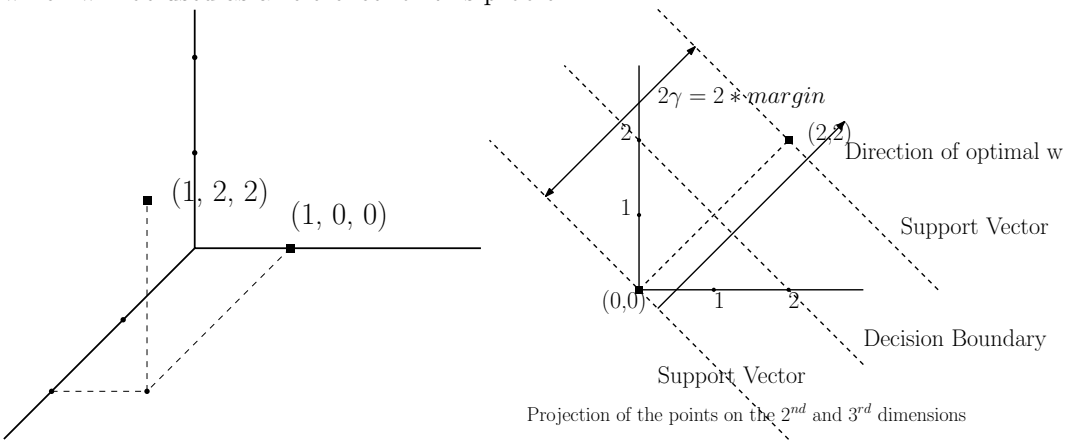
CSE 546 Machine Learning, Autumn 2013
Homework 3

Shrainik Jain, 1323338

1 Fitting an SVM classifier by hand [25 Points]

1. $\phi(x) = [1, \sqrt{2}x, x^2]^T$. This gives $\phi(data) = [1, \sqrt{2} * 0, 0^2]^T, [1, \sqrt{2} * \sqrt{2}, \sqrt{2}^2]^T = [1, 0, 0]^T, [1, 2, 2]^T$. The following figures show the decision boundaries and margin which will be used as a reference for this problem.



From the figures it is clear that optimal w is parallel to the line joining the points $[1, 0, 0]^T \& [1, 2, 2]^T$. One such

$$w = [0, 2, 2] \ \ ...(Ans) \tag{1}$$

2. As shown in the second figure the margin is half of the distance between the points $[1, 0, 0]^T \& [1, 2, 2]^T$.

$$\gamma = \frac{1}{2}\sqrt{(1-1)^2 + (2-0) + (2-0)^2} = \sqrt{2} \ \ ...(Ans) \tag{2}$$

3. Let the optimal $w = [w_1, w_2, w_3]$

$$\gamma = \frac{1}{||w||} \Rightarrow ||w|| = \frac{1}{\sqrt{2}} \Rightarrow (w_1^2 + w_2^2 + w_3^2) = \frac{1}{2} \tag{3}$$

From figures we see that w is parallel to [0, 2, 2] and passes through origin. Hence,

$$w_1 = 0 \ \& \ w_2 = w_3 \tag{4}$$

Plugging into the previous equation we get:

$$2w_2^2 = \frac{1}{2} \Rightarrow [w_1, w_2, w_3] = [0, \frac{1}{2}, \frac{1}{2}] \ \ ...(Ans) \tag{5}$$

4. We have:

$$y_1(w^T \phi(x_1) + w_0) \geq 1 \tag{6}$$

$$y_2(w^T \phi(x_2) + w_0) \geq 1 \tag{7}$$

from these equations and the results from the previous parts, we get:

