CS584 Assignment 4: Report

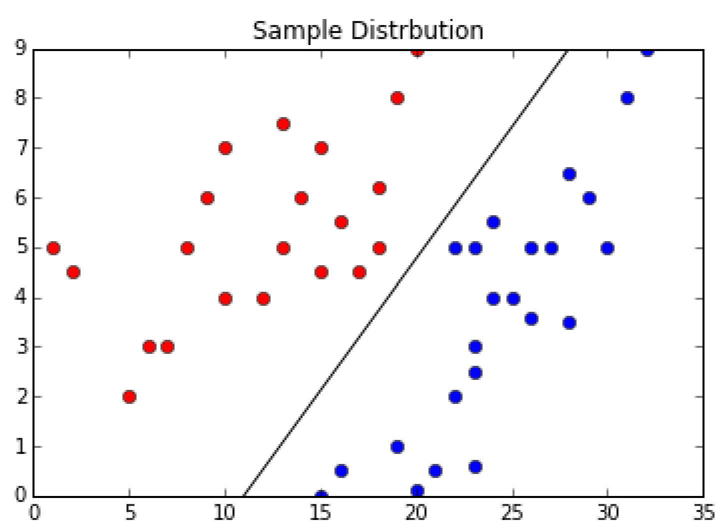
Gady Agam
Department of Computer Science
Illinois Institute of Technology
April 24, 2016

Abstract

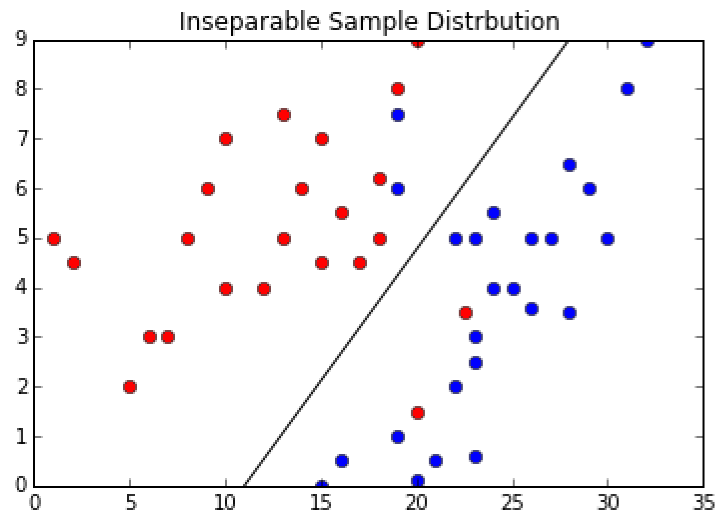
This report is for assignment in CS584. The problems that we confront is Support Vector Machine and its variants like Hard and Soft Margins, usage of linear, polynomial and Gaussian Kernels. Their performance using confusion matrix.

1. Problems Statement:
    a. Generating 2-D linearly separable and inseparable datasets.
    b. Implementation of hard and soft margin based Support Vector Machines algorithms.
    c. Derive the expression for dual objective function.
    d. Implementation of kernel based SVM algorithms.
    e. Effect of skewness in samples.
    f. Classification of external datasets using the implemented SVM algorithms.

2. Implemented Solution:
    a. Linearly Separable data was generated using MS-Excel functions. Its result is stored in Data_2D.csv file.



Class 1: Class 2:



Class 1: ● Class 2: ●

b. Using CVXOPT Solver we find the Lagrange multipliers
(i.e) solvers.qp(P,q,G,h,A,b)

Finding Lagrange multipliers

Using CVXOPT's Quadratic problem solver
$min_x 1/2 X^T P X + q^T X$

Computing Weights W and W_0

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} X^{(i)}$$

$$w_0 = 1/sv \sum_{X^i \in SV} (y^{(i)} - W^T . X^{(i)})$$

Then predict the Y values using the trained parameters.
The parameters differ for both hard and soft margin. The details on the implementation can be found in .ipynb file.

c. Derivation:

