

# Density Based Support Vector Machines for Classification

Zahra Nazari

Department of Information Engineering  
University of the Ryukyus  
Okinawa, Japan

Dongshik Kang

Department of Information Engineering  
University of the Ryukyus  
Okinawa, Japan

**Abstract**—Support Vector Machines (SVM) is the most successful algorithm for classification problems. SVM learns the decision boundary from two classes (for Binary Classification) of training points. However, sometimes there are some less meaningful samples amongst training points, which are corrupted by noises or misplaced in wrong side, called outliers. These outliers are affecting on margin and classification performance, and machine should better to discard them. SVM as a popular and widely used classification algorithm is very sensitive to these outliers and lacks the ability to discard them. Many research results prove this sensitivity which is a weak point for SVM. Different approaches are proposed to reduce the effect of outliers but no method is suitable for all types of data sets. In this paper, the new method of Density Based SVM (DBSVM) is introduced. Population Density is the basic concept which is used in this method for both linear and non-linear SVM to detect outliers. Experiments on artificial data sets, real high-dimensional benchmark data sets of Liver disorder and Heart disease, and data sets of new and fatigued banknotes' acoustic signals can prove the efficiency of this method on noisy data classification and the better generalization that it can provide compared to the standard SVM.

**Keywords**—SVM; Density Based SVM; Classification; Pattern Recognition; Outlier removal

## I. INTRODUCTION

Support Vector Machines is an important example of kernel methods, one of the key areas in machine learning. It is originated from the theoretical foundations of the Statistical Learning Theory and Structural Risk Minimization (SRM) [1, 2]. SVM was introduced by Vapnik and colleagues in 1970's, but its major developments were formed in 1990's. The main idea behind SVM is to find an optimal separating hyperplane with maximized margin. The maximum margin reduces the empirical risks (training errors) and causes a very good generalization performance. SVM became very famous because of its high ability in generalization and good performance in pattern recognition (digit recognition, computer vision, and text & speech categorization, etc.) and have found application in a wide variety of areas [2].

Classification with SVM is formulated as a quadratic programming which can be solved by using optimization algorithms. In binary classification problems the standard SVM can be used and data points will be classified without any misclassification. However in real world problems, sometimes there are many data points which are corrupted by noises or misplaced on the wrong side. These data points are

called outliers and sensitivity of SVM to these outliers is a weak point for this algorithm. There are many approaches proposed to reduce this sensitivity; the Central SVM method (CSVM) which is using class center vectors [3], Adaptive Margin SVM for classification which propose a reformulation of the minimization problem [4], Mapping original input space to normalized feature space for increasing the stability to noise [5], Robust SVM for solving the over fitting problem [6], and Fuzzy SVM [7] are some examples of proposed approaches to reduce the effects of outliers and noises.

Fuzzy SVM is developed on the theory of the SVM and fuzzy membership for each data point shows the attitude of the corresponding point toward one class and also represents the importance of the data points to the decision boundary. The data points with a bigger fuzzy membership will be treated more important and will contribute more to the learning of decision boundary [7].

This paper is organized as follows. The theory of Support Vector Machines will be explained in section II. The Basic concept which is used to develop DBSVM will be explained in section III. Density Based SVM will be introduced in section IV. Experiments and comparison of standard SVM performance to DBSVM performance will be discussed in section V.

## II. SUPPORT VECTOR MACHINES

Data classification process using SVM includes two stages: learning is the first stage, the aim of which is to analyze labeled data and learn a mapping from  $x$  to  $y$  where  $y = \{1, \dots, C\}$  (with  $C$  being the number of classes) and to build a classifier. The second stage is predicting which is using the established model for predicting on novel inputs. SVM is one of the most successful classification algorithms and its important property is that the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum [8]. The basis of the theory of SVM for classification problems will be reviewed in the following.

