Machine Learning (CS 6140)
Homework 2

Instructor: Ehsan Elhamifar
Due Date: November 9, 2017, 11:45am

1. Logistic Regression. We consider the following models of logistic regression for a binary classification with a sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$,

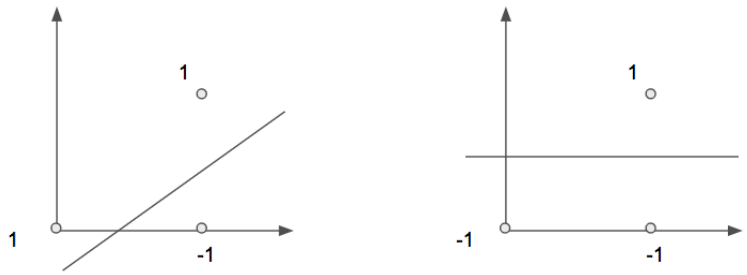
Model 1: $P(Y = 1|X, w_1, w_2) = \sigma(w_1X_1 + w_2X_2)$
Model 2: $P(Y = 1|X, w_0, w_1, w_2) = \sigma(w_0 + w_1X_1 + w_2X_2)$

We have three training examples:

$$(\boldsymbol{x}^1, y^1) = ([1, 1]^\top, 1),\ (\boldsymbol{x}^2, y^2) = ([1, 0]^\top, -1),\ (\boldsymbol{x}^3, y^3) = ([0, 0]^\top, 1)$$

A. How does the the learned value of $\boldsymbol{w} = (w_1, w_2)$ change if we change the label of the third example to -1? How about in Model 2? Explain (Hint: think of the decision boundary on 2D plane.)

Solution: It does not matter in Model 1 because $x^3 = (0, 0)$ makes $w_1x_1 + w_2x_2$ always zero and hence the likelihood of the model does not depend on the value of $\boldsymbol{w}$. But it does matter in Model 2. As for Model 2, the decision boundary before and after changing the label of the third example is shown as below.



B. Now, suppose we train the logistic regression model (Model 2) based on the N training examples $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ and labels $y^1, \ldots, y^n$ by maximizing the penalized log-likelihood of the labels:

$$\sum_i \log P(y^i|\boldsymbol{x}^i, \boldsymbol{w}) - \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

For large λ (strong regularization), the log-likelihood terms will behave as linear functions of $\boldsymbol{w}$.

$$\log \sigma(y^i \boldsymbol{w}^\top \boldsymbol{x}^i) \approx \frac{1}{2} y^{(i)} \boldsymbol{w}^\top \boldsymbol{x}^i$$

Express the penalized log-likelihood using this approximation (with Model 1), and derive the expression for MLE $\boldsymbol{w}$ in terms of λ and training data $\{(\boldsymbol{x}^i, y^i)\}_{i=1}^N$. Based on this, explain how $\boldsymbol{w}$ behaves as λ increases.

Solution: For large λ, the log-likelihood of the labels is

$$\begin{aligned} L(\boldsymbol{w}) &= \sum_i \log P(y^i|\boldsymbol{x}^i, \boldsymbol{w}) - \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \\ &= \sum_i \log \sigma(y^i \boldsymbol{w}^T \boldsymbol{x}^i) - \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \\ &\approx \sum_i \frac{1}{2} y^i \boldsymbol{w}^T \boldsymbol{x}^i - \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \end{aligned}$$

$\frac{\partial^2 L(\boldsymbol{w})}{\partial \boldsymbol{w}^2} = -\frac{\lambda}{2} < 0$, so the likelihood is concave and has a global maximum (i.e., negative likelihood is convex and has a global minima). To find the ML estimate set $\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$:

$$\begin{aligned} &\sum_i \frac{1}{2} y^i \boldsymbol{x}^i - \frac{\lambda}{2} * 2\hat{\boldsymbol{w}} = 0 \\ &\implies \hat{\boldsymbol{w}} = \frac{1}{2\lambda} \sum_i y^i \boldsymbol{x}^i \end{aligned}$$

As λ increases, $\hat{\boldsymbol{w}}$ decreases and goes toward zero.

2. Support Vector Machine. Consider a binary classification problem in one-dimensional space where the sample contains four data points $S = \{(1, -1), (-1, -1), (2, 1), (-2, 1)\}$ as shown in Fig. 1.

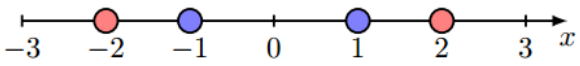


Figure 1: Red points represent instances from class +1 and blue points represent instances from class -1.

