Data assimilation and Markov Chain Monte Carlo techniques

Moritz Schauer, *Gothenburg University, Sweden*

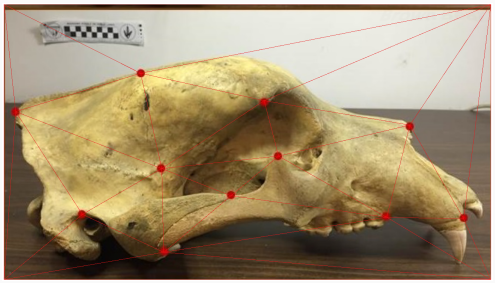
National Institute of Marine Science and Technology, 2019

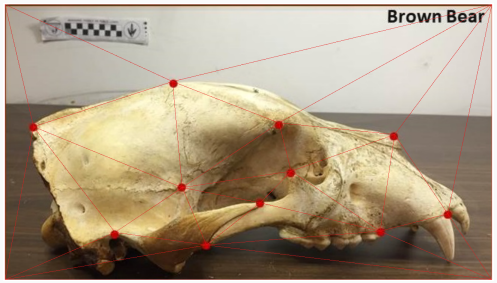
Introduction

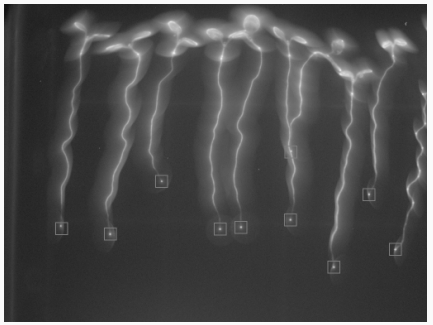
Outline

- Information about content and organisation of the course
- What is MCMC?
- Probability and Bayesian statistics refresher
- Getting you started with the lab

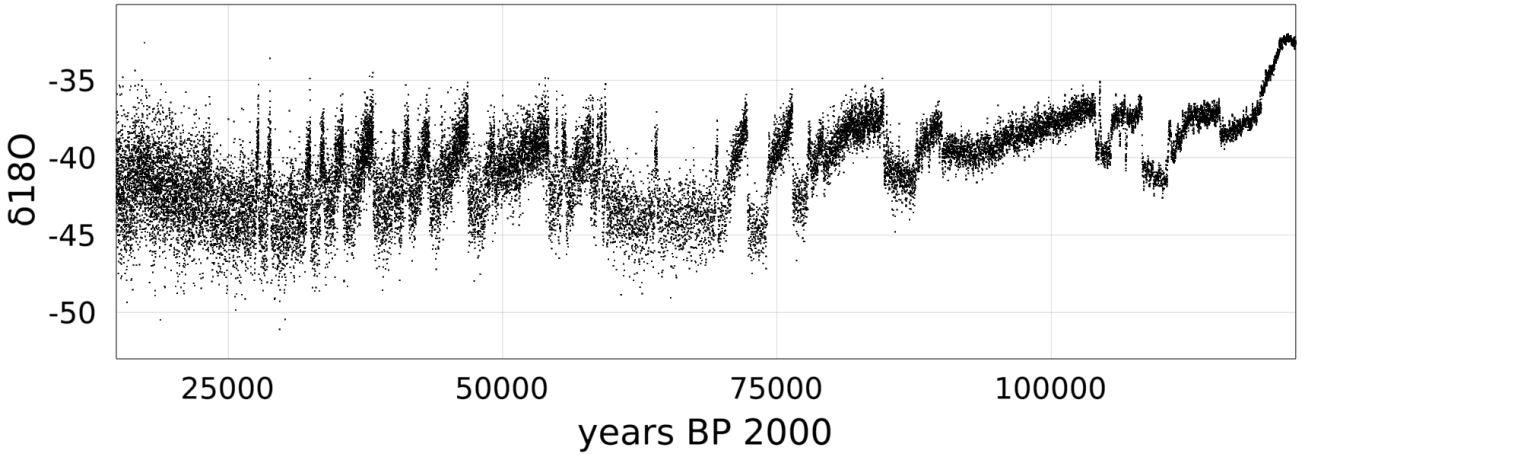
My interests







Statistical models for evolutionary change of plants / Or tracking the growth



Data on past temperature in climate science

Course content

1. Introduction
2. Probability and statistics refresher
3. Markov chain
4. Filtering
5. Hierarchical Bayesian modelling
6. Markov Chain Monte Carlo
7. Stochastic gradient descent
8. Automatic differentiation
9. Synthesis and modern stochastic simulation

Markov Chain Monte Carlo

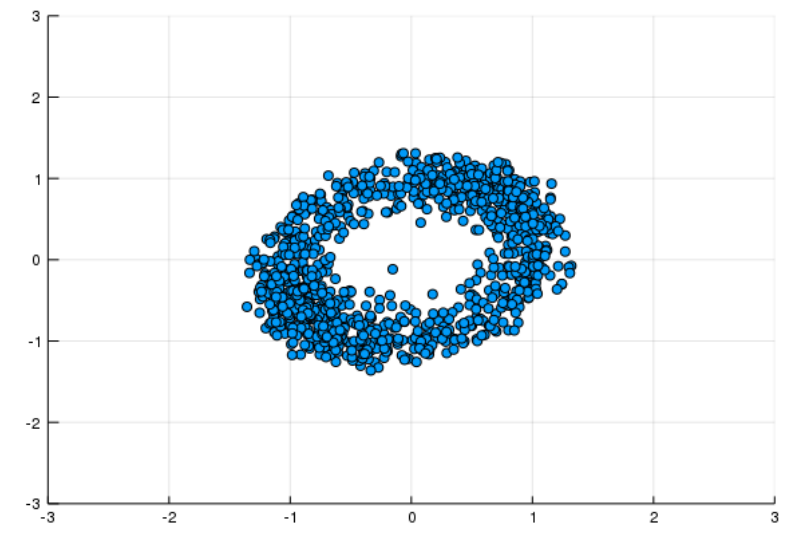
- Markov chain Monte Carlo methods. Computational algorithms for generating samples from a probability distribution say with density

$$x \mapsto p(x)$$

  for example a posterior density in Bayesian statistics.

- The Metropolis-Hastings algorithm, from which many MCMC methods can be derived.

