Statistical Machine Learning (W4400)
Spring 2016
https://courseworks.columbia.edu

John P. Cunningham
jpc2181
Ben Reddy, Phyllis Wan, Ashutosh Nanda
bmr2136, pw2348, an2655

MIDTERM EXAM

Total time: 75 minutes. To be taken in-class, Thursday 10 March 2016.

Do not open this exam until instructed. Carefully read the following instructions.

Write your name, UNI, and the course title on the cover of the blue book. 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). No other materials are allowed (including calculators, textbooks, computers, and other electronics). To receive full credit on multi-point problems, you must 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. (12 points) True or False (briefly explain your answer)
   (a) (2 points) The number of nodes in a decision tree can be larger than the number of features in the data used to train that tree.
   (b) (2 points) The number of nodes in a decision tree can be larger than the number of data points used to train that tree.
   (c) (2 points) A classifier trained on more training data is more likely to over fit.
   (d) (2 points) When training a SVM, removing all the training examples that are not support vectors will not change the learned decision boundary.
   (e) (2 points) No classifier can do better than a naive Bayes classifier if the distribution of the data is known.
   (f) (2 points) If a loss function is not twice differentiable (and thus the Hessian is undefined), then that function cannot be convex.

Solution:

(a) T

(b) F

(c) F

(d) T

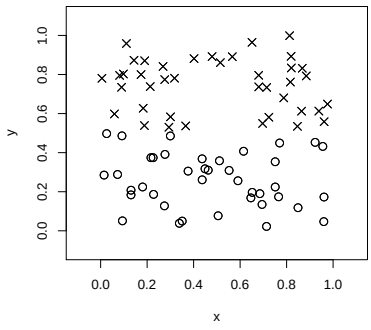
(e) F

(f) F

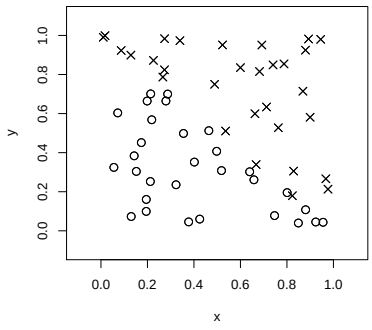
2. (16 points) Decision trees.

(a) (8 points) For each of the following data sets, explain whether or not a basic decision tree of depth 2 will excel in classifying the data. If not, propose a classifier that will.

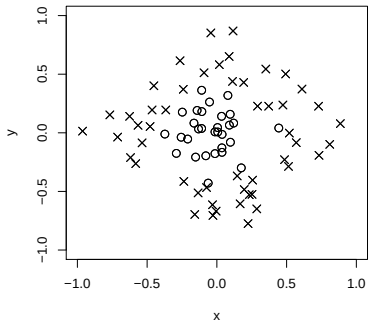
[A]



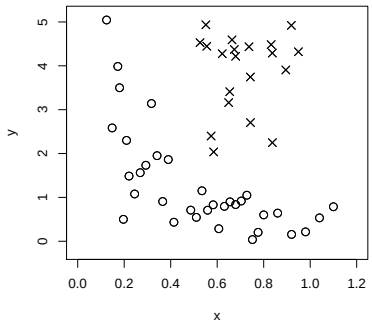
[B]



[C]

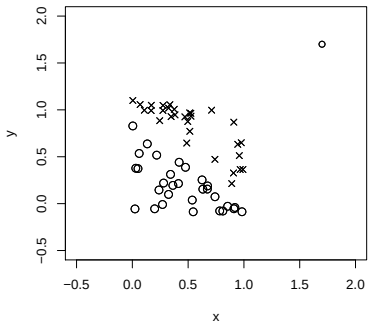


[D]



Solution: A. Yes.
B. No, perceptron.
C. No, boosting with perceptron.
D. Yes.

(b) (4 points) Consider training AdaBoost on the following data set, where the data are linearly separable except for the existence of an outlier. What would happen to the weight assigned to that outlier after many boosting iterations? Would you conclude that AdaBoost is robust to outliers?



Solution: Boosting. Large weight will accumulate on the outlier and steer the classifier towards overfitting. Therefore AdaBoost is not robust to outliers.

