

ACM116 Problem Set 5, Proposed Problem Solutions

Sushant Sundaresh

31 October 2014

Problem 1

(a)

One way to interpret the difference between (“arrivals followed by nonarrivals” or “nonarrivals followed by arrivals”) and (“arrivals followed by arrivals” and “nonarrivals followed by nonarrivals”) is to model the system as follows:

$$\mathbf{M}_1(p, \epsilon) = \begin{cases} \mathbf{P}_{prior}[p, \epsilon] & \sim Uniform \\ \mathbf{P}_{prior}[R_k] & = p \\ \mathbf{P}_{prior}[N_k] & = 1 - p \\ \mathbf{P}[R_k|R_{k-1}] & = p + \epsilon \\ \mathbf{P}[R_k|N_{k-1}] & = p - \epsilon \\ \mathbf{P}[N_k|R_{k-1}] & = 1 - p - \epsilon \\ \mathbf{P}[N_k|N_{k-1}] & = 1 - p + \epsilon \end{cases}$$

This form looks like it might preserve a symmetric data mean for $p \sim 0.5$. Above, “uniform” means over the valid bounds, which is to say, a very lazy prior. Let’s just assert $\epsilon > 0$ and $p + \epsilon < 1$, and $p - \epsilon > 0$ since we know the edge cases (equality) are unreasonable given the data. You could even restrict $\epsilon < 0.1$ in your search as the packet arrival dependence we see in the data looks quite slight.

(b)

There is only a dependence on the previous state in our model so the measured values rr, rw, wr, ww are enough to specify:

$$\mathbf{P}(S|\mathbf{M}_1(p, \epsilon)) = \begin{cases} p(p + \epsilon)^{rr}(p - \epsilon)^{rw}(1 - p - \epsilon)^{wr}(1 - p + \epsilon)^{ww} & S_0 = R_0 \\ (1 - p)(p + \epsilon)^{rr}(p - \epsilon)^{rw}(1 - p - \epsilon)^{wr}(1 - p + \epsilon)^{ww} & S_0 = N_0 \end{cases}$$

and while we're at it,

$$\ln P(S|M_1(p, \epsilon)) = \begin{cases} \ln p + rr \ln(p + \epsilon) + rw \ln(p - \epsilon) + wr \ln(1 - p - \epsilon) + ww \ln(1 - p + \epsilon) & S_0 = R_0 \\ \ln(1 - p) + rr \ln(p + \epsilon) + rw \ln(p - \epsilon) + wr \ln(1 - p - \epsilon) + ww \ln(1 - p + \epsilon) & S_0 = N_0 \end{cases}$$

It's not necessary here, but you might try writing the above as an "independent coin-flipping probability times a correction factor," e.g.

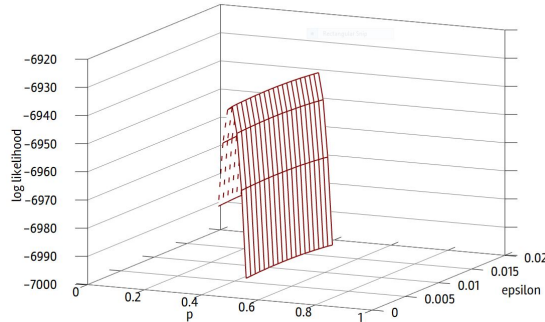
$$P(S|M_1(p, \epsilon)) = p^{rr+rw+1}(1-p)^{wr+ww} \left(1 + \frac{\epsilon}{p}\right)^{rr} \left(1 - \frac{\epsilon}{p}\right)^{rw} \left(1 - \frac{\epsilon}{1-p}\right)^{wr} \left(1 + \frac{\epsilon}{1-p}\right)^{ww}$$

$$P(S|M_1(p, \epsilon)) \sim p^{\text{arrivals}} (1-p)^{\text{non-arrivals}} \exp\left(\frac{\epsilon}{p}(rr - rw) + \frac{\epsilon}{1-p}(ww - wr)\right)$$

for one of the cases above with small ϵ . E. Jaynes discusses this problem in his *Probability: The Logic of Science*. That expression above has a lot of funny consequences - for example, in a long enough sequence of coin flips, this model suggests any finite run length of successes can become exponentially more likely than the binomial representation. Perhaps this is not surprising, but some of the other consequences are labeled as "unsolved" in Jaynes' book. Might be worth a look!

(c)

The Octave script is attached, and commented. The posterior likelihood distribution given our model and priors is shown below. This particular sample sequence had a clear peak at $p = 0.5$ but a log-likelihood difference of barely 0.5 at the peak, near $\epsilon \sim 0.01$. Your data will have a more pronounced peak, but in log-space it'll still look tiny.



(d)

The likelihood should peak around $\epsilon \sim 0.01$ and $p \sim 0.5$. For the data given I believe the ratio of the probability between the peak parameter values and the $p = 0.5, \epsilon = 0$ case was ~ 10 – *fold*. Depending on the sampled run it ranged as low as 0.5 log-units’ difference. It wasn’t a strong signal, despite the amount of data, which was surprising.

