Data analysis: parameters

Mike Nolta

CITA–ICAT

The problem

- Given the data and a theoretical model, what can we infer about the parameters of the model?

Notation

- P(x|y) : conditional probability of x given y
- P(x,y) : joint probability of x and y
- related by : P(x,y) = P(x|y)P(y)
- Unless otherwise indicated, lowercase letters are vectors and uppercase are matrices.

The problem, restated

- Given the data (d) and a theoretical model *H* with parameters θ, what is $P(\theta|d,H)$?

Bayes theorem

- Since P(x,y) = P(x|y)P(y) = P(y|x)P(x),

$$P(\theta|d,\mathcal{H}) = \frac{P(d|\theta,\mathcal{H})P(\theta|\mathcal{H})}{P(d|\mathcal{H})}$$

- In words,

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

What do you mean, "probability"?

- The true theory parameters are fixed, not random variables.
- We're quantifying our *subjective belief* in the parameters.
- What we know now (posterior) is what we knew before (prior) and what the data tells us (likelihood).

Priors

- Typically assume uniform priors, $P(\theta) = 1$
- Priors only tend to be an issue for parameters which are poorly constrained by the data.
  - For example: CosmoMC assumes $40 < H_0 < 100$, which can alter results when you don't assume universe is flat.

Priors (2)

- But note that H(x) and H(log(x)) produce the same model, but P(x) = 1 implies P(log(x)) = x.
- Just be aware that you're usually assuming uniform priors.

Calculating the likelihood

- Easy, in theory:

$$-2\ln P(d|\theta) = \ln[\det(C)] + d^T C^{-1} d + constant$$

- But slow/impossible for large numbers of pixels.
- Need approximations...

WMAP TT likelihood

- Approx (1) assume C_l are Gaussian:

$$-2\ln P_N^* = \sum_{ll'}(\hat{C}_l - C_l)Q_{ll'}(\hat{C}_{l'} - C_{l'})$$

- Approx (2) assume C_l are log-normal:

$$-2\ln P_{LN}^* = \sum_{ll'}(\hat{z}_l - z_l)\mathcal{Q}_{ll'}(\hat{z}_{l'} - z_{l'})$$

- WMAP approx is a blend:

$$\ln P_{WMAP}^* = \frac{1}{3}\ln P_N^* + \frac{2}{3}\ln P_{LN}^*$$

Other likelihood approx.

- $(C_l)^{1/3}$ (Smith, Challinor, & Rocha 2006)
- $(x-\ln(x)-1)^{1/2}$ (Hamimache & Lewis 2008)

Ok, we have $P(d|\theta)$.

Now what?

Want to know...

- Best fit parameters
- Various expectation values:

$$\langle f \rangle = \int d\theta \, P(\theta|d) f(\theta)$$

- For example, $f(\theta) = (n_s - \langle n_s \rangle)^2$

Ok, I'll just..., um…, uh oh

- Let's integrate $P(\theta|d)$ numerically using a grid, with 20 points along each axis.
- Standard LCDM model has 6 parameters.
- Need to evaluate $P(\theta|d)$ $20^6 = 64$ million times.
- If each evaluation takes one second, that's 740 days! Just barely feasible on CITA's cluster.
- With 7 parameters, it'll take 40 years.

Monte Carlo methods

- Generate N random samples $\theta^{(i)}$ from distribution $Q(\theta)$:

$$\int d\theta P(\theta|d) f(\theta) \approx \sum w_i f(\theta^{(i)})$$

- We'll discuss various ways to generate samples: uniform, importance, rejection, Metropolis-Hastings, and Gibbs.

Uniform sampling

- Sample uniformly in volume Ω:

$$w_i = \frac{V_\Omega}{N} P(\theta^{(i)}|d), \quad Q(\theta) = \begin{cases} 1 & \theta \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

- Problem: P(θ|d) is usually concentrated in a small volume around the max-like point. Very inefficient, requiring huge numbers of samples, unless P is approx uniform over Ω.

