Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Reduction of Training Data Using Parallel Hyperplane for Support Vector Machine

Pardis Birzhandi, Kyung Tae Kim, Byungjun Lee & Hee Yong Youn

To cite this article: Pardis Birzhandi, Kyung Tae Kim, Byungjun Lee & Hee Yong Youn (2019) Reduction of Training Data Using Parallel Hyperplane for Support Vector Machine, Applied Artificial Intelligence, 33:6, 497-516, DOI: 10.1080/08839514.2019.1583449

To link to this article: https://doi.org/10.1080/08839514.2019.1583449

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 6, 497–516
https://doi.org/10.1080/08839514.2019.1583449

Taylor & Francis
Taylor & Francis Group

Reduction of Training Data Using Parallel Hyperplane for Support Vector Machine

Pardis Birzhandi, Kyung Tae Kim, Byungjun Lee, and Hee Yong Youn

College of Software, Sungkyunkwan University, Suwon, South Korea

ABSTRACT

Support Vector Machine (SVM) is an efficient machine learning technique applicable to various classification problems due to its robustness. However, its time complexity grows dramatically as the number of training data increases, which makes SVM impractical for large-scale datasets. In this paper, a novel Parallel Hyperplane (PH) scheme is introduced which efficiently omits redundant training data with SVM. In the proposed scheme the PHs are recursively formed while the clusters of data points outside the PHs are removed at each repetition. Computer simulation reveals that the proposed scheme greatly reduces the training time compared to the existing clustering-based reduction scheme and SMO scheme, while allowing the accuracy of classification as high as no data reduction scheme.

Introduction

Support vector machine (SVM) is a powerful technique used to classify the data generated in various fields (Varadwaj, Purohit, and Arora 2009). Basically, the algorithms of machine learning are divided into three categories; Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning. The supervised learning algorithms have been successfully applied to various problems due to the promising performance of classification. Among them, SVM has been shown to be effective for a wide variety of problems such as handwritten character recognition, face detection, pedestrian detection, and text categorization (Cristianini and John 2000).

SVM is a practical approach useful for both linearly and nonlinearly separable data. The main idea employed for nonlinearly separable data is to map low dimension data points into high dimension space using kernel function which makes the data points linearly separable. The key operation

CONTACT Hee Yong Youn youn7147@skku.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

PUBLICATION STATEMENT

Manuscript title: *Reduction of Training Data Using Parallel Hyperplane for Support Vector Machine*

All authors of the paper certify that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the *Applied Artificial Intelligence*.

Authors:

Pardis Birzhandi, Kyung Tae Kim, Byungjun Lee, Hee Yong Youn

of SVM is to find an optimal hyperplane which maximizes the separating margin between the data. Even though it has gained wide acceptance with various classification problems, it requires a large amount of computation and memory for handling the training data. The issue can be alleviated by reducing the number of training data having no or little effect on the construction of the hyperplane, which are called redundant data points.

Only a small group of the training samples called support vector (SV) influence the creation of the hyperplane in SVM. Therefore, the training samples that are not relevant to the SVs can be removed without affecting the construction of appropriate decision function. Here the key is to correctly and efficiently identify the redundant data points among the given training dataset. Various approaches have been proposed to reduce the computation overhead of training with SVM. Among them, combining the clustering algorithm with SVM is one of the common ways employed for reducing the complexity of SVM. Hierarchical clustering (Awad et al. 2004; Heisele et al. 2003; Yu et al. 2005), Fuzzy clustering (Almasi and Rouhani 2016; Cervantes, Li, and Yu 2006; Sohn and Dagli 2001), crisp cluster (Koggalage and Halgamuge 2004), and K-mean clustering (Shen et al. 2013, 2016; Yao et al. 2013) have shown to be effective for diverse classification problems. There also exist the schemes based on the distance from hyperplane (Li, Liu, and Wang 2011; Xia et al. 2015) and neighboring data (Xu and Dong 2016).

While there exist various approaches for the reduction of training data of SVM, little attention has been paid to the manipulation of clusters and hyperplane in an integrated way. In this paper, thus, a new notion of Parallel Hyperplane (PH) is proposed to substantially reduce the amount of training data without degrading the classification accuracy. Here what we call approximate hyperplane is built based on the clusters' centroids, and a PH is parallel to it and passes through the center of the clusters' centroids. The clusters whose data points are not located between the PH and approximate hyperplane are removed from the training dataset. This is because they cannot be included in SVs. A new PH is then constructed with the remaining clusters, and the process is repeated until no reduction is possible. To validate the effectiveness of the proposed scheme, the performance of the proposed PH algorithm is compared with the clustering-based algorithm (Cervantes, Li, and Yu 2006; Li, Cervantes, and Yu 2010) and SMO algorithm (Platt 1998) in terms of training time and accuracy. The simulation with a wide range of data demonstrates that the proposed scheme greatly decreases the training time without lowering the accuracy of classification. This is due to the fact that the proposed scheme preserves the SVs in omitting redundant data points, and thus no harm on the construction of the hyperplane. Also, it turns out to be more effective for the dataset of relatively large standard deviation and a large number of clusters up to a certain point. The main contributions of the paper are summarized below.

- Development of a new notion of parallel hyperplane for effectively omitting redundant data points with SVM. As a result, the training time can be significantly reduced.
- By discretely omitting the data points not belonging to the SVs, high classification accuracy as the scheme of no training data reduction can be achieved.
- Investigation of the effect of the number of clusters and distribution of data. Increasing the number of clusters up to a certain value allows a dramatic reduction of the number of training data points but not beyond it.

The organization of the rest of the paper is as follows: Section 2 introduces the related work, and in Section 3 the proposed PH scheme is presented. Section 4 evaluates the proposed scheme by computer simulation, and finally, the conclusion is given in Section 5.

Related Work

Support Vector Machine

The basic concept of SVM is represented in Figure 1, which finds a hyperplane separating the *d*-dimensional data into two classes based on the maximum margin rule (Joachims 2002). The margin is defined as the geometrical distance of blank space between the two species (Yao et al. 2103). To increase the applicability of SVM, the separating margin needs to be maximized. The maximum margin rule uses a safe distance between the data

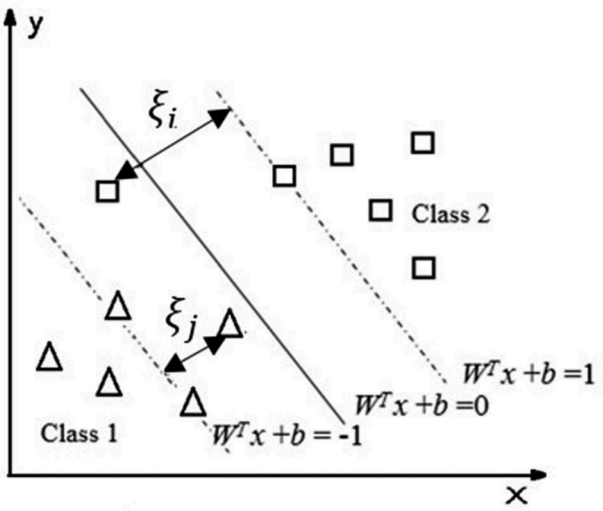

Figure 1. An example of SVM classifier.

points and hyperplane to achieve accurate classification. This margin is called safety margin.

