Support Vector Machine for Linear Classifier

Susan Sapkota[1]

[†]Department of Physics, University of New Mexico

***Abstract*—We introduce idea of Support Vector Machine.Then We perform experiment to classify two dimensional and multidimensional data using linear classifier for identifying how training error and test error changes with number of data, cost parameter (c) and standard deviation (σ) of data. we found out classifier is best with low σ and enough c(neither too high nor too low).**

***Index Terms*—Support Vector Machines, Empirical error, Structural error,Linear Classifier**

I. Introduction

A Support Vector Machine(SVMs) is machine learning algorithm first introduced by [1] in 1963 that used supervised learning method to classify data and sorts into distinct categories. The supervised learning method includes formulation of classifier using training data set by using learning algorithm. The classifier is utilized to classify test data using linear function. [3] developed the idea of non-linear classifier with the help of kernal functions. SVMs have been used on variety of task like pattern recognition, text categorization, image classification and protein classification. The motivation of this paper is to introduce the concept of Support Vector Machines with theoretical background and explain the experiments and results to see how the linear classifier behaves on changing various parameters such as cost parameter and number of datasets. The paper is well structure with theory, experiments and result and discussion of the result.

II. Theory

A. Risk of SVM

Consider m number of observation with pair of data point with unknown probability distribution F(x,y) on plane, vector $\mathbf{x}_n$ where n=1,2,3...k with label y_n which classifies the data in two group. Suppose a machine whose task is to learn the mapping $\mathbf{x}_n \rightarrow y_n$. One can define deterministic machine with adjustable parameters α as $\mathbf{x} \rightarrow f(x,\alpha)$. Trained machine f($\mathbf{x}$,$\alpha$) can be generated with particular α to estimate y with given $\mathbf{x}$ and $f(x)$ [4]. $y - f(x,\alpha)$ is the error or discrepancy between output $f(x,\alpha)$and label y. The expectation of test error (risk) for trained machine can be defined as,

$$R(\alpha) = \int \frac{1}{2}|y - f(x,\alpha)|dF(\mathbf{x},y) \tag{1}$$

We cannot compute the risk in this case because probability distribution is unknown. However, we can approximate the risk defining empirical risk [5]. The empirical risk is defined as measured mean error rate on training set given by,

$$R_{emp}(\alpha) = \frac{1}{2}\sum_{n=1}^{m}|y_n - f(\mathbf{x}_n,\alpha)| \tag{2}$$

$\frac{1}{2}|y_n - f(\mathbf{x}_n,\alpha)|$ is called loss function which takes the value between 0 and 1.

B. Vapnik-Chervonenkis Theorem

We must introduce topic of Vapnik Chervonenkis Dimension (VC Dimension) to find risk. Let us consider m points in R^n. Vapnik theorem states that m points can be shattered by oriented hyperplanes with one point as origin if and only if position vector of remaining points are linearly independent. i.e. The maximum number of vector that can be shattered by a hyperplane in R^n is n+1 called VC dimension [1]. In our case, we have n points with label 2^n to classify into +1 or -1 by linear function. The linear function can shatter three points in two dimension.However, some of the orientation of 2^n cannot be shatter by linear function like 3 points in line but it doesnt create any difficulties as we can find orientation such that all combination can be shattered by linear classifier. The VC dimension in two dimension is 3. In two dimension data, if we take more than 3 points then it cannot be shattered by linear classifier.

Let us take small value η such that $0 \leq \eta \leq 1$ with probability of $1 - \eta$ for loss function. The risk is maximum bounded and given by Vapnik theorem as:-

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(log(\frac{2m}{h})+1) - log(\frac{\eta}{4})}{m}} \tag{3}$$

The second term of the right side is the structural risk (R_s). m is number of points. The left side of eq.(3) is not computable while right side can be computed with knowledge of complexity h. The structural risk is independent of probability distribution. We can minimize the second term of the right hand side of eq.(3) to find lowest upper bound on the risk. The minimization of this risk can lead us to over-fitting or under fitting when estimation function is not complex enough. for case of training data set with linear classifier, the complexity is low which result in high bias. instead of linear, if we choose higher order polynomial like cubic the classifier might exactly classify data with low bias. On tuning complexity, risk might increase. There exit a point called optimal where both structural risk and empirical risk for the risk vs complexity is minimum. The optimal solution leads to neither over-fitting nor under fitting.

