10-601 Machine Learning, Midterm Exam

Instructors: Tom Mitchell, Ziv Bar-Joseph

Monday 22nd October, 2012

There are 5 questions, for a total of 100 points.
This exam has 16 pages, make sure you have all pages before you begin.
This exam is open book, open notes, but *no computers or other electronic devices*.

Good luck!

Name: _____

Andrew ID: _____

Question	Points	Score
Short Answers	20	
Comparison of ML algorithms	20	
Regression	20	
Bayes Net	20	
Overfitting and PAC Learning	20	
Total:	100	

Question 1. **Short Answers**

True False Questions.

(a) [1 point] We can get multiple local optimum solutions if we solve a linear regression problem by minimizing the sum of squared errors using gradient descent.
True False

Solution:
False

(b) [1 point] When a decision tree is grown to full depth, it is more likely to fit the noise in the data.
True False

Solution:
True

(c) [1 point] When the hypothesis space is richer, over fitting is more likely.
True False

Solution:
True

(d) [1 point] When the feature space is larger, over fitting is more likely.
True False

Solution:
True

(e) [1 point] We can use gradient descent to learn a Gaussian Mixture Model.
True False

Solution:
True

Short Questions.

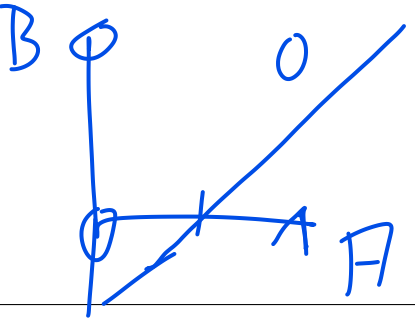
(f) [3 points] Can you represent the following boolean function with a single logistic threshold unit (i.e., a single unit from a neural network)? If yes, show the weights. If not, explain why not in 1-2 sentences.

A	B	f(A,B)
1	1	0
0	0	0
1	0	1
0	1	0

$1 \cdot A \quad -0.5 = B$

$\rightarrow 1 \cdot 1 - 0 - 0.5$

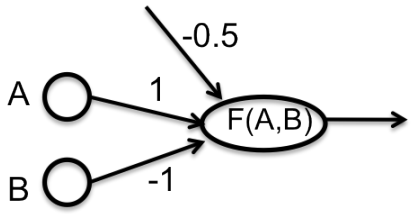
$1 \cdot A - 1 \cdot B - 0.5 = 0$ ← Plane



Solution:

Yes, you can represent this function with a single logistic threshold unit, since it is linearly separable. Here is one example.

$$F(A, B) = 1\{A - B - 0.5 > 0\}$$



(1)

(g) [3 points] Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in the k-means solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences.

Solution:
Yes, k-means assigns each data point to a unique cluster based on its distance to the cluster center. Gaussian mixture clustering gives soft (probabilistic) assignment to each data point. Therefore, even if cluster centers are identical in both methods, if Gaussian mixture components have large variances (components are spread around their center), points on the edges between clusters may be given different assignments in the Gaussian mixture solution.

Circle the correct answer(s).

(h) [3 points] As the number of training examples goes to infinity, your model trained on that data will have:

A. Lower variance B. Higher variance C. Same variance

Solution:
Lower variance

(i) [3 points] As the number of training examples goes to infinity, your model trained on that data will have:

A. Lower bias B. Higher bias C. Same bias

~ but we can increase complexity of model

Solution:
Same bias

(j) [3 points] Suppose you are given an EM algorithm that finds maximum likelihood estimates for a model with latent variables. You are asked to modify the algorithm so that it finds MAP estimates instead. Which step or steps do you need to modify:

A. Expectation B. Maximization C. No modification necessary D. Both

Solution:
Maximization

Question 2. **Comparison of ML algorithms**

Assume we have a set of data from patients who have visited UPMC hospital during the year 2011. A set of features (e.g., temperature, height) have been also extracted for each patient. Our goal is to decide whether a new visiting patient has any of diabetes, heart disease, or Alzheimer (a patient can have one or more of these diseases).

(a) [3 points] We have decided to use a neural network to solve this problem. We have two choices: either to train a *separate* neural network for each of the diseases or to train a single neural network with one output neuron for each disease, but with a shared hidden layer. Which method do you prefer? Justify your answer.

Solution:
1- Neural network with a shared hidden layer can capture dependencies between diseases. It can be shown that in some cases, when there is a dependency between the output nodes, having a shared node in the hidden layer can improve the accuracy.
2- If there is no dependency between diseases (output neurons), then we would prefer to have a separate neural network for each disease.

(b) [3 points] Some patient features are expensive to collect (e.g., brain scans) whereas others are not (e.g., temperature). Therefore, we have decided to first ask our classification algorithm to predict whether a patient has a disease, and if the classifier is 80% confident that the patient has a disease, then we will do additional examinations to collect additional patient features In this case, which classification methods do you recommend: neural networks, decision tree, or naive Bayes? Justify your answer in one or two sentences.

Solution:
We expect students to explain how each of these learning techniques can be used to output a confidence value (any of these techniques can be modified to provide a confidence value). In addition, Naive Bayes is preferable to other cases since we can still use it for classification when the value of some of the features are unknown.
We gave partial credits to those who mentioned neural network because of its non-linear decision boundary, or decision tree since it gives us an interpretable answer.

(c) Assume that we use a logistic regression learning algorithm to train a classifier for each disease. The classifier is trained to obtain MAP estimates for the logistic regression weights W. Our MAP estimator optimizes the objective

$$W \leftarrow \arg\max_W \ln[P(W) \prod_l P(Y^l|X^l, W)]$$

where l refers to the lth training example. We adopt a Gaussian prior with zero mean for the weights $W = \langle w_1 \dots w_n \rangle$, making the above objective equivalent to:

$$W \leftarrow \arg\max_W \ -C \sum_i w_i + \sum_l \ln P(Y^l|X^l, W)$$

Note C here is a constant, and we re-run our learning algorithm with different values of C. Please answer each of these true/false questions, and explain/justify your answer in no more than 2 sentences.

i. [2 points] The average log-probability of the *training data* can never increase as we increase C.
True False

Solution:
True. As we increase C, we give more weight to constraining the predictor. Thus it makes our predictor less flexible to fit to training data (over constraining the predictor, makes it unable to fit to training data).

ii. [2 points] If we start with $C = 0$, the average log-probability of *test data* will likely decrease as we increase C.
True False

Solution:
False. As we increase the value of C (starting from $C = 0$), we avoid our predictor to over fit to training data and thus we expect the accuracy of our predictor to be increased on the test data.

iii. [2 points] If we start with a very large value of C, the average log-probability of *test data* can never decrease as we increase C.
True False

Solution:
False. Similar to the previous parts, if we over constraint the predictor (by choosing very large value of C), then it wouldn't be able to fit to training data and thus makes it to perform worst on the test data.

(d) Decision boundary

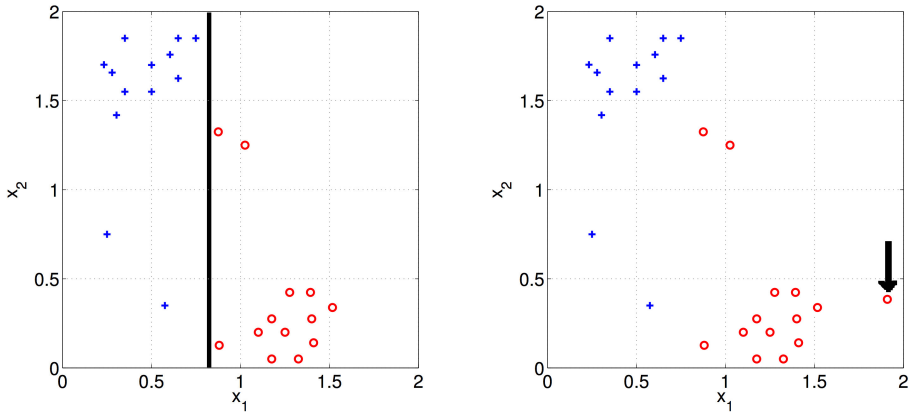


Figure 1: Labeled training set.

i. [2 points] Figure 1(a) illustrates a subset of our training data when we have only two features: X_1 and X_2. Draw the decision boundary for the logistic regression that we explained in part (c).

> **Solution:**
> The decision boundary for logistic regression is linear. One candidate solution which classifies all the data correctly is shown in Figure 1. We will accept other possible solutions since decision boundary depends on the value of C (it is possible for the trained classifier to miss-classify a few of the training data if we choose a large value of C).

ii. [3 points] Now assume that we add a new data point as it is shown in Figure 1(b). How does it change the decision boundary that you drew in Figure 1(a)? Answer this by drawing both the old and the new boundary.

> **Solution:**
> We expect the decision boundary to move a little toward the new data point.

(e) [3 points] Assume that we record information of all the patients who visit UPMC every day. However, for many of these patients we don't know if they have any of the diseases, can we still improve the accuracy of our classifier using these data? If yes, explain how, and if no, justify your answer.

> **Solution:**
> Yes, by using EM. In the class, we showed how EM can improve the accuracy of our classifier using both labeled and unlabeled data. For more details, please look at `http://www.cs.cmu.edu/~tom/10601_fall2012/slides/GrMod3_10_9_2012.pdf`, page 6.

Question 3. **Regression**

Consider real-valued variables X and Y. The Y variable is generated, conditional on X, from the following process:

$$\epsilon \sim N(0, \sigma^2)$$
$$Y = aX + \epsilon$$

where every ϵ is an independent variable, called a *noise* term, which is drawn from a Gaussian distribution with mean 0, and standard deviation σ. This is a one-feature linear regression model, where a is the only weight parameter. The conditional probability of Y has distribution $p(Y|X, a) \sim N(aX, \sigma^2)$, so it can be written as

$$p(Y|X, a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y - aX)^2\right)$$

The following questions are all about this model.

MLE estimation

(a) [3 points] Assume we have a training dataset of n pairs (X_i, Y_i) for $i = 1..n$, and σ is known. Which ones of the following equations correctly represent the maximum likelihood problem for estimating a? Say yes or no to each one. More than one of them should have the answer "yes."

[Solution: *no*] $\arg\max_a \sum_i \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *yes*] $\arg\max_a \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *no*] $\arg\max_a \sum_i \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *yes*] $\arg\max_a \prod_i \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *no*] $\arg\max_a \frac{1}{2}\sum_i (Y_i - aX_i)^2$

[Solution: *yes*] $\arg\min_a \frac{1}{2}\sum_i (Y_i - aX_i)^2$

(b) [7 points] Derive the maximum likelihood estimate of the parameter a in terms of the training example X_i's and Y_i's. We recommend you start with the simplest form of the problem you found above.

Solution:

Use $F(a) = \frac{1}{2}\sum_i (Y_i - aX_i)^2$ and minimize F. Then

$$0 = \frac{\partial}{\partial a}\left[\frac{1}{2}\sum_i (Y_i - aX_i)^2\right] \quad (2)$$

$$= \sum_i (Y_i - aX_i)(-X_i) \quad (3)$$

$$= \sum_i aX_i^2 - X_iY_i \quad (4)$$

$$a = \frac{\sum_i X_iY_i}{\sum_i X_i^2} \quad (5)$$

Partial credit: 1 point for writing a correct objective, 1 point for taking the derivative, 1 point for getting the chain rule correct, 1 point for a reasonable attempt at solving for a. 6 points for correct up to a sign error.

Many people got $\sum y_i / \sum x_i$ as the answer, by erroneously cancelling x_i on top and bottom. 4 points for this answer when it is clear this cancelling caused the problem. If they explicitly derived $\sum x_i y_i / \sum x_i^2$ along the way, 6 points. If it is completely unclear where $\sum y_i / \sum x_i$ came from, sometimes worth only 3 points (based on the partial credit rules above).

Some people wrote a gradient descent rule. We intended to ask for a closed-form maximum likelihood estimate, not an algorithm to get it. (Yes, it is true that lectures never said there exists a closed-form solution for linear regression MLE. But there is. In fact, there is a closed-form solution even for multiple features, via linear algebra.) But we gave 4 points for getting the rule correct; 3 points for correct with a sign error.

For gradient descent/ascent signs are tricky. If you are using the log-likelihood, thus maximization, you want gradient ascent, and thus add the gradient. If instead you're doing the minimization problem, and using gradient descent, need to subtract the gradient. Either way, it comes out to $a \leftarrow a + \eta \sum_i (y_i - ax_i)x_i$. Interpretation: $\sum_i (y_i - ax_i)x_i$ is the correlation of data against the residual. In the case of positive x,y, if the data still correlates with the residual, that means predictions are too low, so you want to increase a.

Here is a lovely book chapter by Tufte (1974) on one-feature linear regression:
`http://www.edwardtufte.com/tufte/dapp/chapter3.html`

MAP estimation

Let's put a prior on a. Assume $a \sim N(0, \lambda^2)$, so

$$p(a|\lambda) = \frac{1}{\sqrt{2\pi}\lambda} \exp(-\frac{1}{2\lambda^2}a^2)$$

The posterior probability of a is

$$p(a \mid Y_1, \ldots Y_n, X_1, \ldots X_n, \lambda) = \frac{p(Y_1, \ldots Y_n | X_1, \ldots X_n, a) p(a|\lambda)}{\int_{a'} p(Y_1, \ldots Y_n | X_1, \ldots X_n, a') p(a'|\lambda) da'}$$

We can ignore the denominator when doing MAP estimation.

(c) [3 points] Under the following conditions, how do the prior and conditional likelihood curves change? Do a^{MLE} and a^{MAP} become closer together, or further apart?

	$p(a\|\lambda)$ prior probability: wider, narrower, or same?	$p(Y_1 \dots Y_n \| X_1 \dots X_n, a)$ conditional likelihood: wider, narrower, or same?	$\|a^{MLE} - a^{MAP}\|$ increase or decrease?
As $\lambda \to \infty$	**[Solution:** wider]	**[Solution:** same]	**[Solution:** decrease]
As $\lambda \to 0$	**[Solution:** narrower]	**[Solution:** same]	**[Solution:** increase]
More data: as $n \to \infty$ (fixed λ)	**[Solution:** same]	**[Solution:** narrower]	**[Solution:** decrease]

(d) [7 points] Assume $\sigma = 1$, and a fixed prior parameter λ. Solve for the MAP estimate of a,

$$\arg\max_a [\ln p(Y_1..Y_n \mid X_1..X_n, a) + \ln p(a|\lambda)]$$

Your solution should be in terms of X_i's, Y_i's, and λ.

Solution:

$$\frac{\partial}{\partial a}[\log p(Y|X,a) + \log p(a|\lambda)] = \frac{\partial \ell}{\partial a} + \frac{\partial \log p(a|\lambda)}{\partial a} \tag{6}$$

To stay sane, let's look at it as maximization, not minimization. (It's easy to get signs wrong by trying to use the squared error minimization form from before.) Since $\sigma = 1$, the log-likelihood and its derivative is

$$\ell(a) = \log\left[\prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right)\right] \tag{7}$$

$$\ell(a) = -\log Z - \frac{1}{2}\sum_i (Y_i - aX_i)^2 \tag{8}$$

$$\frac{\partial \ell}{\partial a} = -\sum_i (Y_i - aX_i)(-X_i) \tag{9}$$

$$= \sum_i (Y_i - aX_i)X_i \tag{10}$$

$$= \sum_i X_i Y_i - aX_i^2 \tag{11}$$

Next get the partial derivative for the log-prior.

$$\frac{\partial \log p(a)}{\partial a} = \frac{\partial}{\partial a}\left[-\log(\sqrt{2\pi}\lambda) - \frac{1}{2\lambda^2}a^2\right] \tag{12}$$

$$= -\frac{a}{\lambda^2} \tag{13}$$

The full partial is the sum of that and the log-likelihood which we did before.

$$0 = \frac{\partial \ell}{\partial a} + \frac{\partial \log p(a)}{\partial a} \tag{14}$$

$$0 = \left(\sum_i X_i Y_i - a X_i^2 \right) - \frac{a}{\lambda^2} \tag{15}$$

$$a = \frac{\sum_i X_i Y_i}{(\sum_i X_i^2) + 1/\lambda^2} \tag{16}$$

Partial credit: 1 point for writing out the log posterior, and/or doing some derivative. 1 point for getting the derivative correct.

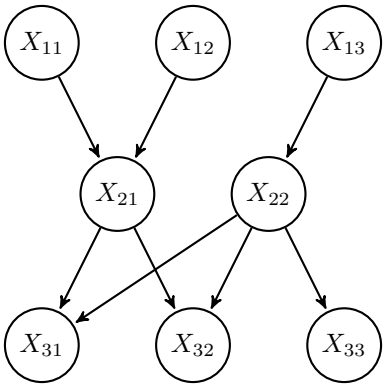
For full solution: deduct a point for a sign error. (There are many potential places for flipping signs). Deduct a point for having n/λ^2: this results from wrapping a sum around the log-prior. (Only the log-likelihood as a $\sum_i$ around it since it's the probability of drawing each data point. The parameter a is drawn only once.)

Some people didn't set $\sigma = 1$ and kept σ to the end. We simply gave credit if substituting $\sigma = 1$ gave the right answer; a few people may have derived the wrong answer but we didn't carefully check all these cases.

People who did gradient descent rules were graded similarly as before: 4 points if correct, deduct one for sign error.

Question 4. **Bayes Net**

Consider a Bayesian network B with boolean variables.



(a) [2 points] From the rule we covered in lecture, is there any variable(s) conditionally independent of X_{33} given X_{11} and X_{12}? If so, list all.

Solution:
X_{21}

(b) [2 points] From the rule we covered in lecture, is there any variable(s) conditionally independent of X_{33} given X_{22}? If so, list all.

Solution:
Everything but X_{22}, X_{33}.

(c) [3 points] Write the joint probability $P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?

Solution:
$P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$
$= P(X_{11})P(X_{12})P(X_{13})P(X_{21}|X_{11}, X_{12})P(X_{22}|X_{13})P(X_{31}|X_{21}X_{22})P(X_{32}|X_{21}X_{22})P(X_{33}|X_{22})$
9 parameters are necessary.

(d) [2 points] Write an expression for $P(X_{13} = 0, X_{22} = 1, X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part **(c)**. Show your work.

Solution:
$P(X_{13} = 0)P(X_{22} = 1|X_{13} = 0)P(X_{33} = 0|X_{22} = 1)$

(e) [3 points] From your answer to **(d)**, can you say X_{13} and X_{33} are independent? Why?

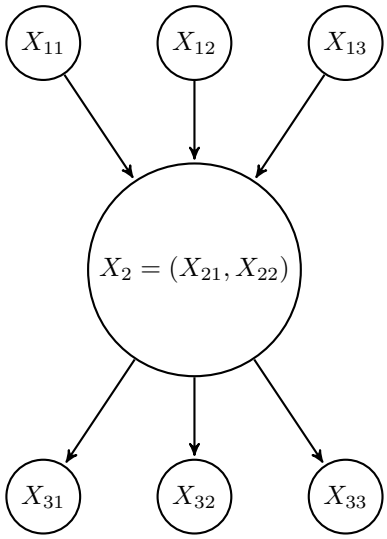
Solution:
No. Conditional independence doesn't imply marginal independence.

(f) [3 points] Can you say the same thing when $X_{22} = 1$? In other words, can you say X_{13} and X_{33} are independent given $X_{22} = 1$? Why?

Solution:
Yes. X_{22} is the only parent of X_{33} and X_{13} is a nondescendant of X_{33}, so by the rule in the lecture we can say they are independent given $X_{22} = 1$

(g) [2 points] Replace $X21$ and $X22$ by a single new variable $X2$ whose value is a pair of boolean values, defined as: $X2 = \langle X21, X22 \rangle$. Draw the new Bayes net B' after the change.

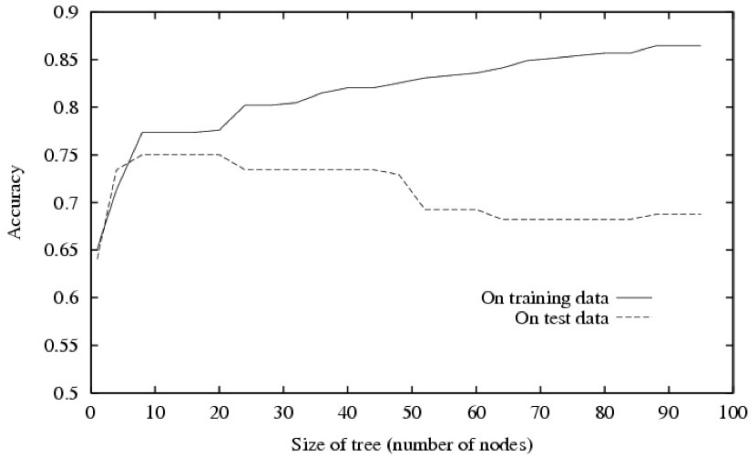
Solution:



(h) [3 points] Do all the conditional independences in B hold in the new network B'? If not, write one that is true in B but not in B'. Consider only the variables present in both B and B'.

Solution:
No. For instance, X_{32} is not conditionally independnt of X_{33} given X_{22} anymore.
* Note: We noticed the problem description was a bit ambiguous, so we also accepted yes as a correct answer

Question 5. **Overfitting and PAC Learning**



(a) Consider the training set accuracy and test set accuracy curves plotted above, during decision tree learning, as the number of nodes in the decision tree grows. This decision tree is being used to learn a function $f : X \rightarrow Y$, where training and test set examples are drawn independently at random from an underlying distribution $P(X)$, after which the trainer provides a noise-free label Y. Note error = 1 - accuracy. Please answer each of these true/false questions, and explain/justify your answer in *1 or 2 sentences.*

i. [2 points] T or F: Training error at each point on this curve provides an unbiased estimate of true error.

Solution:
False. Training error is an optimistically biased estimate of true error, because the hypothesis was chosen based on its fit to the training data.

ii. [1 point] T or F: Test error at each point on this curve provides an unbiased estimate of true error.

Solution:
True. The expected value of test error (taken over different draws of random test sets) is equal to true error.

iii. [1 point] T of F: Training accuracy minus test accuracy provides an unbiased estimate of the degree of overfitting.

Solution:
True. We defined overfitting as test error minus training error, which is equal to training accuracy minus test accuracy.

iv. [1 point] T or F: Each time we draw a different test set from $P(X)$ the test accuracy curve may vary from what we see here.

Solution:
True. Of course each random draw from $P(X)$ may vary from another draw.

v. [1 point] T or F: The variance in test accuracy will increase as we increase the number of test examples.

Solution:
False. The variance in test accuracy will *decrease* as we increase the size of the test set.

(b) Short answers.

i. [2 points] Given the above plot of training and test accuracy, which size decision tree would you choose to use to classify future examples? Give a one-sentence justification.

Solution:
The tree with 10 nodes. This has the highest test accuracy of any of the trees, and hence the highest expected true accuracy.

ii. [2 points] What is the amount of overfitting in the tree you selected?

Solution:
overfitting = training accuracy minus test accuracy = 0.77 - 0.74 = 0.03

Let us consider the above plot of training and test error from the perspective of agnostic PAC bounds. Consider the agnostic PAC bound we discussed in class:

$$m \geq \frac{1}{2\epsilon^2}(\ln |H| + \ln(1/\delta))$$

where ϵ is defined to be the difference between $error_{true}(h)$ and $error_{train}(h)$ for any hypothesis h output by the learner.

iii. [2 points] State in one carefully worded sentence what the above PAC bound guarantees about the two curves in our decision tree plot above.

Solution:
If we train on m examples drawn at random from $P(X)$, then with probability $(1 - \delta)$ the overfitting (difference between training and true accuracy) for each hypothesis in the plot will be less than or equal to ϵ. Note the the true accuracy is the expected value of the test accuracy, taken over different randomly drawn test sets.

iv. [2 points] Assume we used 200 training examples to produce the above decision tree plot. If we wish to reduce the overfitting to half of what we observe there, how many training examples would you suggest we use? Justify your answer in terms of the agnostic PAC bound, in *no more than two sentences*.

Solution:
The bound shows that m grows as $\frac{1}{2\epsilon^2}$. Therefore if we wish to halve ϵ, it will suffice to increase m by a factor of 4. We should use $200 \times 4 = 800$ training examples.

v. [2 points] Give a one sentence explanation of why you are not certain that your recommended number of training examples will reduce overfitting by exactly one half.

Solution:
There are several reasons, including the following. 1. Our PAC theory result gives a bound, not an equality, so 800 examples might decrease overfitting by more than half. 2. The "observed" overfitting is actually the test set accuracy, which is only an estimate of true accuracy, so it may vary from true accuracy and our "observed" overfitting will vary accordingly.

(c) You decide to estimate of the probability θ that a particular coin will turn up heads, by flipping it 10 times. You notice that if repeat this experiment, each time obtaining as new set of 10 coin flips, you get different resulting estimates. You repeat the experiment $N = 20$ times, obtaining estimates $\hat{\theta}^1, \hat{\theta}^2 \ldots \hat{\theta}^{20}$. You calculate the variance in these estimates as

$$var = \frac{1}{N} \sum_{i=1}^{i=N} (\hat{\theta}^i - \theta^{mean})^2$$

where θ^{mean} is the mean of your estimates $\hat{\theta}^1, \hat{\theta}^2 \ldots \hat{\theta}^{20}$.

i. [4 points] Which do you expect to produce a smaller value for var: a Maximum likelihood estimator (MLE), or a Maximum a posteriori (MAP) estimator that uses a Beta prior? Assume both estimators are given the same data. Justify your answer in one sentence.

Solution:
We should expect the MAP estimate to produce a smaller value for var, because using the Beta prior is equivalent to adding in a fixed set of "hallucinated" training examples that will *not* vary from experiment to experiment.

Midterm Exam Solutions

CMU 10-601: Machine Learning (Spring 2016)

Feb. 29, 2016

Name: _____

Andrew ID: _____

START HERE: Instructions

- This exam has 17 pages and 5 Questions (page one is this cover page). Check to see if any pages are missing. Enter your name and Andrew ID above.
- You are allowed to use one page of notes, front and back.
- Electronic devices are not acceptable.
- Note that the questions vary in difficulty. Make sure to look over the entire exam before you start and answer the easier questions first.

