Problem Set 1
10-601 Fall 2012
Due: Friday Sept. 14, at 4 pm

TA: Brendan O'Connor (brenocon@cs.cmu.edu)

Due Date

This is due at **Friday Sept. 14, at 4 pm**. Hand in a hard copy to Sharon Cavlovich, GHC 8215.
This document was last updated Saturday 29th September, 2012, 6:03pm.

Changelog: (9/6) Clarified that graphs need to be printed out and turned in. (9/10) clarified notation on 2.d.1, 2.d.2. (9/11) Added log-scale suggestion for 2.e.2; clarified wording of 2.e.5.

1 Probability Review

[All 1 point except 1d1]

Please show all steps in your solution.

1.a Equation of the Reverend

Prove

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

[Solution:
$P(A|B) = P(AB)/P(B) = P(B|A)P(A)/P(B)$
End solution]

1.b Contingencies

A is a random variable that can take one of two values $\{\diamondsuit, \heartsuit\}$. B is a random variable that can take one of two values $\{\triangle, \square\}$.

There are 117 (A_i, B_i) pairs, with the following "contingency table" of counts: each cell says how many cases there are of that pair type, e.g. 12 cases where $(A, B) = (\diamondsuit, \triangle)$.

	$A = \diamondsuit$	$A = \heartsuit$
$B = \triangle$	12	97
$B = \square$	3	5

Compute the quantities

1. $P(A = \diamondsuit)$
2. $P(A = \diamondsuit$ AND $B = \square)$ (this is a notational equivalent of $P(A = \diamondsuit, B = \square)$.)
3. $P(A = \diamondsuit$ OR $B = \square)$
4. $P(A = \diamondsuit \mid B = \square)$
5. Use the law of total probability to rewrite $P(A)$ in terms of conditional probabilities $P(A|B = \triangle)$ and $P(A|B = \square)$. Compute $P(A = \diamondsuit)$ from this equation. (If this is how you did 1b(1), then compute it with a different, more direct, approach.)

1.c Chain rule

Rewrite $P(X, Y, Z)$ as a *product* of several conditional probabilities, and one unconditioned probability involving a single variable. Your conditional probabilities can use only one random variable on the left side of the conditioning bar. For example, $P(A|C)$ and $P(A)$ would be ok, but $P(A, B|C)$ is not.

[Solution:
$P(XYZ) = P(X|YZ)P(YZ) = P(X|YZ)P(Y|Z)P(Z)$
Other variable orderings are OK too, e.g.
$P(Y|XZ)P(X|Z)P(Z)$
$P(Y|XZ)P(Z|X)P(X)$
End solution]

1.d Total probability and independence

Let X, Y, Z all be binary variables, taking values either 0 or 1.
Assume Y and Z are independent, and $P(Y = 1) = 0.9$ while $P(Z = 1) = 0.8$.
Further, $P(X = 1|Y = 1, Z = 1) = 0.6$, and $P(X = 1|Y = 1, Z = 0) = 0.1$, and $P(X = 1|Y = 0) = 0.2$.

1. *[2 points]*

   Compute $P(X = 1)$. (Hint: use the law of total probability.)

   [Solution:

   0.47.

   $$P(X = 1) = P(X = 1|Y = 0)P(Y = 0) + P(X = 1|Y = 1)P(Y = 1)$$

   The left term is all given. The right term is can be broken down via the background-conditional variant of the law of total probability (so keeping $|Y = 1$ on the right side of the conditional everywhere). We also use a superscript value notation for brevity. (This can be helpful when working things out on paper, too — eliminating the visual distraction of equal signs can clarify thinking. Sometimes I even drop the $P(...)$ notation to just $(...)$, though that is a little extreme.)

   $$\begin{aligned} & P(y^1)[P(x^1|y^1)] && (1)\\ =& P(y^1)[P(x^1|y^1z^0)P(z^0|y^1) + P(x^1|y^1z^1)P(z^1|y^1) && (2)\\ =& P(y^1)[P(x^1|y^1z^0)P(z^0) + P(x^1|y^1z^1)P(z^1) && (3)\end{aligned}$$

   The last step used the independence of Y and Z (that knowing one does not affect your belief about the other), so the final quantities are all given.

