Statistical Inference in Complex Systems: Approximate Bayesian Computation

Jon Wilkins
Santa Fe Institute
wilkins@santafe.edu

Beijing CSSS 2007

Modeling and Inference

- Modeling goals
  - Formalizing hypotheses and assumptions
  - Generating intuitive understandings of systems
- Inference goals
  - Hypothesis testing
  - Parameter estimation
  - Model selection

Normal Statistics

- Many observed distributions are (approx.) Gaussian
- Strictly speaking, Central Limit Theorem requires
  - summation of independent, identically distributed (IID) random variables
  - In practice, systems where these assumptions hold approximately will be nearly Gaussian
- Many basic statistical tools are based on this
  - Many of these tools are used even when they should not be

Inference in Complex Systems

- Analytic solutions are intractable / uninterpretable
  - In these cases, the only tool we have is simulation

- Simulation is good for getting qualitative insights, but what if we want to understand the results quantitatively?

  - How to the macroscopic behaviors of the system vary with changes in microscopic parameters (agent rules, starting conditions)
  - How can we use this type of model in a data-driven way? Given an observed macroscopic behavior, can we infer the microscopic parameters that give rise to that behavior under the model?

- Also for complicated (but not “complex” systems)

The Inference Problem

- The problem: You have
  - Some data -- statistics calculated from some observed (macroscopic) phenomenon
  - A model -- a (microscopic) process or class of processes that can be characterized by some number of parameters

- The goal: You want
  - To estimate the values of these parameters
  - To test some hypothesis (often corresponding to asking if we can reject the null hypothesis that the value of some parameter is zero (or one, or infinity, etc.)

Inference in simple systems

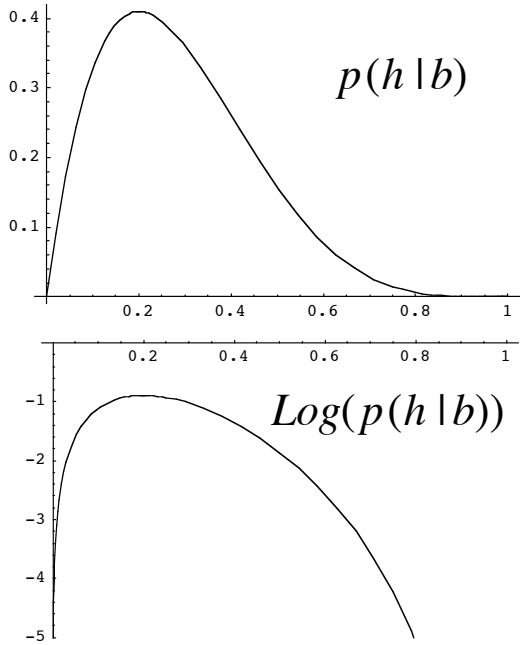
- Maximum Likelihood
  - $p (s | \theta)$ -- probability of data (s) given parameter (θ)
  - maximum likelihood estimator (MLE) is the value of θ that maximizes the value of $p (s | \theta)$
- Bayesian Inference
  - $p (\theta | s) \sim p (\theta) * p (s | \theta)$
  - a lot like maximum likelihood if your prior is uniform
  - in principle, this allows you to incorporate other types of information
  - Bayes estimator: minimize a cost function
    - minimizing average of $(\theta'-\theta)^2$: $\theta' = E [p (\theta | s)]$

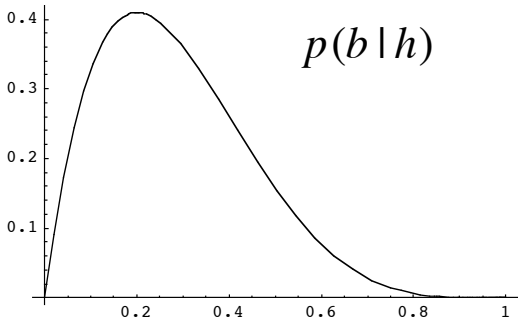
Inference: Likelihood / Uniform Prior

- A biased coin
  - p (heads) = b
  - p (tails) = 1-b
  - 1 parameter: b
  - Data: h = number of heads in 5 tosses
  - Likelihood function:

$$p(h \mid b) = \binom{5}{h} b^h (1-b)^{5-h}$$

$h_{observed} = 1$





Likelihood
95% confidence interval:
0.043 < b < 0.641

b' = 0.2 (mode of p (h|b))

