Support Vector Machine: Influence of the Standard Deviation, Number of data and Cost on a model

Cristian V. Montano
School of Electrical and Computer Engineering
University of New Mexico, USA

***Abstract*—Support vector machines (SVM) is a widely used learning technique applied to classify data samples in many statical learning processes. There are tutorials on SVM construction, applications to various projects, combination with other techniques and a lot of complementary information regarding them for specific uses. However, little research has been done in order to comprehend their classification performance respect to the parameters of construction. This paper consists on building SVMs for sets of data with gaussian distribution in order to make tests for scaled values of sigma, cost and quantity of data, each one independently studied through graphs. The experiments realized expose in a consistent way the performance of the SVM respect to its parameters, the directions where the risk increase or decrease and the magnitude of change.**

I. Introduction

In order to construct SVMs, it is necessary to count with labeled data (the training set) and unlabeled data (the working set). The objective is to assign class labels to the working set such that the best support vector machine (SVM) is constructed first, and later tested in order to measure its performance [1]. The SVM was developed in Russia around the sixties [2, 3]. It is firmly stablished in the area of statical learning theory, or VC theory, which has been studied over the last three decades by Vapnik and Chervonenkis. In a synopsis, VC theory characterizes properties of learning machines which let them predict unseen data. The SVM was developed at AT&T Bell Laboratories by Vapnik and co-workers. Due to practical applications, the SVM investigation has a wide space were can be applied. While there are several studies interested on classification, complementary tools and better ways to improve the SVM modeling, this paper focuses on the components that take part on the construction in order to have a better comprehension. The modeling of SVMs is tested for scaled values of standard deviation, cost and quantity of data, each one independently analized. Each parameter is studied in a specific experiment in order to expose the influence that each parameter cause in the SMV for values concerning the wide spectrum of possibilities. At the time of constructing a SMV classifier, the understanding about the behavior of the model is collected in an abstract way, which derives from logical understanding of the topic. This knowledge about SVM construction is very useful to make an appropriate construction of the SVM. For instance, the purpose of this paper is to make the knowledge of SVM parameters smoother through a deep analysis of the role each constructing parameter contributes. For this reason, this paper builds experiments that expose, in a wide scaled range of values, the influence of the parameters of construction on SVM itself, specifically, the standard deviation, cost and quantity of data. Several tests are done for each parameter and all of them summarized on a graph in order to have a better comprehension.

II. Support Vector Machines

A review of basic principles of SVMs for data classification will be covered in this section. Let's consider $\{(x_i, y_i), i = 1...N\}$ as a training samples set. The ith example $x_i \in R^n$ in an n-dimension input space belongs to one of two classes labeled by $y_i \in \{-1, +1\}$ [4]. The objetive of the SVM is defining a hyperplane in a high-dimensional space, which separates a samples set in the space such that all the points that have the same label are on the same side of the hyperplane.Consequently, we train a SVM in this paper to find w and b so that

$$f(x) = (w^T x + b) \tag{1}$$

where w is the weights vector, and the bias of the hyperplane is represented by b. In this feature space, an optimal separating hyperplane (OSH) that maximizes the margin between the two closest orthogonal vectors to the hyperplanes is constructed. The classification is delimited by the hyperplane in a D-dimensional space is built as

$$y_i[w^T x + b] \geq 1, i = 1...N \tag{2}$$

Among the separating hyperplanes, the one with the maximal distance to the closest point is called the optimal separating hyperplane (OSH), which will result in an optimal generalization. In view of the fact that the distance to the closest point is $1/ \parallel w \parallel$, both parts of the distance are the margin. The margin is the measure of the ability of this hyperplane classification. The greater the margin, the better the generalization will be. Consequently, the OSH is the hyperplane that separates with the maximum magnitude the margin, and obtains the best results. Then, the optimal OSH is reached when $\parallel w \parallel^2$ is miniized subject to the constraint (2). Differently, for data that is non-separable, slack variables are needed, for this reason we introduce ε_i [5]

$$\{ y_i [w^T x + b] \geq 1 - \varepsilon_i, i = 1...N. \tag{3}$$

This method gives the chance to samples violate (2). The necessity of the nonnegative slack variables ε_i is to allow misclassified points to be counted. When the ith example is misclassified by the hyperplane, the corresponding value of $\varepsilon_i \geq 1$. Consequently, $\Sigma_i \varepsilon_i$ is the of upper bound on the quantity of misclassified samples, and the minimization leads to a reduction of the empirical training error. Following the structural risk minimization inductive principle, the SVM approach is to minimize the risk bound as shown:

