Jerry Cain and Chris Piech
CS 109

Section 8 Solution
March 9, 2021

Section 8 Solution

Based on the work of many CS109 instructors and course staff members.

Warmups

1 Parameters and MLE

Suppose $x_1, \ldots, x_n$ are i.i.d. (independent and identically distributed) values sampled from some distribution with density function $f(x|\theta)$, where θ is unknown. Recall that the likelihood of the data is

$$L(\theta) = f(x_1, x_2, \ldots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

Recall we solve an optimization problem to find $\hat{\theta}$ which maximizes $L(\theta)$, i.e., $\hat{\theta} = \arg\max_\theta L(\theta)$.

1. Write an expression for the log-likelihood, $LL(\theta) = \log L(\theta)$.
2. Why can we optimize $LL(\theta)$ rather than $L(\theta)$?
3. Why do we optimize $LL(\theta)$ rather than $L(\theta)$?

Answer.

1. $LL(\theta) = \sum_{i=1}^{n} \log f(x_i|\theta)$
2. The logarithm (for bases > 1) is a monotonically increasing function. This means that if $f(a)$ $> f(b)$, then $\log(f(a)) > \log(f(b))$, so the arg max function is preserved across a logarithmic transformation, i.e., $\arg\max L(\theta) = \arg\max LL(\theta)$.
3. Logs turn products into sums, which makes taking the derivative for maximization or minimization much simpler.

2 Maximum A Posteriori

1. Intuitively, what is MAP? What problem is it trying to solve? How does it differ from MLE?
2. Given a 6-sided die (possibly unfair), you roll the die N times and observe the counts for each of the 6 outcomes as $n_1, \ldots, n_6$. What is the maximum a posteriori estimate of this distribution, using Laplace smoothing? Recall that the die rolls themselves follow a multinomial distribution.

Answer.

1. From the course notes: The paradigm of MAP is that we should choose the value for our parameters that is **the most likely given the data**. At first blush this might seem the same as MLE; however, remember that MLE chooses the value of parameters that **makes the data most likely**. One of the disadvantages of MLE is that it best explains data we have seen and makes no attempt to generalize to unseen data. In MAP, we incorporate prior belief about our parameters, and then we update our posterior belief of the parameters based on the data we have seen.

2. Using a prior which represents one imagined observation of each outcome is called Laplace smoothing and it guarantees that none of your probabilities are 0 or 1. The Laplace estimate for a Multinomial RV is $p_i = \frac{n_i+1}{N+6}$ for $i = 1, ..., 6$.

3 Naive Bayes

Recall the classification setting: we have data vectors of the form $X = (X_1, \dots, X_m)$ and we want to predict a label $Y \in \{0, 1\}$.

1. Recall in Naive Bayes, given a data point x, we compute $P(Y = 1|X = x)$ and predict $Y = 1$ provided this quantity is ≥ 0.5, and otherwise we predict $Y = 0$. Decompose $P(Y = 1|X = x)$ into smaller terms, and state where the Naive Bayes assumption is used.

2. Suppose we are given example vectors with labels provided. Give a formula to estimate (using maximum likelihood) each quantity $P(X_i = x_i|Y = y)$ above, for $i \in \{1, \dots, m\}$ and $y \in \{0, 1\}$. You can assume there is a function count which takes in any number of boolean conditions and returns a count over the data of the number of examples in which they are true. For example, count$(X_3 = 2, X_5 = 7)$ returns the number of examples where $X_3 = 2$ and $X_5 = 7$.

Answer.

1.

$$\begin{aligned} P(Y = 1|X = x) &= \frac{P(Y = 1)P(X = x|Y = 1)}{P(Y = 1)P(X = x|Y = 1) + P(Y = 0)P(X = x|Y = 0)} && \text{(Bayes+LTP)} \\ &= \frac{P(Y = 1)\prod_{i=1}^{m} P(X_i = x_i|Y = 1)}{P(Y = 1)\prod_{i=1}^{m} P(X_i = x_i|Y = 1) + P(Y = 0)\prod_{i=1}^{m} P(X_i = x_i|Y = 0)} && \text{(NB Assumption)} \end{aligned}$$

2. $P(X_i = x_i|Y = y) = \dfrac{\text{count}(X_i = x_i, Y = y)}{\text{count}(Y = y)}$

Problems

1 The Honor Code

We have decided that automated tools should be used to identify if two submissions are suspiciously similar. (N.B. these tools are never used as a basis for community standards cases — those always require professional human opinion.) However, automated tools are never perfect. As active CS109 students, we would like to explore the chance of a false positive in automated tools. For this task, we will consider Breakout, a CS106A assignment where students implement Breakout.

