

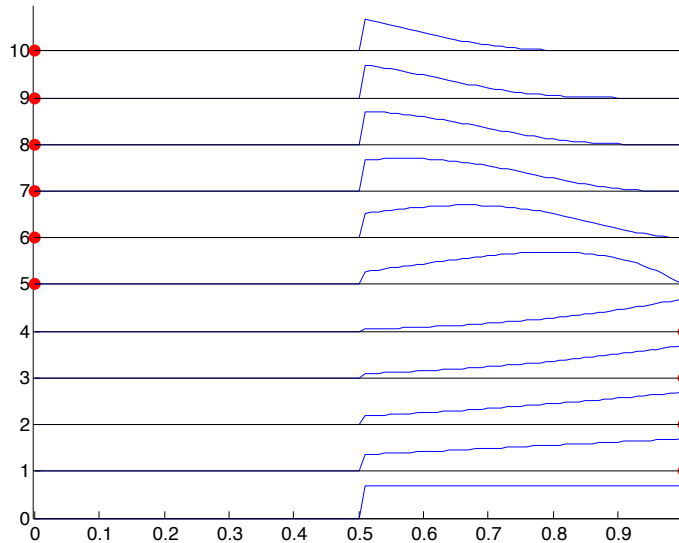
### Math Modeling, Week 3

1. Change the initial prior in the naive Bernoulli estimator. For example, set it to zero for all  $p < .5$  (and remember to normalize). What do you think will happen? Run the model and see, then explain.

We can set the prior to a uniform distribution on  $[.5, 1]$  like this:

`d(p>1/2,1) = 1/sum(p>1/2);` all other values of `d(:,1)` were previously set to zero

This command leaves all other values of  $d$  (for  $p \leq 1/2$ ) at zero. If the prior for a particular hypothesis is zero, the posterior will be zero no matter what data are observed. Therefore the posteriors we see should look like the beta distributions we saw using the uniform  $[0,1]$  prior, except that the left half will be zeroed out (and the right half will be renormalized).



2. Extend the Bernoulli learner to ternary observations. (If you like cover stories: You're training for a roshambo match and watching video of your opponent's past matches to estimate his tendencies.)

a) For binary observations, the latent parameter to be estimated is a single number,  $q \in [0,1]$ . How would you characterize the thing to be estimated for ternary observations? (Answer before moving on.)

Answer given in next question: a vector  $Q = (q_1, q_2, q_3)$ , constrained to satisfy  $\sum(q_i) = 1$ . The probability of rock is  $q_1$ , paper  $q_2$ , and scissors  $q_3$ . Alternatively, you could track only two of the probabilities, e.g.  $(q_1, q_2)$ , because the third is implicit as  $q_3 = 1 - q_1 - q_2$ . Explicitly writing all three makes the following derivations easier.

b) There are multiple good answers for part a, but let's go with this: a vector  $Q = (q_1, q_2, q_3)$ , constrained to satisfy  $\sum(q_i) = 1$ . The probability of rock is  $q_1$ , paper  $q_2$ , and scissors  $q_3$ . Assume a uniform prior for  $Q$ . What's the posterior after a single observation of rock? What's the posterior after observing  $k$  rocks,  $m$  papers, and  $n$  scissors? You can answer without the normalization (using  $\propto$ ), or if you like integrals you can work out the normalization constant.

$$p(Q|\text{rock}) \propto p(Q) \cdot p(\text{rock}|Q) \propto q_1$$

$$p(Q|\{k \text{ rocks, } m \text{ papers, and } n \text{ scissors}\}) \propto p(Q) \cdot p(k \text{ rocks, } m \text{ papers, and } n \text{ scissors}|Q) \propto q_1^k q_2^m q_3^n$$

c) Based on part b, you should be able to tell what the conjugate prior is for a ternary random variable. Try to extend this to the conjugate prior for a general  $n$ -ary random variable. Write out the distribution (with or without normalization), and write out the update rule for how the distribution's parameters change each time a new observation is made.

The conjugate prior is  $p(Q) \propto q_1^{a_1-1} q_2^{a_2-1} q_3^{a_3-1}$ . This generalizes to  $n$ -ary observables as  $p(Q) \propto \prod_{i=1}^n q_i^{a_i-1}$ , where  $a$  is a

vector of  $n$  positive numbers. (The -1 in the exponents is just part of how the Dirichlet distribution is conventionally parameterized.) Each time we make some observation  $d$ , the update rule is to increase the corresponding component of  $a$  by one:  $\Delta a_d = 1$ . Updating  $a$  in this way is equivalent to doing a Bayesian update on the full prior ( $p(Q|d) \propto p(Q)p(d|Q)$ ).

The answer to part c is called a Dirichlet distribution. Look it up in Wikipedia (a reliable resource for this sort of thing) and compare the expression there for the probability density (PDF) to your answer above.

d) Download <http://matt.colorado.edu/teaching/mathmodeling/plotDirichlet.m> for making nice pictures of Dirichlet distributions in the ternary case. (I drew it as an equilateral triangle, with each vertex corresponding to one of the values of the ternary observable variable.) Try different values and make some observations. Try non-integer inputs, including values between 0 and 1. Syntax is `plotDirichlet(a1,a2,a3)`.

e) Write some code for a Bayesian learner observing a series of ternary data, using a Dirichlet conjugate prior. The structure should be something like this:

Set prior, i.e. parameters  $a_1, a_2, a_3$  for Dirichlet

Define true generating probabilities for the observable, i.e.  $q_1, q_2, q_3$  with  $\sum q = 1$

Loop through trials

    Sample an observation according to the probabilities in  $q$  [`o(i)=find(rand<cumsum(q),1)`]

    Update Dirichlet parameters for posterior

Plot posterior

```
%% Multinomial estimation with Dirichlet conjugate prior

n = 100; %number of trials to simulate
q = [.5 .3 .2]; %true parameter value
a = zeros(n+1,3); %parameters of prior, across trials
a(1,:) = [1 1 1]; %parameters of initial prior

%run bayesian model
o = zeros(n,1);
for i=1:n
    o(i) = find(rand<cumsum(q),1); %generate observation
    a(i+1,:) = a(i,:); %copy prior from previous trial
    a(i+1,o(i)) = a(i+1,o(i))+1; %increment Dirichlet parameter corresponding to value of
current observation
end

%plot evolution of posterior
for f=1:6
    i=round(1+n*f/6); %trial to plot
    plotDirichlet(a(i,1),a(i,2),a(i,3),2,3,f)
end
```