ECE 273 Final Project
Using Supporting Vector Machine For Classification

Xufan Xiong[a], **Tianduo Hao**[b]
[a]University of California, San Diego, A53256057, x7xiong@eng.ucsd.edu
[b]University of California, San Diego, A53250540, t6hao@eng.ucsd.edu

Keywords: SVM, hard margin, soft margin, kernel trick.

1 Objective

1.1 Understanding the SVM algorithm

In this project, what we need to do first is understanding supporting vector machine which is a kind of useful supervised machine learning algorithm for classification and regression.

1.2 Thinking about how SVM works in different situations

After understanding the basic SVM method, we have to thinking about how to use the SVM in several different situations, such as linear situation with hard-margin SVM, soft-margin SVM, dual form SVM, and non-linear classification with kernel trick SVM with hard-margin and soft-margin.

1.3 Applying SVM method in real data

This time, we tried to use Matlab codes to implement the SVM algorithm in different situations. For a better visualization, we use different kind of data sets, including two clusters of linear separable data for hard-margin SVM, linear separable data with some noise points for soft-margin SVM, and a nonlinear separable data set for kernel trick SVM. After testing our models on the basic data set we downloaded from Coursera, we apply them on a real data set from Kaggle which contains voices with some features and need to be classified into different generations to find out a suitable model that could best classify the data set.

2 Background and Motivation

Supporting vector machine is a kind of model for binary classification. The basic SVM model is defined to classify for a largest margin in feature space. The purpose for SVM is to maximize the margin so that it can be transferred as a quadratic convex problem. For example, we have a given data set. If every data is labeled as one class or another, then use SVM algorithm could build a model to separate the data so that if a new data get in to the set, it could give it to one of those two categories. For a real data set like classify cat or dog images, if we could maximize the margin in SVM, the model will work better so that less error will be generated such as define a cat photo as a dog. In this project, what we are going to do first is to build some models to separate some generalized points from a online course in Coursea: linear separable, linear separable with some noise points and nonlinear separable. After training our models, we are going to use these to dealing with a real data set. The data set we used is a reviewed data set from Kaggle called "Gender Recognition by Voice" which contains about 3000 labeled data and each has 20 different

voice features, such as frequncies or modes. The goal for us is to build a model that could defined each voice is belonged to male or female correctly.

3 Method

In this section, we are going to talk about the basic concepts for SVM like linear classification and support vectors and algorithms for different kind of SVMs.

3.1 Margins and supporting vectors

To understand the basic supporting vector machine, we have to have an idea of linear classification. Thinking about a binary classification problem represented by some data points, we have to find a hyperplane $W^T x + b = 0$ to separate the two classes. If the data set is linear separable, there will be infinite hyperplanes that could located between the 2 classes. However, if we want to get the best solution, the largest gap between the 2 classes should be achieved which means that we have to find two parallel hyperplanes to separate the data with the largest gap between them. Those hyperplanes that define the gap are the margins. Those data points that located on the margins are the supporting vectors. These points play a dominant role in solving classification problem.

3.2 Hard-margin SVM

In linear SVM, the simplest situation is that all the points could be separated into two classes without any errors. In this time, all the data points can be showed as $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, and y_i is equal to 1 or -1 which means which class does the point belongs to and find two parallel hyperplanes that maximize the margin. So that the points which are far away from the margin are useless and can be removed without influnce the SVM model. The two hyperplanes can be expressed as

$$w^T x + b = 1 \text{ or } \; w^T x + b = -1 \tag{1}$$

The distance between the two hyperplanes can be expressed as $\frac{2}{||w||}$, we have to maximize the distance to get the best separation result, so $||w||$ should be minimized. At the meantime, to make sure that all the points are outside the two hyperplanes, we need to ensure that

$$\begin{aligned} &w^T x_i + b \geq 1 \text{ for all } \; y_i = 1 \text{ or} \\ &w^T x_i + b \leq -1 \text{ for all } \; y_i = -1 \end{aligned} \tag{2}$$

These two formulas can be shown as

$$y_i(w^T x_i + b) \geq 1 \text{ for all } \; i = 1, ..., n \tag{3}$$

And this time, we could summarize the general formula for the hard-margin SVM [1] as

$$\begin{aligned} &\min_{w,b} \; ||w||_2 \\ &\text{s.t. } y_i(w^T x_i + b) \geq 1, i = 1, ..., n \end{aligned} \tag{4}$$