Soft margin

$$L_p = \frac{1}{2} \omega^T \omega + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i \left(y^{(i)} (\omega^T x^i + \omega_0) \right) - \sum_{i=1}^{m} \beta_i \xi_i$$

where

$\omega \rightarrow$ weights

$C \rightarrow$ user selected penalty

$\xi \rightarrow$ slack variable

$\alpha \rightarrow$ Lagrange multipliers

$m \rightarrow$ Sample size

$$\frac{\partial L_p}{\partial \omega} = 0 \Rightarrow \frac{1}{\cancel{2}} \cancel{2} \omega - \sum_{i=1}^{m} \alpha_i y^i x^i + 0 - 0$$

$$\omega - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

$$\boxed{\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}} \quad (1)$$

$$\frac{\partial L_p}{\partial \omega_0} = \frac{1}{2} \cancel{2} \omega - \sum_{i=1}^{m} \alpha_i y_i x^{(i)}$$

$$= 0 + 0 - \sum_{i=1}^{m} \alpha_i y_i \cancel{x^i} + 0$$

$$\Rightarrow \boxed{\sum_{i=1}^{m} \alpha_i y_i = 0} \quad (2)$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 + C - \alpha_i - \beta_i = 0$$

$$= \boxed{C - \alpha_i - \beta_i = 0} \quad - (3)$$

Substituting ①, ② and ③ in L_p

$$L_p = -\frac{1}{2}\left\{\sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)T}\right\}\left(\sum_{i=1}^{m}\alpha_i y^{(i)} x^{i}\right)$$

$$+ C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i y^{(i)}\left[\left(\sum_{j=1}^{m}\alpha_j y^{(j)} x^{(j)T}\right)x^{(i)}\right]$$

$$- \sum_{i=1}^{m}\cancel{\alpha_i y^{(i)}}\, w_0 + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\beta_i\xi_i$$

$$\Rightarrow \left[\because \sum_{i=1}^{m} C\xi_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\beta_i\xi_i = 0\right]$$

and $\sum\alpha_i y^{(i)} = 0$

$$\boxed{L_D = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)} y^{(j)} x^{(i)T} x^{j} + \sum_{i=1}^{m}\alpha_i}$$

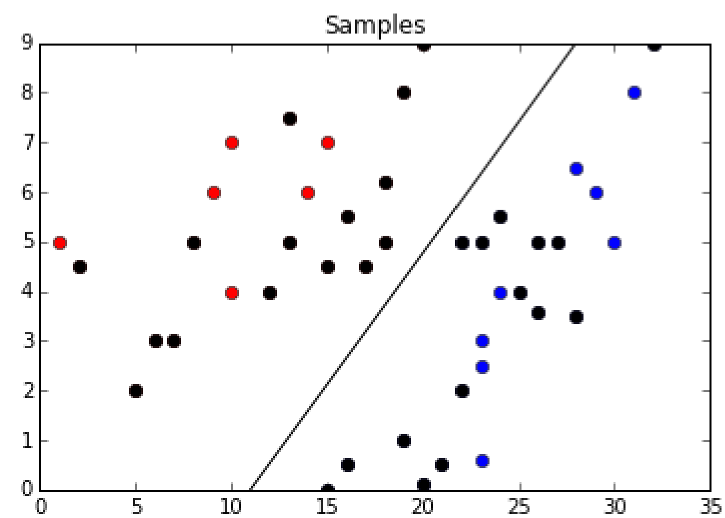
max L_D with Constraints

$$\alpha_i \geq 0, \quad \sum_{i=1}^{m}\alpha_i y^{(i)} = 0, \quad 0 \leq \alpha_i \leq C$$

d. Implementation and classification:

i. Separable Data using hard margin

```
     pcost       dcost       gap    pres   dres
 0: -1.9867e+00 -3.4177e+00  7e+01  7e+00  2e+00
 1: -1.8539e+00 -5.6995e-01  1e+01  1e+00  3e-01
 2: -5.5769e-02 -9.8809e-04  2e-01  3e-02  7e-03
 3: -1.3101e-03 -5.3670e-04  8e-03  8e-04  2e-04
 4: -3.3900e-05 -3.8881e-04  4e-04  3e-19  2e-15
 5: -1.4075e-04 -2.3367e-04  9e-05  8e-20  7e-16
 6: -2.0931e-04 -2.1716e-04  8e-06  6e-20  7e-16
 7: -2.1551e-04 -2.1559e-04  8e-08  2e-21  5e-16
Optimal solution found.
Accuracy        : 1.0
Recall          : 1.0
False Negative  : 0.0
Precision       : 1.0
False Positive  : 0.0
True Negative   : 1.0
F Square        : 1.0
```



Class 1: ● Class 2: ● Support Vectors: ●

Note: Hard Margin SVM was able to classify with an accuracy of 100% for the given sample data.

