STSCI 4780:
Introduction to Bayesian computation

Tom Loredo, CCAPS & SDS, Cornell University

2020-02-27

Recap:
Composite hypotheses

- Simple vs. composite hypotheses
- Classes of problems: single model, multimodel, model checking
- Uncertainty propagation (composite in multivariate cases)
- Marginalization
- Prediction
- Model comparison, model averaging

Theme: Parameter space volume

Bayesian calculations sum/integrate over parameter/hypothesis space! This is *the signature feature* of the Bayesian approach.

(Frequentist calculations average over *sample* space & typically *optimize* over parameter space.)

- Credible regions integrate over parameter space
- Uncertainty propagation integrates over parameter space
- Marginalization weights the profile likelihood by a volume factor for the nuisance parameters
- Prediction integrates over parameter space
- Model (marginal) likelihoods & Bayes factors have Ockham factors resulting from parameter space volume factors

Many/most interesting hypotheses are really *composite*. Many virtues of Bayesian methods can be attributed to accounting for the "size" of parameter spaces when considering composite hypotheses. This idea does not arise naturally in frequentist statistics (but it can be added "by hand").

Roles of the prior

Prior has two roles

- Incorporate any relevant prior information
- Convert likelihood from "intensity" to "measure"
  $\rightarrow$ account for *size of parameter space*

Physical analogy

$$\begin{aligned} \text{Heat} \quad Q &= \int d\vec{r}\,[\rho(\vec{r})c(\vec{r})]\,T(\vec{r}) \\ \text{Probability} \quad P &\propto \int d\theta\; p(\theta)\mathcal{L}(\theta) \end{aligned}$$

Maximum likelihood focuses on the "hottest" parameters.
Bayes focuses on the parameters with the most "heat."

A high-T region may contain little heat if ρc is low or if its volume is small.

A high-$\mathcal{L}$ region may contain little probability if its prior is low or if its volume is small.

Notation focusing on computational tasks

$$
\begin{aligned}
p(\theta|D, M) &= \frac{p(\theta|M)p(D|\theta, M)}{p(D|M)} \\
&= \frac{\pi(\theta)\mathcal{L}(\theta)}{Z} = \frac{q(\theta)}{Z}
\end{aligned}
$$

- M = model specification (context)
- D specifies observed data
- θ = model parameters
- $\pi(\theta)$ = prior pdf for θ
- $\mathcal{L}(\theta)$ = likelihood for θ (likelihood function)
- $q(\theta) \equiv \pi(\theta)\mathcal{L}(\theta)$ = "quasiposterior"
- $Z = p(D|M)$ = (marginal) likelihood for the model

Marginal likelihood:

$$Z = \int d\theta\ \pi(\theta)\mathcal{L}(\theta) = \int d\theta\ q(\theta)$$

Use "Skilling conditional" for common conditioning info:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \qquad || \ M$$

Suppress such conditions when clear from context

Bayesian computational tasks

Multiply, normalize

$$Z = \int d\theta \; \pi(\theta)\mathcal{L}(\theta)$$

Optimize

$$\hat{\theta} = \arg\max_{\theta} p(\theta|D) = \arg\max_{\theta} q(\theta)$$

Moments

$$\mathbb{E}(\theta^{(k)}) = \frac{1}{Z}\int d\theta \; \theta^{(k)} \times q(\theta) \quad \text{for } k\text{'th param}$$

$$\bar{\theta} \equiv \langle\theta\rangle \equiv \mathbb{E}(\theta); \qquad \sigma_\theta^2 \equiv \mathrm{Var}(\theta) = \mathbb{E}(\theta^2 - \bar{\theta}^2)$$

Credible regions

For given probability C, find a region Δ with

$$C = \frac{1}{Z}\int_{\Delta} d\theta\; q(\theta)$$

Eliminate nuisance parameters

For $\theta = (\phi, \eta)$,

$$p(\phi|D,M) = \int d\eta\; p(\phi,\eta|D,M) = \frac{1}{Z}\int d\eta\; q(\phi,\eta)$$

