Homework 1

Luis A. Valbuena Reyes

***Abstract*— In this document, we present the solution for homework 1. We are required to use to function *svmtrain* and *svmpredict* from the package LIBSVM. Trying to follow the guidelines for writing papers, we develop each the requirements of the assignment on Sections II, III, and IV.**

I. INTRODUCTION

This assignment was about how the support vector machine technique behave under different scenarios.

II. THEORY

A. Construction of a classifier with the model parameters

Taking care of conducting matrix operations with compatible dimensions[1], we construct the simplest machine:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w},$$

where it is suggested to calculate $\mathbf{w}$ from the parameters α_i and the data, $\mathbf{X}$. That is:

$$\mathbf{w} = \mathbf{X}^T\alpha \quad \text{with } \alpha = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y},$$

where y are the labels associated with the samples in X. The matrix $(XX^T)^{-1}$ is problematic because it is ill conditioned. Then, using the reduced single value decomposition $X = \mathbf{U\Sigma V}^T$:

$$\begin{aligned}\alpha &= (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y},\\ &= (\mathbf{U\Sigma V}^T(\mathbf{U\Sigma V}^T)^T)^{-1}\mathbf{y},\\ &= (\mathbf{U\Sigma V}^T\mathbf{V\Sigma}^T\mathbf{U}^T)^{-1}\mathbf{y},\\ &= (\mathbf{U\Sigma}^2\mathbf{U}^T)^{-1}\mathbf{y},\\ &= \mathbf{U\Sigma}^{-2}\mathbf{U}^T\mathbf{y}\end{aligned}$$

where matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary. The machine is then:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} = \mathbf{X}\mathbf{X}^T\mathbf{U\Sigma}^{-2}\mathbf{U}^T\mathbf{y}.$$

Then, for a single sample x_i, we have $\hat{\mathbf{y}}_i = x_i\mathbf{w}$. The results are presented on Section III-A.

On the other hand, we have a tool called support vector machines, which finds a balance between complexity and regularization. On the minimization of the mean-squared error, we are increasing the complexity of the solution, which can lead to an over-fitting of the learner: the machine will be learning the noise of the data along. Then given a linear classifier of the form $f(\mathbf{X}) = \mathbf{w}^T\mathbf{X} + b$ we minimize the error measure and also the magnitude of $\mathbf{w}$, which in turn minimizes the complexity. The algorithm is minimize

$$\|\mathbf{w}\| + C\sum_i \xi_i, \text{ subject to } y_i(\mathbf{w}^T x_i + b) \geq 1 - \xi_i.$$

The region between the planes $y_i(\mathbf{w}^T x_i + b) \geq 1$ is called the margin and the only data that takes place in the support vector algorithm are the samples that are inside the margin and the samples that are completely misclassified with respect to the classifier, see Fig. 1

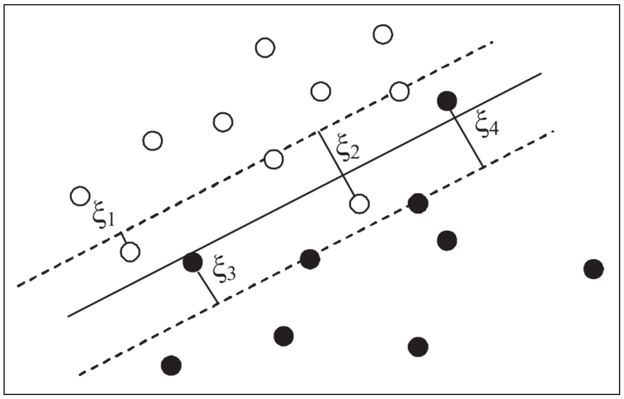


Fig. 1. Support Vector machine schematics. Taken from the class notes.

B. Graphical representation of an SVM

The function *svmtrain* has the option of displaying the data with the separating line and the support vectors. To draw the two margin lines, we created a meshgrid, evaluated the support vector machine given by @svmtrain@ with the margin offset ±1 and plot the contour where $f(X) = 0$. The results are presented on Section III-B.