C. Dual SVM solution

Let us consider linear machine trained on separable data. The training data be ($\mathbf{x}_n$,y_n)where n=1,2,3,....m and y_n be -1 or 1, $\mathbf{x}_n \in \mathrm{R}^d$. suppose we have hyperplane (H) which is

just a linear line that separates positive (x+) and negative (x-) sample. The points lying on hyperplane satisfy the following equation.

$$\mathbf{x}.\mathbf{w} + b = 0 \tag{4}$$

where **w** is normal vector to hyperplane (line). $|b|/||\mathbf{w}||$ is the perpendicular distance from line to origin and $||\mathbf{b}||$ is the norm. Let d be the distance between two seperating hyperplane (H). The points must satisfy following condition.

$$\mathbf{x}_n.\mathbf{w} + b \geq +1 \ - for\, y_n = +1 \tag{5}$$

$$\mathbf{x}_n.\mathbf{w} + b \leq -1 \ - for\, y_n = -1 \tag{6}$$

The equation 5 and 6 can be combined as:-

$$y_n(\mathbf{x}_n.\mathbf{w} + b) - 1 \geq 0 \tag{7}$$

Now define two hyperplane as H_1: $\mathbf{x}_n.\mathbf{w}+b = 1$ and H_2: $\mathbf{x}_n.\mathbf{w}+b = -1$. H_1 and H_2 are parallel lines. Margin (d) is defined as the distance between H_1 and H. let us take $\mathbf{x}_0$ on H such that $\mathbf{w}^T\mathbf{x}_0$ + b = 0. Now we can trace perpendicular line to plane H_1 $\mathbf{x} = \mathbf{x}_0 + \rho\ \mathbf{w}$. One can use property of transpose matrix and write $\mathbf{x}.\mathbf{w}=\mathbf{w}^T\mathbf{x}$. Then intersection with the H_1 is point $\mathbf{x}_1$ such that we get two equation as follows

$$\mathbf{w}^T\mathbf{x}_1 + b = 1$$

$$\mathbf{x}_1 = \mathbf{x}_0 + \rho\mathbf{w}$$

The distance is d = $||\mathbf{x}_1 - \mathbf{x}_0||$ = $\rho||\mathbf{w}||$. Now using above equation, we get

$$\mathbf{w}^T(\mathbf{x}_0 + \rho\mathbf{w}) + b = 1$$

$$\rho||\mathbf{w}||^2 = 1$$

since $\mathbf{w}^T\mathbf{x}_0$ + b = 0 from H line. using distance formula d we get,

$$d = \frac{1}{||\mathbf{w}||} \tag{8}$$

The main aim is to maximize the margin which same as minimizing the norm of the parameter subject to constraint given by

$$\text{minimize}_w\ ||w||^2 \tag{9}$$

subject to condition given by equation 7. This case is for the data that can be seperated completely where empirical risk is just zero. Now we have to introduce the slack variable ξ_n introduce by [2] in 1995. Let us consider a non-separable dataset for slack variable $\xi_n \geq 0$. The empirical risk can be minimized by $R_{emp}(w) = \sum_{n=1}^{m} \xi_n$. Our minimizing condition is as follows:-

$$\text{minimize}_w\ ||w||^2 + C\sum_{n=1}^{m} \xi_n \tag{10}$$

$$\text{subject to } y_n(\mathbf{x}_n.\mathbf{w} + b) \geq 1 - \xi_n \tag{11}$$

where C is cost parameter which gives information of miss classification. Now Lagrange for our optimization in primal (**w**,ξ_n) form is given as:-

$$L = \frac{||w||^2}{2} + C\sum_{n=1}^{m} \xi_n - \sum_{n=1}^{m} \alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 + \xi_n) - \sum_{n=1}^{m} \mu\xi_n \tag{12}$$