Question	Point	Score
1	20	
2	20	
3	20	
4	20	
5	20	
Extra Credit	14	
Total	114	

1 Naive Bayes, Probability, and MLE [20 pts. + 2 Extra Credit]

1.1 Naive Bayes

You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

- sex $\in$ {male,female}
- height $\in$ [0,300] centimeters
- hair $\in$ {brown, black, blond, red, green}
- 3240 men in the data set
- 6760 women in the data set

Under the assumptions necessary for Naive Bayes (not the distributional assumptions you might naturally or intuitively make about the dataset) answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** As height is a continuous valued variable, Naive Bayes is not appropriate since it cannot handle continuous valued variables.

Solution: False. Naive Bayes can handle both continuous and discrete values as long as the appropriate distributions are used for conditional probabilities. For example, Gaussian for continuous and Bernoulli for discrete

(b) [2 pts.] **T or F:** Since there is not a similar number of men and women in the dataset, Naive Bayes will have high test error.

Solution: False. Since the data was randomly split, the same proportion of male and female will be in the training and testing sets. Thus this discrepancy will not affect testing error.

(c) [2 pts.] **T or F:** $P(\texttt{height}|\texttt{sex},\texttt{hair}) = P(\texttt{height}|\texttt{sex})$.

Solution: True. This results from the conditional independence assumption required for Naive Bayes.

(d) [2 pts.] **T or F:** $P(\texttt{height},\texttt{hair}|\texttt{sex}) = P(\texttt{height}|\texttt{sex})P(\texttt{hair}|\texttt{sex})$.

Solution: True. This results from the conditional independence assumption required for Naive Bayes.

1.2 Maximum Likelihood Estimation (MLE)

Assume we have a random sample that is Bernoulli distributed $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$. We are going to derive the MLE for θ. Recall that a Bernoulli random variable X takes values in $\{0, 1\}$ and has probability mass function given by

$$P(X; \theta) = \theta^X (1 - \theta)^{1-X}.$$

(a) [2 pts.] Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$.

Solution:

$$\begin{aligned} L(\theta; X_1, \ldots, X_n) &= \prod_{i=1}^{n} p(X_i; \theta) \\ &= \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} \\ &= \theta^{\sum_{i=1}^n x_i}(1-\theta)^{n-\sum_{i=1}^n x_i}. \end{aligned}$$

Either of the final two steps are acceptable.

(b) [2 pts.] Derive the following formula for the log likelihood:

$$\ell(\theta; X_1, \ldots, X_n) = \left(\sum_{i=1}^{n} X_i\right) \log(\theta) + \left(n - \sum_{i=1}^{n} X_i\right) \log(1-\theta).$$

Solution:

$$\begin{aligned} l(\theta; X_1, \ldots, X_n) &= \log L(\theta; X_1, \ldots, X_n) \\ &= \log\left[\theta^{\sum_{i=1}^n x_i}(1-\theta)^{n-\sum_{i=1}^n x_i}\right] \\ &= (\sum_{i=1}^{n} x_i) \log(\theta) + (n - \sum_{i=1}^{n} x_i) \log(1-\theta) \end{aligned}$$

(c) **Extra Credit:** [2 pts.] Derive the following formula for the MLE: $\hat{\theta} = \frac{1}{n}(\sum_{i=1}^n X_i)$.

Solution: To find the MLE we solve $\frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) = 0$. The derivative is given by

$$\begin{aligned} \frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) &= \frac{d}{d\theta}\left[(\sum_{i=1}^{n} x_i) \log(\theta) + (n - \sum_{i=1}^{n} x_i) \log(1-\theta)\right] \\ &= \frac{\sum_{i=1}^n x_i}{\theta} - \frac{n - \sum_{i=1}^n x_i}{1-\theta} \end{aligned}$$

Next, we solve $\frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) = 0$:

$$\begin{aligned} &\frac{\sum_{i=1}^n x_i}{\theta} - \frac{n - \sum_{i=1}^n x_i}{1-\theta} = 0 \\ \iff &\left(\sum_{i=1}^{n} x_i\right)(1-\theta) - \left(n - \sum_{i=1}^{n} x_i\right)\theta = 0 \\ \iff &\sum_{i=1}^{n} x_i - n\theta = 0 \\ \iff &\hat{\theta} = \frac{1}{n}(\sum_{i=1}^{n} X_i). \end{aligned}$$

1.3 MAP vs MLE

Answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** In the limit, as n (the number of samples) increases, the MAP and MLE estimates become the same.

Solution: True. As the number of examples increases, the data likelihood goes to zero very quickly, while the magnitude of the prior stays the same. In the limit, the prior plays an insignificant role in the MAP estimate and the two estimates will converge.

(b) [2 pts.] **T or F:** Naive Bayes can only be used with MAP estimates, and not MLE estimates.

Solution: False. In Naive Bayes we need to estimate the distribution of each feature X_i given the label Y. Any technique for estimating the distribution can be used, including both the MLE and the MAP estimate.

1.4 Probability

Assume we have a sample space Ω. Answer each question with **T** or **F**. **No justification is required.**

(a) [1 pts.] **T or F:** If events A, B, and C are disjoint then they are independent.

Solution: False. If they are disjoint, i.e. mutually exclusive, they are very dependent!

(b) [1 pts.] **T or F:** $P(A|B) \propto \frac{P(A)P(B|A)}{P(A|B)}$. (The sign '$\propto$' means 'is proportional to')

Solution: False. $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$

(c) [1 pts.] **T or F:** $P(A \cup B) \leq P(A)$.

Solution: False. $P(A \cup B) \geq P(A)$

(d) [1 pts.] **T or F:** $P(A \cap B) \geq P(A)$.

Solution: False. $P(A \cap B) \leq P(A)$

2 Errors, Errors Everywhere [20 pts.]

2.1 True Errors

Consider a classification problem on $\mathbb{R}^d$ with distribution D and target function $c^* : \mathbb{R}^d \to \{\pm 1\}$. Let S be an iid sample drawn from the distribution D. Answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [4 pts.] **T or F:** The true error of a hypothesis h can be lower than its training error on the sample S.

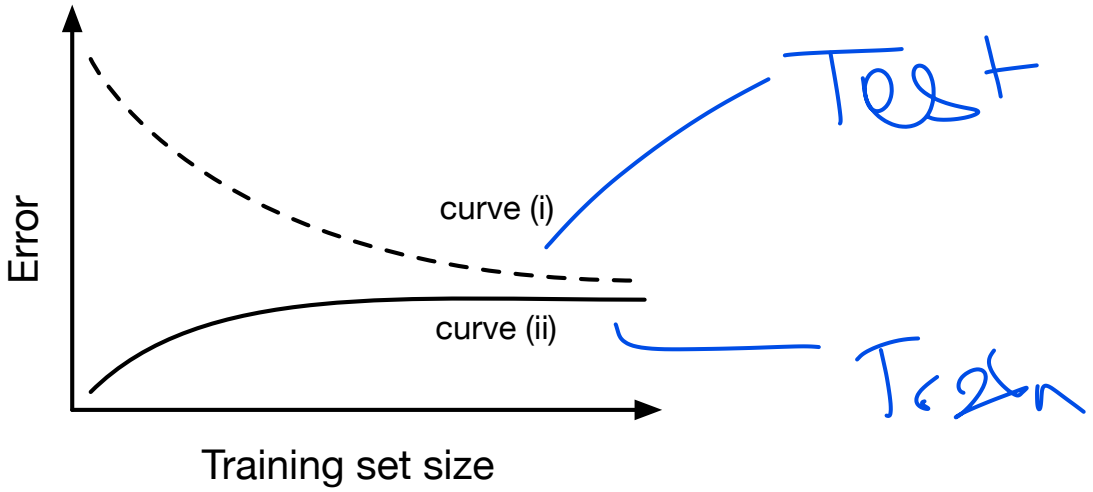
Solution: True. If the sample S happens to favor part of the space where h makes mistakes then the sample error will overestimate the true error. An extreme example is when the hypothesis h has true error 0.5, but the training sample S contains a single sample that h incorrectly classifies.

(b) [4 pts.] **T or F:** Learning theory allows us to determine with 100% certainty the true error of a hypothesis to within any $\epsilon > 0$ error.

Solution: False. There is always a small chance that the sample S is not representative of the underlying distribution D, in which case the sample S may have no relationship to the true error. The sample complexity bounds discussed in class show that this is rare, but not impossible.

2.2 Training Sample Size

In this problem, we will consider the effect of training sample size n on a logistic regression classifier with d features. The classifier is trained by optimizing the conditional log-likelihood. The optimization procedure stops if the estimated parameters perfectly classify the training data or they converge. The following plot shows the general trend for how the training and testing error change as we increase the sample size $n = |S|$. Your task in this question is to analyze this plot and identify which curve corresponds to the training and test error. Specifically:



(a) [8 pts.] Which curve represents the training error? **Please provide 1–2 sentences of justification**.

Solution: It is easier to correctly classify small training datasets. For example, if the data contains just a single point, then logistic regression will always have zero training error. On the other hand, we don't expect a classifier learned from few examples to generalize well, so for small training sets the true error is large. Therefore, curve (ii) shows the general trend of the training error.

(b) [4 pt.] In one word, what does the gap between the two curves represent?

Solution: The gap between the two curves represents the amount of overfitting.

3 Linear and Logistic Regression [20 pts. + 2 Extra Credit]

3.1 Linear regression

Given that we have an input x and we want to estimate an output y, in linear regression we assume the relationship between them is of the form $y = wx + b + \epsilon$, where w and b are real-valued parameters we estimate and ϵ represents the noise in the data. When the noise is Gaussian, maximizing the likelihood of a dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to estimate the parameters w and b is equivalent to minimizing the squared error:

$$\arg\min_{w} \sum_{i=1}^{n} (y_i - (wx_i + b))^2.$$

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line	(b)	(c)	(b)	(a)	(a)

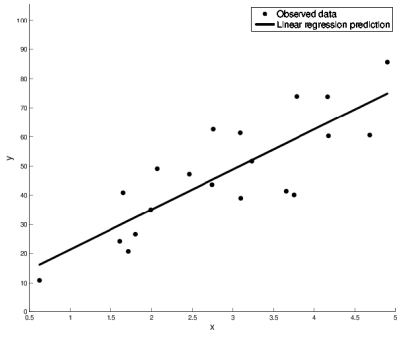
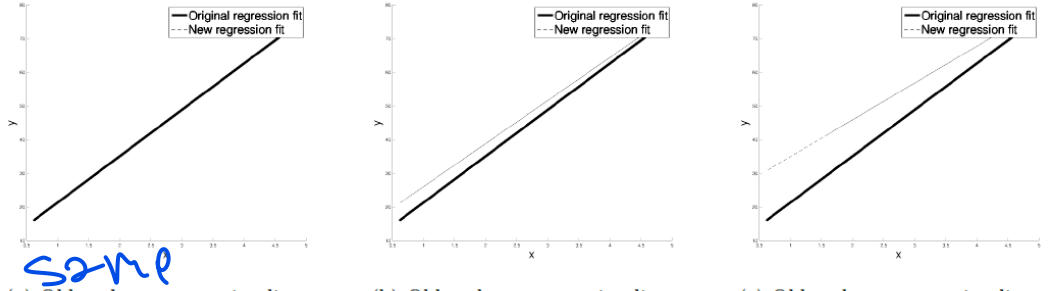
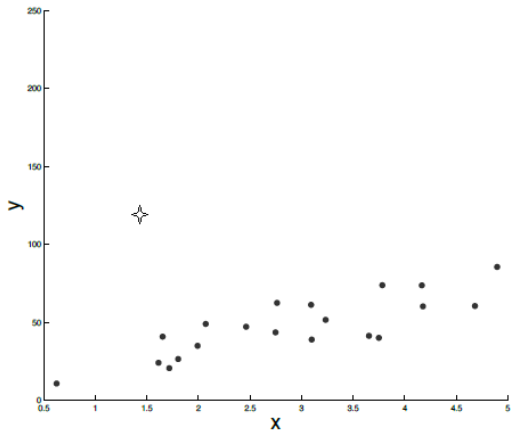


Figure 1: An observed data set and its associated regression line.

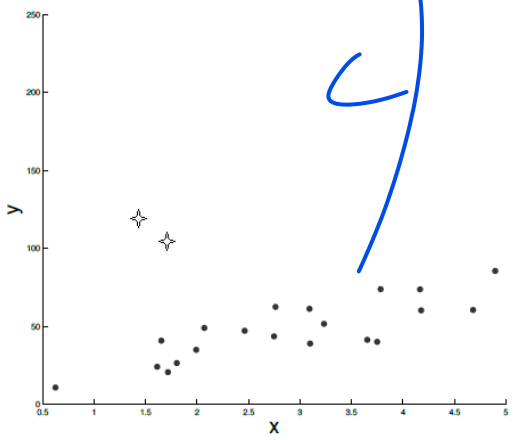


(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

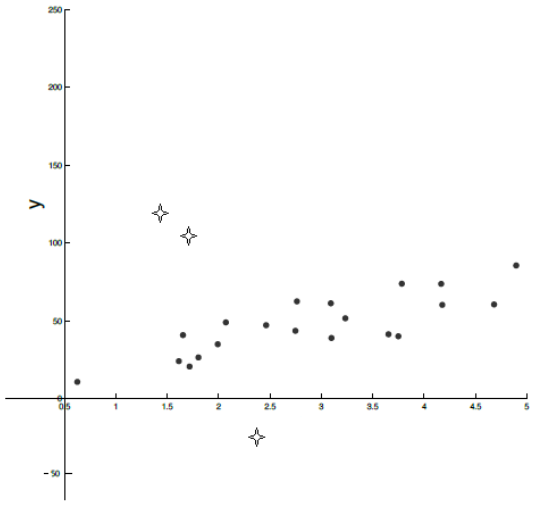
Figure 2: New regression lines for altered data sets S^{new}.



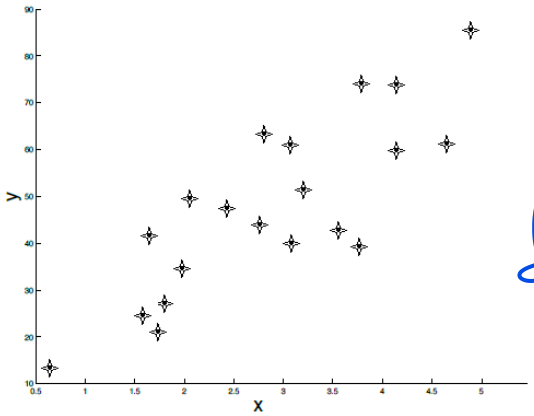
(a) Adding one outlier to the original data set.



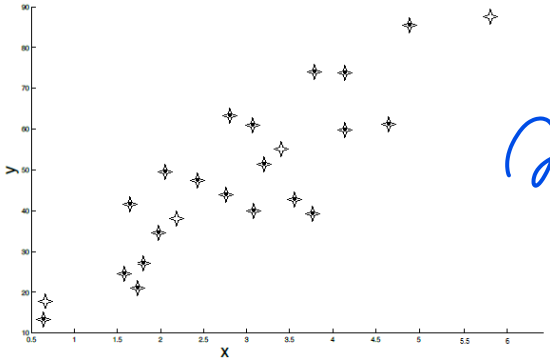
(b) Adding two outliers to the original data set.



(c) Adding three outliers to the original data set. Two on one side and one on the other side.



(d) Duplicating the original data set.



(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

Figure 3: New data set S^{new}.

3.2 Logistic regression

Given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$ where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label, we want to find the parameters $\hat{w}$ that maximize the likelihood for the training set, assuming a parametric model of the form

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.$$

The conditional log likelihood of the training set is

$$\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i, |x_i; w) + (1 - y_i) \log(1 - p(y_i, |x_i; w)),$$

and the gradient is

$$\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w))x_i.$$

(a) [5 pts.] Is it possible to get a closed form for the parameters $\hat{w}$ that maximize the conditional log likelihood? How would you compute $\hat{w}$ in practice?

Solution: There is no closed form expression for maximizing the conditional log likelihood. One has to consider iterative optimization methods, such as gradient descent, to compute $\hat{w}$.

(b) [5 pts.] What is the form of the classifier output by logistic regression?

Solution: Given x, we predict $\hat{y} = 1$, if $p(y = 1|x) \geq p(y = 0|x)$. This is reduces to $\hat{y} = 1$, if $w^T x \geq 0$, which is a linear classifier.

(c) [2 pts.] **Extra Credit:** Consider the case with binary features, i.e, $x \in \{0, 1\}^d \subset \mathbb{R}^d$, where feature x_1 is rare and happens to appear in the training set with only label 1. What is $\hat{w}_1$? Is the gradient ever zero for any finite w? Why is it important to include a regularization term to control the norm of $\hat{w}$?

Solution: If a binary feature was active for only label 1 in the training set then, by maximizing the conditional log likelihood, we will make the weight associated to that feature be infinite. This is because, when this feature is observed in the training set, we will want to predict predict 1 irrespective of everything else. This is an undesired behaviour from the point of view of generalization performance, as most likely we do not believe this rare feature to have that much information about class 1. Most likely, it is spurious co-occurrence. Controlling the norm of the weight vector will prevent these pathological cases.

4 SVM, Perceptron and Kernels [20 pts. + 4 Extra Credit]

4.1 True or False

Answer each of the following questions with **T** or **F** and **provide a one line justification**.

(a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), ..., (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$.

Solution: False. The maximum number of mistakes made by a perceptron is dependent on the margin and radius of the training data, not its dimension or size. The maximum mistake a perceptron will make is $(\frac{R}{\gamma})^2$.

(b) [2 pts.] Suppose $\phi(\mathbf{x})$ is an arbitrary feature mapping from input $\mathbf{x} \in \mathcal{X}$ to $\phi(\mathbf{x}) \in \mathbb{R}^N$ and let $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$. Then $K(\mathbf{x}, \mathbf{z})$ will always be a valid kernel function.

Solution: True. K is a kernel if it is an inner product after applying some feature transformation.

(c) [2 pts.] Given the same training data, in which the points are linearly separable, the margin of the decision boundary produced by SVM will always be greater than or equal to the margin of the decision boundary produced by Perceptron.

Solution: True. SVM explicitly maximizes margin; Perceptron does not differentiate between decision boundaries as long as they lie within the margin of the training data.

4.2 Multiple Choice

(a) [3 pt.] If the data is linearly separable, SVM minimizes $\|w\|^2$ subject to the constraints $\forall i, y_i w \cdot x_i \geq 1$. In the linearly separable case, which of the following may happen to the decision boundary if one of the training samples is removed? **Circle all that apply.**

- Shifts toward the point removed Yes
- Shifts away from the point removed No
- Does not change Yes

(b) [3 pt.] Recall that when the data are not linearly separable, SVM minimizes $\|w\|^2 + C \sum_i \xi_i$ subject to the constraint that $\forall i$, $y_i w \cdot x_i \geq 1 - \xi_i$ and $\xi_i \geq 0$. Which of the following may happen to the size of the margin if the tradeoff parameter C is increased? **Circle all that apply.**

- Increases No
- Decreases Yes

- Remains the same Yes

Proof of part (b):

Let $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$ and let $w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$. Let $c_1 > c_0$. We need $\|w_1^*\|^2 \geq \|w_0^*\|^2$. Define $\xi_{i(0)}$ to be the slack variables under w_0^* and $\xi_{i(1)}$ to be the slack variables under w_1^*.

We first show that for any $\|w'\|^2 < \|w_0^*\|^2$, $\sum_i \xi_i' > \sum_i \xi_{i(0)}$ where ξ_i' are the slack variables under w'.

By contradiction, assume $\|w'\|^2 < \|w_0^*\|^2$ and $\sum_i \xi_i' \leq \sum_i \xi_{i(0)}$. Then, $\|w'\|^2 + c_0 \sum_i \xi_i' < \|w_0^*\|^2 + c_0 \sum_i \xi_{i(0)}$ and $w_0^* \neq \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$.

Thus $\forall \|w'\|^2 \leq \|w_0^*\|^2$, $\sum_i \xi_i' \geq \sum_i \xi_{i(0)}$.

Next, we show that if $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$ and $w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$, then $\|w_1^*\|^2 \geq \|w_0^*\|^2$.

By contradiction, assume $\|w_1^*\|^2 < \|w_0^*\|^2$. Since $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$:

$$\|w_0^*\|^2 + c_0 \sum_i \xi_{i(0)} \leq \|w_1^*\|^2 + c_0 \sum_i \xi_{i(1)}$$

Since $c_1 > c_0$ and $\sum_i \xi_{i(1)} > \sum_i \xi_{i(0)}$, then

$$\|w_0^*\|^2 + c_1 \sum_i \xi_{i(0)} < \|w_1^*\|^2 + c_1 \sum_i \xi_{i(1)}$$

But,

$$w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$$

This yields a contradiction. Thus $\|w_1^*\|^2 \geq \|w_0^*\|^2$.

4.3 Analysis

(a) [4 pts.] In one or two sentences, describe the benefit of using the Kernel trick.

Solution: Allows mapping features into higher dimensional space but avoids the extra computational costs of mapping into higher dimensional feature space explicitly.

(b) [4 pt.] The concept of margin is essential in both SVM and Perceptron. Describe why a large margin separator is desirable for classification.
Solution: Separators with large margin will have low generalization errors with high probability.

(c) [4 pts.] **Extra Credit:** Consider the dataset in Fig. 4. Under the SVM formulation in section 4.2(a),

(1) Draw the decision boundary on the graph.

(2) What is the size of the margin?

(3) Circle all the support vectors on the graph.

Solution: $x_2 - 2.5 = 0$. The size of margin is 0.5. Support vectors are x_2, x_3, x_6, x_7.

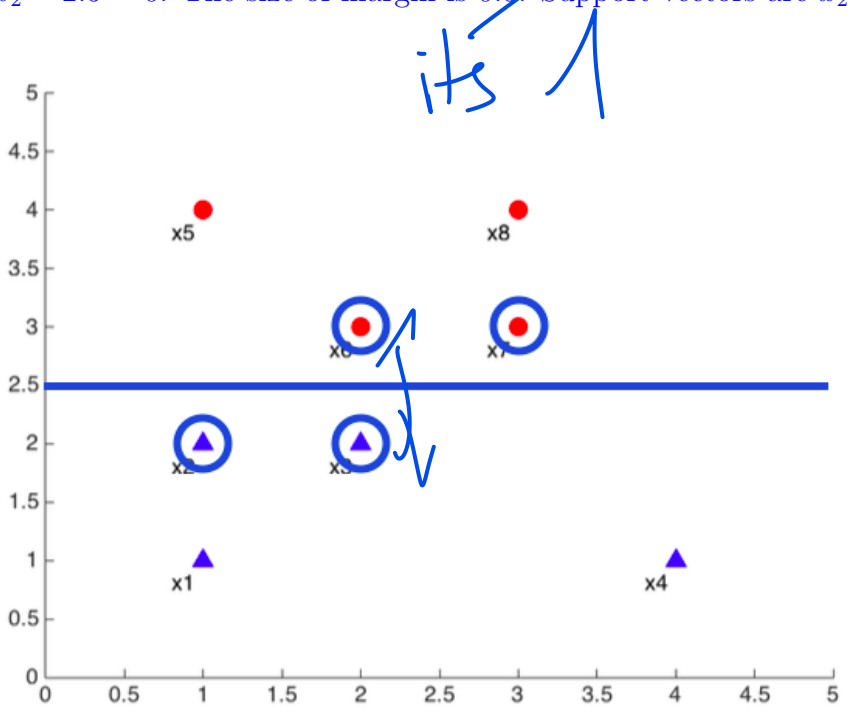


Figure 4: SVM toy dataset

5 Learning Theory [20 pts.]

5.1 True or False

Answer each of the following questions with **T** or **F** and **provide a one line justification.**

(a) [3 pts.] **T or F**: It is possible to label 4 points in $\mathbb{R}^2$ in all possible 2^4 ways via linear separators in $\mathbb{R}^2$.

Solution: F. The VC dimension of linear separator in $\mathbb{R}^2$ is 3, hence it cannot shatter a set of size 4 in all possible ways.

(b) [3 pts.] **T or F**: To show that the VC-dimension of a concept class H (containing functions from X to $\{0, 1\}$) is d, it is sufficient to show that there exists a subset of X with size d that can be labeled by H in all possible 2^d ways.

Solution: F. This is only a necessary condition. We also need to show that no subset of X with size $d + 1$ can be shattered by H.

(c) [3 pts.] **T or F**: The VC dimension of a finite concept class H is upper bounded by $\lceil \log_2 |H| \rceil$.

Solution: T. For any finite set S, if H shatters S, then H at least needs to have $2^{|S|}$ elements, which implies $|S| \leq \lceil \log_2 |H| \rceil$.

(d) [3 pts.] **T or F**: The VC dimension of a concept class with infinite size is also infinite.

Solution: F. Consider all the half-spaces in $\mathbb{R}^2$, which has infinite cardinality but the VC dimension is 3.

(e) [3 pts.] **T or F**: For every pair of classes, H_1, H_2, if $H_1 \subseteq H_2$ and $H_1 \neq H_2$, then $\text{VCdim}(H_1) < \text{VCdim}(H_2)$ (note that this is a strict inequality).

Solution: F. Let H_1 be the collection of all the half-spaces in $\mathbb{R}^2$ with finite slopes and let H_2 be the collection of all the half-spaces in $\mathbb{R}^2$. Clearly $H_1 \subseteq H_2$ and $H_1 \neq H_2$, but $VC(H_1) = VC(H_2) = 3$.

(f) [3 pts.] **T or F**: Given a realizable concept class and a set of training instances, a consistent learner will output a concept that achieves 0 error on the training instances.

Solution: T. Since the concept class is realizable, then the consistent learner can output the oracle labeler by definition, which is guaranteed to achieve 0 error on the training set.

