CS 520: Final exam

Nitin Mali

December 22, 2017

Question 3 - Classification

a) Construct a model to classify images as Class A or B, and train it on the indicated data. What does model predict for each of the unlabeled images? Give the details of your model, its training and the final result. Do the predictions make sense, to you?

Solution: The images in class A and class B both have 5x5 grids with black or white cells. So, based on the data provided in classA.txt and classB.txt each class labeled has 5x5 matrix with binary values 0(black) and 1(white). The location(index) in matrix represents if it's black or white. So i have decided to take all these 25 features for one specific class.

clas	A	B	C	D	E	F	G	H	I	J	K
-1	1	1	0	0	0	0	1	1	0	0	
-1	1	1	0	0	0	0	0	1	0	0	
-1	0	1	0	0	0	1	1	0	0	0	
-1	1	1	0	0	0	1	0	1	0	0	
-1	0	1	1	0	0	1	1	0	0	1	
1	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	1	0	0	0	0	
1	0	1	0	0	0	0	0	0	1	0	
1	1	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	1	0	0	1	

Figure 1: Training data to classify images

All unlabeled data

1	0	0	0	0	1	0	1	0	1
1	1	1	0	0	1	1	1	0	0
0	0	0	0	1	0	0	0	0	1
0	1	1	0	0	1	1	0	0	0
1	0	0	0	1	0	0	0	0	0

Figure 2: Prediction data to classify images

because there were only two classes , I was deciding upon classifier to use. I knew it was a binary classifier application. I selected Support vector Machines(SVM) for two reasons.

1. SVM are great for small data sets with less outlier.

2. SVM handles overfitting.

Hopefully SVM will find optimal hyperplane which maximizes margin between images of class A and class B. I am using Loss function and gradient descent.

Model:

It is a supervised machine learning which can be used for classification. We have 2 labeled classes of data. Classifier formally defined by an optimal hyperplane that separates are classes. New examples that are then mapped into same space which can be categorized based on which side of the gap they fall. Linear classification for N classes with M features , mapping that is a linear combination (Y=mx+b) or multi-dimensional (Y=x+z+b+q). So irrespective of dimensions/features set of 2 classes represents for mapping using Linear function.Following figures show the Loss function, Regularizer term and derivatuve to get gradient.

so the equation consist of two terms Regularizer and Loss. regularizer becomes the balance between margin maximation and loss.

Since we have to learn the rate of change in Loss. The derivative is also mentioned below.

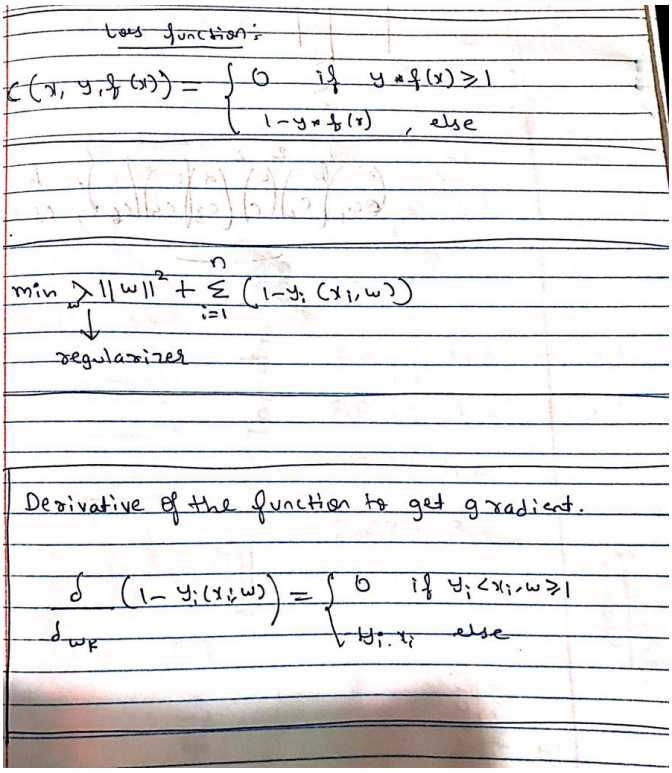

Figure 3: Equations

Updating of weights is done based on mis-classification and classification. If we have a mis-classified sample then the update is made on weight vector W using gradient of both terms else if classified correctly , W is updated by gradient of Regularizer.

W=W+η($y_i x_i$-2λw) –> Mis-classified

W=W+η(-2λw) –> Correctly Classified

So based on gradient descent , λ–> learning rate is length of steps algorithm makes down the gradient descent on error curve.