A. Define $H_t = [t, \infty)$. Consider a class of linear separateors $\mathcal{H} = \{H_t : t \in \mathbb{R}\}$, i.e., for $\forall H_t \in \mathcal{H}$, $H_t(x) = 1$ if $x \geq t$ otherwise -1. Is there any linear separator $H_t \in \mathcal{H}$ that achieves 0 classification error on this sample? If yes, show one of the linear separators that achieves 0 classification error on this example. If not, briefly explain why there cannot be such linear separator.

solution: There is no such linear separator (obvious by visual inspection). To prove, assume that there exists a separator $H_t(x)$ which could achieve 0 classification error on this sample. Then there are only two possibilities:
a) $H_t(-2) = 1$, $H_t(2) = 1$ and $H_t(-1) = -1$ and $H_t(1) = -1$;
b) $H_t(-2) = -1$, $H_t(2) = -1$ and $H_t(-1) = 1$ and $H_t(1) = 1$.

In case a), we have that i) $-2 \geq t$, ii) $2 \geq t$, iii) $-1 < t$ and iv) $1 < t$. From i and ii, we have that $t \leq -2$. From iii and iv we have that $t > 1$. It is not possible that all these four conditions are satisfied. Thus a) is not possible. Similarly, we can show that b) is not possible.

B. Now consider a feature map $\phi : \mathbb{R} \to \mathbb{R}^2$ where $\phi(x) = (x, x^2)$. . Apply the feature map to all the instances in sample S to generate a transformed sample $S' = \{(\phi(x), y) : (x, y) \in S\}$.Let $\mathcal{H}' = \{ax_1 + bx_2 + c \geq 0 : a^2 + b^2 \neq 0\}$ be a collection of half-spaces in R^2. More specifically, $H_{a,b,c}((x_1, x_2)) = 1$ if $ax_1 + bx_2 + c \geq 0$ otherwise -1. Is there any half-space $H' \in \mathcal{H}'$ that achieves 0 classification error on the transformed sample S'? If yes, give the equation of the max-margin linear separator and compute the corresponding margin.

Solution: Yes. $x_2 = 2.5$ is the max-margin linear separator achieves 0 classification error on the transformed sample S'. The corresponding margin is 1.5.

C. What is the kernel corresponding to the feature map $\phi(\cdot)$ in the last question, i.e., give the kernel function $K(x, z) : \mathbb{R} \times \mathbb{R} \to R$.

Solution: The kernel function is $K(x, z) = \phi(x)^T \phi(z) = xz + x^2 z^2 = xz(1 + xz)$.

3. Constructing Kernels. In this question you will be asked to construct new kernels from existing kernels. Suppose $K_1(x, z) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and $K_2(x, z) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ are both kernels, show the following functions are also kernels:

A. $K(x, z) = c_1 K_1(x, z) + c_2 K_2(x, z)$ with $c_1, c_2 \geq 0$.

Solution: Since both $K_1(x, z)$ and $K_2(x, z)$ are kernels, there exists ϕ_1 as a feature map for K_1 and ϕ_2 as a feature map for K_2, where we have $K_1(x, z) = \phi_1(x)^T \phi_1(z)$ and $K_2(x, z) = \phi_2(x)^T \phi_2(z)$. Thus we can write

$$\begin{aligned} K(x, z) =& c_1 K_1(x, z) + c_2 K_z(x, z) \\ =& c_1 \phi_1(x)^T \phi_1(z) + c_2 \phi_2(x)^T \phi_z(z) \\ =& \begin{bmatrix} \sqrt{c_1}\phi_1(x)^T & \sqrt{c_2}\phi_2(x)^T \end{bmatrix} \begin{bmatrix} \sqrt{c_1}\phi_1(z) \\ \sqrt{c_2}\phi_2(z) \end{bmatrix} \end{aligned} \tag{1}$$

Thus, there exists a feature map:

$$\phi(x) = \begin{bmatrix} \sqrt{c_1}\phi_1(x) \\ \sqrt{c_2}\phi_2(z) \end{bmatrix}$$

so that $K(x, z) = \phi(x)^T \phi(x)$. Thus $K(x, z)$ is a kernel.