$$-1(0 * 1 + 1/2 * 0 + 1/2 * 0 + w_0) \geq 1 \Rightarrow w_0 \leq -1 \tag{8}$$

and

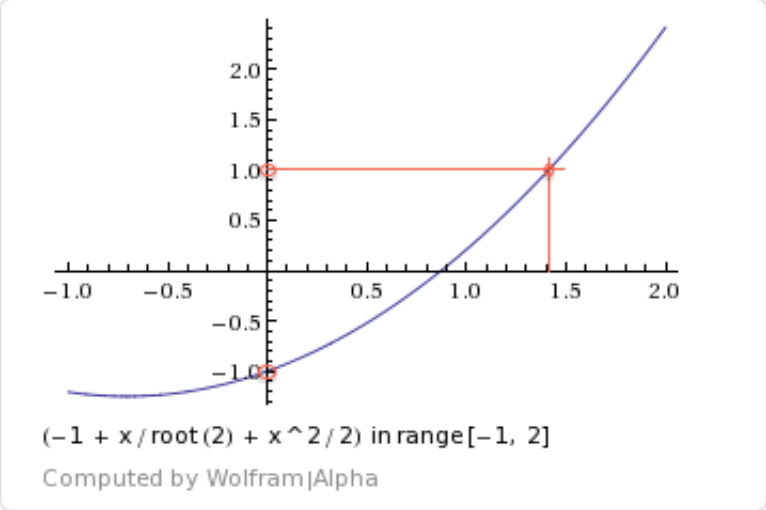
$$1(0 * 1 + 1/2 * 2 + 1/2 * 2 + w_0) \geq 1 \Rightarrow w_0 \geq -1 \tag{9}$$

This implies that $w_0 = -1$ ... (Ans)

5. Using the results from the previous parts:

$$f(x) = -1 + \frac{x}{\sqrt{2}} + \frac{x^2}{2} \tag{10}$$

The plot for the function:



2 Manual calculation of one round of EM for a GMM [30 points]

M step

1. The log likelihood function we are trying to optimize is:

$$Q(\theta, \theta^{(t-1)}) = \sum_i \sum_k r_{ic} log(\pi_c) + \sum_i \sum_k r_{ic} log(p(x_i|\theta_c)) \ ...(Ans) \tag{11}$$

   taken from Murphy, page 351.

2. We know that:

$$\pi_c = \frac{1}{N} \sum_i r_{ic} = \frac{r_c}{N} \tag{12}$$

   Hence,

$$\pi_1 = \frac{1}{3}(1 + 0.4 + 0) = \frac{1.4}{3} = \frac{7}{15} \ ...(Ans) \tag{13}$$

$$\pi_2 = \frac{1}{3}(0 + 0.6 + 1) = \frac{1.6}{3} = \frac{8}{15} \ ...(Ans) \tag{14}$$

3. We have:

$$\mu_c = \frac{\sum_i r_{ic} x_i}{r_c} \tag{15}$$

   Plugging in the values we get:

$$\mu_1 = \frac{1 * 1 + 0.4 * 10 + 0 * 20}{1 + 0.4 + 0} = \frac{5}{1.4} = \frac{25}{7} \ ...(Ans) \tag{16}$$

$$\mu_2 = \frac{0 * 1 + 0.6 * 10 + 1 * 20}{1 + 0.6 + 0} = \frac{26}{1.6} = \frac{65}{4} \ ...(Ans) \tag{17}$$

4. We have,

$$\Sigma_c = \frac{\sum_i (r_{ic})(x_i - \mu_c)(x_i - \mu_c)^T}{r_c} \tag{18}$$

   and

$$\sigma_c = \sqrt{\Sigma_c} \tag{19}$$

   Plugging in the values:

$$\Sigma_1 = \frac{1(1 - \frac{25}{7})^2 + 0.4(10 - \frac{25}{7})^2 + 0(20 - \frac{25}{7})^2}{1 + 0.4 + 0} = 16.53 \Rightarrow \sigma_1 = 4.065 \ ...(Ans) \tag{20}$$

$$\Sigma_2 = \frac{0(1 - \frac{65}{4})^2 + 0.6(10 - \frac{65}{4})^2 + 1(20 - \frac{65}{4})^2}{0 + 0.6 + 1} = 23.4375 \Rightarrow \sigma_2 = 4.84 \ ...(Ans) \tag{21}$$

E step

1. The probability of observation x_i belonging to cluster c:

$$r_{ic} = \frac{\pi_k p(x_i|\theta_k^{(t-1)})}{\sum_{c'} \pi_{c'} p(x_i|\theta_{c'}^{(t-1)})} \quad ...(Ans) \tag{22}$$

where probability follows a Gaussian distribution, i.e.:

$$p(x_i|\theta_c^{(t-1)}) = \frac{1}{\sigma_c\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{x_i - \mu_c}{\sigma_c})^2 \tag{23}$$

taken from Murphy, page 351.