(c) (4 points) In AdaBoost, would you stop the iteration if the error rate of the current weak learner on the weighted training data is 0? Explain.

Solution: Yes. In this case, there are no misclassifications and continue training would only result in the same weak classifier. On the other hand, the weight $\alpha_t = +\infty$ for the current classifier, there is no need to combine it with other classifiers.

3. (27 points) Linear regression.

(a) (2 points) Consider a regression problem on $\mathbb{R}$, that is, the regression function is of the form $f : \mathbb{R} \to \mathbb{R}$. Assume f is linear with least squares solution (weight vector) $\beta = [\beta_0 \ \beta_1]^\top = [-1 \ 4]^\top$. What would be the predictions $f(x)$ and $f(x')$ for points $x = 0.8$ and $x' = 1.0$?

Solution: The solution for a point x is

$$f(x) = \beta_0 + \beta_1 x.$$

For the two specific points in the problem that is

$$f(x) = -1 + 4 \cdot 0.8 = 2.2,$$

and

$$f(x') = -1 + 4 \cdot 1.0 = 3.$$

(b) (2 points) Now consider a more general linear regression problem with n data points each in d dimensions, corresponding to the data matrix $X \in \mathbb{R}^{n \times (d+1)}$ and output vector $y \in \mathbb{R}^n$. What is the form of the ordinary least squares solution for the parameters β? What is the dimensionality of β?

Solution:

$$\beta_{OLS} = (X^\top X)^{-1} X^\top y.$$

Also, $\beta \in \mathbb{R}^{d+1}$.

(c) (3 points) Suppose f is actually a quadratic function, with quadratic parameter $\beta_2 = 1$, with the linear terms as above. What are the new predictions for the above points x and x'?

Solution: The solution for a point x is

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2.$$

For the two specific points in the problem that is

$$f(x) = -1 + 4 \cdot 0.8 + 1 \cdot 0.64 = 2.84,$$

and

$$f(x') = -1 + 4 \cdot 1.0 + 1 \cdot 1.0 = 4.$$

(d) (4 points) Now consider a more general quadratic regression problem with n data points each in d dimensions and output vector $y \in \mathbb{R}^n$. Assume the function f has a simple quadratic relationship with each dimension (no product terms between dimensions). What is the form of the ordinary least squares solution for the parameters β? What is the dimensionality of β? Hint: you may wish to define a feature matrix $\Phi = \left[\phi(x_1)...\phi(x_n)\right]^\top$, for some choice of ϕ. Specify the form of the feature map ϕ.

Solution: Let Φ be the augmented data matrix with $2d+1$ dimensions, corresponding to the intercept term, d linear terms, and d simple quadratic terms. Then, we perform least squares on this new feature matrix.

$$\beta_{OLS}^{quad} = (\Phi^\top \Phi)^{-1} \Phi^\top y.$$

Also, $\beta^{quad} \in \mathbb{R}^{2d+1}$.

(e) (2 points) Given these extra quadratic covariates, we are concerned with overfitting and decide to add a ridge penalty, with some given value λ. What is the form of the ridge regression solution for the parameters β? You should again write your solution in terms of Φ. It is not necessary to derive the solution; you can just write it. What is the dimensionality of β?

Solution:

$$\beta_{RR}^{quad} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top y.$$

Again, $\beta^{quad} \in \mathbb{R}^{2d+1}$.

(f) (2 points) The previous solution for β can be written in the form $\beta = \sum_{i=1}^n \alpha_i \phi(x_i) = \Phi^\top \alpha$ (this general fact is often called the *representer theorem*). What are the values α_i? Hint: you may wish to use the identity: $(\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top y = \Phi^\top (\Phi \Phi^\top + \lambda I)^{-1} y$.

Solution: Using the hint, immediately $\alpha_i = \left[(\Phi \Phi^\top + \lambda I)^{-1} y \right]_i$.