Propagate uncertainty

Model has parameters θ; what can we infer about $\psi = \Psi(\theta)$?

$$\begin{aligned} p(\psi|D,M) &= \int d\theta\; p(\psi,\theta|D,M) \;=\; \int d\theta\; p(\theta|D,M)\, p(\psi|\theta,M) \\ &\propto \int d\theta\; q(\theta)\,\delta[\psi - \Psi(\theta)] \qquad [\textit{single-valued case}] \end{aligned}$$

Prediction

Given a model with parameters θ and present data D, predict future data D' (e.g., for *experimental design*):

$$p(D'|D,M) \;=\; \int d\theta\; p(D',\theta|D,M) \;=\; \int d\theta\; p(\theta|D,M)\, p(D'|\theta,M)$$

Model comparison

Given rival models M_1 and M_2 with parameters θ_1 and θ_2, the Bayes factor is

$$B_{12} = \frac{p(D|M_1)}{p(D|M_2)} = \frac{\int d\theta_1\; \pi_1(\theta_1)\mathcal{L}_1(\theta_1)}{\int d\theta_2\; \pi_2(\theta_2)\mathcal{L}_2(\theta_2)} = \frac{\int d\theta_1\; q_1(\theta_1)}{\int d\theta_2\; q_2(\theta_2)} = \frac{Z_1}{Z_2}$$

Parameter space integrals

For model with m parameters, we need to evaluate integrals like:

$$\int d^m\theta\, g(\theta)\, \pi(\theta)\, \mathcal{L}(\theta) \quad = \quad \int d^m\theta\, g(\theta)\, q(\theta)$$

- $g(\theta) = 1 \rightarrow Z = p(D|M)$ (norm. const., model likelihood)
- $g(\theta) = \theta \rightarrow$ posterior mean for θ
- $g(\theta) =$ 'box' $\rightarrow$ probability $\theta \in$ credible region
- $g(\theta) = 1$, integrate over $< m$ params $\rightarrow$ marginal posterior
- $g(\theta) = \delta[\psi - \psi(\theta)] \rightarrow$ propagate uncertainty to $\psi(\theta)$

Except for optimization, Bayesian computation amounts to *computing the expectation of some function $g(\theta)$ with respect to the posterior dist'n for θ*

Contrast with frequentist computation, which integrates over *sample space*, e.g., via Monte Carlo simulation of data

Bayesian Computation Menu

Large sample size, N: Laplace approximation

- Approximate posterior as multivariate normal $\rightarrow$ det(covar) factors
- Uses ingredients available in χ^2/ML fitting software (MLE, Hessian)
- Often accurate to $O(1/N)$ (better than $O(1/\sqrt{N})$)

Modest-dimensional models ($m \lesssim 10$ to 20)

- Quadrature, cubature, adaptive cubature
- IID Monte Carlo integration (importance & stratified sampling, adaptive importance sampling, quasirandom MC)

High-dimensional models ($m \gtrsim 5$): Non-IID Monte Carlo

- Posterior sampling — create RNG that samples posterior
  - Markov Chain Monte Carlo (MCMC) is the most general framework
- Sequential Monte Carlo (SMC)
- Approximate Bayesian computation (ABC)
- Variational Bayes/variational inference
- ...

The Laplace approximation (1-D)

Motivation

- Approximate integrand in neighborhood of the peak, $\hat{\theta}$, by matching the function value and two derivatives there
- Match derivatives of the *log* integrand, since we want PDFs to be non-negative: For $e^{\Lambda(\theta)} = g(\theta)q(\theta)$, Taylor series to 2nd order gives

$$\Lambda(\theta) \approx \Lambda(\hat{\theta}) + \underbrace{\boxed{\Lambda'(\hat{\theta})}}_{\text{vanishes}}(\theta - \hat{\theta}) + \frac{1}{2}\Lambda''(\hat{\theta})(\theta - \hat{\theta})^2$$