5.2 VC dimension

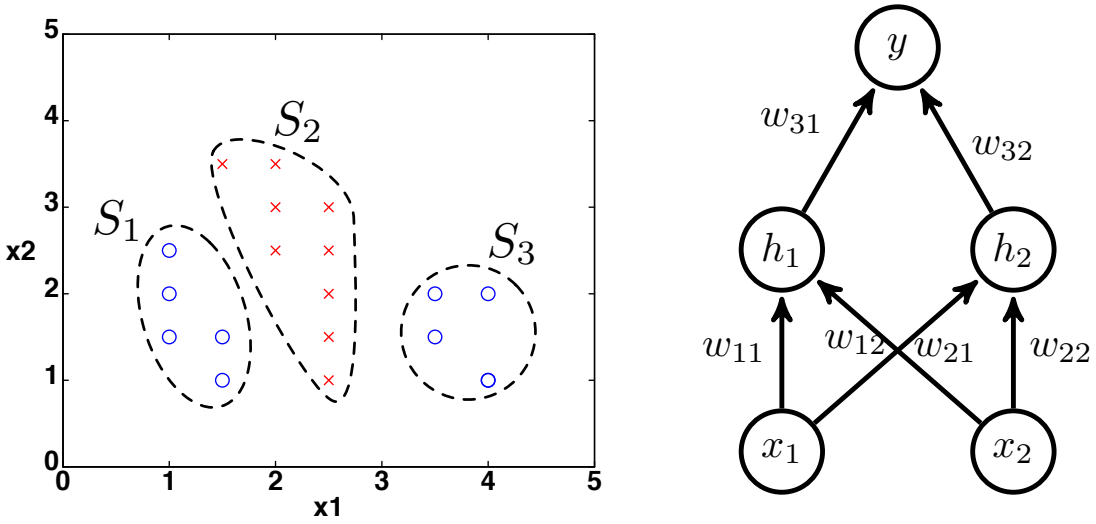
Briefly explain **in 2–3 sentences** the importance of sample complexity and VC dimension in learning with generalization guarantees.

Solution: Sample complexity guarantees quantify how many training samples we need to see from the underlying data distribution D in order to guarantee that uniformly for all hypotheses in the class of functions under consideration we have that their empirical error rates are close to their true errors. This is important because we care abut finding a hypothesis of small true error, but we can only optimize over a fixed training sample. VC bounds are one kind of sample complexity guarantee, where the bound depends on the VC-dimension of the hypothesis class, and they are particularly useful when the class of functions is infinite.

6 Extra Credit: Neural Networks [6 pts.]

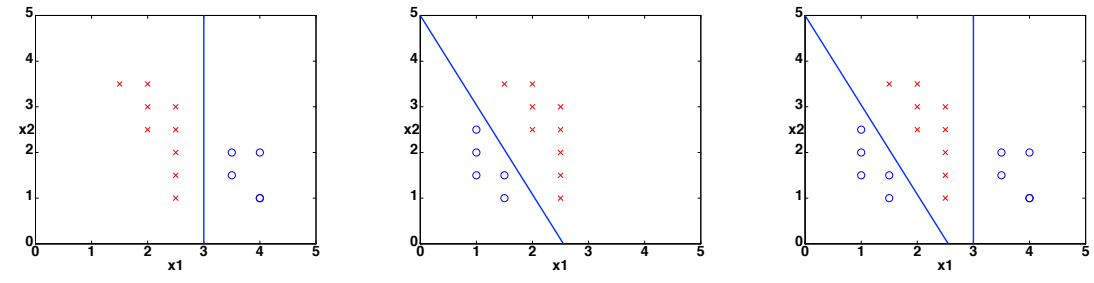
In this problem we will use a neural network to classify the crosses ($\times$) from the circles ($\circ$) in the simple dataset shown in Figure 5a. Even though the crosses and circles are not linearly separable, we can break the examples into three groups, S_1, S_2, and S_3 (shown in Figure 5a) so that S_1 is linearly separable from S_2 and S_2 is linearly separable from S_3. We will exploit this fact to design weights for the neural network shown in Figure 5b in order to correctly classify this training set. For all nodes, we will use the threshold activation function

$$\phi(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0. \end{cases}$$



(a) The dataset with groups S_1, S_2, and S_3. (b) The neural network architecture

Figure 5



(a) Set S2 and S3 (b) Set S1 and S2 (c) Set S1, S2 and S3

Figure 6: NN classification.

(a) First we will set the parameters w_{11}, w_{12} and b_1 of the neuron labeled h_1 so that its output $h_1(x) = \phi(w_{11}x_1 + w_{12}x_2 + b_1)$ forms a linear separator between the sets S_2 and S_3.

(1) [1 pt.] On Fig. 6a, draw a linear decision boundary that separates S_2 and S_3.

(2) [1 pt.] Write down the corresponding weights w_{11}, w_{12}, and b_1 so that $h_1(x) = 0$ for all points in S_3 and $h_1(x) = 1$ for all points in S_2.

Solution: $w_{11} = -1, w_{12} = 0, b_1 = 3$. With these parameters, we have $w_{11}x_1 + w_{22}x_2 + b_1 > 0$ if and only if $-x_1 > -3$, which is equivalent to $x_1 < 3$.

(b) Next we set the parameters w_{21}, w_{22} and b_2 of the neuron labeled h_2 so that its output $h_2(x) = \phi(w_{21}x_1 + w_{22}x_2 + b_2)$ forms a linear separator between the sets S_1 and S_2.

(1) [1 pt.] On Fig. 6b, draw a linear decision boundary that separates S_1 and S_2.

(2) [1 pt.] Write down the corresponding weights w_{21}, w_{22}, and b_2 so that $h_2(x) = 0$ for all points in S_1 and $h_2(x) = 1$ for all points in S_2.

Solution: The provided line has a slope of -2 and crosses the x_2 axis at the value 5. From this, the equation for the region above the line (those points for which $h_2(x) = 1$)) is given by $x_2 \geq -2x_1 + 5$ or, equivalently, $x_2 + 2x_1 - 5 \geq 0$. Therefore, $w_{21} = 2, w_{22} = 1, b_2 = -5$.

(c) Now we have two classifiers h_1 (to classify S_2 from S_3) and h_2 (to classify S_1 from S_2). We will set the weights of the final neuron of the neural network based on the results from h_1 and h_2 to classify the crosses from the circles. Let $h_3(x) = \phi\big(w_{31}h_1(x) + w_{32}h_2(x) + b_3\big)$.

(1) [1 pt.] Compute w_{31}, w_{32}, b_3 such that $h_3(x)$ correctly classifies the entire dataset.

Solution: Consider the weights $w_{31} = w_{32} = 1$ and $b_3 = -1.5$. With these weights, $h_3(x) = 1$ if $h_1(x) + h_2(x) \geq 1.5$. For points in S_1 and S_3 either h_1 or h_2 is zero, so they will be classified as 0, as required. For points in S_2, both h_1 and h_2 output 1, so the point is classified as 1. This rule has zero training error.

(2) [1 pt.] Draw your decision boundary in Fig. 6c.

Use this page for scratch work

10-701 Midterm Exam Solutions, Spring 2007

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be 16 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including my annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are allowed, but no laptops, PDAs, phones or Internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. Note there are extra-credit sub-questions. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.
7. You have 80 minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	21 + 0.911 extra	
2	SVM and slacks	16	
3	GNB	8	
4	Feature Selection	10	
5	Irrelevant Features	14 + 3 extra	
6	Neural Nets	16 + 5 extra	
7	Learning theory	15	

1 [21 Points] Short Questions

The following short questions should be answered with at most two sentences, and/or a picture. For the **(true/false)** questions, answer true or false. If you answer true, provide a short justification, if false explain why or provide a small counterexample.

1. [1 point] **true/false** A classifier trained on less training data is less likely to overfit.

   ★ **SOLUTION:** This is false. A specific classifier (with some fixed model complexity) will be more likely to overfit to noise in the training data when there is less training data, and is therefore more likely to overfit.

2. [1 point] **true/false** Given m data points, the training error converges to the true error as $m \to \infty$.

   ★ **SOLUTION:** This is true, if we assume that the data points are i.i.d. A few students pointed out that this might not be the case.

3. [1 point] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \alpha_1 x_1 x_2^3 + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

   ★ **SOLUTION:** This is true. y is linear in α_1, so it can be learned using linear regression.

4. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = x_1^{\alpha_1} e^{\alpha_2} + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

   ★ **SOLUTION:** This is false. y is not linear in α_1 and α_2, and no simple transformation will make it linear ($\log[x_1^{\alpha_1} e^{\alpha_2} + \epsilon_i] \neq \alpha_1 \log x_1 + \alpha_2 + \epsilon_i$).

5. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \log(x_1^{\alpha_1} e^{\alpha_2}) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

★ **SOLUTION:** This is true. $y_i = \log(x_1^{\alpha_1} e^{\alpha_2}) + \epsilon_i = \alpha_1 \log x_1 + \alpha_2 + \epsilon_i$, which is linear in α_1 and α_2. Also, assuming $x_1 > 0$.

6. [2 points] **true/false** In AdaBoost weights of the misclassified examples go up by the same multiplicative factor.

★ **SOLUTION:** True, follows from the update equation.

7. [2 points] **true/false** In AdaBoost, weighted training error ϵ_t of the t^{th} weak classifier on training data with weights D_t tends to increase as a function of t.

★ **SOLUTION:** True. In the course of boosting iterations the weak classifiers are forced to try to classify more difficult examples. The weights will increase for examples that are repeatedly misclassified by the weak classifiers. The weighted training error ϵ_t of the t^{th} weak classifier on the training data therefore tends to increase.

8. [2 points] **true/false** AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

★ **SOLUTION:** Not if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using. For example consider the EXOR example(In hw2 we worked with a *rotated* EXOR toy dataset) with decision stumps as weak classifiers. No matter how many iterations are performed zero training error will not be achieved.

9. [2 points] Consider a point that is correctly classified and distant from the decision boundary. Why would SVM's decision boundary be unaffected by this point, but the one learned by logistic regression be affected?

★ **SOLUTION:** The hinge loss used by SVMs gives zero weight to these points while the log-loss used by logistic regression gives a little bit of weight to these points.

10. [2 points] Why does the kernel trick allow us to solve SVMs with high dimensional feature spaces, without significantly increasing the running time?

★ **SOLUTION:** In the dual formulation of the SVM, features only appear as dot products which can be represented compactly by kernels.

11. [2 points] Consider a learning problem with 2D features. How are the decision tree and 1-nearest neighbor decision boundaries related?

    ★ **SOLUTION:** In both cases, the decision boundary is piecewise linear. Decision trees do axis-aligned splits while 1-NN gives a voronoi diagram.

12. [2 points] You are a reviewer for the International Mega-Conference on Algorithms for Radical Learning of Outrageous Stuff, and you read papers with the following experimental setups. Would you accept or reject each paper? Provide a one sentence justification. (This conference has short reviews.)

    - **accept/reject** "My algorithm is better than yours. Look at the training error rates!"

      ★ **SOLUTION:** Reject - the training error is optimistically biased.

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for $\lambda = 1.789489345672120002$.)"

      ★ **SOLUTION:** Reject - A λ with 15 decimal places suggests a highly tuned solution, probably looking at the test data.

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ.)"

      ★ **SOLUTION:** Reject - Choosing λ based on the test data?

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ, chosen with 10-fold cross validation.)"

      ★ **SOLUTION:** Accept - Cross validation is the appropriate method for selecting parameters.

13. [Extra credit: 0.911 points] You have designed the ultimate learning algorithm that uses physical and metaphysical knowledge to learn and generalize beyond the quantum P-NP barrier. You are now given the following test example:

    What label will your algorithm output?

(a) Watch a cartoon.

(b) Call the anti-terrorism squad.

(c) Support the Boston Red Sox.

(d) All labels have equal probability.

★ **SOLUTION:** Watching a cartoon earned you 0.39 points. 0.2005 points were given for supporting the Boston Red Sox. 0.666 points were given for calling the anti-terrorism squad. 0.911 points were given for "all labels have equal probability."

■ **COMMON MISTAKE :** Some students skipped this question; perhaps a Moonite Marauders is also scary on paper...

2 [16 Points] SVMs and the slack penalty C

The goal of this problem is to correctly classify test data points, given a training data set. You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much.

For this problem, assume that we are training an SVM with a **quadratic kernel**– that is, our kernel function is a polynomial kernel of degree 2. You are given the data set presented in Figure 1. The slack penalty C will determine the location of the separating hyperplane. Please answer the following questions *qualitatively*. Give a one sentence answer/justification for each and draw your solution in the appropriate part of the Figure at the end of the problem.

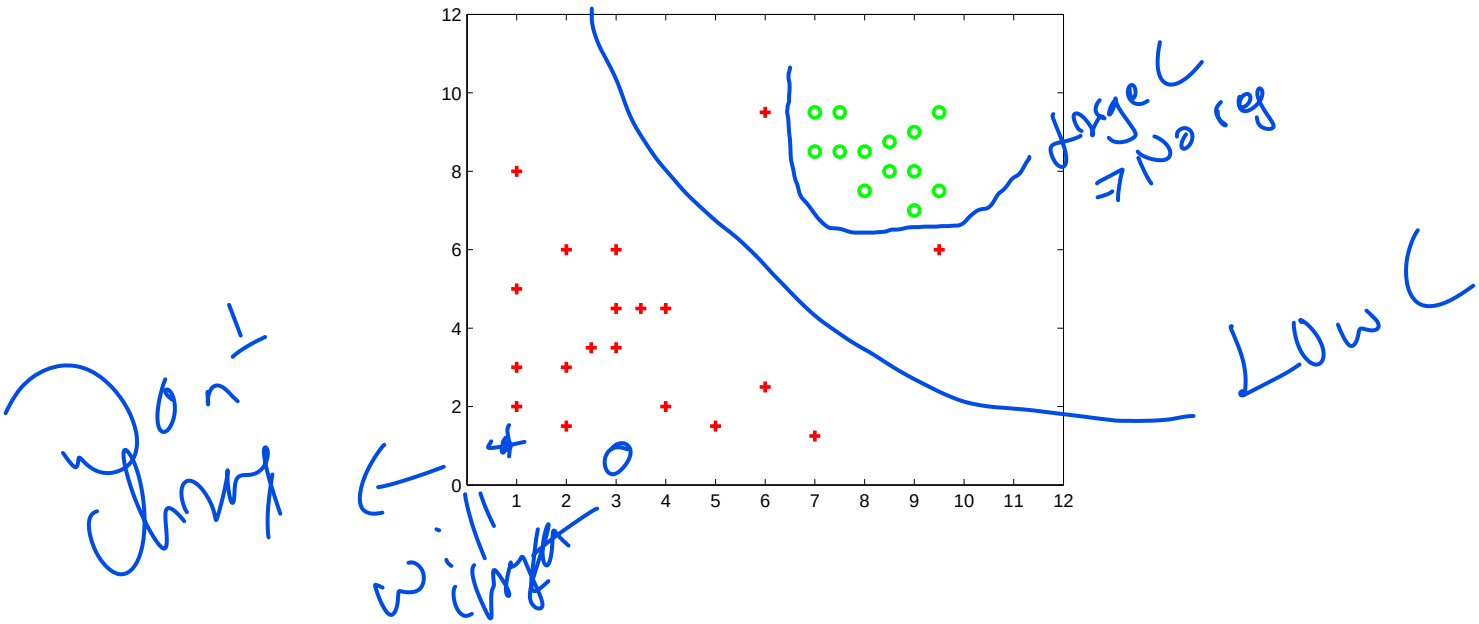


Figure 1: Dataset for SVM slack penalty selection task in Question 2.

1. [4 points] Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? (remember that we are using an SVM with a quadratic kernel.) Draw on the figure below. Justify your answer.

   ★ **SOLUTION:** For large values of C, the penalty for misclassifying points is very high, so the decision boundary will perfectly separate the data if possible. See below for the boundary learned using libSVM and $C = 100000$.

   ■ **COMMON MISTAKE 1:** Some students drew straight lines, which would not be the result with a quadratic kernel.

■ **COMMON MISTAKE 2:** Some students confused the effect of C and thought that a large C meant that the algorithm would be more tolerant of misclassifications.

2. [4 points] For $C \approx 0$, indicate in the figure below, where you would expect the decision boundary to be? Justify your answer.

★ **SOLUTION:** The classifier can maximize the margin between most of the points, while misclassifying a few points, because the penalty is so low. See below for the boundary learned by libSVM with $C = 0.00005$.

3. [2 points] Which of the two cases above would you expect to work better in the classification task? Why?

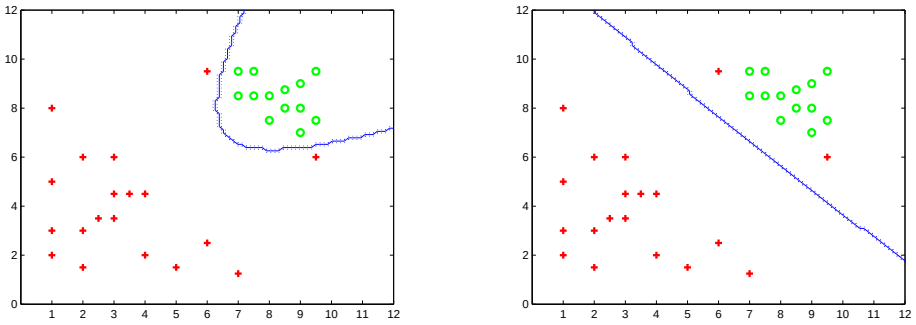
★ **SOLUTION:** We were warned not to trust any specific data point too much, so we prefer the solution where $C \approx 0$, because it maximizes the margin between the dominant clouds of points.

4. [3 points] Draw a data point which will not change the decision boundary learned for very large values of C. Justify your answer.

★ **SOLUTION:** We add the point circled below, which is correctly classified by the original classifier, and will not be a support vector.

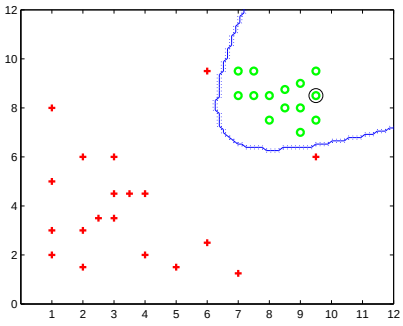
5. [3 points] Draw a data point which will significantly change the decision boundary learned for very large values of C. Justify your answer.

★ **SOLUTION:** Since C is very large, adding a point that would be incorrectly classified by the original boundary will force the boundary to move.

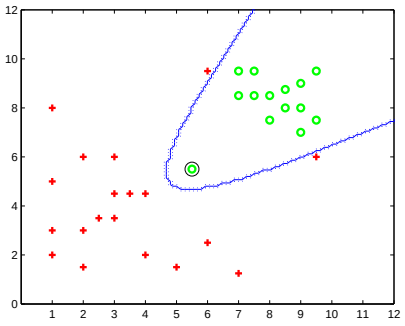


(a) Part 1

(b) Part 2



(c) Part 4



(d) Part 5

Figure 2: Solutions for Problem 2

3 [10 points] Feature selection with boosting

Consider a text classification task, such that the document X can be expressed as a binary feature vector of the words. More formally $X = [X_1, X_2, X_3, ..., X_m]$, where $X_j = 1$ if word j is present in document X, and zero otherwise. Consider using the AdaBoost algorithm with a simple weak learner, namely

$$h(X;\theta) = yX_j$$
$$\theta = \{j, y\} \; j \text{ is the word selector ; } y \text{ is the associated class}$$
$$y \in \{-1, 1\}$$

More intuitively, each weak learner is a word associated with a class label. For example if we had a word `football`, and classes {`sports`,`non-sports`}, then we will have two weak learners from this word, namely

- *Predict* **sports** *if document has word* **football**
- *Predict* **non-sports** *if document has word* **football**.

1. [2 points] How many weak learners are there ?

★ **SOLUTION:** Two weak learners for each word, i.e. 2m weak learners.

2. This boosting algorithm can be used for feature selection. We run the algorithm and select the features in the *order in which they were identified* by the algorithm.

   (a) [4 points] Can this boosting algorithm select the same weak classifier more than once? Explain.

   ★ **SOLUTION:** The boosting algorithm optimizes each new α by assuming that all the previous votes remain fixed. It therefore does not optimize these coefficients jointly. The only way to correct the votes assigned to a weak learner later on is to introduce the same weak learner again. Since we only have a discrete set of possible weak learners here, it also makes sense to talk about selecting the exact same weak learner again.

   (b) [4 points] Consider ranking the features based on their individual mutual information with the class variable y, i.e. $\hat{I}(y; X_j)$. Will this ranking be more informative than the ranking returned by AdaBoost ? Explain.

   ★ **SOLUTION:** The boosting algorithm generates a linear combination of weak classifiers (here features). The algorithm therefore evaluates each new weak classifier (feature) relative to a linear prediction based on those already included. The mutual information criterion considers each feature individually and is therefore unable to recognize how multiple features might interact to benefit linear prediction.

4 [8 points] Gaussian Naive Bayes classifier

Consider the datasets **toydata1** in figure **3(A)** and **toydata2** in figure **3(B)**.

- In each of these datasets there are two classes, '+' and 'o'.
- Each class has the same number of points. -7 Prior is same
- Each data point has two real valued features, the X and Y coordinates.

For each of these datasets, draw the decision boundary that a Gaussian Naive Bayes classifier will learn.

★ **SOLUTION:** For **toydata1** the crucial detail is that GNB learns diagonal covariance matrices yielding axis aligned Gaussians. In figure 4(A) the two circles are the gaussians learned by GNB, and hence the decision surface is the tangent through the point of contact.

For **toydata2** GNB learns two Gaussians , one for the circle inside with small variance , and one for the circle outside with a much larger variance, and the decision surface is roughly shown in figure 4(B).

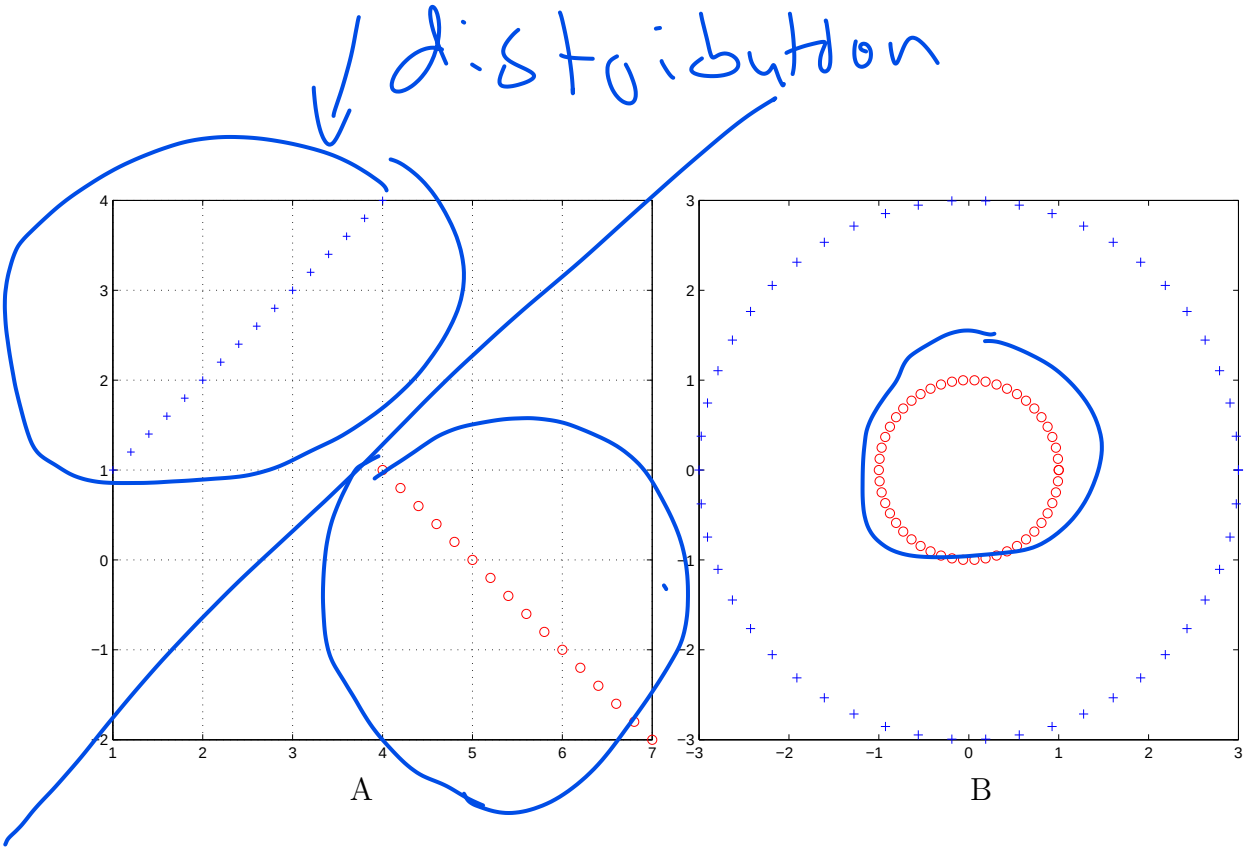


Figure 3: A. **toydata1** in Question 4, B. **toydata2** in Question 4

Remember that a very important piece of information was that all the classes had the *same* number of points, and so we don't have to worry about the prior.