### A. Hard Margin (Linear) SVM

The linearly separable case is the easiest classification problem which is rare in practice. In this case data pairs can be classified perfectly and the empirical risk can be set to zero. In linearly separable cases, among all the separating hyperplanes which minimize the empirical risk, the one with the largest margin is required. This can be expressed as the idea that a

classifier with a smaller margin will have a higher expected risk [2]. Suppose that a set of 2-dimensional labeled training points  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  is given and each of them has a class label  $y \in \{-1, 1\}$  which denotes the two classes separately. During the learning stage the machine finds parameters  $w$  and  $b$  of the decision function  $f(x)$  given as:

$$f(x) = \text{sgn}(x \cdot w^T + b) \quad (1)$$

where  $w$  is the weight vector and  $b$  is the bias. SVM, after learning by training points can produce an output for unknown data point according to above decision function (1). The linearly separable data points can be classified by solving the following quadratic program:

$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ y_i(x_i \cdot w^T + b) \geq 1 \quad i = 1, \dots, N \end{cases} \quad (2)$$

### B. Soft Margin SVM

In previous section the training points were assumed that are linearly separable and the resulting support vector machine will give exact separation of the training points which is not very realistic. Sometimes in real-world problems the training points are overlapped (slightly nonlinear) and some samples cannot be classified correctly and the constraint in (2) will not be satisfied. Therefore classification violation must be allowed in the SVM. In practice the soft margin will be allowed. This approach allows some training points to be on the wrong side of the separating hyperplane, but with a penalty that increases with the distance from hyperplane [2, 9]. To do this, the nonnegative variable  $\xi \geq 0$  will be used to measure the amount of this violation and (2) will be modified to (3):

$$\begin{cases} \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ y_i(x_i \cdot w^T + b) \geq 1 - \xi_i \quad i = 1, \dots, N \\ \xi_i \geq 0 \quad i = 1, \dots, N \end{cases} \quad (3)$$

where  $C \cdot \sum \xi_i$  is the distance of error samples to their correct places. Parameter  $C > 0$  (the only free parameter in SVM) controls the trade-off between slack variable penalty and the margin [8].

### C. Non-linear SVM

In case of considerable class overlapping (seriously nonlinear) of the training points, soft margin SVM classifiers are unable to separate the samples into classes appropriately. Therefore SVM transforms samples  $x$  from original input space to a higher dimensional feature space by a non-linear vector mapping function  $\Phi(x) = R^n \rightarrow F$ . However the vector mapping function ( $\Phi$ ) leads to high computational expenses. Thus, this transformation can be performed by kernel function

which allows more simplified representation of the data. Polynomial, Sigmoidal, and Gaussian (RBF) are some popular kernel functions for this kind of transformation [2, 8, 10, 11].

The different distribution in the feature space enables the fitting of a linear hypersurface in order to separate all samples into the classes. Classification is easier in higher dimensions, but computation is costly. The resulting separating hypersurface in feature space will be optimal in the sense of being a maximal margin classifier with respect to training points [2]. The vector  $\Phi(x_i)$  in the feature space corresponds to vector  $x_i$  in the original space. The solution in the SVM does not depend directly to input vectors, rather to dot product between input vectors, and so the dot product of  $\Phi(x_i) \cdot \Phi(x_j)$  is needed. It would be preferable to be able to define the dot product directly rather than defining the mapping  $\Phi$  explicitly. The kernel function computes the dot product of training points in feature space and there will be no need to define  $\Phi$  explicitly [12]. By using Lagrange multiplier and kernel method, the QP for nonlinear cases is as below:

$$\begin{cases} \max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \sum_{i=1}^N y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1, \dots, N \end{cases} \quad (4)$$

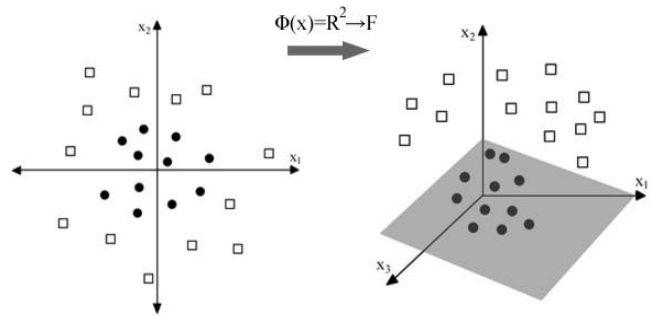


Fig. 1. Transforming non-separable data from original input space to higher dimensional feature space.