Monte Carlo

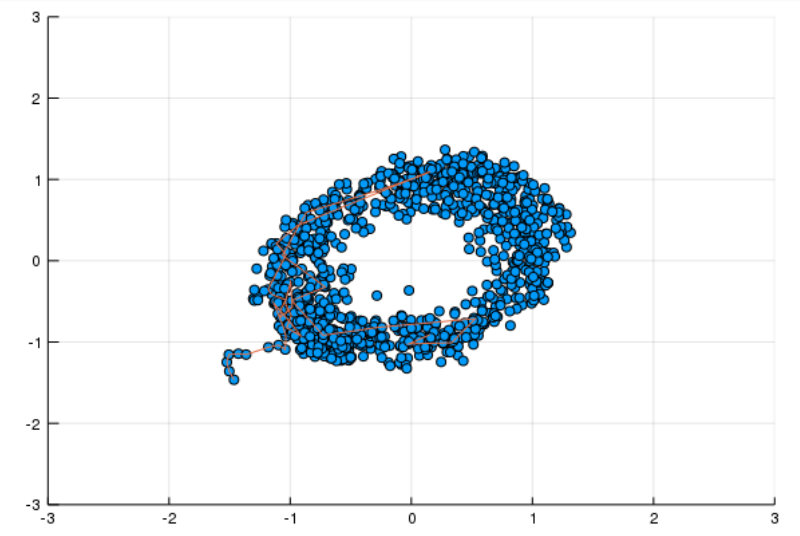


Draw independent samples $X_1, X_2, \ldots$ with probability density p.

By the law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \xrightarrow{a.s.} \int h(x)p(x)\mathrm{d}x.$$

Markov chain Monte Carlo



Generate Markov chain $X_1, X_2, \ldots$ with equilibrium density p.

Under the conditions of the *ergodic theorem* still

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \xrightarrow{a.s.} \int h(x)p(x)\mathrm{d}x,$$

but the convergence becomes slower.

Origin

Markov chain Monte Carlo (MCMC) was invented soon after ordinary Monte Carlo at Los Alamos, one of the few places where adequate computers were available at the time.

Metropolis et al. (1953) invented their algorithm to simulate a liquid in equilibrium with its gas phase.[1]

[1] Their article *Equation of State Calculations by Fast Computing Machines* was deemed important enough to have its own Wikipedia article.

Metropolis-Hastings

Hastings[2] generalized the work to the form now called the Metropolis-Hastings algorithm, the fundamental Markov chain Monte Carlo algorithm.

This algorithm has completely changed the way Bayesian inference is done and where it is used.

Now Markov chain Monte Carlo is the workhorse of Bayesian statistics.

[2] https://www.jstor.org/stable/2334940

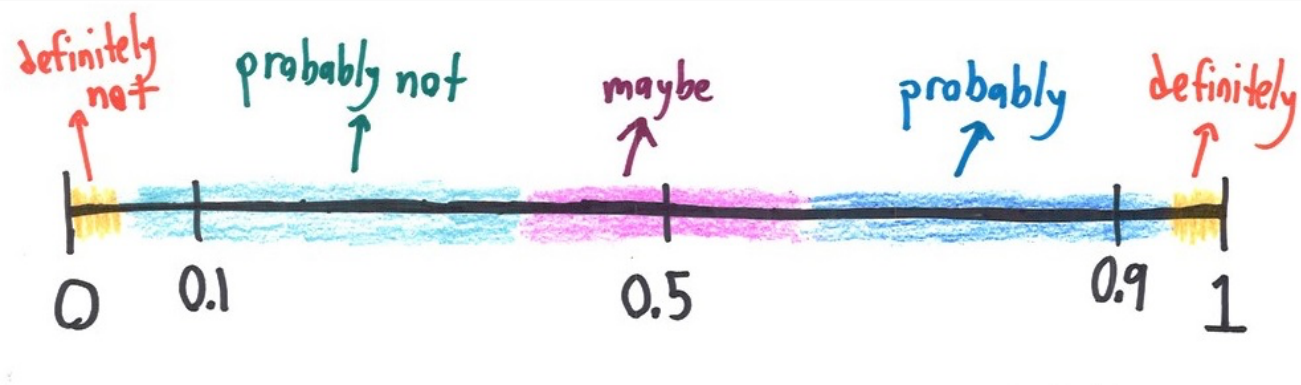
Probability and statistics refresher

Outline

- Probability, random variables
- Bayesian inference
- (Conditional) independence
- Transformations of random variables

Probability

- Probability is a numerical measure of how likely an event is to happen.



- Probability is a *proportion*, a number between 0 and 1.
- Notation: $\mathrm{P}(\textit{something that can happen}) = \textit{a probability}$. E.g. $\mathrm{P}(\text{coin heads-up}) = \frac{1}{2}$.

Figure from `https://mathwithbaddrawings.com/2015/09/23/what-does-probability-mean-in-your-profession/`.

Example: Discrete probability distributions

To specify a discrete probability distribution one lists all possible outcomes and the probabilities with which they occur.

The probability distribution for a fair six-sided die:

Event						
Probability	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

E.g. $P(\{$  ,  $\}) = \frac{1}{3}$.

Random variables

When dealing with several related random experiments, it is convenient to work with random variables.

Random variables are numeric quantities whose value depends on the outcome of a random event.

Random variables

Definition: Random variables are maps, defined on a probability space Ω. Think of Ω as a large set containing all information needed to determine the outcome of experiments, but not specified in detail.

Imagine nature picks a random state ω in the set Ω equipped with a probability measure $\mathbb{P}$.

Random variables

Definition: Random variables are maps, defined on a probability space Ω. Think of Ω as a large set containing all information needed to determine the outcome of experiments, but not specified in detail.

Imagine nature picks a random state ω in the set Ω equipped with a probability measure $\mathbb{P}$.

A random variable X maps ω to a quantity of interest, e.g. the number of eyes of a die shows, $\{1, 2, \ldots, 6\}$.

Then

$$P_X(A) := \mathbb{P}(\{\omega \colon X(\omega) \in A\})$$

defines a measure, the distribution of X (push forward).

Example: Just one die

A random variable X taking values ⚀ to ⚅. If we are taking about X, we do not know what is the result of the experiment, only that

$$P(X = ⚂) = \frac{1}{6}$$

Can you make this formal?

Coupled random variables

If several random variables are defined on the same underlying probability space, they are coupled.

In this case we can talk about dependence and independence etc.

Example?

Conditional distributions

Conditional distributions describe how probabilities of a random variable Y changes if $Y = y$ is known/observed (both coupled).