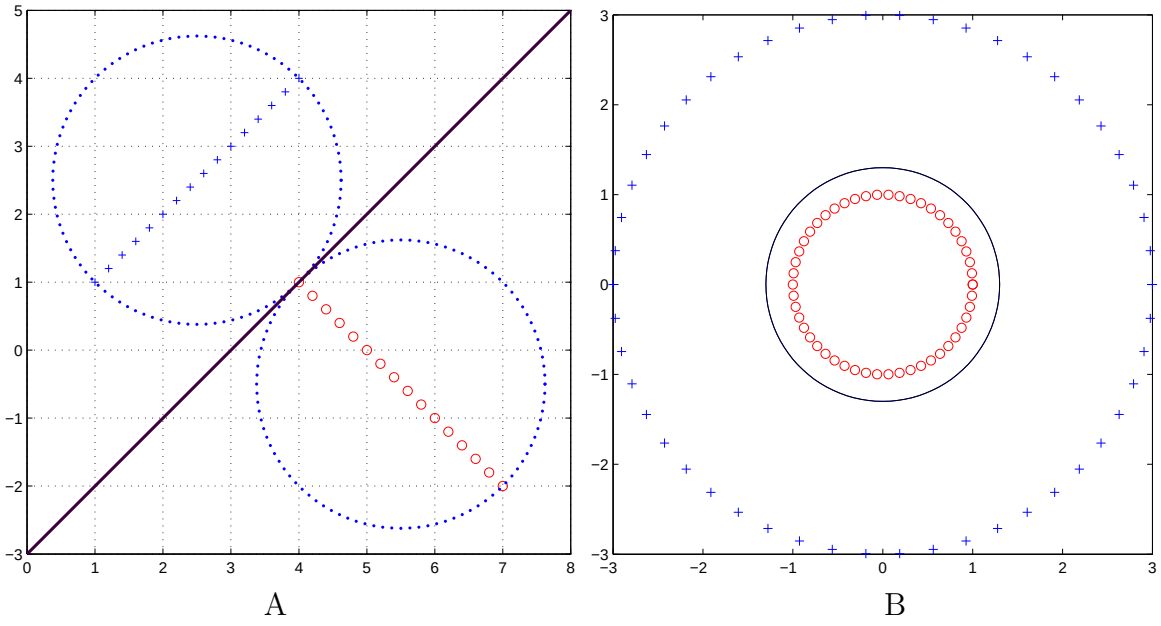
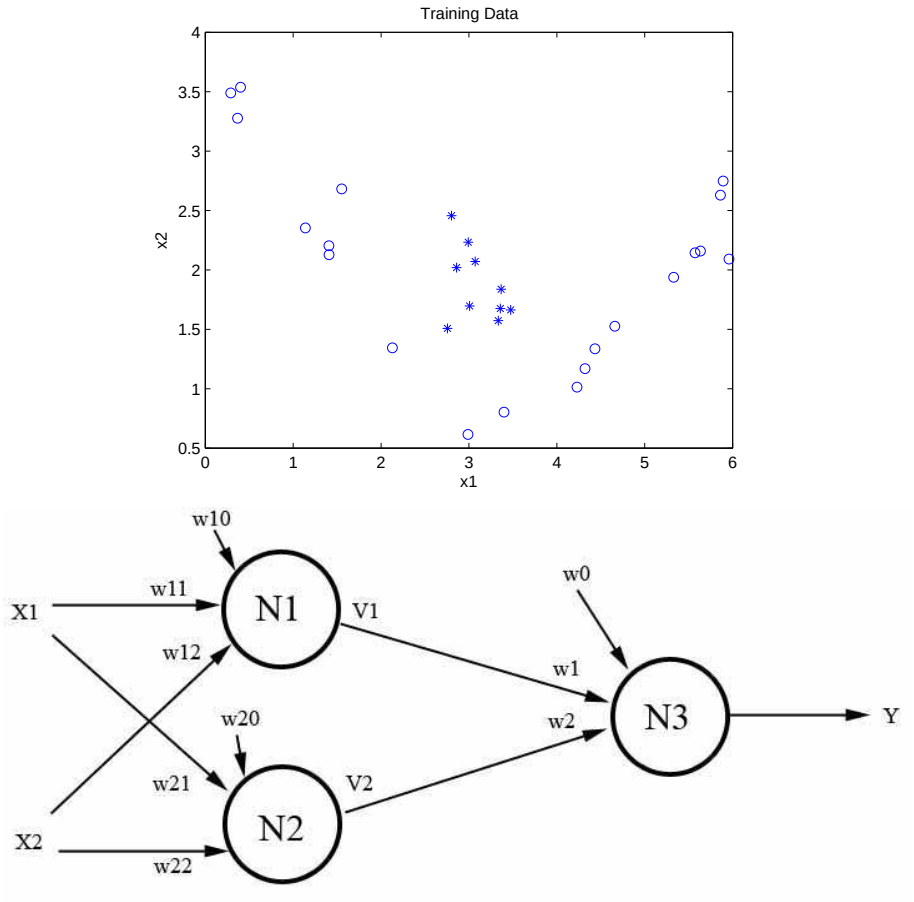


Figure 4: Solutions for A. **toydata1** in Question 4, B. **toydata2** in Question 4

5 [16 Points] Neural Networks

Consider the following classification training data (where "*" = true or 1 and "O" = false or 0) and neural network model that uses the **sigmoid** response function $\left(g(t) = \frac{1}{1+e^{-t}}\right)$.

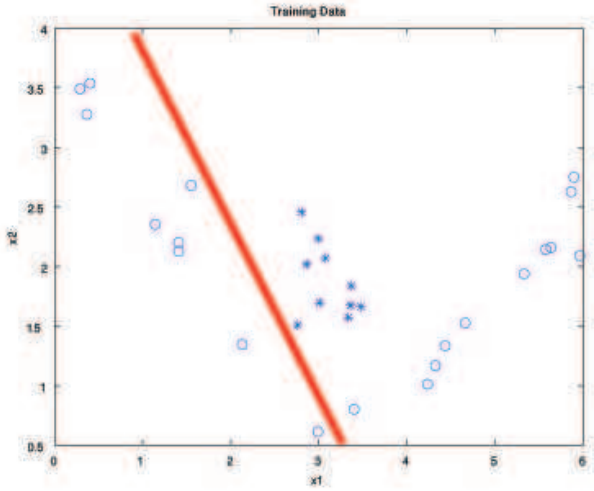


5.1 Weight choice

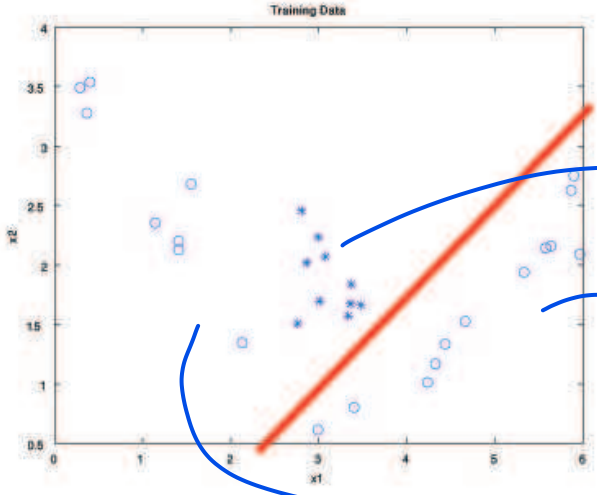
[8 points] We would like to set the weights (w) of the neural network so that it is capable of correctly classifying this dataset. Please plot the decision boundaries for N_1 and N_2 (e.g., for neuron N_1, the line where $w10 + w11 * X1 + w12 * X2 = 0$) on the first two graphs. In the third graph, which has axes V_2 and V_1, plot $\{V_1(x_1, x_2), V_2(x_1, x_2)\}$ for a few of the training points and provide a decision boundary so that the neural net will correctly classify the training data.

All graphs are on the following page!

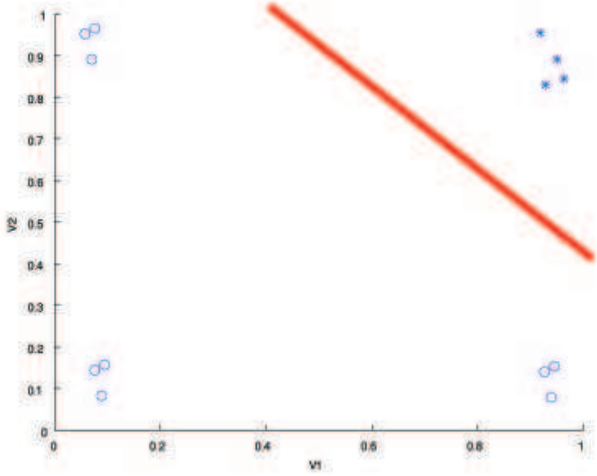
N1 (2 points)



N2 (2 points)



N3 (4 points)



5.2 Regularized Neural Networks

[8 points]

One method for preventing the neural networks' weights from overfitting is to add regularization terms. You will now derive the update rules for the regularized neural network. **Note:** $Y = out(x)$

Recall that the non-regularized gradient descent update rule for w_1 is:

$$w_1^{t+1} \leftarrow w_1^t + \eta \sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right] \tag{1}$$

[4 points] Derive the update rule for w_1 in the regularized neural net loss function which penalizes based on the square of each weight. Use λ to denote the magic regularization parameter.

★ **SOLUTION:** The regularization term is $\lambda(\sum_i w_i^2)$. Differentiating with respect to w_1 yields $2\lambda w_1$. The update rule is

$$w_1^{t+1} \leftarrow w_1^t + \eta(\sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right] - 2\lambda w_1)$$

[4 points] Now, re-express the regularized update rule so that the only difference between the regularized setting and the unregularized setting above is that the old weight w_1^t is scaled by some constant. Explain how this scaling prevents overfitting.

★ **SOLUTION:**

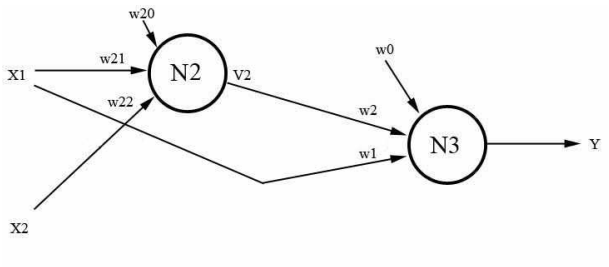
$$w_1^{t+1} \leftarrow w_1^t(1 - 2\eta\lambda) + \eta \sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right]$$

At each update the weight is kept closer to zero by the $(1 - 2\eta\lambda)$ term. This prevents the weights from becoming very large, which corresponds to overfitting.

5.3 Neural Net Simplification [Extra Credit (5 points)]

Please provide a feed-forward neural network with a smaller architecture (i.e., fewer neurons and weights) that is able to correctly predict the entire training set. Justify your answer.

★ **SOLUTION:** One solution follows from the observation that the decision boundary for N1 could be $x_1 = 3.7$. In fact, $N1$ can be removed entirely from the model. This yields a similar decision boundary for $N3$ except that $V1 = x_1$ ranges from 0 to 6.



Other possible solutions are to change the input feature space (e.g., by adding x_1^2 as an input to a single neuron along with x_1, x_2).

6 [14 Points] The Effect of Irrelevant Features

1. (a) [3 points] Provide a 2D dataset where 1-nearest neighbor (1-NN) has lower leave-one-out cross validation error (LOO error) than SVMs.

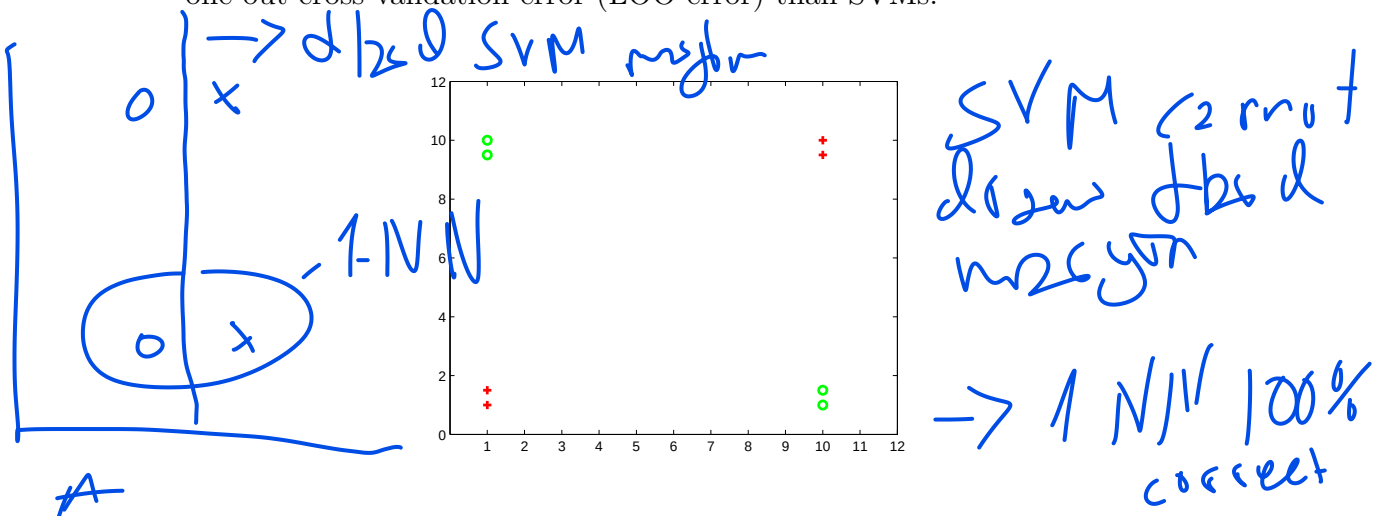


Figure 5: Dataset where 1-NN will do better than SVM in LOOCV.

(b) [3 points] Provide a 2D dataset where 1-NN has higher LOO error than SVMs.

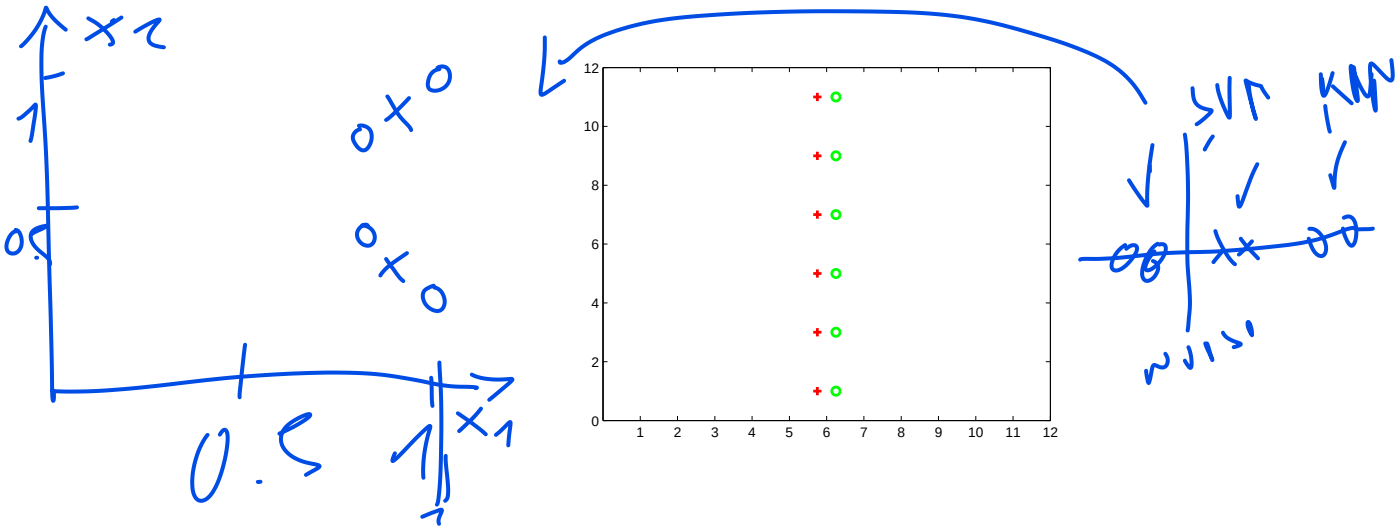


Figure 6: Dataset where SVM will do better than 1-NN in LOOCV.

2. [8 points] You will now generate a dataset to illustrate SVMs' robustness to irrelevant features. In particular, create a 2D dataset with features X_1 and X_2, here X_2 will be the irrelevant feature, such that:

   - If you only use X_1, 1-NN will have lower LOO error than SVMs,
   - but if you use both X_1 and X_2, the SVM LOO error will remain the same, but LOO error for 1-NN will increase significantly.

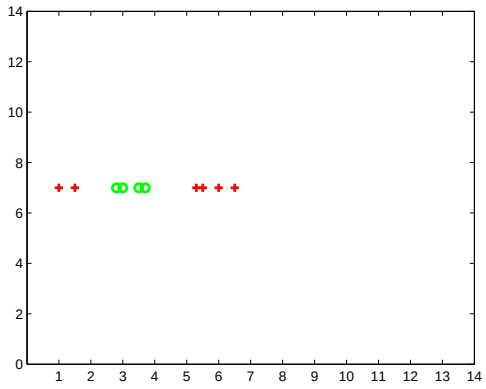


Figure 7: Here the horizontal axis is X_1, and we ignore X_2. 1-NN is perfect, and SVM will get the two points on the left wrong in LOOCV.

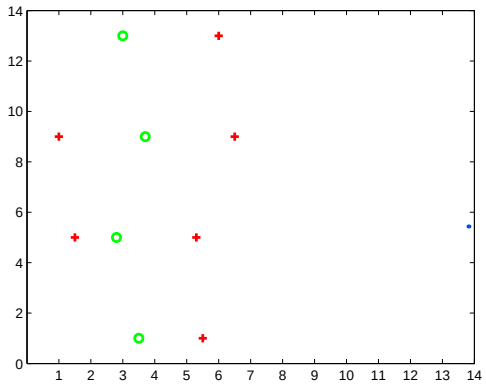


Figure 8: Here the horizontal axis is X_1, and the vertical axis is X_2. X_2 is the irrelevant feature. 1-NN gets every point wrong in LOOCV, while SVM has the same error.

You will receive extra credit if the 1-NN LOO error before adding the irrelevant feature is zero, but the error becomes 100% after adding the feature.

3. [Extra Credit (3 points)] SVMs tend to be robust to irrelevant features. Suppose we run SVMs with features $X_1, \ldots, X_n$, and then add a irrelevant feature X_{n+1} that cannot help increase the margin. How will SVMs automatically ignore this feature? Justify your answer formally.

★ **SOLUTION:** SVMs will automatically ignore this feature because it cannot possibly increase the margin, so giving it non-zero weight keeps the same margin but increases the regularization penalty. Therefore the solution with zero weight is superior to (i.e. has smaller objective function) all feasible solutions of the QP where this feature has non-zero weight.

↳ it sets the $w = 0$

7 [15 points] Learning Theory

Consider binary data-points X in n dimensions, with binary labels Y, i.e. $X \in \{0,1\}^n$; $Y \in \{0,1\}$. We wish to learn a mapping $X \to Y$ using a few different hypothesis classes, but are concerned about the tradeoff between the expressivity of our hypothesis space and the number of training examples required to learn the true mapping probably approximately correctly.

1. Consider the following hypothesis class H: decision stumps that choose a value for Y based on the value of one of the attributes of X. For example, there are two hypotheses in H that involve feature i:

$$h_i(X) = \begin{cases} 1 & \text{if } X_i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad h_{\neg i}(X) = \begin{cases} 0 & \text{if } X_i = 1 \\ 1 & \text{otherwise;} \end{cases}$$

   (a) [3 points] What is the size of this hypothesis class?

   ★ **SOLUTION:** $H = 2n$

   (b) [3 points] For given ϵ, δ how many training examples are needed to yield a decision stump that satisfies the Haussler-PAC bound?

   ★ **SOLUTION:**

$$|H|e^{-m\epsilon} \leq \delta$$

$$\begin{aligned} m &\geq \frac{1}{\epsilon}\left(\log|H| + \log\frac{1}{\delta}\right) \\ &= \frac{1}{\epsilon}\left(\log(2n) + \log\frac{1}{\delta}\right) \end{aligned}$$

2. Now let us define another hypothesis class H', where each hypothesis is a majority over a set of simple decision stumps. Specifically, for each feature i, we either use h_i or $h_{\neg i}$, and the output is the result of a majority vote (in the case of a tie, we predict 1). For example, if we have 5 features, and we choose the stumps $\{h_{\neg 1}, h_2, h_3, h_4, h_{\neg 5}\}$, then the resulting hypothesis is:

$$h'(X) = \begin{cases} 1 & \text{if } h_{\neg 1}(X) + h_2(X) + h_3(X) + h_4(X) + h_{\neg 5}(X) \geq \frac{5}{2} \\ 0 & \text{otherwise} \end{cases}$$

(a) [4 points] What is the size of this hypothesis class?

★ **SOLUTION:** Each element of the hypothesis class here corresponds to picking a subset of n features. Hence:

$$|H| = 2^n$$

(b) [2 points] For given ϵ, δ how many training examples are needed to yield a hypothesis that satisfies the Haussler-PAC bound?

★ **SOLUTION:**

$$m \geq \frac{1}{\epsilon}\left(n \log 2 + \log \frac{1}{\delta}\right)$$

3. [3 points] What can we say about the amount of extra samples necessary to learn this voting classifier? Is this a concern?
Briefly explain the tradeoff between the expressive power of the hypothesis space and the number of training samples required for these two classifier.

★ **SOLUTION:** In the first part of the problem, the required number of samples scales as $O(\log n)$, and in the second part of the problem, it scales as $O(n)$. So we need more samples to get the PAC bound for the second hypothesis class, but scaling linearly in the number of features is probably acceptable.

The tradeoff illustrated here is that greater expressive power (as with H') necessitates more training samples. The smaller and less expressive class H requires fewer training examples.

10-701/15-781 Machine Learning - Midterm Exam, Fall 2010

Aarti Singh
Carnegie Mellon University

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be **15** numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are not necessary. Laptops, PDAs, phones and Internet access are not allowed.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. You have **90** minutes.
7. Good luck!

Question	Topic	Max. score	Score
1	Short questions	20	
2	Bayes Optimal Classification	15	
3	Logistic Regression	18	
4	Regression	16	
5	SVM	16	
6	Boosting	15	
	Total	100	

1 Short Questions [20 pts]

Are the following statements True/False? Explain your reasoning in only 1 sentence.

1. Density estimation (using say, the kernel density estimator) can be used to perform classification.

   True: Estimate the joint density $P(Y, X)$, then use it to calculate $P(Y|X)$.

2. The correspondence between logistic regression and Gaussian Naïve Bayes (with identity class covariances) means that there is a one-to-one correspondence between the parameters of the two classifiers.

   False: Each LR model parameter corresponds to a whole set of possible GNB classifier parameters, there is no one-to-one correspondence because logistic regression is discriminative and therefore doesn't model $P(X)$, while GNB does model $P(X)$.

3. The training error of 1-NN classifier is 0.

   True: Each point is its own neighbor, so 1-NN classifier achieves perfect classification on training data.

4. As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant.

   False: A simple counterexample is the prior which assigns probability 1 to a single choice of parameter θ.

5. Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.

   True: The number of iterations in boosting controls the complexity of the model, therefore, a model selection procedure like cross validation can be used to select the appropriate model complexity and reduce the possibility of overfitting.

6. The kernel density estimator is equivalent to performing kernel regression with the value $Y_i = \frac{1}{n}$ at each point X_i in the original data set.

   False: Kernel regression predicts the value of a point as the weighted average of the values at nearby points, therefore if all of the points have the same value, then kernel regression will predict a constant (in this case, $\frac{1}{n}$) for all values.

7. We learn a classifier f by boosting weak learners h. The functional form of f's decision boundary is the same as h's, but with different parameters. (e.g., if h was a linear classifier, then f is also a linear classifier).

   False: For example, the functional form of a decision stump is a single axis-aligned split of the input space, but the functional form of the boosted classifier is linear combinations of decision stumps which can form a more complex (piecewise linear) decision boundary.

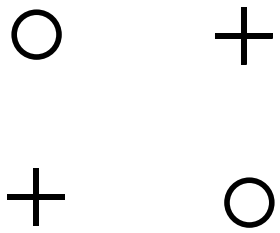
8. The depth of a learned decision tree can be larger than the number of training examples used to create the tree.

   False: Each split of the tree must correspond to at least one training example, therefore, if there are n training examples, a path in the tree can have length at most n.

   Note: There is a pathological situation in which the depth of a learned decision tree can be larger than number of training examples n - if the number of features is larger than n and there exist training examples which have same feature values but different labels. Points have been given if you answered true and provided this explanation.

For the following problems, circle the correct answers:

1. Consider the following data set:



   Circle all of the classifiers that will achieve zero training error on this data set. (You may circle more than one.)

   (a) Logistic regression

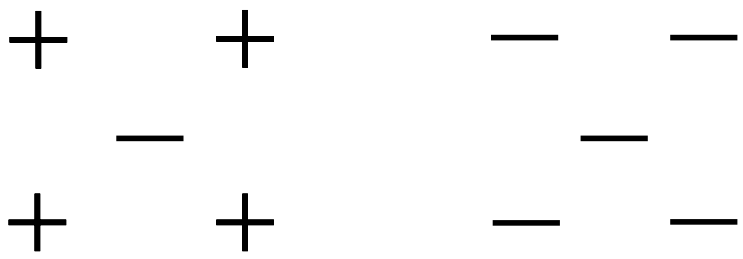
   (b) SVM (quadratic kernel)

   (c) Depth-2 ID3 decision trees

   (d) 3-NN classifier

   Solution: SVM (quad kernel) and Depth-2 ID3 decision trees

2. For the following dataset, circle the classifier which has larger Leave-One-Out Cross-validation error.



a) 1-NN

b) 3-NN

Solution: 1-NN since 1-NN CV err: 5/10, 3-NN CV err: 1/10

2 Bayes Optimal Classification [15 pts]

In classification, the loss function we usually want to minimize is the 0/1 loss:

$$\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$$

where $f(x), y \in \{0, 1\}$ (i.e., binary classification). In this problem we will consider the effect of using an asymmetric loss function:

$$\ell_{\alpha,\beta}(f(x), y) = \alpha\mathbf{1}\{f(x) = 1, y = 0\} + \beta\mathbf{1}\{f(x) = 0, y = 1\}$$

Under this loss function, the two types of errors receive different weights, determined by $\alpha, \beta > 0$.

1. **[4 pts]** Determine the Bayes optimal classifier, i.e. the classifier that achieves minimum risk assuming $P(x, y)$ is known, for the loss $\ell_{\alpha,\beta}$ where $\alpha, \beta > 0$.

