Machine Learning 10-701 Midterm Exam March 4, 2015

Name: ______________________________ **Andrew ID:** ____________________

Instructions

- Anything on paper is OK in arbitrary shape size, and quantity.
- Electronic devices are not acceptable. This includes iPods, iPads, Android tablets, Blackberries, Nokias, Windows phones, Microsoft Surface, laptops, Chromebooks, MP3 players, digital cameras, Google Glass, Android Wear, or anything else that requires electricity.[1]
- **If we catch you cheating or using any such device, the exam is over for you. Your results will not count and you will have failed this exam automatically. There might be more serious consequences, too. No exceptions. Switch off your phones before the exam.**
- There are more questions in the exam than what you are likely to be able to solve in 90 minutes time. Choose wisely and answer those first that you believe you can solve easily. This is an opportunity.
- None of the questions should require more than 1 page to answer. Its OK to be concise if you address the key points of the derivation (it should be human-readable, though).

Point Distribution

Problem	Points
1	/10
2	/5
3	/5
4	/10
5	/5
6	/10
7	/5
8	/10
9	/15
10	/10
Total	/85

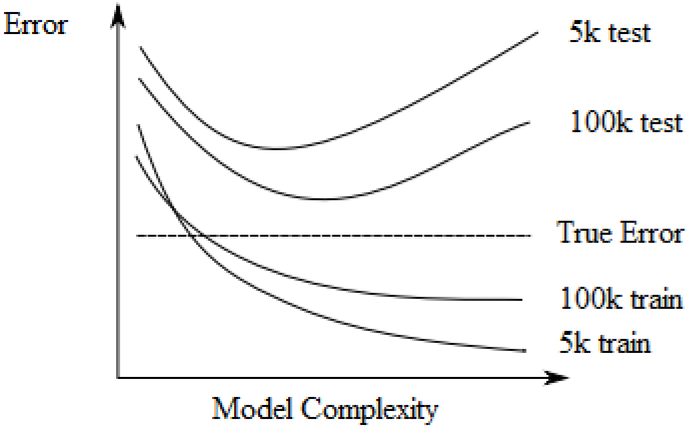
[1] Obvious exceptions for pacemakers and hearing aids.

1 Quick Questions (10 points)

1.1 Model Selection

Assume two data sets sampled from the same distribution where the number of observations for each data is $5,000$ and $100,000$ respectively. Randomly construct the train and test set by dividing the data 90:10.

- Draw two curves for training error and test error for each data set with y-axis denoting the error and x-axis denoting the model complexity.
- You should have total of 4 curves: one training error and one test error curve for each dataset.
- Draw all 4 of them in the same diagram.
- Clearly *mark* all your curves.



Solution:

- The training error decreases when increasing the model complexity, while the test error decreases first but then increases due to overfitting.
- Given the same model complexity, the model has larger training samples is less likely to overfit than the one with less samples. So the two curves representing 100k samples are nearer the dash line the other two curves.

1.2 k Nearest Neighbor Classification

Suppose we have a large training set. Name a drawback when using a k Nearest Neighbor during testing.

Solution:
k-NN is slow in testing phase, since the time complexity for finding k nearest neighbors is $O(knd)$. n is number of training data points. d is number of dimensions.

1.3 Density Estimation

- List two drawbacks of bin counting.

  Solution:
  Not continuous, curse of dimensionality, zero density if no training points in the bin, etc.

- What is the advantage of Parzen Windows compared to bin counting?

  Solution:
  Parzen Window Method gives smoother pdf than bin counting.

- Suggest a method to handle low density regions in a Watson-Nadaraya estimator.

  Solution:
  Use average distance from k nearest neighbors. Non-uniform bandwidth for smoother.

2 Probabilities (5 points)

2.1 Conditional Independence

Construct an example of three random variables X, Y and Z, such that $X \perp Y$ but $X \not\perp Y|Z$.

Solution:
Let X and Y be independent Bernoulli(0.5) random variables. Now define $Z = X \oplus Y$. Then, we can see that $X \perp Y$ but $X \not\perp Y|Z$.

2.2 Tough Machine Class

The *Tough Machine Learning Course* 10-801 is attended by students majoring in ML and some students that don't major in ML. In it only 50% of the ML students and 30% of the non-ML students pass the midterm exam. Unfortunately 60% of the entire class are non-ML students. What is the percentage of ML students among those that actually pass the exam.