In the discrete case,

$$P(X = x \mid Y = y) = \frac{P(\{Y = y\} \cap \{X = x\})}{P(Y = y)}.$$

The probabilistic framework incorporates information (and missing information) naturally.

- (Bayesian) probabilistic updating

Models from random variables

Use random variables to model

- measurement errors
- statistical uncertainty
    - Bayesian inference
- naturally random processes (sic!)

Distinct notions, but reconcilable.[3]

[3] A very Dutch topic: frequentist analysis of Bayesian methods.

Examples

- Unknown mass of the Higgs boson
- Spam filter
- Decay of atoms
- The movement of molecules in a cubic meter of ocean

Random variables with densities

Random variable X with density (probability density function, pdf)

$$f_X : \mathbb{R} \to [0, \infty).$$

Probabilities are integrals

$$P(X \in [a, b]) = \int_{[a,b]} f_X(x) \mathrm{d}x.$$

Discrete random variables

Random variable X with probability mass function $p_X(x) = P(X = x)$,

$$p_X : \mathbb{R} \to [0, \infty)$$

where $p_X(x) = 0$ except for a finite (or countable) set of atoms D for which $p_X(x) > 0$, $x \in D$.

Probabilities are sums,

$$P(X \in A) = \sum_{x \in A \cap D} P(X = x).$$

Mixed random variables

Random variable X with (sub-) probability mass function with support D

$$p_d : \mathbb{R} \to [0, \infty),$$

(sub-) probability density function

$$f : \mathbb{R} \to [0, \infty)$$

Probabilities are computed as

$$P(X \in A) = \int_A f(x)\mathrm{d}x + \sum_{x \in A \cap D} P(X = x).$$

(with $\sum_{x \in D} P(X = x) + \int f(x)\mathrm{d}x = 1$.)

Example: mixed random variable

Let $Z \sim N(0,1)$. Set

$$X = \begin{cases} Z & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

(independent of Z).

What is... $P(X = 0) = ?$

$P(X \geq 0) = ?$

Example: mixed random variable

Let $Z \sim N(0,1)$. Set

$$X = \begin{cases} Z & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

(independent of Z).

What is... $P(X = 0) = 1 - p$

$P(X \geq 0) = (1-p) + p \int_0^\infty \phi(x)\mathrm{d}x = (1-p) + \frac{1}{2}p$

Definition of a probability measure

Start with a set of events $\mathcal{E}$ on a space Ω (σ-algebra.)

A probability measure is a map

$$P\colon \mathcal{E} \to [0,1]$$

with

$$P(\Omega) = 1$$

and

$$\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

for disjoint events $A_i \in \mathcal{E}$.

- Kolmogorov's axioms

Joint and marginal densities

Random variables X, Y with joint bivariate density

$$f_{X,Y} : \mathbb{R} \times \mathbb{R} \to [0, \infty)$$

Marginal densities

$$f_X(x) = \int f_{X,Y}(x, y) \mathrm{d}y, \quad f_Y(y) = \int f_{X,Y}(x, y) \mathrm{d}x$$

Joint and marginal densities

Example bivariate Normal with correlation ρ, means μ_X, μ_Y and variances σ_X^2, σ_Y^2.

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{z}{2(1-\rho^2)}\right),$$

$$z = \frac{(x-\mu_X)^2}{\sigma_X^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \frac{(y-\mu_Y)^2}{\sigma_Y^2}.$$

Conditional densities

The density of X given $Y = y$ (so the density of the conditional distribution of X after observation that Y took value y)

$$\begin{aligned} f_{X|Y=y}(x) &= \frac{f_{X,Y}(x,y)}{\int f_{X,Y}(x',y)\mathrm{d}x'} \\ &= \underbrace{C_y^{-1}}\ f_{X,Y}(x,y) \end{aligned}$$

($C_y = \int f_{X,Y}(x',y)\mathrm{d}x'$ is the normalising constant depending on y.)

Bayes formula

The Bayesian formalism equips unknown parameters with a *prior* distribution. Then the *posterior distribution* over the parameters given data is just the conditional distribution

$p(\theta)$ prior density of the parameter θ
$f(y \mid \theta)$ likelihood (density of observation y given θ)
$p(\theta \mid y)$ posterior parameter density θ

$$p(\theta \mid y) = \frac{f(y \mid \theta)p(\theta)}{\int f(y \mid \theta')p(\theta')\mathrm{d}\theta'}$$

Bayesian convention

Often the subscripts of the densities are dropped.

Random variable X density f_X. Random variable Y with density f_Y. X, Y have joint bivariate density $f_{X,Y}$.

Denote all densities by p and distinguish them by their arguments.

For example the formula for the marginal density becomes

$$p(x) = \int p(x, y) \mathrm{d}y.$$

Bayesian convention

Often the subscripts of the densities are dropped.

Random variable X density f_X. Random variable Y with density f_Y. X, Y have joint bivariate density $f_{X,Y}$.

Denote all densities by p and distinguish them by their arguments.

For example the formula for the marginal density becomes

$$p(x) = \int p(x, y) \mathrm{d}y.$$

The conditional density of X given $Y = y$ becomes

$$p(x \mid y) = \frac{p(x, y)}{\int p(x', y) \mathrm{d}x'}.$$

Radon-Nikodym theorem

For probability measures

$$P(A) = 0 \Rightarrow Q(A) = 0 \text{ for all } A$$

is equivalent to the existence of $\frac{\mathrm{d}Q}{\mathrm{d}P}$ with

$$Q(A) = \int_A \frac{\mathrm{d}Q}{\mathrm{d}P}(\omega)\mathrm{d}P(\omega) \text{ for all } A$$

Transformations of random variables

More tools in your toolbox for working with random variables:

Transformation by function h:

$$Y = h(X)$$

Transformations of random variables

More tools in your toolbox for working with random variables:

Transformation by function h:

$$Y = h(X)$$

Distribution: $P(Y \in A) = P(X \in h^{-1}(A))$.