When solving the problem in a higher dimension, it could be harder for solving quadratic programming problem as the complexity of the problem increases. This time, we need to introduce

Lagrange Function to eliminate the effect of dimension rising. Lagrange function allow us to combine the objective function and constraint function together as

$$L(w, b, \lambda) = 1/2||w||^2 + \sum_{i=1,...n} \lambda_i^T (1 - y_i(w^T x + b)) \tag{5}$$

In this new function, the λ is Lagrange multiplier and we need to assert that $\lambda \geq 0$. To show that this new peoblem is the same as the original one, for any λ' we can say that

$$\min_{w,b}(\max L(w, b, \lambda)) \geq \max(\min_{w,b} L(w, b, \lambda')) \tag{6}$$

In the hard-margin SVM, the objective function is convex and feasible primal, the constraint function is linear, so that it can be defined as strong duality which means the dual solution and the primal solution are the same. In this case, we can remove the sign of inequality. To solve the new problem, we can first do the partial derivative for the variable w and b, respectively and let each one equals 0. For the derivation of b, we can get

$$\sum_{i=1,...,n} \lambda_i y_i = 0 \tag{7}$$

And from the derivative of w, we can get

$$w = \sum_{i=1,...,n} \lambda_i y_i x_i = 0 \tag{8}$$

After transfer the function from maximize to minimize, we finally get the simplest form of the dual form for Lagrange problem [2]

$$\begin{aligned} &\min_{\lambda} \sum_{i=1,...,n} \sum_{j=1,...,n} 1/2 \lambda_i \lambda_j y_i y_j x_i^T x_j - \sum_{i=1,...,n} \lambda_i \\ &\text{s.t. } \lambda \geq 0, \sum_{i=1,...n} \lambda_i y_i = 0 \end{aligned} \tag{9}$$

This time, the original problem has already transformed to a QP problem. However, the hard-margin SVM is limited because it need all the data points can be strictly linearly separated into 2 classes. If there exist some points that located in the other side of the hyperplane, the model will failed. This time we need to consider the situation that the points that linear non-separable.

3.3 Soft-margin SVM

As the hard-margin SVM could be over-fitting for even a few noise points, soft-margin SVM is a good choice for the situation. In this case, we also define two hyperplanes that linearly separate two clusters of the data points. However, those points which are closest to the hyperplanes will not be the supporting vectors in this case anymore, because we allow some points passing through the hyperplanes or even get into the other side of the model because in many cases, there are always noise data points which make the whole data set cannot be strictly linear separated. To show the error points mathematically, we can introduce the slack variable ξ to the objective and constraint

function, so that the problem in hard margin SVM could transformed to this one [3]

$$\begin{aligned} &\min_{w,b} ||w||_2 \\ &\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, ..., n \\ &\xi_i \geq 0; \sum_{i=1,..,n} \xi_i \leq Z \end{aligned} \tag{10}$$

In this form, Z is called tuning parameter that shows the extent of error which controls the total number of points that are not well separated like passing through hyperplanes. Larger Z means that we allow less error points that pass through the margin hyperplane and smaller Z means we allow a lot of error points that not classified correctly. If the point is classified to the right category, the constraint function still be larger than 1 and a larger value of $y_i(w^T x_i + b)$ means the point is far away from the hyperplane. In contrast, if the $y_i(w^T x_i + b)$ is smaller than 1 and the more smaller, the more it get far away from the hyperplane in the wrong direction. In this case, the slack variable ξ works. We can also solve the soft-margin SVM in duality form using two Lagrange multipliers λ, ν

$$\begin{aligned} L(w, b, \xi, \lambda, \nu) &= 1/2||w||^2 + Z \sum_{i=1,...n} \xi_i \\ &+ \sum_{i=1,...n} \lambda_i^T (1 - \xi_i - y_i(w^T x + b)) \\ &+ \sum_{i=1,...n} -\nu_i \xi_i \end{aligned} \tag{11}$$