Solution:

Let S denote that a person pass the midterm. Let M denote that a person is a ML major, and N denotes the otherwise.

$$P(M) = 0.4, P(N) = 0.6, P(S|M) = 0.5, P(S|N) = 0.3$$

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{P(S|M)P(M)}{P(S|M)P(M) + P(S|N)P(N)}$$

$$\frac{0.5 * 0.4}{0.5 * 0.4 + 0.3 * 0.6} = \frac{10}{19}$$

2.3 Gaussians for free

Denote by X_i random variables drawn from the uniform distribution $U[0, 1]$. Design a random variable

$$Z_n := f(X_1, \ldots, X_n)$$

such that Z_n converges to the normal distribution for $n \to \infty$.

Solution:
Many possibilities. One can be $f(X_1, \ldots, X_n) = \frac{1}{n} \sum_{i=1}^{n} X_i$.

3 Naive Bayes Classifier (5 points)

Annabelle Antique is a collector of old paintings. She is sick of getting e-mails offering her fake artwork and wants to train her own Naive Bayes classifier such that she doesn't have to read all the spam any longer. One of the words she knows that corresponds to a fake is the occurrence of the word *replica*. The Naive Bayes classifier doesn't know this yet. All it can do is compute probabilities. Your job is to help it by generating suitable messages:

3.1 Positive and Negative Examples

Generate two messages corresponding to ham and spam respectively, which will lead to the classifier correctly recognizing *replica* as spam.

Solution:
Ham 1: Van Gogh's Starry night in canvas
Ham 2: The original copy of Da Vinci's Vitruvian Man
Spam 1: Van Gogh's Starry night oil paining replica
Spam 2: Da Vinci's Vitruvian Man adapted to oil painting

3.2 Misclassification

Generate a message that would be incorrectly classified as ham based on the four messages generated above. Explain why.

Solution:
An exquisite copy of Van Gogh's Starry night

3.3 Breaking Naive Bayes

Generate a message that would lead to an undefined probability estimate. Explain why. Suggest how to fix the Naive Bayes classifier.

4 Perceptron Algorithm (10 points)

Assume that you are given observations $(x, y) \in \mathbb{R}^2 \times \{\pm 1\}$ in the following order:

Instance	1	2	3	4	5	6	7	8
Label y	+1	-1	+1	-1	+1	-1	+1	+1
Data (x_1, x_2)	(10,10)	(0,0)	(8,4)	(3,3)	(4,8)	(0.5,0.5)	(4,3)	(2,5)

Show the action of the perceptron algorithm for the above sequence of observations. We start with an initial set of weights $w = (1, 1)$ and bias $b = 0$.

Solution:

Let us define $g(y^{(i)}, w, x^{(i)}, b) = y^{(i)}(\langle w, x^{(i)} \rangle + b)$ and simplify notation by $g_i = g(y^{(i)}, w, x^{(i)}, b)$. Rewriting the perceptron algorithm, there is update $w^{(j)} \longleftarrow w^{(j-1)} + y^{(i)}x^{(i)}, b^{(j)} \longleftarrow b^{(j-1)} + y^{(i)}$ if $g_i \leq 0$ and no update otherwise. Following this algorithm instance by instance with the starting point $w^{(0)} = (1, 1),\ b^{(0)} = 0$,

1. $g_1 = +1 \cdot (10 + 10) > 0 \to$ no update:
   $w^{(1)} = (1, 1)$
   $b^{(1)} = 0.$

2. $g_2 = -1 \cdot (0 + 0) \leq 0 \to$ update:
   $w^{(2)} = (1, 1) + (0, 0) = (1, 1)$
   $b^{(2)} = 0 - 1 = -1.$

3. $g_3 = +1 \cdot (12 - 1) > 0 \to$ no update:
   $w^{(3)} = (1, 1)$
   $b^{(3)} = -1.$

4. $g_4 = -1 \cdot (6 - 1) < 0 \to$ update:
   $w^{(4)} = (1, 1) - (3, 3) = (-2, -2)$
   $b^{(4)} = -1 - 1 = -2.$

5. $g_5 = +1 \cdot (-8 - 16 - 2) < 0 \to$ update:
   $w^{(5)} = (-2, -2) + (4, 8) = (2, 6)$
   $b^{(5)} = -2 + 1 = -1.$

6. $g_6 = -1 \cdot (1 + 3 - 1) < 0 \to$ update:
   $w^{(6)} = (2, 6) - (0.5, 0.5) = (1.5, 5.5)$
   $b^{(6)} = -1 - 1 = -2.$

7. $g_7 = +1 \cdot (6 + 16.5 - 2) > 0 \to$ no update:
   $w^{(7)} = (1.5, 5.5)$
   $b^{(7)} = -2.$