Transformations of random variables

More tools in your toolbox for working with random variables:

Transformation by function h:

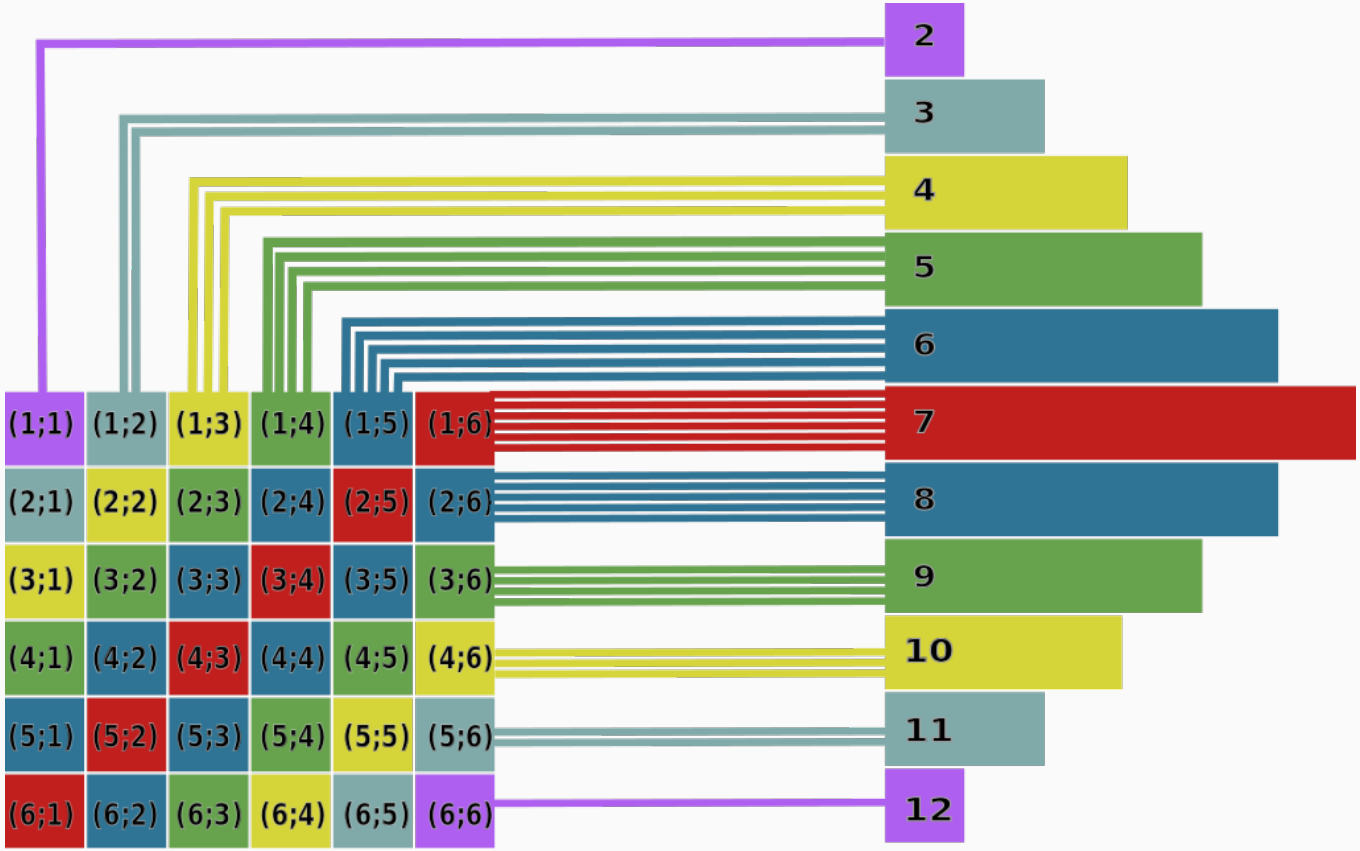
$$Y = h(X)$$

Distribution: $P(Y \in A) = P(X \in h^{-1}(A))$.

Simulation: Generate X, apply h.

Example: Sum of the eyes of two dice

For example, consider two independent throws of six sided dies.



Sums and products of random variables

In some cases the distribution of sums and products of random variables is known in closed form.

Examples: Sum of jointly normal random variables is normal. Ratio of independent centred normal X and root of chi-squared V is t-distributed.

Example: Differential equation with uncertain parameter

Example: Deterministic modelling + uncertainty.

Differential equation

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -\Theta u(t)$$

Example: Differential equation with uncertain parameter

Example: Deterministic modelling + uncertainty.

Differential equation

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -\Theta u(t)$$

with uncertainty about the parameter Θ and starting value $u(0) = U$, quantified by

$$U \sim \mathrm{N}(5, 3)$$
$$\Theta \sim \mathrm{N}(0.5, 0.1).$$

Example: Differential equation

Model is of functional form

$$u(t) = f_t(\Theta, U)$$

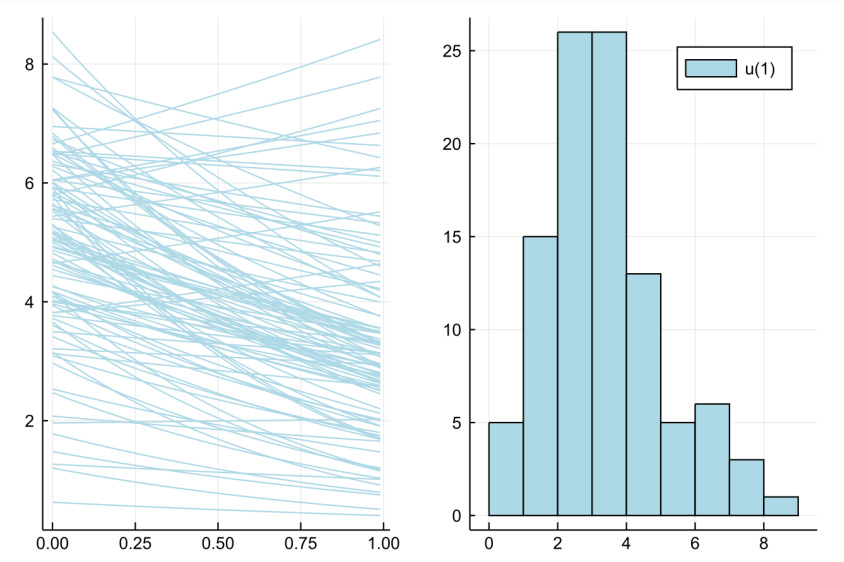
with $f_t(\theta, u) = \exp(-\theta t)u$ the solution of the ODE.

Example: Differential equation

Model is of functional form

$$u(t) = f_t(\Theta, U)$$

with $f_t(\theta, u) = \exp(-\theta t)u$ the solution of the ODE.