The optimal hyperplane (decision boundary) is defined by $w \cdot x + b = 0$, where w is a weighted vector showing the orientation of a discriminant plane. The scalar b determines the offset of the plane from the origin. In Figure 1, the SVs are data points on the dotted lines. Assume that a training dataset is given as $S = \{(x_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in \{1, -1\}\}$, $i = 1, 2, \ldots, N$. Here x_i is i-th data point among the N data points in S. The dimension of the data points is d. $y_i \in \{1, -1\}$ is the output index of x_i. The optimal hyperplane is found by solving the quadratic problem below:

$$\min_{\xi, w} \|\vec{w}\|^2 + c \sum_{i=1}^{N} \xi_i \tag{1}$$

$$\text{s.t. } y_i\left(\vec{w}\vec{x_i} + b\right) \geq 1 - \xi_i, \ (i = 1, 2, \ldots, N)\ \xi_i \geq 0,$$

where ξ_i is the slack variable. The tunable scalar is defined by c which determines the cost of constraint violation. The dual problem of Eq. (1) is represented as below:

$$\max_{\lambda_i} -\frac{1}{2} \sum_{i=1}^{N} \lambda_i + \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j \left(x_i x_j\right) \tag{2}$$

$$\text{s.t. } \sum_{i=1}^{N} \lambda_i y_i = 0 \ 0 \leq \lambda_i \leq c$$

λ_i (i= 1, …, N) is a Lagrange multiplier. The optimal classifier parameters can be expressed as:

$$\begin{aligned} w &= \sum_{i=1}^{N} \lambda_i y_i x_i \\ b &= y_i - \sum_{i=1}^{N} \lambda_i y_i \left(x_i x_j\right) \\ f(x) &= sgn\left(\sum_{i=1}^{N} \lambda_i y_i \left(x_i x_j\right) + b\right) \end{aligned} \tag{3}$$

By converting the problem of Eq. (1) into its dual problem of Eq. (2), the computational complexity becomes dependent only on the number of SVs. Consequently, the optimal discriminant function is determined only by the SVs, whose portion is usually much smaller than that of non-SVs in the training set (Li, Wang, and He 2012). The SVs can be identified from the equations above. The vectors whose components, λ_i, are nonzero are the SVs, which are used to find the optimal separating hyperplane. The quadratic problem (QP) of Eq. (2) is needed to be solved to obtain the SVs. Its complexity exponentially grows as the number of training data points increases. This is the reason why extensive efforts have been put to reduce the complexity of SVM especially for handling a large-scale dataset.

Clustering

Clustering is one of the common tasks handled by unsupervised learning technique. It classifies a set of objects in such a way that the objects in a group are more similar to each other than those in other groups. K-means clustering is an unsupervised algorithm working based on the similarity. It is an iterative algorithm frequently used in the field of data mining, and quite efficient in partitioning the data points. The measure of similarity based on Euclidean distance is the main part of the algorithm, which is used to distinguish the similarity between the data points. There are two important parameters influencing the final result of clustering. The first one is the number of clusters formed, and the second one is the centroids of the initial clusters. Given the number of clusters, k, the algorithm proceeds by alternating between the two steps (Shrivastava and Ahirwal 2013). In the first step, each data are assigned to the cluster whose mean is closest to it as below.

$$if \ \|x - m_i\| \leq \left\|x - m_j\right\|, \ then \ x \in i \tag{4}$$

Here x is a data point, and m_i and m_j are the centroid of cluster-i and cluster-j respectively. In the second step, the new mean of data points is calculated as the centroid of each new clusters using Eq. (5).

$$m_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j, \tag{5}$$

where n_i is the number of data points in cluster-i.

Hierarchical clustering (Awad et al. 2004; Heisele et al. 2003; Yu et al. 2005), has been adopted for SVM, where the main issue is the variation within a class of objects. With fuzzy clustering (Sohn and Dagli 2001), fuzzy class membership is used for each sample in the training set, while the concept of crisp cluster (Koggalage and Halgamuge 2004) was introduced to identify irrelevant samples from the training data points.

K-SVM algorithm (Yao et al. 2013) was proposed to find the SVs while reducing the number of training data, which is useful for binary classification with small-scale data points. The k-mean clustering was applied to large-scale data points (Shen et al. 2013, 2016), where the clusters far from hyperplane are removed using the Max-min cluster distance scheme. A measure in kernel space based on the distance from hyperplane was also proposed to extract a subset of the data which includes the SVs (Xia et al. 2015). A fast classification algorithm (Sun et al. 2017) was proposed for multi-label large-scale datasets by applying the method handling approximately extreme points. The proposed scheme is presented next.

The Proposed Scheme

The Parallel Hyperplane

Only a small number of the training samples called SVs have a dominant effect on creating the hyperplane in the SVM technique since they lie close to the decision boundary. Therefore, the training samples irrelevant to the SVs can be removed without the deterioration of the construction of appropriate decision function. The data points which are far from the hyperplane are deemed to be not part of the SVs. Removing them can significantly improve the performance of SVM in terms of computation and memory overhead. In order to efficiently find such data points, the PH scheme is proposed which consists of seven steps as explained below:

Step 1: Clustering of data

The k-mean clustering algorithm is used to cluster the original training data points into k clusters. The value of k is selected by the user. Note that the final result of SVM is substantially dependent on the value of k. After clustering, some clusters may contain the data of two class labels (called duo-cluster), while others do only one class label (distinct cluster). The duo-clusters need to be divided into two distinct clusters. For this, the k clusters are further classified into two subsets as $U = \{u_i \mid i = 1, 2 \ldots, r\ (1 < r < k)\}$ and $V = \{v_i \mid i = r + 1, r+ 2, \ldots, k\}$. Set U is constructed by the clusters of only one class label, while set V of the clusters of two class labels. The set of all clusters, C, is thus $C = U \cup V$. In the classification process, only distinct clusters are acceptable. Therefore, every cluster in V should be divided into two distinct clusters. Assume that the clusters in set V are further divided into L sub-clusters $V_L = \{v_{lj} \mid j = 1, \ldots, l\}$. In the SVM technique, the SVs lie on the bindery of convex hull of two distinct classes, and consequently it is highly likely that the SVs are in the clusters of the V_L (Heisele et al. 2003). Therefore, v_{lj} clusters are not removed. In Figure 2(a) the center cluster of two sub-clusters is an example of duo- cluster. Note that, if there exist $(k–r)$ duo-clusters, the total number of clusters after dividing them into two becomes $r + 2(k–r) = (2k–r)$.