This problem combines our knowledge of the Central Limit Theorem and Maximum Likelihood Estimation. Exciting!

1.1 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?

Answer. 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 $Y \sim N(1000p', 1000p'(1-p'))$.

1.2 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): $y^{(1)} \dots y^{(1300)}$. Set up (but do not solve) simultaneous equations we could solve to find the values of μ and β.

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

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta) \\
LL(\theta) &= \log \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta) \\
&= \sum_{i=1}^{n} \log f(Y^{(i)} = y^{(i)} \mid \theta) \\
&= \sum_{i=1}^{n} \log \frac{1}{\beta} e^{-(z_i + e^{-z_i})} && \text{where } z_i = \frac{y^{(i)} - \mu}{\beta} \\
&= \sum_{i=1}^{n} \log \frac{1}{\beta} + \sum_{i=1}^{n} -(z_i + e^{-z_i}) \\
&= -n \log(\beta) + \sum_{i=1}^{n} -(z_i + e^{-z_i})
\end{aligned}$$

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 would instead use an approximation method called gradient ascent/descent (to be taught later this week) to solve for these.

2 Multiclass Bayes

In this problem we are going to explore how to write Naive Bayes for multiple output classes. We want to predict a single output variable Y which represents how a user feels about a book. Unlike in your homework, the output variable Y can take on one of the *four* values in the set {Like, Love, Haha, Sad}. We will base our predictions off of three binary feature variables $X_1, X_2,$ and X_3 which are indicators of the user's taste. All values $X_i \in \{0, 1\}$.

We have access to a dataset with 10,000 users. Each user in the dataset has a value for X_1, X_2, X_3 and Y. You can use a special query method **count** that returns the number of users in the dataset with the given *equality* constraints (and only equality constraints). Here are some example usages of **count**:

count($X_1 = 1, Y =$ Haha)	returns the number of users where $X_1 = 1$ and $Y =$ Haha.
count($Y =$ Love)	returns the number of users where $Y =$ Love.
count($X_1 = 0, X_3 = 0$)	returns the number of users where $X_1 = 0$, and $X_3 = 0$.

You are given a new user with $X_1 = 1$, $X_2 = 1$, $X_3 = 0$. What is the best prediction for how the user will feel about the book (Y)? You may leave your answer in terms of an argmax function. You should explain how you would calculate all probabilities used in your expression. Use **Laplace estimation** when calculating probabilities.

Answer. We can make the Naive Bayes assumption of independence and simplify argmax of $P(Y|\mathbf{X})$ to get an expression for $\hat{Y}$, the predicted output value, and evaluate it using the provided **count** function.

$$\begin{aligned}
\hat{Y} &= \arg\max_y \frac{P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y)}{P(X_1 = 1, X_2 = 1, X_3 = 0)} \\
&= \arg\max_y P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y) \\
&= \arg\max_y P(X_1 = 1|Y = y)P(X_2 = 1|Y = y)P(X_3 = 0|Y = y)P(Y = y), \text{ where:}
\end{aligned}$$

$$\begin{aligned}
P(X_1 = 1|Y = y) &= \frac{\mathbf{count}(X_1 = 1, Y = y) + 1}{\mathbf{count}(Y = y) + 2} \\
P(X_2 = 1|Y = y) &= \frac{\mathbf{count}(X_2 = 1, Y = y) + 1}{\mathbf{count}(Y = y) + 2} \\
P(X_3 = 1|Y = y) &= \frac{\mathbf{count}(X_3 = 1, Y = y) + 1}{\mathbf{count}(Y = y) + 2} \\
P(X_1 = 0|Y = y) &= \frac{\mathbf{count}(X_1 = 0, Y = y) + 1}{\mathbf{count}(Y = y) + 2} \\
P(X_2 = 0|Y = y) &= \frac{\mathbf{count}(X_2 = 0, Y = y) + 1}{\mathbf{count}(Y = y) + 2} \\
P(X_3 = 0|Y = y) &= \frac{\mathbf{count}(X_3 = 0, Y = y) + 1}{\mathbf{count}(Y = y) + 2}
\end{aligned}$$

You don’t need to use MAP to estimate $P(Y)$: $P(Y = y) = \mathbf{count}(Y = y)/10,000$