Explore model
through simulations

`https://nextjournal.com/mschauer/probabilistic-programming`

Exact computations

Example: Sum of uniform random variables

$$Z = X_1 + X_2, \quad \text{where } X_1, X_2 \sim \mathcal{U}[0,1] \text{ independent.}$$

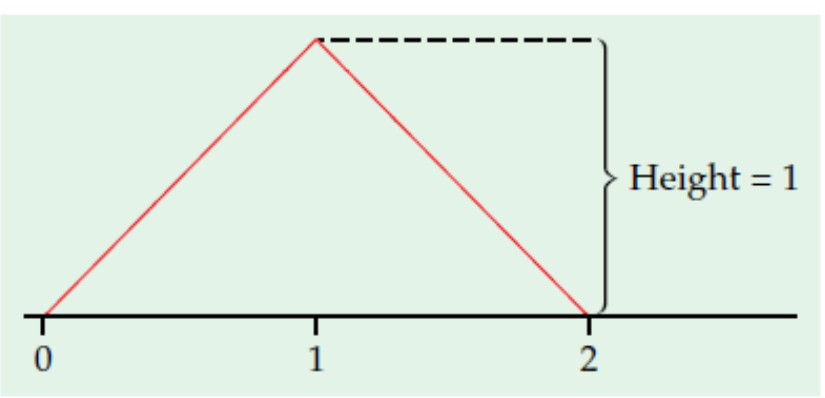
Exact computations

Example: Sum of uniform random variables

$$Z = X_1 + X_2, \quad \text{where } X_1, X_2 \sim \mathcal{U}[0,1] \text{ independent.}$$

Z has triangular distribution with density

$$f_Z(x) = \begin{cases} x & 0 \le x \le 1 \\ 2 - x & 1 \le x \le 2. \end{cases}$$



Working with mean and variance

Sometimes it is sufficient to determine mean and variance of the result of operations on random variables.

For sums of r.v. X and Y: $\mathrm{E}[X + Y] = \mathrm{E}X + \mathrm{E}Y$.

$\mathrm{Var}(aX + Y) = a^2 \mathrm{Var}(X) + \mathrm{Var}(Y)$ if X, Y independent

Working with mean and variance

Sometimes it is sufficient to determine mean and variance of the result of operations on random variables.

For sums of r.v. X and Y: $\mathrm{E}[X + Y] = \mathrm{E}X + \mathrm{E}Y$.

$\mathrm{Var}(aX + Y) = a^2 \mathrm{Var}(X) + \mathrm{Var}(Y)$ if X, Y independent

In the situation of the central limit theorem, a sum of random variables can approximated by a normal with the right mean and variance.

Error propagation

Independent $X_1 \dots X_n$ random with mean 10 and std. deviation $0.3 = \sqrt{0.09}$. $Z = \sum_{i=1}^{n} X_i$.

Z is approx. $\mathrm{N}(10n, 0.09n)$ distributed

$$Z = \underbrace{10n}_{mean} \pm \underbrace{0.3\sqrt{n}}_{std.dev}$$

Error propagation

Independent $X_1 \dots X_n$ random with mean 10 and std. deviation $0.3 = \sqrt{0.09}$. $Z = \sum_{i=1}^{n} X_i$.

Z is approx. $\mathrm{N}(10n, 0.09n)$ distributed

$$Z = \underbrace{10n}_{mean} \pm \underbrace{0.3\sqrt{n}}_{std.dev}$$

Very different from how the engineer argues:

$X_1, \dots, X_n$ are quantities approximately in the interval $[9.7, 10.3]$

$$Z \in [9.7n, 10.3n] \quad \text{or} \quad Z = 10n \pm 0.3n.$$

Julia programming language

I have prepared some things using the programming language Julia.

You can follow the course and lab using a programming language you are familar with, but you might enjoy giving Julia a try.

There are many online ressources at `julialang.org/learning/`.

Setup

- Install Julia. Download the Julia 1.1 for your operating system from `julialang.org`

or

- Make a free account on `https://nextjournal.com/`. Nextjournal is an online notebook environment running Julia (you need a working internet connection to use it.)
- In any case, download `https://www.dropbox.com/s/yvwgz71zgvkj7sg/icecoredata.csv` and read `https://nextjournal.com/mschauer/bayesian-filter` for the lab.

Markov chain

Simple Markov chain

Recursively, let

$$X_i = h(X_{i-1}, Z_i), \quad i > 0$$

and $X_0 \sim g$

using independent identically distributed innovation or noise random variables

$$Z_1, \ldots, Z_n, \ldots.$$

Example: Autoregressive chain

Autoregressive chain: Take $\eta \in (0, 1)$,

$$h(x, z) = \eta x + z$$

and

$$Z_1, \ldots, Z_n, \overset{i.i.d.}{\sim} N(0, 1).$$

So

$$X_i = \eta X_{i-1} + Z_i, \quad i > 0.$$

Example: Autoregressive chain

Then

$$\mathrm{Var}(X_i) = \mathrm{Var}(\eta X_{i-1} + Z_i) = \eta^2 \,\mathrm{Var}(X_{i-1}) + 1.$$

Example: Autoregressive chain

Then

$$\mathrm{Var}(X_i) = \mathrm{Var}(\eta X_{i-1} + Z_i) = \eta^2 \,\mathrm{Var}(X_{i-1}) + 1.$$

Assume that X_0 is normal with mean zero. Then all X_i are equally normal with mean zero.

Setting $\mathrm{Var}(X_i) \equiv \sigma^2$

Example: Autoregressive chain

Then

$$\mathrm{Var}(X_i) = \mathrm{Var}(\eta X_{i-1} + Z_i) = \eta^2 \,\mathrm{Var}(X_{i-1}) + 1.$$

Assume that X_0 is normal with mean zero. Then all X_i are equally normal with mean zero.

Setting $\mathrm{Var}(X_i) \equiv \sigma^2$ and solving

$$\sigma^2 = \eta^2 \sigma^2 + 1$$

gives

$$\sigma^2 = \frac{1}{1-\eta^2}.$$

$\{X_1, X_2, \dots\}$ is a Markov chain with stationary distribution

$$X_i \sim \mathrm{N}(0, \tfrac{1}{1-\eta^2}).$$

Filtering

Smoothing/Filtering

Start with joint density factorized as

$$f(x, y) = f(y \mid x) f(x)$$

This is convenient if e.g. x is a parameter with prior and the density of Y given $X = x$ is determined by the model.

Y is observed.

Smoothing/Filtering

Start with joint density factorized as

$$f(x, y) = f(y \mid x) f(x)$$

This is convenient if e.g. x is a parameter with prior and the density of Y given $X = x$ is determined by the model.

Y is observed.

