10-715 Advanced Introduction to Machine Learning

Mid Term Exam *Oct 27th 2014*

Name: : ______________________________

Andrew Id: : ______________________________

Department: : ______________________________

Guidelines

1. **Please don't turn this page until instructed.**
2. Write your name, Andrew Id and department in the space provided above.
3. You have seventy **(70)** minutes for this exam.
4. This exam has **eleven (11)** pages on eleven sheets of paper including this page.
5. This exam has **four (4)** questions. The number of points allocated for each question is indicated next to the question. The total number of points is 80.
6. This exam is **close book**. You cannot use any books, class notes or cheat sheets during the exam.
7. The questions vary in difficulty. The points allocated per question **do not** entirely reflect the level of difficulty. Don't spend too much time on one question.
8. If any question is unclear, you may write your own interpretation and answer the question.
9. The questions appear only on side of the paper. You may use the other side for rough work. If you still need extra paper, please ask one of the instructors.

1 Regression

1.1 Linear Regression and Neural Networks

Your are given some data $(x_i, y_i)_{i=1}^n$, where $x_i = [x_i^{(1)}, \dots, x_i^{(D)}]^\top \in \mathbb{R}^D$, $y_i \in \mathbb{R}$. There is additive Gaussian noise in the label, i.e. $y_i = f(x_i) + \epsilon$ for some unknown function f and $\epsilon \sim \mathcal{N}(0, \eta^2)$. You are asked to use techniques from linear regression to learn a model to predict y from x. You may assume that $n \gg D$ and that η^2 is small compared to the variation in the y_i values.

You split your training data into two halves: a training set on which you wish to apply your learning algorithm and a test set on which you wish to evaluate your model to choose model hyper parameters.

You begin with a simple linear basis $\phi(x) = [1, x^{(1)}, \dots, x^{(D)}]^\top$. To estimate w, you minimize a cost function of the form

$$J(w) = \sum_{i=1}^{n} (y_i - w^\top \phi(x_i))^2 \tag{1}$$

For questions 1-5 circle the correct answer (5 × 2 Points).

1. You notice that both the training error and the test set error are unacceptably high. This is most likely because our model has,

   (a) High bias

   (b) High variance

   Ans: (a)

2. You realize that a linear basis is not rich enough for the problem. So you use a quadratic basis containing all first and second order interactions as shown below,

$$\phi(x) = [1, x^{(1)}, \dots, x^{(D)}, {x^{(1)}}^2, \dots, {x^{(D)}}^2, \, x^{(1)}x^{(2)}, \, x^{(1)}x^{(3)}, \, \dots, x^{(D-1)}x^{(D)}]^\top \tag{2}$$

   When compared to the linear basis, when using the quadratic basis

   (a) The bias increases but the variance decreases.

   (b) The bias decreases but the variance increases.

   (c) Both the bias and variance increase.

   (d) Both the bias and variance decrase.

   Ans: (b)

3. You realize that the quadratic basis achieves good training error but poor test error. So you modify the cost function with a quadratic penalty: $J(w) = \sum_{i=1}^n (y_i - w^\top \phi(x_i))^2 + \lambda \|w\|^2$. Now, the training error increases slightly but the test set error decreases significantly. This is because by introducing a penalty,

   (a) We reduce both the variance and the bias.

   (b) We increase the variance slightly but significantly reduce the bias.

   (c) We increase the bias slightly but significantly reduce the variance.

   Ans: (c)

4. Consider the prediction y_* at a point x_*.

   (a) y_* is a linear combination of the labels $(y_i)_{i=1}^n$ when we use a linear basis but this is not the case when we use a quadratic basis.

(b) y_* is a linear combination of the labels $(y_i)_{i=1}^n$ when we use both the linear and quadratic bases.

(c) y_* has a complicated nonlinear connection to $(y_i)_{i=1}^n$ as we need to invert a matrix to solve linear regression.

Ans: (b)

5. Now, instead of using classical regression techniques, you decide to use neural networks to train the model. The multi layer perceptron (MLP) is a commonly used neural network and is trained using the back propagaion (BP) algorihtm. BP performs,

   (a) Gradient descent on the MLP cost function.

