Programming Assignment 6: Support Vector Machine

Introduction

In this assignment, you will **implement the Support Vector Machine (SVM)** algorithm. To get started, you will need to download the starter code and unzip its contents to the directory where you wish to complete the assignment. The following files are included,

#	File name	Description	Add your code?
1	PA6test.m	MATLAB script that steps you through the first part of the assignment	No (you may change the C parameter)
2	PA6.m	MATLAB script that steps you through the second part of the assignment	No
3	PA6data1.mat	Example Dataset 1	No
4	PA6data2.mat	Example Dataset 2	No
5	PA6data3.mat	Example Dataset 3	No
6	linearKernel.m	Define the linear kernel	Yes
7	gaussianKernel.m	Define the gaussian kernel	Yes
8	Ddataset3Params.m	Function to select the best model (varying C, sigma)	Yes
9	plotData.m	Plot the dataset	No
10	svmTrain	SVM training algorithm	No
11	svmPredict.m	Return prediction	No
12	visualizeBoundary.m	Plot linear non-linear decision boundary	No
13	visualizeBoundaryLinear.m	Plot linear decision boundary	No
14	PA2data1.m	Dataset from PA2	No

After you complete all the 3 files (#6-8) that you need to add code to, put all 14 files above and 1 text file (.doc, or pdf) for your Task 2 and files for Task 5 (if you choose to work on the bonus task) into a folder with your name (Folder name format: FirstnameLastname-PA6). Zip the folder and submit it through BrightSpace.

!!Assignments that are not submitted in this format will NOT BE GRADED.

1. Session I

In the first part of this assignment, you will be using support vector machines (SVMs) with various example 2D datasets. Experimenting with these datasets will help you gain an intuition of how SVMs work and how to use a linear kernel or a Gaussian kernel with

SVMs. The provided script, PA6test.m, will help you step through the first half of the exercise.

1.1. Example dataset 1

PA6data1.mat contains a 2D example dataset that is linearly separable. The script PA6test.m will plot the training data (Figure 1). The positions of the positive examples (indicated with green o) and the negative examples (indicated with red x) suggest a natural separation indicated by the gap. However, notice that there is an outlier positive example on the far left at about (0.1*;* 4.1). As part of this exercise, you will experiment to see how this outlier affects the SVM decision boundary.

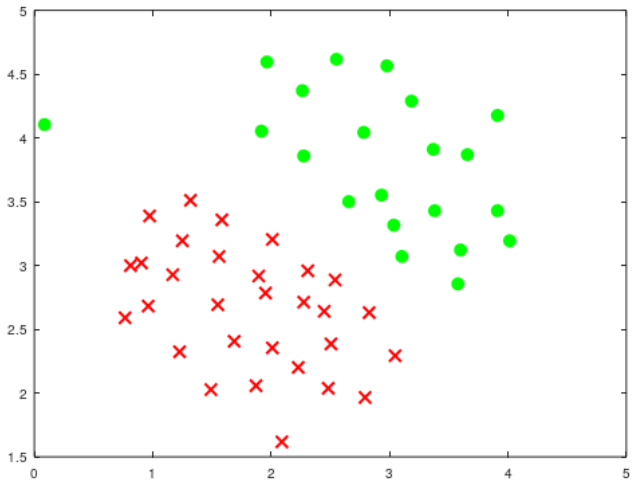

Figure 1 Dataset 1

Task 1. Complete linearKernel.m

The Linear kernel is the simplest kernel function. It is given by the inner product <x,y>. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts.

$$k(x, y) = x^T y$$

Task 2. Examine the role of the *C* parameter

After loading and visualizing the dataset from PA6data1.mat, part 2 in PA6test.m will run the SVM training with *C* =1 using SVM software (svmTrain.m). You should find the decision boundary as show in figure 2.

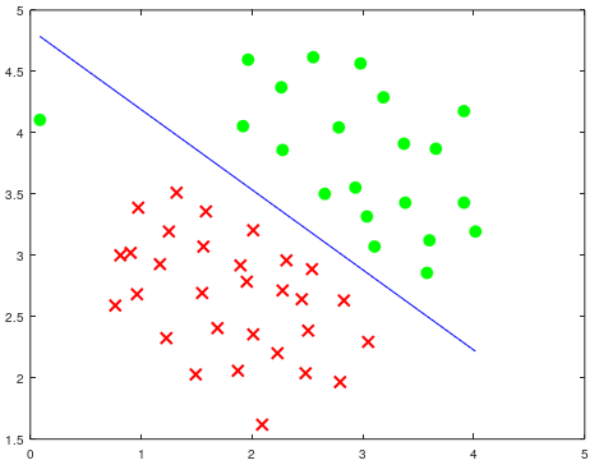

Figure 2 SVM Decision Boundary with C = 1 for dataset 1

Informally, the *C* parameter is a positive value that controls the penalty for misclassified training examples. A large *C* parameter tells the SVM to try to classify all the examples correctly. *C* plays a role similar to 1/*λ*, where *λ* is the regularization parameter that we were using previously for logistic regression.

Your task is to try different values of C on the dataset. Vary the *C* parameter in PA6test.m part 2. How does the C parameter affect the decision boundary? Report what you find in a .doc or .pdf file. You should include figures in your report.