   Another approach is to use the joint form of the law of total probability, and the product version of the definition of independence.

   $$\begin{aligned} & P(x^1y^1) && (4)\\ =& P(x^1y^1z^0) + P(x^1y^1z^1) && (5)\\ =& P(x^1|y^1z^0)P(y^1z^0) + P(x^1|y^1z^1)P(y^1z^1) && (6)\\ =& P(x^1|y^1z^0)P(y^1)P(z^0) + P(x^1|y^1z^1)P(y^1)P(z^1) && (7)\end{aligned}$$

   End solution]

2. Compute the expected value $E[Y]$.

   [Solution:

   0.9

   End solution]

3. Suppose that instead of Y attaining values 0 and 1, it takes one of two values 115 and 20, where $P(Y = 115) = 0.9$. Compute the expected value $E[Y]$.

   [Solution:

   $.9 \times 115 + .1 \times 20 = 105.5$

   End solution]

2 Decision Trees

Untergang der Titanic by Willy Stöwer, 1912

Below is a dataset of the 2201 passengers and crew aboard the RMS Titanic, which disastrously sunk on April 15th, 1912. For every combination of three variables (Class, Gender, Age), we have the counts of how many people survived and did not. We've also included rollups on individual variables for convenience.

Next to the table is a *mosaic plot*, which simply visualizes the counts as proportional areas.[1]

2.a Train a decision tree

[5 points] We are interested in predicting the outcome variable Y, survival, as a function of the input features C, G, A.

Use the information gain criterion to choose which of the three features C, G or A to use at the root of the decision tree. In fact, your task here is to learn a depth 1 decision tree that uses only this root feature to classify the data (such depth-1 decision trees are often called "decision stumps"). Please show all work, including the information gain calculations for each candidate feature.

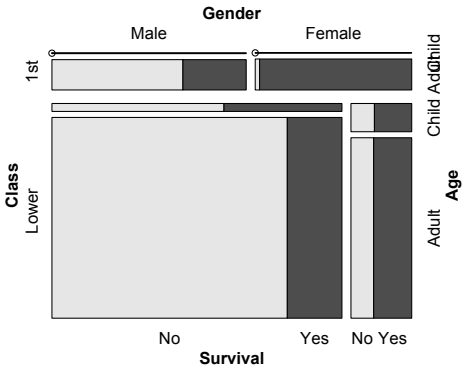
Hint: to make information gain easier to calculate, you may wish to use this formula for conditional entropy:

$$-H(Y|X) = \sum_{x,y} p(x, y) \log p(y|x)$$

[Solution:
First split: find X for highest $H(Y) - H(Y|X)$.
Sufficient to find highest $-H(Y|X) = \sum_{x,y} p(xy) \log p(y|x)$.
Here we'll use natural logarithms; base-2 or base-10 will all get the same answer.

[1] From R packages *vcd* and *reshape2*, and built-in dataset *Titanic*. The original data has four values for Class; we collapsed 2nd, 3rd, and Crew into "Lower".

Class	Gender	Age	No	Yes	Total
1st	Male	Child	0	5	5
1st	Male	Adult	118	57	175
1st	Female	Child	0	1	1
1st	Female	Adult	4	140	144
Lower	Male	Child	35	24	59
Lower	Male	Adult	1211	281	1492
Lower	Female	Child	17	27	44
Lower	Female	Adult	105	176	281
		totals:	1490	711	2201



Class	No	Yes	Total
1st	122	203	325
Lower	1368	508	1876

Gender	No	Yes	Total
Male	1364	367	1731
Female	126	344	470

Age	No	Yes	Total
Child	52	57	109
Adult	1438	654	2092

For gender

$$-H(Y|G) = p(Male, Yes)\log p(Yes|Male) + p(Male, No)\log p(No|Male) +$$

$$p(Female, Yes)\log p(Yes|Female) + p(Female, No)\log p(No|Female)$$

$$= (1/2201) * (367 * log(367/1731) + 1364 * log(1364/1731) + 344 * log(344/470) + 126 * log(126/470)$$