Step 2: Identifying the centroid of each cluster

Applying k-mean clustering, the centroid of each cluster ($u_i \in U$) is determined. Also, for each cluster, the distances between the centroid and the data points inside the cluster are decided. $M_U = \{m_{u1}, m_{u2}, \ldots, m_{uk}\}$ is the set of the centroids of U. For each cluster of v_l, the centroid of the cluster is calculated using Eq. (6).

$$m_{v_l} = (1/n) \sum_{i=1}^{n} x_i, \tag{6}$$

where n is the number of data points in cluster v_l. $M_{VL}= \{m_{vl1}, m_{vl2}, \ldots, m_{vl}\}$ is defined as the set of centroids of the clusters of V_L. Then, the distance between

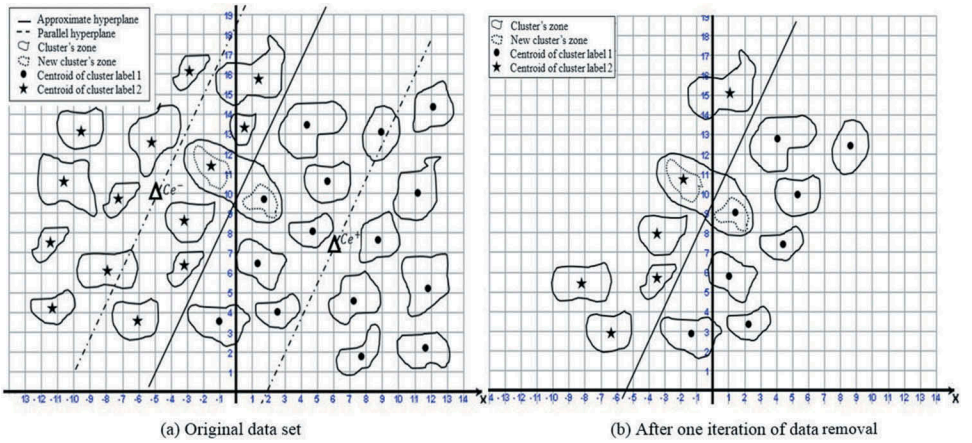

Figure 2. An example of the proposed scheme.

each data point and its corresponding centroid in each cluster is decided, and the maximum distance in each cluster, $d(max)_i$ (i= 1, …, ($2k$–r)), is determined.

Step 3: Obtaining approximate hyperplane

In this step approximate hyperplane is found with only the centroid of the clusters of the training data instead of all data points. This hyperplane is represented by a normal vector $\vec{w}$ and bias b.

Step 4: Finding the centroid of the clusters' centroids in the two sides of approximate hyperplane

Assume that $M^+ = \{(m_i, y_i) \mid m_i \in (M_U \cup M_{VL}), y_i = 1\}$ is the set of the centroids of the clusters of label_1 and $M^- = \{(m_i, y_i) \mid m_i \in (M_U \cup M_{VL}), y_i = -1\}$ is for the label_(−1). The centroid of set M^+ and M^- are denoted as Ce^+ and Ce^-, respectively. They are decided by Eq. (7), assuming n clusters in each side.

$$Ce^+ = (1/n)\sum_{i=1}^{n} m_i, \qquad \forall i \in M^+ \tag{7}$$

$$Ce^- = (1/n)\sum_{i=1}^{n} m_i, \qquad \forall i \in M^-$$

Step 5: Finding PH

The PHs each passes Ce^+ and Ce^- are found. They are denoted as PH^+ and PH^-, respectively. Note that the hyperplane is defined by $ax + by + cz + d = 0$. The normal vector to this plane is defined as $\vec{w} = (a, b, c)$. Then, the PH passing through the point (x_0, y_0, z_0) is found as below:

$$PH: \ \vec{w}.((x - x_0), \ (y - y_0), \ (z - z_0)) \tag{8}$$
$$= a(x - x_0) + b\ (y - y_0) + c\ (z - z_0) = 0$$

In Figure 2(a), an example of clustered data points of two distinct labels is shown. The dot and star indicate the center of each cluster of label-1 and label-2, respectively. The approximate hyperplane is represented by a solid line. The triangles marked as Ce^+ and Ce^- are the centroids of the centroids

of the clusters in the two different sides of the approximate hyperplane. The PHs passing through Ce^+ or Ce^- are shown by dotted line. Observe from Figure 2(a) that the clusters beyond the PHs have little influence on the approximate hyperplane and thus eventually on the classification process. Efficiently identifying them is the main objective of the proposed scheme, and Figure 2(b) is the result of one iteration of data removal.

Step 6: Removing the clusters lying at the positive side of PH^+ and negative side of PH^-

The main goal of this step is to eliminate the training data far from the approximate hyperplane. For this, the clusters lying at the positive side of PH^+ and negative side of PH^- are removed. Here, the position of each data point in set M^+ and M^- need to be compared with PH^+ and $PH-$, respectively. Denote set I^+ and I^- the data points in M^+ lying at the positive side of PH^+ and those of M^- at the negative side of $PH-$, respectively. Note that the positive side is a half space of hyperplane in the direction of the upward normal vector of the hyperplane. Similarly, another half-space is determined from the negative side.

In order to identify the position of the data points with respect to hyperplane, dot product is used. In Figure 3, A is a point on the hyperplane. The position of another point B with respect to the hyperplane is found by dot product of $\left(\overrightarrow{B-A}\right)$ and $\vec{w}$ (Eq. (9)). A positive result indicates that the vectors form an acute angle (θ) and the data point, B, lies in front of the plane. On the contrary, a negative value does that the vectors form an obtuse angle and the data point lies in the back of the plane like point C in Figure 3.

$$\vec{w} \times \left(\overrightarrow{B-A}\right) = |B|\ |A| \cos\theta \tag{9}$$

To find out the clusters to be removed, $d(max)_i$ in set I^+ and the distance between m_i and PH^+ ($d_{m_i-PH^+}$) is compared. Similarly, $d(max)_i$ in set I^- and the distance between m_i and PH^- ($d_{m_i-PH^-}$) are compared. m_i is removed if

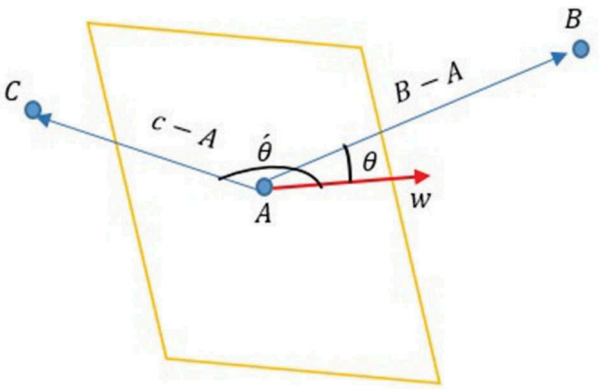

Figure 3. The position of a data with respect to the plane.