$$minimize\ L_p(w, \varepsilon_i) = \frac{1}{2} \parallel w \parallel^2 + c\Sigma_i \varepsilon_i \tag{4}$$

subject to (3). Based on [1] and (4), it is known that the first term of (4) is reduced to the minimum in order to control the VC dimension for the class of learning machines, for instance the learning capacity of the learning machines. It also amounts to maximizing the margin of a separating hyperplane in the feature space. The second term of the equation manages the number of misclassified samples, in other words, this controls the empirical risk term of the guaranteed risk. The positive constant c is required to be set up before solving (4). A higher value of c means a greater penalty is assigned to empirical errors. This minimization is resumed on a structural risk minimization for a set of functions if a proper positive constant c is chosen. [3, 5]

III. Vapnik-Chervonenkis dimension

Consider some set of m points in R^D and choose any one of the points as origin, and Then the m points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent [6]. The maximum number of vectors that can be shattered by a hyperplane in a space R^D is $D + 1$, since we can always choose $D + 1$ points, and then choose one of the points as origin, such that the position vectors of the remaining D points are linearly independent but can never choose $D + 2$ such points since no $D + 1$ vectors in R^D can be linearly independent. Consider some set of m points in R^D. Choose any one of the points as origin. Then the m points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent. The maximum number of vectors that can be shattered by a hyperplane is called the VC dimension [1]. The VC dimension also gives a measure of the complexity of linear functions. If the VC dimension of an estimator is higher than the number of vectors to be classified, then the estimator is guaranteed to overfit if an empirical risk is minimized over the data, since all vectors will be correctly classified regardless of their statistical properties. The linear empirical risk [6] is defined as:

$$R_{emp}(\alpha) = \frac{1}{2N} \sum_{n=1}^{N} \mid y - f(x, \alpha) \mid \tag{5}$$

where $f(x, \alpha)$ is defined so that the loss function $\mid y - f(x, \alpha) \mid$ can only take the values 0 or 1. Then, with probability $1 - \eta$ the following bound holds. [6]

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(log(2N/h) + 1) - log(\eta/4)}{N}} \tag{6}$$

IV. SVM construction

To illustrate the method of the paper, I realize SVMs construction and analysis with the use of Matlab libraries svmtrain and smvpredict. The data to be classified is generally given or collected, but for the experiments in this paper, the data is generated. This is going to allow the test count with the necessary number of samples and as many times as required.

A. *Standard deviation on SVM*

For the standard deviation analysis, 4 Gaussians around four different centroids are generated with samples are randomly taken, all of them independent and identically distributed. All data samples are stored in an input matrix X with size N x D (where N is the number of training vectors and D is the dimension of the space). The chosen centroids for the Gaussian samples are $\{1\ 1;\ 2\ 1,5;\ 2\ 1;\ 3\ 1,5\}$. These four coordinates form a rhomboid in the plane. The use of a Rhomboid corners make possible a proper classification of the data and it also gives a clear way to comprehend the manner the classifier is working. For this experiment, the data has two different labels. The data for the two first centroids $\{1\ 1;\ 2\ 1,5\}$ correspond to the first label, and the data related to the other two centroids $\{2\ 1;\ 3\ 1,5\}$, correspond to the second label. Consequently, the data has two to labels distributed around four different centroid and can be classified with a simple straight line. All the samples generated around the centroid have the same standard deviation (σ) in order to ease and focus the comprehension of the experiment on the SVM construction. The figures 1, 2, 3 and 4 show with the symbol '+' the first two centroids of the data, and with the symbol 'o' the other two.

To illustrate the method of the paper, I construct an SVM to solve the two-class pattern classification problem for the data with the characteristics previously mentioned.

