SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018
Homework 3

Due 2 March 2018, 5pm

This homework will be graded by Thilini Cooray

Question 1. Download and install the widely used SVM implementation LIBSVM (`https://github.com/cjlin1/libsvm`, or `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`; clicking on either link takes you to the webpage). We expect you to install the package on your own – this is part of learning how to use off-the-shelf machine learning software. Read the documentation to understand how to use it.

Download `promoters.zip`. In that folder are `training.txt` and `test.txt`, which respectively contain 74 training examples and 32 test examples in LIBSVM format. The goal is to predict whether a certain DNA sequence is a promoter[1] or not based on 57 attributes about the sequence (this is a binary classification task).

Run LIBSVM to classify promoters with different kernels (0-3), using default values for all other parameters. What is your test accuracy for each kernel choice? *(5 points)*

Question 2.

(a) In logistic regression, we find parameters of a logistic (sigmoid) function that maximize the likelihood of a set of training examples $((x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)}))$. The likelihood is given by

$$\prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}) \tag{1}$$

However, we re-express the problem of maximizing the likelihood as minimizing the following expression:

$$\frac{1}{n}\sum_{i=1}^{n} \log\big(1 + \exp\big(-y^{(i)}(\theta \cdot x^{(i)} + \theta_0)\big)\big). \tag{2}$$

(Note that both maximization and minimization problems have the same optimal θ and θ_0.) What *computational* advantage does Equation 2 have over Equation 1? *(Hint: try randomly*

[1] A promoter is a region of DNA that facilitates the transcription of a particular gene. The ability to predict promoters is of practical importance in searching for new promoter sequences.

generating, say, 1,000 probabilities in Python and multiplying them together as in Equation 1.) (5 points)

(b) You are given a training set `diabetes_train.csv`. Each row in the file contains whether a patient has diabetes (+1: yes, -1: no), followed by values of 20 unknown features. **Write code to train a logistic regression model with stochastic gradient descent (SGD)**. Run SGD for 10,000 iterations, and save the model weights after every 100 iterations. Plot the log-likelihood of the training data given by your model at every 100 iterations. (Log-likelihood is $\log \prod_{i=1}^{n} P(y^{(i)}|x^{(i)}) = \sum_{i=1}^{n} \log P(y^{(i)}|x^{(i)})$ where $(x^{(i)}, y^{(i)})$ is an example.) Provide crystal clear instructions along with the source code on how to execute it. (*Hints: If your stochastic gradient descent code in the previous homework is written modularly enough, you could save time by reusing it here. Try a learning rate of 0.1*) *(10 points)*

Question 3. We can compute the *leave-one-out-cross-validation* (LOOCV) error of a binary classifier as follows. Given a training set of n examples, hold out (i.e., remove) one example i from the set, train a binary classifier on the remaining $n-1$ examples, and then test the classifier on example i. Repeat the procedure for every example. The LOOCV error is the fraction of incorrectly classified held-out examples. For example, we are given a training set $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $(x^{(3)}, y^{(3)})$. We first train a binary classifier on $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$ and test it on $x^{(3)}$ (and it gets the label correct). Next we train the classifier on $(x^{(1)}, y^{(1)})$, $(x^{(3)}, y^{(3)})$ and test it on $x^{(2)}$ (and it again gets the label correct). Finally we train the classifier on $(x^{(2)}, y^{(2)})$, $(x^{(3)}, y^{(3)})$ and test it on $x^{(1)}$ (and it gets the label wrong). Because the classifier wrongly predicted one of the three examples, the LOOCV error is $\frac{1}{3}$.

Give an upper bound on the LOOCV error of the SVM in Figure 1, which was trained on 13 examples (i.e., LOOCV error $\leq$?). Explain your answer. *(Hint: Recall that only the support vectors are used to define the SVM's hyperplane.) (5 points)*

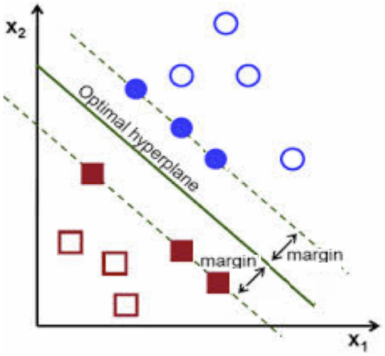

Figure 1: The bold line is the decision boundary of an SVM, and the dotted lines are its margin boundaries. The circle are positive examples, and the squares are negative examples. Filled circles and squares are support vectors.