   Solution: We can write

$$\begin{aligned}
\arg\min_f \mathbb{E}\ell_{\alpha,\beta}(f(x), y) &= \arg\min_f \mathbb{E}_{X,Y}[\alpha\mathbf{1}\{f(X) = 1, Y = 0\} + \beta\mathbf{1}\{f(X) = 0, Y = 1\}] \\
&= \arg\min_f \mathbb{E}_X[\mathbb{E}_{Y|X}[\alpha\mathbf{1}\{f(X) = 1, Y = 0\} + \beta\mathbf{1}\{f(X) = 0, Y = 1\}]] \\
&= \arg\min_f \mathbb{E}_X[\int_y \alpha\mathbf{1}\{f(X) = 1, y = 0\} + \beta\mathbf{1}\{f(X) = 0, y = 1\} dP(y|x)] \\
&= \arg\min_f \int_x [\alpha\mathbf{1}\{f(x) = 1\}P(y = 0|x) + \beta\mathbf{1}\{f(x) = 0\}P(y = 1|x)] dP(x)
\end{aligned}$$

   We may minimize the integrand at each x by taking:

$$f(x) = \begin{cases} 1 & \beta P(y = 1|x) \geq \alpha P(y = 0|x) \\ 0 & \alpha P(y = 0|x) > \beta P(y = 1|x). \end{cases}$$

2. **[3 pts]** Suppose that the class $y = 0$ is extremely uncommon (i.e., $P(y = 0)$ is small). This means that the classifier $f(x) = 1$ for all x will have good risk. We may try to put the two classes on even footing by considering the risk:

$$R = P(f(x) = 1|y = 0) + P(f(x) = 0|y = 1)$$

   Show how this risk is equivalent to choosing a certain α, β and minimizing the risk where the loss function is $\ell_{\alpha,\beta}$.

   Solution: Notice that

$$\begin{aligned}
E\ell_{\alpha,\beta}(f(x), y) &= \alpha P(f(x) = 1, y = 0) + \beta P(f(x) = 0, y = 1) \\
&= \alpha P(f(x) = 1|y = 0)P(y = 0) + \beta P(f(x) = 0|y = 1)P(y = 1)
\end{aligned}$$

   which is same as the minimizer of the given risk R if $\alpha = \frac{1}{P(y=0)}$ and $\beta = \frac{1}{P(y=1)}$.

3. [**4 pts**] Consider the following classification problem. I first choose the label $Y \sim$ Bernoulli($\frac{1}{2}$), which is 1 with probability $\frac{1}{2}$. If $Y = 1$, then $X \sim$ Bernoulli(p); otherwise, $X \sim$ Bernoulli(q). Assume that $p > q$. What is the Bayes optimal classifier, and what is its risk?

Solution: Since label is equally likely to be 1 or 0, to minimize prob of error simply predict the label for which feature value X is most likely. Since $p > q$, $X = 1$ is most likely for $Y = 1$ and $X = 0$ is most likely for $Y = 0$. Hence $f^*(X) = X$. Baye's risk $= P(X \neq Y) = 1/2 \cdot (1 - p) + 1/2 \cdot q$.

Formally: Notice that since $Y \sim$ Bernoulli($\frac{1}{2}$), we have $P(Y = 1) = P(Y = 0) = 1/2$.

$$\begin{aligned} f^*(x) = \arg\max_y P(Y = y|X = x) &= \arg\max_y P(X = x|Y = y)P(Y = y) \\ &= \arg\max_y P(X = x|Y = y) \end{aligned}$$

Therefore, $f^*(1) = 1$ since $p = P(X = 1|Y = 1) > P(X = 1|Y = 0) = q$, and $f^*(0) = 0$ since $1 - p = P(X = 0|Y = 1) < P(X = 0|Y = 0) = 1 - q$. Hence $f^*(X) = X$. The risk is $R^* = P(f^*(X) \neq Y) = P(X \neq Y)$.

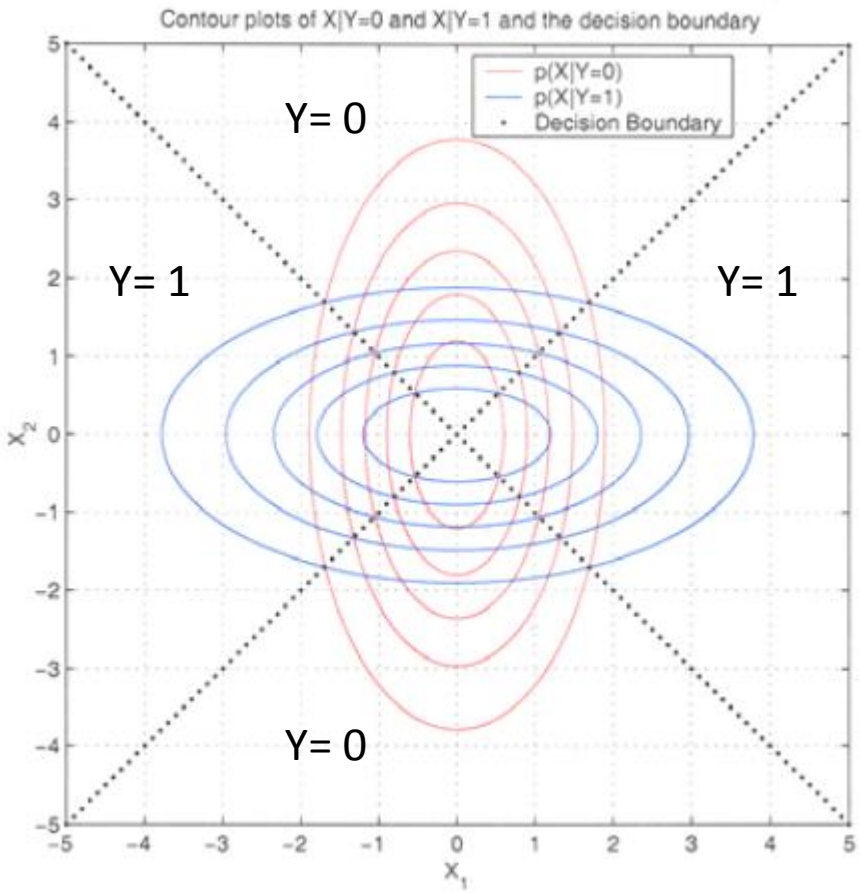
$$R^* = P(Y = 1)P(X = 0|Y = 1) + P(Y = 0)P(X = 1|Y = 0) = \frac{1}{2} \cdot (1 - p) + \frac{1}{2} \cdot q.$$

4. [**4 pts**] Now consider the regular 0/1 loss ℓ, and assume that $P(y = 0) = P(y = 1) = 1/2$. Also, assume that the class-conditional densities are Gaussian with mean μ_0 and co-variance Σ_0 under class 0, and mean μ_1 and co-variance Σ_1 under class 1. Further, assume that $\mu_0 = \mu_1$.

For the following case, draw contours of the level sets of the class conditional densities and label them with $p(x|y = 0)$ and $p(x|y = 1)$. Also, draw the decision boundaries obtained using the Bayes optimal classifier in each case and indicate the regions where the classifier will predict class 0 and where it will predict class 1.

$$\Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

Solution: next page


Contour plots of X|Y=0 and X|Y=1 and the decision boundary
p(X|Y=0)
p(X|Y=1)
Decision Boundary
Y= 0
Y= 1
Y= 1
Y= 0
X2
X1

3 Logistic Regression [18 pts]

We consider here a discriminative approach for solving the classification problem illustrated in Figure 1.

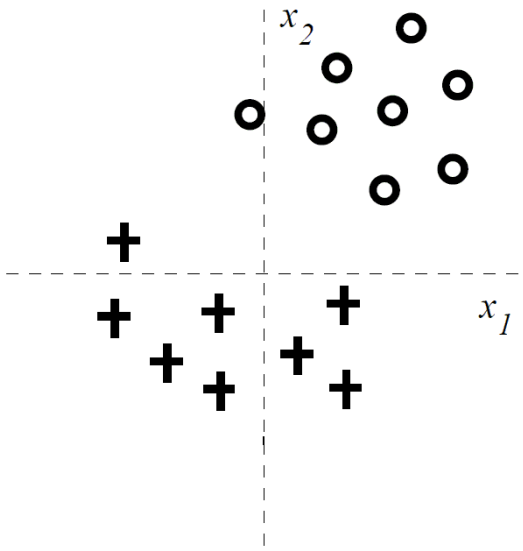


Figure 1: The 2-dimensional labeled training set, where '+' corresponds to class y=1 and 'O' corresponds to class $y = 0$.

1. We attempt to solve the binary classification task depicted in Figure 1 with the simple linear logistic regression model

$$P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1x_1 + w_2x_2) = \frac{1}{1 + exp(-w_0 - w_1x_1 - w_2x_2)}.$$

Notice that the training data can be separated with *zero* training error with a linear separator.

Consider training *regularized* linear logistic regression models where we try to maximize

$$\sum_{i=1}^{n} \log\ (P(y_i|x_i, w_0, w_1, w_2)) - Cw_j^2$$

for very large C. The regularization penalties used in penalized conditional log-likelihood estimation are $-Cw_j^2$, where $j = \{0, 1, 2\}$. In other words, only one of the parameters is regularized in each case. Given the training data in Figure 1, how does the training error change with regularization of each parameter w_j? State whether the training error increases or stays the same (zero) for each w_j for very large C. Provide a brief justification for each of your answers.

(a) By regularizing w_2 **[2 pts]**

SOLUTION: Increases. When we regularize w_2, the resulting boundary can rely less and less on the value of x_2 and therefore becomes more vertical. For very large C, the training error increases as there is no good linear vertical separator of the training data.

(b) By regularizing w_1 **[2 pts]**

SOLUTION: Remains the same. When we regularize w_1, the resulting boundary can rely less and less on the value of x_1 and therefore becomes more horizontal and the training data can be separated with *zero* training error with a horizontal linear separator.

(c) By regularizing w_0 **[2 pts]**

SOLUTION: Increases. When we regularize w_0, then the boundary will eventually go through the origin (bias term set to zero). Based on the figure, we can *not* find a linear boundary through the origin with *zero* error. The best we can get is one error.

2. If we change the form of regularization to L1-norm (absolute value) and regularize w_1 and w_2 only (but not w_0), we get the following penalized log-likelihood

$$\sum_{i=1}^{n} \log P(y_i|x_i, w_0, w_1, w_2) - C(|w_1| + |w_2|).$$

Consider again the problem in Figure 1 and the same linear logistic regression model $P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1x_1 + w_2x_2)$.

(a) **[3 pts]** As we increase the regularization parameter C which of the following scenarios do you expect to observe? (Choose only one) Briefly explain your choice:

( ) First w_1 will become 0, then w_2.

( ) First w_2 will become 0, then w_1.

( ) w_1 and w_2 will become zero simultaneously.

( ) None of the weights will become exactly zero, only smaller as C increases.

SOLUTION: First w_1 will become 0, then w_2.

The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e. making $w_1 = 0$. Initially we might prefer to have a non-zero value for w_1 but it will go to zero rather quickly as we increase regularization. Note that we pay a regularization penalty for a non-zero value of w_1 and if it does not help classification why would we pay the penalty? Also, the absolute value regularization ensures that w_1 will indeed go to *exactly* zero. As C increases further, even w_2 will eventually become zero. We pay higher and higher cost for setting w_2 to a non-zero value. Eventually this cost overwhelms the gain from the log-probability of labels that we can achieve with a non-zero w_2.

(b) **[3 pts]** For very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly. (Note that the number of points from each class is the same.) (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. Note that when $w_1 = w_2 = 0$, the log-probability of labels becomes a finite value, which is equal to n log(0.5), i.e. $w_0 = 0$. In other words, $P(y = 1|\vec{x}, \vec{w})$=$P(y = 0|\vec{x}, \vec{w})$=0.5. We expect so because the number of elements in each class is the same and so we would like to predict each one with the same probability, and w_0=0 makes $P(y = 1|\vec{x}, \vec{w})$=0.5.

(c) **[3 pts]** Assume that we obtain more data points from the '+' class that corresponds to y=1 so that the class labels become unbalanced. Again for very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly. (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. With unbalanced classes where the number of '+' labels are greater than that of 'o' labels, we want to have $P(y = 1|\vec{x}, \vec{w}) > P(y = 0|\vec{x}, \vec{w})$. For that to happen the value of w_0 should be greater than zero which makes $P(y = 1|\vec{x}, \vec{w}) > 0.5$.

4 Kernel regression [16 pts]

Now lets consider the non-parametric kernel regression setting. In this problem, you will investigate univariate locally linear regression where the estimator is of the form:

$$\widehat{f}(x) = \beta_1 + \beta_2 x$$

and the solution for parameter vector $\beta = [\beta_1 \ \beta_2]$ is obtained by minimizing the weighted least square error:

$$J(\beta_1, \beta_2) = \sum_{i=1}^{n} W_i(x)(Y_i - \beta_1 - \beta_2 X_i)^2 \quad \text{where} \quad W_i(x) = \frac{K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)},$$

where K is a kernel with bandwidth h. Observe that the weighted least squares error can be expressed in matrix form as

$$J(\beta_1, \beta_2) = (Y - A\beta)^T W (Y - A\beta),$$

where Y is a vector of n labels in the training example, W is a $n \times n$ diagonal matrix with weight of each training example on the diagonal, and

$$A = \begin{bmatrix} 1 \ X_1 \\ 1 \ X_2 \\ \cdots \\ 1 \ X_n \end{bmatrix}$$

1. [**4 pts**] Derive an expression in matrix form for the solution vector $\hat{\beta}$ that minimizes the weighted least square.

   Solution: Differentiating the objective function wrt β, we have:

   $$\frac{\partial J(\beta)}{\beta} = 2A^T W A\beta - 2A^T W^T Y.$$

   Therefore, the solution $\hat{\beta}$ satisfies the following normal equations:

   $$A^T W A\beta = A^T W^T Y$$

   And if $A^T W A$ is invertible, then the solution is $\hat{\beta} = (A^T W A)^{-1} A^T W^T Y$. (Note that $W = W^T$, so the solution can be written in terms of either).

2. [**3 pts**] When is the above solution unique?

   Solution: When $A^T W A$ is invertible. Since W is a diagonal matrix, $A^T W A = (W^{1/2}A)^T (W^{1/2}A)$ and hence rank$(A^T W A) = \min(n, 2)$ - Refer TK's recitation notes. Since a matrix is invertible if it is full rank, a unique solution exists if $n \geq 2$.

3. [**3 pts**] If the solution is not unique, one approach is to optimize the objective function J using gradient descent. Write the update equation for gradient descent in this case. Note: Your answer must be expressed in terms of the matrices defined above.

Solution: Let $\alpha > 0$ denote the step-size.

$$\begin{aligned}\beta^{(t+1)} &= \beta^{(t)} - \frac{\alpha}{2}\frac{\partial J(\beta)}{\partial \beta} \\ &= \beta^{(t)} - \alpha A^T W(A\beta - Y)\end{aligned}$$

4. [**3 pts**] Can you identify the signal plus noise model under which maximizing the likelihood (MLE) corresponds to the weighted least squares formulation mentioned above?

   Solution: $Y = \beta_1 + \beta_2 X + \epsilon$, where $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_i^2)$ for $i = 1, \ldots, n$. Here $\sigma_i^2 \propto 1/W_i(x)$.

5. [**3 pts**] Why is the above setting non-parametric? Mention one advantage and one disadvantage of nonparametric techniques over parametric techniques.

   Solution: The above setting is non-parametric since it performs locally linear fits, therefore number of parameters scale with data. Notice that $W_i(x)$, and hence the solution $\hat{\beta}$, depends on x. Thus we are fitting the parameters to every point x - therefore total number of parameters can be larger than n.

   Nonparametric techniques do not place very strict assumptions on the form of the underlying distribution or regression function, but are typically computationally expensive and require large number of training examples.

5 SVM [16 pts]

5.1 L2 SVM

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ be a set of l training pairs of feature vectors and labels. We consider binary classification, and assume $y_i \in \{-1, +1\}\ \forall i$. The following is the primal formulation of L2 SVM, a variant of the standard SVM obtained by squaring the hinge loss:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \ \geq\ 1 - \xi_i, \quad i \in \{1, \ldots, l\}, \\ & \xi_i \ \geq\ 0, \quad i \in \{1, \ldots, l\}. \end{aligned}$$

1. **[4 pts]** Show that removing the last set of constraints $\{\xi_i \geq 0\ \forall i\}$ does not change the optimal solution to the primal problem.

   Solution: Let $(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*)$ be the optimal solution to the problem without the last set of constraints. It suffices to show that $\xi_i^* \geq 0\ \forall i$. Suppose it is not the case, then there exists some $\xi_j^* < 0$. Then we have

   $$y_j((\mathbf{w}^*)^\top\mathbf{x}_j + b^*) \ \geq\ 1 - \xi_j^* \ >\ 1,$$

   implying that $\xi_j' = 0$ is a feasible solution and yet gives a smaller objective value since $(\xi_j')^2 = 0 < (\xi_j^*)^2$, a contradiction to the assumption that ξ_j^* is optimal.

2. **[3 pts]** After removing the last set of constraints, we get a simpler problem:

   $$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \ \geq\ 1 - \xi_i, \quad i \in \{1, \ldots, l\}. \end{aligned} \tag{1}$$

   Give the Lagrangian of (1).

   Solution: The Lagrangian is

   $$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \ :=\ \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i),$$

   where $\alpha_i \geq 0,\ \forall i$ are the Lagrange multipliers.

3. **[6 pts]** Derive the dual of (1). How is it different from the dual of the standard SVM with the hinge loss?

Solution: Taking partial derivatives of the Lagrangian wrt $\mathbf{w}$, b and ξ_i,

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \;,$$

$$\partial_b L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \sum_{i=1}^{l} \alpha_i y_i \;=\; 0,$$

$$\partial_{\xi_i} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \xi_i = \alpha_i / C \;.$$

Plugging these back to the Lagrangian, rearranging terms and keeping constraints on the Lagrange multipliers we obtain the dual

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top (Q + I/C)\boldsymbol{\alpha} + \mathbf{1}^\top \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{y}^\top \boldsymbol{\alpha} \;=\; 0, \quad \alpha_i \;\geq\; 0 \;\forall i, \end{aligned}$$

where $\mathbf{1}$ is a vector of ones, I is the identity matrix, $\mathbf{y}$ is the vector of labels y_i's, and Q is the l-by-l kernel matrix such that $Q_{ij} = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$. Compared with the dual of the standard SVM, the quadratic term is regularized by an additional positive diagonal matrix, and thus has stronger convexity leading to faster convergence. The other difference is that the dual variables here are only bounded from below, but in the standard SVM the dual variables are bounded both from above (by C) and from below. In fact, for L2 svms the solution does not depend on the tradeoff parameter C.

5.2 Leave-one-out Error and Support Vectors

[3 pts] Consider the standard two-class SVM with the hinge loss. Argue that under a given value of C,

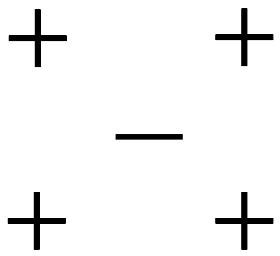
$$\text{LOO error} \;\leq\; \frac{\#\text{SVs}}{l},$$

where l is the size of the training data and #SVs is the number of support vectors obtained by training SVM on the entire set of training data.

Solution: Since the decision function only depends on the support vectors, removing a non-support vector from the training data and then re-training an SVM would lead to the same decision function. Also, non-support vectors must be classified correctly. As a result, errors found in the leave-one-out validation must be caused by removing the support vectors, proving the desired result.

6 Boosting [15 pts]

1. Consider training a boosting classifier using decision stumps on the following data set:

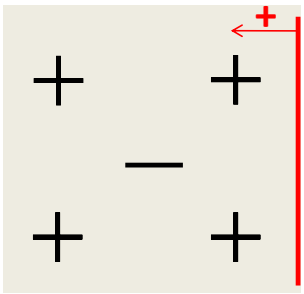


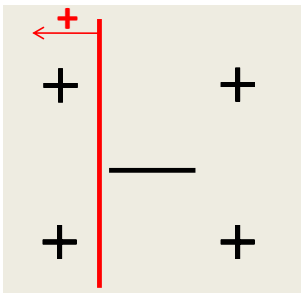
(a) **[3 pts]** Which examples will have their weights increased at the end of the first iteration? Circle them.

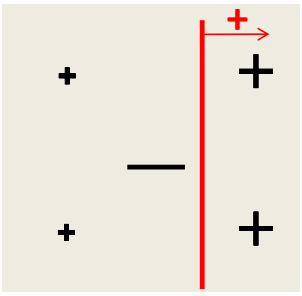
Solution: The negative example since the decision stump with least error in first iteration is constant over the whole domain. Notice this decision stump only predicts incorrectly on the negative example, whereas any other decision stump predicts incorrectly on at least two training examples.

(b) **[3 pts]** How many iterations will it take to achieve zero training error? Explain.

Solution: At least three iterations. The first iteration misclassifies the negative example, the second iteration misclassifies two of the positive examples as the negative one has large weight. The third iteration is needed since a weighted sum of the first two decision stumps can't yield zero training error, and misclassifies the other two positive examples. See Figures below.







(c) **[3 pts]** Can you add one more example to the training set so that boosting will achieve zero training error in two steps? If not, explain why.

Solution: No. Notice that the simplest case is adding one more negative example in center or one more positive example between any two positive examples, as it still yields three decision regions with axis-aligned boundaries. If only two steps were enough, then a linear combination of only two decision stumps $sign(\alpha_1 h_1(x) + \alpha_2 h_2(x))$

should be able to yield three decision regions. Also notice that at least one of h_1 or h_2 misclassifies two positive examples. If only h_2 misclassifies two positive examples, the possible decisions are (1) $sign(\alpha_1 - \alpha_2)$ on those two positive examples, (2) $sign(\alpha_1 + \alpha_2)$ on the remaining positive examples and (3) $sign(\alpha_1 - \alpha_2)$ on the negative examples - which don't yield zero training error since signs on (1) and (3) agree. If both h_1 and h_2 misclassify two positive examples, we have (1) $sign(\alpha_1 - \alpha_2)$ on two positive examples, (2) $sign(-\alpha_1 + \alpha_2)$ on the remaining positive examples and (3) $sign(-\alpha_1 - \alpha_2)$ on the negative - which again don't yield zero training error since signs on (1) and (2) don't agree.

2. [**2 pts**] Why do we want to use "weak" learners when boosting?

   Solution: To prevent overfitting, since the complexity of the overall learner increases at each step. Starting with weak learners implies the final classifier will be less likely to overfit.

3. [**4 pts**] Suppose AdaBoost is run on m training examples, and suppose on each round that the weighted training error ϵ_t of the t^{th} weak hypothesis is at most $1/2 - \gamma$, for some number $\gamma > 0$. After how many iterations, T, will the combined hypothesis H be consistent with the m training examples, i.e., achieves zero training error? Your answer should only be expressed in terms of m and γ. (Hint: What is the training error when 1 example is misclassified?)

   Solution: Training error when 1 example is misclassified = $1/m$. Therefore, we need to guarantee that training error is $< 1/m$. Since $\epsilon_t \leq 1/2 - \gamma$, from class notes we know that

   $$\text{Training err of the combined hypothesis } H \leq \exp(-2T\gamma^2)$$

   The upper bound is $< 1/m$ if $T > \ln m/2\gamma^2$.

10-601 Machine Learning
Midterm Exam
Fall 2011

Tom Mitchell, Aarti Singh
Carnegie Mellon University

1. Personal information:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be **11** numbered pages in this exam.
3. This exam is open book, open notes. No computers or internet access is allowed.
4. You do not need a calculator.
5. If you need more room to answer a question, use the back of the page and clearly mark on the front of the page if we are to look at the back.
6. Work efficiently. Answer the easier questions first.
7. You have **80** minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	35	
2	MLE/MAP	15	
3	Bayes Nets	15	
4	EM	15	
5	Regression	20	
	Total	100	

1 Short Questions [35 pts]

Answer True/False in the following 8 questions. Explain your reasoning in 1 sentence.

1. **[3 pts]** Suppose you are given a dataset of cellular images from patients with and without cancer. If you are required to train a classifier that predicts the probability that the patient has cancer, you would prefer to use Decision trees over logistic regression.

   ★ **SOLUTION:** FALSE. Decision trees only provide a label estimate, whereas logistic regression provides the probability of a label (patient has cancer) for a given input (cellular image).

2. **[3 pts]** Suppose the dataset in the previous question had 900 cancer-free images and 100 images from cancer patients. If I train a classifier which achieves 85% accuracy on this dataset, it is it a good classifier.

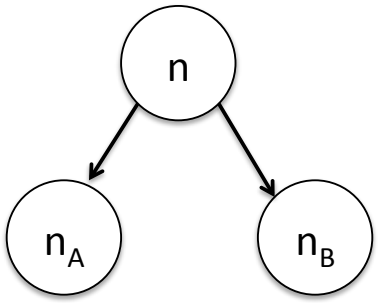
   ★ **SOLUTION:** FALSE. This is not a good accuracy on this dataset, since a classifier that outputs "cancer-free" for all input images will have better accuracy (90%).

3. **[3 pts]** A classifier that attains 100% accuracy on the training set and 70% accuracy on test set is better than a classifier that attains 70% accuracy on the training set and 75% accuracy on test set.

   ★ **SOLUTION:** FALSE. The second classifier has better test accuracy which reflects the true accuracy, whereas the first classifier is overfitting.

4. **[3 pts]** A football coach whispers a play number n to two players A and B independently. Due to crowd noise, each player imperfectly and independently draws a conclusion about what the play number was. A thinks he heard the number n_A, and B thinks he heard n_B. True or false: n_A and n_B are marginally dependent but conditionally independent given the true play number n.

   ★ **SOLUTION:** TRUE. Knowledge of n_A value tells us something about n_B therefore $P(n_A|n_B) \neq P(n_A)$ hence they are marginally dependent, but given n, n_A and n_B are determined independently. Also follows from following Bayes Net:



5. **[3 pts]** Assume m is the minimum number of training examples sufficient to guarantee that with probability $1-\delta$ a consistent learner using hypothesis space H will output a hypothesis with true error at worst ϵ. Then a second learner that uses hypothesis space H' will require $2m$ training examples (to make the same guarantee) if $|H'| = 2|H|$.