## III. BASIC CONCEPT

Population Density is the basic concept which is used to develop Density Based SVM. Population density is the way of measuring the population per unit of area or volume. The term of population density was used by Henry Drury Hames in 1937 for the first time, then widely used to measure the decrement and increment of densities and finally applied as an indicator to compare the area's population density. The concept of population density indicates the relationship between number of population and the occupied space by them.

$$\text{Population density} = \frac{\text{number of population}}{\text{area}} \quad (5)$$

By using this concept, the densely populated and less populated areas can be determined. Considering the training points as the population, those samples placed in less populated areas or the areas with low population densities can be treated as outliers. These outliers are not very important, but dramatically are affecting on performance of SVM algorithm.

Outliers are unusual data points that are inconsistent with other observations. In statistics an outlier is an observation with an abnormal distance from most other observations. Generally presence of an outlier may cause some sort of problems. An outlier may be due to gross measurement error, coding/recording error, and abnormal cases, but a frequent cause of outliers is a mixture of two distributions and they can be occurred by chance in any distribution [13, 14, 15]. There are two strategies to deal with outliers: first, outlier detection or removal as a part of preprocessing; second, developing a robust modeling method to be insensitive to outliers [14, 15]. Density Based SVM is based on the first strategy.

#### IV. DENSITY BASED SVM

The main goals of Density Based SVM is reducing effects of outliers, maximizing margin, providing better generalization, and adjusting the decision boundary according to the density of data sets. Meanwhile Density Based SVM reduces the number of support vectors which decreases computational complexity. It is noteworthy that in Density Based SVM, input vectors are those which are in highest-confidence area of data set and they are more informative than other input vectors.

Density Based SVM can detect outliers or data points which are out of the densely populated area. To detect these outliers, first the densely populated area of a data set should be determined. The data points which are located in the densely populated area will be considered as important (meaningful) points and other as less important (meaningless) which can be misclassified or ignored. Although the concept of population density is used to develop Density Based SVM, the formula is different with (5). In this method the distance (Euclidean & Mahalanobis) between data points of one data set plays the main role to determine the area with high population density.

##### A. Density Based SVM with Euclidean Distance

Euclidean distance measures the distance between two points by formula (6) in Euclidean space [16]. Suppose that a set of 2-dimensional data  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  is given. First the Euclidean distance between all data points of one class should be calculated. For example the Euclidean distance between point 1 and 2, 3, ... , n and the Euclidean distance between point 2 to 1, 3, ..., n and so on.

$$D(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (6)$$

The next step is summing up all distances for each point. For example the total distance for point 1 is  $d_1 = [D(1, 2) + D(1, 3) + \dots + D(1, n)]$  where n is the number of data points in one data set. The total distances for all data points of one data set is needed to calculate the average distance which will be used to determine data points which are inside and outside of densely populated area. The average distance can be calculated as follows:

$$\text{Average}_d = \frac{\sum_{j=1}^N \sum_{i=1}^N \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{n} \quad (7)$$

if  $d_i > \text{Average}_d \rightarrow x_i = \text{outlier}$

After calculating the *Average\_d* by (7), those data points with  $(d \leq \text{Average}_d)$  should go to group 1 which is the new training set and others to group 2. The space which is occupied by group 1 is the area with high population density and data points inside this group will be considered as important data points. Those data points in group 2 will be considered as less important or outliers and they will not contribute in training phase [17].

---

#### Algorithm 1:

---

- 1- For each data point  $x_i$ :
    - Calculate the Euclidean distance between  $x_i$  and all other data points by (6)
    - Sum up all the distances calculated for one point as  $d$
  - 2- Sum up all  $d$  values as  $total\_d$ .
  - 3- Divide  $total\_d$  by number of data points of one set as  $Average\_d$  by (7)
  - 4- Set all data points with  $(d \leq Average\_d)$  in one group
  - 5- New group contains the most important data points and others will be considered as outliers.
- 

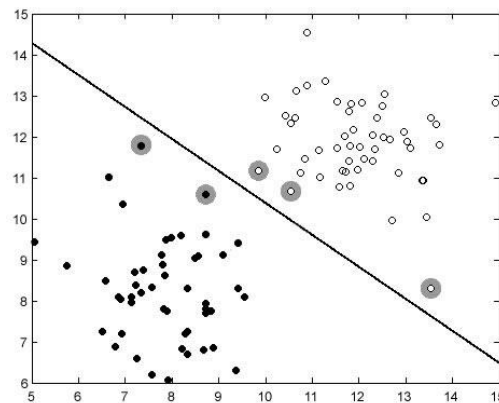


Fig. 2. Result of standard SVM; Outliers exist and margin is small.