$$= -0.530438$$

For age

$$-H(Y|A) = p(Child, Yes)\log p(Yes|Child) + p(Child, No)\log p(No|Child) +$$

$$p(Adult, Yes)\log p(Yes|Adult) + p(Adult, No)\log p(No|Adult)$$

$$= (1/2201) * (57 * log(57/109) + 52 * log(52/109) + 654 * log(654/2092) + 1438 * log(1438/2092))$$

$$= -0.6246924$$

For Class

$$-H(Y|C) = p(1st, Yes)\log p(Yes|1st) + p(1st, No)\log p(No|1st) +$$

$$p(Lower, Yes)\log p(Yes|Lower) + p(Lower, No)\log p(No|Lower)$$

$$= (1/2201) * (203 * log(203/325) + 122 * log(122/325) + 508 * log(508/1876) + 1368 * log(1368/1876)$$

$$= -0.5955177$$

Therefore Gender is the first split: highest log-likelihood (lowest cross-entropy).
Decision stump is:
IF Gender=Female THEN predict Yes; if Gender=Male THEN predict No.
End solution]

2.b Evaluation

1. *[1 point]* What is the accuracy rate of your decision stump (depth 1 decision tree) on the training data?

   [Solution:

   Female num correct = 344. Male num correct = 1364. Acc = $(344 + 1364)/2201 = 1708/2201 = 0.776$.

   End solution]

2. *[1 point]* If you grew a complete decision tree that used all three variables, what would its accuracy be over the training data? [Hint: you don't actually need to grow the tree to figure out the answer.]

   [Solution:

   This classifier would predict the majority class for every row in the first table, so sum the larger Y count from every row. $5+118+1+140+35+1211+27+176 = 1713$ and accuracy is $1713/2201 = 0.778$.

   End solution]

2.c Decision Trees and Equivalent Boolean Expressions

[1 point] The decision tree is a function $h(C, G, A)$ that outputs a binary value. Therefore, it can be represented as a boolean logic formula.

Write a decision tree that is equivalent to the following boolean formula (i.e., a decision tree that outputs 1 when this formula is satisfied, and 0 otherwise).

$$(C \wedge \neg A \wedge \neg G) \vee (C \wedge A) \vee (\neg C \wedge G)$$

[Solution:

```
check C
    if C=1: check A
        if A=1: classify Yes
        if A=0: check G
            if G=1: classify No
            if G=0: classify Yes
    if C=0: check G
        if G=1: classify Yes
        if G=0: classify No
```

End solution]

2.d Model complexity and data size

Let's think about a situation where there is a true boolean function underlying the data, so we want the decision tree to learn it. We'll use synthetic data generated by the following algorithm. To generate an $(\vec{x}, y)$ pair, first, six binary valued $x_1, \ldots, x_6$ are randomly generated, each independently with probability 0.5. This six-tuple is our $\vec{x}$. Then, to generate the corresponding y value:

$$f(\vec{x}) = x_1 \ \vee \ (\neg x_1 \ \wedge \ x_2 \ \wedge \ x_6) \tag{8}$$

$$y = f(\vec{x}) \text{ with prob } \theta, \text{ else } (1 - f(\vec{x})) \tag{9}$$

So Y is a possibly corrupted version of $f(X)$, where the parameter θ controls the noisiness. $\theta = 1$ is noise-free. $\theta = 0.51$ is very noisy.

1. *[0.5 points]* What is $P(Y = 1 \mid (X_1 \vee (\neg X_1 \wedge X_2 \wedge X_6)) = 1)$?

   [Solution:

   θ

   End solution]

2. *[0.5 points]* What is $P(Y = 1 \mid \neg((X_1 \vee (\neg X_1 \wedge X_2 \wedge X_6))) = 1)$?

   [Solution:

   $1 - \theta$

   End solution]

3. *[1 point]* Does $P(Y = 1|X_2 = 1) = P(Y = 1)$? Why?

   [Solution:

   No. If $X_2 = 0$, the right disjunct is false, so 50% chance of $X_1 = 1$ to get $f(\vec{x}) = 1$. But if $X_2 = 1$, the right disjunct could turn out true, so there's a higher chance of getting $f(\vec{x}) = 1$.

   End solution]

4. *[1 point]* Does $P(Y = 1|X_4 = 1) = P(Y = 1)$? Why?

   [Solution:

   Yes. X_4 is irrelevant to the outcome of $f(\vec{x})$.