(g) (12 points) Combining the above parts, we now derive *kernel* ridge regression. Just as we extended a linear SVM to a nonlinear SVM with the kernel trick, we can do the same to create nonlinear kernel regression. Specifically, find the solution to:

$$\arg\min_\beta \sum_{i=1}^n \left(y_i - \langle \beta, \phi(x_i) \rangle_{\mathcal{F}} \right)^2 + \lambda \|\beta\|_{\mathcal{F}}^2$$

where $\langle \cdot, \cdot \rangle_{\mathcal{F}} = k(\cdot, \cdot)$ in the usual 'kernel trick' way. Then, using that solution, write how to predict $\hat{f}(x^\star)$ at a new point $x^\star$. Your prediction $f(x^\star)$ should be in terms of the kernel matrix $K = \left\{ \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{F}} \right\}_{i,j=1,\ldots,n}$, and elements in $\mathcal{F}$ should not appear explicitly in this prediction (since that could be infinite dimensional). **Hint**: the key step is to use the representer theorem $\beta = \Phi^\top \alpha$, which holds here as it does in the case above.

Solution:

$$
\begin{aligned}
\sum_{i=1}^{n} \left(y_i - \langle \beta, \phi(x_i) \rangle_{\mathcal{F}}\right)^2 + \lambda \|\beta\|_{\mathcal{F}}^2 &= \|y - \left\langle \beta, \Phi^\top \right\rangle\|_2^2 + \lambda \|\beta\|_{\mathcal{F}}^2 \\
&= \|y - \left\langle \Phi^\top \alpha, \Phi^\top \right\rangle\|_2^2 + \lambda \|\Phi^\top \alpha\|_{\mathcal{F}}^2 \\
&= y^\top y - 2\alpha^\top \Phi\Phi^\top y + \alpha^\top \Phi\Phi^\top \Phi\Phi^\top \alpha + \alpha^\top \Phi\Phi^\top \alpha \\
&= y^\top y - 2\alpha^\top K y + \alpha^\top (K^2 + \lambda K)\alpha \\
&\Rightarrow \\
2Ky &= 2K(K + \lambda I)\alpha \\
&\Rightarrow \\
\alpha_{KRR} &= (K + \lambda I)^{-1} y,
\end{aligned}
$$

just as in the previous quadratic case. Then we have $\beta_{KRR} = \Phi^\top \alpha_{KRR}$, and thus:

$$
\begin{aligned}
f(x^\star) &= \langle \beta, \phi(x^\star) \rangle_{\mathcal{F}} \\
&= \left\langle \Phi^\top \alpha, \phi(x^\star) \right\rangle_{\mathcal{F}} \\
&= \begin{bmatrix} k(x^\star, x_1) \\ \vdots \\ k(x^\star, x_n) \end{bmatrix}^\top (K + \lambda I)^{-1} y.
\end{aligned}
$$

4. (15 points) Class imbalance.

   Many datasets have class imbalance: each label is not *a priori* equally likely to occur.

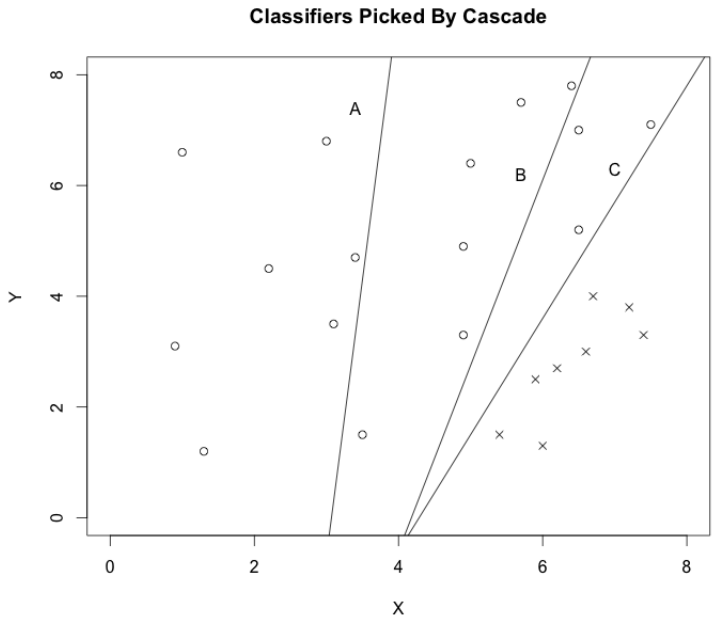
   (a) (5 points) We dealt with class imbalance in face detection; specifically, very few patches of an image typically contain faces, so there are far more examples of non-faces than faces. Name the technique we used to address this issue, and explain (≈ 2 sentences) how it works.

