Course: STAT W4400
Title: Statistical Machine Learning
Semester: Fall 2014
Instructor: John P. Cunningham

MIDTERM EXAM

Explanation

This exam is to be done in-class. You have 75 minutes to complete the entirety. All solutions should be written in the accompanying blue book. No other paper (including this exam sheet) will be graded. **To receive credit for this exam, you must submit blue book with the exam paper placed inside.** As reference you may use one sheet of 8.5 × 11in paper, on which any notes can be written (front and back). A calculator may be used for simple calculations only (e.g., no stored formulas). No other materials are allowed (including textbooks, computers, and other electronics). To receive full credit on multi-point problems, you must thoroughly explain how you arrived at your solutions. Each problem is divided up into several parts. Many parts can be answered independently, so if you are stuck on a particular part, you may wish to skip that part and return to it later. Good luck.

1. (25 points)
A particular gambling scheme involves n rounds of play. In the first round, the payoff for is $X_1 \sim exp(\theta, \lambda)$ for some payoff parameters θ and λ. For the ith round, the payoff is $X_i | X_{i-1} \sim exp(\theta + X_{i-1}, \lambda + X_{i-1})$. Recall: we say $X \sim exp(a, b)$ if:

$$f_X(x) = ae^{-a(x-b)} \mathbb{1}\{x \geq b\}$$

(a) (5 points) I give you the observed payouts from all n rounds, $X_1, ..., X_n$. What is the maximum likelihood estimate of λ, namely λ_{ML}?

Solution:

We start by writing out the likelihood. Let $X_0 := 0$. Then

$$\begin{aligned} L(\theta, \lambda) =& \prod_{i=1}^{n} (\theta + X_{i-1}) e^{-(\theta + X_{i-1})(X_i - X_{i-1} - \lambda)} \mathbb{1}\{X_i \geq X_{i-1} + \lambda\} \\ =& \mathbb{1}\{\lambda \leq \Delta\} \prod_{i=1}^{n} (\theta + X_{i-1}) e^{-(\theta + X_{i-1})(X_i - X_{i-1} - \lambda)}, \end{aligned}$$

where $\Delta := \min_i (X_i - X_{i-1})$. For $\lambda \leq \Delta$, the likelihood is non-negative and strictly increasing in λ. For $\lambda > \Delta$, the likelihood is zero. So, $\lambda_{ML} = \Delta$.

Note that the usual approach of finding the MLE by taking the derivative of the log-likelihood, setting it equal to zero, and solving for λ. You could instead formulate the problem as a constrained optimization problem, with the constraint $\lambda - \Delta \leq 0$, in which case you immediately find that $\lambda_{ML} = \infty$ if the constraint is not active (but is not a situation of interest), or that $\lambda_{ML} = \Delta$ if the constraint is active.

(b) (8 points) Assume we know λ_{ML}. I give you the observed payouts from all n rounds, $X_1, ..., X_n$. Write down an optimization problem for θ_{ML}. Be sure to transform the optimization to a more computationally tractable form.

Solution:

We found λ_{ML} in part (a), so we can write our log-likelihood as

$$\ell(\theta, \lambda_{ML}) = \sum_{i=1}^{n} [\log(\theta + X_{i-1}) - (\theta + X_{i-1})(X_i - X_{i-1} - \lambda_{ML})]$$

The optimization problem we want to solve is then

$$\theta_{ML} = \arg\max_{\theta \geq 0} \left\{ \sum_{i=1}^{n} [\log(\theta + X_{i-1}) - (\theta + X_{i-1})(X_i - X_{i-1} - \lambda_{ML})] \right\}$$

(c) (8 points) Write down an optimization algorithm that will optimize the function from the previous part. There are several choices; you may choose the simplest. Describe how the algorithm proceeds, and write an expression for the update rule, which should only involve θ, λ_{ML}, and the data $X_1, ..., X_n$.