C. Estimating Structural Risk

In [1], for a group of l predictors, the empirical risk (error) is given as:

$$R_{emp}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l} \mid y_i - f(\mathbf{x}_i, \alpha) \mid, \tag{1}$$

where $\mathbf{x}_i$ and y_i are the data from the training set. The instructor advised to use the same expression using $\mathbf{x}_i$ and y_i from the test set to calculate the actual risk. The structural risk is the difference between the actual risk and empirical risk. The results are presented on Section III-C.

Luis Valbuena is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, 87131-0001, {lavalbuenar@unm.edu

[1] The dimensions of the quantities in the code are not compatible with the matrix multiplications presented on some of the slides

D. SVM for Regression

This experiment requires us to simulate the linear process

$$\begin{aligned} y[n] &= 0.03y[n-1] - 0.01y[n-2] \\ &\quad + 3x[n] - 0.5x[n-1] + 0.2x[n-2], \end{aligned}$$

where $x[n]$, $x[n-1]$ and $x[n-2]$ are the inputs. Preliminarily, it was assumed that $x[n]$, $x[n-1]$ and $x[n-2]$ were the attributes and $y[n]$ the label, but note that $y[n-1]$ and $y[n-2]$ also take part on $y[n]$. Therefore, it is required to precompute the kernel used by the support vector machine algorithm to specify what are the labels and the attributes. The input is $N(0,1)$ and the signal $y[n]$ is disturbed by additive noise of the form $e[n]$ $N(0,0.01)$. The idea of this theoretical example is that we only have $x[n]$ and $y[n]+e[n]$ and we reconstruct $y[n]$. Let $o[n] = y[n]+e[n]$. Based on the advice given by the instructor, the new attributes and labels are of the form:

$$Z_i = \begin{bmatrix} x[i-2] \\ x[i-1] \\ x[i] \\ o[i-2] \\ o[i-1] \end{bmatrix}, \quad Y_i = o[i],$$

for $i \geq 2$. Then $\mathbf{Z} = [Z_2 \quad Z_3 \quad \ldots \quad Z_N]$ and $\mathbf{Y} = [o_2 \quad o_3 \quad \ldots \quad o_N]^T$. The precomputed kernel has the form $\mathbf{K} = \gamma\mathbf{I} + \mathbf{Z}^T\mathbf{Z}$.

The results are presented on Section III-D. The RMS error is calculated between the output $y[n]$ and the estimated $\hat{o}[n]$ signal given by the calculated support vector machine.

III. Experiments

This is the output of the command @svmtrain@ given on the sample code, see Fig. 2(a)

A. Construction of a classifier with the model parameters

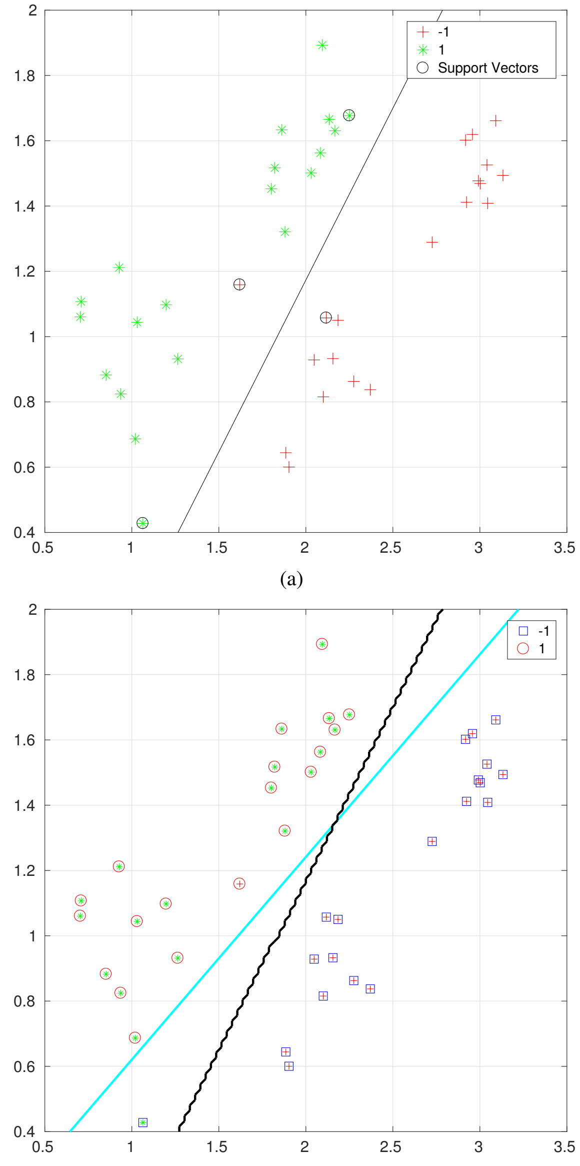
The comparizon requested can be appretiated on Fig. 2(a) and Fig. 2(b).