subject to $\alpha_n, \mu \geq 0$. Now we can start taking gradient of equation 12 with respect to primal **w**,b and ξ. For the optimal value, all the gradient is zero. Gradient with respect to w yields **w**=$\sum_{n=1}^{m} \alpha_n y_n \mathbf{x}_n$. Gradient with respect to b yields $C = \alpha_n + \mu$. Gradient with respect to ξ yields $\sum_{n=1}^{m} \alpha_n y_n = 0$ with complementarity property over constraints as $\mu\xi_n = 0$ and $\alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 + \xi_n) = 0$. Then we can write our estimator $y_k = \mathbf{w}^T\mathbf{x}_k + b$ as

$$y_k = \sum_{n}^{m} y_n\alpha_n x_n^T x_k + b \tag{13}$$

We can get dual representation of Lagrange by plugging value of **b**,C and ξ obtained above given by,

$$L_{dual} = \sum_{n=0}^{m} \alpha_n - \frac{1}{2}\sum_{n}^{m}\sum_{k}^{m} \alpha_n y_n x_n^T x_k y_k \alpha_k \tag{14}$$

On the equation 12, we do have information of **w** but α is still without adequate information. Looking at equation 12 and 14, the equation 12 is computationally costly when dimension is gets higher since we have to compute many dot product. For the dual Lagrange, computational cost is cheap because lot of terms are zero. We just need to compute dot product between **x** and support vector.So, it is good idea to use dual Lagrange. The miss-classified data and inside the margin have $_n = C$. The data on the margin have $\alpha_n < C$. The data which are well shattered by classifier have $\alpha_n = 0$ as evidence of above analysis.

III. Experiments and Results

A. Task 1 and 2

We performed experiment to investigate the above theory. In the task 1 and 2, we construct a machine using the parameter of training that classifies the training data. The classifier used in the process is linear. We used $svmtrain$ to construct model by passing a label vector **y** and training matrix X and condition to classify data into two class. We used $C - SVC$ while constructing our model to add trade or cost parameter C can be determined.

After constructing model, we computed **w** and b. we then used **w** and b to construct classifier y using the given equation

$$y = \mathbf{x}.\mathbf{w} + b \tag{15}$$

Here the **w** have two value w_1 and w_2.w_2 is normalizing factor. The equation 15 takes the form of

$$y = \frac{xw_1 + b}{w_2} \tag{16}$$

The above equation gives linear classifier (line). We can then compute the other two lines as follows:-

$$\mathbf{x.w} + b = 1, \text{for upper hyperplane} \tag{17}$$

$$\mathbf{x.w} + b = -1, \text{for lower hyperplane} \tag{18}$$

Finally, we plot the points and hyperplane (line) and compute the slope and margin length for different standard deviation (σ) and cost parameter (C). we also computed the error between Y and prediction.

B. Task 3

In this task, we compute average training error, test error and difference of them on varying parameter like cost (C), number of data.

First of all, we vary C from $10^{-1.5}$ to 10 for the total number of observation (N)=100. We computed X and Y and generate model from the data. After generating model, we generate other sets X and Y test data and predict the test data. We compute test error with updated set Y and prediction. We average down the error for varying C iteration. And then compute difference between test and training data for varying C iteration and average the error.

For the second test, we vary N from 10 to 500 with fixed C=1. we generate a model from the data and then predict the X and Y. we compute training error and generate another set of data X and Y to predict current set. Then we utilize new X and Y to compute test error using Y for the 100 iteration and average down the error. Finally, compute difference between two error and plot respectively.