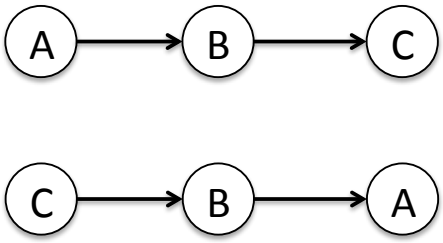
★ **SOLUTION:** FALSE. Minimum number of training examples sufficient to make an (ϵ, δ)-PAC guarantee depends logarithmically on hypothesis class size ($\ln |H|$) and not linearly.

6. **[3 pts]** If you train a linear regression estimator with only half the data, its bias is smaller.

F

★ **SOLUTION:** FALSE. Bias depends on the model you use (in this case linear regression) and not on the number of training data.

7. **[3 pts]** The following two Bayes nets encode the same set of conditional independence relations.



★ **SOLUTION:** TRUE. Both models encode that C and A are conditionally independent given B. Also

$$P(A)P(B|A)P(C|B) = \frac{P(A,B)P(B,C)}{P(B)} = P(C)P(B|C)P(A|B)$$

8. **[3 pts]** A, B and C are three Boolean random variables. The following equality holds without any assumptions on the joint distribution $P(A, B, C)$

$$P(A|B) = P(A|B, C = 0)P(C = 0) + P(A|B, C = 1)P(C = 1).$$

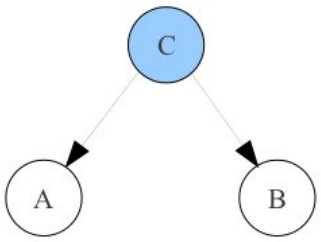
★ **SOLUTION:** TRUE. Since C is a Boolean random variable, we have

$$\begin{aligned} P(A|B) &= P(A, C = 0|B) + P(A, C = 1|B) \\ &= P(A|B, C = 0)P(C = 0) + P(A|B, C = 1)P(C = 1) \end{aligned}$$

where last step follows from definition of conditional probability.

The following three short questions are not True/False questions. Please provide explanations for your answers.

9. **[3 pts]** The Bayes net below implies that A is conditionally independent of B given C ($A \perp\!\!\!\perp B|C$). Prove this, based on its factorization of the joint distribution, and on the definition of conditional independence.



★ **SOLUTION:** Using factorization of joint distribution

$$P(A, B, C) = P(C)P(A|C)P(B|C)$$

and using definition of conditional independence

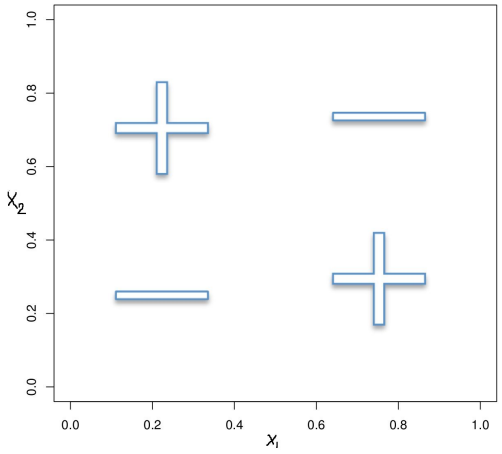
$$P(A, B, C) = P(C)P(A, B|C)$$

Therefore, we have:

$$P(A, B|C) = P(A|C)P(B|C)$$

i.e. A is conditionally independent of B given C ($A \perp\!\!\!\perp B|C$).

10. [**3 pts**] Which of the following classifiers can perfectly classify the following data:



(a) Decision Tree

(b) Logistic Regression

(c) Gaussian Naive Bayes

★ **SOLUTION:** Decision Tree only. Decision trees of depth 2 which first splits on X_1 and then on X_2 wil perfectly classify it. Logistic regression leads to linear decision boundaries, hence cannot classify this data perfectly. Due to conditional independence requirement, it is not possible to fit a Gaussian that peaks at the labels of only one class and has no covariance between features, so Gaussian Naive Bayes cannot classify this data perfectly.

11. [**5 pts**] Boolean random variables A and B have the joint distribution specified in the table below.

A	B	$P(A, B)$
0	0	0.32
0	1	0.48
1	0	0.08
1	1	0.12

Given the above table, please compute the following five quantities:

★ **SOLUTION:** $P(A = 0) = P(A = 0, B = 0) + P(A = 0, B = 1) = 0.32 + 0.48 = 0.8$

$P(A = 1) = 1 - P(A = 0) = 0.2$

$P(B = 1) = P(B = 1, A = 0) + P(B = 1, A = 1) = 0.48 + 0.12 = 0.6$

$P(B = 0) = 1 - P(B = 1) = 0.4$

$P(A = 1|B = 0) = P(A = 1, B = 0)/P(B = 0) = 0.08/0.4 = 0.2$

Are A and B independent? Justify your answer.

★ **SOLUTION:** YES. Using the calculations above,

$$P(A = 0)P(B = 0) = 0.8 * 0.4 = 0.32 = P(A = 0, B = 0)$$
$$P(A = 0)P(B = 1) = 0.8 * 0.6 = 0.48 = P(A = 0, B = 1)$$
$$P(A = 1)P(B = 0) = 0.2 * 0.4 = 0.08 = P(A = 1, B = 0)$$
$$P(A = 1)P(B = 1) = 0.2 * 0.6 = 0.12 = P(A = 1, B = 1)$$

2 MLE/MAP Estimation [15 pts]

In this question you will estimate the probability of a coin landing heads using MLE and MAP estimates.

Suppose you have a coin whose probability of landing heads is $p = 0.5$, that is, it is a fair coin. However, you do not know p and would like to form an estimator $\hat{\theta}$ for the probability of landing heads p. In class, we derived an estimator that assumed p can take on any value in the interval $[0, 1]$. In this question, you will derive an estimator that assumes p can take on only two possible values: 0.3 or 0.6.

Note: $P_{\hat{\theta}}[\text{heads}] = \hat{\theta}$.

Hint: All the calculations involved here are simple. You do not require a calculator.

1. **[5 pts]** You flip the coin 3 times and note that it landed 2 times on tails and 1 time on heads. Find the maximum likelihood estimate $\hat{\theta}$ of p over the set of possible values $\{0.3, 0.6\}$.

   Solution:

   $$\begin{aligned}\hat{\theta} &= \text{argmax}_{\theta \in \{0.3, 0.6\}}\ P_\theta[D] \\ &= \text{argmax}_{\theta \in \{0.3, 0.6\}}\ P_\theta[\text{heads}] P_\theta[\text{tails}]^2 \\ &= \text{argmax}_{\theta \in \{0.3, 0.6\}}\ \theta(1-\theta)^2\end{aligned}$$

   We observe that

   $$\frac{P_{\theta=0.3}[D]}{P_{\theta=0.6}D]} = \frac{0.3 * 0.7^2}{0.6 * 0.4^2} = \frac{0.49}{0.32} > 1$$

   which implies that $\hat{\theta} = 0.3$.

2. **[4 pts]** Suppose that you have the following prior on the parameter p:

   $$P[p = 0.3] = 0.3 \quad \text{and} \quad P[p = 0.6] = 0.7.$$

   Again, you flip the coin 3 times and note that it landed 2 times on tails and 1 time on heads. Find the MAP estimate $\hat{\theta}$ of p over the set $\{0.3, 0.6\}$, using this prior.

   Solution:

   $$\hat{\theta} = \text{argmax}_{\theta \in \{0.3, 0.6\}}\ P_\theta[D] P[\theta]$$

   We observe that

   $$\frac{P_{\theta=0.3}[D] P[\theta = 0.3]}{P_{\theta=0.6}[D] P[\theta = 0.6]} = \frac{0.3 * 0.7^2 * 0.3}{0.6 * 0.4^2 * 0.7} = \frac{0.21}{0.32} < 1$$

   which implies that $\hat{\theta}_{\text{MAP}} = 0.6$.

3. **[3 pts]** Suppose that the number of times you flip the coin tends to infinity. What would be the maximum likelihood estimate $\hat{\theta}$ of p over the set $\{0.3, 0.6\}$ in that case? Justify your answer.

   Solution:
   With the number of flips tending to infinity, proportion of heads to the total number of flips tends to 0.5. The MLE would be 0.6 as this is closer to 0.5.

4. **[3 pts]** Suppose that the number of times you flip the coin tends to infinity. What would be the MAP estimate $\hat{\theta}$ of p over the set $\{0.3, 0.6\}$, using the prior defined in part 2 of this question? Justify your answer.

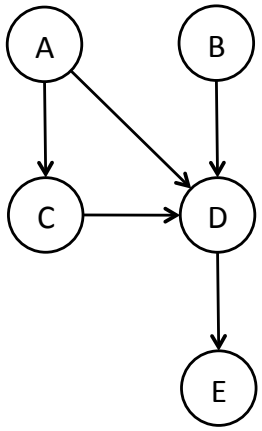
   Solution:
   With the number of flips tending to infinity, the effect of the prior becomes negligible. Therefore, the MAP estimate will be the same as the MLE.

3 Bayes Nets [15 pts]

1. (a) **[3 pts]** Please draw a Bayes net which represents the following joint distribution:

$$P(A, B, C, D, E) = P(A)P(B)P(C|A)P(D|A, B, C)P(E|D)$$

Solution:



(b) **[2 pts]** For the graph that you drew above, assume each variable can take on the values 1, 2 or 3. Also assume that you are given values for the probabilities $P(D = 1)$, $P(D = 2)$ and $P(D = 3)$. Please specify the smallest set of Bayes net parameters you would need in order to calculate $P(E = 1)$. *Solution:*
We can write:

$$\begin{aligned} P(E = 1) =& P(E = 1|D = 1)P(D = 1) + P(E = 1|D = 2)P(D = 2) + \dots \\ & P(E = 1|D = 3)P(D = 3) \end{aligned}$$

Thus we need three parameters: $P(E = 1|D = 1)$, $P(E = 1|D = 2)$ and $P(E = 1|D = 3)$.

2. **[4 pts]** Please draw a single Bayes net which encodes all the following conditional independence assumptions over the variables A, B, C and D:

   (a) A is independent of D given B and C

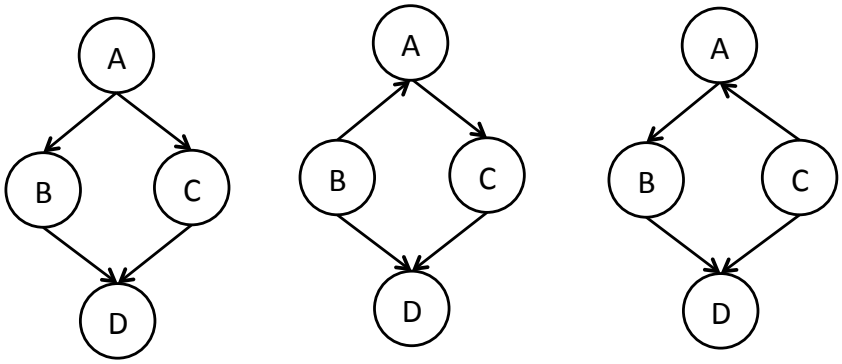
   (b) A is *not* independent of D given only B

   (c) A is *not* independent of D given only C

(d) B is independent of C given only A

(e) B is *not* independent of C given A and D

Solution: Any of the following satisfy the above:



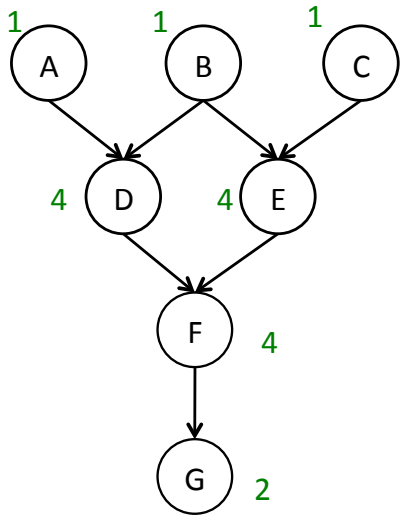
3. [**4 pts**] Consider the graph drawn below. Assume that each variable can only take on values *true* and *false*.

   (a) How many parameters are necessary to specify the joint distribution $P(A, B, C, D, E, F, G)$ for this Bayes net? You may answer by writing the number of parameters directly next to each graph node.

   Solution: See below for the number of parameters needed for each node. Total is 17.

   (b) Please give the **minimum** number of Bayes net parameters required to fully specify the distribution $P(G|A, B, C, D, E, F)$. Briefly justify your answer.

   Solution: Note that the Markov blanket for G consists only of F. Thus, $P(G|A, B, C, D, E, F) = P(G|F)$ and only two parameters are need to specify this distribution.



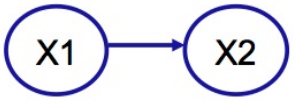
4. [**2 pts**] Given the graph provided above, please state if the following are **true** or **false**.

   (a) E is conditionally independent of G given F. *Solution:* True.

   (b) A is conditionally independent of C given B and G. *Solution:* False.

4 EM [15 pts]

In this question you will apply EM to train the following simple Bayes net:



using the following data set, for which $X2$ is unobserved in training example 4.

Example	$X1$	$X2$
1.	0	1
2.	0	0
3.	1	0
4.	1	?
5.	0	1

The EM process has run for several iterations. At this point the parameter estimates are:

$$\hat{\theta}_{X1=1} = \hat{P}(X1 = 1) = 0.4$$
$$\hat{\theta}_{X2=1|X1=1} = \hat{P}(X2 = 1|X1 = 1) = 0.4$$
$$\hat{\theta}_{X2=1|X1=0} = \hat{P}(X2 = 1|X1 = 0) = 0.66$$

1. **[2 pts]** What is calculated in the next E step?

   Answer: The expected value of $X2$ for example 4: $P(X2 = 1|X1 = 1; \theta)$

2. **[5 pts]** What precisely is the result of the next E step? Show your work.

$$\hat{P}(X2 = 1|X1 = 1) = \hat{\theta}_{X2=1|X1=1} = 0.4$$

3. **[3 pts]** What is calculated in the next M step?

   New estimates for $\hat{\theta}_{X1=1}$, $\hat{\theta}_{X2=1|X1=0}$ (which do not change), and $\hat{\theta}_{X2=1|X1=1}$

4. **[5 pts]** What precisely is the result of the next M step? Show your work.

$$\hat{\theta}_{X1=1} = \frac{2}{5} = 0.4$$
$$\hat{\theta}_{X2=1|X1=0} = \frac{2}{3} = 0.66$$
$$\hat{\theta}_{X2=1|X1=1} = \frac{0.4}{2} = 0.2$$

5 Bias and Variance in Linear Regression [20 pts]

In this question, we will explore bias and variance in linear regression. Assume that a total of N data points of the form (x_i, y_i) are generated from the following (true) model:

$$x_i \sim Unif(0,1), \quad y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0,1), \quad f(x) = x$$

We assume $x_i \perp \epsilon_j \; \forall i, j$ and $\epsilon_i \perp \epsilon_j \; \forall i \neq j$ (note $a \perp b$ means a and b are independent).

You may find the following pieces of information useful when solving this problem:

- $bias^2 = \int_x (E_D[h_D(x)] - f(x))^2 p(x) dx$
- $variance = \int_x E_D[(h_D(x) - E_D[h_D(x)])^2] p(x) dx$
- $\hat{\mu} \sim N(\mu, \frac{1}{N})$ if $\hat{\mu}$ is the MLE estimator with N data points
- If $x \sim Unif(0,1)$, then $\int_0^1 p(x)dx \; = 1$, and therefore $p(x) = 1$.

We begin by examining the case where we are not aware that y depends on x. Instead, our (incorrect) model is that $f(x)$ has some constant value $f(x) = \mu$, and therefore

$$x_i \sim Unif(0,1), \quad y_i \sim N(\mu, 1) \text{ with } x_i \perp y_i.$$

We use the MLE estimator for μ. That is, we let $\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} y_i$. The prediction of our trivial regression model for the value of y_i is $\hat{\mu}$, regardless of the value of x_i.

1. **[2 pts]** What is the value for $E_D[h_D(x)]$ in this case? Here E_D refers to the expected value over different training data sets of size N, and $h_D(x)$ is the predictor learned from a specific data set D.

   ★ **SOLUTION:** $E_D[h_D(x)] = E_D[\hat{\mu}] = \frac{1}{2}$

2. **[3 pts]** What is the bias of this trivial regression model?

   ★ **SOLUTION:** $Bias^2 = \int_0^1 (\frac{1}{2} - x)^2 (1) \; dx = -\frac{1}{3}(\frac{1}{2} - x)^3 |_0^1 = \frac{1}{12}$.
   The bias is thus $\sqrt{\frac{1}{12}}$.

3. **[2 pts]** What is the variance of this trivial regression model?

★ **SOLUTION:** The variance is the variance of the MLE estimator. By the third bullet, this is $\frac{1}{N}$.

4. [**1 pts**] What is the unavoidable error in this learning setting?

   ★ **SOLUTION:** The unavoidable error is introduced by ϵ_i, and is 1 by assumption.

5. [**2 pts**] How do each of bias, variance, and unavoidable error change as $N \to \infty$?

   ★ **SOLUTION:** The unavoidable error and bias do not change. The variance goes to 0 as $N \to \infty$.

Now assume we notice that y in fact depends on x. Therefore, we change to a linear regression model (with zero intercept), which assumes the data are generated as follows:

$$x_i \sim Unif(0,1), \quad y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0,1), \quad f(x) = ax$$

We also assume (as in the true model) that $x_i \perp \epsilon_j \ \forall i, j$ and $\epsilon_i \perp \epsilon_j \ \forall i \neq j$.

6. **[3 pts]** We choose our estimator $\hat{a}$ for a to minimize the squared sum of errors. That is, we choose $\hat{a}$ such that

$$\hat{a} = argmin_a \ \frac{1}{2} \sum_{i=1}^{N} (y_i - ax_i)^2$$

Derive the closed form expression for $\hat{a}$. Once we have chosen the value of $\hat{a}$, we now have a regression model that predicts $y_i = \hat{a}x_i$.

★ **SOLUTION:** Let $f(a) = \frac{1}{2} \sum_{i=1}^{N} (y_i - ax_i)^2$. Then,

$\frac{\partial(f)}{\partial a} = \sum_{i=1}^{N} -x_i(y_i - ax_i)$

Setting the derivative to 0, we obtain:

$\sum_{i=1}^{N} -x_i(y_i - ax_i) = 0$

$\implies \sum_{i=1}^{N} x_i y_i = \sum_{i=1}^{N} a x_i^2$

$$\implies \hat{a} = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2}.$$

7. **[2 pts]** What is the bias of this linear regression model?

★ **SOLUTION:** The bias of the regression model is 0.

8. **[2 pts]** As $N \to \infty$, what is the variance of this linear regression model?

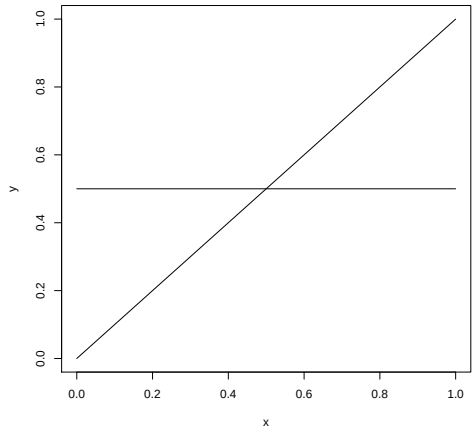
★ **SOLUTION:** The variance of the linear regression models goes to 0 as $N \to \infty$.

9. **[1 pts]** What is the unavoidable error in this learning setting?

★ **SOLUTION:** The unavoidable error is still introduced by ϵ_i, and is 1 by assumption.

10. [**2 pts**] In the figure below, draw the two learned regression models if we have an infinite number of data points.

★ **SOLUTION:** Model 1 (the trivial model) is the horizontal line. Model 2 (the linear regression model) is the diagonal line.



Question 1 – Probability and estimation (12 points)

a) (4 points) Let a stick X_0 of unit length be broken at random at any position along the length with uniform probability. Let X_1 be the bigger piece. What is the expected length of X_1 ?

The stick is broken at position $x \sim U(0,1)$.

$\therefore$ Expected length $= E_x[\max(x, 1-x)]$

$$= \int_0^{1/2} (1-x) \cdot f_x(x)\,dx + \int_{1/2}^{1} x \cdot f_x(x)\,dx$$

$$= \int_0^{1/2} (1-x) \cdot 1 \cdot dx + \int_{1/2}^{1} x \cdot 1 \cdot dx$$

$$= \left[x - \frac{x^2}{2}\right]_0^{1/2} + \left[\frac{x^2}{2}\right]_{1/2}^{1}$$

$$= \frac{1}{2} - \frac{1}{8} + \frac{1}{2} - \frac{1}{8} = \frac{3}{4}$$

b) (4 points) Let a loss function be defined as the expected squared discrepancy between the actual and estimated value of the parameter, i.e., $g(\theta,\hat{\theta}) = E\left[(\theta-\hat{\theta})^2\right]$

and assume that we sample i.i.d. from the following distribution: $X_1, X_2, \ldots, X_n \sim N(\theta,1)$

Let $\hat{\theta_1}=X_1$ be an estimator which estimates the mean as the value of the first sample. What is the loss for this estimator?

$$g(\theta,\hat{\theta}_1) = E\left[(X_1-\theta)^2\right] = \sigma^2 \quad \left[\text{where } \sigma^2 = \text{variance of the normal distrib., where } X_1 \sim N(\theta,\sigma^2) \text{ from defn.}\right]$$

$$= 1 \quad [\because \sigma^2 = 1]$$

c) (4 points) Let $\hat{\theta}_2 = \dfrac{\sum_i x_i}{n}$ be another estimator - the sample mean. What is the loss function for this estimator? [Hint: For any variable or function $E(y^2) = E(y)^2 + Var(y)$]

$$g(\theta,\hat{\theta}_2) = E\left[(M_n-\theta)^2\right] \quad (\text{let } M_n = \text{sample mean on } n \text{ samples})$$

$$= \left[E(M_n-\theta)\right]^2 + Var(M_n-\theta) \quad (\text{from corollary above})$$

$$= 0 + Var(M_n-\theta) \quad (\because \text{sample mean is unbiased estimator of } \theta)$$

$$= Var(M_n) - Var(\theta)$$

$$= Var\left(\frac{X_1+X_2+\cdots+X_n}{n}\right) - 0 = \frac{1}{n^2} \cdot n\,Var(X_1) \quad [\because \text{iid}]$$

$$= \frac{1}{n^2} \cdot 1 = \frac{1}{n^2}$$

Question 2 – Naïve Bayes (16 points)

About 2/3 of your email is spam so you downloaded an open source spam filter based on word occurrences that uses the Naive Bayes classifier. Assume you collected the following regular and spam mails to train the classifier, and only three words are informative for this classification, i.e., each email is represented as a 3-dimensional binary vector whose components indicate whether the respective word is contained in the email.

'study'	'free'	'money'	Category
1	0	0	Regular
0	0	1	Regular
1	0	0	Regular
1	1	0	Regular
0	1	0	Spam
0	1	0	Spam
0	1	0	Spam
0	1	0	Spam
0	1	1	Spam
0	1	1	Spam
0	1	1	Spam
0	1	1	Spam

a) (2 points) You find that the spam filter uses a prior p(spam) = 0.1. Explain (in one sentence) why this might be sensible.

Answer: It is worse for regular emails to be classified as spam than it is for spam email to be classified as regular email.

b) (4 points) Give the following model parameters when estimated as maximum-likelihood with add-one smoothing (i.e., using pseudocounts of one).

P(study|spam) = 1/10
P(study|regular) = 2/3
P(free|spam) = 9/10
P(free|regular) = 1/3
P(money|spam) = 1/2
P(money|regular) = 1/3

c) (5 points) Based on the prior and conditional probabilities above, give the model probability P(spam|***s***) that the sentence ***s***="money for psychology study" is spam.

Answer: p(spam) P(study|spam) (1-P(free|spam)) (P(money|spam)) =
p(spam) * 1/200
(1-p(spam)) P(study|regular) (1-P(free|regular)) (P(money|regular)) = (1-p(spam))*4/27
P(spam|s)= p(spam)/200 / [p(spam)/200 + (1-p(spam))*4/27]
= p(spam)/200 / [p(spam)/200 + 4/27 - p(spam)*4/27]
= p(spam)/200 / [4/27 - p(spam)*773/5400] = .003736

d) (5 points) What should be the value of the prior p(spam) if we would like the above sentence to have the same probability as being spam as not spam, i.e., it would be classified as spam with probability 0.5?

0.5 = p(spam)/200 / [4/27 - p(spam)*773/5400] iff
4/27 - p(spam)*773/5400 = p(spam)/100 iff p(spam) = .9674

Question 3 – Regression (8 points)

We are dealing with samples x where x is a single value. We would like to test two alternative regression models:

1. $y = ax + e$
2. $y = ax + bx^2 + e$

We make the same assumptions we had in class about the distribution of e ($e \sim N(0,s^2)$).

a. (4 points) Assume we have n samples: $x_1 \ldots x_n$. with their corresponding y values, : $y_1 \ldots y_n$. Derive the value assigned to b in model 2. You can use a in the equation for b.

$$b = \frac{\sum_i y_i x_i^2 - a x_i^3}{\sum_i x_i^4}$$

b. (2 points) Which of the two models is more likely to fit the *training* data better?

   a. model 1
   b. model 2
   c. both will fit equally well
   d. impossible to tell

Answer: b. (model 2). Since it has more parameters it is likely to provide a better fit for the training data.

c. (2 points) Which of the two models is more likely to fit the *test* data better?

   a. model 1
   b. model 2
   c. both will fit equally well
   d. impossible to tell

Answer: d. It depends on the underlying model of the data and the amount of data available for training. If the data indeed comes from a linear model and we do not have a lot of data to train on model 2 will lead to overfitting and model 1 would do better. On the other hand if the data comes from an underlying quadratic model, model 2 would be better.