And then we could transfer the original problem to the dual problem as

$$\max_{\lambda,\nu}(\min_{w,b,\xi} L(w, b, \xi, \lambda, \nu)) \tag{12}$$

The idea for solving this problem is similar to the the hard-margin SVM, we need to take the derivative of the 3 variables:w,b,ξ. First, as we take the derivative of the ξ

$$\begin{aligned} &\frac{dL}{d\xi} = Z - \lambda_i - \nu_i = 0 \\ &\implies \nu_i = Z - \lambda_i \end{aligned} \tag{13}$$

We can also get the same result as hard-margin SVM when taking the derivative of w and after eliminating the ξ_i in the formula, we can get the following one

$$\max_{0 \leq \lambda_i \leq Z}(\min_{w,b}(1/2||w||^2 + \sum_{i=1,...n} \lambda_i^T (1 - y_i(w^T x + b))) \tag{14}$$

From that equation which is similar to the hard-margin SVM and only add a constraint function, we can get the simple form for the dual problem of soft-margin SVM [4]

$$\begin{aligned} &\min_{\lambda} \sum_{i=1,\dots,n} \sum_{j=1,\dots,n} 1/2\lambda_i\lambda_j y_i y_j x_i^T x_j - \sum_{i=1,\dots,n} \lambda_i \\ &\text{s.t. } 0 \leq \lambda_i \leq Z \\ &\sum_{i=1,\dots n} \lambda_i y_i = 0 \end{aligned} \tag{15}$$

This time, we could solve the dual problem without worry about the new variable of the primal problem. And soft-margin SVM could dealing with the most linear non-separable data. However, there will be some situations like the total data set cannot be linearly separated by the hyperplane, such as one cluster of data points contains in the other one, what we need to do is to use kernel trick to deal with it.

3.4 Non-linear classification via SVM

3.4.1 Kernel SVM

In the previous parts, we have derived the dual form of the hard-margin and soft-margin SVM. In this section, we will talk about how to use kernel trick to solve non-linear separable problems via these two kinds of SVMs. The dual problem for the hard-margin SVM is

$$\begin{aligned} &\min_{\lambda} \sum_{i=1,\dots,n} \sum_{j=1,\dots,n} 1/2\lambda_i\lambda_j y_i y_j x_i^T x_j - \sum_{i=1,\dots,n} \lambda_i \\ &\text{s.t. } \lambda \geq 0, \sum_{i=1,\dots n} \lambda_i y_i = 0 \end{aligned} \tag{16}$$

For the objective function, we can find that the complexity of the function focus on the $y_i y_j x_i x_j$ which is called Q matrix and to dealing with the Q matrix, we have to do the inner product with the data mapping in a higher dimensional feature space to make the linear separating though the original problem cannot do which is hard to solve. In this case, we can use kernels for a easier way to solve the problem. First, we need to transfer the Q matrix as [5]

$$q_{i,j} = y_i y_j x_i^T x_j = y_i y_j \phi(x_i)^T \phi(x_j) \tag{17}$$

In this step, the ϕ is called feature map that could map the data from the original n dimensional space to a higher dimensional Euclidean space. To solve the new dot product, we introduce a function k which is called a kernel function for representing the inner product of the feature map as

$$k(x, x') = \langle \phi(x'), \phi(x) \rangle \tag{18}$$

which make the inner product happens on the original dimension to realize the inner product of feature maps that avoid the extra computing on the high dimension complexity. Using a second order kernel as the example, if we choose 2 data which from the original n-dimensional space, the

feature map of the each data is

$$\phi(x) = (1, x_1, x_2, ..., x_n, x_1^2, x_1x_2, ..., x_1x_n, ..., x_nx_1, x_2^2, ..., x_n^2) \tag{19}$$

And the inner product of the feature maps can be shown as

$$\begin{aligned} \phi(x)\phi(x') &= 1 + \sum_{i=1,...,n} x_i x_i' + \sum_{i=1,...,n} \sum_{j=1,...,n} x_i x_j x_i' x_j' \\ &= 1 + xx' + (xx')^2 \end{aligned} \tag{20}$$

And when we use the kernel trick, we can define that

$$k_\phi(x, x') = \langle \phi(x'), \phi(x) \rangle \tag{21}$$

Now we can replace some of the complex part in dual formula by the kernel function and solve it. Such as the Q matrix can be expressed as

$$q = y_i y_j k((x_i)^T(x_j)) \tag{22}$$

And we can use it to solve the dual problem. [6]