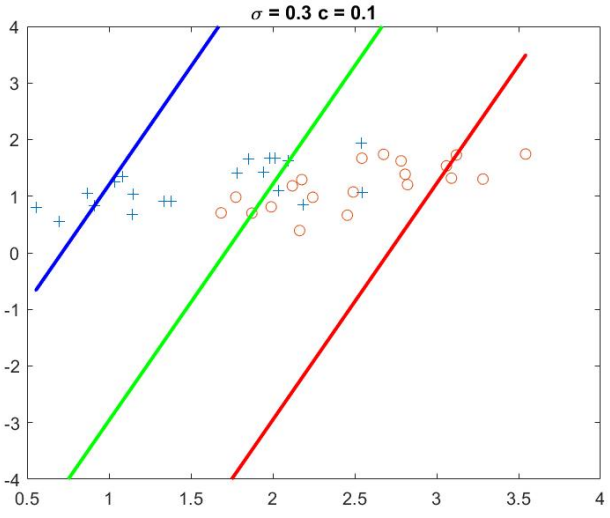

Fig. 1: Classifier with $\sigma = 0.3$ and C=0.1

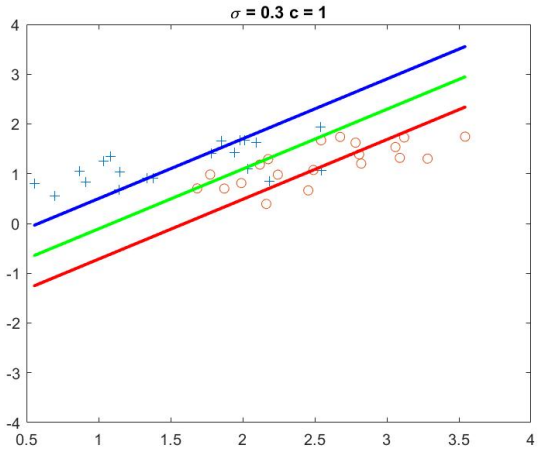

Fig. 2: Classifier with $\sigma = 0.3$ and C=1

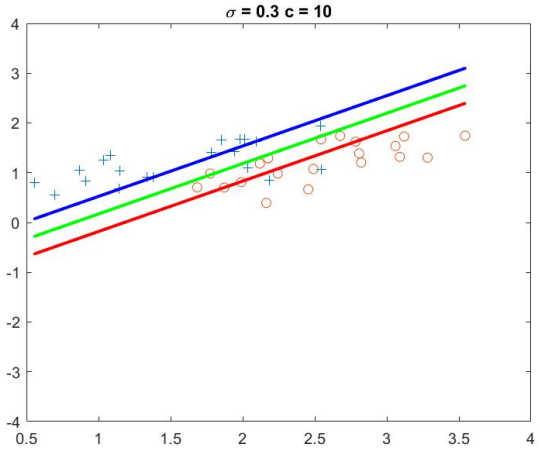

Fig. 3: Classifier with $\sigma = 0.3$ and C=10

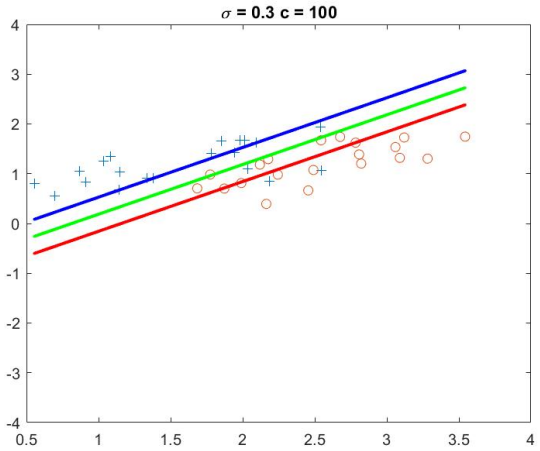

Fig. 4: Classifier with $\sigma = 0.3$ and C=100

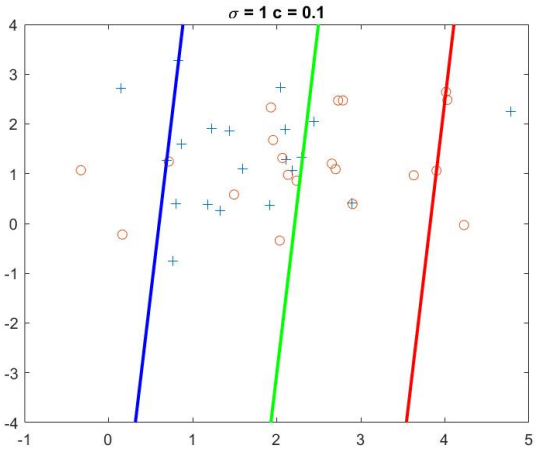

Fig. 5: Classifier with $\sigma = 1$ and C=0.1

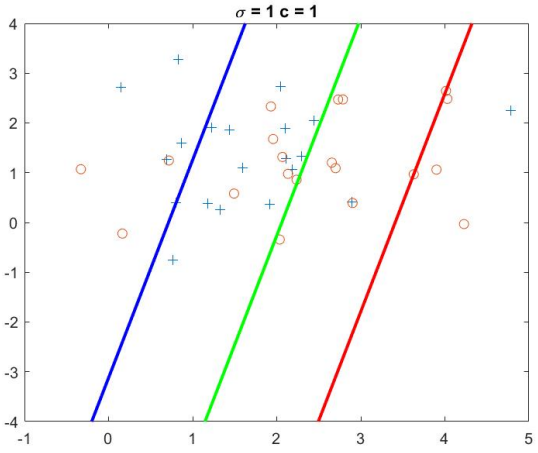

Fig. 6: Classifier with $\sigma = 1$ and C=1

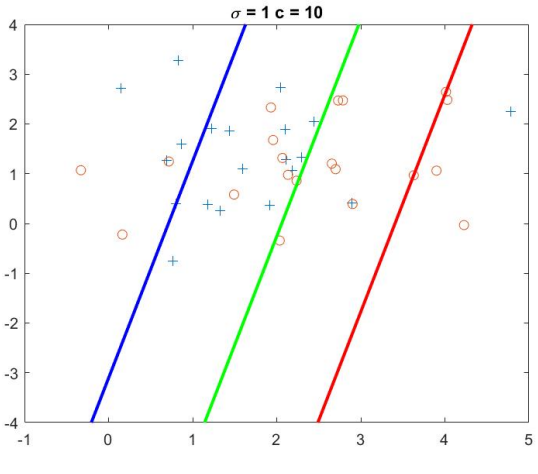

Fig. 7: Classifier with $\sigma = 1$ and C=10

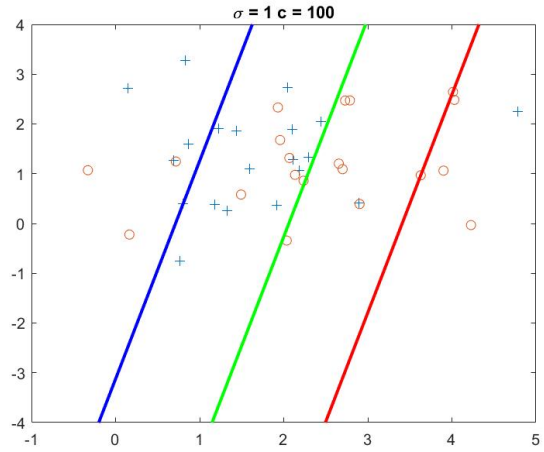

Fig. 8: Classifier with $\sigma = 1$ and C=100

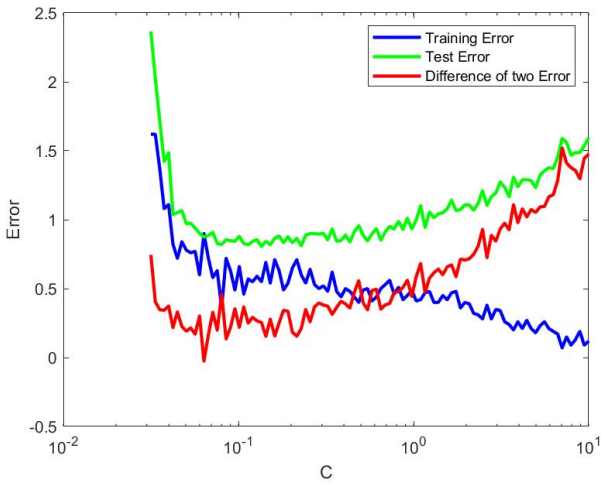

Fig. 9: Variation of Error with Cost and N=100

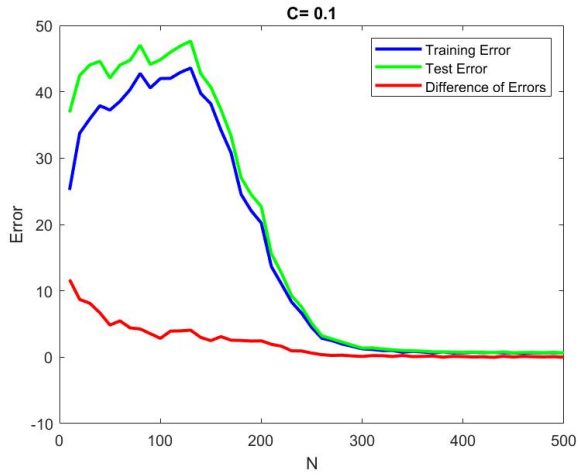

Fig. 10: Variation of Error with N and fixed C=0.1

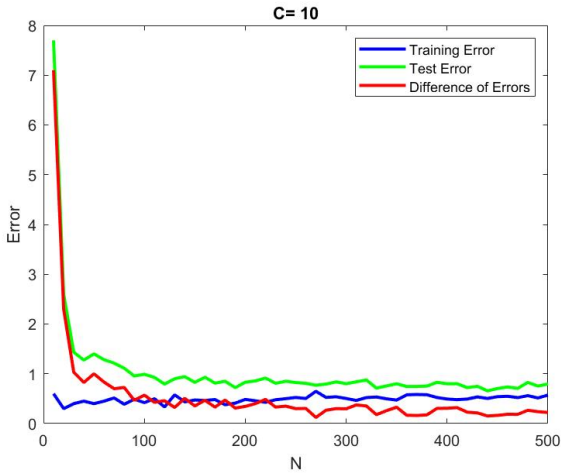

Fig. 11: Variation of Error with N and fixed C=10

IV. Discussion

A. Task 1 and 2

Figure 1 to 8 represents task 1 and 2 with varying parameter C and standard deviation σ. we can see that on figure 1,2,3 and 4 with fixed $\sigma = 0.3$ and increasing cost parameter (C). The classifier with high C have narrower margin and with low c has wider margin. The structural risk is dominant in case of low C whereas empirical risk is dominant in case of high C. So, Optimal classifier is on some medium value of C. Furthermore, we can see on 5,6,7 and 8 with fixed $\sigma = 