Name:	**Student ID:**

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018
Midterm Exam
Sample Solutions
Date: 16 March 2018
Time: 14:30 - 16:30

Instructions:

1. This is a closed-book exam.
2. Write your name and student ID at the top of this page.
3. This paper consists of 7 main questions and 11 printed pages.
4. The problems are not necessarily in order of difficulty. We recommend that you scan through all the questions first, and then decide on the order to answer them.
5. Write your answers in the space provided.
6. You are allowed to use non-programmable calculators.
7. You may NOT refer to any other material.
8. You may NOT access the Internet.
9. You may NOT communicate via any means with anyone (aside from the invigilators).

For staff's use:

Problem 1	/8
Problem 2	/10
Problem 3	/5
Problem 4	/6
Problem 5	/12
Problem 6	/6
Problem 7	/3
Total	**/50**

Problem 1: True or False (8 Points)

Please indicate whether the following statements are true (**T**) or false (**F**).

1. The perceptron algorithm will converge after a finite number of steps if the underlying training set is linearly separable. *(1 point)*

   Answer : *T*

2. Logistic Regression is a model that is used for classification but not regression. *(1 point)*

   Answer : *T*

3. We can use the backpropagation algorithm to learn the parameters of a multi-layer feed-forward neural network. *(1 point)*

   Answer : *T*

4. The initialization is important for the k-means algorithm. It is possible that if you change your initialization, you may arrive at very different solutions after running the k-means algorithm. *(1 point)*

   Answer : *T*

5. The stochastic gradient descent algorithm should only be used to minimize a function that is convex. If the function is not convex, then we should never use it since it will not let us reach the global optimal value. *(1 point)*

   Answer : *F*

6. Consider a training set consisting of input-output pairs $(x^{(i)}, y^{(i)})$ for binary classification. The solution returned by a linear SVM is of the form $\theta \cdot x + \theta_0 = 0$ where θ is a linear combination of the $x^{(i)}$'s (i.e. the inputs of the training instances). *(1 point)*

   Answer : *T*

7. To perform non-linear classification, we can use SVM with kernels. Typically we can play the kernel trick based on the dual form, replacing the dot product $x^{(i)} \cdot x^{(j)}$ with a predefined kernel function $K(x^{(i)}, x^{(j)})$ in the dual form's objective function. *(1 point)*

   Answer : *T*

8. The Naive Bayes model is a generative model, where strong conditional independence assumptions are made. Specifically, it assumes two words (say words a and b) in a document are conditionally independent of each other given the document's label (y). *(1 point)*

   Answer : *T*

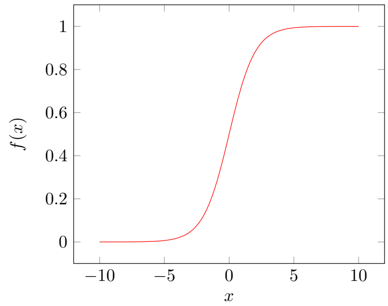
Problem 2: Multiple Choice Questions (10 Points)

Please answer the questions by ticking (✓) the correct choices.

1. Which of the following statements are true (select all that apply)? *(2 points)*

   - (✓) In classification, we are learning a mapping between two spaces – the input space is a vector space, and the output space is a set of discrete variables.
   - (✓) The k-means algorithm is an unsupervised learning algorithm that maps each input vector to a discrete variable which is the cluster ID/label.
   - ( ) Semi-supervised learning algorithms typically make use of a small amount of unlabeled data and a very large amount of labeled data for learning.
   - (✓) We are able to use the RBF kernel to map original input vectors to new vectors that reside in an infinite dimensional space.

2. This plot can be associated with which of the following functions (select all that apply)? *(2 points)* (*Hint: check what happens when $x \to -\infty$ or $x \to +\infty$.*)



   - ( ) $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
   - (✓) $f(x) = \frac{1}{1+e^{-x}}$
   - (✓) $f(x) = \frac{e^x}{1+e^x}$
   - ( ) $f(x) = \log(1 + e^x)$

3. In class we mentioned that roughly speaking a convex function typically has a U-shape (there are boundary cases where the surface can be flat, though. In that case the function is both convex and concave). In the homework we also learned that a convex function has the property that $\frac{f(x_1)+f(x_2)}{2} \geq f(\frac{x_1+x_2}{2})$. A convex function has the nice property that its local optimum is also a global optimum. Which of the following functions are convex in $(-\infty, +\infty)$ (select all that apply)? (*Hint: as we have learned in the homework, the sum/max of two convex functions is again convex*) *(2 points)*