   End solution]

5. *[1 point]* Consider learning a decision tree classifier h. Assume the learning algorithm outputs a decision tree h that exactly matches f (despite the noise in the training data, it has so much data that it still learns f correctly). Assume the training data was generated by the above process. What should h's accuracy rate be on the training data?

   [Solution:

   θ

   (Technically, this is the expected accuracy rate, but asking that would have required introducing the notion of resampling a training set, which seemed too hard for this assignment. I was hoping the word "should" would be an acceptably precise-enough fuzzification of this concept; a few people noticed the issue.)

   End solution]

6. *[1 point]* Assume new test data is also generated from the same process. What should its accuracy rate be on this new test data (assuming plenty of test data)?

   [Solution:

   θ

   End solution]

7. *[1 point]* Decision trees can overfit, so let's think about controlling the tree's model complexity. Instead of using pruning like we learned in lecture, here we use a maximum depth parameter.

   Assuming a very large amount of training data, what's the smallest maximum-depth setting necessary to perfectly learn the generating function f?

   [Solution:

   3, because only 3 input variables are used to make the decision.

   End solution]

2.e Train/Test Experiments

Now we experimentally investigate the relationships between model complexity, training size, and classifier accuracy. Get code and test data from: `http://www.cs.cmu.edu/~tom/10601_fall2012/hw/hw1_code.tgz`

We provide a Matlab implementation of ID3, without pruning, but featuring a maxdepth parameter: *train_tree(trainX, trainY, maxdepth)*. It returns an object representing the classifier, which can be viewed with *print_tree(tree)*. Classify new data via *classify_with_tree(tree, testX)*. We also provide the simulation function to generate the synthetic data: *generate_data(N, theta)*, that you can use to create training data. Finally, there is a fixed test set for all experiments (generated using $\theta = 0.9$).

See `tt1.m` for sample code to get started.

Include printouts of your code and graphs.

1. *[1 point]* For a depth=3 decision tree learner, learn classifiers for training sets size 10 and 100 (generate using $\theta = 0.9$). At each size, report training and test accuracies.

2. *[8 points]* Let's track the learning curves for simple versus complex classifiers.

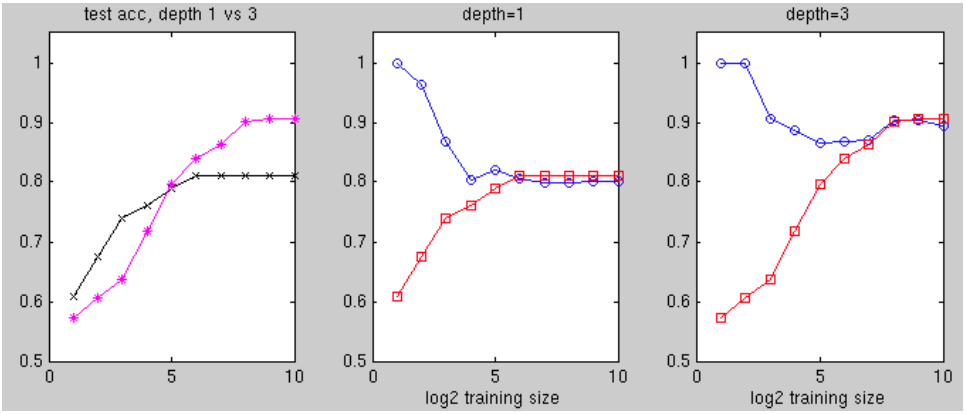
   For maxdepth=1 and maxdepth=3, perform the following experiment. For each training set size $\{2^1, 2^2, \ldots, 2^{10}\}$, generate a training set, fit a tree, and record the train and test accuracies. For each (depth,trainsize) combination, average the results over 20 different simulated training sets.

   Make three learning curve plots, where the horizontal axis is training size, and vertical axis is accuracy. First, plot the two testing accuracy curves, for each maxdepth setting, on the same graph. For the second and third graphs, have one for each maxdepth setting, and on each plot its training and testing accuracy curves. Place the graphs side-by-side, with identical axis scales. It may be helpful to use a log-scale for data size.

   [Solution:

   Pink stars: depth=3. Black x's: depth=1.

   Blue circles: training accuracy. Red squares: testing accuracy.