8. $g_8 = +1 \cdot (3 + 27.5 - 2) > 0 \to$ no update:
   $w^{(8)} = (1.5, 5.5)$
   $b^{(8)} = -2.$

5 Representer Theorem (5 points)

Denote by $\ell : \mathbb{R} \to RR$ a nonnegative convex differentiable function, i.e. a loss function. Moreover, denote by $\|\cdot\|_2$ the Euclidean norm. Finally, let $y_i \in \mathbb{R}$ be scalars and $x_i, w \in \mathbb{R}^d$ be vectors (i.e. $d \in \mathbb{N}$). Prove that the solution of the optimization problem

$$\underset{w}{\text{minimize}} \sum_{i=1}^{n} \ell(y_i - \langle x_i, w \rangle) + \|w\|_2$$

can be written as

$$w^* = \sum_i \alpha_i x_i \text{ for some } \alpha_i \in \mathbb{R}.$$

Solution:

You can express w in terms of linear combination of x_i and v, i.e. $w = \alpha_0 v + \sum_{k=1}^n \alpha_k x_k$, where $v \in null\{x_1, \ldots, x_n\}$ (in other expression $v \perp \text{span}\{x_1, \ldots, x_n\}$). Then, plugging in the $w = \alpha_0 v + \sum_{k=1}^n \alpha_k x_k$ expression in to the objective function,

$$\begin{aligned}
\sum_i^n \ell(y_i - \langle x_i, w \rangle) + \|w\|_2 &= \sum_i^n \ell(y_i - \langle x_i, \alpha_0 v + \sum_{k=1}^n \alpha_k x_k \rangle) + \|\alpha_0 v + \sum_{k=1}^n \alpha_k x_k\|_2 \\
&= \sum_i^n \ell(y_i - \sum_{i=1}^n \alpha_i' x_i) + \sqrt{\|\alpha_0 v\|_2^2 + \|\sum_{k=1}^n \alpha_k x_k\|_2^2} \quad \text{using } v \perp \text{span}\{x_1, \ldots, x_n\} \\
&\geq \sum_i^n \ell(y_i - \sum_{i=1}^n \alpha_i' x_i) + \|\sum_{k=1}^n \alpha_k x_k\|_2
\end{aligned}$$

We just proved that the objective function is smaller when the wieght of v, w_0 is 0, and therefore we conclude that $\hat{w} = \sum_i^n \alpha_i x_i$.

6 Bundle Methods (10 points)

Let us revisit the bundle method problem we discussed in class. In each iterative step, we need to solve the following optimization problem:

$$\underset{w}{\text{minimize}} \max_{i \in \{1, \ldots k\}} \langle a_i, w \rangle + b_i + \frac{1}{2} \|w\|_2^2$$

where $w, a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ for $1 \leq i \leq k$.

6.1 Constrained Optimization Problem

Rewrite the optimization problem as the quadratic programming problem. Hint: substitute the maximum with an auxiliary slack variable ξ.

Solution:
Given $A = [a_1, a_2, \ldots, a_k]$, $z \in \mathbb{R}$, and $\mathbf{1} \in \mathbb{R}^n$, we can rewrite the primal problem as

$$\begin{aligned} \min_{x,\xi} \quad & \xi + \frac{1}{2}\|w\|_2^2 \\ \text{s.t. } & A^T w + b \leq \xi \mathbf{1} \end{aligned} \tag{1}$$

Caution In order to have the same minimization problem, you have to minimize with ξ as well on the final expression. -1 if this point is missed. In order to remove $\underset{i}{max}$, you need b_i incorporated with the constraint related to the ξ. -1 if this point is missed and hence still has the term $b_i$in the final expression.

6.2 Dual Problem

Derive the dual optimization problem.

Solution:
Writing the Lagrangian function with slack variable $u \geq 0, \ u \in \mathbb{R}^n$.

$$L(w, \xi, u) = \xi + \frac{1}{2}\|w\|_2^2 + u^T(A^T w + b - \xi \mathbf{1}),$$

the dual problem of (6.2) is defined as

$$\max_{u \geq 0} \min_{w,\xi} L(w, \xi, u) = \max_{u \geq 0} \min_{w,\xi} \ \xi + \frac{1}{2}\|w\|_2^2 + u^T(A^T w + b - \xi \mathbf{1}).$$

Let's solve over $\min_{w,\xi}$, first minimizing over w, we obtain the following

$$\frac{\partial L(w, \xi, u)}{\partial w} = 0 \Longleftrightarrow w = -Au.$$

Now, plugging uA into the optimization problem becomes

$$\max_{u \geq 0} \min_{\xi} L(\xi, u) = \max_{u} \min_{\xi} \xi \mathbf{1} - \frac{1}{2} u^T A^T A u + u^T b - \xi u^T \mathbf{1}.$$