(The LOOCV tells us something about how well the model generalizes to unseen data. From this question we can understand how the sparse solution as given by SVM may help generalization.)

Question 4. You have trained a simple linear SVM from a large collection of data points, where the input x vectors are from a 3-d space, of the form (x_1, x_2, x_3). Now you would like to explore the trained model a little further.

(a) You found the margin boundaries are $3x_1+12x_2+4x_3+1=0$ and $3x_1+12x_2+4x_3+3=0$. What is the decision boundary? What is the size of the margin? (*Hint: Recall that the size of the margin is the distance between the decision boundary and the margin boundary, and* $3^2+4^2+12^2=13^2$) *(4 points)*

(b) Next, in the same training dataset, you found the following points (of the form $\big((x_1^{(t)}, x_2^{(t)}, x_3^{(t)}), y^{(t)}\big)$): $\big(($-1, 1, -2$)$, +1$\big)$, $\big(($0, 1, -4$)$, +1$\big)$, $\big(($-1, 1, -3$)$, -1$\big)$, $\big(($0, 0, 0$)$, -1$\big)$. Is this dataset linearly separable? Clearly explain why. *(2 points)*

(c) You checked the values for the Lagrangian multipliers α's used in the dual form for the above points as returned by the optimizer, and found that the value of the first data point $\alpha_1 = 1.6$. What are the exact values of θ and θ_0, respectively? Which of the above 4 points are support vectors? (*Hint: Also think about: what if* $\alpha_1 = 0$*? Would this make any difference to your answer? – you don't have to submit an answer for this case though.*) *(9 points)*

$3(-1)+12(1)+4(-2)+1 \neq 0$ but 2

$3(-1)+12(1)+4(-2)+3 \neq 0$ but 4

Question 2.

(a) In logistic regression, we find parameters of a logistic (sigmoid) function that maximize the likelihood of a set of training examples $((x^{(1)}, y^{(1)}), ..., (x^{(n)}, y^{(n)}))$. The likelihood is given by

$$\prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}) \tag{1}$$

However, we re-express the problem of maximizing the likelihood as minimizing the following expression:

$$\frac{1}{n}\sum_{i=1}^{n} \log\big(1 + \exp\big(-y^{(i)}(\theta \cdot x^{(i)} + \theta_0)\big)\big). \tag{2}$$

(Note that both maximization and minimization problems have the same optimal θ and θ_0.) What *computational* advantage does Equation 2 have over Equation 1? *(Hint: try randomly generating, say, 1,000 probabilities in Python and multiplying them together as in Equation 1.) (5 points)*

2 (a) In equation 2, the use of the log function would lead to the gradient of a single example being minimised before being used for the weight update. Finding the gradient of a single example would not require the use of the chain rule. Instead equation 2 would require the use of a summation function.

In equation 1, the product notation leads to the use of chain rule in finding the gradient which is computationally intensive than the method used for equation 2. Eliminating the need for chain rule is very computationally advantageous which is the aim for equation 2.

Question 3. We can compute the *leave-one-out-cross-validation* (LOOCV) error of a binary classifier as follows. Given a training set of n examples, hold out (i.e., remove) one example i from the set, train a binary classifier on the remaining $n-1$ examples, and then test the classifier on example i. Repeat the procedure for every example. The LOOCV error is the fraction of incorrectly classified held-out examples. For example, we are given a training set $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)})$. We first train a binary classifier on $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)})$ and test it on $x^{(3)}$ (and it gets the label correct). Next we train the classifier on $(x^{(1)}, y^{(1)}), (x^{(3)}, y^{(3)})$ and test it on $x^{(2)}$ (and it again gets the label correct). Finally we train the classifier on $(x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)})$ and test it on $x^{(1)}$ (and it gets the label wrong). Because the classifier wrongly predicted one of the three examples, the LOOCV error is $\frac{1}{3}$.