This time, the samples that the constructed machine classifies as 1 are assigned a red circle marker, while the others are assigned a blue square marker. The stepped black[2] line on Fig. 2(b) is the black line on Fig. 2(a), and the cyan line is a contour line of the plane perpendicular to $\mathbf{w}$ as $x_i\mathbf{w} = 0$. Note that these planes are not parallel. Also note that for this particular experiment, the sample in the region $1 \leq x \leq 1.5$ and $0.4 \leq y \leq 0.6$ is misclassified by the constructed machine.

B. Graphical representation of an SVM

We chose $\sigma = \{0.05, 1, 1.4\}$, the plots of the separating line, the two margin lines and the support vectors are presented in Fig. 3(a), Fig. 3(b), and Fig. 3(c).

For $\sigma = 0.05$ on Fig. 3(a), the samples of each classification group are close together creating clusters, hence there are no red markers on the upper left area or green markers on the lower right region, there are not even markers inside the margin area. The transversal length of the margin area is $l_m = 0.29$ units.



(a)

(b)

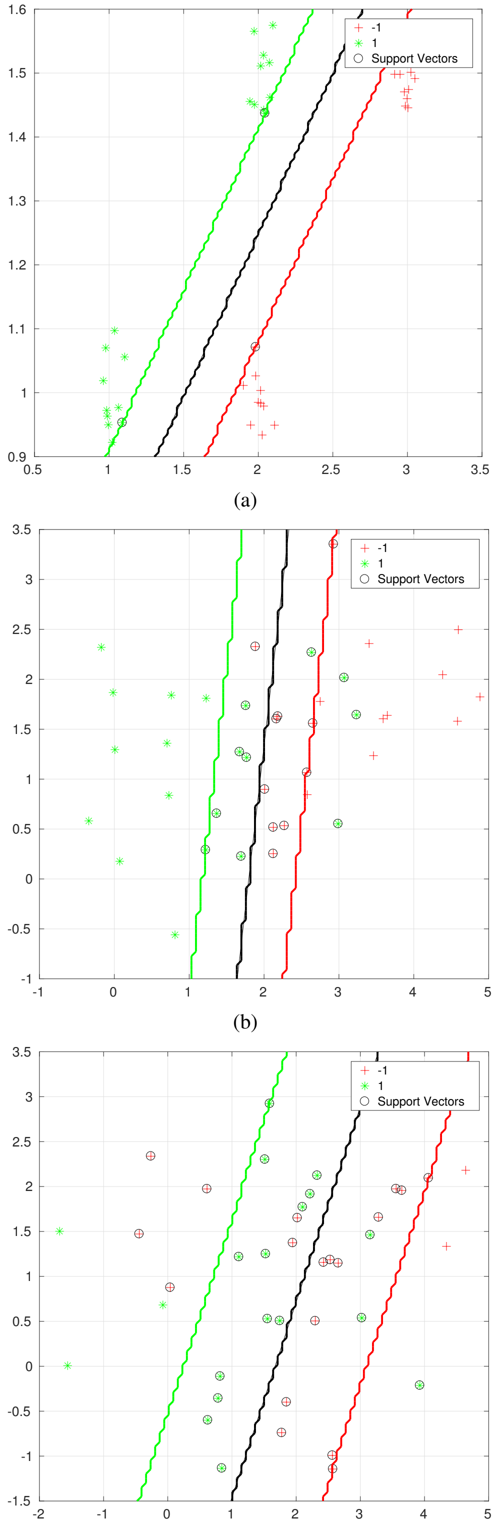
Fig. 2. Output of *svmtrain*. (a), Normal Execution. (b) The constructed machine.