2. Using the two equations above and plugging in the values, we get:
$p(x_1|\theta_1) = \frac{1}{4.065\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{1-\frac{25}{7}}{4.065})^2 = 0.0803$
$p(x_1|\theta_2) = \frac{1}{4.84\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{1-\frac{65}{4}}{4.84})^2 = 0.000575$
$p(x_2|\theta_1) = \frac{1}{4.065\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{10-\frac{25}{7}}{4.065})^2 = 0.028$
$p(x_2|\theta_2) = \frac{1}{4.84\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{10-\frac{65}{4}}{4.84})^2 = 0.0358$
$p(x_3|\theta_1) = \frac{1}{4.065\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{20-\frac{25}{7}}{4.065})^2 = 0.0000278$
$p(x_3|\theta_2) = \frac{1}{4.84\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{20-\frac{65}{4}}{4.84})^2 = 0.061$

Hence, $r_{11} = \frac{\frac{7}{15}0.0803}{\frac{7}{15}0.0803+\frac{8}{15}0.000575} = 0.992$
$r_{12} = 1 - r_{11} = 1 - 0.992 = 0.008$
$r_{21} = \frac{\frac{7}{15}0.028}{\frac{7}{15}0.028+\frac{8}{15}0.0358} = 0.406$
$r_{22} = 1 - r_{21} = 1 - 0.406 = 0.594$
$r_{31} = \frac{\frac{7}{15}0.0000278}{\frac{7}{15}0.0000278+\frac{8}{15}0.061} = 0.00039$
$r_{32} = 1 - r_{31} = 1 - 0.00039 = 0.99961$

$$R_{new} = \begin{pmatrix} 0.992 & 0.008 \\ 0.406 & 0.594 \\ 0.00039 & 0.99961 \end{pmatrix}$$

...(Ans)

3 Programming Question [45 Points]

3.1 Dataset

No question in this part.

3.2 Perceptron

1. From the lecture on Oct 28^{th}, slide 7, we know the following:

$$\hat{y} = sign(\phi(x).w^{(t)}) = sign(\sum_{j \in M^{(t)}} y^{(j)}\phi(x).\phi(x^{(j)})) \tag{24}$$

and

$$sign(\sum_{j \in M^{(t)}} y^{(j)} \phi(x).\phi(x^{(j)})) = sign(\sum_{j \in M^{(t)}} y^{(j)} k(x, x^{(j)})) \tag{25}$$

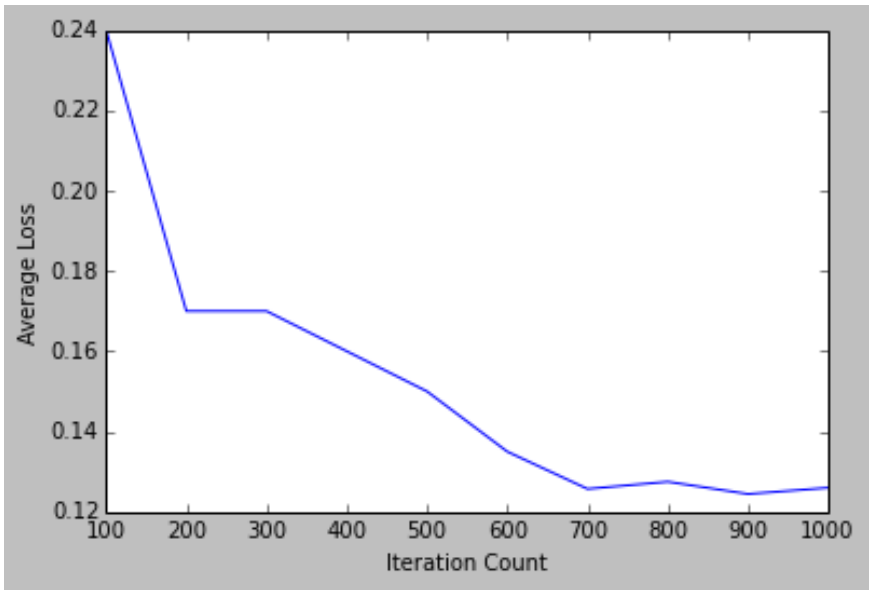
hence the prediction rule is:

