SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018
Homework 3

Due 2 March 2018, 5pm

Sample Solutions

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)*

$$\begin{aligned}
\text{Kernel 0 (linear) accuracy} &= 27/32 = 84\% \\
\text{Kernel 1 (polynomial) accuracy} &= 26/32 = 81\% \\
\text{Kernel 2 (RBF) accuracy} &= 29/32 = 91\% \\
\text{Kernel 3 (Sigmoid) accuracy} &= 14/32 = 44\%
\end{aligned}$$

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

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

[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.

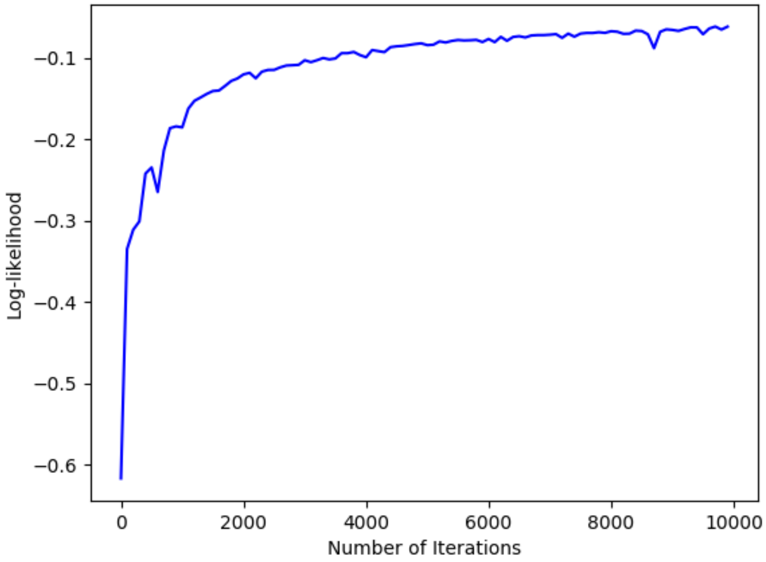
However, we re-express the problem of maximizing the likelihood as minimizing the following expression:

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

(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)*

Progressively multiplying many probabilities together as in Equation 1 quickly gives a result that is too small to be representable in computer memory (this is known as an *underflow* problem). In contrast, Equation 2 uses a sum over terms that makes this problem less likely to occur.

(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. (If your 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)*

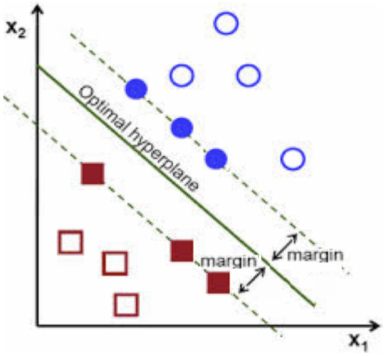


Figure 1: The bold line is the decision boundary of an SVM, and the dotted lines are its margin boundaries. The circle are positive examples, and the squares are negative examples. Filled circles and squares are support vectors.

(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.)

If we hold out a training example that is not a support vector, the SVM's hyperplane does not change. The example was classified correctly when included, and will continue to be classified correctly when excluded from the training set (because the hyperplane does not change). An error **may** occur if we hold out a support vector. Thus the (conservative) bound is:
LOOCV error $\leq \frac{6}{13}$ (six support vectors out of thirteen examples).

Question 4. You have trained a simple linear SVM from a large collection of data, 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)*

The two margin boundaries are as follows:

$$3x_1 + 12x_2 + 4x_3 + 1 = 0 \tag{3}$$
$$3x_1 + 12x_2 + 4x_3 + 3 = 0 \tag{4}$$

The decision boundary is exactly in the middle of the above two parallel hyperplanes. That means the decision boundary is:

$$\frac{(3x_1 + 12x_2 + 4x_3 + 1) + (3x_1 + 12x_2 + 4x_3 + 3)}{2} = 0 \tag{5}$$
$$\tag{6}$$

which gives you:

$$3x_1 + 12x_2 + 4x_3 + 2 = 0 \tag{7}$$

The margin is calculated as the distance between the decision boundary and the margin boundary. It can be calculated using 2 data points – one on the margin boundary and one on the decision boundary.