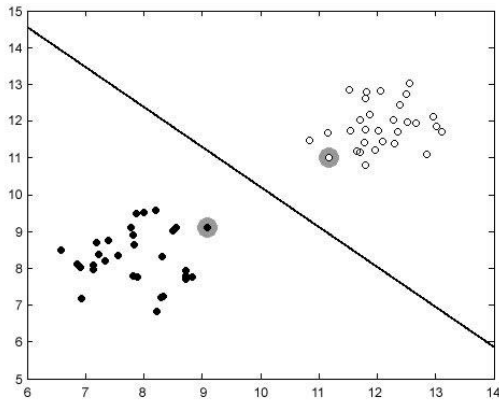


Fig. 3. Result of DBSVM; outliers are removed and margin is bigger.

Applying algorithm 1 as preprocessing on both data sets of the classification problem in Fig .2, can help to detect outliers, reducing the number of support vectors, and maximizing the margin. The difference of margin with and without outliers is shown in Fig .2 and Fig. 3 respectively.

The described algorithm can be applied on both linear and non-linear SVM. In non-linear cases it can be done either in original input space or feature space and there will be no difference in result. In case of applying algorithm in input space, removing outliers from data set will also reduce the dimensionality of data points in feature space and there will be no need to change the algorithm and all should be done like as previous description. However in case of applying mentioned algorithm in feature space, there will be a small difference. Since the kernel matrix should be positive semi-definite and symmetric, after removing outliers it will become asymmetric. In this case by removing each data point, the corresponding column also should be removed. For example point  $x_3$  in below matrix is an outlier, consequently in addition to the 3<sup>rd</sup> row, the 3<sup>rd</sup> column also should be removed from kernel matrix.

$$\begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & k(x_1, x_3) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & k(x_2, x_3) & \dots & k(x_2, x_n) \\ k(x_3, x_1) & k(x_3, x_2) & k(x_3, x_3) & \dots & k(x_3, x_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & k(x_n, x_3) & \dots & k(x_n, x_n) \end{bmatrix}$$

### B. Density Based SVM with Mahalanobis Distance

In this section the Mahalanobis distance will be used instead of Euclidean distance. Euclidean distance measures the distance between two points by formula (6) in Euclidean space. The Mahalanobis distance is the distance from  $x$  to the quantity  $\mu$ . This distance is based on the correlation between variables or the variance-covariance matrix. Mahalanobis distance is unit less and it takes into accounts the correlation

of the data set and does not depend on the scale of measurement [16]. The Mahalanobis distance of point  $x$  to the mean of distribution can be calculated by formula (8) and Mahalanobis distance of point  $x$  to point  $y$  can be calculated by formula (9):

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

$$D_m(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \quad (9)$$

where  $\mu$  is the mean of the distribution and  $S^{-1}$  is the inverse covariance matrix. Here, to determine the densely populated area, the Mahalanobis distance of each point to the mean  $\mu$  of the data set is used. Same as the previous section, the average distance should be calculated and then, those data points with  $(D_m) \leq Average\_D$  should go to group 1 as important points and others to group 2 as outliers.

$$Average\_D = \frac{\sum_{i=1}^N \sqrt{(x_i - \mu)^T S^{-1} (x_i - \mu)}}{n} \quad (10)$$

$$\text{if } D_m > Average\_D \rightarrow x_i = \text{outlier}$$

---

### Algorithm 2:

---

- 1- For each data point  $x_i$ :
    - Calculate the Mahalanobis distance of  $x_i$  to the mean  $\mu$  of data set as  $D$  by (8)
  - 2- Sum up all  $D$  values as  $total\_D$ .
  - 3- Divide  $total\_D$  by number of data points of one set as  $Average\_D$  by (10)
  - 4- Set all data points with  $D \leq Average\_D$  in one group
  - 5- New group contains the most important data points and others will be considered as outliers.
- 

### C. Density Based SVM for Special Cases

So far, the considered data sets had one center and the distribution of data points were around that center. However sometimes data points are distributed very widely and it seems they have more than one center. To deal with this problem, before applying algorithm 1 or 2, the method of K-means clustering should be used to cluster data points and then algorithm 1 or 2 can be applied for each cluster separately.