Bayesian
95% credible interval:
0.043 < b < 0.641

b' = 0.2 (mean of p (b|h))

Inference: Non-Uniform Prior

- A biased coin
  - Prior:

$$p(b) = 2 - 4\left|b - 0.5\right|$$

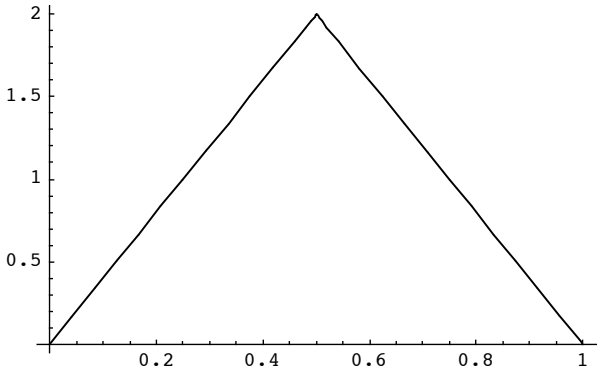
  - Likelihood function:

$$p(h \mid b) = \binom{5}{h} b^h (1-b)^{5-h}$$

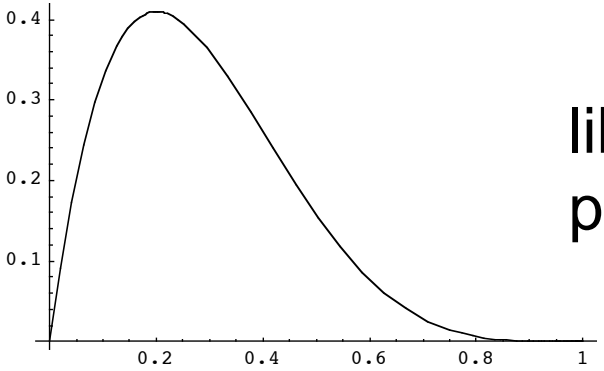
  - Posterior distribution

$$p(b \mid h) \sim p(b)\, p(h \mid b)$$

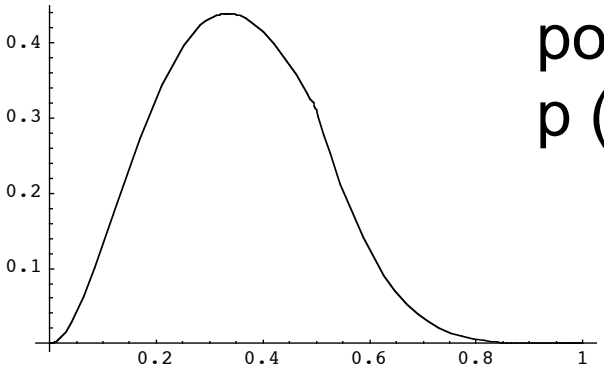
$$\sim \binom{5}{h} b^h (1-b)^{5-h} \left(2 - 4\left|b - 0.5\right|\right)$$



prior
p (b)



likelihood
p (h=1 | b)



posterior
p (b | h=1)

Monte Carlo Methods

- For many systems (especially complex ones), we can not write down a likelihood function
- Simulate a LOT of data to find parameters that are most likely to produce your data.
- THE PROBLEM: For many systems the probability of producing EXACTLY your data might be zero
- We need to find parameters that are most likely to produce data that is SIMILAR TO the observed data

Approximate Bayesian Computation

- Calculate summary statistics from observed data
- Define and parameterize a model
- Define a prior distribution over parameter values
- Simulate many of datasets, drawing parameter values from the prior
- Calculate the same summary statistics from each simulated dataset
- Identify the simulations that produce data that is “most like” the observed data
- Parameter values associated with these simulations approximate the posterior distribution

Approximate Bayesian Computation

- THE PROBLEM: How do we identify simulated data that is “most like” our observed data

- If the data is discrete (e.g., number of times a coin toss comes up heads), and the number of possible outcomes is small, we simply require that the simulated and observed data be identical
  - For example, if we were simulating sets of five coin tosses, we would collect all of the simulations for which heads came up exactly one time

Approximate Bayesian Computation

- MANY POSSIBLE OUTCOMES:
  - If we require an exact match, we might have to throw away almost all of our simulations!