   End solution]

   Next, answer several questions with *no more than three sentences* each:

3. *[1 point]* When is the simpler model better? When is the more complex model better?

   [Solution:

   It is good to have high model complexity when there is lots of training data. When there is little training data, the simpler model is better.

   End solution]

4. *[1 point]* When are train and test accuracies different? If you're experimenting in the real world and find that train and test accuracies are substantially different, what should you do?

[Solution:

They're different when you're overfitting. If this is happening you have two options. (1) decrease your model complexity, or (2) get more data.

End solution]

5. *[2 points]* For a particular maxdepth, why do train and test accuracies converge to the same place? Comparing different maxdepths, why do test accuracies converge to different places? Why does it take smaller or larger amounts of data to do so?

   [Solution:

   (1) They converge when the algorithm is learning the best possible model from the model class prescribed by maxdepth: this gets same accuracy on the training and test sets. (2) The higher complexity (maxdepth=3) model class learns the underlying function better, thus gets better accuracy. But, (3) the higher complexity model class has more parameters to learn, and thus takes more data to get to this point.

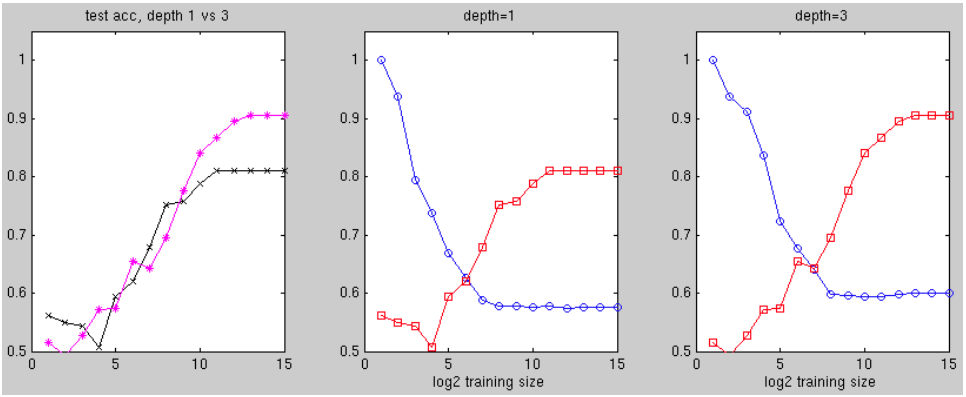
   End solution]

6. *[3 points]* For maxdepths 1 and 3, repeat the same vary-the-training-size experiment with $\theta = 0.6$ for the training data. Show the graphs. Compare to the previous ones: what is the effect of noisier data?

   [Solution:

   It's much harder for the complex model to do better. Also, it takes much longer for all test curves to converge. (Train/test curves don't converge to same place because noise levels are different.)

   Colors and styles same as previous plots. These plots test all the way up to 2^{15} training examples: you can see where they converge to, which is not completely clear with only 2^{10} examples.



   End solution]

[Note:

We asked for your graphs to be side-by-side, but only some people did this. We decided to be lenient, but did deduct half a point if you wasted an entire page for each plot. The point of making graphs is to analyze them and achieve insights. If you can't see them next to each other on the same page, it's harder to draw comparisons visually and you don't learn as much. We observed that students who put the graphs next to each other seemed to understand the phenomena better.

Another common problem was to not label the different lines in the graph, or to differentiate them with colors, but print on black-and-white. This lost points as well. If you are printing on a black-and-white printer, use line styles (e.g. dashed or dotted lines) and/or point shapes (e.g. squares, circles etc.). If you are sending something electronically, it is best to plan for a worst-case scenario that the reader may print on black-and-white.

Another, rarer, problem was to put three graphs stacked in a landscape orientation, so the training size x-axis stretched 10 inches (long way on the page), and the heights were very small (2 inches or so). This

is an extremely poor layout decision: it is very difficult to see the variation and change in accuracy. We deducted points for this layout.

Finally, a log-scale on the x-axis made it much easier to see the important trends, though this did not impact the grade.

Just as it is hard to communicate research findings without good layout and visual distinctions, it is much more time-consuming for your TA's to grade problems where the answers span across dozens of sheets of paper and it is unclear what is being shown on a plot. This increases the risk of being misgraded, and makes your TA's frustrated with you.