On Fig. 3(b) where $\sigma = 1$, we can see that the two group of data is spread wider and they share a common area, which the support vector machine algorithm marks as the margin region. This time, there are some green markers on the lower right region and some red markers on the upper left region. This time the transversal length of the margin area is $l_m = 1.16$ units.

Finally, for $\sigma = 1.4$ the data is mixed together and the support vector machine algorithm[3] requires more support vectors, see Fig. 3(c). The transversal length of the margin

[2] This line is a contour line for which the classifier is zero. The step feature of the line is due to the mesh grid used.

[3] The value $\sigma = 1.4$ was critical, for slightly bigger values, the algorithm could not converge



(a)

(b)

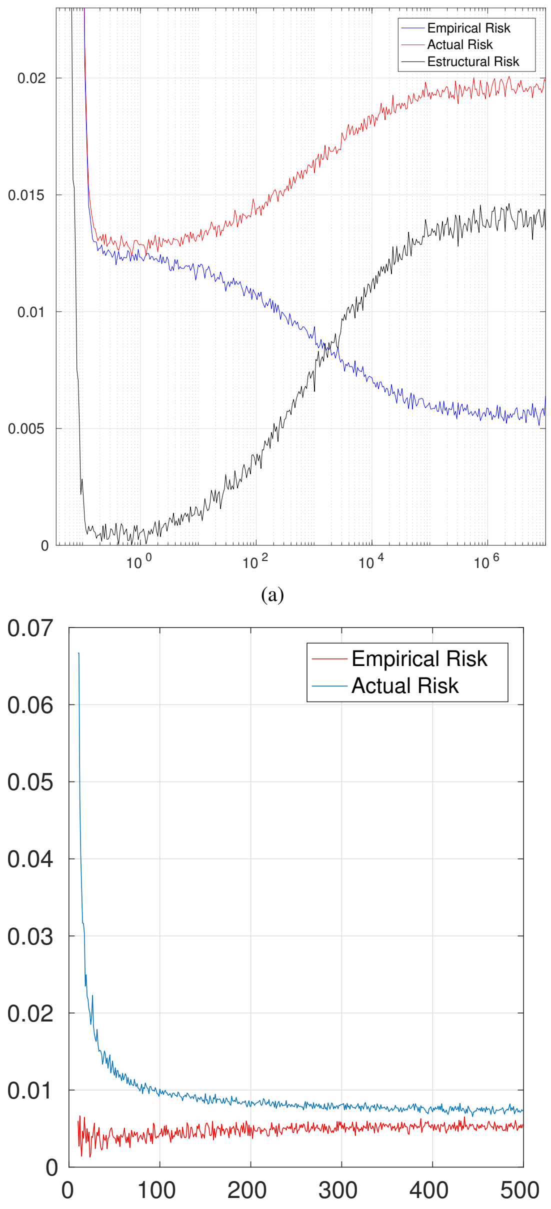
(c)

Fig. 3. Output of *svmtrain* with different values of σ. (a), $\sigma = 0.05$. (b), $\sigma = 1$. (c), $\sigma = 1.4$.

area is $l_m = 2.64$ units.

C. Estimating Structural Risk

The two requested plots are presented on Fig. 4(a), and Fig. 4(b). The purpose of these two graphs was to corroborate that as the empirical risk reduces as the value of C tends to infinity, the actual and structural risks increase, while the actual risk bounds both the empirical and estructural risk.