The construction of the SVM is done with the use of library svmtrain from Matlab. Once the centroids are defined, 10 samples for each centroid were delimited, hence there are forty samples to be classified. The data sample quantity is chosen to ease the graph comprehension. Next, it is necessary to assign a standard deviation to the samples. Here I make the construction of four models, each one with a different value of the standard deviation. The reason for this is to analyze the behavior of the classifier, in one hand, when the samples have a low dispersion from the centroid with a low possibly of overlapping with other label samples, and in the other, when the samples have higher dispersion greater possibility of overlapping with other label samples. For each SVM construction the values for sigma are 0,1, 0,2, 0,4, and 0,6 respectively. First, the sample data generated is stored on a Matrix X. This matrix contains information corresponding to each label of the data stored on each column, hence as generated data for a two-class pattern, the experiment needs

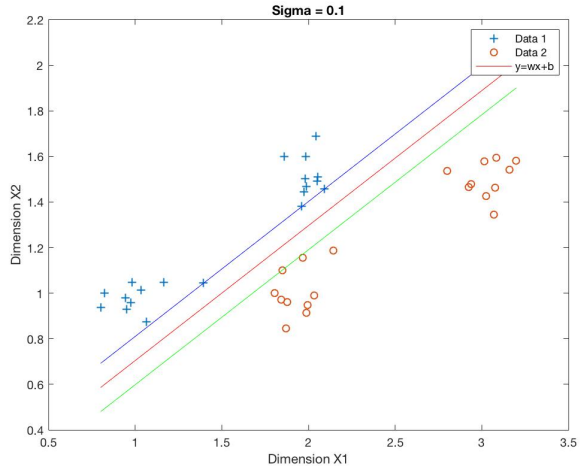

Fig. 1: 100% samples properly classified for $\sigma = 0.1$

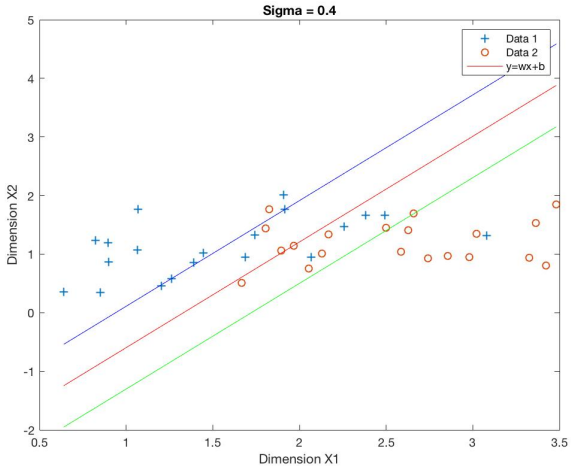

Fig. 3: 80% samples properly classified for $\sigma = 0.4$

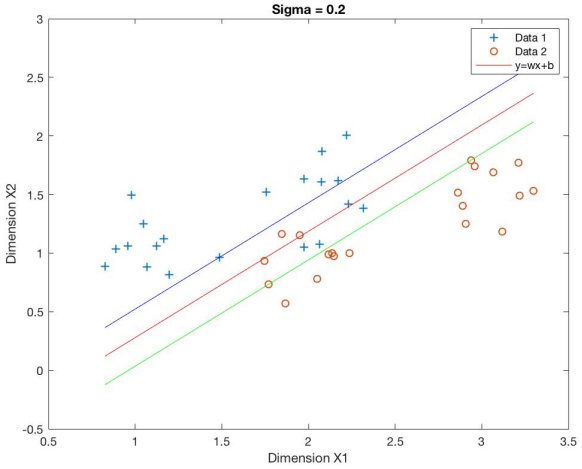

Fig. 2: 87,5% samples properly classified for $\sigma = 0.2$

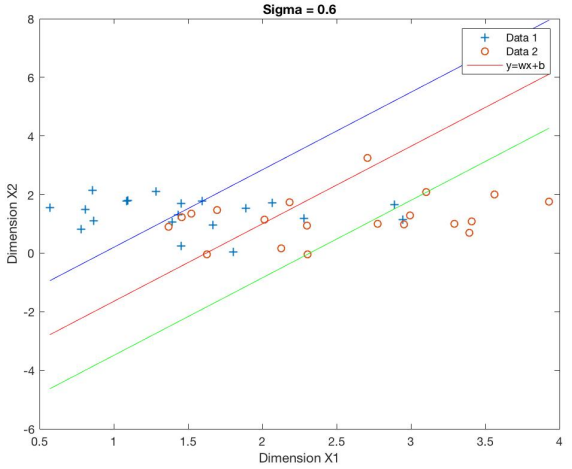

Fig. 4: 72,5% samples properly classified for $\sigma = 0.6$

two columns on matrix X. In the same way, each data sample requires needs to be stored on a row of the matrix X, and therefore, for a 40-sample experiment 40 rows are needed. Consequently, X is a 40x2 matrix. Next, it is necessary to label the stored data to construct the SVM. In this experiment, each data sample is labeled with -1 or 1 depending the label they belong to. For this reason, a vector Y is necessary to store the label for each data sample. Thus, Y is a vector of containing 40 elements. According matrix X, the first 20 samples belong to one label, and the second 20 samples belong to the other label. Thus, the 20 first elements of vector Y contain a value of 1, and the following 20 contain a value -1. Matlab library svmtrain is applied to build a linear model (-t 0), classifier (-s 0) and with a cost of 100 (-c 100). The model is constructed with the data inputs (X) and labels (Y). Correspondently, it gives as a return the bias stored in the variable rho, and w can be computed by multiplying $model.SVs'$ with $model.sv\_coef$. Both variables are obtained from svmtrain once the model is constructed. The behavior of the model for each standard deviation is depicted on figure 1, figure 2, figure 3 and figure 4 correspondently. For the SVM constructed in figure 1, there is no overlap in the data samples with $\sigma = 0,1$, therefore, the SVM classifies correctly all samples