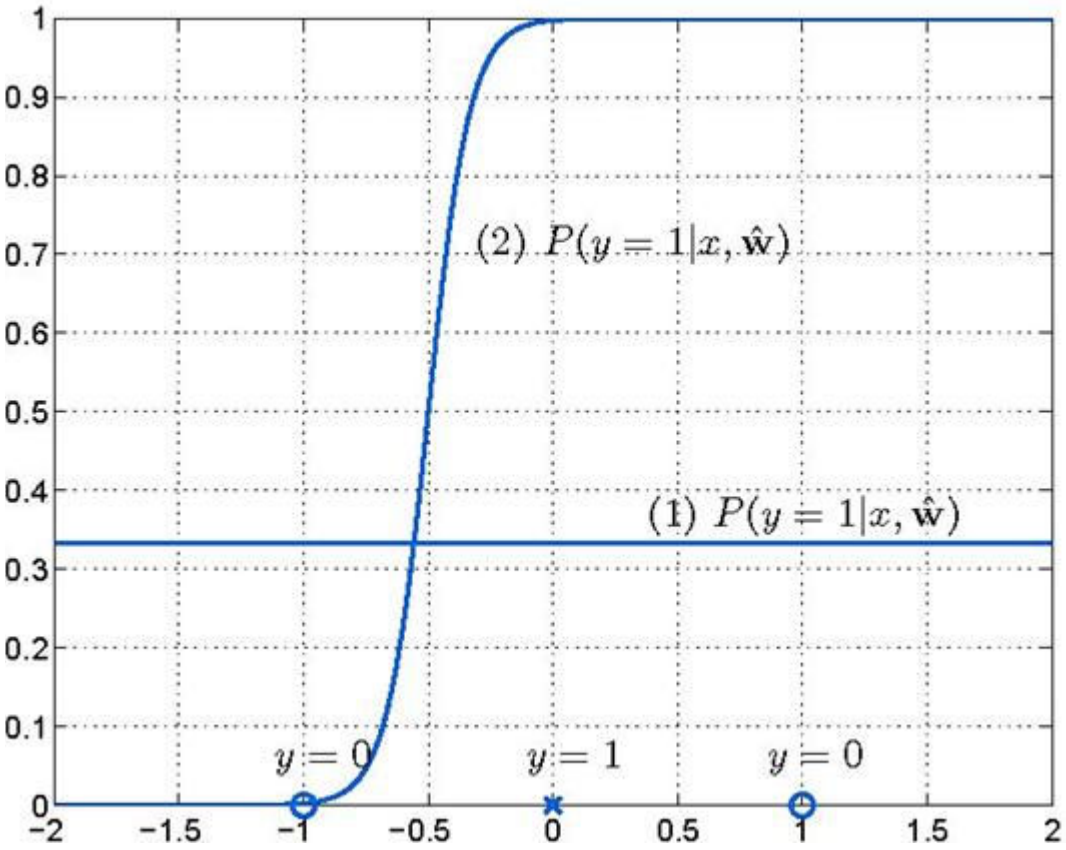
Question 4 – Logistic Regression (12 points)

Consider a simple one dimensional logistic regression model

$$P(y=1|x, \mathrm{w}) = g(w_0 + w_1 x)$$

where g(z) = 1/(1+exp(z)) is the logistic function.

The following figure shows two possible conditional distributions P (y =1|x; w), viewed as a function of x, that we can get by changing the parameters w.



(a) (4 points) Please indicate the number of classification errors for each conditional given the labeled examples in the same figure.

Conditional (1) makes (1) classification errors

Conditional (2) makes (1) classification errors

(b) (4 points) One of the two classifiers corresponds to the maximum likelihood setting of the parameters w based on the labeled data in the figure, i.e. its parameters maximize the joint probability

P(y=0|x=-1; w) P(y=1|x=0; w) P(y=0|x=1; w)

Circle which one is the ML solution and briefly explain why you chose it:

Classifier 1 or Classifier 2

Answer: Class. 1 b/c it can't be classifier 2, for which P(y=0|x-1)=0

(c) (4 points) Would adding a regularization penalty $|w_1|^2 / 2$ to the log-likelihood estimation criterion affect your choice of solution (Y/N)? (Note that the penalty above only regularizes w_1, not w_0.)? Briefly explain why.

Answer: no, because w_1 is zero for Classifier 1, so no penalty is incurred. Therefore, if it was the ML solution before, it must still be the ML solution.

Question 5 – Decision Trees (14 points)

Decision trees

a. (2 points) What is the biggest advantage of decision trees when compared to logistic regression classifiers?

Answer: Decision trees do not assume independence of the input features and can thus encode complicated formulas related to relationship between these variables whereas logistic regression treats each feature independently.

b. (2 points) What is the biggest weakness of decision trees compared to logistic regression classifiers?

Answer: Decision trees are more likely to overfit the data since they can split on many different combination of features whereas in logistic regression we associate only one parameter with each feature.

For the next problem consider *n* two dimensional vectors ($x = \{x_1, x_2\}$) that can be classified using a regression classifier. That is, there exists a w such that

$$\begin{cases} y = +1 & \text{if } w^T x+b>0 \\ \quad -1 & \text{if } w^T x+b \pounds 0 \end{cases}$$

c. (5 points) Can a decision correctly classify these vectors? If so, what is an upper bound on the depth of the corresponding decision tree (as tight as possible)? If not, why not?

Answer: Yes. One possible strategy is to split the points according to their x_1 values. Since the data is linearly separable, for each x_1 value there is a cutoff on x_2 so that values above it are in class 1 and below it in class -1. Splitting the data based on the x_1 values can be done with a log(n) depth tree and then we only need at most one more node to correctly classify all points so the total depth is *O(log n)*.

d. (5 points) Now assume that these n inputs are not linearly separable (that is, no w exists for correctly classifying all inputs using linear regression classifier). Can a decision tree correctly classify these vectors? If so, what is an upper bound on the depth of the corresponding decision tree (as tight as possible)? If not, why not?

Answer: Yes. Similar to what we did for c we can split the points according to their x_1 values. However, since they are not linearly separable we cannot assume a cutoff on x_2 anymore. Instead, we may need to consider different values for x_2 again, the can be done in at most log(n) depth for a total of 2log(n) and an O(log n) depth.

Question 6 – Neural Networks (15 points)

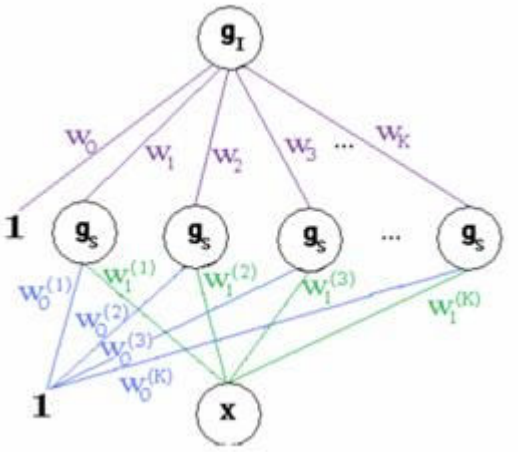
Suppose that you have two types of activation functions at hand:

- Identity function: $g_I(x) = x$
- Step function: $g_s(x)=1$ if x >= 0, 0 otherwise

So, for example, the output of a neural network with one input x, a single hidden layer with K units having step function activations, and a single output with identity activation can be written as

$$out(x) = g_I\,(w_0 + \Sigma_i\, w_i\, g_S\,(w_0^{(i)} + w_1^{(i)}\, x))$$

and can be drawn as follows:



1. (7 points) Consider the step function: u(x)=c if x<a, 0 otherwise (where a and c are fixed real-valued constants). Construct a neural network with one input x and one hidden layer whose response is u(x). Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_S), and specify the values for all weights (in terms of a and c).
   Answer: $g_I\,(c - c*g_s(x-a))$

   No points deduced for this (though not correct for x=a): $g_I\,(c*g_s(a-x))$

2. (8 points) Now, construct a neural network with one input x and one hidden layer whose response for fixed real-valued constants a, b and c is
   c if x ϵ [a, b), and 0 otherwise.

   Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_S), and specify the values for all weights (in terms of a, b, and c).

   Answer: $g_I\,[\,c*g_s\,(x-a) - c*g_s(x-b)\,]$

Question 7 – Learning Theory (12 points)

Suppose you want to use a Boolean function to pick spam emails. Each email has n = 10 binary features (e.g. contains/ does not contain the keyword "sale").

a) Suppose the emails are generated by some unknown Boolean function of the n binary features (if the outcome of the boolean function is 1, it generates a spam otherwise regular emails).

Question:

i) (2 points) If our hypothesis space is all Boolean function, what is the error of the best hypothesis in our space?

Answer: 0.

Since the hypothesis space contains the true concept.

ii) (4 points) How many sample emails are sufficient to get a Boolean function with probability at least 95% that its error is less than 10%?

Answer: 7128

$$m \geq \frac{1}{\varepsilon}(\ln|H| + \ln\frac{1}{\delta})$$

$$\varepsilon = 0.1, \quad \delta = 0.05, \quad |H| = 2^{2^n} = 2^{1024}$$

$$m \geq 7127.8$$

b) Suppose the emails are generated by the following process: 75% of the emails are generated by Boolean function as described above. 25% of the emails are just random ones (randomly spam or not).

Question:

i) (2 points) If our hypothesis space is all Boolean function, what is the error of the best hypothesis in our space?

Answer: 12.5%

Since the accuracy will be 75% + 25% / 2 = 87.5%

ii) (4 points) How many sample emails are sufficient to get a Boolean function with probability at least 95% that its error is less than 15%?

Answer: 570223

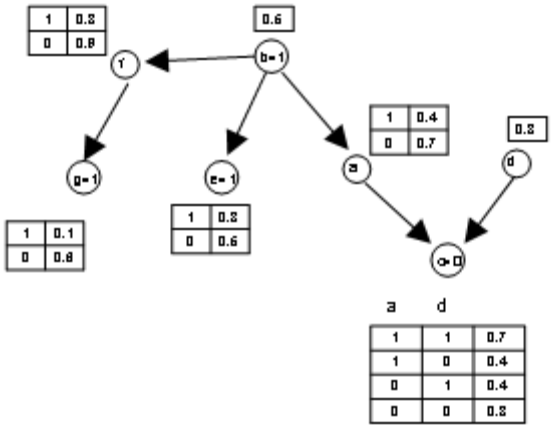
$$m \geq \frac{1}{2\varepsilon^2}(\ln|H| + \ln\frac{1}{\delta})$$

$$\varepsilon = 0.15 - 0.125 = 0.025, \quad \delta = 0.05, \quad |H| = 2^{2^n} = 2^{1024}$$

$$m \geq 570222.7$$

Question 8 – Bayesian Networks (11 points)

a. (6 points) The Bayesian network in the following figure contains both observed (denoted by 0 or 1 on the corresponding nodes) and unobserved variables.



All variables are binary. The CPTs for the probability of 1 for each variable are provided.
For example, p(c = 1| parent =1) = 0.3
p(c = 1| parent =0) = 0.9

Given the values observed, what is the value of p(a=1)?

Answer: Note that a is conditionally independent of all other nodes given its Markov blanket. Thus:

P(a=1|value of all nodes) = p(a=1|b,c,d)=

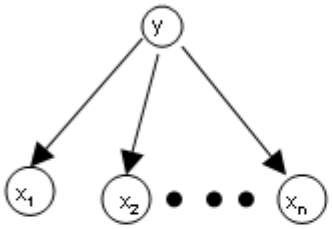
$$\frac{\sum_d p(a=1,b=1,c=0,d)}{\sum_d p(a=1,b=1,c=0,d)+\sum_d p(a=0,b=1,c=0,d)}$$

$$=\frac{0.5*0.4*(0.3*0.3+0.6*0.7)}{0.5*0.4*(0.3*0.3+0.6*0.7)+0.5*0.6*(0.6*0.3+0.2*0.7)}=0.515$$

b. (5 points) Assume we are using Naïve Bayes for classifying an n dimensional vector

$x = \{x_1 \ldots x_n\}$

that can belong to one of two classes (0 or 1). Draw the graph for corresponding Bayesian network. Note that there should be n+1 variables in your network. There is no need to provide any CPT.



10-601 Machine Learning, Midterm Exam

Instructors: Tom Mitchell, Ziv Bar-Joseph

Wednesday 12th December, 2012

There are 9 questions, for a total of 100 points.
This exam has 20 pages, make sure you have all pages before you begin.
This exam is open book, open notes, but *no computers or other electronic devices*.

This exam is challenging, but don't worry because we will grade on a curve. Work efficiently.

Good luck!

Name: _____

Andrew ID: _____

Question	Points	Score
Short Answers	11	
GMM - Gamma Mixture Model	10	
Decision trees and Hierarchical clustering	8	
D-separation	9	
HMM	12	
Markov Decision Process	12	
SVM	12	
Boosting	14	
Model Selection	12	
Total:	100	

Question 1. **Short Answers**

(a) [3 points] For data D and hypothesis H, say whether or not the following equations must always be true.

- $\sum_h P(H = h|D = d) = 1$ ... is this always true?

Solution:
yes

- $\sum_h P(D = d|H = h) = 1$ ... is this always true?

Solution:
no

- $\sum_h P(D = d|H = h)P(H = h) = 1$ ... is this always true?

Solution:
no

(b) [2 points] For the following equations, describe the relationship between them. Write one of four answers:

(1) "=" (2) "$\leq$" (3) "$\geq$" (4) "(depends)"

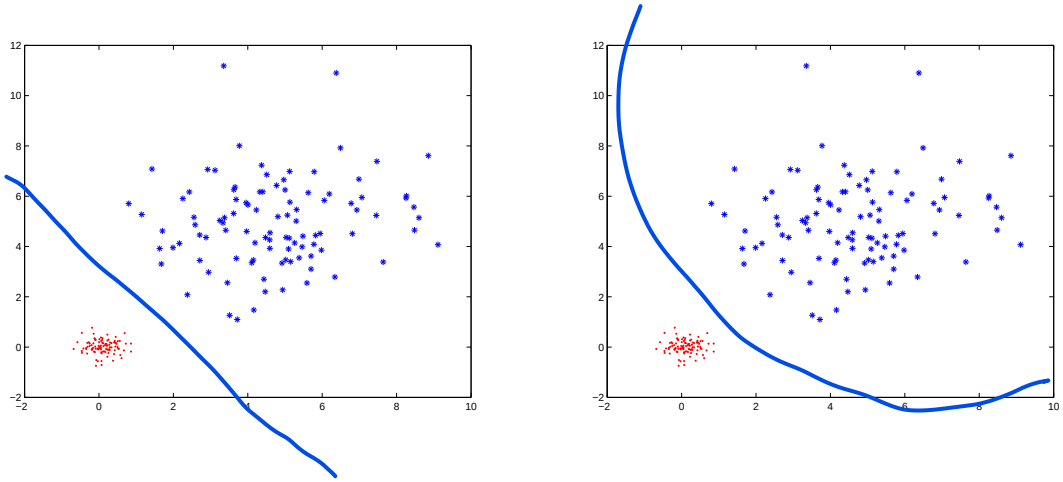
Choose the most specific relation that always holds; "(depends)" is the least specific. Assume all probabilities are non-zero.

$P(H = h\|D = d)$	$P(H = h)$
$P(H = h\|D = d)$	$P(D = d\|H = h)P(H = h)$

Solution:
$P(H|D)$ (DEPENDS) $P(H)$
$P(H|D) \geq P(D|H)P(H)$ .. this is the numerator in Bayes Rule, have to divide by the normalizer $P(D)$, which is less than 1. Tricky... $P(H|D) = P(D|H)P(H)/P(D) > P(D|H)P(H)$.

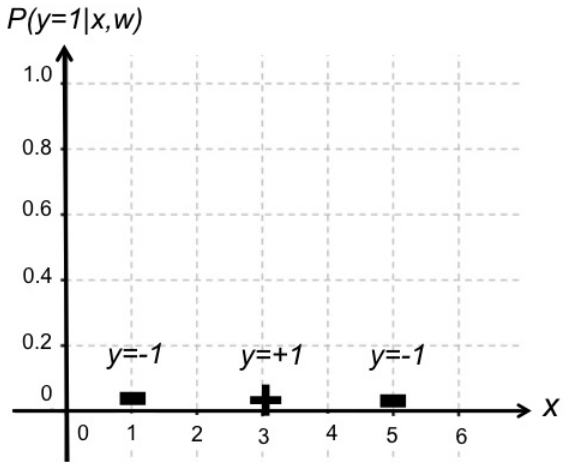
(c) [2 points] Suppose you are training Gaussian Naive Bayes (GNB) on the training set shown below. The dataset satisfies Gaussian Naive Bayes assumptions. Assume that the variance is independent of instances but dependent on classes, i.e. $\sigma_{ik} = \sigma_k$ where i indexes instances $X^{(i)}$ and $k \in 1, 2$ indexes classes. Draw the decision boundaries when you train GNB

a. using the **same** variance for both classes, $\sigma_1 = \sigma_2$
b. using separate variance for each class $\sigma_1 \neq \sigma_2$



Solution:
The decision boundary for part a will be linear, and part b will be quadratic.

(d) [2 points] Assume that we have two possible conditional distributions ($P(y = 1|x, w)$) obtained by training a logistic regression on the dataset shown in the figure below:

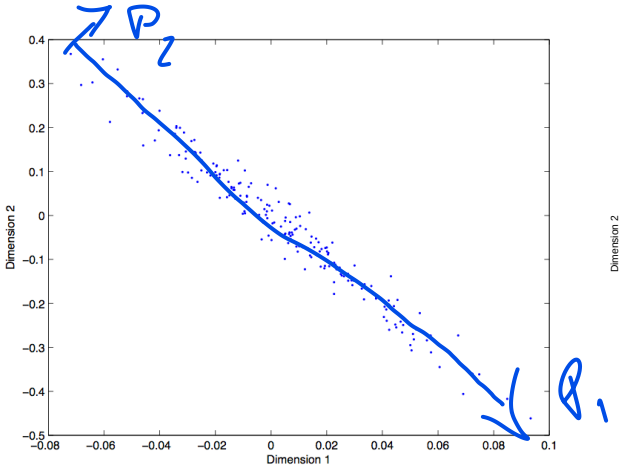


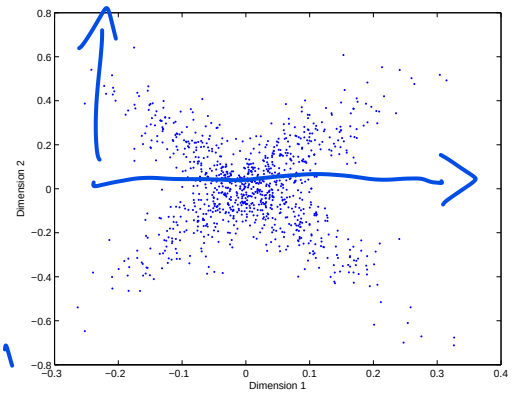
In the first case, the value of $P(y = 1|x, w)$ is equal to 1/3 for all the data points. In the second case, $P(y = 1|x, w)$ is equal to zero for $x = 1$ and is equal to 1 for all other data points. One of these conditional distributions is obtained by finding the maximum likelihood of the parameter w. Which one is the MLE solution? Justify your answer in at most three sentences.

↳ Sum whose $\frac{1}{N} = \frac{1}{3}$

Solution:
The MLE solution is the first case where the value of $P(y = 1|x, w)$ is equal to 1/3 for all the data points.

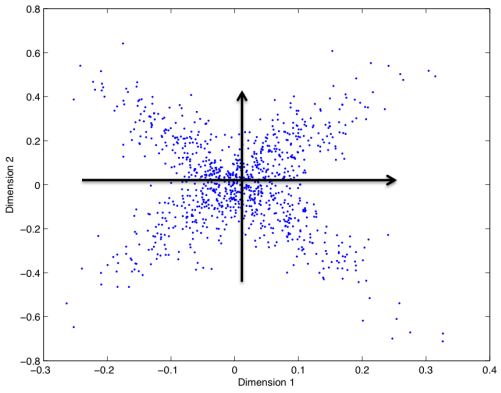
(e) [2 points] Principal component analysis is a dimensionality reduction method that projects a dataset into its most variable components. You are given the following 2D datasets, draw the first and second principle components on each plot.

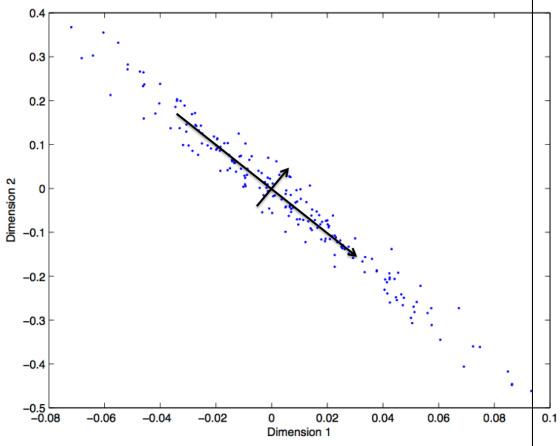




← Min reconstruction error

Solution:





Question 2. **GMM - Gamma Mixture Model**

A Assume each data point $X_i \in \mathbb{R}^+$ $(i = 1...n)$ is drawn from the following process:

$$Z_i \sim Multinomial(\pi_1, \pi_2, ..., \pi_K)$$

$$X_i \sim Gamma(2, \beta_{Z_i})$$

The probability density function of $Gamma(2, \beta)$ is $P(X = x) = \beta^2 x e^{-\beta x}$.

(a) [3 points] Assume $K = 3$ and $\beta_1 = 1$, $\beta_2 = 2$, $\beta_3 = 4$. What's $P(Z = 1|X = 1)$?

Solution:
$P(Z = 1|X = 1) \propto P(X = 1|Z = 1)P(Z = 1) = \pi_1 e^{-1}$
$P(Z = 2|X = 1) \propto P(X = 1|Z = 2)P(Z = 2) = \pi_2 4e^{-2}$
$P(Z = 3|X = 1) \propto P(X = 1|Z = 3)P(Z = 3) = \pi_3 16e^{-4}$

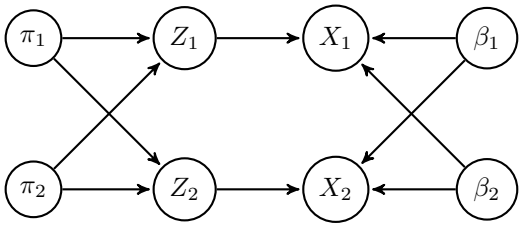
$P(Z = 1|X = 1) = \frac{\pi_1 e^{-1}}{(\pi_1 e^{-1} + \pi_2 4e^{-2} + \pi_3 16e^{-4})}$

(b) [3 points] Describe the E-step. Write an equation for each value being computed.

Solution:
For each $X = x$,

$$P(Z = k|X = x) = \frac{P(X = x|Z = k)P(Z = k)}{\sum_{k'} P(X = x|Z = k')P(Z = k')} = \frac{\beta_k^2 x e^{-\beta_k x}\pi_k}{\sum_{k'} \beta_{k'}^2 x e^{-\beta_{k'} x}\pi_{k'}}$$

(c) [2 points] Here's the Bayes net representation of the Gamma mixture model for $k = n = 2$. Note that we are treating π's and β's as variables – we have priors for them.



Would you say π's are independent given the observations X? Why?

Solution:
No. $\pi_1 \to Z_1 \to \pi_2$ is an active trail since X is given.

(d) For the following parts, choose true or false with an explanation in **one sentence**

i. [1 point] Gamma mixture model can capture overlapping clusters, like Gaussian mixture model.

Solution:
(All or none. 1 pt iff you get the answer and the explanation correct) true. in the e-step it does soft assignment

ii. [1 point] As you increase K, you will **always** get better likelihood of the data.

Solution:
(All or none. 1 pt iff you get the answer and the explanation correct) false. Won't improve after $K > N$

Question 3. **Decision trees and Hierarchical clustering**

Assume we are trying to learn a decision tree. Our input data consists of N samples, each with k attributes ($N \gg k$). We define the depth of a tree as the maximum number of nodes between the root and any of the leaf nodes (including the leaf, not the root).

(a) [2 points] If all attributes are binary, what is the maximal number of leaf (decision) nodes that we can have in a decision tree for this data? What is the maximal possible depth of a decision tree for this data?

Solution:
$2^{(k-1)}$. Each feature can only be used once in each path from root to leaf. The maximum depth is O(k).

(b) [2 points] If all attributes are continuous, what is the maximum number of leaf nodes that we can have in a decision tree for this data? What is the maximal possible depth for a decision tree for this data?

Solution:
Continuous values can be used multiple times, so the maximum number of leaf nodes can be the same as the number of samples, N and the maximal depth can also be N.

(c) [2 points] When using **single link** what is the maximal possible depth of a hierarchical clustering tree for the data in 1? What is the maximal possible depth of such a hierarchical clustering tree for the data in 2?

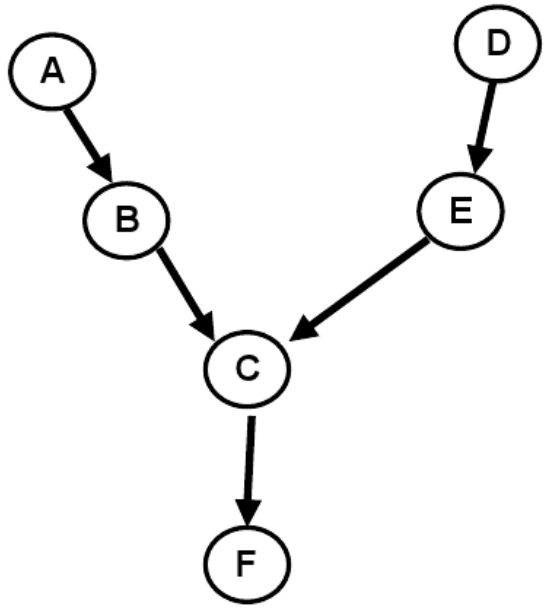
Solution:
When using single link with binary data, we can obtain cases where we are always growing the cluster by 1 node at a time leading to a tree of depth N. This is also clearly the case for continuous values.

(d) [2 points] Would your answers to (3) change if we were using **complete link** instead of **single link**? If so, would it change for both types of data? Briefly explain.

Solution:
While the answer for continuous values remain the same (its easy to design a dataset where each new sample is farther from any of the previous samples) for binary data, if k is small compared to N we will not be able to continue to add one node at a time to the initial cluster and so the depth will change to be lower than N.

