The KNN approach is outstanding with its simple training and classifying algorithms, due to the fact that KNN implements instant-based learning algorithm. Hence, it

does not need construction of a feature space until it receives a command to classify an input sample. Besides this, it can manage multiclass data set sufficiently by requiring a small amount of training samples. Despite that, most distance metrics, such as that of nearest neighbors, are expected to become less sensitive as the dimensionality of the noisy data increases, thus limiting the performance of NN when applied to data classification. In other words, the performance of KNN classifier is severely degraded by noises. In many cases, KNN is often used as the preliminary classifier to prune the training samples and eliminate those samples effectively to reduce the sample size for better classification.

In general, SVM has been commonly reported as one of the discriminative classification schemes which are recognized to be more accurate than many of the other classification approaches. The good classification performance of SVM is due to the fact that it implements Structural Risk Minimization (SRM) principle that which entails finding a decision surfaces, called optimal separating hyper-planes, to guarantee the lowest true error. However, the classification performance of SVM is not very high when samples are close to the hyper-planes. The samples intermixed in another class or in the overlapped boundary region may cause the decision boundary too complex and may be harmful to improve the precise of SVM. Another issue of SVM classifier involves time-consuming optimization and computation of pair-wise distances. SVM classifier suffers from the problem of computational intensive due to the convoluted training and classifying processes, especially when the number of training samples and the number of features for each sample are huge.

Thus, in order to improve the efficacy of classification performance, many recent researchers focus their attention in combining the SVM and KNN methods to classify various types of data with high dimensionality (in terms of and number of training samples and number of features), such as gene expression, illness, visual object, as well as email spam filtering data.

The hybridized models could bring a number of benefits. In the cases that have been reviewed in this study, the hybridized models can be used for large and multiclass datasets, reasonably lower computational complexity in training and at run time and better classification accuracy. In such hybridized models, the KNN is typically used to reduce the size of training set. Hybrid models of SVM combines KNN can perform better and faster since SVM will not be trained with the entire training set. Also, there exists evidence that SVM-KNN

hybrid models can be used to reduce bias caused by NN finite samples due to the curse of high dimensionality. These severe partialities can happen in the NN rule, which if not properly dealt with can strongly affect classification performance. In almost every instance, if not all, it had been concluded that SVM-KNN classifier has higher accuracy compared to SVM and KNN used independently, with a few reported that it was due to more information obtained after training is performed. Also, the number of classes used in the training process will not have high effect to the classifier than most other kinds of classifiers. Misclassification rate for the prediction or testing set was found to be lower with samples pruning used than that of KNN and SVM, which indicated the stability of the classification performance of the algorithm. However, the common but major drawback of most SVM-KNN classifier is low processing speed, due to the fact that it needs to calculate distance for each of the sample data, thus increasing the accuracy of the classification at the price of extra computation.

In conclusion, while it is usually costly in building a hybridized SVM-KNN method for classification, the increase in classification performance over conventional KNN and SVM methods outweighs this additional rate. If speed is not the primary concern, its classification performance makes it a promising choice of classifier due to its high stability in most applications.

REFERENCES

- Abd Alla, A.N., 2006. Three phase induction motor faults detection by using radial basis function neural network. *J. Applied Sci.*, 6: 2817-2820.
- Apte, C., F. Damerou and S.M. Weiss, 1994. Automated learning of decision rules for text categorization. *ACM Trans. Inform. Syst.*, 12: 233-251.
- Blanzieri, E. and A. Bryl, 2007a. Instance-based spam filtering using SVM nearest neighbor classifier. *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference*, May 7-9, Key West, Florida, USA, pp: 441-442.
- Blanzieri, E. and A. Bryl, 2007b. Evaluation of the highest probability SVM nearest neighbor classifier with variable relative error cost. *Proceedings of the 4th Conference on Email and Anti- Spam*, Aug. 2-3, Mountain View, California, USA., pp: 5-9.
- Brown, M.P.S., D. Lin, W.N. Grundy, N. Cristianini and C. Sugnet *et al.*, 1999. *Support Vector Machine Classification of Microarray Gene Expression Data*. University of California, Santa Cruz, California.