K-means is one of the most popular clustering algorithms, and it is an iterative descent clustering method. K-means finds  $k$  clusters in a given data set and number of  $k$  should be defined by user. Each cluster is described by a single point called centroid. Centroid means it's at the center of all the data points in a cluster. K-means is a simple algorithm based on

similarity and the measure of similarity plays an important role in the process of clustering [18, 19, 20].

The  $k$ -means algorithm works like this: First  $k$  randomly centroids will be placed, next, each point in the data set will be assigned to the nearest centroid by measuring the Euclidean distance between point and all centroids. After this step, the centroids will be updated by taking the mean value  $\mu$  of all the points assigned to them. This process will be repeated until the assignments stop changing. The result of  $k$ -means depends to two factors: first the value of  $k$ ; second the initial selection of centroids [21, 22, 23].

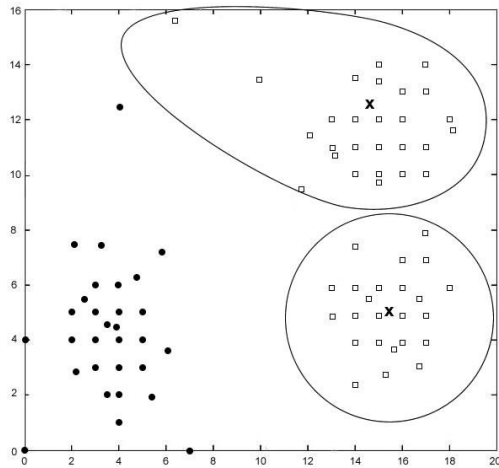


Fig. 4. Result of applying k-means clustering method on one data set.

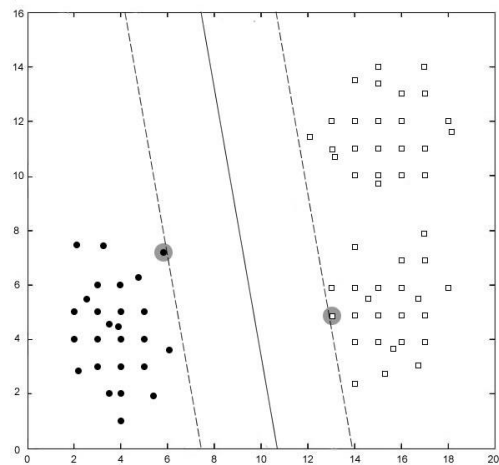


Fig. 5. Result of DBSVM after clustering the widely distributed data set.

### V. EXPERIMENTS & RESULTS

In order to validate the performance of Density Based SVM, two types of Experiments were performed with different data sets of binary classification problems. The first type of experiment was performed using 2 and 3 dimensional artificial data sets and the second type was performed using two high dimensional benchmark data sets and one set of high

dimensional banknote data. The K-fold Cross Validation method is used for all data sets.

#### A. Artificial data classification

The artificial data sets which are used for this experiment are generated at random with normal distribution. These data sets are used with different standard deviations, and below tables represent the results of applying standard SVM and Density Based SVM on them by 2-fold cross validation.

TABLE I. CHARACTERISTICS OF ARTIFICIAL DATA SETS

Linearly separable data sets	# of instances
set1_a & set1_b ( $\mu=8$ & 12, SD=1)	tr=200, ts=200
set1_c & set1_d ( $\mu=8$ & 12, SD=2)	tr=200, ts=200
Non-linear data sets	# of data
set2_a & set2_b ( $\mu=8$ & 12, SD=3)	tr=200, ts=200
set2_c & set2_d ( $\mu=8$ & 12, SD=4)	tr=200, ts=200

TABLE II. LINEAR ARTIFICIAL DATA CLASSIFICATION

Data sets	SVM (Linear)	DBSVM (Linear+Euc)	DBSVM (Linear+Mah a)
set1_a & set1_b	99%	99%	99%
#SV	35	30	30
set1_c & set1_d	80%	83%	83%
#SV	47	36	37

TABLE III. NON-LINEAR ARTIFICIAL DATA CLASSIFICATION

Data sets	SVM (RBF)	DBSVM (RBF+Euc)	DBSVM (RBF+Maha)
set2_a & set2_b	76%	82%	82%
#SV	70	45	48
set2_c & set2_d	62%	68%	70%
#SV	86	50	56

According to above results, Density Based SVM with Euclidean distance can perform better than with Mahalanobis distance on data sets which have smaller standard deviations and are linearly separable or slightly non-linear. However Density Based SVM with Mahalanobis distance performs better on data sets with bigger standard deviation values that are seriously nonlinear.

