Math Modeling, Week 3

Urns

Game 1
- A: all red
- B: 50% red, 50% blue
- Draw red. Pr[A]?

Game 2
- A: 70% red, 30% blue
- B: 30% red, 70% blue
- Draw {3 red, 1 blue}. Pr[A]?
- Draw {10 red, 7 blue}. Pr[A]?

Bayes' rule

Hypotheses, data, subjective probability

Hypothesis equivalent to probability distribution over data

Posterior $\propto$ Prior * Likelihood

$\Pr[H|D] = \Pr[H] * \Pr[D|H] / \Pr[D]$

Simple derivation from joint probability, Pr[H,D]
- $\Pr[H|D] \propto \Pr[H,D]$

Bernoulli sampling

$x \in \{0,1\}$, $\Pr[x = 1] = q$

Estimate $q \in [0,1]$

Prior density $p(q)$

Likelihood
- $\Pr[x = 1|q] = q$ (graph)
- $\Pr[x = 0|q] = 1 - q$ (graph)

Posterior density
- $p(q|x) = p(q) \cdot \Pr[x|q] / \int p(q')\Pr[x|q']dq'$

Many observations
- $\mathbf{x} = (x_1,\ldots x_{m+n})$, $\Sigma(x_i=0) = m$, $\Sigma(x_i=1) = n$
- $\Pr(\mathbf{x}|q) = (1-q)^m q^n$

Assume uniform prior, $p(q) = 1$
- $p(q|\mathbf{x}) = (1-q)^m q^n / \int (1-r)^m r^n dr = (1-q)^m q^n / B(n,m)$
- Beta function: $B(\alpha,\beta) = \Gamma(\alpha+\beta)/\Gamma(\alpha)/\Gamma(\beta) = (\alpha+\beta-1)!/(\alpha-1)!/(\beta-1)!$
- Beta distribution: $q|\mathbf{x} \sim \text{Beta}(n+1,m+1)$

Assume arbitrary beta prior, $q \sim \text{Beta}(\alpha,\beta)$
- $p(q|\mathbf{x}) \propto (1-q)^\beta q^\alpha / B(\alpha,\beta) \cdot (1-q)^m q^n \propto B(\alpha+n,\beta+m)$
- simple updating rule: just count 0s and 1s
- α,β "virtual counts"

Conjugate prior

Given a parameterized family of likelihoods
- Bernoulli, q
- Gaussian, μ

a conjugate family of distributions
- Beta, m, n
- Gaussian, m, s^2

If prior is in conjugate family, posterior is too
- Conjugate family closed under multiplication by likelihood
- Likelihood viewed as function of parameter being learned (q, μ)
- Efficiency of Bayesian updating

Kalman Filter

Tracking a stochastic process with noisy observation

Generative model

Dynamics: $x_n = x_{n-1} + \eta_n$

$\eta_n \sim \mathcal{N}(0,\sigma_\eta^2)$

Gaussian random walk

Observation: $y_n = x_n + \varepsilon_n$

$\varepsilon_n \sim \mathcal{N}(0,\sigma_\varepsilon^2)$

Gaussian noise

Independence: $\perp\{\eta_n,\varepsilon_n|n\in\mathbb{N}\}$

Causal graphical model

Conjugate prior

Gaussian likelihood, parameterized by the mean: $y_n \sim \mathcal{N}(x_n, \sigma_\varepsilon^2)$

Gaussian prior, parameterized by mean and variance

$x_n \sim \mathcal{N}(a,b)$

Posterior

$$p(x_n|y_n) \propto e^{-\frac{1}{2b}(x_n-a)^2} \cdot e^{-\frac{1}{2\sigma_\varepsilon^2}(x_n-y_n)^2}$$

$$\propto e^{-\left[\left(\frac{1}{2b}+\frac{1}{2\sigma_\varepsilon^2}\right)x_n^2-2\left(\frac{1}{2b}a+\frac{1}{2\sigma_\varepsilon^2}y_n\right)x_n\right]}$$

$$\propto e^{-\left(\frac{1}{2b}+\frac{1}{2\sigma_\varepsilon^2}\right)\left(x_n-\frac{\frac{1}{b}a+\frac{1}{\sigma_\varepsilon^2}y_n}{\frac{1}{b}+\frac{1}{\sigma_\varepsilon^2}}\right)^2}$$

Precision-weighted averaging

$$x_n|y_n \sim \mathcal{N}\left(\frac{\frac{1}{b}a+\frac{1}{\sigma_\varepsilon^2}y_n}{\frac{1}{b}+\frac{1}{\sigma_\varepsilon^2}}, \frac{1}{\frac{1}{b}+\frac{1}{\sigma_\varepsilon^2}}\right)$$

Iterative prior

$x_n|y_n \sim \mathcal{N}(c,d)$

$x_{n+1}|y_n = x_n + \eta_n \sim \mathcal{N}(c, d+\sigma_\eta^2)$

Convolution: $p(\alpha+\beta=X) = \int p(\alpha=Z)p(\beta=X-Z)\mathrm{d}Z$

Update rules

$x_n|\mathbf{y}_{n-1} \sim \mathcal{N}(\mu_n, s_n^2)$

$$\mu_{n+1} = \frac{\frac{1}{s_n^2}\mu_n+\frac{1}{\sigma_\varepsilon^2}y_n}{\frac{1}{s_n^2}+\frac{1}{\sigma_\varepsilon^2}} = \frac{\sigma_\varepsilon^2\mu_n+s_n^2y_n}{\sigma_\varepsilon^2+s_n^2}$$

$$s_{n+1}^2 = \frac{1}{\frac{1}{s_n^2}+\frac{1}{\sigma_\varepsilon^2}} + \sigma_\eta^2 = \frac{\sigma_\varepsilon^2 s_n^2}{\sigma_\varepsilon^2+s_n^2} + \sigma_\eta^2$$

Exercises

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.

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.)

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 $\Sigma(q_i) = 1$. 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.

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 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(a_1,a_2,a_3).

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 $\Sigma q = 1$
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