($d_{m_i-PH^+}$) ≥ $d(max)_i$ and m_i □ M_{VL}. As it is highly likely that SVs are in the clusters of the V_L, $m_i \in M_{VL}$ are not removed in the proposed scheme. Figure 2(b) shows the remaining clusters of data points of Figure 2(a) after applying the first iteration of the proposed PH algorithm. After removing the redundant clusters in Step 6, the process is repeated from Step 4 with a reduced number of clusters as far as there exist some clusters in the positive or negative side of the approximate hyperplane satisfying the condition of cluster removal. The remaining clusters are considered as the training data for the final SVM. The following is the procedure of the proposed scheme.

(1) Put each point $m_i \in M_U \cup M_{V_L}$ whose class label is (1) in set M^+
(2) Put each point $m_i \in M_U \cup M_{V_L}$ whose class label is (−1) in set M^-
(3) Co$_m$pute the centroid of set M^+ which is called Ce^+
(4) Find the PH^+ which pass through Ce^+
(5) **For** (each data point m_i in M^+,) do
(6) Compare the position of each data points with PH^+
(7) Put each data points of M^+ which is in the positive side of parallel hyperplane in set Γ^+
(8) **End for**
(9) **For** (each data point m_i in Γ^+), do
(10) Compute the distance between each data points and PH^+ and call them $d_{m_i-PH^+}$
(11) **If** $d_{m_i-PH^+} \geq d(max)i$ **then**
(12) **If** $m_i \notin M_{V_L}$ **then**
(13) Remove m_i from sets M^+ and Γ^+
(14) **End if**
(15) **End for**
(16) **If** there were any m_i which removed from set M^+ **then**
(17) Go back to line 3
(18) **Else**, consider M^+ as a final set of cluster's centroid
(19) **End if**
(20) Compute the centroid of set M^- which is called Ce^-
(21) Find the PH^- which pass through Ce^-
(22) **For** (each data point m_i in M^-), do
(23) Compare the position of each data points with PH^-
(24) Put each data points of M^+ which is in the negative side of parallel hyperplane in set Γ^-
(25) **End for**
(26) **For** (each data point m_i in Γ^-), do
(27) Compute the distance between each data points and PH^- and call them $d_{m_i-PH^-}$
(28) **If** $d_{m_i-PH^-} \geq d(max)i$ **then**
(29) **If** $m_i \notin M_{V_L}$ **then**

(30) Remove m_i from sets M^- and Γ^-
(31) **End if**
(32) **End for**
(33) **If** there were any m_i which removed from set M^- **then**
(34) Go back to line 3
(35) **Else**, consider M^- as a final set of cluster's centroid
(36) **End if**
(37) Keep the data points of clusters whose centroids belong to M^-, M^+ and M_{V_L} as remaining data points.
(38) Remove the data points of the other clusters
(39) Apply final SVM to the remaining data points as training dataset

Data Modeling

There are some parameters directly affecting the performance of the proposed PH scheme. In the following, they are discussed along with the process of modeling of the data points. Normal and exponential distribution are used for randomly generating the training data. The probability density function (PDF) of the normal distribution and exponential distribution are defined in Eqs. (10) and (11).

$$f(x) = \frac{1}{\sqrt{2\sigma^2\pi}}\, e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{10}$$

$$f(x) = \lambda e^{-\lambda x} \tag{11}$$

Here μ and σ are the mean and standard deviation of the normal distribution, and λ is the rate parameter of the exponential distribution. A low standard deviation indicates that the data points lie close to the mean of the data of the set, while a large value implies widespread. Therefore, standard deviation is an important parameter deciding the distribution of the distances between each pair of data points in a cluster. The effect of increasing the standard deviation on the normal distribution of the data points is illustrated in Figure 4. The upper set is distributed with μ = (4, 4) and σ = 0.8, while the lower set with μ = (−4, −4) and σ = 2. The figure shows that the distance between the data points grows by increasing the value of σ. The distance factor also affects the result of k-mean clustering algorithm. Consequently, the effectiveness of the PH scheme is affected by the value of standard deviation.

Besides standard deviation, the number of clusters in k-mean clustering is another parameter affecting the performance of the PH scheme. Given a set of n data points X= (x_1, x_2, ..., x_n), k-mean clustering aims to partition the n data points into ($k \leq n$) clusters. With a fixed number of data points, the

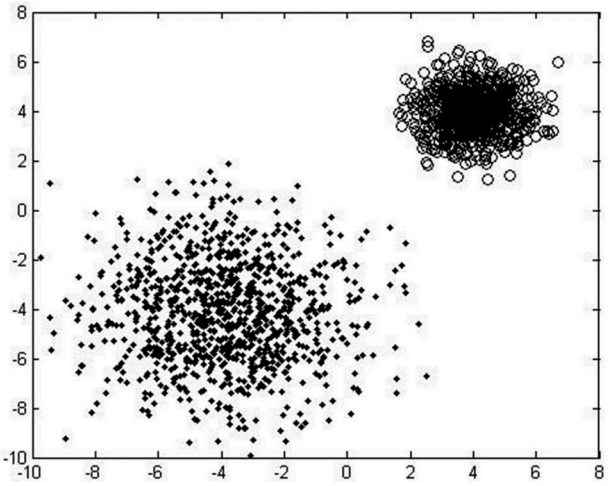

Figure 4. The distribution of data with different σ.

more clusters, the less data points in each cluster. Note that the variance of the distance in each cluster is thus influenced by the number of clusters. The maximum distance between the data points and the centroid of each cluster depends on k, and it affects the process of identification of removable clusters in the proposed scheme.

In order to investigate the effect of the number of clusters on the maximum distance between the data points, the Fisher's Iris dataset of 100 data points is divided into two different numbers of clusters of 3 and 6, as shown in Figure 5. The variance of each cluster in Figure 5(a) is greater than that of the clusters in Figure 5(b). The maximum distance of each cluster is listed in Table 1 with different numbers of clusters. Notice that increasing the number of clusters from 2 to 9 reduces the average distance from 1.6 to 0.45.

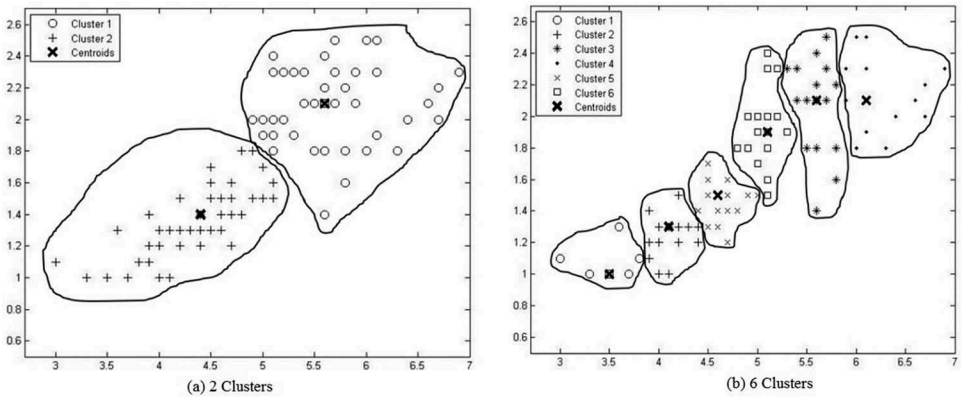

Figure 5. Clustering with Fisher's Iris dataset.