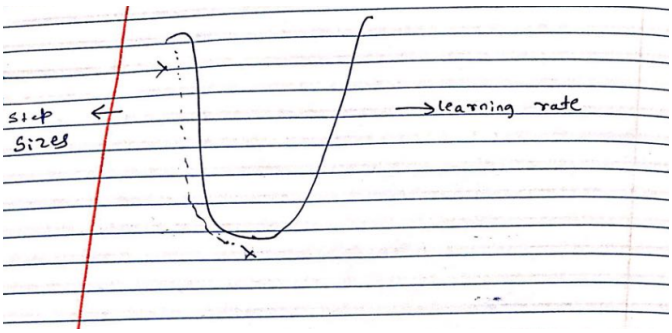

Figure 4: Learning

1. Learning rate too high (λ) algorithm might overshoot optimal low point and can get stuck in infinite loop.
2. Learning rate too low(λ) could take long time to converge or never converge.

The η Regularizer too high -> Overfit and the η Regularizer too low -> Underfit

I am trying to approximate a function and co-efficients are its weights and they are updated over time. I am using gradient descent to update and approximate function. Update weights are printed in the end for each of them. So for each of the unlabeled image based on the feaures, it is predicting few images accurately but few of them are not so accurate.
Below is the Figure which shows prediction.

```
This is my class labels where A=-1 and B=1
[-1 -1 -1 -1 -1  1  1  1  1  1]
[1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0]
B
[1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0]
A
[0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 1 1]
B
[0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0]
A
[1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1]
B
```

Figure 5: SVM prediction on Unlabeled data

1) B 2) A 3) B 4) A 5) B

The training and testing data is attahced in Zip file.
I have also plotted the misclassified data points with iterations. The error reduces as iterations increases.

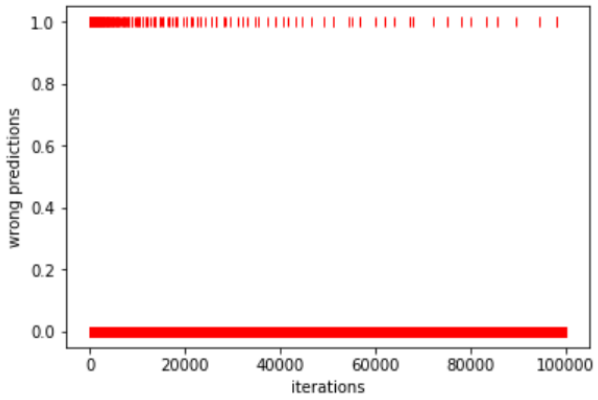

Figure 6: Error vs Misclassified

b)The data provided is quite small and overfitting is a serious risk. what steps can you take to avoid it?

Solution: SVM is very good for high dimensions but the data provided is too small and overfitting is a big task. I have selected svm because it handles overfitting in a better way. In svm the complexity of resulting classifier is characterized by number of support vectors rather than the dimensionality of the transformed space. Hence, they are less prone to problems of over-fitting then some other methods.

I tried adding noise to the data to avoid over-fitting but the result was negatively affected. I took off the noisy data and did some tuning for regularization. In machine learning model always depends on goodness of fit. It measures how well the approximation of function matches the target function. So, overfitting happens when a model learns the detail and noise in training data to extent that it negatively impacts the performance of the model on new data.

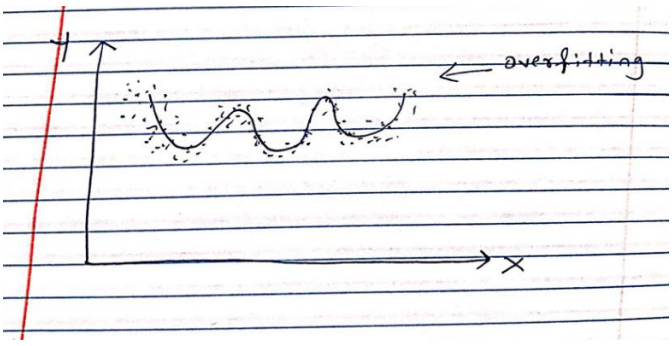

Figure 7: Overfitting

1) Maximise the margin and took soft margin into consideration.
2) Performance of the svm was sensitive to the tuning of Regularization(λ)or c parameter. Tuning to the extent can reduce.
3)I thought of reducing the features from 25 to 5 usinf feature selection or PCA but the sample was already too small.

So went with 25 features.

c) Construct and train a second type of model to.Give the details of your model, how do prediction compare to the first model?

Solution: When i thought about second model. I was confused. I thought of selecting other then logistic regression but end up choosing logistic. It is a linear classification algorithms for two-class problems.

Input values(X)are combined linearly using weights or coefficient values to predict an output value(y).In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In logistic regression, the outcome (dependent variable) has only a limited number of possible values. Logistic Regression is used when response variable is categorical in nature.

