Introduction and Markov chain Monte Carlo (MCMC)

MA8702: Advanced Computer Intensive Statistical Methods (Spring 2022)

Andrea Riebler

Logistics

- **Instructor:** Andrea Riebler, `andrea.riebler@ntnu.no`
- **Time:** Wednesdays 10:15-12:00,
  specified Mondays 10:15 - 11:00 (12:00). OK?????
- **Place:** Throughout January via Zoom,
  afterwards in Sentralbygg II, 734.
- **Textbook:** There is no textbook but lecture notes will be available on the class website. Supplementary papers will be discussed throughout the semester in Monday sessions.

Course outline

Prequesities: It is recommended to have already experience with computer-intensive statistical methods, for example the idea of sampling (Monte Carlo sampling), Markov chains, Bayesian inference etc., but also GLMs, multivariate normal distribution

The course is divided into three topics:

1. Markov chain Monte Carlo (MCMC) techniques
2. Gaussian Processes and Integrated nested Laplace approximations (INLA)
3. Sequential Monte Carlo methods

To each topic there is an obligatory project to be solved. In addition, two papers for each topic will be shortly presented and discussed in Monday sessions.

Grading

The grade for this course is pass/fail, and 70/100 score is required to pass (this is the standard rule for PhD courses at NTNU).

The projects need to be satisfactory in order to be admitted to the exam. In the exam one of the projects has to be presented and discussed.

Statistical software R

`R` is available for free download at The Comprehensive R Archive Network (http://cran.r-project.org) for Windows, Linux and Mac.

- Rstudio `http://www.rstudio.org` is an integrated development environment (system where you have your script, your run-window, overview of objects and a plotting window).
- A nice introduction to `R` is the book P. Dalgaard: Introductory statistics with R, 2nd edition, Springer which is also available freely to NTNU students as an ebook.

My computer-intensive life

- I have studied Bioinformatics and implemented my first MCMC algorithm during my master thesis in the field of Genetics.
- During my PhD in Biostatistics I have implemented various MCMC algorithms using Gibbs sampling, addition of auxiliary variables, Metropolis Hastings, etc. I have also started to use INLA.
- Today, I sometimes implement MCMC algorithms myself. INLA has become a good alternative for latent Gaussian models, and Stan is a very general software to do MCMC.

I have taught three times TMA4300 "Computer-intensive Statistical Methods" and TMA4265 "Stochastic Modelling" and have enjoyed it a lot. This is the second time that I teach this PhD course.

Covid-19 logistics

This course will be digitally at least until January, 24th. I would nevertheless very much encourage you to actively participate with quesions, comments, discussion.

It would be very nice if you could switch-on your cameras during our sessions. It is much nicer to talk to faces than to black squares.

It would be very nice if each of you could shortly introduce him/herself indicating background, study level, thesis project and supervisor, expectations into the course.

Quality ensurance: Reference group

- At least two participants
- The reference group should
  - have an ongoing dialogue with other participants throughout the semester.
  - meet three times throughout the semester with the lecturer.
  - Evaluate the course and write a report which will be published at the end.

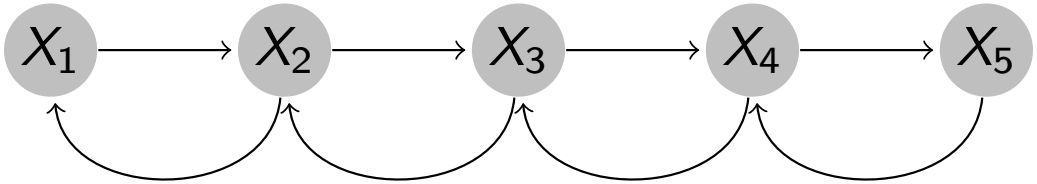
Introduction to Markov chain Monte Carlo (MCMC)

en.wikipedia.org/wiki/Markov_chain

Andrey Markov (1856 – 1922),

Russian mathematician.

Markov chain:



Given the previous observation X_{i-1}, X_i is independent of the sequence of events that preceded it.

Introduction to Markov chain Monte Carlo (MCMC)

- Usual Stochastic Processes Markov chains; a transition matrix P from which one can (sometimes) derive the (unique) stationary distribution π.
- MCMC, one would like to find, for a given distribution π, the transitions P which has π as limiting distribution.