   - (✓) $f(x) = (x - 1)^2$
   - (✓) $f(x) = \max(x^2, 2^x)$
   - (✓) $f(x) = |x| - x$
   - (✓) $f(x) = |x| + x$

4. Let us consider a Naive Bayes model for document classification. Assume we have two documents: $\Big((a,b,c),+1\Big)$ and $\Big((c,e),-1\Big)$ in the training set. This means in the first document we have 3 words "a,b,c" and the label of the document is $+1$, while in the second document we have 2 words "c,e" and its label is -1. Based on what we discussed in class, we can use the maximum likelihood estimator to find the most probable parameters based on such a training set. Which of the following estimated model parameters are correct: (select all that apply)? (*Hint: we follow the notation used in class and in the notes here, i.e.,* $\theta_w^+ = p(w|y=+1)$*, and* $\theta_w^- = p(w|y=-1)$) *(2 points)*

   - (✓) $\theta_a^+ = 1/3$
   - (✓) $\theta_c^+ = 1/3$
   - ( ) $\theta_a^- = 1/2$
   - (✓) $\theta_c^- = 1/2$

5. Support Vector Machines optimize the following objective function:

$$\min \ \frac{\lambda}{2}||\theta||^2 + \sum_{t=1}^{n} \xi_t$$

   subject to $y^{(t)}(\theta \cdot x^{(t)} + \theta_0) \geq 1 - \xi_t$, $\xi_t \geq 0$, $t = 1, \ldots, n$.

   We can alternatively perform the optimization with its dual form:

$$\max \ \sum_{t=1}^{n} \alpha_t - \frac{1}{2}\sum_{t=1}^{n}\sum_{t'=1}^{n} \alpha_t \alpha_{t'} y^{(t)} y^{(t')} (x^{(t)} \cdot x^{(t')})$$

   subject to:

$$0 \leq \alpha_t \leq 1/\lambda, \ \ \sum_{t=1}^{n} \alpha_t y^{(t)} = 0$$

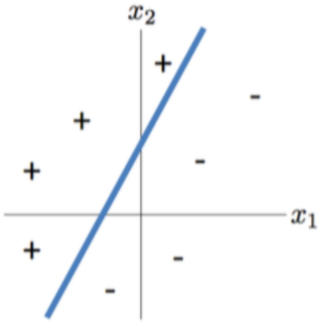
   where α_t is the Lagrangian multiplier associated with the t-th instance $(x^{(t)}, y^{(t)})$.

   After a Support Vector Machine has been trained under the dual formulation, we obtain $\hat{\alpha}_t$ for each t. Which of the following conditions are definitely true (select all that apply)? (*Hint: try to recall in class how we used Lagrangian to draw the conclusions.*) *(2 points)*

   - (✓) $\hat{\alpha}_t = 0 \Rightarrow y^{(t)} \left(\sum_{t'=1}^{n} \hat{\alpha}_{t'} y^{(t')} (x^{(t')} \cdot x^{(t)}) + \hat{\theta}_0 \right) \geq 1$
   - (✓) $\hat{\alpha}_t \in (0, 1/\lambda) \Rightarrow y^{(t)} \left(\sum_{t'=1}^{n} \hat{\alpha}_{t'} y^{(t')} (x^{(t')} \cdot x^{(t)}) + \hat{\theta}_0 \right) = 1$
   - ( ) $\hat{\alpha}_t = 1/\lambda \ \Rightarrow \ y^{(t)} \left(\sum_{t'=1}^{n} \hat{\alpha}_{t'} y^{(t')} (x^{(t')} \cdot x^{(t)}) + \hat{\theta}_0 \right) > 1$
   - ( ) $\hat{\alpha}_t = 1/\lambda \ \Rightarrow \ y^{(t)} \left(\sum_{t'=1}^{n} \hat{\alpha}_{t'} y^{(t')} (x^{(t')} \cdot x^{(t)}) + \hat{\theta}_0 \right) = 1$

Problem 3: Perceptron (5 Points)

1. You are in the middle of running the perceptron algorithm. The current decision boundary as returned by the perceptron in the figure below crosses the x_1 axis at -2 and the x_2 axis at 6. What are the possible weights θ and θ_0 of the perceptron? (providing one possible solution is sufficient) *(2 points)*