- EXAMPLE
  - DNA Sequence data - we could run simulations (coalescent or traditional) to generate DNA sequences, and collect the simulations that produce sequences exactly like the ones we see in nature

1. — ATTCTTCAGGATATAGCTACTTACGGGCTAGGCAT —
2. — ATTCTTGAGGACTTAGCTACTTACAGGCTAGGCAT —
3. — ATTCTTGAGGACTTAGCTACTTACAGGCTAGCCAT —
4. — ATTCTTCAGGACTTAGCTACTTACGGGCTAGGCAT —

Genealogical topologies

n	Random-joining trees (nodes ordered in time)	Unrooted bifurcating trees (nodes not ordered in time)
2	1	1
3	3	1
4	18	3
5	180	15
6	2700	105
7	56700	945
8	1587600	10395
9	57153600	135135
10	2571912000	2027025
100	1.37×10^{284}	1.70×10^{182}
1000	3.02×10^{4831}	1.91×10^{2860}

Approximate Bayesian Computation

- ALSO, CONTINUOUS DATA:
  - If we require an exact match, we will never collect any simulations at all (an infinite number of possible outcomes)

- So, in practice, we will do two things
  - Calculate summary statistics from simulated and observed data
  - Collect simulations for which the summary statistics are CLOSE TO the summary statistics calculated from the observed data

1. [Observed Data] --> SS_{obs}
2. θ_k (sampled from prior) --> [Simulated Data] --> SS_k
3. If $|SS_{obs} - SS_k| < \delta$, add the point θ_k to the posterior
4. Repeat 2 and 3 many times
5. Estimate the posterior density from the θ_k that met the criterion

Approximate Bayesian Computation

- But, what if we have multiple summary statistics?
- We could use a weighted euclidean distance:
  - Calculate n different summary statistics ss_i ($1 \le i \le n$)
  - collect simulation k if

$$\sum_{i=1}^{n} w_i (ss_{k,i} - ss_{obs,i})^2 < \delta$$

  - How do we choose the values of the w_i?

Approximate Bayesian Computation

- Defining the appropriate euclidean distance: $\sum_{i=1}^{n} w_i (ss_{k,i} - ss_{obs,i})^2$
- w_i should be large for “good” summary statistics
- w_i should be small for “bad” summary statistics
- summary statistics are good if they are
  - not noisy $Var\left(ss_{k,i} \mid \theta\right) \quad is\ small$
  - responsive to changes in parameters $\frac{\partial}{\partial \theta} E\left(ss_{k,i} \mid \theta\right) \quad is\ large$
  - not redundant $Co\,\mathrm{var}\left(ss_{k,i}, ss_{k,j} \mid \theta\right) \quad is\ small$

Approximate Bayesian Computation

- Choosing good summary statistics
- If the focus is the macroscopic behavior
  - choose the features that you want to explain
- If the focus is the microscopic rules
  - choose statistics that are sensitive to changes in those rules
- In a complex system, how would we ever be able to choose?
- In general, what is a good statistic will vary across the parameter space

Approximate Bayesian Computation

- The algorithm

1. Rescale parameters so that you are equally sensitive to errors in each parameter
2. Regress summary statistics on parameters
3. Orthogonalize the residuals
4. Rescale the transformed summary statistics so that they all have equal variance
5. Compute the Fisher Information Matrix for the parameters at the center of the prior
6. Calculate the euclidean distance for each simulation
7. Construct the posterior distribution from the closest simulations
8. Repeat from step 2 until the posterior converges