   (b) Newton's method on the MLP cost function.

   Ans: (a)

1.2 Multi task Regression

Now you are given K batches of data $(y_{ki}, x_{ki})_{i=1}^{n_k}$, $k = 1, \ldots, K$, $x_{ki} \in \mathbb{R}^D, y_{ki} \in \mathbb{R}$ for all i, k. Therefore you have K tasks. For each of the k tasks, you should fit *different* linear regression models for the data. However, you wish to combine them through a complexity penalty for the parameters. Precisely we wish to minimize an objective of the form,

$$\sum_{k=1}^{K}\sum_{i=1}^{n_k}(y_{ki} - \theta_k^\top x_{ki})^2 \quad + \quad R(\Theta)$$

where, $\theta_k \in \mathbb{R}^D$ are the parameters for the k^{th} task, $\Theta = [\theta_1^\top; \theta_2^\top; \ldots; \theta_K^\top] \in \mathbb{R}^{K \times D}$ and R is a complexity penalty for Θ.

Below we describe three different situations and then four options for R. To each situation, match an appropriate penalty function. Each penalty function should be matched to **only one** situation. Exactly, one penalty function will not be matched.
You do not need to provide justification.

Situations

1. You know that most of the D features are irrelevant and that the y values for each task can be predicted by a *small unknown* subset of the features. Further, these features are the *same* for all tasks.
2. You know that most of the D features are irrelevant and that the y values for each task can be predicted by a *small unknown* subset of the features. However, these features could be *different* for each task.
3. You know that all D features are relevant for all tasks. However, the amount of data n_k we have for each task is either smaller than or not significantly greater than D.

Penalty Functions

(a) $\sum_{k=1}^{K}\sum_{d=1}^{D}|\Theta_{kd}|$

(b) $\sum_{k=1}^{K}\sum_{d=1}^{D}\Theta_{kd}^2$

(c) $\sum_{k=1}^{K}\sqrt{\sum_{d=1}^{D}\Theta_{kd}^2}$

(d) $\sum_{d=1}^{D}\sqrt{\sum_{k=1}^{K}\Theta_{kd}^2}$

Give your matchings in the table below. **(3 × 2 Points)**

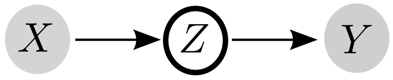
Situation	Penalty Function
1	(d)
2	(a)
3	(b)

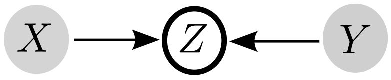
2 EM for Supervised Learning

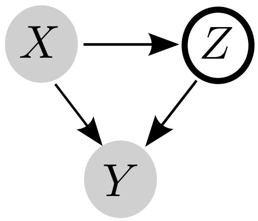
In class and in the homeworks we applied EM for unsupervised learning. Here we will look at EM for a classification problem.

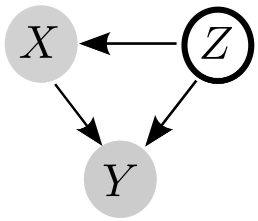
Consider the following problem in astrophysics. We want to assign a “size” label Y to a star indicating if it is a dwarf ($Y = 0$) or giant ($Y = 1$) using its spectrum X observed through telescopes. For this task we have training data $(X_i, Y_i)_{i=1}^n$. However, using X alone we cannot build a good classifier. Astronomers group stars into spectral classes Z such as brown, red, white. Note that the spectral class Z is *not* observed. However, it can be determined using the spectrum X. Further, the size labels Y can be determined using *both* X and Z. We wish to incorporate this domain knowledge when building our classifier.

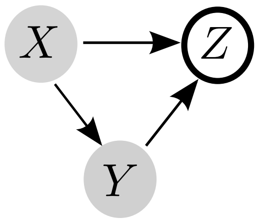
1. Circle the Bayesian network below that best captures the above scenario. We have shaded the observed variables. **(2 Points)**











Ans: First graph in second row.