#### B. Benchmark data classification

Two benchmark data sets are used. They are medical data of Liver Disorder and Heart Disease which are obtained from real life and can be downloaded from the Repository of machine learning databases of the well-known University of California at Irvine (UCI) [25]. These data sets are used by 2-fold and 5-fold cross validations.

a) Liver Disorder Data

TABLE IV. CHARACTERISTICS OF LIVER DISORDER DATA SET

<b>Data set characteristics:</b>	Multivariate
<b>Attribute characteristics:</b>	Categorical, Integer, Real
<b>Number of instances:</b>	345
<b>Number of attributes:</b>	7

TABLE V. LIVER DISORDER CLASSIFICATION BY LINEAR KERNEL

Data sets	SVM (Linear)	DBSVM (Linear+Euc)	DBSVM (Linear+Maha)
<b>Liver Disorder (2-fold)</b>	66.9%	70.8%	69.7%
<b>#SV</b>	127	83	70
<b>Liver Disorder (5-fold)</b>	68.5%	70.5%	71.6%
<b>#SV</b>	200	125	122

TABLE VI. LIVER DISORDER CLASSIFICATION BY POLYNOMIAL KERNEL

Data sets	SVM (Poly)	DBSVM (Poly+Euc)	DBSVM (Poly+Maha)
<b>Liver Disorder (2-fold)</b>	57.5%	59.12%	57.7%
<b>#SV</b>	120	85	90
<b>Liver Disorder (5-fold)</b>	63.69 %	63.69%	67.73%
<b>#SV</b>	210	135	124

TABLE VII. LIVER DISORDER CLASSIFICATION BY RBF KERNEL

Data sets	SVM (RBF)	DBSVM (RBF+Euc)	DBSVM (RBF+Maha)
<b>Liver Disorder (2-fold)</b>	60.3%	60.3%	60.3%
<b>#SV</b>	172	131	115
<b>Liver Disorder (5-fold)</b>	57.5%	58%	58%
<b>#SV</b>	225	160	150

b) Heart Disease Data

TABLE VIII. CHARACTERISTICS OF HEART DISEASE DATA SET

<b>Data set characteristics:</b>	Multivariate
<b>Attribute characteristics:</b>	Categorical, Real
<b>Number of instances:</b>	270
<b>Number of attributes:</b>	13

TABLE IX. HEART DISEASE CLASSIFICATION BY POLYNOMIAL KERNEL

Data sets	SVM (Poly)	DBSVM (Poly+Euc)	DBSVM (Poly+Maha)
<b>Heart Disease (2-fold)</b>	72 %	71%	79.99%
<b>#SV</b>	86	50	30
<b>Heart Disease (5-fold)</b>	70.36 %	70.36%	79.99%
<b>#SV</b>	86	45	30

TABLE X. HEART DISEASE CLASSIFICATION BY RBF KERNEL

Data sets	SVM (RBF)	DBSVM (RBF+Euc)	DBSVM (RBF+Maha)
<b>Heart Disease (2-fold)</b>	60.6 %	60.6%	60.6%
<b>#SV</b>	135	85	75
<b>Heart Disease (5-fold)</b>	60%	60%	60%
<b>#SV</b>	96	50	58

C. New and Fatigued Banknote classification

To classify the new and fatigued banknotes, two sets of acoustic signals of new and fatigued U.S. one dollar banknote which are recorded by measurement system of acoustic signal are used for both training and testing. In this case the amplitude differences are considered as the characteristic value. The acoustic signal sets are mapped from very high-dimensional to four-dimensional data. Steps for converting data to four-dimensional are as follows [26]:

Step 1. Calculating the amplitude difference from the sample data of forward and backward (see Fig.6).