End note]

3 Maximum Likelihood and MAP Estimation

This question asks you to explore a simple case of maximum likelihood and MAP estimation. The material for this question will not be covered in class until Tuesday, September 11, so you might want to wait until then to attempt it. Please print out all plots and code used to create them.

Our data is a set of n Boolean (0 or 1) values drawn independently from a single Bernoulli probability distribution, for which $P(X = 1) = \theta$, and therefore $P(X = 0) = 1-\theta$. We define n Boolean-valued random variables, $X_1...X_n$ to represent the outcomes of these n distinct draws. This problem asks you to explore how to estimate the value of θ from the observed values $X_1...X_n$.

Turn in printouts of your graphs.

3.a Maximum Likelihood Estimate

1. *[1 point]* Write a formula for $P(X_1...X_n|\theta)$ in terms of θ. This is called the dataset's *likelihood*. We write $L(\theta) = P(X_1...X_n|\theta)$, to indicate that the likelihood of the data $X_1 \ldots X_n$ is a function of θ.

   [Solution:

   $$L(\theta) = \theta^{\#\{X=1\}}(1-\theta)^{\#\{X=0\}}$$

   End solution]

2. *[2 points]* Assume a dataset size $n = 9$, consisting of 6 heads and then 3 tails:

   $$(X_1, ..., X_n) = (1, 1, 1, 1, 1, 1, 0, 0, 0)$$

   Plot the likelihood curve as a function of θ, using a fine-grained grid of θ values, say for $\theta \in \{0, 0.01, 0.02, ...1\}$. For the plot, the x-axis should be θ and the y-axis $L(\theta)$. Scale your y-axis so that you can see some variation in its value. Make sure to turn in both the plot and code that made it (should only be 3 or so lines of code). [Hint: In Matlab, it's easiest to first create the vector of θ values, then compute a corresponding vector of $L(\theta)$ values.]

3. *[1 point]* In class we discussed that the maximum likelihood estimate of θ, which we call θ^{MLE} is the value that maximizes the likelihood $L(\theta)$:

   $$\theta^{MLE} = \arg\max_{\theta} L(\theta)$$

   On your plot, mark the value of θ along the x-axis that maximizes the likelihood. Does your θ^{MLE} agree with the following closed-form maximum likelihood estimator for a binomial distribution, which we learned in class?

   $$\theta^{MLE} = \frac{\sum_i X_i}{n}$$

   [Solution:

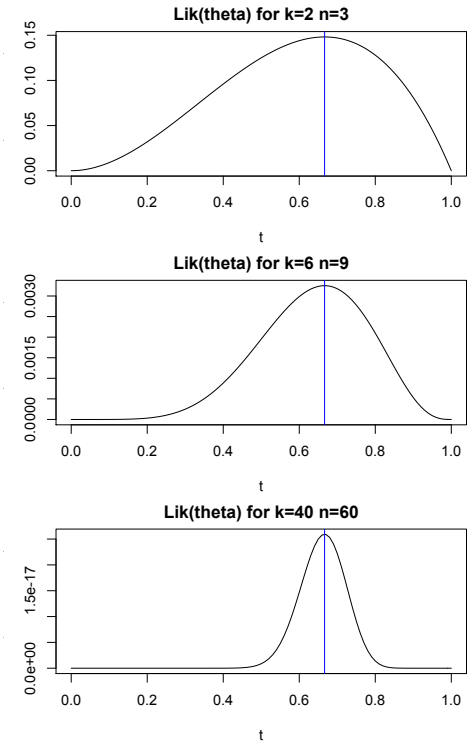
   Yes, it agrees at $\theta^{MLE} = 0.667$.

   End solution]

4. *[1 points]* Create two more likelihood plots: one for a dataset of 2 heads and 1 tail; and one for a dataset of 40 heads and 20 tails.

   [Solution:

   for 3a, 3b, 3c:



   End solution]

5. *[1 points]* Describe how the likelihood curves, maximum likelihoods, and maximum likelihood estimates compare?

   [Solution:

   1. The maximum likelihood estimates are always at the same place. 2. The likelihood curves get narrower around the MLE: with a large dataset, there is increased confidence — i.e., a narrower range of θ values are reasonably compatible with the data.