   > *Solution:* The cascade works by building many classifiers in a sequence that correspond with stages of the cascade; the cascade overcomes class imbalance by consecutively trimming the percentage of non-face labels. This is done by taking all examples labeled as a $+1$ in the first stage and then training a classifier on just those examples; this process can be repeated until the until the number of face labels and non-face labels are approximately equal. In classification of new data points, if any of the classifiers think the example is not a face, then the example is marked as not a face; equivalently, a new data point is only marked a face if all classifiers believe the example is a face.

   (b) (5 points) Assume that 99% of training data are labeled as non-faces. Further, assume we have a naive classifier (that did not consider class imbalance) that has misclassification rate of 10%. State and justify a decision rule that would outperform the naive classifier.

   > *Solution:* The current misclassification rate is 10%. To improve upon this, we may simply label all data points as not a face. This would result in a new misclassification rate as 1% because 99% of data points will be labeled correctly.

   (c) (5 points) Consider the given figure below:



   State and justify the order of the classifiers, assuming they were generated by a cascade.

Solution: The order should be A, then B, and then C because each is progressively more aggressive in finding the boundary between the two classes.

5. (30 points) Logistic regression.

A classical statistical method often used for binary classification is logistic regression. For independent observations $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$, with $y_i \in \{0, 1\}$ and $\mathbf{x}_i \in \mathbb{R}^d$, the logistic regression model posits

$$\mathbb{P}[y_i = 1|\mathbf{x}_i] := p(\mathbf{x}_i, \boldsymbol{\beta}) = \frac{e^{\beta_0 + \sum_{j=1}^d \beta_j \mathbf{x}_{ij}}}{1 + e^{\beta_0 + \sum_{j=1}^d \beta_j \mathbf{x}_{ij}}} = \frac{e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}},$$

where $\mathbf{z}_i = \binom{1}{\mathbf{x}_i}$. (Note that this is of the same general class of models as linear regression, where we have $f(\mathbf{z}_i, \boldsymbol{\beta}) = \boldsymbol{\beta}^T \mathbf{z}_i$. In the case of binary valued Y, we pass $\boldsymbol{\beta}^T \mathbf{z}_i$ through the logistic function $\sigma(f) = \frac{e^f}{1+e^f}$, which "squashes" $\boldsymbol{\beta}^T \mathbf{z}_i$ onto the unit interval.)

(a) (10 points) In the simplest case, we don't have covariates $\mathbf{x}_i$ and β_0 is the only parameter to estimate. In this case, what is the maximum likelihood estimator (MLE), $\hat{\beta}_0$? (**Hint:** You might find it simpler to find the MLE $\hat{p}$ for the Bernoulli distribution and then express $\hat{\beta}_0$ in terms of $\hat{p}$.)

Solution: The likelihood is

$$L(y_1, \ldots, y_n) = \prod_{i=1}^n p^{y_i}(1-p)^{1-y_i}$$

The log-likelihood is therefore

$$\ell(y_1, \ldots, y_n) = \sum_{i=1}^n y_i \log p + (1 - y_i) \log(1 - p)$$

Taking the derivative w.r.t. p and setting to zero, we get

$$\begin{aligned} \frac{\partial \ell}{\partial p} &= \sum_{i=1}^n \frac{y_i}{p} - \frac{1 - y_i}{1 - p} = 0 \\ \frac{n\bar{Y}}{p} &= \frac{n(1 - \bar{Y})}{1 - p} \\ \rightarrow \hat{p} &= \bar{Y}. \end{aligned}$$

