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: $P(Y = 1|X, w_1, w_2) = \sigma(w_1 X_1 + w_2 X_2)$
Model 2: $P(Y = 1|X, w_0, w_1, w_2) = \sigma(w_0 + w_1 X_1 + w_2 X_2)$

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.)

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.

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.

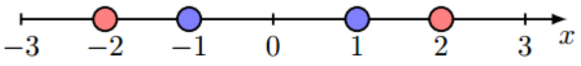


Figure 1: Red points represent instances from class +1 and blue points represent instances from class -1.

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.

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.

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

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 $c_1, c_2 \geq 0$.

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

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

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

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

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

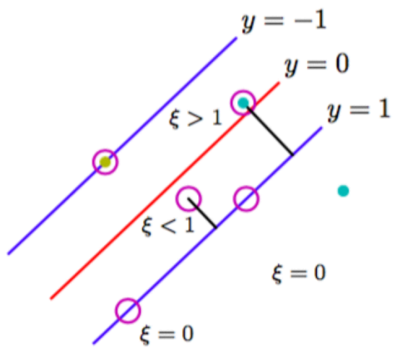
$$\begin{array}{lll}
\underset{\mathbf{w}, \xi_i}{\text{minimize}} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} \xi_i & \\
\text{subject to} & y_i(\mathbf{w}^T \phi(x_i)) \geq 1 - \xi_i & \forall i = 1, \cdots, n \\
 & \xi_i \geq 0 & \forall i = 1, \cdot, n
\end{array}$$

which is equivalent to the following dual optimization

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

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 $0 < \xi_i \le 1$ 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 **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.

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 $C \to \infty$.

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 **x**, , how to make prediction on the instance without explicitly computing the feature vector $\phi(\mathbf{x})$?

5. Generalized Lagrangian Function. Consider the optimization problem

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

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

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

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.

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”.