Computation (chapter 10)

- Crude estimation
  - posterior mode
  - “empirical Bayes´´
- How many simulation draws are needed
  - Monte Carlo error

Slide 1

About distributions

- Log-densities
  - to prevent over- and underflows in floating point computation log-densities are often used
  - exponentiation should be made last
  - eg. in Metropolis-algorithm instead of computing ratio of densities, compute difference of log-densities

Slide 2

About distributions

- Normalized and unnormalized distributions
  - often computing normalization is difficult
  - often unnormalized is sufficient for computation
  - $q(\theta|y)$ is unnormalized density if $q(\theta|y)/p(\theta|y)$ is constant depending only y
  - eg. $p(y|\theta)p(\theta)$ is unnormalized posterior density

Slide 3

Crude estimation

- Before using complicated computational methods it is good to make crude estimate
  - sensibility check
  - initial guess for more sophisticated models
- Hierarchical models
  - approximate hyperparameters

Slide 4

- Posterior modes
  - find joint- or marginal mode(s) using optimization algorithm
  - normal, mixture normal, etc. approximation

Crude estimation

- If model is so complex, that it is difficult to make simple posterior approximation

  ⟶ start with simpler model
  - simpler model gives baseline accuracy
  - works as a sensibility check

Slide 5

Monte Carlo - history*

- Used before computers, eg.
  - Buffon (1700's)
  - De Forest, Darwin, Galton (1800's)
  - Pearson (1800's)
  - Gosset (ie. Student, 1908)
- "Monte Carlo method" terms was proposed by Metropolis, von Neumann or Ulam in the end of 1940's
  - Metropolis, Ulam and von Neumann worked together in A-bomb project
  - Metropolis and Ulam, "The Monte Carlo Method", 1949
- Users of Bayesian methods started to have enough cheap computational in 1990's
  - before usage was rare, although some Bayesians developed MCMC methods

Slide 6

Monte Carlo

- Sample from distribution
- Compute and plot
  - averages and variances
  - quantiles
  - histograms
  - marginals
  - etc.

Slide 7

How many simulation draws are needed?

- Expectation

$$\mathrm{E}(\theta) \approx \frac{1}{L} \sum_{l} \theta^{(l)}$$

if L large and $\theta^{(l)}$ independent samples, may assume asymptotic normality with variance σ_θ^2/L

  - this variance is independent of number of dimensions
  - combined variance is sum of data variance and simulation variance

$$\sigma_\theta^2 + \sigma_\theta^2/L = \sigma_\theta^2(1 + 1/L)$$

  - e.g. if $L = 100$, simulation inflates the variance by $\sqrt{1 + 1/L} = 1.005$
  - remember the counter examples to asymptotic normality

Slide 8

How many simulation draws are needed?

- Posterior probability

$$p(\theta \in A) \approx \frac{1}{L}\sum_l I(\theta^{(l)} \in A)$$

where $I(\theta^{(l)} \in A) = 1$ if $\theta^{(l)} \in A$

  - $I(\cdot)$ binomially distributed with parameters $p(\theta \in A)$
  - deviation is $\sqrt{p(1-p)/L}$ (s. 577)
  - if $L = 100$ and p about 0.5, $\sqrt{p(1-p)/L} = 0.05$ i.e. 5%-unit accuracy (deviation)
  - with $L = 2500$ samples, accuracy 1%-unit

Slide 9

- To estimate small probabilities need many samples
  - enough many samples have to have $\theta^{(l)} \in A$, i.e. $L \gg 1/p$

How many simulation draws are needed?

- Quantiles
  - for q-quantile, choose a for which

$$p(\theta < a) = q$$

ie

$$\frac{1}{L}\sum_l I(\theta^{(l)} < a) \approx q$$

Slide 10

  - for good estimate, need many samples for which $\theta^{(l)} < a$ or $\theta^{(l)} > a$, and thus $L \gg 1/q$ or $L \gg 1/(1-q)$
  - cf. previous slide

How many simulation draws are needed?

- Monte Carlo error can be estimated using simulation too
  - use approximative distribution for samples and use Bayesian inference
- e.g. non-parametric approach using Dirichlet-model (Rubin, 1981)
  - works for non-normal distributions, too

Slide 11

How many simulation draws are needed?