Solution:

We can use any variant of gradient ascent to find the maximum. To do find the updates, we need the gradient of $\ell(\theta, \lambda_{ML})$ with respect to θ.

$$\frac{\partial \ell(\theta, \lambda_{ML})}{\partial \theta} = \sum_{i=1}^{n} \frac{1}{\theta + X_{i-1}} - (X_i - X_{i-1} - \lambda_{ML})$$

To simplify notation, let $\Delta_i := X_i - X_{i-1}$, and $\bar{\Delta} := \frac{1}{n}\sum_{i=1}^{n} \Delta_i$. Then our update is

$$\theta_{t+1} := \theta_t + \alpha_t \left(n\lambda_{ML} - n\bar{\Delta} + \sum_{i=1}^{n} \frac{1}{\theta_t + X_{i-1}} \right)$$

for some appropriately chosen step size, α_t. Alternatively, we can use the second derivative to use the Newton-Raphson algorithm:

$$\theta_{t+1} := \theta_t + \left(-\sum_{i=1}^{n} \frac{1}{(\theta + X_{i-1})^2} \right)^{-1} \left(n\lambda_{ML} - n\bar{\Delta} + \sum_{i=1}^{n} \frac{1}{\theta_t + X_{i-1}} \right)$$

(d) (4 points) In terms of the data $X_1, ..., X_n$, what are the smallest and largest values that λ_{ML} can be?

Solution:

$\lambda_{ML} = \min_i(X_i - X_{i-1})$.

2. (25 points)
The following questions all consider a binary classifier $f : \mathbb{R}^d \to \{-1, +1\}$.

(a) (3 points) Do most machine learning algorithms use risk $R(f)$ or empirical risk $\hat{R}_n(f)$, and why?

Solution:

Empirical risk, due to uncertainty about the true distribution of the data.

(b) (3 points) If the training data $\{(x_1, y_1), ..., (x_n, y_n)\}$ for a fixed classifier f are n iid draws from the true underlying distribution of the data, what is:

$$\lim_{n\to\infty} \left| R(f) - \hat{R}_n(f) \right|$$

Please make a simple argument; no proof is required. (Technical note: you may assume that $R(f)$ is well behaved such that questions of convergence are all appropriately satisfied).

Solution:

0.

(c) (3 points) Under the usual 01 loss, what is the range of $R(f)$? With this answer, interpret $R(f)$ in words as a probability (one sentence will suffice).

Solution:

$[0, 1]$. $R(f)$ under the 01 loss is the probability that a given classifier f makes an error.

(d) (2 points) Training procedure 1 chooses linear classifiers f^1 entirely at random. Now the risk $R(f^1)$ is a random variable (a function of the random variable f^1). What is $E\left(R\left(f^1\right)\right)$ under the 01 loss?

Solution:

0.5

(e) (2 points) Training procedure 2 uses a soft-margin SVM to choose a linear classifier f^2 according to a training set $\{(x_1, y_1), ..., (x_n, y_n)\}$ drawn iid from the true underlying distribution. By analogy to the previous part, you can consider that training procedure 2 chooses linear classifiers f^2 *better than* entirely at random. Do you expect $E\left(R\left(f^2\right)\right)$ to be larger or smaller than $E\left(R\left(f^1\right)\right)$, again under the same 01 loss?

Solution:

Smaller. Risk under the 01 loss is an error measure, so the soft-margin SVM will do better.

(f) (8 points) Training procedure 3 repeats training procedure 2 independently m times (assume m is odd), each time with a new training set drawn iid from the true underlying distribution, producing classifiers $f_1^2, f_2^2, ...f_m^2$. If I let $f^3(x) = \text{sign}\left(\sum_{k=1}^m f_k^2(x)\right)$. What is $E\left(R\left(f^3\right)\right)$ in terms of $E(R(f^2))$? Do not try to simplify the solution entirely.

