CMPS 242: Homework #4

Due: *December 13, 2016*

S V N Vishwanathan

Kostas Zampetakis 1567380

Panos Karagiannis 1309484

Question 1

For this assignment generate the training dataset as follows: draw 1000 points from the 2-d Gaussian distribution with $\mu_+ = (10)$, $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and label them as +1. Similarly, draw another 1000 points from the 2-d Gaussian distribution with $\mu_+ = (10)$, $\Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$ and label them as −1. From the same distributions as above draw 500 points from each class for the test dataset.

(a) Download and install LibSVM from `http://www.csie.ntu.edu.tw/ cjlin/libsvm/`. Then, set $C = 1$ and train Support Vector Machines (SVM) with Gaussian Kernels whose kernel widths are given by $\gamma = \{1, 10, 100, 1000\}$. Finally, plot the decision boundary for each value of γ. Also report the test accuracy. Comment on what you observe.

(b) Fix $\gamma = 10$ and let $C \in \{1, 10, 100, 1000\}$. As before plot the decision boundary and report the test accuracy. What do you observe?

Answer:

(a) First we generate test and train points coming form a 2-d Gaussian distribution. By making use of the fact that that Σ_+ and Σ_- are both diagonal we can draw each coordinate of the vectors $\mathbf{x}_+$, $\mathbf{x}_-$ independently therefore simplyfing significantly our implementation. After downloading and installing LibSVM we use the Gaussian Kernel (RBF) with $C = 1$ with different parameter values for $\gamma = \{1, 10, 100, 1000\}$ in order to train our SVM. The formula for the Gaussian Kernel K is given by:

$$K(\mathbf{x}_1, \mathbf{x}_2) = exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$$

The Gaussian Kernel is a measure of similarity between two vectors. As we can see from the formula, as long as $\gamma > 0$ we can conclude that for points where $\|\mathbf{x}_1 - \mathbf{x}_2\|$ (i.e. the distance) is small the similarity is large, while the opposite is true as well. The parameter γ determines the width of the bell-shaped curve K. Increasing γ decreases the width of the curve, therefore decreasing the similarity of points. Based on these observations we expect that as we increase γ our misclassification rate will grow as well. This is because our model will not give emphasis in classifying points in the same "neighborhood" with the same label.

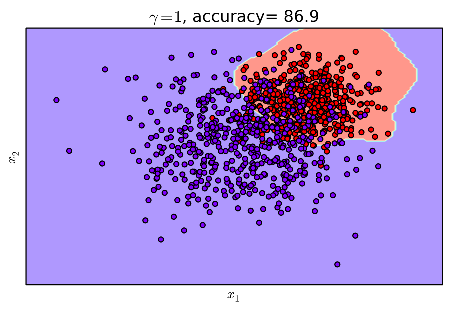
After reading the documentation of LibSVM we train the SVM using the command:
```
./svm-train -s 0 -c 1 -t 2 -g gamma values.train
```
And we test:
```
./svm-predict values.test values.train.model outFile.txt
```

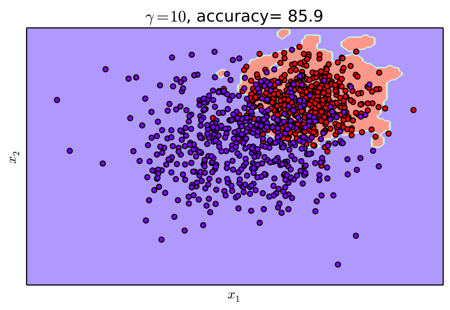
Where `gamma` is just a placeholder for the different values of γ. It is worth mentioning that we did not employ scaling since our data are within one order of magnitude. The classification percentages are displayed in Table 1:

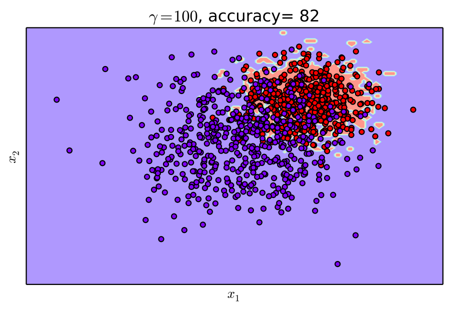
Table 1: Contraceptive Method γ

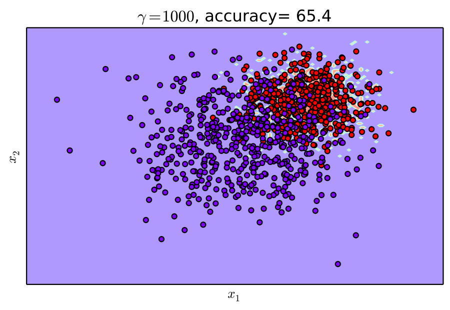
Age	Ster.	Other	None
15-19	3	61	232
20-24	80	137	400
25-29	216	131	301
30-34	268	76	203