This has become a very popular method for Monte Carlo sampling during the last $20 - 30$ years. (Hastings original paper was in 1970.)

Monte Carlo sampling

Suppose we want to approximate some expectation:

$$m = E[f(x)] = \sum_x f(x)\pi(x)$$

We assume it is relatively easy to simulate from the π distribution.

Monte Carlo algorithm:

- Sample x^b, $b = 1, \ldots, B$ from distribution π.
- Approximate by averaging:

$$\hat{m} = \frac{1}{B} \sum_{b=1}^{B} f(x^b)$$

Under weak regularity conditions, $\hat{m} \to m$, when $B \to \infty$.

Idea of Markov chain Monte Carlo

Idea

Simulate a Markov chain $X_1, \ldots, X_i, \ldots$, which is designed in a way such that $\mathrm{P}(X_i = x)$ converges to the target distribution $\pi(x)$, e.g. the posterior distribution.

Properties:

- After convergence, one obtains random samples from the target distribution, which can be used to estimate posterior characteristics.
- Samples will typically be dependent.

Central algorithms:

- Metropolis-Hasting algorithm
- Gibbs sampling

Review: Discrete-time Markov chains

A Markov chain is a discrete-time stochastic process $\{X_i\}_{i=0}^{\infty}$, $X_i \in S$, where given the present state, past and future states are independent (Markov assumption):

$$P(X_{i+1} = x_{i+1} \mid X_0 = x_0, X_1 = x_1, \ldots, X_i = x_i) = P(X_{i+1} = x_{i+1} \mid X_i = x_i)$$

Review: Markov chains

A Markov chain with stationary transition probabilities can be specified by:

- the initial distribution $P(X_0 = x_0) = g(x_0)$
- the transition matrix

$$P(x^\star \mid x) = P(X_{i+1} = x^\star \mid X_i = x) \quad [= P_{xx^\star}]$$

Review: Markov chains

A Markov chain has a unique limiting distribution $\pi(x)$ if the chain is irreducible, aperiodic, and positive recurrent. If so, the limiting distribution $\pi(x) = \lim_{i\to\infty} P(X_i = x)$ is given by

$$\begin{aligned} \pi(x^\star) &= \sum_{x\in S} \pi(x) P(x^\star \mid x) \quad \text{for all } x^\star \in S \\ \sum_{x\in S} \pi(x) &= 1 \end{aligned} \tag{1}$$

A sufficient condition for (1) is the detailed balance condition:

$$\pi(x)P(x^\star \mid x) = \pi(x^\star)P(x \mid x^\star) \quad \text{for all } x, x^\star \in S \tag{2}$$

which gives a time-reversible Markov chain.

Reversible Markov chains

- In a reversible MC we cannot distinguish the direction of simulation from inspecting a realisation of the chain (even if we know the transition matrix).
- Most MCMC algorithms are based on reversible Markov chains.

Problem statement

In stochastic processes course: The Markov chain is given, i.e. $P(x^\star \mid x)$ is given, find $\pi(x)$.

Now: $\pi(x)$, $x \in S$ is given, want to find $P(x^\star \mid x)$, $x, x^\star \in S$ so that

$$\pi(x^\star) = \sum_{x \in S} \pi(x) P(x^\star \mid x) \quad \text{for all } x^\star \in S$$

$$\sum_{x \in S} \pi(x) = 1$$

However, # unknowns: $|S| \cdot (|S| - 1)$; # equations: $|S|$.

$$\Rightarrow \text{many solutions exist – we want one!}$$

(Note: $|S|$ can be huge, so solving this as a matrix equation is not possible.)

Idea

Focus on (2) the detailed balance condition instead. We want to find $P(x^\star \mid x)$ that solves

$$\pi(x)P(x^\star \mid x) = \pi(x^\star)P(x \mid x^\star) \quad \text{for all } x, x^\star \in S$$

Here, we still have many solutions. However, we do not need a general solution, one (good) solution is enough.

Idea II

Try:

$$P(x^\star \mid x) = Q(x^\star \mid x)\alpha(x^\star \mid x), \quad x^\star \neq x$$

$$P(x \mid x) = 1 - \sum_{x^\star \neq x} Q(x^\star \mid x)\alpha(x^\star \mid x)$$

so that $\sum_{x^\star \in S} P(x^\star \mid x) = 1$.