1$ and varying C. We relative choose higher σ where relative increase in C doesnot narrow the margin like of the previous figure but also the margin gets narrower when increasing the parameter C. We see clearly the data are less spread for lower sigma in figure 1,2,3 and 4. However, Data are more spread for higher σ in figure 5,6,7 and 8. The support vector increase in number on increasing the σ which is evident on comparing on figure 1 to 4 with low σ vs figure 5 to 8 resulting on wrong classification of data. So, Error increases with increasing σ.

B. Task 3

Figure 9 show behaviour of training error and test error with cost parameter C. For the low value of C, the structural error is dominant than empirical error. On increasing C to 1 or 2, we can see that both empirical and structural error has equal contribution to error where the optimal value of parameter exists. On further increasing C, we see that the empirical error is dominant.

The complexity is low when the value of c is lower which leads to under fitting. On increasing C, the complexity is higher resulting test data poorly classified and over-fitting. We know risk can only be upper bounded so there is no physical or numerical way to calculate structural risk.

The structural risk is bounded by total risk and empirical risk by equation 3. Theoretical model predicts the actual risk is double of empirical risk where both meets but it doesnot intersect in practice. On increasing N on figure 10 (C=0.1), both test and training error goes to zero quickly. On increasing N on figure 11 (C=10), the test error decreases quickly and training increases because the dimension of the classifier is held fixed at 10. From the VC theorem, we can only shattered 11 points. On increasing the number of points, the classifier is not good enough to shattered point and training error increases. With increasing number of points, there is support vector with large margin which might be the reason for decreasing the test error.

V. Conclusion

We have presented theoretical framework of support vector machine developed by [1], [2], [3], [4] and [**?**] in the paper. We showed nature of empirical and structural error on varying the parameter like cost, standard deviation and number of observation. The optimal C is hard to find due to over-fitting when increasing C and standard deviation and under-fitting when decreasing C. we found out the error goes to zero when increasing number of observation for low C but doesnot converge to zero for high C.

References

[1] Vapnik, Vladimir N., Alexey Ya Chervonenkis, and N. Moskva. "Pattern recognition theory." Statistical Learning Problem (1974).

[2] Cortes, Corinna, and Vladimir Vapnik."Support-vector networks." Machine learning 20.3 (1995): 273-297

[3] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992.

[4] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167

[5] Schlkopf, Bernhard, et al. "Support vector method for novelty detection." Advances in neural information processing systems. 2000.