Approximate Bayesian Computation

1. Rescale parameters so that you are equally sensitive to errors in each parameter

Parameters θ_1 and θ_2

Parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$

Cost function $C = c_1 \left(\hat{\theta}_1 - \theta_1\right)^2 + c_2 \left(\hat{\theta}_2 - \theta_2\right)^2$

Define new, rescaled parameters $\phi_1 = \sqrt{c_1}\ \theta_1$ and $\phi_2 = \sqrt{c_2}\ \theta_2$ such that

$C = (\hat{\phi}_1 - \phi_1)^2 + (\hat{\phi}_2 - \phi_2)^2$

Approximate Bayesian Computation

2. Regress summary statistics on parameters

Summary statistics $s_{k,i}$ and rescaled parameters $\phi_{k,j}$

k is the index over simulation runs

i are the different statistics

j are the different parameters

Do multiple linear regression, so that we express the summary statistics as

$$s_{k,i} = s_{0,i} + \sum_j b_{i,j} \phi_{k,j} + \epsilon_{k,i}$$

$$s_k = s_0 + b\phi_k + \epsilon_k$$

(matrix notation)

where $\sum_k \sum_i \epsilon_{k,i}^2$ is minimized

Approximate Bayesian Computation

3. Orthogonalize the residuals

Construct the variance-covariance matrix of the ϵ_k terms: $\mathcal{K}$

Diagonalize the matrix:

$$\mathcal{K} = \Lambda^{-1} \mathcal{V} \Lambda$$

$$s_k = s_0 + b\phi_k + \epsilon_k$$

$$\zeta_k = \Lambda (s_0 + b\phi_k + \epsilon_k)$$

4. Rescale the transformed summary statistics so that they all have equal variance

$$\xi_k = \mathcal{V}^{-1} \zeta_k = \mathcal{V}^{-1} \Lambda (s_0 + b\phi_k + \epsilon_k)$$

Approximate Bayesian Computation

5. Compute the Fisher Information Matrix for the parameters at the center of the prior

Fisher Information: how much information about a parameter do we get from an observation

$$\mathcal{I}_{m,n}(\phi) = E\left[-\frac{\partial}{\partial \xi_m}\frac{\partial}{\partial \xi_n}\mathrm{Log}(P(\xi \mid \phi))\right]$$

Calculate euclidean distances for each simulation $\mathcal{D}_k = (\xi_k - \xi_{\mathrm{obs}})^2$

Collect the simulations with the smallest value of $\mathcal{D}$ (say, 1%)
Now, look at the distribution of parameters ϕ_k associated with these simulations

The nice thing is, all we have to do is calculate the variance-covariance matrix of this

$\mathcal{I}$ = variance-covariance matrix of the selected ϕ_k

Approximate Bayesian Computation

6. Calculate the euclidean distance for each simulation

Change each summary statistic vector into a vector of distances from the observed data:

$$r_k = |\xi_k - \xi_{\text{obs}}|$$

Invert the Fisher Information Matrix (calculate I^{-1})
Calculate the euclidean distance

$$\mathcal{D}_k = \boldsymbol{r}_k \, I^{-1} \, \boldsymbol{r}_k$$

Approximate Bayesian Computation

7. Construct the posterior distribution from the closest simulations

Rank order the simulations in order of increasing $\mathcal{D}_k$

The values of ϕ_k with the smallest $\mathcal{D}$ form the basis of the posterior

Select ~ 0.1%

The important thing is to collect enough points that you have reasonable sampling from the posterior, but not so many points that you are including things that don't actually match your data very well

Approximate Bayesian Computation

8. Repeat from step 2 until the posterior converges

Many of the quantities that we have calculated (the variance-covariance matrices, the linear regression of statistics against parameters, the Fisher Information Matrix) will be different in different regions of the parameter space. So, first, we calculated these using the whole prior.

Now, we have a set of points that are sampled from the posterior. These points will be concentrated in the region that we care about. The next time, the regression and covariance matrices should be based on points in this region. One way to do it is to use points in the prior that are closest to the points in our posterior. For example, look at each point in the posterior, and use the 10 points in the prior that are closest to it.

After a few rounds, the posterior should converge.