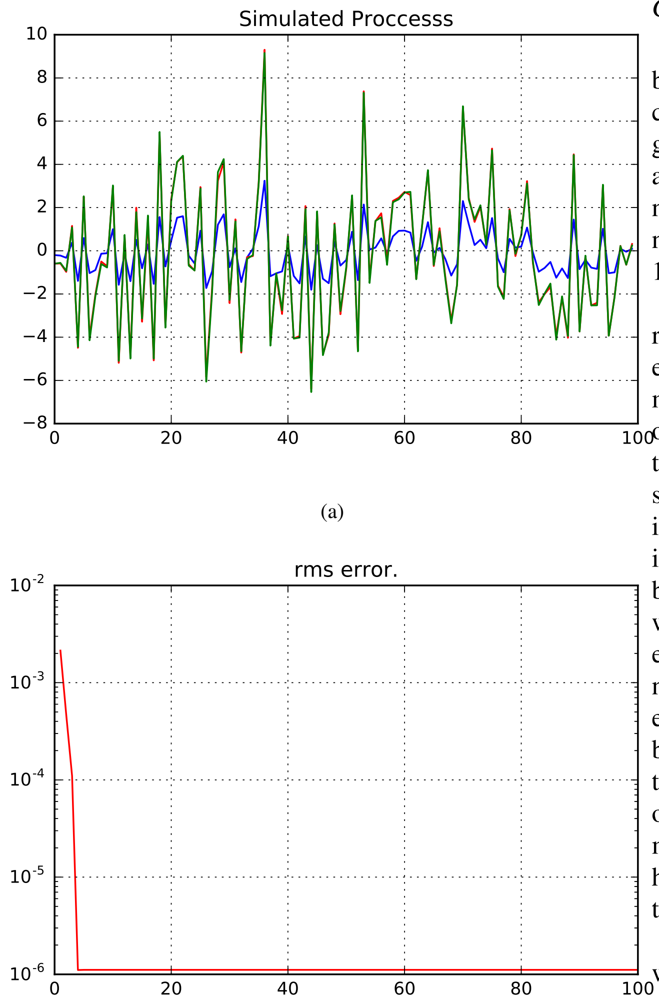


(a)

(b)

Fig. 4. Evolution of empirical and actual risk with respect to (a), value C. (b), the number of predictors N.

The file `data.m` was used to generate both the training and test data. A key element to obtain the results described earlier on is to compute several executions and then average them, that is, we conducted 1000 realizations; additionally, the choice of the interval for the free parameter C is instrumental, the interval of C is $[10^{-2}, 10^{7}]$.



(a)

(b)

Fig. 5. SVM for regression (a), Simulated process: the blue represents the input $x[n]$, the red line is the evolution of the process $y[n]$, and the green line is $o[n]$. (b), RMS Error.

D. SVM for Regression

IV. Conclusion

A. Construction of a classifier with the model parameters

The key differences are: the constructed machine is not conducting a regularization

B. Graphical representation of an SVM

the support vector machine algorithm performs well for Fig. 3(a). it does not have problems at classifying because the data is already clustered according to its label.

For Fig. 3(b), the support vector machine include more samples as support vectors and on Fig. 3(c) even more.

As the variance is bigger, the support vector machine algorithm is pushed harder.

C. Estimating Structural Risk

We want to know the actual risk of a trained machine because we want to know how susceptible our trained machine is under a new subset of data. However, the definition given on [1] requires knowledge of the probability function associated with the data, which is not available. Hence, we need to estimate this quantity. We do that by generating a new data set and use it as our test set, and then use Equation 1 with the test set.

We think the reason behind of why the minimum actual risk does not coincide with the intersection of the other two errors has to do with complexity. Note that on Figure. 4(a) the minimum of the actual risk is almost the same as the value of the empirical risk for low values of C. This is because the trained machine has a low value C so the machine has the same clearance in both the train and test set. However, as C increases, the complexity does too, so the trained machined is forced to map better the training data to the training labels, but at the same time the machine is not going to perform as well with a new set of data as the test set. Then the actual error is going to increase. The minimum of the actual risk is not going to coincide with the intersection of the other two errors because the manipulation of the value C is a trade off between the complexity and regularization: The minimum of the actual error is one side of the coin; the machine does not overfit but it is too simple. The intersection happens in the middle of the logarithmic domain, where C is bigger and you have made your machine more complex so it can describe the data closer.

For the second question, the procedure seems consistent, we have more data, but after $N \approx 70$ you don't gain more.

References

[1] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1009715923555