Below follow the graphical representations of the decision boundaries obtained for each different value of γ:









(b) Now instead of varying the value of γ we set $\gamma = 10$ and we vary the value of the constant $C = \{1, 10, 100, 1000\}$. As we know, in the context of SVM we want to solve following convex minimization problem:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\| + \frac{C}{N}\sum_{i=0}^{N}\xi_i \\ \text{subject to} \quad & t_i(<\mathbf{w}, x_i> + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

The term ξ_i translates to the "slack" that we allow our model to have. It is related to the penalty that we pay for each misclassification that we need to make. Therefore, increasing

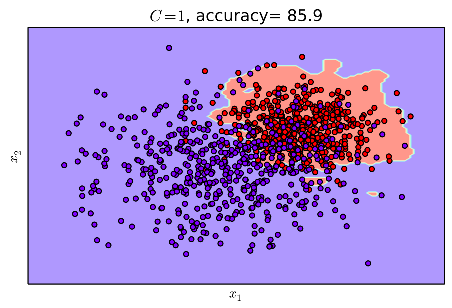
Question 1 [(b)] continued on next page...

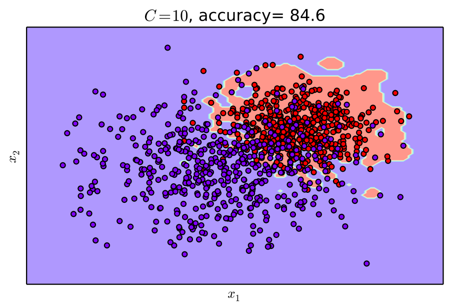
the value of C would imply that we need as few misclassifications in our training data as possible which leads to overfitting. Therefore, we would expect that as we increase the value of the paramater C the misclassification error becomes larger. The results we get by fixing $\gamma = 10$ are summarized in Table 2:

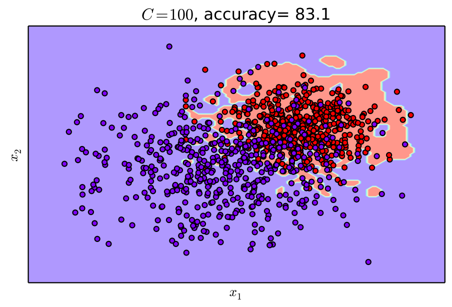
Table 2: Accuracy for different values of C

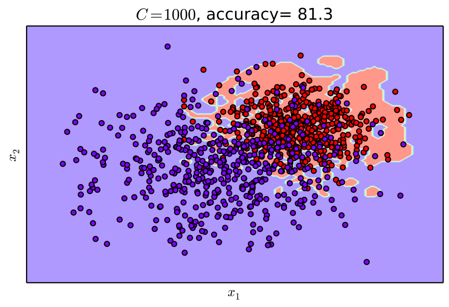
C	Accuracy(%)
1	85.9
10	84.6
100	83.1
1000	81.3

Below follow the graphical representations of the decision boundaries obtained for each value of C:









Question 2

Consider a dataset $\{(x_n, t_n)\}$ for $n = 1 \ldots N$, where $x_n \in \mathbb{R}^D$. For each of the statements below, answer either True or False. In either case justify your answer for full credit. In some cases, the answer can be neither True nor False.

(a) In the case of the linear SVM, it is always preferable to solve the Primal optimization problem.

Question 2 continued on next page. . .

(b) If $N \gg D$ then it is preferable to solve the Linear SVM problem in the dual.

(c) If we use the feature map $\phi(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^2 \end{bmatrix}$, where x is a vector given by squaring every entry of x, then the primal is a 2D dimensional optimization problem.

(d) In the above case, the dual is a 2N dimensional optimization problem.