- Brucher, H., G. Knolmayer and M.A. Mittermayer, 2002. Document classification methods for organizing explicit knowledge. Proceedings of the 3rd European Conference on Organizational Knowledge, Learning and Capabilities, (ECOKLC'02), Institute of Information Systems, University of Bern, Engehaldenstrasse Bern, Switzerland, Athens, Greece, pp: 124-126.
- Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Data Min. Knowledge Discovery*, 2: 121-167.
- Chakrabarti, S., S. Roy and M.V. Soundalgekar, 2003. Fast and accurate text classification via multiple linear discriminant projection. *VLDB J.*, 12: 170-185.
- Domingos, P. and M. Pazzani, 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29: 103-130.
- Dror, G., R. Sorek and R. Shamir, 2005. Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, 21: 897-901.
- Drucker, H., D. Wu and V.N. Vapnik, 1999. Support vector machines for spam categorization. *IEEE Trans. Neural Network*, 10: 1048-1054.
- Eyheramendy, S., A. Genkin, W.H. Ju, D. Lewis and D. Madigan, 2003. Sparse bayesian classifiers for text categorization. Department of Statistics, Rutgers University. <http://www.stat.rutgers.edu/~madigan/PAPERS/jicrd-v13.pdf>.
- Godbole, S., 2006. Inter-class relationship in text classification. Ph.D. Thesis, Indian Institute of Technology, Bombay, India. <http://www.it.itb.ac.in/~shantanu/work/thesis.pdf>.
- Greiner, R. and J. Schaffer, 2001. *AIExploratorium-Decision Trees*. Department of Computing Science, Alberta, Canada.
- Gutschoven, B. and P. Verlinde, 2000. Multi-modal identity verification using support vector machines (SVM). *Proc. 3rd Int. Conf. Inform. Fusion*, 2: 3-8.
- Han, E.H., G. Karypis and V. Kumar, 1999. Text categorization using weight adjusted k-nearest neighbour classification. Department of Computer Science and Engineering, Army HPC Research Center, University of Minnesota. <http://glaros.dtc.umn.edu/gkhome/fetch/papers/wknnPAKDD01.pdf>.
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*. 2nd Edn., Prentice Hall, New Jersey, USA., ISBN: 8120323734, pp: 443-484.
- Isa, D., L.H. Lee, V.P. Kallimani and R. Prasad, 2008a. Polychotomiser for case-based reasoning beyond the traditional bayesian classification approach. *Comput. Inform. Sci.*, 1: 57-68.
- Isa, D., L.H. Lee, V.P. Kallimani and R. Rajkumar, 2008b. Text document pre-processing with the bayes formula for classification using the support vector machine. *IEEE Trans. Knowledge Data Eng.*, 20: 1264-1272.
- Isa, D., L.H. Lee, V.P. Kallimani and R. Rajkumar, 2008c. Text document pre-processing with the bayes formula for classification based on the vector space model. *Comput. Inform. Sci.*, 1: 79-90.
- Isa, D., V.P. Kallimani and L.H. Lee, 2009. Using the self-organizing map for clustering of text documents. *Expert Syst. Appl.*, 36: 9584-9591.
- Joachims, T., 1999. Making Large-Scale SVM Learning Practical. In: *Advances in Kernel Methods-Support Vector Learning*, Scholkopf, B., C.J.C. Burges and A.J. Smola, (Eds.). MIT Press, Cambridge, MA., ISBN-10: 0-262-19416-3, pp: 169-184.
- Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, Apr. 21-23, Springer Berlin/Heidelberg, pp: 137-142.
- Kim, S.B., H.C. Rim, D.S. Yook and H.S. Lim, 2002a. Effective methods for improving naive bayes text classifiers. *Lecture Notes Comput. Sci.*, 2417: 479-484.
- Kim, K.I., K.C. Jung, S.H. Park and H.J. Kim, 2002b. Support vector machines for texture classification. *IEEE Trans. Pattern Anal. Machine Intell.*, 24: 1542-1550.
- Lee, L.H., 2008. Domain specific classification and hybrid classification techniques for text document categorization. Ph.D. Thesis, The University of Nottingham. <http://www.nottingham.edu.my/Faculties/Science/CSIT/Documents/ThesisLam.pdf>.
- Lee, L.H., D. Isa, W.O. Choo and W.Y. Chue, 2010. Tournament structure ranking techniques for Bayesian text classification with highly similar categories. *J. Applied Sci.*, 10: 1243-1254.
- Li, S., J.T. Kwok, H. Zhu and Y. Wang, 2003. Texture classification using the support vector machines. *Pattern Recognition*, 36: 2883-2893.
- Lin, Y., 1999. Support vector machines and the bayes rule in classification. Technical Report No. 1014, Department of Statistics, University of Wisconsin, Madison. <http://www.stat.wisc.edu/public/ftp/yilin/tr1014.ps>.
- McCallum, A. and K. Nigam, 2003. A comparison of event models for naive bayes text classification. *J. Machine Learning Res.*, 3: 1265-1287.
- Mei, Z., Q. Shen and B. Ye, 2009. Hybridized KNN and SVM for gene expression data classification. *Life Sci. J.*, 6: 61-66.