Step 2. Assigning each calculated data to the horizontal and vertical axes.

Step 3. Converting from Cartesian coordinate into the polar coordinate; all elements are divided by the fan-shaped domain.

Step 4. Number of elements which are distributed over each domain gives a four-dimensional data for each banknote acoustic signal (see Fig. 7).

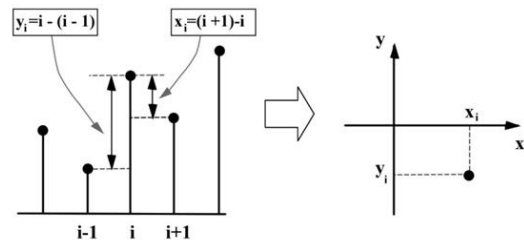


Fig. 6. Calculating amplitude difference from sample data of forward and backward.

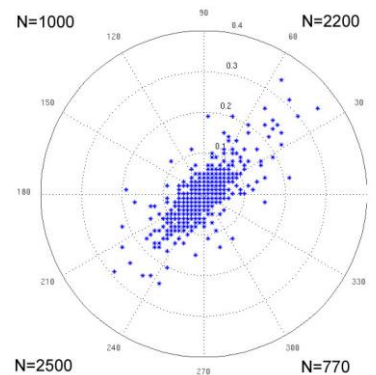


Fig. 7. Sample of data in polar coordinate 4-dim  $x_i = (2200, 1000, 2500, 770)$

The data set which is used for this experiment contains 48 four-dimensional data of new banknotes and 128 four-dimensional data of fatigued banknotes in four different levels. The data set is divided at random into two disjoint subset of the same size for 2-fold cross validation. Results of experiments with standard SVM with different kernels and Density Based SVM (Euclidean distance & Mahalanobis distance) are shown in following tables.

TABLE XI. CHARACTERISTICS OF BANKNOTE DATA SETS

Banknote Data Sets	# of instances	# of attributes
S00. New banknote	48	4
S01. Fatigued banknote (level 1)	32	4
S02. Fatigued banknote (level 2)	32	4
S03. Fatigued banknote (level 3)	32	4
S04. Fatigued banknote (level 4)	32	4

TABLE XII. BANKNOTE DATA CLASSIFICATION BY LINEAR KERNEL

Banknote Data	SVM (Linear)	DBSVM (Linear+Eu c)	DBSVM (Linear+Maha )
Average Result of Binary Classification	66.77%	72.55%	69%
#SV	17	9	12
Multiclass Classification	39.56%	45%	42%
#SV	55	30	45

TABLE XIII. BANKNOTE DATA CLASSIFICATION BY POLYNOMIAL KERNEL

Banknote Data	SVM (Polynomial)	DBSVM (Poly+Eu c)	DBSVM (Poly+Maha )
Average Result of Binary Classification	63.68%	68.85%	65%
#SV	13	8	10
Multiclass Classification	38%	43%	37.5%
#SV	50	32	36

TABLE XIV. BANKNOTE DATA CLASSIFICATION BY SPLINE KERNEL

Banknote Data	SVM (Spline)	DBSVM (Spline+Eu c)	DBSVM (Spline+Maha )
Average Result of Binary Classification	62.24%	65%	63.83%
#SV	17	10	11

TABLE XV. BANKNOTE DATA CLASSIFICATION BY RBF KERNEL

Banknote Data	SVM (RBF)	DBSVM (RBF+Eu c)	DBSVM (RBF+Maha )
Average Result of Binary Classification	69%	70.5%	69%
#SV	32	20	20

Banknote Data	SVM (RBF)	DBSVM (RBF+Eu c)	DBSVM (RBF+Maha )
Multiclass Classification	35%	35%	35%
#SV	75	45	48

According to the results of different types of experiments on artificial data sets, benchmark data sets and banknote data sets, it can be claimed that Density Based SVM can provide better generalization ability, reduces the effects of outliers, and it can decrease the number of support vectors. Number of support vectors has a direct influence on the time required to evaluate the SVM decision function and also on the time required to train the SVM.

Considering the results presented in previous tables, algorithm 1 can be useful for linearly separable and slightly overlapping classes, and algorithm 2 can be useful for those classes with considerable overlapping (seriously nonlinear).