Now for simplicity, we will assume that $X \in \mathbb{R}^D$ and both Z and Y are binary valued. We will not make any assumption on the distribution of X. We will use logistic regression to model Y and Z as explained below.

Recall that in logistic regression we model the label V given the features U with parameter β via,

$$p(v|u;\theta) = \sigma(\beta^\top u)^v (1 - \sigma(\beta^\top u))^{1-v}$$

where σ is the logistic function[1].

In this problem, we will model $Z|X$ via logistic regression with parameter θ and $Y|X, Z = i$, (where $i = 0, 1$) via logistic regression with parameter α_i. The parameters of the model are $\theta, \alpha_0, \alpha_1 \in \mathbb{R}^D$.

2. **(3 Points)** First write down an expression for the conditional probability $p(y|x)$. Use this to write down the conditional log likelihood of the parameters $\theta, \alpha_0, \alpha_1$.

   Ans: The conditional probability can be written as,

   $$p(y|x) = \sum_{z=0,1} p(y, z|x) = \sum_{z=0,1} p(y|z, x)p(z|x) = \sum_{z=0,1} \sigma(\alpha_z^\top x)^z(1-\sigma(\alpha_z^\top x))^{1-z}\sigma(\theta^\top x)^z(1-\sigma(\theta^\top x))^{1-z}$$

   Therefore the log likelihood is

   $$\ell(\theta, \alpha_0, \alpha_1) = \sum_{i=1}^n \log\left(\sum_{z=0,1} \sigma(\alpha_z^\top x_i)^z(1 - \sigma(\alpha_z^\top x_i))^{1-z}\sigma(\theta^\top x_i)^{y_i}(1 - \sigma(\theta^\top x_i))^{1-y_i}\right)$$

[1] $\sigma(t) = \frac{1}{1+e^{-t}}$. But you can write your answers in terms of σ

3. **(1 Point)** Are we using a discriminative model or a generative model for classification ?
Ans: discriminative

4. **(3 Points)** We can learn the parameters of the model via Expectation Maximization. Write down the E step update at the $(t+1)^{th}$ iteration. Your answer should be in terms of the estimates $\theta^{(t)}, \alpha_0^{(t)}, \alpha_1^{(t)}$ at the t^{th} iteration. **You do not need to derive the M-step.**
Ans: We can lower bound the log likelihood using Jensen's inequality,

$$\ell(\theta, \alpha_0, \alpha_1) \leq \sum_i \sum_{z=0,1} R_{iz} \log \left(\frac{p(y_i, z; x_i)}{R_{iz}} \right)$$

In the E-step we compute R via

$$R_{iz} = p(z|y_i, x_i; \theta^{(t)}, \alpha_0^{(t)}, \alpha_1^{(t)}) = \frac{p(y_i|z, x_i; \alpha_z^{(t)}) p(z|x_i; \theta^{(t)})}{p(y_i|z=0, x_i; \alpha_0^{(t)}) p(z=0|x_i; \theta^{(t)}) + p(y_i|z=1, x_i; \alpha_1^{(t)}) p(z=1|x_i; \theta^{(t)})}$$

5. **(5 Points)** Consider the following statements about the EM algorithm *in general*. State if they are True (T) or False (F).

   (a) The EM Algorithm maximizes a lower bound on the log likelihood at each iteration.

   (b) The EM Algorithm minimizes an upper bound on the log likelihood at each iteration.

   (c) The EM algorithm *never* decreases the log likelihood.

   (d) EM is guaranteed to converge to a unique globally optimal solution.

   (e) It is *impossible* to use gradient descent/ ascent to learn the parameters instead of EM.