B. $K(x, z) = K_1(x, z) \cdot K_2(x, z)$.

Solution: Let

$$\phi_1(x) = [f_1(x), f_2(x), \ldots]$$
$$\phi_2(x) = [g_1(x), g_2(x), \ldots]$$

be the feature map of K_1 and K_2 respectively. We have:

$$\begin{aligned} K(x,z) =& K_1(x,z) \cdot K_z(x,z) \\ =& \phi_1(x)^T \phi_1(z) \phi_2(x)^T \phi_2(z) \\ =& \left(\sum_{i=1}^{\infty} f_i(x) f_i(z) \right) \left(\sum_{j=1}^{\infty} g_j(x) g_i(z) \right) \\ =& \sum_{i,j} f_i(x) f_i(z) g_j(x) g_j(z) \\ =& \sum_{i,j} (f_i(x) g_j(x))(f_i(z) g_j(z)) \\ =& \sum_{i,j} h_{i,j}(x) h_{i,j}(z) \end{aligned} \tag{2}$$

where, we defined $h_{i,j} = f_i(x) g_j(x)$. As a result, we can write $K(x,z) = \phi(x)^T \phi(x)$, where $\phi(x)$ is a $M \times N$-dimensional vector of all pairs of $h_{i,j}$. Hence, $K(x,z)$ is a kernel.

C. Let $q(t) = \sum_{i=0}^{p} c_i t^i$ be a polynomial function with nonnegative coefficients, i.e., $c_i \geq 0, \forall i$. Show that $K(x,z) = q(K_1(x,z))$ is a kernel.

Solution: We have $K(x,z) = q(K_1(x,z)) = \sum_{i=0}^{p} c_i K_1(x,z)^i$. First, we want to prove that $K_1(x,z)^i$ ($i \geq 0$, $i \in \mathcal{Z}$) is a kernel. We prove this by induction:
1) When $i = 1$, we have $K_1(x,z)^1 = K(x,z)$, thus $K_1(x,z)^i$ being a kernel holds for $i = 1$.
2) We want to prove that if $K_1(x,z)^i$ is a kernel when $i = k$, then it still holds when $i = k+1$. This is easy to show, since $K_1(x,z)^{k+1} = K_1(x,z)^k K_1(x,z)^1$. From part B, we know that product of kernels is a valid kernel.
Next, we want to prove that $K(x,z)$ is a kernel for any p. This also follows easily from part A, since write $K(x,z) = \sum_{i=0}^{p} c_i K_1(x,z)^i$, and as we know sum of positively scaled valid kernels is also a kernel.

D. $K(x,z) = \exp(K_1(x,z))$. (Hint: you can use the previous results to prove this.)

Solution: Using Taylor series expansion of exponential function, we have

$$\begin{aligned} \exp(x) &= \lim_{i \to \infty} 1 + x + \cdots + \frac{x^i}{i!} \\ &= \lim_{p \to \infty} \sum_{i=0}^{p} c_i x^i \\ &= \lim_{p \to \infty} q(x) \end{aligned} \tag{3}$$

In part C, we proved that $\forall p, q(K_1(x,z))$ is a kernel. Thus, $K(x,z) = \exp(K_1(x,z))$ is a kernel.

E. Let A be a positive semidefinite matrix and define $K(x,z) = x^T A z$.

Solution: Since A is a positive semi-definite matrix, there exists B that $A = B^T B$. Thus, we have $K(x,z) = x^T A z = x^T B^T B z = (Bx)^T (Bz) = \phi(x)\phi(z)$ where $\phi(x) = Bx$. Thus $K(x,z)$ is a kernel.

F. $K(x,z) = \exp(-\|x-z\|_2^2)$.

solution: We can write

$$\begin{aligned} K(x,z) &= \exp(-\|x-z\|_2^2) \\ &= \exp(-(x-z)^T(x-z)) \\ &= \exp(-x^Tx - z^Tz + 2x^Tz) \\ &= \big(\exp(-x^Tx)\exp(-z^Tz)\big)\exp(2x^Tz) \end{aligned} \tag{4}$$

If we define $K_1(x,z) = 2x^Tz = \phi_1(x)^T\phi_1(z))$ where $\phi_1(x) = \sqrt{2}x$, then we have $K_1(x,z)$ is a kernel. Then according to part D, $K_2(x,z) = \exp(2x^Tz) = \exp(K_1(x,z))$ must be a kernel. Let $K_3(x,z) = \phi_3(x)\phi_3(z)$, where $\phi_3(x) = \exp(-x^Tx)$, then $K_3(x,z)$ is also a kernel. So $K(x,z) = K_3(x,z) \cdot K_2(x,z)$ is a kernel.