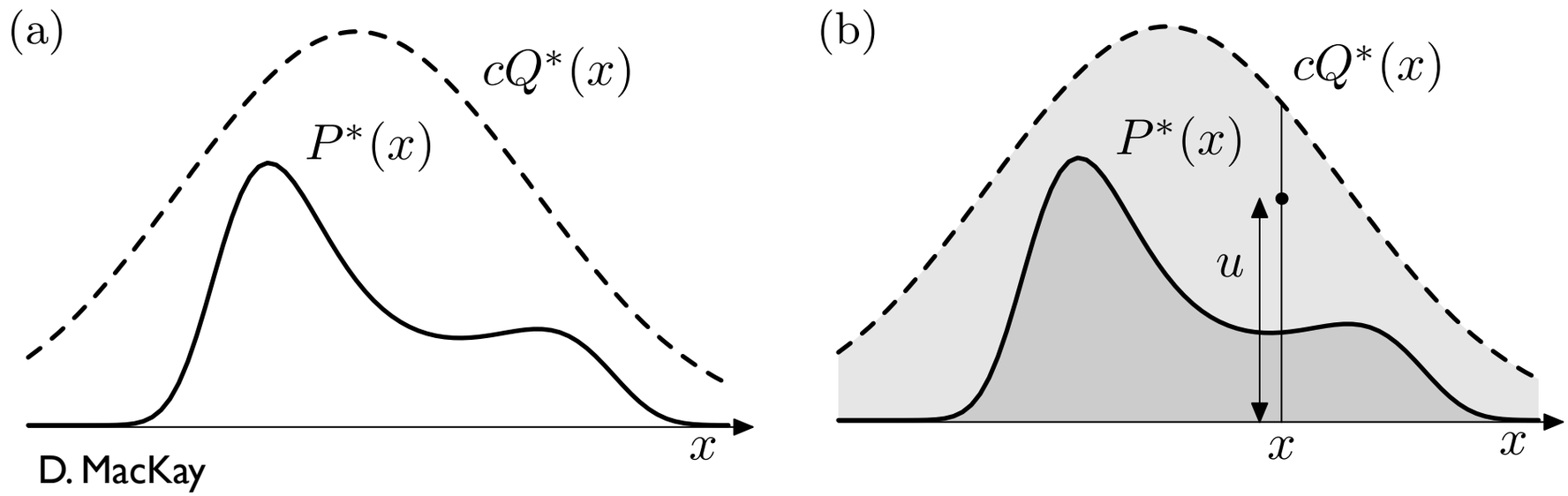
Normalization

- In practice, only know $P(\theta|d)$ up to a constant, because evidence $P(d)$ is expensive to calculate.
- Define $P = P^*/Z$, where Z is the evidence.
- Since $$\int d\theta\, P(\theta|d) = 1 \approx \frac{V_\Omega}{N} \sum P(\theta^{(i)}|d)$$
- In practice, $$w_i = \frac{P^*(\theta^{(i)}|d)}{\sum P^*(\theta^{(i)}|d)}$$

Importance sampling

- Like uniform sampling, but with a Q designed to roughly match $P(\theta|d)$ (typically Gaussian).
- Problem: if $P(\theta|d)f(\theta)$ is large where Q is small, may never converge to correct answer.

Rejection sampling



D. MacKay

- Choose sample from Q, where Q(θ|d) > P(θ|d) for all θ.
- Accept/reject sample w/ probability P(θ|d)/ Q(θ|d).

Disadvantages

- Importance and rejection sampling are prohibitively slow unless Q matches P* pretty well.
- On the other hand, if you pick the wrong Q, e.g., if P* falls off more slowly than Q, you'll get the wrong answer.
- So you have to be conservative choosing Q, and that slows things down.

Fixed Q is a problem

- Getting Q right implies knowing P* very well, yet we're running a Monte Carlo precisely because we don't know P* very well!
- Can we "explore" P*? What if Q was allowed to vary?

Markov chain

- A Markov chain is a sequence of random variables $x_0, \ldots, x_n, x_{n+1}$ such that:

$$P(x_{n+1}|x_n, \ldots, x_0) = P(x_{n+1}|x_n)$$

- Given the present state, future states are independent of the past.
- A random walk is an example of a Markov chain.

Andrei Markov

- Invented Markov chains in 1906 as a purely theoretical generalization of independent trials: $P(x_{n+1}|x_n, \ldots, x_0) = P(x_{n+1})$
- Couldn't (or wouldn't) think of a practical example.

World's first Markov Chain

- Markov (1913) analyzed a sample of 20k letters from Pushkin's *Eugene Onegin* as a Markov chain, finding:

$$\begin{array}{c} \\ \text{vowel} \\ \text{consonant} \end{array} \begin{array}{c} \begin{array}{cc} \text{vowel} & \text{consonant} \end{array} \\ \begin{pmatrix} .128 & .872 \\ .663 & .337 \end{pmatrix} \end{array}.$$

- Possibly inspired by encryption schemes, such as the "Nihilist transposition cipher" used by Russian revolutionaries.