$$\hat{y} = sign(\sum_{j \in M^{(t)}} y^{(j)} k(x, x^{(j)})) \ \ ...(Ans) \tag{26}$$

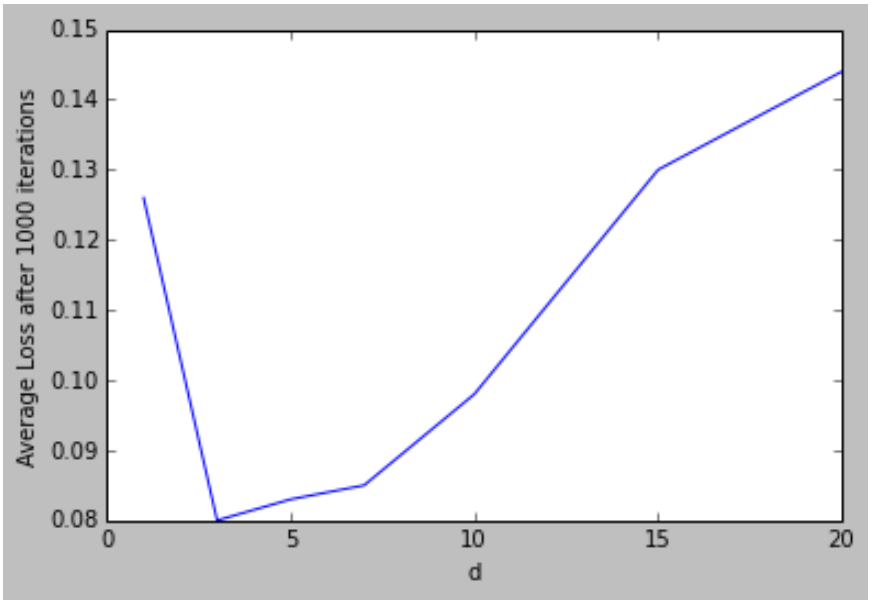
where $M^{(t)}$ = mistakes till iteration t.

2. See attached code.

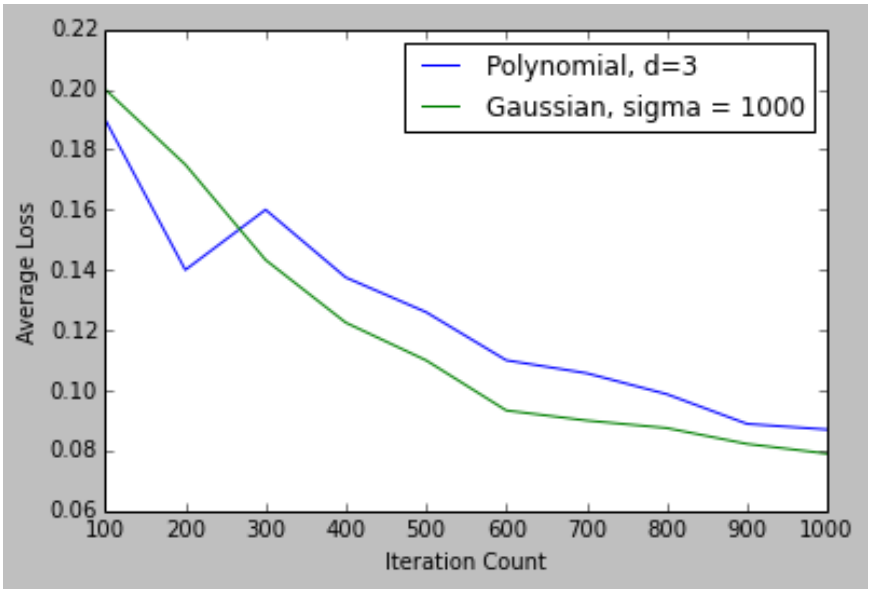
3. Plot for average loss at every 100 steps:



4. Plot for the average loss after 1000 iterations for the kernels with increasing degree of polynomials:



5. Plot for average loss at every 100 steps, polynomial kernel with d = 3 vs. gaussian kernel with $\sigma = 1000$:



3.3 SVM

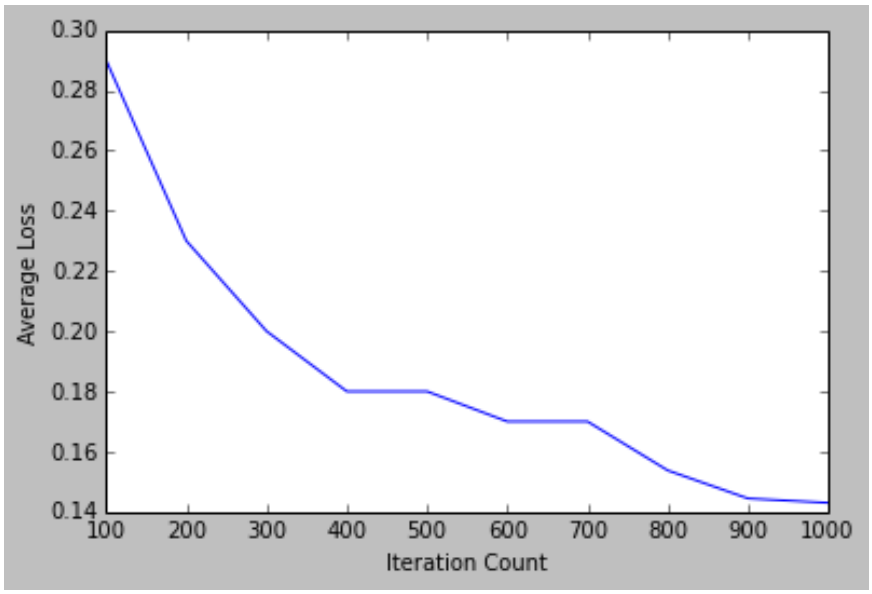
1. Update rules for linear SVM Stochastic gradient descent:

$$w^{(t+1)} = w^{(t)} + \eta(C\mathbb{I}((1 - y^{(t)}(w^{(t)}.x^{(t)} + w_0^{(t)}) > 0)) - 2w^{(t)}) \tag{27}$$

$$w_0^{(t+1)} = w_0^{(t)} + \eta C y^{(t)} \quad ...(Ans) \tag{28}$$

2. See attached code.

3. Average loss after every 100 steps for $\eta = 10^{-5}$ and $C = 1$:



4. For SVMs, C is the measure of how much we want to penalize misclassification of data, ie. the categorization error. The objective of SVM is:

$$argmin \ \ ||w||_2^2 + C \sum_{j=1}^{N} (1 - y^j(w.x^j + w_0))_+ \tag{29}$$

   We started the experiment with C=1 and got some decision rule d1, which misclassified some data points due to linear inseparablity of data. Irrespective of how much we increase C, we will always misclassify these specific linearly inseparable data points (since we are not using features of features). Hence the decision rule doesnt change. It just means that the margin keeps on decreasing.

5. As stated in the previous part, C is the measure of how much we want to penalize misclassification of data. If we increase C, we penalize the mistakes more and decrease the margin. A higher values of C imply that we will get good separation in the data but a smaller margin. Lower values of C imply that we want bigger margins and and care lesser about the number of mistakes in training.
   The relation with perceptron can be seen as the following:

   - Both SVM and perceptron try to minimize the hinge loss, but SVM has regularization.

- Lack of regularization in perceptron gives it an exact upper bound on the number of mistakes. In SVM, we can control this upper bound by the regularization constant C. Specifically, lower C $\Rightarrow$ higher bound on number of allowed errors. (which in turn implies we are willing to tolerate errors to get a better separation.)