Problem 2

How do we calculate this probability? Well, let’s get the computer to count for us. Have it walk over the entire sample space of packet arrival sequences that we think are possible, one step at a time, but instead of keeping track of the sequence itself, which results in an exponentially branching tree, keep track of $N+1$ states: whether you’re currently in a run of length 0, 1, ... $N-1$, or, critically, if you were at any point ever in a run of length N (or greater). If you try to branch this set of states, pretending in your mind to simulate another arrival, another arrival, another arrival, you’ll find that the branching tree loops back on itself, it doesn’t explode out anymore. The states trade probability amongst each other.

For example, if I start out in a run of length 0, i.e. $S_0 = N_0$, then the probability that I am in a run of length 1 by the next time step is p , or $p - \epsilon$, depending on the model. If I am in a run of length 1, the probability that I return to a run of length 0 in the next step is just the probability that I see no arrival in the next step. That’s $1 - p - \epsilon$! You can continue this reasoning. The critical step is, suppose I am in a run of length $N - 1$. If I see an arrival in my next step, I’m at a run of length $\geq N$. No matter what happens past this point for this sequence, it satisfies the criteria we set out to count. So we should accumulate this probability, for any and all sequences that arrive at this state. You shouldn’t be able to leave the “greater than or equal to N ” state, only to enter it.

Let’s write our states as follows:

$$\begin{bmatrix} P(RL_0) \\ P(RL_1) \\ P(RL_1) \\ P(RL_1) \\ \dots \\ P(RL_{\geq N}) \end{bmatrix}$$

This is read as “the probability that we’re in a run of length k ” and, at the end, “the probability that we saw a run of length greater than or equal to N .” Our

initial state is:

$$\begin{bmatrix} 1-p \\ p \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

Those are just our priors; just start in $[1 \ 0 \ 0 \ 0 \ 0 \dots]$ instead and ask what the next state should be. Here's what the state update process would look like for the independent arrivals case:

$$\begin{bmatrix} P(RL_0) \\ P(RL_1) \\ P(RL_1) \\ P(RL_1) \\ \dots \\ P(RL_{\geq N}) \end{bmatrix}_{t=M} = \begin{bmatrix} 1-p & 1-p & \dots & 1-p & 0 \\ p & & & & \\ & p & & & \\ & & \dots & & \\ & & & p & 1 \end{bmatrix}_{(N+1) \times (N+1)}^{M-1} \begin{bmatrix} 1-p \\ p \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}_{t=0}$$

That's an $N + 1$ square transition matrix. Try multiplying out the matrix and interpreting the results in words. For example, from the initial state, to the second arrival step, multiplying the first row by the input state, we get:

$$P(RL_0)_{t=1} = \begin{bmatrix} 1-p & 1-p & \dots & 1-p & 0 \end{bmatrix} \begin{bmatrix} 1-p \\ p \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}_{t=0}$$

Verbally, we might say, "the probability on the second step that we ended up in a run of length 0 is the probability that we started off in a run of length 0, $1 - p$, multiplied by the probability that we saw no arrival on the second step, $1 - p$. To this we add the probability that we got an arrival in our first step, p , multiplied by the probability that the second step saw a non-arrival, which broke the run, $1 - p$."

Continue this reasoning; you'll see that multiplying by the matrix counts the probability you're interested in. You keep track of the length of your runs till you hit N , then you save that probability, and let the rest of the sequences you're mentally simulating keep evolving. Note, eventually, every sequence will end up in that final state, given enough time. But, we've limited the evolution to M time steps.

For the dependent arrivals case, we get something very similar:

$$\begin{bmatrix} P(RL_0) \\ P(RL_1) \\ P(RL_1) \\ P(RL_1) \\ \dots \\ P(RL_{\geq N}) \end{bmatrix}_{t=M} = \begin{bmatrix} 1-p+\epsilon & 1-p-\epsilon & \dots & 1-p-\epsilon & 0 \\ p-\epsilon & & & & \\ & p+\epsilon & & & \\ & & \dots & & \\ & & & p+\epsilon & 1 \end{bmatrix}_{(N+1) \times (N+1)}^{M-1} \begin{bmatrix} 1-p \\ p \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}_{t=0}$$

On a representative data set (not the one you're using), I saw the following. You'll see something similar.

- Independent, unbiased flips, chance of success p at 50.00% and ϵ at 0.0%
 - Percent chance we saw a run of length ≥ 15 is 2.971%
- Dependent, unbiased flips, chance of success p at 50.00% and ϵ at 1.0%
 - Percent chance we saw a run of length ≥ 15 is 3.752%
- Independent, biased flips, chance of success p at 50.92% and ϵ at 0.0%
 - Percent chance we saw a run of length ≥ 15 is 3.750%

As stated in 1(d), at our p, ϵ and M , the difference is not too large; if the asymmetry model is correct, you'd see such runs maybe 1% more often. You can get a similar effect by just changing p alone, in a standard binomial distribution, albeit with the mean slightly off. The difference in the predictions of the two models gets quite large for $M = 10000$, but both models predict greater than a 50% likelihood of runs greater than 15 then, so it's questionable whether our model refinement was of any use at all. Still, I hope you had some fun. The idea behind these problems was to give you a chance to think about using and tweaking the independence models you've seen so far; it's pretty neat how Bayes' rule works with actual data!