David Varodayan
CS 109

Section #7
February 26-28, 2020

Section #7 Solutions

Adapted for Winter 2020 by Alex Tsun

This Week in Review: Last week, we wrapped up a major portion of the class by using probabilistic models (with given parameters) to answer probability questions. Now we're moving into a new framework: can we start from data and look at it to learn the parameters themselves? We've seen a few specific cases before of approximating parameters; the Beta distribution approximates a Bernoulli p, and sample means and variances can approximate parameters of a normal. Models that attempt to capture complex data will have lots of parameters, though, and we now need a more general approach to approximating them from data. We're going to explore MLE and gradient ascent today in the context of something we're all familiar with: the Stanford Honor Code.

The Honor Code: The Stanford Student Honor Code is one of the policies that make our University so unique. Beyond the practical implications, it has the beautiful effect of creating a society where we all operate on the basis of mutual trust. In departments from English to Computer Science it has been decided that automated tools should be used to identify if two submissions are suspiciously similar. (Aside: these tools are never used as a basis for community standards cases–that always requires professional human opinion.) Automated tools are never perfect, and students and teachers alike worry of even the tiny probability of a false accusation. That becomes especially important for short assignments, like the ones at the start of CS106A. As active Stanford citizens we would like to further explore the chance of a false match in automated tools. In today's section we are going to specifically explore the Breakout assignment, one of the shortest in CS106A.

1. **Single Match**

   Say there are 1000 decision points when writing Breakout. Assume: Each decision point is binary. Each student makes all 1000 decisions. For each decision there is a probability p that a student takes the more popular choice. What is the probability distribution for the number of matching decisions (we are going to refer to this as the "score")? If possible, could you approximate this probability?

   Let A_i be the event that decision point i is matched. We note that a match occurs when both students make the more popular choice or when both students make the less popular choice. $P(A_i) = P(\text{Both more popular}) + P(\text{Both less popular}) = p^2 + (1-p)^2$.

   Let M be a random variable for the number of matches. It is easy to see that each of the 1000 decisions is an independent Bernoulli experiment with probability of success $p' = p^2 + (1-p)^2$. Therefore $M \sim Bin(1000, p')$.

   We can use a Normal distribution to approximate a binomial. We approximate $M \sim Bin(1000, p')$ with Normal random variable $Y \sim N(1000p', 1000p'(1-p'))$.

2. **Maximum Match**

When we look at two assignments, the probability of a false match is exceedingly small. What would the max similarity score look like when we compare one student to 5000 historical breakout submissions? Let X_i be the similarity score between a student who worked on their own and student i. Let Y be the highest match score between the student and all other submissions:

$$Y = \max_i X_i$$

The Central Limit Theorem tells us about the distribution of the sum of IID random variables. A more obscure theorem, the Fisher-Tippett-Gnedenko theorem, tells us about the *max* of IID random variables. It says that the max of IID exponential or normal random variables will be a "Gumbel" random variable.

$$Y \sim \text{Gumbel}(\mu, \beta) \qquad \text{The max of IID vars}$$

$$f(Y = k) = \frac{1}{\beta} e^{-(z+e^{-z})} \text{ where } z = \frac{k-\mu}{\beta} \qquad \text{The Gumbel PDF}$$

You are given data of 1300 students' max scores from three quarters (we believe they all worked independently): $y^{(1)} \ldots y^{(1300)}$. Set up (but do not solve) simultaneous equations we could solve to find the values of μ and β.

For this problem, we use Maximum Likelihood Estimator (MLE) to estimate the parameters $\theta = (\mu, \beta)$.

$$L(\theta) = \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$LL(\theta) = \log \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$= \sum_{i=1}^{n} \log f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$= \sum_{i=1}^{n} \log \frac{1}{\beta} e^{-(z_i + e^{-z_i})} \qquad \text{where } z_i = \frac{y^{(i)} - \mu}{\beta}$$

$$= \sum_{i=1}^{n} \log \frac{1}{\beta} + \sum_{i=1}^{n} -(z_i + e^{-z_i})$$

$$= -n \log(\beta) + \sum_{i=1}^{n} -(z_i + e^{-z_i})$$

Now we must choose the values of $\theta = (\mu, \beta)$ that maximize our log-likelihood function. First, we need to find the first derivative of the log-likelihood function with respect to our parameters.