generated. In figure 2 the data samples are more disperse, and both labels get mixed; as a result the SVM classifies correctly only 87,5% for $\sigma = 0,2$. In the figure 3 can be observed that the misclassified samples increases for $\sigma = 0,4$ and the high dispersion of the data allows the model make a correct classification for 80% of the samples. Finally, figure 4 depicts a correct classification for 72,5% of the samples, in this case $\sigma = 0,6$. It is possible to infer from the graphs that the standard deviation an important role on the SVM effectiveness. It is needed a very low standard deviation in order to have 100% of data classification, and as the standard deviation increases, the overlap of data causes a reduction on the model accuracy.

V. SVM FUNCTION OF THE COST AND NUMBER OF DATA

In the second experiment, the objective is to plot the behavior of the SVM in function of the number of data in one hand, and cost in the other hand.

A. *Cost value test*

In order to build the data, I generated eight points describing a cube of dimension σ. Four of the points were at one side of the plane described by the equation $f(x) = (w^T x + b)$ where $w_i = 1/\sqrt{10}$ and $b = 0$, at a distance $\sigma/2$, and they were labelled with -1. The other four, at the other side of the plane and be labelled as +1. Each pattern is generated by choosing at random one of the eight points and then adding a 10-dimensional Gaussian noise of standard deviation $0, 2\sigma$. One hundred data samples are generated with a standard deviation equal to $\sigma/2$. Then it is necessary to build the model. For the construction of the model, the library svmtrain was used. The constructed model is linear (-t 0), classifier (-s 0) and with with 100 different values for the cost, ranging from $10^{1,5}$ to 10 in a logarithmic scale generated by the Matlab function logspace(-1.5,1,100). I used the logarithmic scale because it gives a plot easy to comprehend respect to error analysis. In order to compute the empirical error, the function smvpredict of Matlab is used in this experiment. The function is needed two times: first, in order to compute the training error, and second, to compute the test error. For consistent results, I made the computation of the error for each value of the cost 1000 times and the took the mean. The difference of of both error are also depicted on the figure 5.

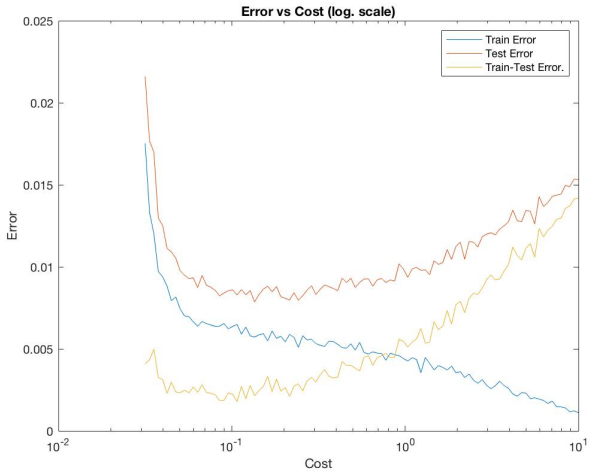

Fig. 5: SVM Error function of Cost values

From the graph it is possible to observe that the training error starts with 1,75x10^{-2} for a cost = $10^{-1,5}$ and it is the higher value reached. Starting there, it has an abrupt decrease until it reaches 7x10^{-3} at a cost around 0,1. From that point, as the value of the cost increases the error keeps decreasing, however, in a gradual way, until it reaches 2,25x10^{-3} when the cost is 10.