Answer:

(a) FALSE: Solving the Primal problem would be much more inefficient in the case where $D \gg N$. Even if solving each problem gives the same result the complexities of the solutions are very different. This is obvious from the formulas of the two equivalent problems:

Primal:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\| + \frac{C}{M}\sum_{i=0}^{M}\xi_i \\ \text{subject to} \quad & t_i(<\mathbf{w}, x_i> + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

In the Primal we see that we optimize over $\mathbf{w}$ which is a D-dimensional vector. Therefore if the feature space is large, then it would imply large complexity in solving the above problem. On the contrary, by manipulating the Lagrangian we obtain the formula for the Dual:

Dual:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \alpha^T H\alpha - \alpha^T e \\ \text{subject to} \quad & \alpha^T \mathbf{t} = 0 \\ & 0 \leq \alpha \leq \frac{C}{M} \end{aligned}$$

Which is a minimization over the N-dimensional vector α. This means that now our problem only depends on the number of data points. Moreover in order to retrieve $\mathbf{w}$ we can calculate:

$$\mathbf{w} = \sum_{i=0}^{M} \alpha_i t_i x_i$$

But that still would very inefficient since we would have to store and maintain a D-dimensional vector. Instead we can use kernel functions to overcome this barrier. If we let K be the kernel associated with the RKHS we can say that:

$$<\mathbf{w}, x> = \sum_{i=0}^{M} \alpha_i t_i K(x_i, x)$$

To further illustrate why the Dual is often more efficient, consider the example of classification of emails. Then let $\mathbf{x}_i$ be the binary vector containing information on whether an email

Question 2 [(a)] continued on next page...

contains word i. Then if we attempt to incorporate bigrams and trigrams in in our vector $\mathbf{x}_i$ we see that the dimension D grows exponentially. Therefore, representing richer and richer concepts in our vector explodes the complexity in the Primal space. As a result, this is an example in which solving the Dual is a much better solution.

(b) Neither: Even if $N \gg D$ we don't know what is the dimension of the feature map $\phi(\mathbf{x})$. If the dimension of $\phi(\mathbf{x})$ is small then we should solve the primal problem, but since this information is not available to us we cannot decide whether the primal or the dual will have a smaller complexity.

(c) True: As we can see from our argument in Question(2a) the dimension of $\phi(\mathbf{x})$ constitutes the dimension of the primal minimization problem. Since the dimension of $\phi(\mathbf{x})$ is $2D$ then that implies that the primal is a $2D$ dimensional problem.

(d) False: That does not need to be the case since the number of data points N is independent of the D-dimensional feature space as well as the dimension of the feature map $\phi(\mathbf{x})$. So the dual could not be a 2N dimensional problem.

Question 3

For each of the datasets below, indicate if one should use a CRF for classification. Justify your answer.
(a) Scanned OCR digits,where the task is to predict the label $\in \{0 \dots 9\}$

(b) Stock price of AMZN, where the task is to predict the value of the stock in the future

(c) The set of movies that a user has watched, where the task is to predict the next movie that the user is going to watch

(d) News articles from NYTimes, where the task is to predict if a word is a proper noun.

Answer:

(a) False: There does not seem to be a sequential nature to this problem, since observing a number at position $i-1$ is independent of observing another number at position i. Therefore we could deal with this problem just by considering each number individually rather than a sequence of numbers. As a result CRF is not applicable.

(b) True: In this case the stock price at a given time depends on the price of the stock at previous times. The labels in this case can be any hidden factors that affect the price of the stock.

continued on next page. . .

(c) TRUE: We could use CRF to predict the next movie someone is going to watch since it is natural to assume that there is a dependency of past and future movies (ex. movie sequels). For example if someone liked a given actor in a specific movie genre, then he is much more likely to watch a movie of the same actor and genre as the previous one. Hence, there is a clear dependency of past and future movies that a user will watch.

(d) TRUE: In this case using CRF would be a good solution since the label of an improper noun depends on the previous sequence of words.

Question 4

(a) Consider a Hidden Markov Model where the the hidden state y takes on two values namely 1 and 2, with the corresponding state transition matrix given by $M = \begin{bmatrix} 0.3 & 0.7 \\ 0.1 & 0.9 \end{bmatrix}$ Moreover, let the observed state x take on values 0 or 1. The emission probabilities are given by:

$$p(x = 0 \mid y = 1) = 0.1$$
$$p(x = 1 \mid y = 1) = 0.9$$
$$p(x = 0 \mid y = 2) = 0.5$$
$$p(x = 1 \mid y = 2) = 0.5$$

Use the forward-backward algorithm to compute the probability of observing the sequence 0101. You may assume that at the beginning both the hidden states have equal probability.