3.4.2 Some kernels

There are some useful kernels to solving nonlinear data. The first one is called polynomial kernel which is defined by

$$\begin{aligned} &k(x, x') = (\langle (x'), (x) \rangle + c)^m \\ &c \geq 0, m \geq 0 \end{aligned} \tag{23}$$

If $m = 1$ and $c = 0$, the polynomial kernel will be called as linear kernel which is useful for solving linear classification rpoblem and it is more powerful to solve the primal problem than the dual problem. Another important kernel is called Gaussian kernel, which is also called Radial Basis Function, RBF that

$$k(x, x') = exp(-\gamma||x - x'||^2), \gamma \geq 0 \tag{24}$$

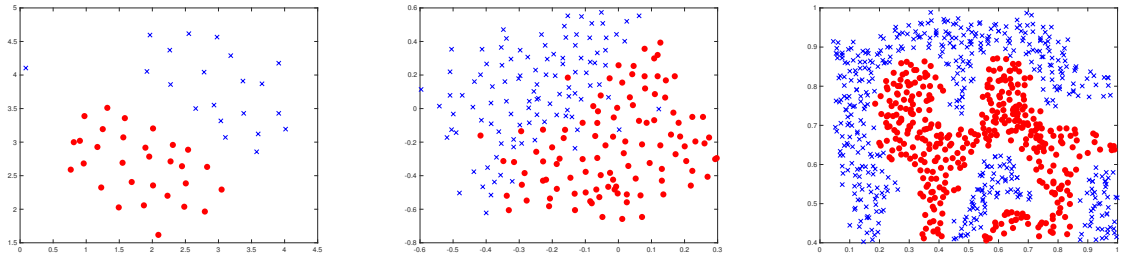
The result of using Gaussian kernel is heavily affected by the choice of γ because the increasing of γ means the stand deviation of the Gaussian function decreasing so that the Gaussian function becomes sharp and makes the margin become unevenness or even creates some separated margins, so it is important to choose a suitable γ to over-fitting.
Comparing those kernels, we can find that each of them has some advantages and shortages. For the linear kernel, it is the simplest one and should be tried first; However, it can be only used for a linear separable data set. For the polynomial kernel, it can be used to solve non-linear separable data; However, it has more parameters than the others which is difficult to decide each one and need to pay attention to the choice of m. And the last one, Gaussian kernel, which is the one we used in this project, is easy to over-fitting, but it is more powerful than the others and only have one parameter that need to control and can better classify the non-linear separable data.

4 Result and discussion

In this part, we will show our results based on some data sets downloaded from Internet. In order to solve the SVM problems, we import CVX packages into MATLAB and get the hyperplanes by solving the convex optimization problems.

4.1 The data sets

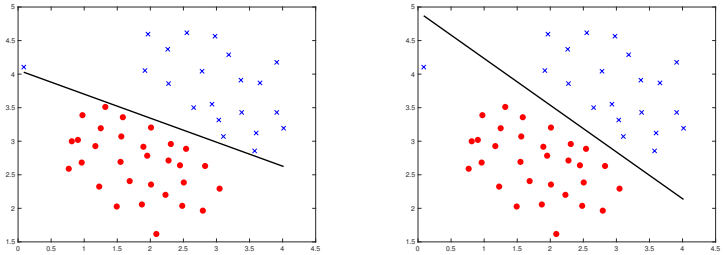
The data sets we introduced in this project are divided into two parts. At first, we used three data sets, which have two features in each, to illustrate and visualize the principal of SVM in 2-dimensional features space (As shown in Fig.1). The data were downloaded from 'Machine learning' course of Andrew Ng. Furthermore, 'Gender Recognition by Voice' data sets from Kaggle [7], which have 20 features, was introduced to train some SVM models based on hard margin, soft margin, dual and kernel trick. Based on the result of these models, we try to analyze performance of these model.