Solution:

Note that probability of a correct classification under the 0-1 loss is $p = 1 - E(R(f^2))$. Then use Condorcet's jury theorem.

$$\begin{aligned}
P(correct) &= 1 - E(R(f^3)) \\
&= \sum_{j=\frac{m+1}{2}}^{m} \binom{m}{j} p^j (1-p)^{m-j} \\
&= \sum_{j=\frac{m+1}{2}}^{m} \binom{m}{j} (1 - E(R(f^2)))^j (E(R(f^2)))^{m-j} \\
&\Rightarrow \\
E(R(f^3)) &= \sum_{j=1}^{\frac{m+1}{2}} \binom{m}{j} (1 - E(R(f^2)))^j (E(R(f^2)))^{m-j}
\end{aligned}$$

(g) (4 points) Training procedure 4 uses AdaBoost, with m classifiers of type f^2, on a single training set $\{(x_1, y_1), ..., (x_n, y_n)\}$ to produce f^4. Do you expect $E\left(R\left(f^4\right)\right)$ to be larger or smaller than $E\left(R\left(f^3\right)\right)$, again under the same 0-1 loss?

Solution:

Larger. Adaboost has correlated data sets.

3. (25 points)
Consider a soft-margin, linear support vector machine:

$$\begin{aligned}
\min_{\mathbf{v}_H, b, \xi} \quad & \|\mathbf{v}_H\|^2 + C\sum_{i=1}^{n} \xi_i^2 \\
\text{s.t.} \quad & y_i(\langle \mathbf{v}_H, x_i \rangle - b) \geq 1 - \xi_i \quad \text{for } i = 1, \ldots, n \\
& \xi_i \geq 0, \quad \text{for } i = 1, \ldots, n
\end{aligned}$$

(a) (3 points) For increasing C, will the margin increase or decrease, and why?

Solution:

Errors cost more, so the margin has to decrease to avoid making errors.

(b) (3 points) For increasing C, will $||\mathbf{v}_H||$ increase or decrease, and why?

Solution:

Decreasing margin means increasing $||\mathbf{v}_H||$.

(c) (4 points) For increasing C, will training error increase or decrease, and why?

Solution:

Errors cost more, so training error will go down.

(d) (4 points) For increasing C, should testing error increase or decrease, and why?

Solution:

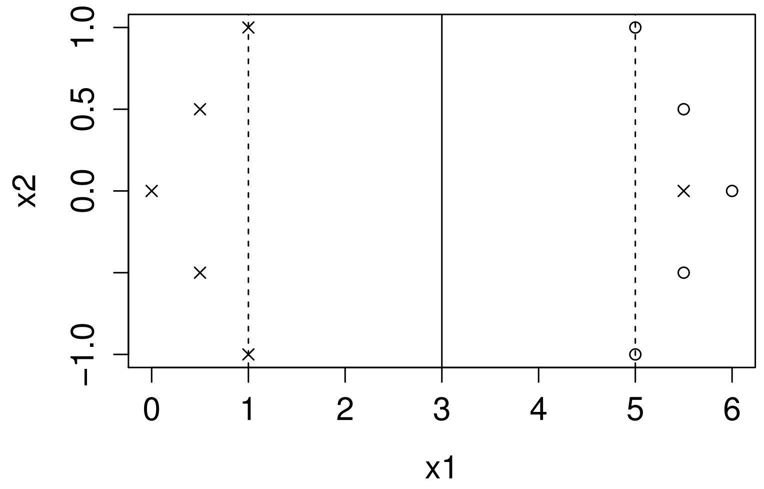
Testing error should increase, as we risk overfitting by being oversensitive to errors.

(e) (7 points) Consider the following training data in $\mathbb{R}^2$. For a large but finite value of C, what will be the training error? (Note: do not try to run an SVM by hand. You should draw the data and argue the answer in a few sentences.)

x_i^1	0.0	1.0	1.0	0.5	0.5	5.0	5.0	6.0	5.5	5.5	5.5
x_i^2	0.0	-1.0	1.0	0.5	-0.5	1.0	-1.0	0.0	-0.5	0.5	0.0
y_i	$+1$	$+1$	$+1$	$+1$	$+1$	-1	-1	-1	-1	-1	$+1$

Solution:

$\frac{1}{11}$. With a drawing, as below.



(f) (4 points) Consider the data in the previous part. For a large but finite value of C, what are the support vectors?

Solution:

Points 2,3,6,7,11. The points 2,3 and 5,6 will be on the margin, and point 11 is misclassified and so is a support vector.

