1 Maximum likelihood estimation of multinomials

Suppose $X \in \{1, 2\}$ and $Y \in \{1, 2, 3\}$. Define the joint distribution $P(X = j, Y = k) = \theta_{j,k}$. Consider the training data $D$ below

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Find the maximum likelihood estimates

$$\hat{\theta}_{jk} = \arg \max_n \prod_{i=1}^{n} p(x_i, y_i | \theta)$$

where there are $n = 6$ training points. Hint: just normalize the counts! (The answer should be a $2 \times 3$ table of numbers that sum to one.)

2 Presidential debate


On September 25, 1988, the evening of a presidential campaign debate in the USA, ABC News conducted two surveys of voting intentions, one before and after the debate, with these results:

<table>
<thead>
<tr>
<th>Survey</th>
<th>Bush</th>
<th>Dukakis</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre</td>
<td>294</td>
<td>307</td>
<td>38</td>
</tr>
<tr>
<td>Post</td>
<td>288</td>
<td>332</td>
<td>19</td>
</tr>
</tbody>
</table>

Let us ignore the “other” responses. Let $\pi_j$ represent the fraction of voters who prefer Bush in survey $j$ ($j = 1$ is pre debate survey, $j = 2$ is post debate survey). Assume that the two surveys are independent samples from the population of registered voters. Let $\pi_j$ have a $\text{Beta}(1, 1)$ prior before the survey. Hence we have $p(\pi_1 | S_1) = \text{Beta}(295, 308)$ and $p(\pi_2 | S_2) = \text{Beta}(289, 333)$, where $S_j$ is the $j$'th survey. What is the probability that there was a shift towards Bush as a result of the debate?

Answer: The probability there was a shift towards Bush is given by

$$p(\pi_2 > \pi_1 | S_1, S_2) = EI(\pi^2 > \pi^1)$$

where we have used the trick that the probability of a binary event, $X = \pi_2 > \pi_1$, is the expectation of its indicator, $EI(X)$, where $I(X) = 1$ if $X$ is true and $I(X) = 0$ otherwise. We can further simplify this integral thus

$$p(\pi_2 > \pi_1 | S_1, S_2) = \int_0^1 \int_0^{\pi_2} p(\pi_1 | S_1) p(\pi_2 | S_2) d\pi_1 d\pi_2$$


which shows that we are just computing the posterior probability mass above the diagonal line $\pi_1 = \pi_2$.

We can approximately solve this integral using **Monte Carlo integration**

\[
E[f(\theta)|D] = \int f(\theta)p(\theta|D) \approx \frac{1}{N} \sum_{i=1}^{N} f(\theta^i) \tag{5}
\]

where $\theta^i \sim p(\theta|D)$ is a sample from the appropriate posterior and $N$ is the number of samples (say, 1000).

In this case, we can use

\[
p(\pi_2 > \pi_1|S_1, S_2) = EI(\pi_2 > \pi_1) \approx \frac{1}{N} \sum_{i=1}^{N} I(\pi_{2i} > \pi_{1i}) \tag{6}
\]

where $\pi_{1i} \sim Beta(295,308)$ and $\pi_{2i} \sim Beta(289,333)$.

**Question**: implement Equation 7 in matlab. Turn in your code and numerical answer.

**Hint**: in the matlab statistics toolbox, you can use `betarnd` to draw samples from a beta distribution. If you don’t have the statistics toolbox, you can use `randbeta` from the lightspeed toolbox, which is freely available (google Tom Minka’s web page).

**Bonus**: plot the exact (factored) posterior $p(\pi_1, \pi_2|S_1, S_2)$ on a grid of points, superimpose the line $\pi_1 = \pi_2$ and your sampled points. The fraction of points lying above the line is your estimate. Use numerical integration to compute the exact answer.

## 3 Bayesian concept learning

In this question, you will implement the Bayesian concept learning framework for the “number game” we discussed in class. You are provided the following functions

- **hypoSpace = mkHypoSpace()** which creates the hypothesis space (a structure). The only field you should need is called 'hyps', which is a cell array. To extract the set of integers defined by the h’th hypothesis (this is called the support or extension of the hypothesis), use the following:

  \[
  \text{hypSpace.hyps}(h)
  \]

  There are **hypSpace.Hmath=23** mathematical hypotheses, and **hypSpace.Nint =5050** interval hypotheses, stored in order in order of increasing size. Thus

  \[
  \text{hypSpace.hyps}(24) = 1, \\
  \text{hypSpace.hyps}(25) = 2, \\
  \text{hypSpace.hyps(124)} = \{1,2\}
  \]

  etc.