   Ans:

   (a) T

   (b) F

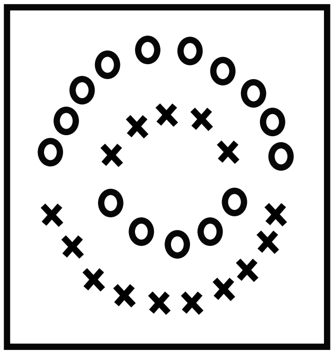
   (c) T

   (d) F

   (e) F

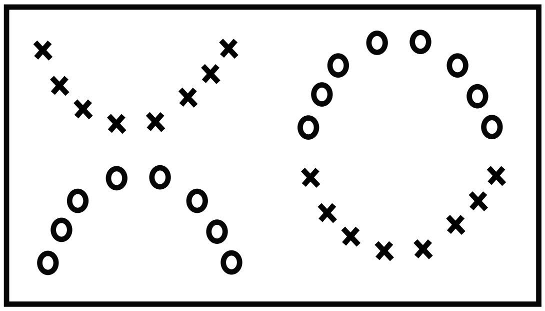
6. **(3 × 2 Points)** Given below are three different datasets on which we wish to apply this model (without any transformation of the variables). Which datasets can we perfectly classify using this model ? We have indicated the two classes using circles ($\circ$) and crosses ($\times$). A simple Yes/ No answer is sufficient.

   (a)



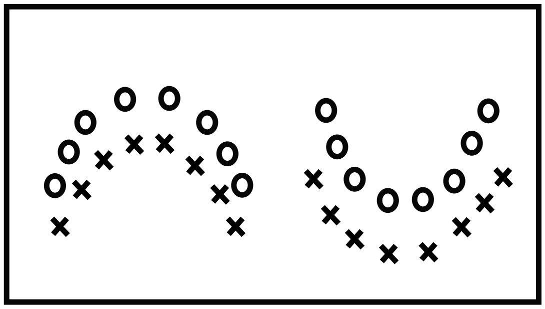
Ans: No

(b)



Ans: Yes

(c)

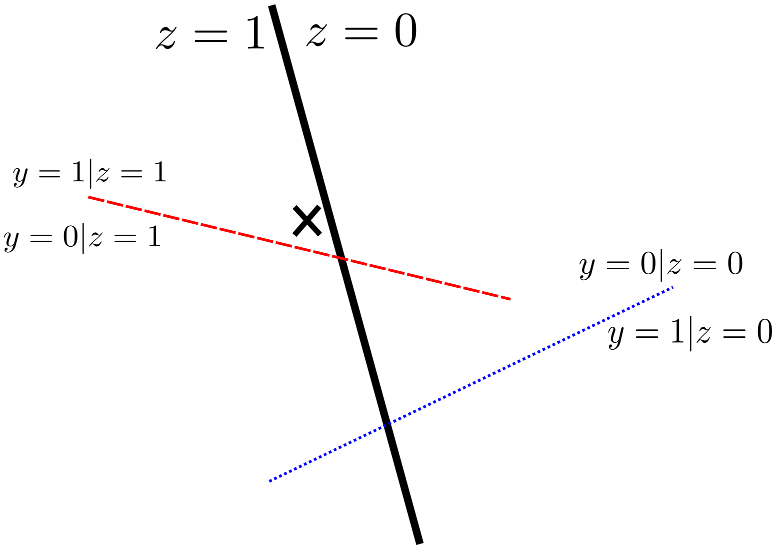


Ans: No

7. **(4 Points)** Consider the following two decision rules to classify a new point x_* using this model.

- First choose $z_* = \operatorname{argmax} p(z|x_*;\theta)$. Then choose $y_* = \operatorname{argmax} p(y|z_*, x_*; \alpha_{z_*})$.
- Choose $y_* = \operatorname{argmax} p(y|x_*; \theta, \alpha_0, \alpha_1)$

Show that these two decision rules are not equal.
You do not need to provide a proof but should sufficiently explain why this occurs. One option, is to illustrate via a two dimensional example. You can depict the various logistic regression classifiers via straight lines in your illustration.



Ans:

You were not required to provide such a descriptive answer. A simple explanation that conveys the idea is sufficient.

In the figure above, the black solid line is the decision boundary for $z|x$. The red dashed curve is the decision boundary for $y|x, z = 1$ and the blue dotted line is the decision boundary for $y|x, z = 0$. The cross(×) is the point we wish to classify. First decision rule: Since the point is on the left of the solid line the we will choose $z = 1$ and then since it is above the red line we will choose $y = 1$.