- Less samples are needed if marginalisation is used
  - density can be often factored and lowest level marginalized

$$\mathrm{E}(\theta) \approx \frac{1}{L}\sum_{l} \mathrm{E}(\theta|\phi^{(l)})$$

    where $\phi^{(l)}$ are samples from the marginal of hyper parameters

  - almost always can be used for predictive densities

Slide 12

- SAT-example
  - probability that effect of school A is larger than 50
  - with plain simulation 3 samples of 10000 larger than 50
  - computing analytically $\Pr(\theta_1 > 50|\mu, \tau, y)$, good accuracy achieved with 200 samples

Direct simulation

- Direct simulation produces independent samples
- Requirement is (pseudo) random number from uniform distribution
  - in Bayesian analysis good pseudo random number generators when used appropriately are good enough
    - eg. Matlab's default generator is excellent (Mersenne Twister algorithm) and for special cases latest version includes alternatives

Slide 13

Direct simulation

- Uniform random numbers can be used to get samples from some basic distributions using transformations and factoring (see e.g. appendix A)
- 1–3 dimensionals can be handled also with inverse-cdf/grid-approach

Slide 14

Example of transformation*

- Box-Muller -methods:
  If U_1 and U_2 are independent samples from distribution $\mathrm{U}(0,1)$, and

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

  then X_1 and X_2 are independent from the distribution $\mathrm{N}(0,1)$
  - not the fastest choice due to trigonometric computations
  - for normal distribution more than ten different methods
  - Matlab uses fast Ziggurat method
- For basic distributions usually functions available

Slide 15

Grid sampling

- Generalizes inverse-cdf
- Suffers from curse of dimensionality

Slide 16

Grid sampling

- E.g.: SAT
  - 10 parameters
  - if location of essential posterior mass is unknown
    - lower and upper limits for discretization need to be loose
    - need to have enough grid points, so that some of them falls to high density area
  - e.g. 1000 grid points per dimension
    $\longrightarrow$ 1000^{10} = total of 1e30 grid points
  - Matlab computes normal density function about 4 million times per second
    $\longrightarrow$ evaluation in all grid points would take about 1e18 years

Slide 17

Curse of dimensionality

- Example
  - reasonable guess having posterior mass in 1/3 of the guessed limits
    - 1 parameter $\longrightarrow$ 1/3 evaluations in interesting area
    - 2 parameters $\longrightarrow$ 1/9 evaluations in interesting area . . .
    - 3 parameters $\longrightarrow$ 1/27 evaluations in interesting area . . .
    - d parameters $\longrightarrow$ $1/3^d$ evaluations in interesting area . . .

Slide 18

Markov chain Monte Carlo (MCMC) (chapter 11)

- Markov chain
  - a sequence of variables $\theta^1, \theta^2, \ldots$, for which with all t, distribution of θ^t depends only on θ^{t-1}
  - starting point θ^0
  - transition distribution $T_t(\theta^t|\theta^{t-1})$
  - suitably constructed Markov chain converges to unique stationary distribution $p(\theta|y)$

Slide 19

- Pros/cons
  - + general use
  - + chain tends to find where the mass is
  - - dependent samples
  - - construction of efficient transition distribution may be difficult

Metropolis-algorithm

- Metropolis-algorithm and its generalizations are base of all MCMC-algorithms
- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick proposal θ^* from proposal distribution $J_t(\theta^*|\theta^{t-1})$
     proposal distribution has to be symmetric, ie.
     $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) compute ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$
  - transition distribution is mixture of point a point mass at $\theta^t = \theta^{t-1}$ and a weighted version of the proposal distribution $J_t(\theta^*|\theta^{t-1})$

Slide 20

Metropolis-algorithm

Slide 21

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick proposal θ^* from proposal distribution $J_t(\theta^*|\theta^{t-1})$
     proposal distribution has to be symmetric, ie.
     $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) compute ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$
  - instead of $p(\theta|y)$, unnormalized $q(\theta|y)$ can be used
  - step c is done by using uniform random number $\mathrm{U}(0, 1)$
  - rejection of proposal is also one iteration (ie t increases by one)

Metropolis algorithm

Slide 22

- Example: one observation (y_1, y_2)
  - normal model with unknown mean and known variance
  $$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \Bigg| y \sim \mathrm{N}\left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$
  - proposal distribution $J_t(\theta^*|\theta^{t-1}) = \mathrm{N}(\theta^*|\theta^{t-1}, 0.8^2)$
- Esim7_1.m

Burn-in and convergence diagnostics

- How long it does to chain converge?

  → burn-in = remove samples from the beginning of the chain

Slide 23

Dependent samples and auto-correlation

- Autocorrelation describes how much samples correlate on average with samples with certain lag
  - how quickly chain forgets previous states
  - how efficient algorithm is
- Autocorrelation can be used to estimate the effective number of samples

Slide 24

Why Metropolis-algorithm works

- Intuitively more samples are accepted from higher density areas

1. Prove, that simulated series is Markov-chain, which has unique stationary distribution
2. Prove, that stationary distribution is desired target distribution

Slide 25

Why Metropolis-algorithm works

1. Prove, that simulated series is Markov-chain, which has unique stationary distribution; show that chain chain is
   a) *irreducible*
      - positive probability to reach any state from any other state
   b) *aperiodic*
      - return time i can be any number
      - holds for random walk and any proper distribution except for trivial exceptions
   c) *recurrent / not transient*
      - probability to return to state i is 1
      - holds for random walk and any proper distribution except for trivial exceptions