$$\begin{aligned}
\frac{\partial LL(\theta)}{\partial \mu} &= \frac{\partial}{\partial \mu}\Big[- n \log(\beta) + \sum_{i=1}^{n} -(z_i + e^{-z_i})\Big] \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial \mu}\Big[- (z_i + e^{-z_i})\Big] \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial z_i}\Big[- (z_i + e^{-z_i})\Big]\frac{\partial z_i}{\partial \mu} && \text{By the Chain Rule} \\
&= \sum_{i=1}^{n} \Big[- 1 + e^{-z_i}\Big]\Big[- \frac{1}{\beta}\Big] \\
&= \frac{1}{\beta}\sum_{i=1}^{n} 1 - e^{-z_i}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial LL(\theta)}{\partial \beta} &= \frac{\partial}{\partial \beta}\Big[- n \log(\beta) + \sum_{i=1}^{n} -(z_i + e^{-z_i})\Big] \\
&= -\frac{n}{\beta} + \sum_{i=1}^{n} \frac{\partial}{\partial \beta}\Big[- (z_i + e^{-z_i})\Big] \\
&= -\frac{n}{\beta} + \sum_{i=1}^{n} \frac{\partial}{\partial z_i}\Big[- (z_i + e^{-z_i})\Big]\frac{\partial z_i}{\partial \beta} && \text{By the Chain Rule} \\
&= -\frac{n}{\beta} + \sum_{i=1}^{n} \Big[- 1 + e^{-z_i}\Big]\Big[\frac{\mu - y^{(i)}}{\beta^2}\Big] && \text{Where the last term equals } \frac{\partial z_i}{\partial \beta}
\end{aligned}$$

We want to find a simultaneous solution for both, but this is algebraically not possible. We will instead use an approximate method (gradient ascent) to solve for these, which will be taught next week.

Variance of Hemoglobin Levels: *sampling and bootstrapping*

A medical researcher treats patients with dangerously low hemoglobin levels. She has formulated two slightly different drugs and is now testing them on patients. First, she administered drug A to one group of 50 patients and drug B to a separate group of 50 patients. Then, she measured all the patients' hemoglobin levels post-treatment.

For simplicity, assume that all variation in the patient outcomes is due to their different reactions to treatment.

The researcher notes that the sample mean is similar between the two groups: both have mean hemoglobin levels around 10g/dL. However, drug B's group has a **sample variance** that is 3 $(\text{g/dL})^2$ **greater** than drug A's group. The researcher thinks that patients respond to drugs A and B differently. Specifically, she wants to make the scientific claim that drug A's patients will end up with a significantly different spread of hemoglobin levels compared to drug B's.

You are skeptical. It is possible that the two drugs have practically identical effects and that the observed different in variance was a result of chance and a small sample size, i.e. the **null hypothesis**. Calculate the probability of the null hypothesis using bootstrapping. Here is the data. Each number is the level of an independently sampled patient:

Hemoglobin Levels of Drug A's Group ($S^2 = 6.0$):
13, 12, 7, 16, 9, 11, 7, 10, 9, 8, 9, 7, 16, 7, 9, 8, 13, 10, 11, 9, 13, 13, 10, 10, 9, 7, 7, 6, 7, 8, 12, 13, 9, 6, 9, 11, 10, 8, 12, 10, 9, 10, 8, 14, 13, 13, 10, 11, 12, 9

Hemoglobin Levels of Drug B's Group ($S^2 = 9.1$):
8, 8, 16, 16, 9, 13, 14, 13, 10, 12, 10, 6, 14, 8, 13, 14, 7, 13, 7, 8, 4, 11, 7, 12, 8, 9, 12, 8, 11, 10, 12, 6, 10, 15, 11, 12, 3, 8, 11, 10, 10, 8, 12, 8, 11, 6, 7, 10, 8, 5

Discuss: How would this calculation be different if you were interested in looking at the statistical significance of the difference in sample mean? 95th percentile?

```
def bootstrap(sample1, sample2):
  \# make the universal population
  totalSample = copy.deepcopy(sample1)
  totalSample.extend(sample2)

  \# Run a bootstrap experiment
  countDiffGreaterThanObserved = 0
  print 'starting bootstrap'
  for i in range(50000):
      \# resample and recalculate the statistic
      resample1 = resample(totalSample, len(sample1))
      resample2 = resample(totalSample, len(sample2))
      resampleStat1 = calcSampleVariance(resample1)
      resampleStat2 = calcSampleVariance(resample2)
      diff = abs(resampleStat2 - resampleStat1)
      \# count how many times the statistic is more extreme
      if diff >= 3:
         countDiffGreaterThanObserved += 1
  \# compute the p-value
  p = float(countDiffGreaterThanObserved) / 50000
  print 'p-value:', p
```

For this data, the two-tailed (e.g. using absolute value) test returns a null hypothesis probability **p = 0.12**. There is a pretty decent chance that the observed difference in sample variance was from random chance – and it doesn't fall under what scientists often call "statistically significant."