Second decision rule: I will throw in some numbers here for ease of explanation. Since the point is just on the left of the black line (it was only marginally determined to be $z = 1$), say, $p(z = 1|x) = 0.51$ and $p(z = 0|x) = 0.49$. Again, since it is just above the red line, say, $p(y = 1|z = 1, x) = 0.51$ and $p(y = 0|z = 1, x) = 0.49$. However, since it is well above the blue line, say, $p(y = 0|z = 0, x) = 0.99$ and $p(y = 1|z = 0, x) = 0.01$. Therefore we have,

$$\begin{aligned} p(y = 1|x) &= p(y = 1|z = 0, x)p(z = 0|x) + p(y = 1|z = 1, x)p(z = 1|x) \\ &= 0.01 \times 0.49 + 0.51 \times 0.51 = 0.265 \end{aligned}$$

Therefore $p(y = 0|x) = 0.735$ and we will choose $y = 0$ with the decision rule. You don't need to give all these numbers–an intuitive explanation is sufficient.

Also, the distance from the decision boundary alone does not determine the probability–the magnitude of the weights also matter. But, this should convey the intuition.

Solution to Problem 3

1. (a) We usually solve the dual because it can be kernelized, and it has easy box constraints. However, note the paper by Chappelle on how to kernelize the SVM primal.

   (b) i. $\{(x_i, y_i)|i = 1, 2, 3, 4, 9, 10, 11, 12\}$ as α_i is non-zero for those points.

   ii. All except $(x_1, y_1), (x_{12}, y_{12})$ are definitely not misclassified because the dual variable should be $< C$ for those points.

   (c) Points on the margins or those crossing the margins correspond to support vectors. C' is smaller because it results in more support vectors. Informally, think of what happens when C grows very large or very small.

2. No, because g is not a concave function.

3. The given points are separable under the kernel given by the feature map $\phi(x) = (x, (x-1)^2)$.

4. The RKHS is

$$\mathcal{H} = \left\{ f : \mathbb{R}^d \to \mathbb{R} | \exists w \in \mathbb{R}^d \ni f(x) = \langle w, x \rangle, \forall x \in \mathbb{R}^d \right\}$$

   and the inner product between $f = \langle w_1, . \rangle$ and $g = \langle w_2, . \rangle$ is $\langle w_1, w_2 \rangle$. $\mathcal{H}$ is a vector space spanned by $\{b_i = \langle e_i, . \rangle | i \in [d]\}$ where $e_i \in \mathbb{R}^d$ with all zeros except a 1 in position i. Further the b_i's are orthogonal to each other in $\mathcal{H}$. So the dimension of $\mathcal{H}$ is d.

5. It reduces the infinite dimensional problem to a finite dimensional one when we have finite amount of data.

Solution to Problem 4

1. (a) $2^T - 1$

   (b) The graphical model should have directed edges to X_{i+2} from X_i, X_{i+1} for $i = 1, 2, \cdots, T-2$ and an edge to X_2 from X_1. It should have no other edges.

   (c) $4 * T - 5$. Four each for the conditional distribution of $X_{i+2}|X_{i+1}, X_i$ for $i = 1, 2, \cdots, T-2$, two for $X_2|X_1$ and one for the marginal of X_1.

2. (a) Viterbi

   (b) No, because posterior decoding might result in a suboptimal decoding. See lecture slides for an example.

   (c) It is used for learning an HMM. Yes, it is an EM algorithm.

   (d) False, CRF models $P(Y|X)$ and so it is a discriminative model, whereas HMM is a generative model.

   (e) By normalizing globally instead of locally.

3. (a) Detailed balance is a desirable property in MCMC sampling because it sufficient to give stationarity to the distribution in the detailed balance.

   (b) We discard first several samples in MCMC because the chain might not have mixed well for those samples. In general, it is hard to decide how many samples need to discarded.

   (c) $T(x \to x') = p(x'_i|x_{-i})$ where $x'_{-i} = x_{-i}$, when the ith variable is being updated.