The testing error, in the same way, starts with the higher average value of 2,25x10^{-2} for a cost = $10^{-1,5}$ and has an abrupt decrease until it reaches a minimum around 8x10^{-3} for a cost value between 0,07 and 0,2. From that point, as the value of the cost gets higher than 0,1, the testing error increases, however, in a gradual way, until it reaches 0,015 when the cost is 10.

From the difference of both the training and testing error, it is possible to recognize that both have a high similitude for a cost value interval value between 0,05 and 0,2; and the difference increases as the cost gets far from there.

B. *Data sample quantity test*

For this experiment, the same random data generating function mentioned in subsection A is used. The objective in this section is to obtain a graph where the actual risk and the empirical risk can be analyzed respect to an scaled quantity of data. In order to train the SVM, I used a standard deviation of 0,3, a linear model (-t 0), classifier (-s 0) and with a cost of 1 (-c 1). This values ease the model understanding. I made the training vector classification for 50 different amounts of data, between the range 10 to 500, with a difference of 10 units one to another. To make the computation of the risk, the function smvpredict is used two times: first, to compute the training risk, and second, to compute the testing risk. Consistent results are obtained when the calculation is done 1000 times for the training and the testing error respectively, and ploted the mean. The difference of both is also exposed in the figure 6

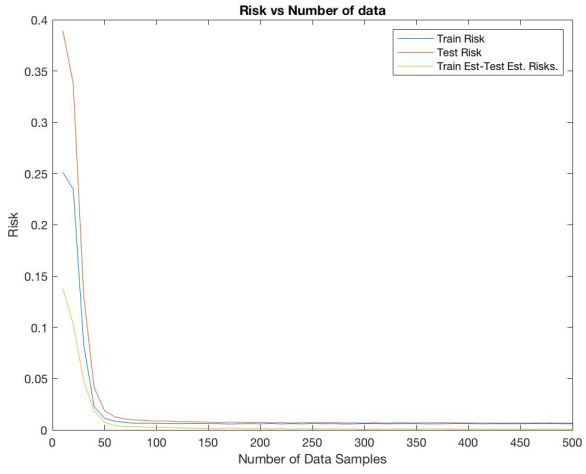

Fig. 6: SVM Risk function of Number of Data

It is possible to distinguish from the figure 6 that as the number of data increases the risk decreases, for this reason, the model gets more accurate. However, two segments are very notorious: first, a fast risk decrease for a quantity of data minor that 50, and the model establishes with higher values; that means, no major change as the number of data increases more than certain point.

VI. Conclusions

This paper exposes the behavior of an SVM classifier when constructed with many different values of standard deviation, number of data samples and cost. Each parameter is independently analyzed on specific experiments thru graphical representation.

For the first experiment, an SVM is constructed with four different standard deviation in a two class data situation. A graph of the SVM construction for data samples is obtained for each standard deviation. From the graphs, the accuracy of the classifier for each case is computed and expressed thru percentage of assertion. In some experiments, the SVM is built with totally separated data, while in others, the data is overlapped as a consequence of a high standard deviation.

Also, in this paper it is presented a graphical representation of the cost influence respect to the training and testing risk in SVM's construction. A test for various values of the cost is done and the empirical and structural risk is computed and graphed. An analysis based on the influence of the cost to the empirical and structural risk derives from the graph for this experiment.

Similarly, an experiment to represent the influence of the number of data samples when constructing a SVM is done. For this experiment only value to test is the number of data. For each test with an determinate number of data, the empirical and structural risk is computed. As a result, a graph of empirical and structural risk vs number of data samples is obtained. An analysis concerning the influence of the cost respect to the empirical and structural risk derives from the obtained graph for this experiment. [1]

References

[1] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, NY, USA, 2001, vol. 1, no. 10.

[2] V. Vapnik and A. Y. Lerner, "Recognition of patterns with help of generalized portraits," *Avtomat. i Telemekh*, vol. 24, no. 6, pp. 774–780, 1963.

[3] V. Vapnik and A. Chervonenkis, "On a perceptron class," *Automation and Remote Control*, vol. 25, pp. 112–120, 1964.

[4] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[6] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.