Question 4. **D-separation**

Consider the following Bayesian network of 6 variables.



(a) [3 points] Set $X = \{B\}$ and $Y = \{E\}$. Specify two distinct (not-overlapping) sets Z such that: $X \perp\!\!\!\perp Y | Z$ (in other words, X is independent of Y given Z).

Solution:
$Z = \{A\}$ and $Z = \{D\}$

(b) [2 points] Can you find another distinct set for Z (i.e. a set that does not intersect with any of the sets listed in 1)?

Solution:
The empty set $Z = \{\}$

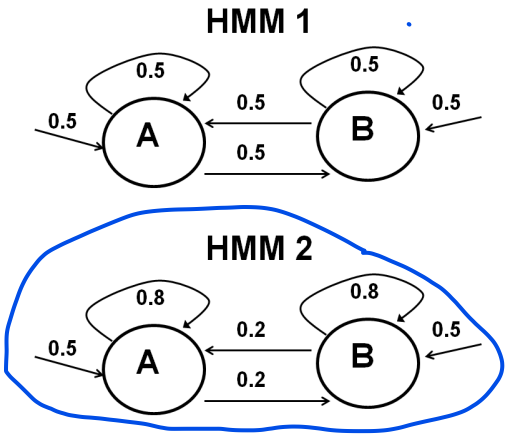
(c) [2 points] How many distinct Z sets can you find if we replace B with A while Y stays the same (in other words, now $X = \{A\}$ and $Y = \{E\}$)? What are they?

Solution:
$Z = \{\}$, $Z = \{B\}$ and $Z = \{D\}$

(d) [2 points] If $W \perp\!\!\!\perp X | Z$ and $X \perp\!\!\!\perp Y | Z$ for some distinct variables W, X, Y, Z, can you say $W \perp\!\!\!\perp Y | Z$? If so, show why. If not, find a counterexample from the graph above.

Solution:
No. $A \perp\!\!\!\perp F | B$ and $D \perp\!\!\!\perp A | B$ but D and F are not independent given B.

Question 5. **HMM**



The figure above presents two HMMs. States are represented by circles and transitions by edges. In both, emissions are deterministic and listed inside the states.

Transition probabilities and starting probabilities are listed next to the relevant edges. For example, in HMM 1 we have a probability of 0.5 to start with the state that emits A and a probability of 0.5 to transition to the state that emits B if we are now in the state that emits A.

In the questions below, O_{100}=A means that the 100th symbol emitted by the HMM is A.

(a) [3 points] What is $P(O_{100} = A, O_{101} = A, O_{102} = A)$ for HMM1?

Solution:
Note that P(O100=A, O101=A, O102=A) = P(O100=A, O101=A, O102=A,S100=A, S101=A, S102=A) since if we are not always in state A we will not be able to emit A. Given the Markov property this can be written as:

P(O100=A, O101=A, O102=A,S100=A, S101=A, S102=A) = P(O100=A|S100=A) P(S100=A) P(O101=A|S101=A) P(S101=A|S100=A) P(O102=A|S102=A) P(S102=A|S101=A)

The emission probabilities in the above equation are all 1. The transitions are all 0.5. So the only question is: What is P(S100=A)? Since the model is fully symmetric, the answer to this is 0.5 and so the total equation evaluates to: 0.5^3

(b) [3 points] What is $P(O_{100} = A, O_{101} = A, O_{102} = A)$ for HMM2?

Solution:
$0.5 * 0.8^2$

(c) [3 points] Let P_1 be: $P_1 = P(O_{100} = A, O_{101} = B, O_{102} = A, O_{103} = B)$ for HMM1 and let P_2 be: $P_2 = P(O_{100} = A, O_{101} = B, O_{102} = A, O_{103} = B)$ for HMM2. Choose the correct answer from the choices below and briefly explain.

1. $P_1 > P_2$
2. $P_2 > P_1$
3. $P_1 = P_2$
4. Impossible to tell the relationship between the two probabilities

Solution:
(a). P1 evaluates to 0.5^4 while P2 is $0.5 * 0.2^4$ so clearly $P_1 > P_2$.

(d) [3 points] Assume you are told that a casino has been using one of the two HMMs to generate streams of letters. You are also told that among the first 1000 letters emitted, 500 are As and 500 are Bs. Which is of the following answers is the most likely (briefly explain):

1. The casino has been using HMM 1
2. The casino has been using HMM 2
3. Impossible to tell

do not have to be in order

Solution:
(c). While we saw in the previous question that it is much more less likely to switch between A and B in HMM2, this is only true if we switch at every step. However, when aggregating over 1000 steps, since the two HMMs are both symmetric, both are likely to generate the *same* number of As and Bs.

Question 6. **Markov Decision Process**

Consider a robot that is moving in an environment. The goal of the robot is to move from an initial point to a destination point as fast as possible. However, the robot has the limitation that if it moves fast, its engine can overheat and stop the robot from moving. The robot can move with two different speeds: *slow* and *fast*. If it moves fast, it gets a reward of 10; if it moves slowly, it gets a reward of 4. We can model this problem as an MDP by having three states: *cool*, *warm*, and *off*. The transitions are shown in below. Assume that the discount factor is 0.9 and also assume that when we reach the state *off*, we remain there without getting any reward.

s	a	s'	$P(s'\|a,s)$
cool	slow	cool	1
cool	fast	cool	1/4
cool	fast	warm	3/4
warm	slow	cool	1/2
warm	slow	warm	1/2
warm	fast	warm	7/8
warm	fast	off	1/8

(a) [2 points] Consider the **conservative** policy when the robot always moves slowly. What is the value of $J^*(cool)$ under the conservative policy? Remember that $J^*(s)$ is the expected discounted sum of rewards when starting at state s

Solution:
$J^*(cool) = 4 + 0.9J^*(cool)$
$J^*(cool) = 40$

(b) [3 points] What is the optimal policy for each state?

Solution:
If in state *cool* then move *fast*. If in state *warm* then move *slow*.

(c) [2 points] Is it possible to change the discount factor to get a different optimal policy? If yes, give such a change so that it results to a **minimum** changes in the optimal policy and if no justify your answer in at most two sentences.

Solution:
Yes, by decreasing the discount factor. For example by choosing the discount factor equal to zero the robot always chooses an action that gives the highest immediate reward.

(d) [2 points] Is it possible to change the immediate reward function so that J^* changes but the optimal policy remains unchanged? If yes, give such a change and if no justify your answer in at most two sentences.

Also eg cool>fast>cool is $J^*(s) = R(s,a) + \gamma \sum_{s'} p(s'|s,a) \cdot J^*(s)$

$1/4 \cdot 10 = 2.5$

10

≥ 27.5

Solution:
Yes, for example by multiplying all the rewards by two.

(e) [3 points] One of the important problems in MDPs is to decide what should be the value of the discount factor. For now assume that we don't know the value of discount factor but an expert person tells us that action sequence {*fast, slow, slow*} is preferred to the action sequence { *slow, fast, fast*} if we start from either of states *cool* or *warm*. What does it tell us about the discount factor? What ranges of discount factor is consistent with this preference?

Solution:
The discounted sum of future rewards using discount factor λ is calculated by: $r + r(\lambda) + r(\lambda^2) + \ldots$.
So by solving the below equation, we would be able to find a range for discount factor λ:
$10 + 4\lambda + 4\lambda^2 > 4 + 10\lambda + 10\lambda^2$

↳ actually needs 7/8 probs inside

$10 + \frac{7}{8}(4\lambda + 4\lambda^2) > 4 + 10\lambda + \frac{7}{8}10\lambda^2$

Question 7. **SVM**

(a) Kernels

i. [4 points] In class we learnt that SVM can be used to classify linearly inseparable data by transforming it to a higher dimensional space with a kernel $K(x,z) = \phi(x)^T\phi(z)$, where $\phi(x)$ is a feature mapping. Let K_1 and K_2 be $R^n \times R^n$ kernels, K_3 be a $R^d \times R^d$ kernel and $c \in R^+$ be a positive constant. $\phi_1 : R^n \to R^d$, $\phi_2 : R^n \to R^d$, and $\phi_3 : R^d \to R^d$ are feature mappings of K_1, K_2 and K_3 respectively. Explain how to use ϕ_1 and ϕ_2 to obtain the following kernels.

a. $K(x,z) = cK_1(x,z)$

b. $K(x,z) = K_1(x,z)K_2(x,z)$

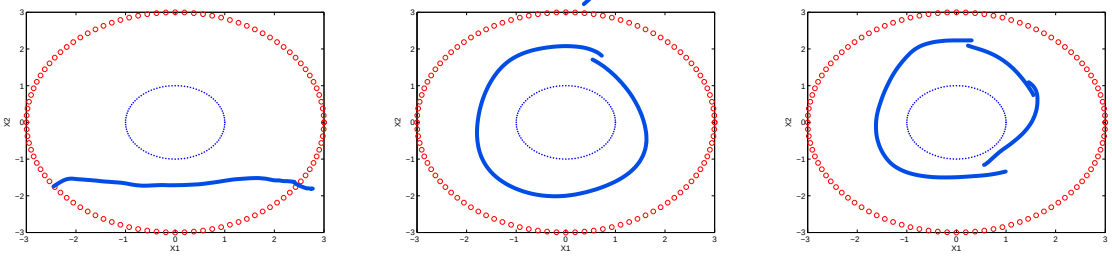
Solution:
a. $\phi(x) = \sqrt{(c)}\phi_1(x)$
b. $\phi(x) = \phi_1(x)\phi_2(x)$

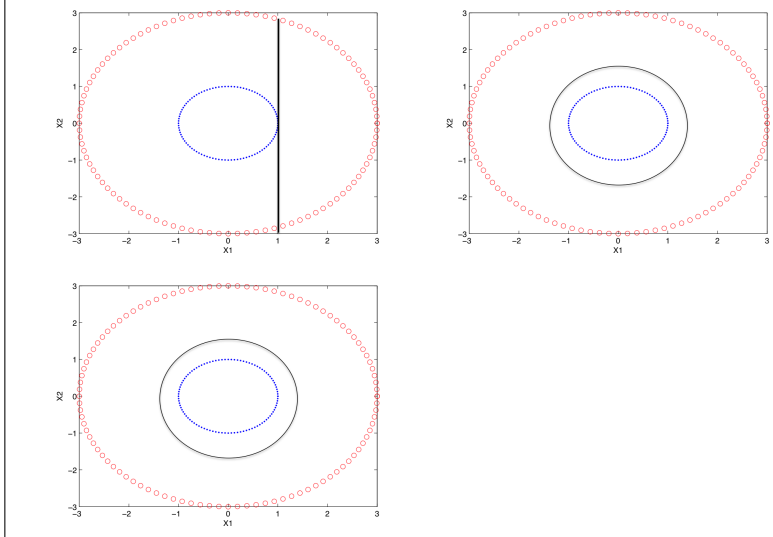
ii. [2 points] One of the most commonly used kernels in SVM is the Gaussian RBF kernel: $k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right)$. Suppose we have three points, z_1, z_2, and x. z_1 is geometrically very close to x, and z_2 is geometrically far away from x. What is the value of $k(z_1, x)$ and $k(z_2, x)$?. Choose one of the following:
a. $k(z_1, x)$ will be close to 1 and $k(z_2, x)$ will be close to 0.
b. $k(z_1, x)$ will be close to 0 and $k(z_2, x)$ will be close to 1.
c. $k(z_1, x)$ will be close to c_1, $c_1 \gg 1$ and $k(z_2, x)$ will be close to c_2, $c_2 \ll 0$, where $c_1, c_2 \in R$
d. $k(z_1, x)$ will be close to c_1, $c_1 \ll 0$ and $k(z_2, x)$ will be close to c_2, $c_2 \gg 1$, where $c_1, c_2 \in R$

Solution:
Correct answer is a, RBF kernel generates a "bump" around the center x. For points z_1 close to the center of the bump, $K(z_1, x)$ will be close to 1, for points away from the center of the bump $K(z_2, x)$ will be close to 0.

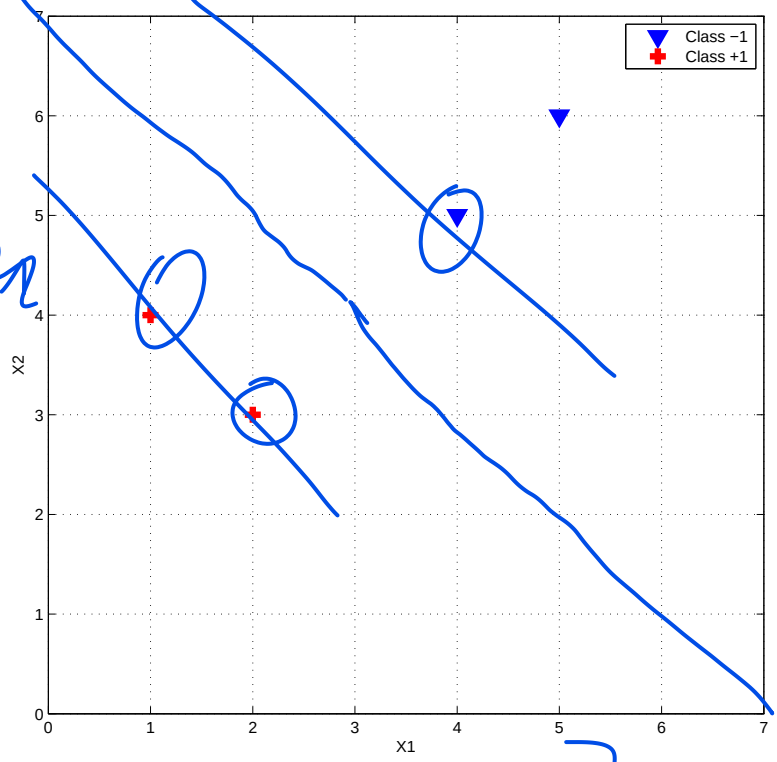
iii. [3 points] You are given the following 3 plots, which illustrates a dataset with two classes. Draw the decision boundary when you train an SVM classifier with linear, polynomial (order 2) and RBF kernels respectively. Classes have equal number of instances.

Solution:


X1
X2


X1
X2

(b) [3 points] Hard Margin SVM



Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM on a tiny dataset with 4 points shown in Figure 2. This dataset consists of two examples with class label -1 (denoted with plus), and two examples with class label +1 (denoted with triangles).

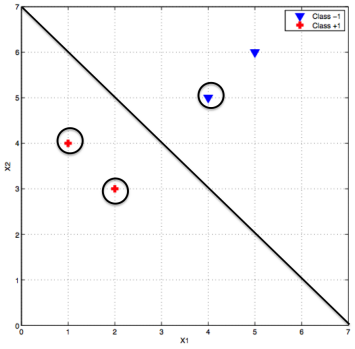
i. Find the weight vector **w** and bias b. What's the equation corresponding to the decision boundary?

Solution:
SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1. Thus the line equation is $(x_2 - 4) = -1(x_1 - 3) = x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations:
$2w_1 + 3w_2 + b = 1$ and
$4w_1 + 5w_2 + b = -1$

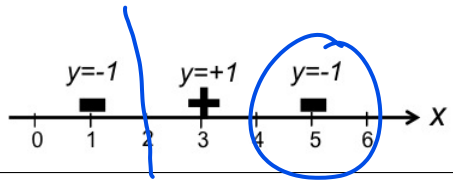
Hence $w_1 = w_2 = -1/2$ and $b = 7/2$

ii. Circle the support vectors and draw the decision boundary.

Solution:



Question 8. **Boosting**



Solution:

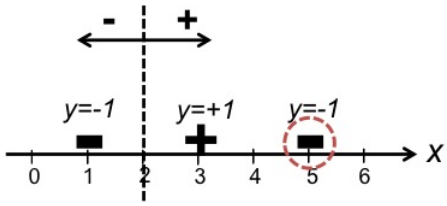


Figure 1: Sample training data for boosting algorithm.

In this problem, we study how boosting algorithm performs on a very simple classification problem shown in Figure 1. We use decision stump for each weak hypothesis h_i. Decision stump classifier chooses a constant value c and classifies all points where $x > c$ as one class and other points where $x \leq c$ as the other class.

(a) [2 points] What is the initial weight that is assigned to each data point?

Solution:
$\frac{1}{3}$

(b) [2 points] Show the decision boundary for the first decision stump (indicate the positive and negative side of the decision boundary).

Solution:
One possible solution is shown in the figure.

(c) [3 points] Circle the point whose weight increases in the boosting process.

Solution:
One possible solution is shown in the figure.

(d) [3 points] Write down the weight that is assigned to each data point after the first iteration of boosting algorithm.

Solution:
$\epsilon_t = \frac{1}{3}$
$\alpha_t = \frac{1}{2}\ln(2) = 0.3465$
For data points that are classified correctly $D_2(i) = \frac{1/3*\exp(-0.3465)}{Z_2} \approx 0.25$ and for the data point that is classified incorrectly $D_2(i) = \frac{1/3*\exp(0.3465)}{Z_2} \approx 0.5$ where Z_2 is the normalization factor.

(e) [3 points] Can boosting algorithm perfectly classify all the training examples? If no, briefly explain why. If yes, what is the minimum number of iteration?

Solution:
No, since the data is not linearly separable.

(f) [1 point] **True/False** The training error of boosting classifier (combination of all the weak classifier) monotonically decreases as the number of iterations in the boosting algorithm increases. Justify your answer in at most two sentences.

Solution:
False, boosting minimizes loss function: $\sum_{i=1}^{m} exp(-y_i f(x_i))$ which doesn't necessary mean that the training error monotonically decrease. Please look at slides 14-18 `http://www.cs.cmu.edu/~tom/10601_fall2012/slides/boosting.pdf`.

Question 9. **Model Selection**

(a) [2 points] Consider learning a classifier in a situation with 1000 features total. 50 of them are truly informative about class. Another 50 features are direct copies of the first 50 features. The final 900 features are not informative.

Assume there is enough data to reliably assess how useful features are, and the feature selection methods are using good thresholds.

- How many features will be selected by mutual information filtering?

Solution:
about 100

- How many features will be selected by a wrapper method?

Solution:
about 50

(b) Consider k-fold cross-validation. Let's consider the tradeoffs of larger or smaller k (the number of folds). For each, please select one of the multiple choice options.

i. [2 points] With a higher number of folds, the estimated error will be, on average,

- (a) Higher.
- (b) Lower.
- (c) Same.
- (d) Can't tell.

Extreme Leave one out
Train on all except 1

Solution:
Lower (because more training data)

(c) [8 points] Nearly all the algorithms we have learned about in this course have a tuning parameter for regularization that adjusts the bias/variance tradeoff, and can be used to protect against overfitting. More regularization tends to cause less overfitting.

For each of the following algorithms, we point out one such tuning parameter. If you increase the parameter, does it lead to MORE or LESS regularization? (In other words, MORE bias (and less variance), or LESS bias (and more variance)?) For every blank, please write MORE or LESS.

Naive Bayes: MAP estimation of binary features' $p(X\|Y)$, using a $Beta(\alpha, \alpha)$ prior.	Higher α means…	6.	regularization
Logistic regression, linear regression, or a neural network with a $\lambda \sum_j w_j^2$ penalty in the objective	Higher λ means…	MORE	regularization
Bayesian learning for real-valued parameter θ, given a prior $p(\theta)$, which might a wide or narrow shape. (For example, a high vs. low variance gaussian prior.)	Higher width of the prior distribution means…	1.	regularization
Neural Network: number of hidden units, n	Higher n means…	LESS	regularization
Feature selection with mutual information scoring: Include a feature in the model only if its MI(feat, class) is higher than a threshold t.	Higher t means…	7.	regularization
Decision tree: n, an upper limit on number of nodes in the tree.	Higher n means…	LESS	regularization
Boosting: number of iterations, n	Higher n means…	LESS	regularization
Dimension reduction as preprocessing: Instead of using all features, reduce the training data down to k dimensions with PCA, and use the PCA projections as the only features.	Higher k means…	LESS	regularization

Solution:

- NB α: more
- λ L2 penalty: more
- Bayesian prior width: less
- Num. hidden units: less
- MI threshold: more
- Num. dtree nodes: less
- Num. boosting iter: less
- Num. PC's: less

PRACTICE QUESTIONS

- If you train a linear regression estimator with only half the data, its bias is smaller.

- Suppose you are given a dataset of cellular images from patients with and without cancer. If you are required to train a classifier that predicts the probability that the patient has cancer, you would prefer to use Decision trees over logistic regression.

- Suppose the dataset in the previous question had 900 cancer-free images and 100 images from cancer patients. If I train a classifier which achieves 85% accuracy on this dataset, it is it a good classifier.

- A classifier that attains 100% accuracy on the training set and 70% accuracy on test set is better than a classifier that attains 70% accuracy on the training set and 75% accuracy on test set.

- Assume m is the minimum number of training examples sufficient to guarantee that with probability $1 - \delta$ a consistent learner using hypothesis space H will output a hypothesis with true error at worst ε. Then a second learner that uses hypothesis space H' will require 2m training examples (to make the same guarantee) if $|H'| = 2|H|$.

- Which of the following classifiers can perfectly classify the XOR data:

  \+ -

  \- +

  (a) Decision Tree, (b) Logistic Regression, (c) Gaussian Naive Bayes

- When the feature space is larger, overfitting is less likely.

- Non-parametric models are usually more efficient than parametric models in terms of

model storage.

- Boosting decision stumps can result in a quadratic decision boundary.

- Suppose you wish to predict age of a person from his/her brain scan using regression, but you only have 10 subjects and each subject is represented by the brain activity at 20,000 regions in the brain. You would prefer to use least squares regression instead of ridge regression.

- When doing kernel regression on a memory-constrained device, you should prefer to use a box kernel instead of a Gaussian kernel.

- To predict the chance that Steelers football team will win the Super Bowl Championship next year, you should prefer to use logistic regression instead of decision trees.

- The kmeans algorithm finds the global optimum of the kmeans cost function.

- Unlike the k-means objective, it is computationally feasible to find the optimal parameters for Gaussian mixture models exactly since the cluster assignments in GMM are soft.

PRACTICE QUESTION ANSWERS

- If you train a linear regression estimator with only half the data, its bias is smaller.

SOLUTION: FALSE. Bias depends on the model you use (in this case linear regression) and not on the number of training data.

- Suppose you are given a dataset of cellular images from patients with and without cancer. If you are required to train a classifier that predicts the probability that the patient has cancer, you would prefer to use Decision trees over logistic regression.

SOLUTION: FALSE. Decision trees only provide a label estimate, whereas logistic regression provides the probability of a label (patient has cancer) for a given input (cellular image).

- Suppose the dataset in the previous question had 900 cancer-free images and 100 images from cancer patients. If I train a classifier which achieves 85% accuracy on this dataset, it is it a good classifier.

⋆ SOLUTION: FALSE. This is not a good accuracy on this dataset, since a classifier that outputs ”cancer-free” for all input images will have better accuracy (90%).

- A classifier that attains 100% accuracy on the training set and 70% accuracy on test set is better than a classifier that attains 70% accuracy on the training set and 75% accuracy on test set.

⋆ SOLUTION: FALSE. The second classifier has better test accuracy which reflects the true accuracy, whereas the first classifier is overfitting.

- Assume m is the minimum number of training examples sufficient to guarantee that with probability $1 - \delta$ a consistent learner (i.e. a learner that correctly classifies the training data) using hypothesis space H will output a hypothesis with true error at worst ε. Then a second learner that uses hypothesis space H' will require 2m training examples (to make the same guarantee) if $|H'| = 2|H|$.

⋆ SOLUTION: FALSE. Minimum number of training examples sufficient to make an (ε,δ)-PAC guarantee depends logarithmically on hypothesis class size ($\ln|H|$) and not linearly.

- Which of the following classifiers can perfectly classify the XOR data:

+	-
-	+

(a) Decision Tree (b) Logistic Regression (c) Gaussian Naive Bayes

⋆ SOLUTION: Decision Tree only. Decision trees of depth 2 which first splits on X_1 and then on X_2 wil perfectly classify it. Logistic regression leads to linear decision boundaries, hence cannot classify this data perfectly. Due to conditional independence requirement, it is not possible to fit a Gaussian that peaks at the labels of only one class and has no covariance between features, so Gaussian Naive Bayes cannot classify this data perfectly.

- When the feature space is larger, overfitting is less likely.

⋆ SOLUTION: False. The more the number of features, the higher the complexity of the model and hence greater its ability to overfi the training data.

- Non-parametric models are usually more efficient than parametric models in terms of model storage.

⋆ SOLUTION: False. Non-parametric models either need to look at the entire dataset to predict the label of test points or require the number of parameters to scale with the dataset size, hence require more storage.

- Boosting decision stumps can result in a quadratic decision boundary.

⋆ SOLUTION: False. The sign of a finite linear combination of decision stumps always results in a piecewise linear decision boundary.

- Suppose you wish to predict age of a person from his/her brain scan using regression, but you only have 10 subjects and each subject is represented by the brain activity at 20,000 regions in the brain. You would prefer to use least squares regression instead of ridge regression.

⋆ SOLUTION: False. When the number of datapoints (subjects) is less than number of features, the least squares solution needs to be regularized to prevent overfitting, hence we prefer ridge regression.

- When doing kernel regression on a memory-constrained device, you should prefer to use a box kernel instead of a Gaussian kernel.

⋆ SOLUTION: True. A box kernel only uses a few data points for prediction and hence does not need to load the entire dataset into memory unlike Gaussian kernel which assigns non-zero weight to all training data points.

- To predict the chance that Steelers football team will win the Super Bowl Championship next year, you should prefer to use logistic regression instead of decision trees.

⋆ SOLUTION: True. Logistic regression will characterize the probability (chance) of label being win or loss, whereas decision tree will simply output the decision (win or loss).

- The kmeans algorithm finds the global optimum of the kmeans cost function.

⋆ SOLUTION: False. The kmeans cost function is non-convex and the algorithm is only guaranteed
to converge to a local optimum.

- Unlike the k-means objective, it is computationally feasible to find the optimal parameters for Gaussian mixture models exactly since the cluster assignments in GMM are “soft."

⋆ SOLUTION: False. The maximum likelihood optimization for GMM is still non-convex.