4. (25 points)
Let $\mathcal{G}$ be a convex, differentiable constraint set on $\mathbb{R}^d$, and $f : \mathbb{R}^d \to \mathbb{R}$ be the following convex objective:

$$f(x) = \frac{1}{2}x^\top Px + a^\top x + b.$$

Let the constraint function $g(x)$ be:

$$g(x) = \begin{cases} < 0 & x \in in(\mathcal{G}) \\ = 0 & x \in \partial(\mathcal{G}) \\ > 0 & x \notin \mathcal{G}. \end{cases}$$

Let $x^* = \arg\min_x f(x)$ (unconstrained) and $x^*_{\mathcal{G}}$ be the constrained solution to:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \end{aligned}$$

(a) (3 points) Say $g(x^*) \leq 0$. What is $\nabla f(x^*)$?

Solution:

0.

(b) (3 points) Say $g(x^*) > 0$. What is $\nabla f(x^*)$?

Solution:

0.

(c) (4 points) Using the given form of $f(x)$, what is the update step for a Newton's method in the unconstrained problem?

Solution:

The update step is

$$\begin{aligned} x_{t+1} :=& x_t - \left[\nabla^2 f(x_t)\right]^{-1} \nabla f(x_t) \\ :=& x_t - P^{-1}(Px_t + a) \\ :=& x_t - x_t - P^{-1}a \\ :=& - P^{-1}a \end{aligned}$$

(d) (4 points) What does this answer indicate about the convergence of Newton's method for this particular choice of $f(x)$?

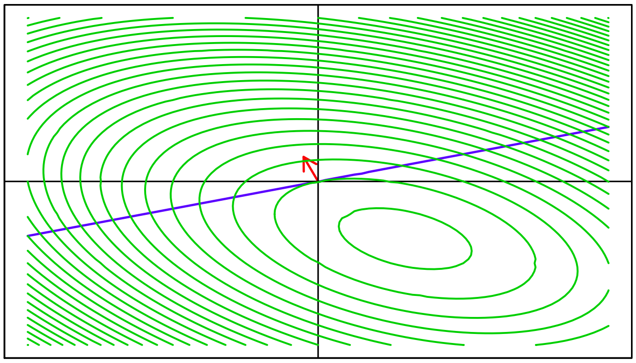
Solution:

The local Hessian approximation is globally exact, and Newton's method will converge in one step. This is expected, as $f(x)$ is a quadratic function.

(e) (4 points) Let $g(x) = |c^\top x|$. Draw the constrained problem if $d = 2$. Include representations of $f(x), g(x), c$ (you need not represent P, a, b explicitly).

Solution:

One acceptable example is the following picture, with
$P = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}, a = \begin{bmatrix} -0.1 \\ 0.3 \end{bmatrix}, b = 0, c = \begin{bmatrix} -0.2 \\ 0.6 \end{bmatrix}$.
The green contours represent $f(x)$, the blue line is $g(x) = 0$, and the small red line is the vector c, which is orthogonal to $g(x) = 0$.



(f) (7 points) Write out the optimality conditions explicitly, and simplify as much as possible. Hint: it is possible to write these conditions as a single matrix-vector solve, which allows us to find $\begin{bmatrix} x^*_{\mathcal{G}} \\ \lambda \end{bmatrix}$ in closed form.

Solution:

It is easiest to notice that $g(x) = |c^\top x|$ implies that $in(\mathcal{G})$ is empty, and thus we have an equality constraint $g(x) = c^\top x = 0$. This fact simplifies the KKT conditions:

$$\begin{aligned} c^\top x^*_{\mathcal{G}} &= 0 \\ \nabla f(x^*_{\mathcal{G}}) + \lambda \nabla g(x^*_{\mathcal{G}}) &= 0 \end{aligned}$$

which in our setting becomes:

$$\begin{aligned} c^\top x^*_{\mathcal{G}} &= 0 \\ P x^*_{\mathcal{G}} + a + \lambda c &= 0. \end{aligned}$$

In a single matrix-vector solve, that looks like:

$$\begin{bmatrix} P & c \\ c^\top & 0 \end{bmatrix} \begin{bmatrix} x^*_{\mathcal{G}} \\ \lambda \end{bmatrix} = \begin{bmatrix} -a \\ 0 \end{bmatrix}.$$