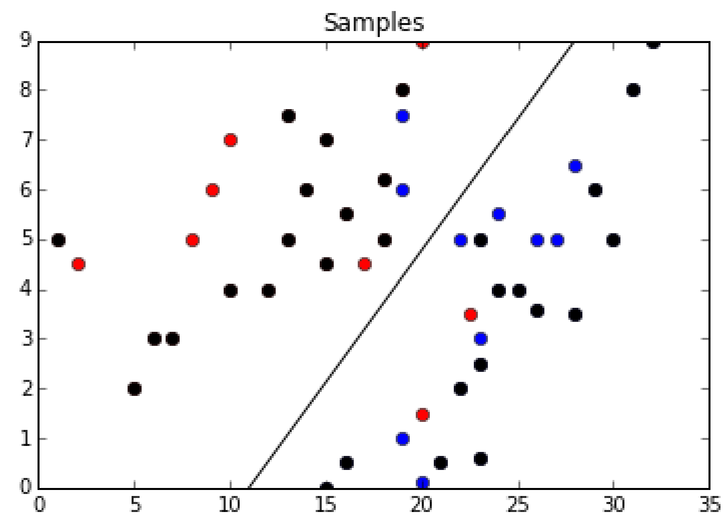
ii. Inseparable Data using soft margin:
Using soft margin, the classifier was not accurate, it predicted with an accuracy of 67%

Accuracy	: 0.666666666667
Recall	: 1.0
False Negative	: 0.0
Precision	: 0.6
False Positive	: 0.666666666667
True Negative	: 0.333333333333
F Square	: 0.75

iii. Inseparable Data using soft margin and Gaussian kernel

Usage of Gaussian kernel lead to a drastic improvement in the accuracy of the classifier. We can get close to 100% accuracy using Gaussian kernel.

```
     pcost       dcost       gap    pres   dres
 0: -1.8318e-01 -3.2828e+00  7e+01  8e+00  3e-16
 1: -1.7549e-01 -2.8663e+00  3e+00  1e-01  3e-16
 2: -1.6840e-01 -3.1845e-01  2e-01  1e-03  3e-16
 3: -1.7958e-01 -1.8946e-01  1e-02  4e-05  1e-16
 4: -1.8141e-01 -1.8288e-01  1e-03  5e-06  1e-16
 5: -1.8194e-01 -1.8203e-01  9e-05  1e-07  1e-16
 6: -1.8199e-01 -1.8199e-01  2e-06  1e-09  1e-16
 7: -1.8199e-01 -1.8199e-01  2e-08  1e-11  1e-16
Optimal solution found.
Accuracy        : 1.0
Recall          : 1.0
False Negative  : 0.0
Precision       : 1.0
False Positive  : 0.0
True Negative   : 1.0
F Square        : 1.0
```



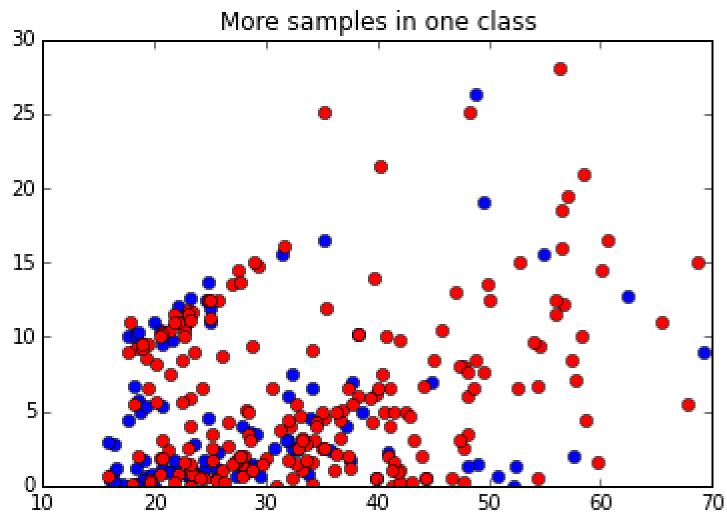
Class 1: ● Class 2: ● Support Vectors: ●

In comparison with linear kernel, Gaussian kernel based SVM algorithms performed well, as problem that are not linearly separable in 2-D space can become linearly separable in Higher Dimension by a linear plane whereas Gaussian kernel achieves the same result using a non-parametric approach. The intuition behind the improved accuracy while using Gaussian is usage of Euclidean distance measure between each and every point is considered while classification which tend to smoothen the PCA.

After conducting experiments with various C and Kernel values, we find that with c=0.2 and kernel = Gaussian Kernel works best.

e. We further experiment with two new datasets, Credit Score and Spam datasets.

In one experiment we chose the number of training samples to be skewed towards one class, we find that the classifier doesn't work well. It is due to the fact that number of support vectors are high whose sample size is high.



Class 1: Class 2:

Accuracy : 0.53
Recall : 1.0
False Negative : 0.0
Precision : 0.53
False Positive : 1.0
True Negative : 0.0
F Square : 0.692810457516

Number of samples with Class 1 - 158 Number of samples with Class 2 - 42

Print the Label of Support Vectors: Class1 - 39, Class 2 - 0