- Osuna, R.G., 2002. Lecture Notes CS 790: Introduction to Pattern Recognition. Wright State University, Dayton, Ohio, USA.
- Peng, J., D.R. Heisterkamp and H.K. Dai, 2003. LDA/SVM driven nearest neighbor classification. *IEEE Trans. Neural Networks*, 14: 940-942.
- Quinlan, J.R., 1993. *C4.5: Program for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA., ISBN: 1-55860-238-0, pp: 302.
- Rangwala, H. and G. Karypis, 2005. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21: 4239-4247.
- Rish, I., 2001. An empirical study of the naive bayes classifier. *Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence, (WEMAI'01)*, Watson Research Center, Hawthorne, pp: 41-46.
- Sani, M.M., K.A. Ishak and S.A. Samad, 2009. Classification using adaptive multiscale retinex and support vector machine for face recognition system. *J. Applied Sci.*, 10: 506-511.
- Shen, X. and Y. Lin, 2004. Gene expression data classification using SVM-KNN classifier. *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing*, Oct. 20-22, Hong Kong, China, pp: 149-152.
- Shin, K., A. Abraham and S.Y. Han, 2006. Improving kNN text categorization by removing outliers from training set. *Proceedings of the 7th International Conference of Computational Linguistics and Intelligent Text Processing*, Feb. 19-25, Mexico City, Mexico, Germany, pp: 563-566.
- Soltanizadeh, H. and B.S. Shahriar, 2008. Feature extraction and classification of objects in the rosette pattern using component analysis and neural network. *J. Applied Sci.*, 8: 4088-4096.
- Tan, S., 2006. An effective refinement strategy for KNN Text classifier. *Expert Syst. Appl.*, 30: 290-298.
- Tao, D., X. Tang, X. Li and X. Wu, 2006. Asymmetric bagging and random subspace for support vector machines based relevance feedback in image retrieval. *IEEE Trans. Patt. Anal. Mach. Int.*, 28: 1088-1099.
- Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. 1st Edn., Springer-Verlag, New York, ISBN: 0-387-94559-8.
- Yang, Y. and X. Liu, 1999. A re-examination of text categorization method. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 15-19, New York, USA., pp: 42-49.
- Yang, Y.M. and J. Pedersen, 1997. A comparative study on feature selection in text categorization. *Proceedings of the 14th International Conference on Machine Learning, (ICML'97)*, Morgan Kaufmann, pp: 412-420.
- Zhang, H., A.C. Berg, M. Maire and J. Malik, 2006. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Volume 2*, June 17-22, Washington, DC, USA., pp: 2126-2136.
- Zhang, J.P., 2004. Chapter 6: The Structural Risk Minimization Principle. *Intelligent Information Processing Laboratory, Fudan*.