Table 1. The maximum distance of each cluster with a different number of clusters.

Number of clusters	Maximum distance									Avg
2	1.7	1.5	–	–	–	–	–	–	–	1.6
3	0.7	1.5	1.1	–	–	–	–	–	–	1.1
4	0.4	1.2	0.9	0.9	–	–	–	–	–	0.86
5	0.8	0.51	0.7	0.62	0.95	–	–	–	–	0.72
6	0.9	0.7	0.4	0.6	0.4	0.4	–	–	–	0.57
7	0.7	0.6	0.4	0.4	0.4	0.6	0.4	–	–	0.5
8	0.4	0.4	0.6	0.4	0.7	0.6	0.4	0.3	–	0.47
9	0.2	0.4	0.7	0.5	0.4	0.4	0.6	0.6	0.3	0.45

As the number of clusters and the standard deviation are important factors, they are treated as variables in the performance evaluation of the proposed scheme presented next. Figure 6 illustrates the main steps of the proposed scheme. Figure 6(a) shows a binary class dataset-1 with 2000 data points which is generated using two normal distributions of N (3, 1.69) for the upper set and N (−3, 1.69) for the lower one with class labels of 1 and 2, respectively. In this paper, the goal is to find effective classifier using only a small group of data points to reduce training time and memory

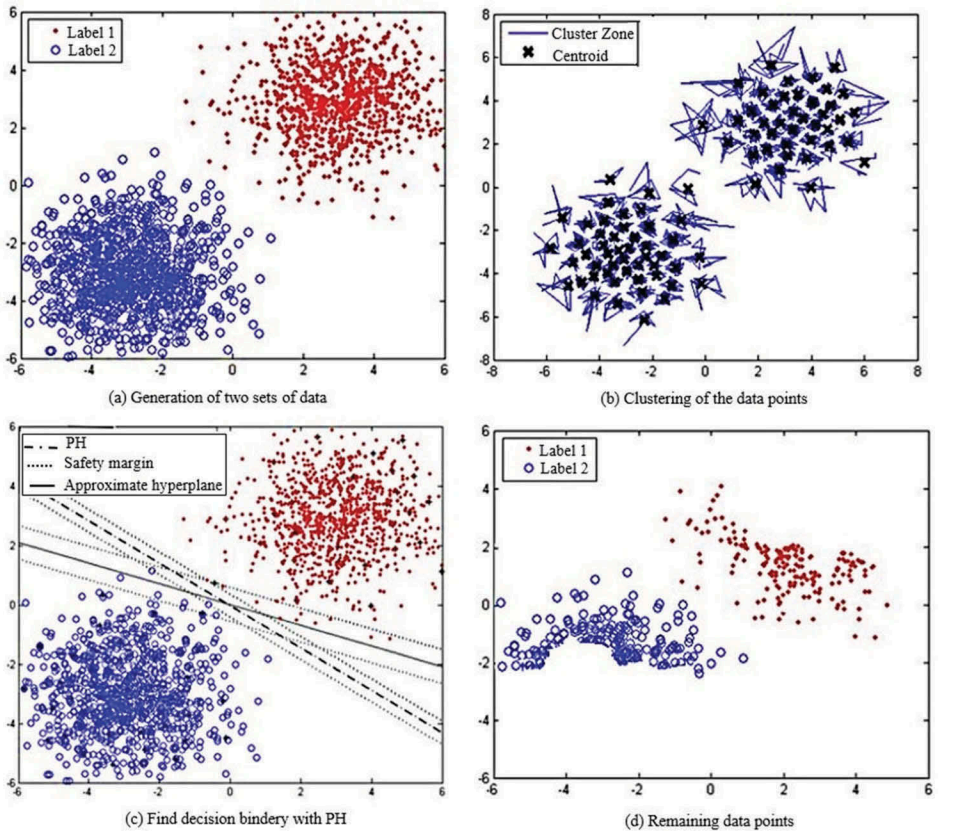

Figure 6. The steps of the proposed PH scheme with normally distributed data.

requirement. The case of 90 clusters is shown in Figure 6(b), which is obtained using the k-mean clustering algorithm.

The approximate hyperplane obtained with the centroids of the clusters is displayed by the solid line in Figure 6(c). The lines of safety margin of the approximate hyperplane are also shown as dotted lines. With the proposed PH scheme, the redundant data points having no potential to be the SVs are removed from the training data. The remaining data points are illustrated in Figure 6(d). Only 313 data points remain as the training data. Using them, the final separating classifier is obtained which is represented by dash-dotted line in Figure 6(c).

Figure 7 shows binary class dataset-2 with 10000 data points which are generated using normal and exponential distributions. The upper set is of the normal distribution with μ = (25, 25) and σ = 2.5, while the lower set is an exponential distribution with the mean parameter λ^{-1} = 3.5. The remaining data points after the proposed scheme is applied are illustrated in Figure 7(b), showing only 238 data points. Applying SVM to the remaining data points, the final separating classifier is obtained which is shown by dash-dotted line in Figure 7(a). Observe from the figure that the data points are dense around the origin and (25, 25), while sparse around the separating hyperplane.

Performance Evaluation

In this section, the performance of the proposed PH scheme is evaluated using various datasets. The skin segmentation dataset from the UCI machine learning database repository (Bhatt and Dhall 2009) is selected as the real world dataset, while several large-scale datasets are generated randomly as artificial datasets. The experiments are conducted on the PC of Core i5-4690 3.50 GHz CPU using the MATLAB programing and applying the CVX modeling system (Grant and Boyd 2013). To investigate the effectiveness of the proposed scheme, the performance of the PH algorithm

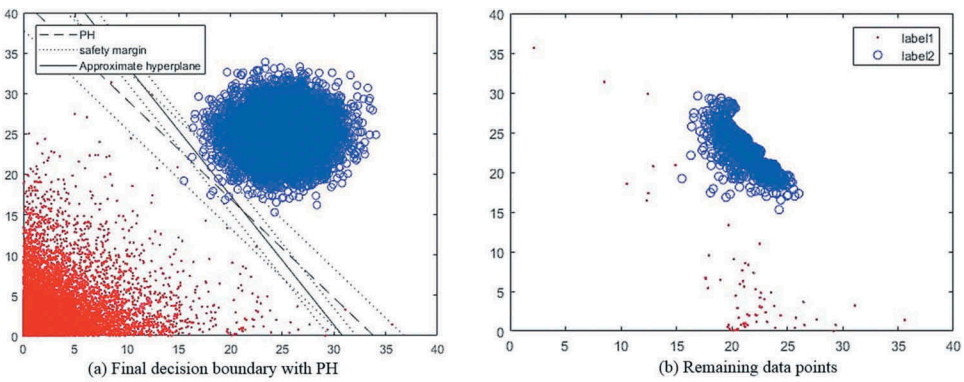

Figure 7. The Result of the proposed PH scheme.

is compared with the clustering-based algorithm (Cervantes, Li, and Yu 2006; Li, Cervantes, and Yu 2010) and SMO algorithm (Platt 1998) in terms of training time and accuracy with different number of clusters and various standard deviations.