Solving over the $\min_{\xi}$,

$$\frac{\partial L(\xi, u)}{\partial \xi} = 0 \Longleftrightarrow u^T \mathbf{1} = 1 \Leftrightarrow \|u\|_1 = 1 \quad (\because u \geq 0)$$

Then, we can remove the terms related to the ξ and finally obtain the quadratic programming problem.

$$\begin{aligned} \max_{u} - \frac{1}{2} u^T A^T A u + u^T b \\ \text{s.t. } u \geq 0, \|u\|_1 = 1, \end{aligned} \quad \text{or rewriting as} \quad \begin{aligned} -\min_{u} \frac{1}{2} u^T A^T A u - u^T b \\ \text{s.t. } u \geq 0, \|u\|_1 = 1 \ . \end{aligned}$$

7 Entropy of Exponential Family (5 points)

The entropy of a random variable X distributed according to a probability density function $p(\cdot)$ is given by

$$H[X] = -\int p(x) \log p(x) dx$$

What is the entropy if the probability density function belonged to exponential family, i.e.

$$p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))$$

Express your answer in terms of $\theta, g(\theta)$ and $\nabla g(\theta)$, i.e. the gradient of $g(\theta)$.

Solution:

Multiple ways to solve it. One is as follows: Let us first pre-compute $\nabla g(\theta) = \int p(x|\theta)\phi(x)dx$ Now evaluating the entropy using the definition:

$$\begin{aligned} H[X] &= -\int p(x|\theta) \log p(x|\theta) dx \\ &= -\int p(x|\theta)(\langle \phi(x), \theta \rangle - g(\theta))dx \\ &= -\int p(x|\theta)\langle \phi(x), \theta \rangle dx + \int p(x|\theta) g(\theta) dx \\ &= -\left\langle \int p(x|\theta)\phi(x)dx, \theta \right\rangle + g(\theta) \int p(x|\theta)dx \\ &= -\langle \nabla g(\theta), \theta \rangle + g(\theta) \end{aligned} \tag{2}$$

8 Kernels (10 points)

Let $k(\cdot, \cdot)$ and $l(\cdot, \cdot)$ with $k, l : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be Mercer kernels. That is, k, l are symmetric and any Gram matrix constructed based on them is positive semidefinite.

8.1 Linear Combinations

Prove that $\alpha k(x, x') + l(x, x')$ is a valid kernel for any $\alpha \geq 0$.

Solution:

One proof strategy is based on explicit feature map construction by concatenation. Specifically, let $k(x, x') = \langle \phi(x), \phi(x') \rangle$ and $l(x, x') = \langle \psi(x), \psi(x') \rangle$. Then the feature map for the linear combination is $\begin{pmatrix} \sqrt{\alpha}\phi \\ \psi \end{pmatrix}$, i.e. $\alpha k(x, x') + l(x, x') = \langle \begin{pmatrix} \sqrt{\alpha}\phi \\ \psi \end{pmatrix}(x), \begin{pmatrix} \sqrt{\alpha}\phi \\ \psi \end{pmatrix}(x') \rangle$

8.2 Upper Bound

Prove the inequality $k^2(x, x') \leq k(x, x)k(x', x')$.

Solution:

The proof follows from Cauchy-Schwartz inequality in the feature space. Specifically, let $k(x, x') = \langle \phi(x), \phi(x') \rangle$. Then $k(x, x) = \|\phi(x)\|^2$ and $k(x', x') = \|\phi(x')\|^2$. By Cauchy-Schwartz, we have $\|\phi(x)\|\|\phi(x')\| \geq \langle \phi(x), \phi(x') \rangle$ from which the result follows immediately.

8.3 Kernel Product

Prove that $k(x, x')l(x, x')$ is a kernel. Hint: design an explicit feature map for $k(x, x')l(x, x')$.

Solution:

One proof strategy is to show that elementwise product of any two positive semidefinite matrices is always a positive semidefinite matrix.

Another proof strategy is based on explicit feature map construction as kronecker product. Specifically, let $k(x, x') = \langle \phi(x), \phi(x') \rangle$ and $l(x, x') = \langle \psi(x), \psi(x') \rangle$. Then the feature map for the kernel product is $\phi \otimes \psi$, i.e. $k(x, x')l(x, x') = \langle (\phi \otimes \psi)(x), (\phi \otimes \psi)(x') \rangle$

9 Load Balancing (15 points)

In general, load balancing is a problem to distribute tasks among multiple resources. This has useful application across computer science and in particular large scale distributed machine learning algorithms.