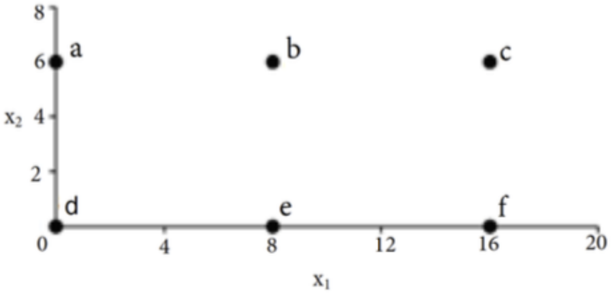
$x_2 = \frac{6}{2}x_1 + 6 \Rightarrow -3x_1 + x_2 - 6 = 0$
Thus, $\theta = [-3, 1], \theta_0 = -6$. Any proportional scaling of the weights are accepted.

2. You continued running the perceptron algorithm for a few more steps and you checked the decision boundary again. Do you think there is a possibility that the decision boundary may be different by now? Clearly explain why. (*Hint: do we know the normal vector of the previous decision boundary?*)*(2 points)*

"No" if the normal vector is defined as above. (But "Yes" if the normal vector is defined as $[3, -1]$, this could happen when one randomly initialize θ and θ_0 for the perceptron.)

Problem 4: k-Means (6 Points)

In the figure below is a set of 6 points: $a = (0, 6), b = (8, 6), c = (16, 6), d = (0, 0), e = (8, 0), f = (16, 0)$. Let us run the k-means algorithm over these points with $k = 3$ and with the squared Euclidean distance measure.



Given an initialization of the 3 initial representatives of the 3 clusters z_1, z_2, z_3 such that the algorithm will definitely converge to the following clusters: $\{(a, d), (b, e), (c, f)\}$. *(3 points)*

$z_1 = ($ 0 , 3 $)$

$z_2 = ($ 8 , 3 $)$

$z_3 = ($ 16 , 3 $)$

Other solutions exist.

Give another initialization such that the algorithm will definitely not converge to the following clusters: $\{(a, d), (b, e), (c, f)\}$. *(3 points)*

$z_1 = ($ 4 , 6 $)$

$z_2 = ($ 4 , 0 $)$

$z_3 = ($ 16 , 3 $)$

Other solutions exist.

Problem 5: Kernels (12 Points)

Recall in support vector machines (SVMs), a kernel function $K(x, x')$ implicitly does the following two tasks at the same time:

- Map the two input vectors $x, x' \in R^n$ into two new vectors $z, z' \in R^m$ respectively with a certain function $f : R^n \to R^m$.
- Return the dot product between z and z', i.e., $z \cdot z'$.

In other words, if a function $K(x, x') : R^n \times R^n \to R$ is a valid kernel, then there exists a function $f : R^n \to R^m$ such that $K(x, x') = f(x) \cdot f(x')$.

Indicate whether the following functions are valid kernels, and explain your answers (i.e. provide a formal proof to justify your answer).

Hint: how to prove if a function is not a kernel? As discussed in class, you may try to check if $K(x, x)$ is not always non-negative for any x (why?). Or you may check if $K(x, x') = K(x', x)$ is not always true for any choice of x and x' (why?).

(a) $K(x, x') = 10$. *(2 points)*

Is this a kernel? yes (*yes* or *no*).
Explanations:

$f(x) = \sqrt{10}$

(b) $K(x, x') = -1$. *(2 points)*

Is this a kernel? no (*yes* or *no*).
Explanations:

Assume it is valid and $K(x, x') = f(x) \cdot f(x')$ for some f. Then we have $-1 = K(x, x) = ||f(x)||^2 \geq 0$ for any x. Contradiction!

(c) $K(x, x') = x \cdot x' + 1$ *(2 points)*

Is this a kernel? yes (*yes* or *no*).
Explanations:

$x \cdot x'$ is a kernel with $f(x) = x$. 1 is also a kernel. The sum of two kernels is a kernel.

(d) $K(x, x') = x \cdot x' - 1$ *(3 points)*

Is this a kernel? no (*yes* or *no*).
Explanations:

Assume it is a kernel, then $K(x, x) \geq 0$. However for certain choices of x this cannot be satisfied. Specifically if $||x|| < 1$. Thus we have a contradiction.