Here:

- $Q(x^\star \mid x)$ is an almost arbitrary transition matrix (proposal kernel) for some other irreducible Markov chain.
- $\alpha(x^\star \mid x) \in [0, 1]$ is an acceptance probability.

Simulation algorithm (Metropolis-Hastings)

1: Init $x_0 \sim g(x_0)$
2: **for** $i = 1, 2, \ldots$ **do**
3: Generate a proposal $x^\star \sim Q(x^\star | x_{i-1})$
4: Compute the acceptance probability $\alpha(x^\star \mid x_{i-1})$
5: $u \sim \mathrm{U}(0, 1)$
6: **if** $u < \alpha(x^\star \mid x_{i-1})$ **then**
7: $x_i \leftarrow x^\star$
8: **else**
9: $x_i \leftarrow x_{i-1}$
10: **end if**
11: **end for**

What is Q and α?

See derivation

History of Metropolis-Hastings

- The algorithm was presented 1953 by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller from the Los Alamos group. It is named after the first author Nicholas Metropolis.
- W. Keith Hastings extended it to the more general case in 1970.
- It was then ignored for a long time.
- Since 1990 it has been used more intensively.

What about

- Irreducible: Must be checked in each case. Must choose $Q(x^\star \mid x)$ so that this is ok.
- Aperiodic: Sufficient that $P(x \mid x) > 0$ for one $x \in S$, so sufficient that $\alpha(x^\star \mid x) < 1$ for one pair $x^\star, x \in S$.
- Positive recurrent: for finite S, irreducibility is sufficient. More difficult in general, but if Markov chain is not recurrent we will see this as drift in the simulations. (In practice usually no problem).

Remarks on the Metropolis-Hastings algorithm

- Under some regularity conditions it can be shown that the Metropolis-Hasting algorithm converges to the target distribution regardless of the specific choice of $Q(x|x_{i-1})$.
- However, the speed of convergence and the dependence between the successive samples depends strongly on the proposal distribution.
- Since we only need to compute the ratio $\pi(x^\star)/\pi(x_{i-1})$, the proportionality constant is irrelevant.
- Similarly, we only care about $Q(.)$ up to a constant.
- Often it is advantageous to calculate the acceptance probability on log-scale, which makes the computations more stable.

Metropolis-Hastings algorithm

1: Init $x_0 \sim g(x_0)$
2: **for** $i = 1, 2, \ldots$ **do**
3: Generate a proposal $x^\star \sim Q(x^\star | x_{i-1})$
4: $u \sim \mathrm{U}(0, 1)$
5: **if** $u < \underbrace{\min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})} \times \underbrace{\frac{Q(x_{i-1}|x^\star)}{Q(x^\star|x_{i-1})}}_{\text{Proposal ratio}}\right)}_{\text{Acceptance probability } \alpha}$ **then**
6: $x_i \leftarrow x^\star$
7: **else**
8: $x_i \leftarrow x_{i-1}$
9: **end if**
10: **end for**

Special cases of the Metropolis-Hastings algorithm

Depending on the choice of $Q(x^\star|x_{i-1})$ different special cases result. In particular, two classes are important

- The independence proposal
- The Metropolis algorithm: random walk proposal
- Metropolis-adjusted Langevin algorithm (MALA): Langevin proposal

Independence proposal

- The proposal distribution does not depend on the current value x_{i-1}

$$Q(x|x_{i-1}) = Q(x).$$

- $Q(x)$ is an approximation to $\pi(x)$
  $\Rightarrow$ Acceptance rate should be close to 1.
- The sampler is closer to rejection sampler. However, here if we reject, then we retain the sample.

Experience:

- Performance is either very good or very bad, usually very bad.
- The tails of the proposal distribution should be at least as heavy as the tails of the target distribution.