Slide 26

Why Metropolis-algorithm works

v

2. Prove, that stationary distribution is desired target distribution
   - start at time $t-1$ by picking θ^{t-1} from the target distribution $p(\theta|y)$
   - choose two points θ_a and θ_b, which have been picked from $p(\theta|y)$ and named so that $p(\theta_b|y) \geq p(\theta_a|y)$
   - density for transition from θ_a to θ_b

Slide 27

$$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_t(\theta_b|\theta_a),$$

where acceptance probability is 1 due to selected naming

- density for transition from θ_b to θ_b

$$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|y)J_t(\theta_a|\theta_b)\left(\frac{p(\theta_a|y)}{p(\theta_b|y)}\right) \\ &= p(\theta_a|y)J_t(\theta_a|\theta_b), \end{aligned}$$

which is same as for transition from θ_a to θ_b since $J_t(\cdot|\cdot)$ is symmetric

- since joint distribution is symmetric, marginal of θ^t and θ^{t-1} are same and thus $p(\theta|y)$ is stationary distribution of the Markov-chain

Slide 28

Metropolis-Hastings algorithm

- Generalization of Metropolis algorithm to asymmetric proposal distributions
- sometimes Hastings dropped
- asymmetry is taken into account in computation of acceptance probability

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{t-1})}{p(\theta^{t-1}|y)/J_t(\theta^{t-1}|\theta^*)} = \frac{p(\theta^*|y)J_t(\theta^{t-1}|\theta^*)}{p(\theta^{t-1}|y)J_t(\theta^*|\theta^{t-1})}$$

Slide 29

  - possible to use more efficient proposal distributions
  - proof as previously, but name θ_a and θ_b so that $p(\theta_b|y)J_t(\theta_a|\theta_b) \geq p(\theta_a|y)J_t(\theta_b|\theta_a)$

Metropolis-Hastings -algorithm

- Generalization of Metropolis algorithm to asymmetric target distribution
- More efficient algorithms
  - proposal distribution can resemble more target distribution
    - more efficient acceptance
  - eg. proposal distribution which leans on direction of gradient (*Langevin-Hastings-algorithm*)
    - chain has tendency to travel towards higher mass

Slide 30

Metropolis-Hastings-Green algorithm (s. 338-339)

- Reversible jump Markov chain Monte Carlo (RJMCMC)
- Metropolis-Hastings generalised to jumps between different parameter spaces
  - trans-dimensional method

Slide 31

Metropolis-Hastings

- Ideal proposal distribution is the target distribution
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance 1
  - independent samples
- Good proposal resembles the target distribution
- Good scale can be selected by using rejection rate of 60–90%

Slide 32

Metropolis-Hastings

- Updates
  - jointly
  - blocked
  - single-component

Slide 33

Gibbs sampling

- Called Gibbs sampling by Geman & Geman (1984)
  in physics also known as heat bath method
- Gibbs sampling is special case of Metropolis-Hastings
  - single component (usually)
  - proposal distribution is the full conditional distribution of given parameter
    $\longrightarrow$ proposal and target distributions are same
    $\longrightarrow$ acceptance probability is 1

Slide 34

Gibbs sampling

- Sample from each full conditional full conditional distribution

$$p(\theta_j|\theta_{-j}^{t-1}, y)$$

where θ_{-j}^{t-1} is

$$\theta_{-j}^{t-1} = (\theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1})$$

Slide 35

- in one time step t update all parameters θ_j (although not necessary)

Gibbs sampling

- Example: one observation (y_1, y_2)
  - normal model with unknown mean and known variance

$$\left.\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}\right| y \sim \mathrm{N}\left(\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

  - conditional distributions (book s. 86 and 288)

Slide 36

$$\begin{aligned}\theta_1|\theta_2, y &\sim \mathrm{N}(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)\\ \theta_2|\theta_1, y &\sim \mathrm{N}(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)\end{aligned}$$

- Esim7_2.m

Gibbs sampling

- Use of semi-conjugate priors in hierarchical models, produces often nice conditional distributions
  - WinBUGS/OpenBUGS
- No tunable algorithm parameters
- If some of the conditionals not in nice form may use e.g.. grid sampling, Metropolis-Hastings or slice sampling
- Sometimes blocking used (cf. Metropolis-Hastings)

Slide 37

Burn-in and convergence diagnostics

- Start with visual inspection
  - Esim7_3.m

Slide 38

Use of several chains

- Initialization of chains
  - start from different points
  - overdispersed starting points
  - different random number generator seeds
- Compare interesting scalars, eg:
  - parameters
  - future predictions
  - log-posterior density
  - log-predictive density

Slide 39

MCMC samples not independent

- Monte Carlo estimates still valid
- Monte Carlo error estimates slightly more difficult
  - time series analysis
  - thinning
  - batching
- Estimation of the effective number of samples
  - comparison of independent chains
  - time series analysis

Slide 40