Assume that we have M machines and N independent tasks that require the same work to execute. Ideally we want to assign the tasks to machines so that every machine gets approximately the same amount of work. For this we use the following load balancing algorithm

$$m(t) := \operatorname*{argmin}_{m' \in \{1, \dots M\}} h(t, m').$$

Here $t \in \{1, \dots N\}$ is the task and h is an ideal hash function. That is, for all practical purposes consider the value associated with (t, m) as a random variable, drawn independently from any other pair (t', m') over all integers $\left\{1, 2^{64} - 1\right\}$ (you do not need to worry about collisions).

9.1 Expected Load

Show that the *expected fraction* of work per machine is $1/M$ regardless of the total number of tasks.

Solution:
Just need to identify uniform distribution.

9.2 Maximum Load for a Machine

Bound the probability that the load for a machine, e.g. machine #42, will exceed $1/M$ by ϵ. Hint: use Hoeffding's theorem.

Solution:
Let X_m denote the load for machine m. Applying this Hoeffding's inequality to the problem, we have:

$$\Pr\left(X_m > \frac{1}{M} + \epsilon\right) \leq \exp\left(-2N\epsilon^2\right) \tag{3}$$

9.3 Worst Case for the Cluster

Give a lower bound on the probability that *none of the machines* in the cluster will need to perform more than a fraction of $1/M + \epsilon$ work. Hint: why can you treat the machines as if they were independent.

Solution:

$$\begin{aligned} \Pr\left(\forall m : X_m < \frac{1}{M} + \epsilon\right) &= 1 - \Pr\left(\exists m : X_m > \frac{1}{M} + \epsilon\right) \\ \text{By union bound} &\geq 1 - \sum_m \Pr\left(X_m > \frac{1}{M} + \epsilon\right) \\ &\geq 1 - M \exp\left(-2N\epsilon^2\right) \end{aligned} \tag{4}$$

Compute a bound on the number of independent work packages N in terms of ϵ and the confidence $1 - \delta$.

Solution:

Being conservative, we need to have:

$$\Pr\left(\forall m : X_m < \frac{1}{M} + \epsilon\right) \geq 1 - \delta \tag{5}$$

But we can only compute the lower bound for the probability. So we would like to have the lower bounder to be higher than the minimum confidence required, i.e.

$$\begin{aligned} 1 - M \exp\left(-2N\epsilon^2\right) &\geq 1 - \delta \\ M \exp\left(-2N\epsilon^2\right) &\leq \delta \\ N &\geq \frac{1}{2\epsilon^2} \log \frac{M}{\delta} \end{aligned} \tag{6}$$

9.4 Not enough Tasks

Now assume that $M = N$. What is the probability that a particular machine is empty? What is the expected fraction of machines sitting idle for $M = N \to \infty$.

Solution:

Probability a particular machine being empty $= (1 - \frac{1}{N})^N$
As $N \to \infty$, we have $(1 - \frac{1}{N})^N \to \frac{1}{e}$
Therefore approximately $\frac{N}{e}$ computers are sitting idle.

10 Shoe Distribution (10 points)

Assume that each student in class has on average 3 pairs, i.e. 6 shoes. Moreover, assume that the standard deviation in the number of shoes is 1.5. Consider the event that a randomly selected student has at least 9 shoes.

10.1 Markov's Inequality [2 pts]

Use Markov's inequality to compute an upper bound on the probability of this event.

Solution:

$$P(X \geq 9) \leq \frac{E[X]}{9} = \frac{2}{3}$$

10.2 Chebyshev's Inequality [3 pts]

Use Chebyshev's inequality to compute an upper bound on the probability of this event.

Solution:

Similar to the previous one, two versions are acceptable.

$$P(X \geq 9) \leq P(|X - E[X]| \geq 3) \leq \frac{Var[X]}{3^2} = \frac{1.5^2}{3^2} = \frac{1}{4}$$

10.3 A tighter fit [2 pts]

Explain how you could tighten the upper bound computed above (hint — shoes come in pairs).

Solution:

Using $P(X \geq 10)$ with Chebyshev's Inequality will get you a tighter bound.

$$P(X \geq 10) \leq P(|X - E[X]| \geq 4) \leq \frac{Var[X]}{4^2} = \frac{1.5^2}{4^2} = \frac{9}{64}$$

(This page was left blank intentionally)

(This page was left blank intentionally)