For example, we can take the first point $\vec{p} = [1, 0, -1]$ that is on the margin boundary $3x_1 + 12x_2 + 4x_3 + 3 = 0$, and the second point $\vec{q} = [2, -1, 1]$ that is from the decision boundary.

The size of the margin is going to be the absolute value of the projection of $(\vec{q} - \vec{p})$ along the direction of the normal vector of the decision boundary. The normal vector is $\vec{n} = [3, 12, 4]$.

$$\text{margin} = \frac{|(\vec{p} - \vec{q}) \cdot \vec{n}|}{||\vec{n}||} = \frac{1}{\sqrt{3^2 + 12^2 + 4^2}} = \frac{|[1, -1, 2] \cdot [3, 12, 4]|}{\sqrt{3^2 + 12^2 + 4^2}} = \frac{1}{13}$$

(*Note that at this point, we are not able to know the exact values for θ and θ_0 yet, why?*)

(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)*

A dataset is linearly separable if no points of different sign appear on the same side of the decision boundary.

Let us look at the following two points: $(0, 1, -2)$ and $(0, 1, -4)$:

Following table checks whether above condition is satisfied by all data points

$\vec{x}$	$3x_1 + 12x_2 + 4x_3 + 2$	y
$(-1, 1, -2)$	$+3$	$+1$
$(0, 1, -4)$	-2	$+1$

This shows these two points, though they are both labeled as positive points, appear on different sides of the decision boundary. This means the dataset must not be linearly separable.

(*Some of you may ask: is it possible to have a case where the dataset is linearly separable but SVM still returns you the above solution because of a particular choice of the C or $1/\lambda$ coefficient? Think about it!*)

(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)*

The fact that $\alpha_1 \neq 0$ shows that this point is a support vector. From here we can conclude that this point must satisfy the following condition:

$$y(\theta \cdot \vec{x} + \theta_0) \leq 1$$

On the other hand, we can rewrite the two margin boundaries and the decision boundary as:

$$3x_1 + 12x_2 + 4x_3 + 2 = +1 \tag{8}$$

$$3x_1 + 12x_2 + 4x_3 + 2 = 0 \tag{9}$$

$$3x_1 + 12x_2 + 4x_3 + 2 = -1 \tag{10}$$

Note that from here we still do not know the values for θ and θ_0 as we may have the following two options:

$$\theta = [3, 12, 4], \theta_0 = 2 \tag{11}$$

$$\theta = [-3, -12, -4], \theta_0 = -2 \tag{12}$$

However since the first point satisfy $y(\theta \cdot x + \theta_0) \leq 1$, we can conclude that the second option above for θ and θ_0 is correct, since for the first option we have:

$$(+1)\big((3) \times (-1) + (12) \times (1) + (4) \times (-2) + 2\big) = 3 \not\leq 1$$

but for the second option we have:

$$(+1)\big((-3) \times (-1) + (-12) \times (1) + (-4) \times (-2) - 2\big) = -3 \leq 1$$

Thus now we can conclude

$$\theta = [-3, -12, -4], \theta_0 = -2 \tag{13}$$

Now we can decide support vectors(SV) based on above model parameters:

$\vec{x}, y$	$\theta \cdot x + \theta_0$	$y(\theta \cdot x + \theta_0)$	is SV?
$(-1, 1, -2), +1$	-3	-3 (<1)	yes
$(0, 1, -4), +1$	$+2$	$+2$ (>1)	no
$(-1, 1, -3), -1$	$+1$	-1(<1)	yes
$(0, 0, 0), -1$	-2	$+2$ (>1)	no

Therefore $\big((-1, 1, -2), +1\big)$ and $\big((-1, 1, -3), -1\big)$ are support vectors.

(*Can you figure out the solution for the case where $\alpha_1 = 0$? In the case the answer is different. Think about it!*)