Artificial Dataset

The parameters of dataset-1 are used to generate 100000 data points. Figures 8 and 9 show the training time and the accuracy of the three classification schemes for the 100000 data points when the number of clusters varies from 50 to 200. Unlike the proposed and the clustering-based scheme, the SMO algorithm classifies data points without clustering. Therefore, it shows constant performance regardless of the number of clusters.

Figure 8 shows that the proposed scheme is much faster than the other schemes since it allows a significant reduction in the number of training data points. The figure demonstrates that increasing the number of clusters up to 150 notably reduces the training time of classification. However, a further increase beyond it shows little impact on the reduction of training time. This property will be more delved at the end of this section.

The accuracies of the three schemes are compared in Figure 9. The results indicate that the accuracy of the proposed scheme is almost the same as the SMO algorithm and higher than the clustering-based algorithm. This is due to the fact that the proposed scheme preserves the SVs in omitting redundant data points, and thus no harm on the construction of the hyperplane. The proposed PH scheme consistently provides high accuracy while taking less training time compared to the other schemes.

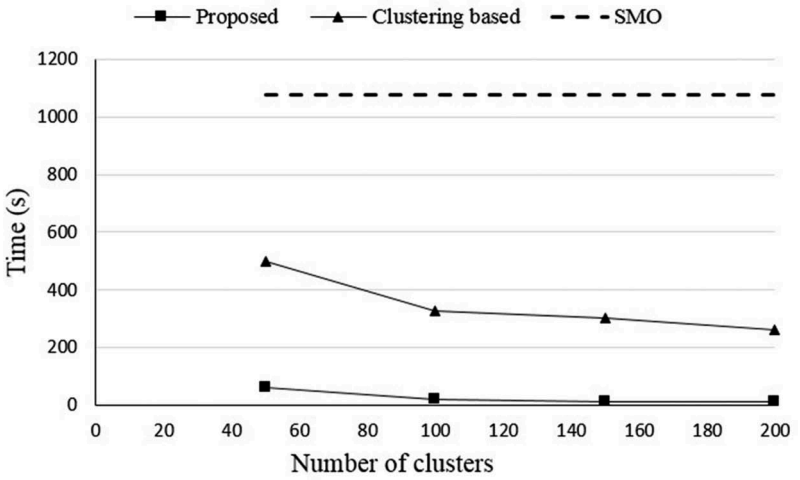

Figure 8. The comparison of training times with the data of normal distribution.

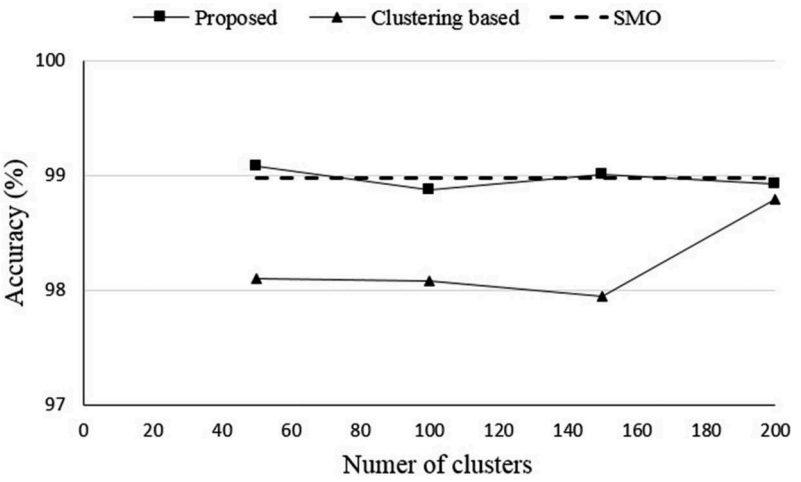

Figure 9. The comparison of classification accuracies with the data of normal distribution.

The statistical parameters of the mixed distribution of dataset-2 are used to generate 100000 data points. Figures 10 and 11 show the training time and accuracy achieved by the three classification schemes with the 100000 data points of mixed distribution. The figures illustrate that the proposed scheme still classifies the data of the mixed distribution faster than the SMO and Clustering-based schemes. Its accuracy is also higher than the other schemes. Note that the density of data points of exponential distribution around the hyperplane is lower than that with a normal distribution. As a result, the training time of the proposed scheme of Figure 10 is smaller than that of Figure 8.

Figure 12 shows the percentage of the remaining data points with the proposed scheme for different sizes of a dataset, the number of clusters, and

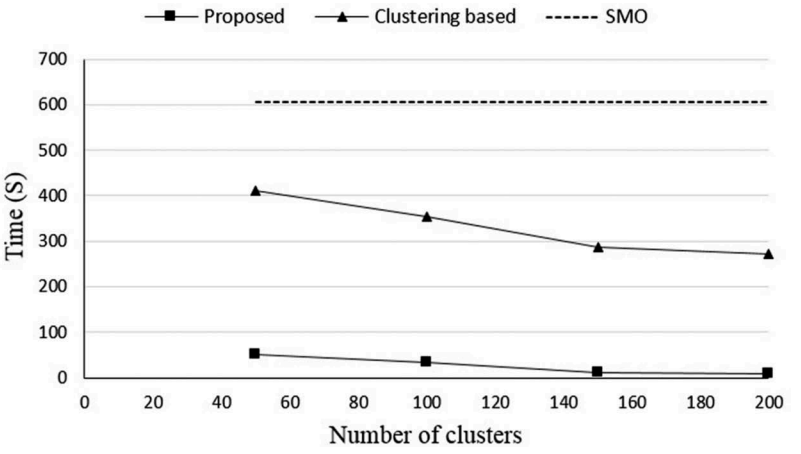

Figure 10. The comparison of training times with the data of mixed distribution.