(e) $K(x, x') = ||x + 2x'||^2$ *(3 points)*

Is this a kernel? no (*yes* or *no*).
Explanations:

Assume it is a kernel. Then $K(x, x') = K(x', x)$ is always true for any x and x'. This means we must have $||x + 2x'||^2 = ||x' + 2x||^2$ for any x and x'. However this is not always true. For example, choosing $x' = 2x \neq 0$, we have $||5x||^2 = ||4x||^2$ which is impossible. Thus we have a contradiction.

Problem 6: SVM (6 Points)

John and Mary are both working on their own SVM implementation as part of their homework for the machine learning class. Both of them used the following objective function as the primal form:

$$\min_{\theta,\theta_0} \lambda||\theta||^2 + \sum_{t=1}^{n} \max\left(1 - y^{(t)}(x^{(t)} \cdot \theta + \theta_0), 0\right)$$

John and Mary then respectively used the same online optimization tools to help them find the exact optimal solutions to θ and θ_0. Both of them trained on the same dataset which consists of a large collection of instances for binary classification and both set $\lambda = 1$.

1. After he finished the training process, unfortunately, John found his implementation has a bug. Specifically John incorrectly flipped the signs for all y's in the dataset when importing the data (the x's were imported correctly). In other words, in his implementation, all positive instances in the training set were labeled as negative instances and all negative instances were labeled as positive instances. John would like to fix this bug. However Mary told John that he did not have to fix the bug and re-train the model – he just needed to make some simple changes to the learned θ and θ_0 (based on his buggy impelmentation) to obtain the desired decision boundary. What should John do here? Clearly explain why. (*Hint: one idea is to think about the objective of the dual form of SVM, and think about how θ/θ_0 are defined with respect to the training instances.*) *(3 points)*

Flipping the sign leads to the optimization of the following objective instead:

$$\min_{\theta,\theta_0} \lambda||\theta||^2 + \sum_{t=1}^{n} \max\left(1 - (-y^{(t)})(x^{(t)} \cdot \theta + \theta_0), 0\right)$$

which is

$$\min_{\theta,\theta_0} \lambda||\theta||^2 + \sum_{t=1}^{n} \max\left(1 + y^{(t)}(x^{(t)} \cdot \theta + \theta_0), 0\right)$$

or:

$$\min_{-\theta,-\theta_0} \lambda||-\theta||^2 + \sum_{t=1}^{n} \max\left(1 - y^{(t)}(x^{(t)} \cdot (-\theta) + (-\theta_0)), 0\right)$$

This means $-\theta$ and $-\theta_0$ are what we wanted. From here we can see that we can flip the sign of θ and θ_0 returned by the buggy implementation.

Other explanations exist. However you are supposed to clearly explain why flipping the sign makes sense.

2. Unfortunately, Mary also found a bug in her implementation – she realized in her implementation she actually used the following objective function (instead of the one listed above):

$$\min_{\theta,\theta_0} \lambda||\theta||^2 + \sum_{t=1}^{n} \max\left(1 + y^{(t)}(x^{(t)} \cdot \theta + \theta_0), 0\right)$$

In other words, she used the + sign instead of the - sign in front of the term $y^{(t)}(x^{(t)} \cdot \theta + \theta_0)$ in the objective function. Mary wanted to fix this bug and re-run the model as well. However, this time John told Mary that she also does not have to do so – she also just needed to make some simple changes to the learned θ and θ_0 (based on her buggy implementation) to obtain the desired decision boundary. What should Mary do here? Clearly explain why. (*Hint: one idea is to think about the relation between this question and the previous question.*) *(3 points)*

This is equivalent to the previous question. See the solution to the previous question.

Problem 7: LOOCV (3 Points)

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 wrongly 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 1/3.

Assume you are running a learning experiment on a new algorithm for binary classification. You have a dataset consisting of 50 positive and 50 negative examples, each with k discrete input features. You plan to use leave-one-out cross validation. As a baseline, you decide to compare your algorithm to a simple *majority classifier*, i.e., one which predicts the class that is most common in the training data (choosing randomly in the case of a tie) regardless of the input features. You expect the majority classifier to get about 50% LOOCV error, but instead it performs very poorly. What is its LOOCV error and why does it get such an error? (*Answer concisely.*) *(3 points)*

For every iteration of LOOCV, the majority label of the training data will be different from the label of the validation data instance. For example, if the validation data example is negative, the training data will consist of 50 positive examples and 49 negative examples, and thus the majority label of the training data is positive. Hence the accuracy will always be 0% for every iteration. This leads to LOOCV error of 100%.