The logistic function, also called the sigmoid function is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.Where e is the base of the natural logarithms (Euler's number or and value is the actual numerical value that i want to transform.
Look at the below filgure.

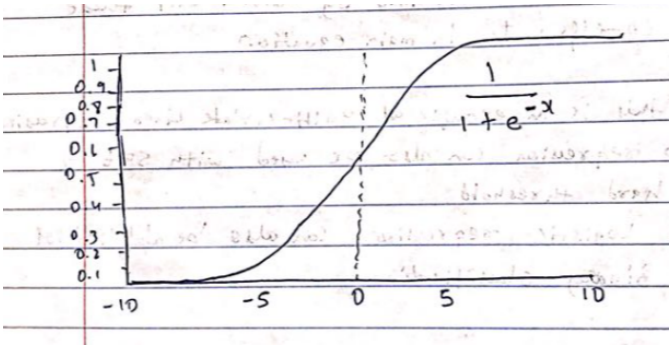

Figure 8: sigmoid function

so if the function is of the following:
$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_4.....\beta_nx_n$
where y is the logit function logit(y),β_0 is intercept,$\beta_1.....\beta_n$ are co-efficients and $x_1....x_n$ are features.

logarithm of OR(X), called the log-odds function. I am modeling the probability that an input (X) belongs to the default class (Y=1). The probability prediction must be transformed into a binary values (0 or 1) in order to actually make a probability prediction. Logistic regression is a linear method, but the predictions are transformed using the logistic function.
I am using a Gradient Descent where it is the process of minimizing a function by following the gradients of the cost function.

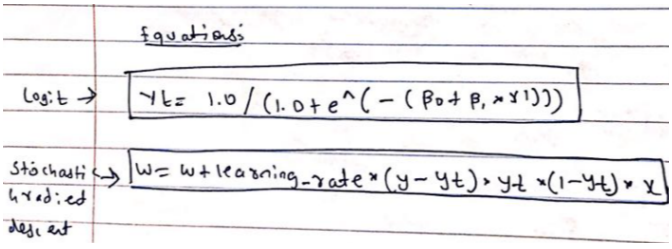

Figure 9: Equations

This involves knowing the form of the cost as well as the derivative.In machine learning, we can use a technique that evaluates and updates the coefficients every iteration called stochastic gradient descent to minimize the error of a model on our training data.The way this optimization algorithm works is that each training instance is shown to the model one at a time. The model makes a prediction for a training instance, the error is calculated and the model is updated in order to reduce the error for the next prediction.

This procedure can be used to find the set of coefficients in a model that result in the smallest error for the model on the training data. Each iteration, the coefficients w in machine learning language are updated using the equation:

I am trying to approximate a function and co-efficients are its weights and they are updated over time. I am using gradient descent to update and approximate function. Update weights are printed in the end for each of them. So for each of the unlabeled image based on the feaures, it is predicting few images accurately but few of them are not so accurate.
Below is the Figure which shows prediction.

```
This is my class labels where A=-1 and B=1
[-1 -1 -1 -1 -1  1  1  1  1  1]
[1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0]
B
[1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0]
A
[0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 1 1]
B
[0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 0]
A
[1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1]
A
```

Figure 10: SVM prediction on Unlabeled data

1) B 2) A 3) B 4) A 5) A

The training and testing data is attahced in Zip file.

When compared to previous model prediciton. There is only one change in my prediction. For the final prediction the Instance is Class B for SVM where as for Logistic it belongs to class A. The rest of the unlabeled data is predicting correctly. I have compared the prediction with a library called sci-kit learn. It is predicting same for both logistic and SVM. I tried tuning my learning rate for Logistic but the results were erroneous. so I hope these models work for other unlabeled images as well.

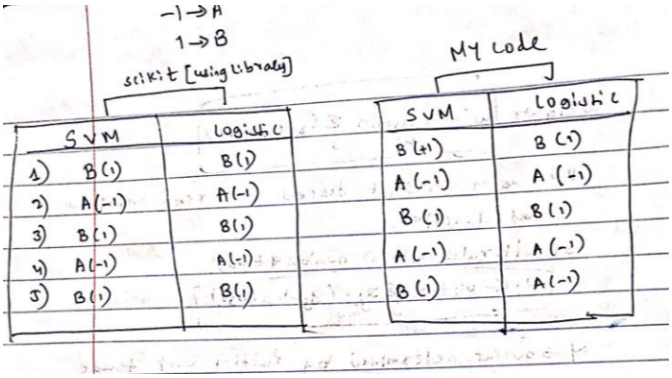

Figure 11: sci-kit learn vs my code