Give an upper bound on the LOOCV error of the SVM in Figure 1, which was trained on 13 examples (i.e., LOOCV error $\leq$?). Explain your answer. *(Hint: Recall that only the support vectors are used to define the SVM's hyperplane.) (5 points)*

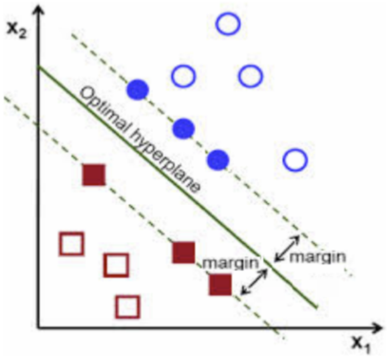

Figure 1: The bold line is the decision boundary of an SVM, and the dotted lines are its margin boundaries. The circle are positive examples, and the squares are negative examples. Filled circles and squares are support vectors.

(The LOOCV tells us something about how well the model generalizes to unseen data. From this question we can understand how the sparse solution as given by SVM may help generalization.)

3 LOOCV $\leq$ # of support vectors / no of examples

LOOCV $\leq$ 6 / 13.

First, to understand the answer, there is a need to know the term support vectors. The examples that lie exactly on the margin boundaries are called support vectors. Next, the 7 examples that lie outside the margin boundary would be classified correctly regardless of whether they are part of the training set. Not so for support vectors that are key to define the linear separator and cause error in classification.

Question 4. You have trained a simple linear SVM from a large collection of data points, where the input x vectors are from a 3-d space, of the form (x_1, x_2, x_3). Now you would like to explore the trained model a little further.

(a) You found the margin boundaries are $3x_1+12x_2+4x_3+1=0$ and $3x_1+12x_2+4x_3+3=0$. What is the decision boundary? What is the size of the margin? (*Hint: Recall that the size of the margin is the distance between the decision boundary and the margin boundary, and* $3^2+4^2+12^2=13^2$) *(4 points)*

(b) Next, in the same training dataset, you found the following points (of the form $\big((x_1^{(t)}, x_2^{(t)}, x_3^{(t)}), y^{(t)}\big)$): $\big($(-1, 1, -2), +1$\big)$, $\big($(0, 1, -4), +1$\big)$, $\big($(-1, 1, -3), -1$\big)$, $\big($(0, 0, 0), -1$\big)$. Is this dataset linearly separable? Clearly explain why. *(2 points)*

(c) You checked the values for the Lagrangian multipliers α's used in the dual form for the above points as returned by the optimizer, and found that the value of the first data point $\alpha_1 = 1.6$. What are the exact values of θ and θ_0, respectively? Which of the above 4 points are support vectors? (*Hint: Also think about: what if $\alpha_1 = 0$? Would this make any difference to your answer? – you don't have to submit an answer for this case though.*) *(9 points)*

4(a) When $x_1=x_2=0$ in the plane $3x_1+12x_2+4x_3+1=0$, there exists the point $(0,0,-\frac{1}{4})$, We can find distance between point and a plane via eqn

$$D=\frac{|ax_1+bx_2+cx_3+d|}{\sqrt{a^2+b^2+c^2}}$$

$$=\frac{|3(0)+12(0)+4(-\frac{1}{4})+3|}{\sqrt{3^2+12^2+4^2}}$$

$$=\frac{|-1+3|}{\sqrt{169}}$$

$$=\frac{2}{13}$$

→ Size of the margin $=\frac{1}{2}\left(\frac{2}{13}\right)=\frac{1}{13}$

Decision Boundary would be $3x_1+12x_2+4x_3+2=0$

4(b) $+1: 3(-1)+12(1)+4(-2)+2=3$

$+1: 3(0)+12(1)+4(-4)+2=-1$ (Wrongly Classified)

$-1: 3(-1)+12(1)+4(-3)+2=-1$

$-1: 3(0)+12(0)+4(0)+2=2$ (Wrongly Classified)

→ Hence, Dataset is not linearly separable.

4(c) $\alpha_1=1.6>0$ → $(-1,1,-2)$ is a support vector.

→ $\theta=[3,12,4]$ and $\theta_0=2$

Since point 1 is definitely a support vector but does not lie on margin boundaries, Soft vector machine is used here. As a result, points 2, 3 and 4 cannot be determined as support vectors. Hence there is no answer for points 2, 3 and 4.