The Metropolis algorithm

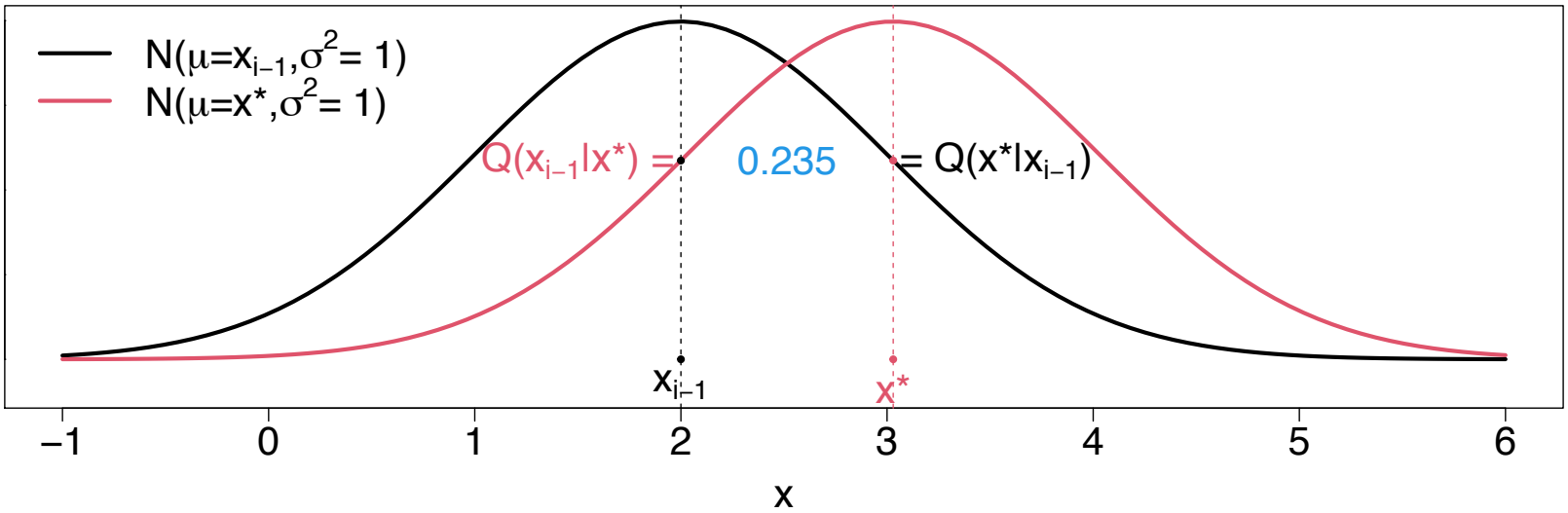
The proposal density is symmetric around the current value, that means

$$Q(x_{i-1}|x^\star) = Q(x^\star|x_{i-1}).$$

Hence,

$$\alpha = \min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})} \times \frac{Q(x_{i-1}|x^\star)}{Q(x^\star|x_{i-1})}\right) = \min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})}\right)$$

A particular case is the random walk proposal, defined as the current value x_{i-1} plus a random variate of a 0-centred symmetric distribution.



Examples for random walks proposal

Assume x is scalar.

Then all proposal kernels, which add a random variable generated from a zero-symmetrical distribution to the current value x_{i-1}, are random walk proposals. For example:

$$x^\star \sim \mathcal{N}(x_{i-1}, \sigma^2)$$

$$x^\star \sim t_\nu(x_{i-1}, \sigma^2)$$

$$x^\star \sim \mathcal{U}(x_{i-1} - d, x_{i-1} + d)$$

Efficiency of the Metropolis-Hastings algorithm

The efficiency and performance of the Metropolis-Hastings algorithm depends crucially on the relative frequency of acceptance.

An acceptance rate of one is not always good. Consider the random walk proposal:

- Too large acceptance rate $\Rightarrow$ Slow exploration of the target density.
- Too small acceptance rate $\Rightarrow$ Large moves are proposed, but rarely accepted.

Tuning the acceptance rate:

- For random walk proposals, acceptance rates between 20% and 50% are typically recommended. They can be achieved by changing the variance of the proposal distribution.
- For independence proposals a high acceptance rate is desired, which means that the proposal density is close to the target density.

Example: Random walk proposal

Exploration of a standard Gaussian distribution ($\mathcal{N}(0,1)$) using a random walk Metropolis algorithm. As proposal assume a Gaussian distribution with variance σ^2, where.

- $\sigma = 0.24$
- $\sigma = 2.4$
- $\sigma = 24$

See R-code `demo_mcmcRW.R`.