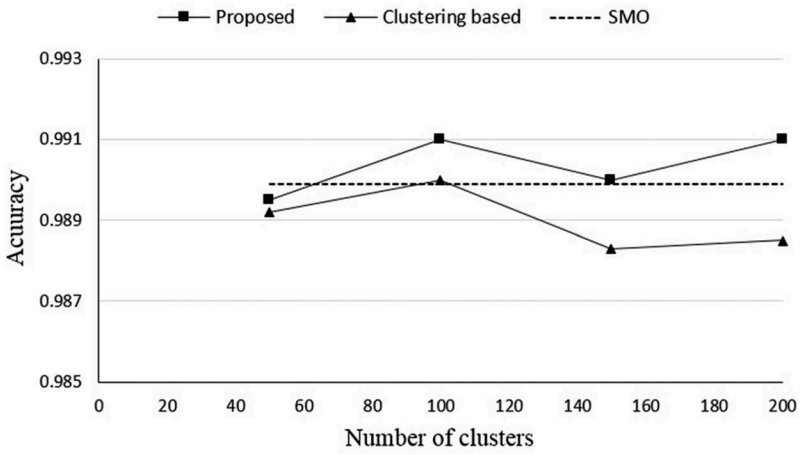

Figure 11. The comparison of classification accuracies with the data of mixed distribution.

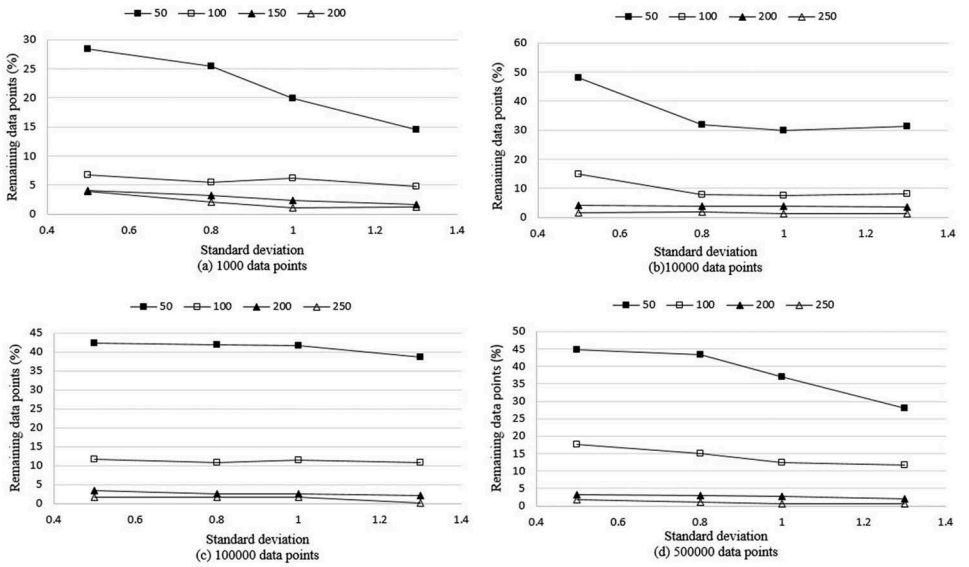

Figure 12. The percentage of the number of remaining data points with a different number of clusters.

σ. Here the data points are generated using two normal distributions of μ_1 = (3, 3) and μ_2 = (−3, −3) in four different sizes of 1000, 10000, 100000, and 500000. Four values of standard deviations of 0.5, 0.8, 1, and 1.3 are considered for the generation of each case. The results in the figures are the average value of five runs for each case. As identified from the figures, a decreasing trend in the percentage of remaining data is observed when the standard deviation increases. This means that the performance of the proposed algorithm is higher for the data of larger standard deviation. The

same characteristics are observed as the number of clusters increases. The performance gain, however, is not monotonous. Increasing the number of clusters from 50 to 100 allows a significant reduction in the number of remaining data points, while no notable decrement between 200 and 250. Increasing the number of clusters up to a certain point can dramatically reduce the number of training data points. However, a further increase beyond it shows little impact on the reduction of data points.

Figure 12(d) demonstrates that the proposed PH scheme effectively reduces the number of training data points from 500000 to 4948 using 250 clusters. This result shows 99% removal of redundant data points in the training dataset. The result of the proposed PH scheme for the data of the combination of the normal and exponential distribution is illustrated in Figure 13. A significant decrease in the number of remaining data points is observed with the increase of the number of clusters up to a certain point. Compared to Figure 12, a lower number of remaining data points are observed for some number of training data points and clusters. As an example, for 100000 training data points with 100 clusters, the number of remaining data points for normal distribution in Figure 12 is 10884 while it is 2639 in Figure 13.

Real Dataset

In addition to linearly separable datasets, the proposed PH scheme can be applied to nonlinearly separable datasets where most of the data points are linearly separable while a small portion of data points are located in the overlapped region. The Skin Segmentation dataset constructed using 3D skin textures of 245057 face images taken from people of different ages, genders, and races, is an example of this kind of datasets (Bhatt and Dhall 2009). The

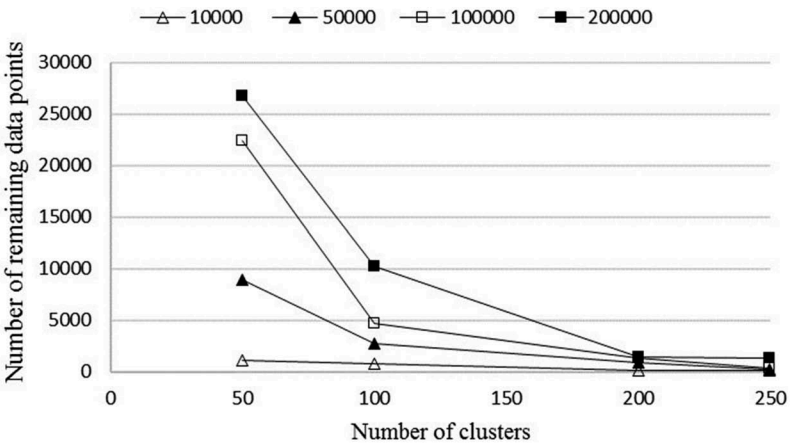

Figure 13. The number of remaining data points with mixed distribution.

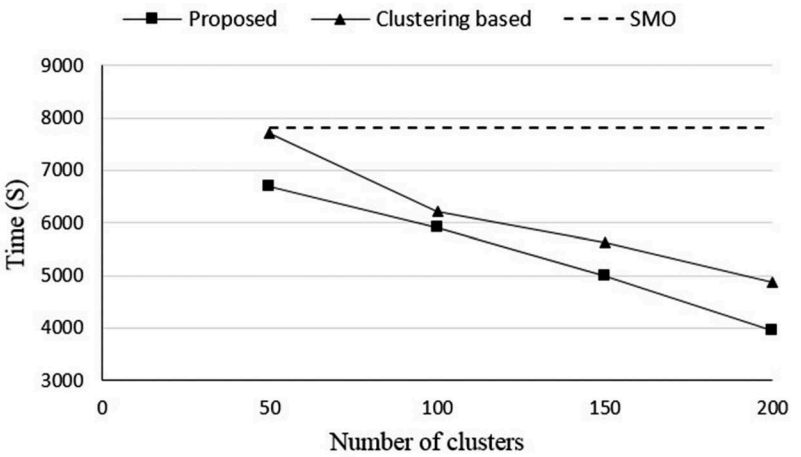

Figure 14. The comparisons of the training times and accuracies with Skin Segmentation dataset.

performance of the proposed scheme is compared with the other schemes over the Skin Segmentation dataset, as the number of clusters varies from 5o to 200. Figure 14 illustrates that the training time of the proposed PH algorithm is substantially lower than that of the other schemes while allowing high accuracy as the SMO scheme.