Then as before

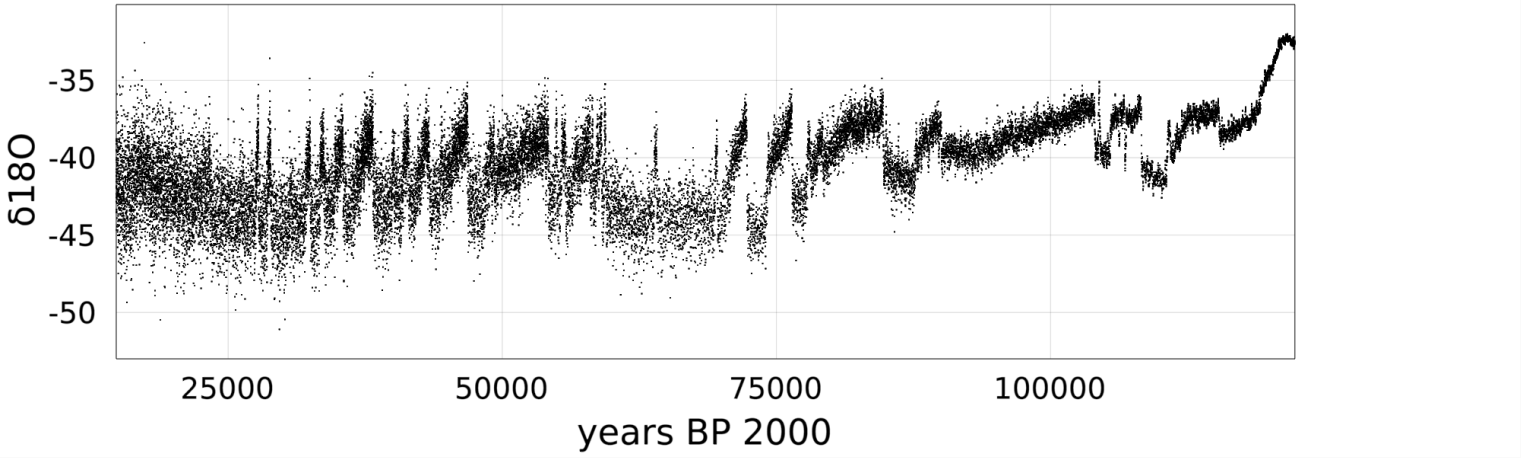
$$f(x \mid y) = \frac{f(y \mid x) f(x)}{\int f(y \mid x') f(x') \mathrm{d}x'} = \frac{f(x, y)}{\int f(x', y) \mathrm{d}x'}.$$

Random walk filter example for the lab

See `https://nextjournal.com/mschauer/bayesian-filter` for example and details.

Random walk filter example for the lab

The North Greenland Ice Core Project obtained a core sample of ice from drilling through the arctic ice shield. Scientists are interested in a quantity X_t (the past $\delta^{18}O$ values) which changes over time, written as $t = 1, \ldots, n$.



Measurements

They have produced a sequence of n measurements

$$Y_1, \ldots, Y_n$$

and, as the measurement errors stem from a variety of unrelated causes, model the measurement errors ϵ_t at time t,

$$Y_t = X_t + \epsilon_t.$$

as independent random variables ϵ_t with Gaussian distribution $N(0, \sigma^2)$ with variance σ^2 for each $t = 1, \ldots, n$.

Model

Model

$$X_{t+1} = X_t + \eta_t,$$

for $t = 1, \ldots, n-1$, where η_t are independent normally distributed $N(0, s^2)$ with mean zero and variance s^2 for $t = 1, \ldots, n$.

Filter

The *filtered* distribution of X_t is the conditional distribution of X_t at time t given observations $Y_1 = y_1, Y_2 = y_2$ up to and including $Y_t = y_t$. It captures what we know about X_t if we have seen the measurement points $y_1, y_2, \ldots, y_t$.

Kalman filtering equations

For a random walk model with independent observation errors as above the filtered distribution of X_t is a normal distribution and its mean μ_t and variance p_t^2 can be computed recursively using the filtering equations

$$p_t^2 = \sigma^2 K_t$$
$$\mu_t = \mu_{t-1} + K_t(y_t - \mu_{t-1})$$

with K_t the so called Kalman gain

$$K_t = \frac{s^2 + p_{t-1}^2}{s^2 + p_{t-1}^2 + \sigma^2}, \quad t > 1$$

and $K_1 = \frac{p_0^2}{p_0^2 + \sigma^2}$.

Kalman filtering equations

The mean of the filtered distribution, μ_t, serves as estimate of the unknown location of X_t and the standard deviation p_t can be seen as measure of uncertainty about the location.

Full Kalman filter

The more general Kalman filter for multivariate linear models

$$x_k = \Phi x_{k1} + b + w_k, \qquad w_k \sim N(0, Q)$$
$$y_k = H x_k + v_k, \qquad v_k \sim N(0, R)$$

proceeds similarly.

Hierarchical Bayesian modelling

Outline

- Hierarchical probabilistic model
- Bayesian inference
- Monte Carlo methods
    - Metropolis–Hastings

Hierarchical models

Use hierarchical models to deal with

- dependency between variables
- indirect observations and missing data
    - smoothing/filtering
- Generally: If the data generating process is somewhat understood

A dilemma when modelling groups

Pool all groups together.

- Ignore latent differences.
- Observations are not independent.

Model each group separately.

- Small sample sizes problematic.
- Many more parameters to estimate.
- Ignores latent similarity.

Hierarchical modelling shows a way out...

Example: Group means model

Latent group means

$$b_i \sim \mathrm{N}(\mu, \sigma_b^2)$$

with density ϕ_i for m groups.

Specimen j from group i

$$y_{ij} \sim \mathrm{N}(b_i, \sigma_y^2)$$

with density $\phi_{ij}(\cdot - b_i)$.

Factorisation of the joint density

$$\prod_{i \in I} \phi_i(b_i) \prod_{j \in J_i} \phi_{ij}(y_{ij} - b_i)$$

Example: Group means model

Latent group means

$$b_i \sim \mathrm{N}(\mu, \sigma_b^2)$$

with density ϕ_i for m groups.

Specimen j from group i

$$y_{ij} \sim \mathrm{N}(b_i, \sigma_y^2)$$

with density $\phi_{ij}(\cdot - b_i)$.