(a) (b) (c)

Fig 1: Visualization of three 2D datasets

4.2 Hard margin and soft margin



(a) (b)

Fig 2: Result of first data set with hard margin and soft margin

As shown in the Fig.1(a) and Fig.1(b), we use these two data sets to illustrate hard margin and soft margin with dual or not. Fig.2(a) shows the data that can be separated by hard margin SVM by solving the convex optimization problem we have mentioned in Equation(4). Since the two classes of sample are separated with a large margin, hard-margin SVM could get an better result

of this problem. However, there is a point that can affect the result of SVM. Hence, to develop our solution more significantly, we use soft-margin SVM and set the tuning parameter Z to 5. As shown in Fig.2(b), after using soft margin SVM and ignoring the abnormal point, we get a much better hyperplane that can separated two classes with a higher accuracy.

As shown in Fig.1(b), these two classes of sample cannot be separated with an obviously wide boundary. Some points are mixed in another class. In this situation, we need use soft margin SVM which can allow some noise points passing through the hyperplanes. By solving the convex optimization problem in Equation(10) with tuning parameter $Z = 50$, as shown in Fig.3, a soft margin hyperplane can divide the second dataset into two classes. The accuracy of this model is nearly 92.5%, but it still have some method to increase the accuracy of the model. We will discuss our kernel trick model in the following parts.

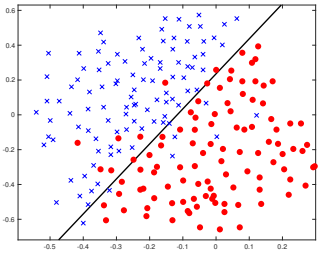
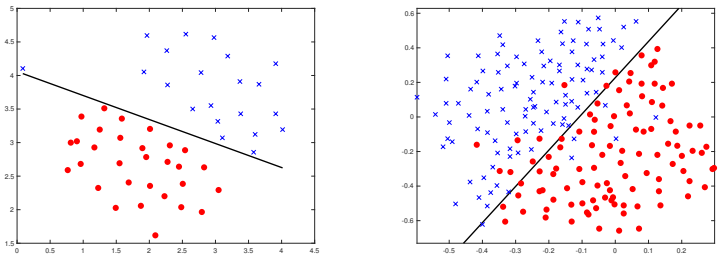


Fig 3: Result of second data set with soft margin

4.3 Dual

As discussed earlier, if the dimension of the features space increase, the difficulties of solving the former convex optimization problems will rise at the meantime. Therefore, we use $Lagrange\ Function$ and introduce $Dual$ method to eliminate the effect of dimension rising. As shown in Fig.4, the results of hard margin and soft margin with dual method are similar to previous results. It shows that primal problem and dual problem is equivalent in these two data sets.



(a) (b)

Fig 4: Result of the first two data set with dual

4.4 Kernel trick

Some data sets, like the third data set shown in Fig.1(c), can't be separated linearly. So, we need kernel method to project existing current feature space to higher dimensional feature space that will make the data set linearly separable.

At first, we use *Polynomial Kernel*, which is $k\left(\overrightarrow{x_i}, \overrightarrow{x}_j\right) = \left(\overrightarrow{x}_i \overrightarrow{x}_j + 1\right)^9$ and the tuning parameter is $Z = 10$. Then, we get hyperplane of soft margin with kernel as shown in Fig.5. The accuracy of this model is 94.3%, which is larger than previous model.

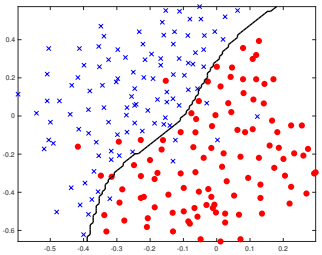
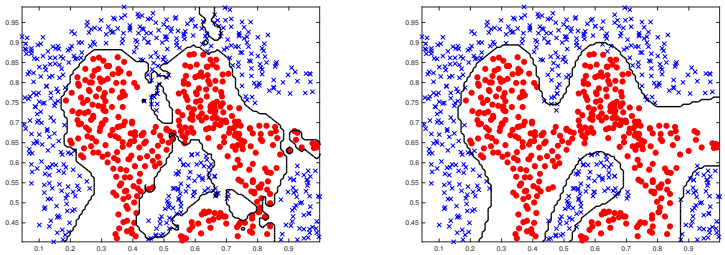


Fig 5: Result of second data set with soft margin and kernel trick

In addition, we try to separate the third data set shown in Fig.1(c). In this situation, Gaussian kernel which was discussed before are introduced to solved this SVM problems. At first, we try to seek a hard margin SVM with Gaussian kernel by solving the Equation(16). The solution of hyperplane as shown in Fig.6(a). However, it's obvious that this model is over-fitting. So, we require a soft margin SVM with Gaussian kernel to solve the over-fitting problem. By solving the Equation(15) with tuning parameter $Z = 10$, the hyperplane of this problem as shown in Fig.6(b). The accuracy of this model is nearly 98.5%. Therefore, the best hyperplane of the third data is obtained by soft margin with Gaussian kernel.