Conclusion

In this paper, we have introduced the PH scheme which effectively removes redundant data points from the training dataset to reduce the training time of SVM. In the proposed method the k-mean clustering algorithm is adopted to divide the given data points into different clusters. Then, the data points of the cluster which are not potentially support vectors are removed from the training dataset. MATLAB was used to verify the robustness of the proposed scheme. It demonstrates that the proposed scheme removes a significant amount of redundant data points, and eventually reduces the training time without affecting the accuracy of the classification. It also shows that the proposed scheme is much faster than the existing cluster-based scheme. The effect of the number of clusters in k-mean clustering and the data distribution was also studied. The results show that the proposed scheme is more effective for the dataset of relatively high standard deviation. Also, increasing the number of clusters up to a certain value can dramatically reduce the number of training data points. The number of clusters and the distribution of the data points influence the effectiveness of the proposed approach. In the future, we will model the impact of these factors on the performance of the proposed approach, and develop the scheme further enhancing the performance. Also, the proposed scheme will be expanded to large-scale fully nonlinearly separable data points.

Funding

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00133, Research on Edge computing via collective intelligence of hyperconnection IoT nodes), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information & communications Technology Promotion) (2015-0-00914), Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2016R1A6A3A11931385, Research of key technologies based on software defined wireless sensor network for real-time public safety service, 2017R1A2B2009095, Research on SDN-based WSN Supporting Real-time Stream Data Processing and Multiconnectivity), the second Brain Korea 21 PLUS project, and Samsung Electronics.

References

Almasi, O. N., and M. Rouhani. 2016. Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets. *Turkish Journal of Electrical Engineering and Computer Sciences* 24 (1):219–33. doi:10.3906/elk-1304-139.

Awad, M., L. Khan, F. Bastani, and I.-L. Yen. 2004. An effective support vector machines (SVMs) performance using hierarchical clustering. Paper presented at 16th IEEE International Conference on Tools with Artificial Intelligence, USA, November 15–17.

Bhatt, R., and A. Dhall. 2009. 'Skin segmentation dataset', UCI machine learning repository, https://archive.ics.uci.edu/ml/datasets/skin+segmentation

Cervantes, J., X. Li, and W. Yu. 2006. Support vector machine classification based on fuzzy clustering for large datasets. Paper presented at Mexican International Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, November.

Cristianini, N., and S.-T. John. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge, UK: Cambridge university press.

Grant, M., and S. Boyd. 2013. CVX: Matlab software for disciplined convex programming, version 2.0 beta. Accessed September 2013. http://cvxr.com/cvx.

Heisele, B., T. Serre, S. Prentice, and T. Poggio. 2003. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition* 36 (9):2007–17. doi:10.1016/S0031-3203(03)00062-1.

Joachims, T. 2002. *Introduction to support vector machines*. Cambridge university press.

Koggalage, R., and S. Halgamuge. 2004. Reducing the number of training samples for fast support vector machine classification. *Neural Information Process-Letters and Reviews* 2 (3):57–65.

Li, C., K. Liu, and H. Wang. 2011. The incremental learning algorithm with support vector machine based on hyperplane-distance. *Applied Intelligence* 34 (1):19–27. doi:10.1007/s10489-009-0176-9.

Li, X., J. Cervantes, and W. Yu. 2010. A novel SVM classification method for large datasets. Paper presented at Granular Computing (GrC), 2010 IEEE International Conference, August. San Jose, CA, USA.

Li, Y., Y. Wang, and G. He. 2012. Clustering-based distributed support vector machine in wireless sensor networks. *Journal of Information & Computational Science* 9 (4):1083–96.

Platt, J. 1998. *Sequential minimal optimization: A fast algorithm for training support vector machines*. Technical Report MSR-TR-98-14, Microsoft Research.

Shen, X., Z. Li, Z. Jiang, and Y. Zhan. 2013. Distributed SVM classification with redundant data removing. Paper presented at 2013 IEEE International Conference on Green Computing and

Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, August. Beijing, China.

Shen, X.-J., L. Mu, L. Zhen, H.-X. Wu, J.-P. Gou, and X. Chen. 2016. Large-scale support vector machine classification with redundant data reduction. *Neurocomputing* 172:189–97. doi:10.1016/j.neucom.2014.10.102.

Shrivastava, A., and R. R. Ahirwal. 2013. A SVM and K-means Clustering based Fast and Efficient Intrusion Detection System. *International Journal of Computer Application* 72 (6):25-29.

Sohn, S., and C. H. Dagli. 2001. Advantages of using fuzzy class memberships in self-organizing map and support vector machines. Paper presented at international Joint Conference on Neural Networks. Proceedings. IJCNN '. Washington, DC, USA.

Sun, Z., Z. Guo, C. Liu, X. Wang, J. Liu, and S. Liu. 2017. Fast extended one-versus-rest multi-label support vector machine using approximate extreme points. *IEEE Access* 5:8526–35. doi:10.1109/ACCESS.2017.2699662.

Varadwaj, P., N. Purohit, and B. Arora. 2009. Detection of splice sites using support vector machine. Paper presented at International Conference on Contemporary Computing. Springer, Berlin, Heidelberg.

Xia, S., Z. Xiong, Y. Luo, and L. Dong. 2015. A method to improve support vector machine based on distance to hyperplane. *Optik - International Journal for Light Electron Optics* 126 (20):2405–10. doi:10.1016/j.ijleo.2015.06.010.

Xu, W., and L. Dong. 2016. A novel relative density based support vector machine. *Optik - International Journal for Light Electron Optics* 127 (22):10348–54. doi:10.1016/j.ijleo.2016.08.027.

Yao, Y., Y. Liu, Y. Yu, H. Xu, W. Lv, Z. Li, and X. Chen. 2013. K-SVM: An effective SVM algorithm based on K-means clustering. *Journal of Computers* 8 (10):2632–39. doi:10.4304/jcp.8.10.2632-2639.

Yu, H., J. Yang, J. Han, and X. Li. 2005. Making SVMs scalable to large datasets using hierarchical cluster indexing. *Data Min Knowledge Discovery* 11 (3):295–321. doi:10.1007/s10618-005-0005-7.