Factorisation of the joint density

$$\prod_{i \in I} \phi_i(b_i) \prod_{j \in J_i} \phi_{ij}(y_{ij} - b_i)$$

Only the specimens are observed, the conditional density is

$$p_{(y_{ij})}(b_1, \ldots, b_m) = \int \prod_{i \in I} \phi_i(b_i) \prod_{j \in J_i} \phi_{ij}(y_{ij} - b_i) \mathrm{d}b_1 \cdots \mathrm{d}b_m$$

Bayesian net

This is an example of a Bayesian net. The variables with nodes together with the parent operator pa form a DAG, a directed, acyclic graph.

In general, the joint density in a Bayesian net can be factorises in the form

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid \mathrm{pa}(x_i))$$

Bayesian net, factor graph representation

$$p(x_1, \ldots, x_d) = \prod_{\alpha \in F} f_\alpha(x_\alpha)$$

where $\alpha \subset \{1, \ldots, d\}$ and x_α is the tuple of variables $(x_i)_{i \in \alpha}$.

Bayes' formula for hierarchical models

Markov Chain Monte Carlo

What is on the menu

- Markov chain Monte Carlo methods. Computational algorithms for generating samples from a probability distribution say with density

$$x \mapsto p(x)$$

  for example a posterior density in Bayesian statistics.

- The Metropolis-Hastings algorithm, from which many MCMC methods can be derived.

Recap: Markov chain

A Markov chain

- is a sequence of random variables X_1, X_2, ... (with values in some state space E)
- with the Markov property:

$$P(X_i \in B \mid X_{i-1} = x_{i-1}, \ldots, X_1 = x_1) = P(X_i \in B \mid X_{i-1} = x_{i-1}).$$

Recap: Markov chain

A Markov chain

- is a sequence of random variables X_1, X_2, ... (with values in some state space E)
- with the Markov property:

$$P(X_i \in B \mid X_{i-1} = x_{i-1}, \ldots, X_1 = x_1) = P(X_i \in B \mid X_{i-1} = x_{i-1}).$$

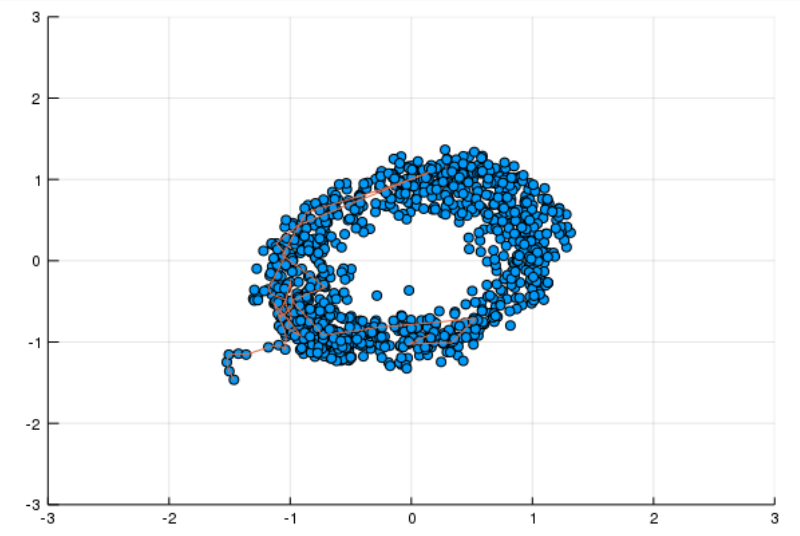
The transition kernel

$$Q(x; B) = P(X_i \in B \mid X_{i-1} = x)$$

of a time-homogenous Markov chain fully describes the evolution of the chain.

Denote its density in x' by $q(x; x')$.

Markov chain Monte Carlo



Generate Markov chain $X_1, X_2, \ldots$ with equilibrium density p.

Under the conditions of the ergodic theorem still

$$\frac{1}{n}\sum_{i=1}^{n} f(X_i) \xrightarrow{a.s.} \int f(x)p(x)\mathrm{d}x,$$

but the convergence becomes slower.

Bayesian inference

A typical application of MCMC is in Bayesian inference.

Posterior density $\pi(\cdot \mid y)$ is proportional to

$$p_\theta(y)\pi(\theta).$$

(y is an observation with density $p_\theta(y)$ and π is a prior distribution of a parameter θ)

Statistical questions should have answers of the form

$$\int f(\theta)\mathrm{d}\pi(\theta \mid y).$$

Expectations

Statistical questions should have answers of the form

$$\int f(\theta) \mathrm{d}\pi(\theta \mid y).$$

For example posterior mean and variance or probabilities arise this way.

"Statistical" questions which cannot be posed as expectations of the posterior are very problematic.

What would be interesting choices of f?

Expectations

Statistical questions should have answers of the form

$$\int f(\theta) \mathrm{d}\pi(\theta \mid y).$$

For example posterior mean and variance or probabilities arise this way.

"Statistical" questions which cannot be posed as expectations of the posterior are very problematic.

What would be interesting choices of f?

Probabilities are expectations of indicators and the median is minimiser of expected deviation.

Probabilistic programming

Hierarchical Bayesian model

$$
\begin{aligned}
s^2 &\sim \text{InverseGamma}(2, 3) \\
m &\sim \mathrm{N}(0, s^2) \\
x &\sim \mathrm{N}(m, s^2) \\
y &\sim \mathrm{N}(m, s^2)
\end{aligned}
$$

What is the posterior distribution of parameters s and m given observations $x = 1.5$, $y = 2$?

Probabilistic programming

```julia
using Turing
using StatsPlots

# Define a simple Normal model with unknown mean and variance.
@model gdemo(x, y) = begin
  s ~ InverseGamma(2.0, 3.0)
  m ~ Normal(0.0, sqrt(s))
  x ~ Normal(m, sqrt(s))
  y ~ Normal(m, sqrt(s))
end

#  Run sampler, collect results
chn = sample(gdemo(1.5, 2), HMC(1000, 0.1, 5));
```