Now, we have $p = \frac{e^{\beta_0}}{1+e^{\beta_0}}$. Plugging in our solution for $\hat{p}$ and solving for β_0 yields

$$\boxed{\hat{\beta}_0 = \log\left(\frac{\bar{Y}}{1 - \bar{Y}}\right)}$$

(b) (10 points) When covariates $\mathbf{x}_i$ are incorporated, the MLE for each β_j no longer has an analytic form. Show that the MLE $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{d+1}$ satisfies the systems of equations

$$\sum_{i=1}^n \frac{e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}} \mathbf{z}_{ij} = \sum_{i=1}^n y_i \mathbf{z}_{ij}, \quad j = 1, \ldots, d+1$$

Solution: The likelihood in this case is

$$L(y_1, \ldots, y_n) = \prod_{i=1}^{n} \left(\frac{e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}} \right)^{y_i} \left(\frac{1}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}} \right)^{1 - y_i},$$

which gives the log-likelihood

$$\ell(y_1, \ldots, y_n) = \sum_{i=1}^{n} \left(y_i \boldsymbol{\beta}^T \mathbf{z}_i - \log \left(1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i} \right) \right).$$

Taking the derivative w.r.t. β_j, setting equal to zero and rearranging, we have

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \left(y_i \mathbf{z}_{ij} - \frac{e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}} \mathbf{z}_{ij} \right) = 0$$
$$\sum_{i=1}^{n} \frac{e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i}} \mathbf{z}_{ij} = \sum_{i=1}^{n} y_i \mathbf{z}_{ij}$$

(c) (4 points) Describe how the Newton's method proceeds at each step s and write down an expression for the update rule $\boldsymbol{\beta}_{(s-1)} \to \boldsymbol{\beta}_{(s)}$. (**Note:** In the interest of limiting explicit calculations, you do not need to include an explicit expression for the gradient or the Hessian.)

Solution: Starting from some initial value $\boldsymbol{\beta}_{(0)}$, iteratively updates until our specified convergence criteria are met (e.g., $||\boldsymbol{\beta}_{(s)} - \boldsymbol{\beta}_{(s-1)}|| < \epsilon$). The update step is given by

$$\boldsymbol{\beta}_{(s)} \leftarrow \boldsymbol{\beta}_{(s-1)} - \mathbf{H}_{\beta}^{-1}(\boldsymbol{\beta}_{(s-1)}) \cdot \nabla_{\beta} \ell(\boldsymbol{\beta}_{(s-1)}),$$

where $\mathbf{H}_{\beta}^{-1}(\boldsymbol{\beta}_{(s-1)})$ is the inverse of the Hessian matrix evaluated at $\boldsymbol{\beta}_{(s-1)}$ and the gradient $\nabla_{\beta} \ell(\cdot)$ was derived in part (b).

(d) (6 points) The Hessian matrix is given by

$$\begin{aligned} \mathbf{H}_{jk} = \frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} &= \sum_{i=1}^{n} \left(\frac{\mathbf{z}_{ij} \mathbf{z}_{ik} e^{\boldsymbol{\beta}^T \mathbf{z}_i}}{\left(1 + e^{\boldsymbol{\beta}^T \mathbf{z}_i} \right)^2} \right) \\ &= \sum_{i=1}^{n} \mathbf{z}_{ij} \mathbf{z}_{ik} p(\mathbf{x}_i, \boldsymbol{\beta})(1 - p(\mathbf{x}_i, \boldsymbol{\beta})) \\ &= \mathbf{Z}^T \mathbf{V} \mathbf{Z}, \end{aligned}$$

where $\mathbf{V}$ is the diagonal matrix with entries $\mathbf{V}_{ii} = p(\mathbf{z}_i, \boldsymbol{\beta})(1 - p(\mathbf{z}_i, \boldsymbol{\beta}))$, the estimated Bernoulli variance of each observation. Under what conditions on $\mathbf{H}_{\beta}$ would we expect Newton's algorithm to be stable, i.e. it takes controlled steps at each iteration and converges reasonably quickly?

Solution: The update step for Newton's method will be stable when $\mathbf{H}_\beta$ is invertible and "far from singular". The condition for invertibility is that $\mathbf{H}_\beta$ is positive definite (i.e., all of its eigenvalues are positive); the latter condition is met when the spectral condition $c(\mathbf{H}_\beta) = \frac{\lambda_{\min}}{\lambda_{\max}}$ is reasonably large.