   [Optional: the maximum likelihood decreases with more data.]

   End solution]

3.b MAP Estimation

This section asks you to explore Maximum Aposteriori Probability (MAP) estimation of θ, in contrast to Maximum Likelihood estimation. Whereas the maximum likelihood estimate chooses a θ to maximize $P(X_1 \dots X_n|\theta)$, the MAP estimate instead chooses the θ that maximizes $P(\theta|X_1 \dots X_n)$. That is,

$$\theta^{MAP} = \arg\max_{\theta} P(\theta|X_1 \dots X_n)$$

which, by Bayes rule, is the same as

$$\theta^{MAP} = \arg\max_{\theta} \frac{P(X_1 \dots X_n|\theta)P(\theta)}{P(X_1 \dots X_n)}$$

and since the denominator $P(X_1 \dots X_n)$ is independent of θ this is equivalent to the simpler

$$\theta^{MAP} = \arg\max_{\theta} P(X_1 \dots X_n|\theta)P(\theta) \tag{10}$$

Thus, to find θ^{MAP} we just need to find the θ that maximizes $P(X_1 \dots X_n|\theta)P(\theta)$. This requires that we choose some probability distribution $P(\theta)$ that represents our prior assumptions about which values of θ are most probable before we have seen the data. For this, we will use the $Beta(\theta; \beta_H, \beta_T)$ distribution:

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} = Beta(\theta; \beta_H, \beta_T) \tag{11}$$

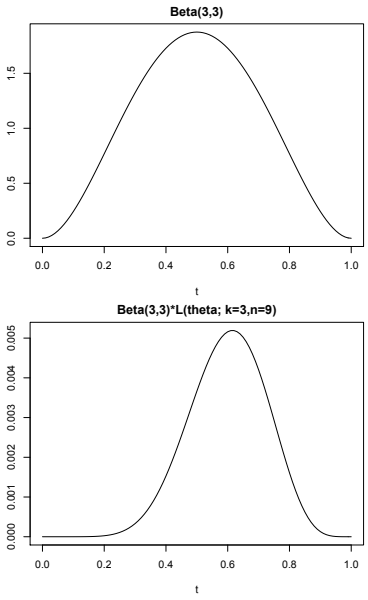
where the denominator $B(\beta_H, \beta_T)$ is a normalizing function that does not depend on θ. Therefore, we ignore this denominator when maximizing θ.

1. *[1 point]* Let's use a $Beta(\theta; 3, 3)$ distribution as our prior $P(\theta)$. Plot this as a function of θ. [Hint: The value of the normalizing denominator $B(3, 3) = 0.0333$].

2. *[1 point]* Now plot the expression in the argmax of Equation 10, versus θ. Use your earlier data set containing 6 heads and 3 tails, and use $P(\theta) = Beta(\theta; 3, 3)$ as your prior. Where is the maximum on this plot? How does your θ^{MAP} from this plot compare with your earlier θ^{MLE} estimate for the same 6 heads, 3 tails data?

   [Solution:

   For 3a1, 3b2:



   For 3b2: θ^{MAP} is a little less than θ^{MLE}: it's shrunken towards the middle (the prior belief).

   End solution]

3. *[2 points]* Above you used a $Beta(\theta; 3, 3)$ prior. Can you pick a different $Beta(\theta; \beta_H, \beta_T)$ distribution that, when used as a prior along with the earlier 2 heads and 1 tail data, wil yield a $P(\theta|D)$ that has the same shape as your likelihood plot for 6 heads and 3 tails? If so, explain *in at most two sentences* why you are sure these two curves will have identical shape. If not, explain *in at most two sentences* why this is impossible.

   [Solution:

It is possible: with a prior $Beta(5,3)$, the $(n_H, n_T) = (2,1)$ data results in a $Beta(7,4)$ posterior, whose density is $(1/Z)\theta^{7-1}(1-\theta)^{4-1}$, which is proportional to the raw likelihood function $L(\theta;6,3) = \theta^6(1-\theta)^3$.

If the answer indicated a different understanding of what "same shape" meant, and it was clear the student understood the issue, we marked it as correct. Answers that disagreed with the above but did not state their assumptions lost points.

End solution]