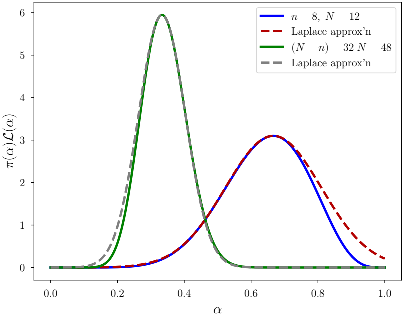
  Leading order dependence on θ is *Gaussian* with:
  mode (mean) $\hat{\theta}$ and *variance* $\sigma^2 = -1/\Lambda''(\hat{\theta})$

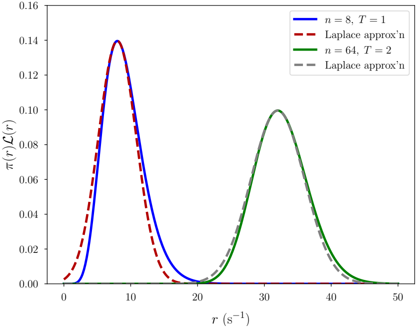
- In many settings, asymptotics $\rightarrow$ expect $q(\theta)$ to be $\approx$ Gaussian so $gq \approx$ Gaussian if $g(\theta)$ varies slowly

LA fits a Gaussian function to the peak of the *integrand*, and estimates the original integral using the fitted Gaussian

Example—Laplace approximation for Z

Beta, gamma distribution examples:





Analytic Laplace approximations

Let $f(\theta) \equiv g(\theta)q(\theta) = e^{\Lambda(\theta)}$, so $\Lambda(\theta) = \ln f(\theta)$. Note that

$$\frac{\mathrm{d}\Lambda}{\mathrm{d}\theta} = \frac{1}{f}\frac{\mathrm{d}f}{\mathrm{d}\theta}; \qquad \frac{\mathrm{d}^2\Lambda}{\mathrm{d}\theta^2} = \frac{1}{f}\frac{\mathrm{d}^2 f}{\mathrm{d}\theta^2} - \frac{1}{f^2}\frac{\mathrm{d}f}{\mathrm{d}\theta} = \frac{1}{f}\frac{\mathrm{d}^2 f}{\mathrm{d}\theta^2} \text{ at } \hat{\theta}$$

Tip: If $f(\theta) = Ck(\theta)$,

$$\frac{1}{\sigma^2} = - \left. \frac{1}{k}\frac{\mathrm{d}^2 k}{\mathrm{d}\theta^2}\right|_{\hat{\theta}}$$

I.e., we need only keep the θ-dependent "kernel" of $f(\theta)$

Tip: Express derivatives in terms of factors multiplying k when possible

- Normal: $k(\theta) = \exp\left[-\frac{(\theta-\mu)^2}{2\sigma^2}\right] \quad \Rightarrow \quad \hat{\theta} = \mu;\ -k''/k = 1/\sigma^2$
- Gamma: $k(r) = r^{a-1}e^{-r/s} \quad \Rightarrow \quad \hat{r} = (a-1)s;\ -k''/k = \frac{a-1}{\hat{r}^2}$
  so $\sigma^2 = (a-1)s^2$
- Beta: $k(\alpha) = \alpha^n(1-\alpha)^{N-n}$ ...

Numerical Laplace approximations

Find $\hat{\theta}$ with an optimizer

Estimate derivatives numerically via *finite differencing*, e.g.,

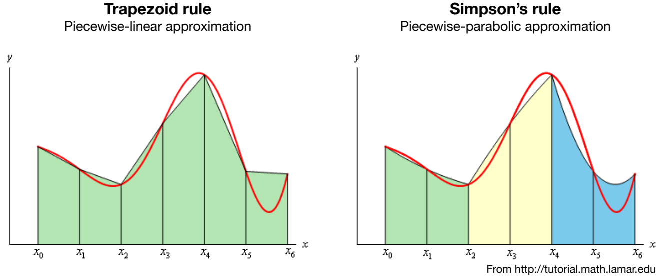
$$
\begin{aligned}
f'(x) &\approx \frac{f(x+h) - f(x)}{h} \quad \text{(forward difference)} \\
&\approx \frac{f(x+h/2) - f(x-h/2)}{h} \quad \text{(central difference)}
\end{aligned}
$$