4. Support Vectors. In question 2, we explicitly constructed the feature map and find the corresponding kernel to help classify the instances using linear separator in the feature space. However in most cases it is hard to manually construct the desired feature map, and the dimensionality of the feature space can be very high, even infinity, which makes explicit computation in the feature space infeasible in practice. In this question we will develop the dual of the primal optimization problem to avoid working in the feature space explicitly. Suppose we have a sample set $S = (x_1, y_1), ..., (x_n, y_n)$ of labeled examples in $\mathbb{R}^d$ with label set $\{+1, -1\}$. Let $\phi : \mathbb{R}^d \to R^D$ be a feature map that transform each input example to a feature vector in $\mathbb{R}^D$. Recall from the lecture notes that the primal optimization of SVM is given by

$$\begin{aligned} \underset{\mathbf{w},\xi_i}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(x_i)) \geq 1 - \xi_i \qquad \forall i = 1, \cdots, n \\ & \xi_i \geq 0 \qquad \forall i = 1, \cdot, n \end{aligned}$$

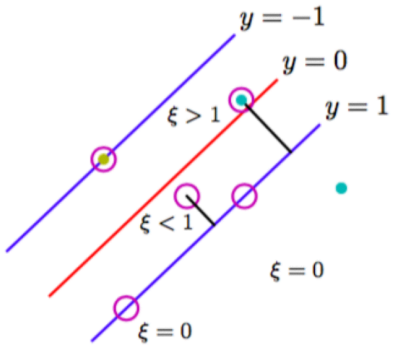
which is equivalent to the following dual optimization

$$\begin{aligned} \underset{\alpha_i}{\text{minimize}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \qquad \forall i = 1, \ldots, n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \forall i = 1, \ldots, n \end{aligned}$$

Recall from the lecture notes $\xi_1, ..., \xi_n$ are called slack variables. The optimal slack variables have intuitive geometric interpretation as shown in Fig. 3. Basically, when $\xi_i = 0$, the corresponding feature vector $\phi(x_i)$ is correctly classified and it will either lie on the margin of the separator or on the correct side of the margin. Feature vector with $0 < \xi_i \leq 1$ lies within the margin but is still be correctly classified. When $\xi_i > 1$, the corresponding feature vector is misclassified. Support vectors correspond to the instances with $\xi_i > 0$ or

instances that lie on the margin. The optimal vector $\mathbf{w}$ can be represented in terms of $\alpha_i, i = 1, \cdots, n$ as $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)$.

A. Suppose the optimal $\xi_1, ..., \xi_n$ have been computed. Use the ξ_i to obtain an upper bound on the number



of misclassified instances.

Solution: The upper bound on the number of misclassified instances is $\sum_{i=1}^{n} \xi_i$. If the ith data point is misclassified by the hyperplane, then we have $y_i(\mathbf{w}^\top \phi(x_i)) \leq 0$, i.e., $\xi_i \geq 1 - y_i(\mathbf{w}^\top \phi(x_i)) \geq 1$. Thus, $\sum_{i=1}^{n} \xi_i$ provides an upper bound of the number of misclassified instances.

B. In the primal optimization of SVM, what?s the role of the coefficient C? Briefly explain your answer by considering two extreme cases, i.e., $C \to 0$ and $C \to \infty$.

Solution: C is used to control the tolerance of misclassification. When $C \to \infty$, we are emphasizing on constraints on the slack variable, i.e., we hope that $\sum_{i=1}^{n} \xi_i \to 0$ and subsequently $\xi_i = 0$. So large C means we want most data points to be correctly classified and very few data points violate the margin condition. When $C \to 0$, we put less emphasis on the constraint on the slack variables, and we allow the model to have more violations of the margin condition (more points can fall on the wrong side of the hyperplane).

C. Explain how to use the kernel trick to avoid the explicit computation of the feature vector $\phi(\mathbf{x}_i)$? Also, given a new instance $\mathbf{x}$, , how to make prediction on the instance without explicitly computing the feature vector $\phi(\mathbf{x})$?
Solution: By using kernel, we could replace $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ in the objective function with $K(x_i, x_j)$. Thus, we don't have to compute $\phi(\mathbf{x}_i)$. Since we have that $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)$, So given a new instance $\mathbf{x}$, we can predict it by:

$$\begin{aligned} \mathbf{w}^T x &= \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) \\ &= \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \end{aligned} \tag{5}$$

Thus, we don't have to compute the feature vector $\phi(\mathbf{x})$.