(a) (b)

Fig 6: Result of the third dataset

4.5 Gender Recognition by Voice

To further elaborate SVM, we use a more complex data set – 'Gender Recognition by voice' – to train the models. This data set has 20 features of voice of every male or female. Hence, it need to separate the labels 'male' and 'female' in 20 dimensional feature space. For training the model, we divided the data set into training set and test set. The ratio of these two sets is 7:3.

In order to judge if the data set can be separated linearly or not, we try to use the hard-margin SVM at first. However, there is no feasible solution to this convex optimization problem. Hence, it means the data set cannot be separated linearly with hard margin.

So, in this situation, we try to obtain the hyperplane by solving soft margin SVM. Firstly, we train the model with linear kernel ($k\left(\overrightarrow{x_i}, \overrightarrow{x}_j\right) = \left(\overrightarrow{x}_i \overrightarrow{x}_j\right)$) and set the tuning parameter $Z = 10$,

the accuracy of test set is 97.79%. If $Z = 5,\ Z = 50$, the accuracy will be 97.58%. In addition, Polynomial kernel ($k\left(\overrightarrow{x_i}, \overrightarrow{x}_j\right) = \left(\overrightarrow{x}_i \overrightarrow{x}_j + 1\right)^5$) is introduced to solve the problem. If the tuning parameter is $Z = 5,\ 10,\ 50$, the accuracy of test set is 97.68%. We also try other number of power, such as 3, 4, 6 and 7, the accuracies of test set are almost similar. At last, Gauss kernel is also introduced to train the model. If $Z = 5,\ 10,\ 50$, all of these three accuracies will be 50.74%. Therefore, it's inappropriate to use Gauss kernel for this data set.

To sum up, for this data set, it cannot be separated linearly by hard margin. So, kernel trick with soft margin is introduced to solve this convex optimization problem. From the previous result, Gauss kernel couldn't be used in this problem. Linear kernel and polynomial kernel show a better performance on this problem.

5 Conclusion

In this project, we try to illustrate and understand the principal of support vector machine and the derivative convex optimization problem. Based on MATLAB and CVX package, we implement hard-margin, soft-margin SVMs on solving the linear separable and non-separable problem. Furthermore, to get a non-linear classification model with a higher complexity, kernel trick is introduced to the hard and soft SVMs so that non-linear problems can be solved linearly in a higher dimensional space. We implement our kernel trick SVMs on the real data to verify the feasibility and found that linear kernel and polynomial kernel work better in this case though Gaussian kernel is widely used in most of time. After learning convex optimization, we have a deeper understanding of the principals about supporting vector machine and We will keep learning SVMs and convex problems in the future.

References

[1] K. P. Bennet and C. Campbell, "Support vector machine: Hyper or hallelujah," *SIGKDD Explorations* **2**, 1 (2000).

[2] N. Vasconcelos, "The support vector machine," (2009). http://www.svcl.ucsd.edu/courses/ece271B-F09/handouts/SVMs.pdf.

[3] C. J. BURGES, *A Tutorial on Support Vector Machines for Pattern Recognition*, Kluwer Academic Publishers, Boston., Bell Laboratories, Lucent Technologies (1998).

[4] N. Vasconcelos, "The soft-margin support vector machine," (2009). http://www.svcl.ucsd.edu/courses/ece271B-F09/handouts/SoftSVMs.pdf.

[5] C. Frogner, "Support vector machines," (2011). https://yeolab.weebly.com/uploads/2/5/5/0/25509700/class06-svm.pdf.

[6] D. Fradkin and I. Muchnik, *Mathematics Subject Classification*, Addison-Wesley, Reading, Mass. (2004).

[7] K. Becker, "Gender recognition by voice – identify a voice as male or female," (2016). https://www.kaggle.com/primaryobjects/voicegender/.