<u>**Answer:**</u>

(a) First we need to calculate the probability of observing any given sequence $\mathbf{x}$. In order to do this we use the formula:

$$P(\mathbf{x}) = \sum_t P(\mathbf{x}, t) = \sum_t P(\mathbf{x} \mid t) P(t)$$

Using the Markovian property and letting $\mathbf{x} = \{\zeta_1 \dots \zeta_D\}$ and the corresponding $t = \{y_1 \dots y_D\}$, (where $y_i \in \{1, 2, \cdots, l\}$) we can further expand the expression to obtain:

$$P(\mathbf{x}) = \sum_{y_1 \dots y_D} \prod_{i=1}^{D} P(\zeta_i \mid y_i) P(y_i \mid y_{i-1})$$

Now if we let:

$$\Psi_i(\zeta_i, y_i, y_{i-1}) = P(\zeta_i \mid y_i) P(y_i \mid y_{i-1})$$

Then we simply have to calculate

$$\sum_{y_1 \dots y_D} \prod_{i=1}^{D} \Psi_i(\zeta_i, y_i, y_{i-1})$$

In order to calculate this sum efficiently we employ the forward algorithm used in class to reduce its complexity. We re-write the expression in an equivalent way by pushing summation inside the product, to get the following result:

$$\sum_{y_1 \dots y_D} \prod_{i=1}^{D} \Psi_i(\zeta_i, y_i, y_{i-1}) = [1 \dots 1] \left[\prod_{i=2}^{D} \psi_i\right] [P(\zeta_1|y_1 = 1)P(y_1 = 1) \dots P(\zeta_1|y_1 = l)P(y_1 = l)]^T$$

where now ψ_i are square matrices such that the kj entry is equal to:

$$[\psi_i]_{kj} = \Psi(\zeta_i, k, j), \quad k, j \in \{1, 2, \cdots, l\}$$

In this example we see that $\mathbf{x} = 0101$. Applying these results to our case we get that $D = 2$, $x_i \in \{0, 1\}$ and $y_i \in \{1, 2\}$. By assumption we know that:

$$[P(\zeta_1 = 0|y_1 = 1)P(y_1 = 1), P(\zeta_1 = 0|y_1 = 2)P(y_1 = 2)]^T = [0.1{\times}0.5, 0.5{\times}0.5]^T = [0.05, 0.25]^T$$

Next we show how to calculate ψ_2:

$$\psi_2 = \begin{bmatrix} P(\zeta_2 = 1 \mid y_2 = 1)P(y_2 = 1 \mid y_1 = 1) & P(\zeta_2 = 1 \mid y_2 = 1)P(y_2 = 1 \mid y_1 = 2) \\ P(\zeta_2 = 1 \mid y_2 = 2)P(y_2 = 2 \mid y_1 = 1) & P(\zeta_2 = 1 \mid y_2 = 2)P(y_2 = 2 \mid y_1 = 2) \end{bmatrix} = \begin{bmatrix} 0.9 \times 0.3 & 0.9 \times 0.1 \\ 0.5 \times 0.7 & 0.5 \times 0.9 \end{bmatrix}$$

In a similar manner we calculate ψ_3 to get:

$$\psi_3 = \begin{bmatrix} 0.1 \times 0.3 & 0.1 \times 0.1 \\ 0.5 \times 0.7 & 0.5 \times 0.9 \end{bmatrix} = \begin{bmatrix} 0.03 & 0.01 \\ 0.35 & 0.45 \end{bmatrix}$$

Since $\zeta_2 = \zeta_4 = 1$ we can conclude that $\psi_4 = \psi_2$. Hence the final expression that we obtain is:

$$P(\mathbf{x}) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.27 & 0.09 \\ 0.35 & 0.45 \end{bmatrix} \begin{bmatrix} 0.03 & 0.01 \\ 0.35 & 0.45 \end{bmatrix} \begin{bmatrix} 0.27 & 0.09 \\ 0.35 & 0.45 \end{bmatrix} \begin{bmatrix} 0.05 & 0.25 \end{bmatrix}^T =$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.0070416 & 0.032828 \end{bmatrix}^T = 0.0398696$$