2nd order central differencing gives:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Quadrature rules

Trapezoid and Simpson's rules



Trapezoid rule:

$$\int \mathrm{d}x\, f(x) \approx \Delta x \left[\frac{1}{2} f(x_0) + f(x_1) + f(x_2) + \cdots + \frac{1}{2} f(x_n) \right]$$

Simpson's rule:

$$\int \mathrm{d}x\, f(x) \approx \frac{\Delta x}{3} \left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + f(x_n) \right]$$

Generic quadrature rule

Weighted sum of integrand at nodes $\{x_i\}$:

$$\int_a^b \mathrm{d}x\, f(x) \approx \sum_{i=0}^{n} w_i f(x_i)$$

Closed rules have $x_1 = a$ and $x_n = b$; *open* rules have all nodes inside (a, b) (useful for infinite ranges)

Error terms (exact integral − approx):

$$\begin{aligned} \text{Trapezoid:} \qquad & -\frac{b-a}{12}\left(\frac{b-a}{n}\right)^2 f''(\xi) \\ \text{Simpson's:} \qquad & -\frac{b-a}{180}\left(\frac{b-a}{n}\right)^4 f^{(4)}(\xi) \end{aligned}$$

for *some* (unspecified) ξ in the interval.

In practice, error is often estimated by applying rules with two different choices of n

Gaussian quadrature rules

Write integrand $f(x) = h(x)\omega(x)$, where a simple *weight function* $\omega(x)$ captures (very) rough behavior (e.g, constant, polynomial, exponential, Gaussian)

Absorb ω into quadrature rule weights:

$$\begin{aligned}\int_a^b \mathrm{d}x\, f(x) &\approx \sum_{i=1}^{n} w_i f(x_i) \\ &= \sum_{i=1}^{n} w_i' \frac{f(x_i)}{\omega(x_i)} \quad \text{with } w_i' = w_i \omega(x_i)\end{aligned}$$

Pick $2n$ values $\{(x_i, w_i')\}$ to make the quadrature exact for polynomial $h(x)$—this can work up to degree $2n-1$

$\{(x_i, w_i')\}$ determined by *orthogonal polynomials* wrt $\omega(x)$

Error is proportional to $f^{(2n)}(\xi)$ and falls quickly with n

Common weight functions for various types of intervals:

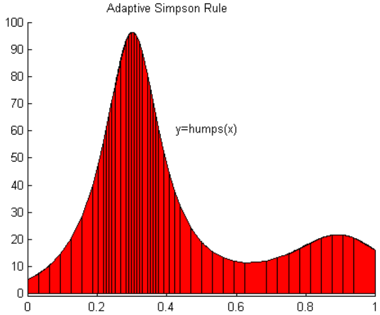
- $[a, b]$: $\omega(x) = 1 \quad \Rightarrow \quad$ Gauss-Legendre
- $[-1, 1]$: $\omega(x) = (1-x)^\alpha(1+x)^\beta \quad \Rightarrow \quad$ Gauss-Jacobi
- $[0, \infty]$: $\omega(x) = e^{-x} \quad \Rightarrow \quad$ Gauss-Laguerre
- $[0, \infty]$: $\omega(x) = x^\alpha e^{-x} \quad \Rightarrow \quad$ Gen. Gauss-Laguerre
- $[-\infty, \infty]$: $\omega(x) = e^{-x^2} \quad \Rightarrow \quad$ Gauss-Hermite

Rules are open, with *unequally spaced* nodes (at roots of orthogonal polynomials); note rules are available for *infinite* intervals

Gaussian quadratures accurately integrate non-polynomial functions by factoring out the weight function