Markov chain example: Google

- Consider the "random surfer", who starts on a random page, randomly clicks links, never goes back, and who every now and then starts on a completely new random page.
- The probability that the random surfer visits a page is its PageRank:

$$PR(A) = (1-d) + d \sum_{T \to A} \frac{PR(T_i)}{C(T_i)}$$

Monte Carlo Markov Chain (MCMC)

- MCMC is a technique for drawing samples $\theta^{(i)}$ from $P(\theta|d)$, and thus:

$$\int d\theta P(\theta|d) f(\theta) \approx \frac{1}{N} \sum f(\theta^{(i)})$$

Metropolis sampling

- given a point $\theta^{(i)}$:
- (1) choose new point θ^* from proposal density $Q(\theta^*;\theta^{(i)})$
- (2) calculate the ratio $\alpha=P^*(\theta^*|d)/P^*(\theta^{(i)}|d)$
- (3) set $\theta^{(i+1)}=\theta^*$ with probability α, otherwise set it to $\theta^{(i)}$

Physicist 'Proof' of MCMC

- Consider a thermodynamic system with states x and transition rates W(x→x'):

$$\frac{dP(x,t)}{dt} = -\sum_{x'} W(x \to x')P(x,t) + \sum_{x'} W(x' \to x)P(x',t)$$

- In thermodynamic equilibrium dP/dt=0 and $P(x,t) = P_{eq}(x)$.
- Then by the principle of detailed balance,

$$W(x \to x')P_{eq}(x) = W(x' \to x)P_{eq}(x')$$

Physicist 'Proof' of MCMC (2)

- We know the equilibrium distribution (β=1/kT): $P_{eq}(x) = e^{-\beta E(x)}/Z$
- Since $W \leq 1$, and assuming E(x')>E(x),

$$\frac{W(x \to x')}{W(x' \to x)} = \frac{e^{-\beta[E(x')-E(x)]}}{1}$$

- Metropolis sampling is like simulating a gas particle in a potential.

MCMC Advantages

- Since we can't screw up the chain by choosing a bad Q, we can be very aggressive in trying to match Q to P*.
- Robust & fast.

Gibbs sampling

- Gibbs sampling is a method for drawing samples from joint distributions, e.g., P(x,y|d), if you know the conditional probabilities P(x|y,d) & P(y|x,d).
- To sample, first take $x^{(i+1)} \sim P(x|y^{(i)},d)$, and then $y^{(i+1)} \sim P(y|x^{(i+1)},d)$.

Convergence

- A MCMC chain is "converged" if its drawing fair samples from its stationary distribution, and the chain has explored the posterior well.
- Unfortunately there is no unambiguous yes/no test for convergence.

Gelman-Rubin test

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2 = \frac{1}{m(n-1)}\sum_{t}(x_{it} - \langle x_{i.}\rangle)^2$$

$$B/n = \sum_{i=1}^{m}(\langle x_{i.}\rangle - \langle x_{..}\rangle)^2/(m-1)$$

- Run multiple chains, and check the intrachain & interchain variance.

$$\hat{V}^2 = \frac{n-1}{n}W + \frac{m+1}{mn}B \qquad \hat{R} = \frac{\hat{V}}{W}\left(\frac{df}{df-2}\right)$$

- R is a prediction for how much better the chain might get.

In practice

Parameter estimation is a solved problem

- Step 1: download CosmoMC (http://cosmologist.info/cosmomc/)
- Step 2: plug in your likelihood code
- Step 3: run CosmoMC
- Step 4: profit!

Choosing the proposal density

- Run a sample chain, compute the covariance matrix, and then use that for the real chain.
- Rule of thumb is that acceptance rate should be ~20%.

Some terminology

- "Burn-in": The portion of the beginning of the chain which are not fair samples from the posterior distribution.
- "Thinning": throwing away points in the chain, so that the remaining points are all independent samples. Typical example: keeping only every 20 samples.

WMAP parameters are almost completely automated

- `$ ./runchain.py lcdm+sz+lens/wmap5+bao`
- http://lambda.gsfc.nasa.gov/cgi-bin/chainplot/index.py
- (demo)

References

- A nice textbook is *Information Theory, Inference, and Learning Algorithms* by David J. C. McKay (Cambridge University Press)