5. Generalized Lagrangian Function. Consider the optimization problem

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) \quad \text{s.t.} \quad g_j(\boldsymbol{w}) \leq 0, \forall j = 1, \ldots, m, \; h_j(\boldsymbol{w}) = 0, \forall j = 1, \ldots, p \tag{6}$$

Show that for the generalized Lagrangian function, defined by

$$L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \triangleq f(\boldsymbol{w}) + \sum_{j=1}^{n} \alpha_j g_j(\boldsymbol{w}) + \sum_{j=1}^{p} \beta_j h_j(\boldsymbol{w})$$

the following always holds

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \ \leq \ \min_{\boldsymbol{w}} \max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Solution: We can prove the result using the following steps:

$$\begin{aligned}
&\min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \ \ \forall \boldsymbol{w}, \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta} \\
\Longrightarrow &\max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq \max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \ \ \forall \boldsymbol{w} \\
\Longrightarrow &\max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq \min_{\boldsymbol{w}} \max_{\boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})
\end{aligned}$$

6. Dual Optimization. Consider the optimization program

$$\min_{\boldsymbol{x}} \frac{1}{2} \boldsymbol{x}^\top P_0 \boldsymbol{x} + q_0^\top \boldsymbol{x} + r_0 \ \ \text{s.t.} \ \ \frac{1}{2} \boldsymbol{x}^\top P_i \boldsymbol{x} + q_i^\top \boldsymbol{x} + r_i \leq 0, \ i = 1, \ldots, m$$

where P_0 and all P_i are assumed to be positive semi-definite matrices. A) Form the generalized Lagrangian function. B) Compute the Lagrange dual function. C) Derive the dual maximization problem.

Solution of A: The generalized Lagrangian function is given by:

$$L(x, \alpha) = \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 + \sum_{i=1}^{m} \alpha_i (\frac{1}{2} x^\top P_i x + q_i^\top x + r_i)$$

Solution of B: The Lagrange dual function is shown below:

$$\begin{aligned}
\theta_D(x) &= \min_{x} L(x, \alpha) \\
&= \min_{x} \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 + \sum_{i=1}^{m} \alpha_i (\frac{1}{2} x^\top P_i x + q_i^\top x + r_i)
\end{aligned}$$

To get the minimum, we set

$$\begin{aligned}
\left. \frac{\partial L(x, \alpha)}{\partial x} \right|_{x^*} &= 0 \\
P_0 x^* + q_0 + \sum_{i=1}^{m} \alpha_i (P_i x^* + q_i) &= 0 \\
(P_0 + \sum_{i=1}^{m} \alpha_i P_i) x^* &= -(q_0 + \sum_{i=1}^{m} \alpha_i q_i)
\end{aligned}$$

Let $P(\alpha) \triangleq P_0 + \sum_{i=1}^{m} \alpha_i P_i,\ q(\alpha) \triangleq q_0 + \sum_{i=1}^{m} \alpha_i q_i,\ r(\alpha) = r_0 + \sum_{i=1}^{m} \alpha_i r_i$. Then, $x^* = -P(\alpha)^{-1} q(\alpha)$. Pluggin this back into the Lagrangian function, we get

$$q(\alpha) = -1/2 q(\alpha)^\top P(\alpha)^{-1} q(\alpha) + r(\alpha).$$

Solution of C: The dual optimization is given by

$$\max_{\alpha} -1/2 q(\alpha)^\top P(\alpha)^{-1} q(\alpha) + r(\alpha) \ \ \text{s.t.} \ \ \alpha \geq 0.$$

7. Logistic Regression Implementation.
A) Write down a code in Python whose input is a training dataset $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^N, y^N)\}$ and its output is the weight vector $\boldsymbol{w}$ in the logistic regression model $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x})$.
B) Download the dataset on the course webpage. Use 'dataset1'. Run the code on the training dataset to compute $\boldsymbol{w}$ and evaluate on the test dataset. Report $\boldsymbol{w}$, classification error on the training set and classification error on the test set. Plot the data (use different colors for data in different classes) and plot the decision boundary found by the logistic regressions.
C) Repeat part B using 'dataset2'. Explain the differences in results between part A and B and justify your observations/results.

8. SVM Implementation.
Implement SVM with the SMO algorithm and train it on the provided dataset. For your implementation, you only have to use the linear kernel. In addition, run SVM using the LIBSVM package and compare the results. You can implement the simplified SMO, as described in http://cs229.stanford.edu/materials/smo.pdf
A) Apply the SVM on the 'dataset1' and report the classification error (on both training and test sets) as a function of the regularization parameter C.
B) Repeat part A using 'dataset2'. Explain the differences in results between part A and B and justify your observations/results.

Homework Submission Instructions:
– Submission of Written Part: You must submit your written report in the class BEFORE CLASS STARTS. The written part, must contain the plots and results of running your code on the provided datasets.
–Submission of Code: You must submit your Python code (.py file) via email, BEFORE CLASS STARTS. For submitting your code, please send an email to me and CC both TAs.
- The title of your email must be "CS6140: Code: HW2: Your First and Last Name".
- You must attach a single zip file to your email that contains all python codes and a readme file on how to run your files.
- The name of the zip file must be "HW2-Code: Your First and Last Name".