- **prior = mkPrior(hypSpace)**, which creates a (row) vector, in which $\text{prior}(h) = p(h)$, for $h=1:5073$.

Use these to answer the following questions

1. Write a function $\text{lik} = \text{likelihood(hypSpace, X)}$ which computes

   \[
   \text{lik}(h) = p(X|h) = \begin{cases} 
   \frac{1}{\text{size(h)}} & \text{if all } x_1, \ldots, x_n \in h \\
   0 & \text{if any } x_i \notin h
   \end{cases}
   \]

   where $\text{lik}$ is a vector, which one element for each possible hypothesis. Turn in your code.
2. Write a function \( \text{post} = \text{mkPost}(\text{hypSpace}, X) \) which computes
\[
\text{post}(h) = \frac{p(X|h)p(h)}{\sum_{h'} p(X|h')p(h')}
\]
where \( \text{post} \) is a vector. Turn in your code.

3. Suppose \( X = [32] \). Compute the posterior \( \text{post}(h) = p(h|X) \). Plot the posterior over the mathematical hypotheses \( \text{post}(1:32) \) Turn in your plot.

4. What is the maximum a posterior (MAP) hypothesis \( h_{\text{MAP}} = \arg \max_h p(h|X) \)? Print out the extension of \( h_{\text{MAP}} \) (i.e. its list of integers). From the extension, you should be able to infer the name of the “rule” that defines it. e.g., if \( h_{\text{MAP}} \) is \([2, 4, \ldots, 96, 98, 100]\), then the rule is “even numbers”; if \( h_{\text{MAP}} \) is \([33, 34, 35]\), then the rule is “interval 33:35”. (You can also look at \text{mkHypSpace.m} to figure out the rule from the index \( h \).) What is the rule corresponding to \( h_{\text{MAP}} \)?

5. Sort the hypotheses into decreasing order of posterior probability. What are the top 5 most probable hypotheses? (Return their names/rules in addition to their numeric ids.) Hint: you may find the function \text{celldisp} helpful.

6. Draw a sample of 5 hypotheses from the posterior. Return their names/rules in addition to their numeric ids. Please set the random number seed as shown below, to ensure everyone’s results are the same
\[
\text{seed} = 0; \\
\text{rand('state', seed);} \\
\text{randn('state', seed);}
\]

Hint: you can use the provided function \text{data = sample\_discrete(prob, 1, n)} to sample \( n \) points from a discrete probability distribution.

7. Write a function to compute the posterior predictive distribution
\[
\text{pred}(x) = p(y(x) = 1|X) = \sum_{h \in \mathcal{H}} p(y(x) = 1|h)p(h|X)
\]
where \( \text{pred}(x) \) is a vector, with one element for each \( x = 1:100 \), \( X \) is the training data, and \( y(x) = 1 \) if \( x \) is in the concept, and \( y(x) = 0 \) otherwise. (Obviously \( y(x) = 1 \) for all \( x \in X \); the goal is to generalize beyond the training set, i.e., to predict which other numbers are in the concept class.) Plot \( \text{pred}(x) \) as a histogram. Turn in your code and plot.

8. What is \( p(y(6) = 1|X) \)? What is \( p(y(7) = 1|X) \)? What is \( p(y(8) = 1|X) \)?

9. Write a function to compute the maximum likelihood estimate
\[
\hat{h}_{\text{ML}} = \arg \max_h p(X|h)
\]

Turn in your code. What is \( \hat{h}_{\text{ML}} \)? (Give its number and name/rule.)

10. Write a function to compute the plug-in estimate
\[
\text{predML}(x) = p(y(x) = 1|\hat{h}_{\text{ML}}(X))
\]
Plot \( \text{predML}(x) \) as a histogram. Turn in your code and plot. Why is \( \text{predML} \) worse than \( \text{pred} \)?

11. Now repeat steps 3-10 using \( X = [32, 2, 44, 64, 88, 2, 10] \). Turn in your new plots and numbers. What are the main qualitative differences when using this larger, less ambiguous sample?