## VI. CONCLUSION

In this paper, the new method of Density Based Support Vector Machines is introduced. Density Based SVM tries to decrease the effects of outliers on SVM performance. The basic concept which is used in this method is population density. By using this concept, the densely populated area of each data set can be found. Those data points which are inside this area are located in highest confidence area of data set and will be considered as most important points and others as outliers. To find this area, two algorithms are proposed; algorithm 1 uses Euclidean distance and algorithm 2 uses Mahalanobis distance.

SVM finds the optimal separating hyperplane under the effects of outliers, but this method first removes outliers as a preprocessing and adjusts the separating hyperplane/decision boundary according to the density of data sets. Support vectors in Density Based SVM are from high confidence area of data set. Although the main goal of Density Based SVM is removing outliers, it is also maximizing margin, reducing number of support vectors which results reducing computational complexity and gives better generalization ability. Different experiments on artificial data sets and real high dimensional data sets are performed to prove the validity of this method. Considering the results of experiments, the Density Based SVM can be useful on different types of noisy data sets. It increases the SVM performance and considerably reduces number of support vectors.

The future work to be done is to make some changes in this method to become more effective on RBF kernels. Because according to the results of experiments, Density Based SVM only reduces the number of support vectors and computational complexity, but does not increase the generalization ability while using RBF kernel.

## ACKNOWLEDGMENT

The authors would like to thank Mr. Ikugo Mitsui, Mr. Nobuo Shoji, and Mr. Kenji Ozawa from Japan Cash Machine

Co., Ltd, for their support and providing the measurement system of acoustic signals for experiment C.

REFERENCES

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer 2000.
- [2] V. Kecman, *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, The MIT Press 2001.
- [3] X. Zhang, *Using Class Center Vectors to Build Support Vector Machines*, IEEE, pp.4-6, 1999.
- [4] R. Herbrich, J. Watson, *Adaptive Margin Support Vector Machines for Classification*, Microsoft Research, pp. 2-4, 1999.
- [5] AB. A. Graf, *Classification in a Normalized Feature Space Using Support Vector Machines*, IEEE, pp. 1-3, 2003.
- [6] Q. Song, *Robust Support Vector Machine with Bullet Hole Image Classification*, IEEE, pp. 3-4, 2002.
- [7] C. F. Lin, *Fuzzy Support Vector Machines*, IEEE, pp. 3-4, 2002.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.
- [9] A. R. Webb and K. D. Copsey, *Statistical Pattern Recognition*, John Wiley & Sons 2011.
- [10] V. N. Vapnik, *Statistical Learning Theory*, AT&T Research Laboratories, John Wiley & Sons, 1998.
- [11] S. Abe, *Support Vector Machines for Pattern Classification*, Springer 2010.
- [12] B. Scholkopf & A. J. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization & Beyond*, The MIT Press 2002.
- [13] V. Cherkassky and F. Mulier, *Learning from Data, Concepts, Theory and Methods*, IEEE Press 2007.
- [14] D. Ripley, *Robust Statistics*, M.Sc. in Applied Statistics MT2004.
- [15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press 1999.
- [16] Y. Dodge, *The Concise Encyclopedia of Statistics*, Springer 2008.
- [17] Z. Nazari, D. Kang, and H. Endo, *Density Based Support Vector Machines*, The 29<sup>th</sup> International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), pp.1-3, 2014.
- [18] P. Cichosz, *Data Mining Algorithms Explained Using R*, John Wiley & Sons 2015.
- [19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, data mining, inference and prediction*, Springer 2009.
- [20] P. Harrington, *Machine Learning in Action*, Manning Publications Co 2012.
- [21] S. Marsland, *Machine Learning an Algorithmic Perspective*, CRC Press 2009.
- [22] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press 2010.
- [23] J. Bell, *Machine Learning Hands-on for Developers and Technical Professionals*, John Wiley & Sons 2015.
- [24] T. Segaran, *Programming Collective Intelligence*, O'REILLY Media. Inc 2007.
- [25] *Machine Learning Repository*, available online at: (<https://archive.ics.uci.edu/ml/datasets/Liver+Disorders>).
- [26] M. Higa, D. Kang, H. Miagi, *Classification of Fatigue Bill based on Acoustic Signals*, pp. 1-4, 2012.