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.

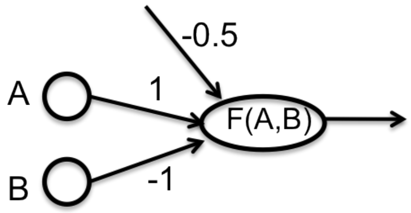
(f) [3 points] Can you represent the following boolean function with a single logistic threshold unit (i.e., a single unit from a neural network)? If yes, show the weights. If not, explain why not in 1-2 sentences.

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

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

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

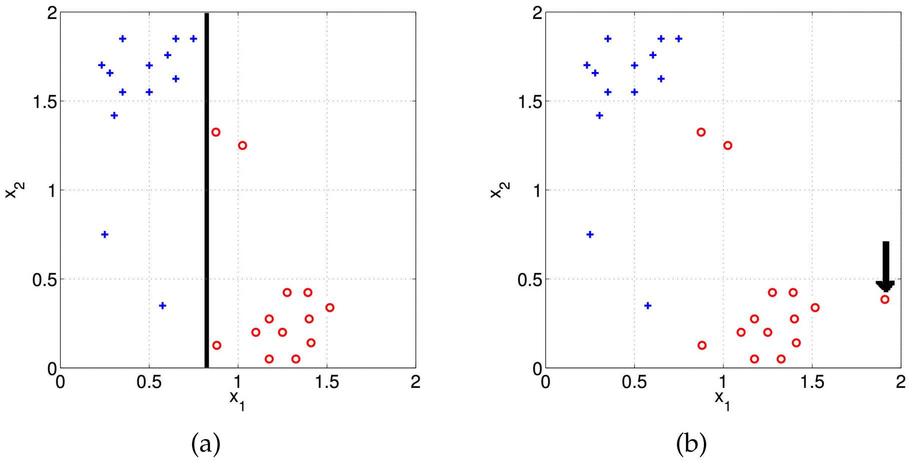


Figure 1: Labeled training set.

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

$$
\begin{aligned}
0 &= \frac{\partial}{\partial a}\left[\frac{1}{2}\sum_i (Y_i - aX_i)^2\right] && (2)\\
&= \sum_i (Y_i - aX_i)(-X_i) && (3)\\
&= \sum_i aX_i^2 - X_iY_i && (4)\\
a &= \frac{\sum_i X_iY_i}{\sum_i X_i^2} && (5)
\end{aligned}
$$

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 $\sum x_iy_i / \sum x_i^2$ 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

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

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,

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

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

Solution:

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

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

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

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

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

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

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

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.

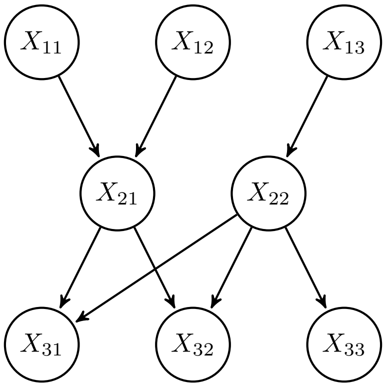
For full solution: deduct a point for a sign error. (There are many potential places for flipping signs). Deduct a point for having n/λ^2: this results from wrapping a sum around the log-prior. (Only the log-likelihood as a $\sum_i$ around it since it's the probability of drawing each data point. The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

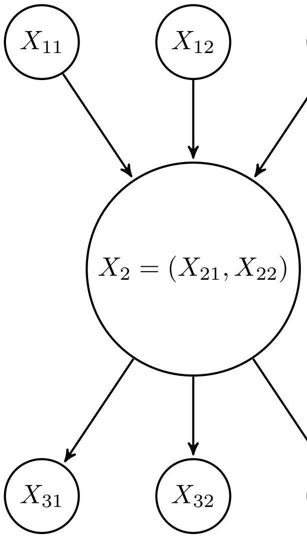
Solution:
No. Conditional independence doesn't imply marginal independence.

(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:



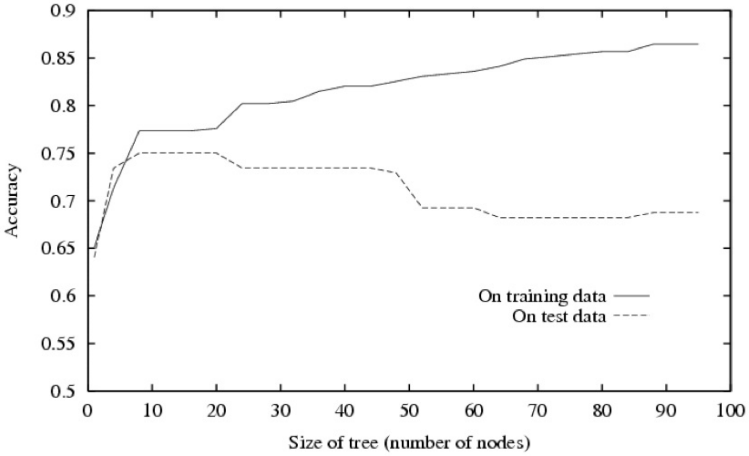
(h) [3 points] Do all the conditional independences in B hold in the new network B'? If not, write one that is true in B but not in B'. Consider only the variables present in both B and B'.

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.