Adaptive quadrature

1. Estimate integral over $[a, b]$
2. Estimate error (e.g., using higher-n rule that reuses nodes)
3. If error too large, subdivide interval, and repeat in subintervals
4. When error criterion met, sum subinterval quadratures



From J. T. King

`scipy.integrate.quad()` uses adaptive Clenshaw-Curtis or Fourier rules

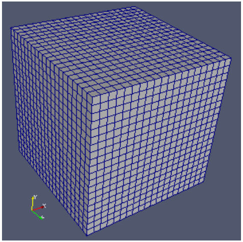
Cubature rules for modest-D integrals

Quadrature rules for 1-D integrals (with weight function $h(\theta)$):

$$\begin{aligned}\int d\theta\, f(\theta) &= \int d\theta\, h(\theta)\, \frac{f(\theta)}{h(\theta)} \\ &\approx \sum_i w_i\, f(\theta_i) + O(n^{-2}) \text{ or } O(n^{-4})\end{aligned}$$

Smoothness $\rightarrow$ fast convergence in 1-D

Curse of dimensionality: Cartesian product rules converge slowly, $O(n^{-2/m})$ or $O(n^{-4/m})$ in m-D



Wikipedia

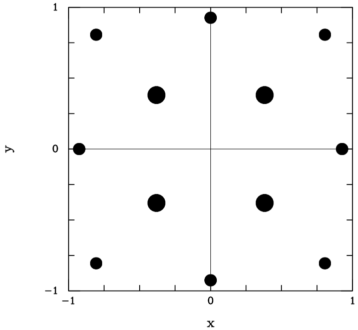
Monomial Cubature Rules

Seek rules exact for multinomials ($\times$ weight) up to fixed monomial degree with desired lattice symmetry; e.g., for a 7th-degree rule:

$$f(x, y, z) = \mathrm{MVN}(x, y, z) \sum_{ijk} a_{ijk} x^i y^j z^k \qquad \text{for } i + j + k \leq 7$$

Number of points required grows much more slowly with m than for Cartesian rules (but still quickly)

A 7th order rule in 2-d



See:

- Ronald Cools's Encyclopaedia of Cubature Formulas
- quadpy

Adaptive Cubature

- Subregion adaptive cubature: Use a pair of monomial rules (for error estim'n); recursively subdivide regions w/ large error. Concentrates points where most of the probability lies. See: `ADAPT`, `CUHRE`, `BAYESPACK`, `Cuba`, `cubature`, quadpy; various languages
- Adaptive grid adjustment: Naylor-Smith method
  Iteratively update abscissas and weights to make the (unimodal) posterior approach the weight function.

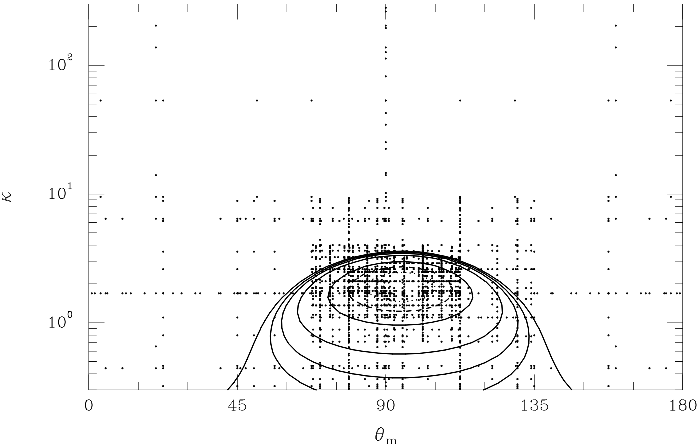
These provide diagnostics (error estimates or measures of reparameterization quality).

nodes used by ADAPT's 7th order rule

$$2^d + 2d^2 + 2d + 1$$

Dimen	2	3	4	5	6	7	8	9	10
# nodes	17	33	57	93	149	241	401	693	1245

Analysis of Galaxy Polarizations



TL, Flanagan, Wasserman (1997)