Note: Most SVM software packages (including svmTrain.m) automatically add the extra feature $x_0 = 1$ for you and automatically take care of learning the intercept term θ_0. So when passing your training data to the SVM software, there is no need to add this extra feature $x_0 = 1$ yourself. In particular, in MATLAB your code should be working with training examples $x \in R^n$ (rather than $x \in R^{n+1}$); for example, in the first example dataset $x \in R^2$.

1.2. Example dataset 2

Task 3. Complete gaussianKernel.m

You will use SVMs with gaussian kernels (also called radical basis function kernel, RBF kernel) for non-linear classification. Complete the code in gaussianKernel.m to compute the Gaussian kernel between two examples, ($x^{(i)}$, $x^{(j)}$). The Gaussian kernel function is defined as:

$$K_{gaussian}\left(x^{(i)}, x^{(j)}\right) = \exp\left(-\frac{\left\|x^{(i)} - x^{(j)}\right\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{k=1}^{n}(x_k^{(i)} - x_k^{(j)})^2}{2\sigma^2}\right)$$

PA6test.m part 3 will test your gaussianKernel function. You should expect a value of 0.324652 if your guassianKernel is defined correctly.

PA6data2.mat contains a 2D example dataset that is not linearly separable. The script PA6test.m will plot the training data (Figure 3).

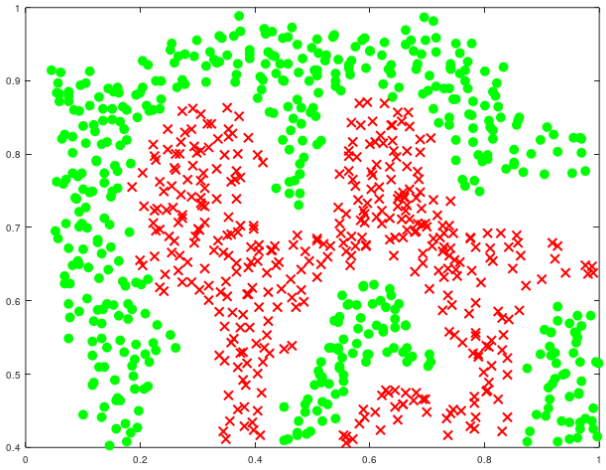

Figure 3 Dataset 2

After loading and visualizing the dataset from PA6data2.mat, part 4 in PA6test.m will run the SVM training with *C* =1, sigma = 0.1 using svmTrain.m with your gaussianKernel. You should find the decision boundary as show in figure 4.

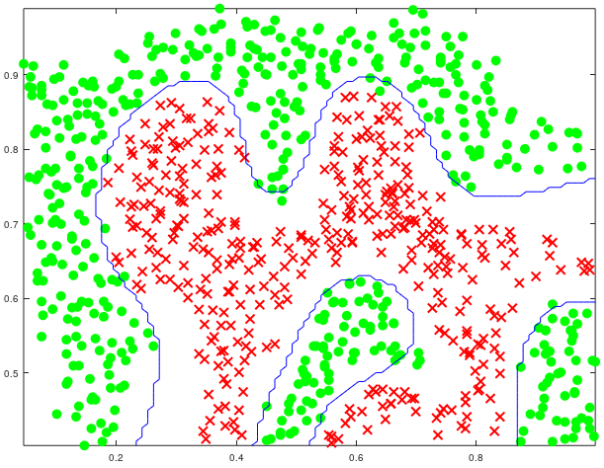

Figure 4 SVM (Gaussian Kernel) Decision Boundary for Example Dataset 2

2. Session 2

2.1. Example dataset 3

PA6data3.mat contains a different 2D example dataset that is also not linearly separable. The script PA6test.m part 6 will plot the training data. In this exercise you will perform a model selection for the best C and sigma for the dataset.

Task 4. Complete datset3Params.m

PA6.m will load and plot the dataset from PA6data3.m for you. You will have X, y, Xval, and yval in your environment. Xval and yval are the cross-validation dataset. Train SVM models with C = 0.01, 0.03, 0.1, 0.3, 1, 3, 10, or 30, and sigma = 0.01, 0.03, 0.1, 0.3, 1, 3, 10, or 30. As a result, you will train 8x8 = 64 SVM models. You will test your SVM models on the cross validation dataset.

You can use svmPredict to predict the labels on the cross validation set. For example,

```
predictions = svmPredict(model, Xval);
```

You can compute the prediction error using

```
mean(double(predictions ~= yval));
```

The combination of C and sigma that produces the smallest prediction error will be returned by dataset3Params.

PA6.m will used the selected C and sigma to compute the decision boundary for the training set and report C, sigma and minimal Error on screen.

If your dataset3Params.m runs correctly, you will see the minimal error = 0.03.

3. Session 3

Task 5. (optional, bonus 20 points) Apply SVM on PA2 dataset

Compare your result (accuracy of prediction on half-year death probability) with what we got using logistic regression in PA2. Upload your code and answers.

Submission and Grading

Session	Submitted File	Task	Points
Session I PA6test.m	linearKernel.m	Task1: Complete linearKernel.m	**10**
	Results	Task2: Examine the role of the *C* parameter	**10**
	gaussianKernel.m	Task3: Complete gaussianKernel.m	**10**
Session II PA6.m	datset3Params.m	Task4: Complete datset3Params.m	**20**
Session III (Bonus)	You own file(s)	Task5: Apply SVM on PA2 dataset. What is the best prediction accuracy you get?	**20**
Total points			**50+20**(bonus)