✓ 3.2s Julia Julia 1.1

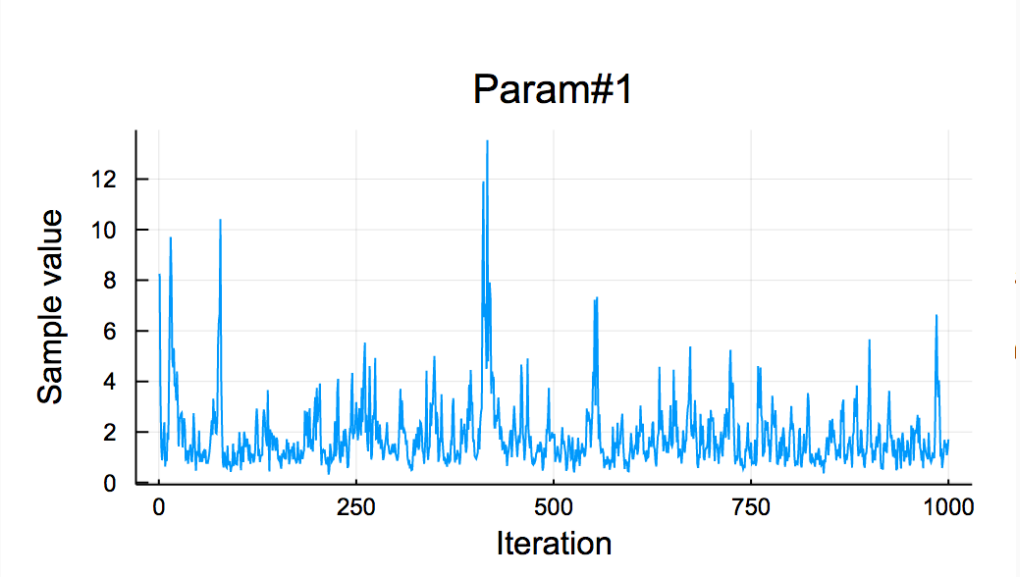
Short illustration using the *Turing.jl* package in julia.

`https://nextjournal.com/mschauer/probabilistic-programming`

Probabilistic programming

```
plot(Chains(chn[:s]))
```

✓ 1.0s Julia Julia 1.1

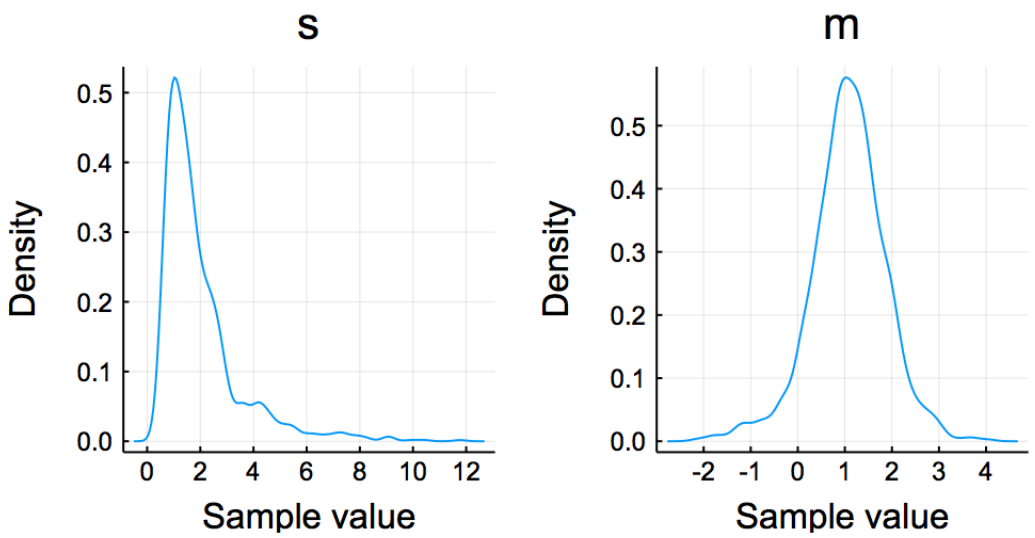


Generated Markov chain.

Probabilistic programming

```
plot(plot(density(Chains(chn[:s])), title="s"),
     plot(density(Chains(chn[:m])), title="m"))
```

✓ 1.0s Julia Julia 1.1



Posterior density estimates.

Markov chains and detailed balance

The chain is in detailed balance if there is a density p such that

$$p(x)q(x;x') = p(x')q(x';x),$$

i.e. symmetry of the joint distribution of two iterates X, X' when starting the chain in $X \sim p$.

Implies that p is stationary density

$$\begin{aligned} p(x) &= \int p(x)q(x;x')\mathrm{d}x' \\ &= \underbrace{\int p(x')q(x';x)\mathrm{d}x'}_{\text{the density after one step}}. \end{aligned}$$

(A sufficient condition.)

Metropolis-Hastings algorithm

Starting point is a Markov chain with some transition density

$$q(x; x').$$

For example, a symmetric random walk,

$$q(x; x') = C \exp(-\frac{\|x - x'\|^2}{2\sigma_{mh}^2}) = q(x'; x).$$

How to obtain a chain with stationary density p

Define a new chain, for which detailed balance holds for p by modifying the transition kernel.

Metropolis-Hastings algorithm

Define the *Metropolis-Hastings acceptance probability*

$$\alpha(x; x') = \min\left(\frac{p(x')q(x'; x)}{p(x)q(x; x')}, 1\right).$$

Metropolis-Hastings algorithm

Define the *Metropolis-Hastings acceptance probability*

$$\alpha(x; x') = \min\left(\frac{p(x')q(x'; x)}{p(x)q(x; x')}, 1\right).$$

Starting from X_1, we recursively...

- Given X_i, draw random proposal X_{i+1}° from the density $q(X_i; \cdot)$
- With probability $\alpha(X_i; X_{i+1}^{\circ})$

$$\text{accept and move: } X_{i+1} = X_{i+1}^{\circ},$$

else

$$\text{reject and stay: } X_{i+1} = X_i.$$

Detailed balance

Proposition: This creates a Markov chain with a new Markov kernel $Q^{MH}(x, \cdot)$ such that detailed balance holds for p.

Detailed balance

Proposition: This creates a Markov chain with a new Markov kernel $Q^{MH}(x, \cdot)$ such that detailed balance holds for p.

That means hat the joint law $p(x)Q^{MH}(x, \mathrm{d}x')\mathrm{d}x$ is symmetric:

$$p(x)Q^{MH}(x; \mathrm{d}x')\mathrm{d}x = p(x')Q^{MH}(x'; \mathrm{d}x)\mathrm{d}x'.$$

Detailed balance

Proposition: This creates a Markov chain with a new Markov kernel $Q^{MH}(x, \cdot)$ such that detailed balance holds for p.

That means hat the joint law $p(x)Q^{MH}(x, \mathrm{d}x')\mathrm{d}x$ is symmetric:

$$\int_A \int_B p(x)Q^{MH}(x; \mathrm{d}x')\mathrm{d}x = \int_B \int_A p(x')Q^{MH}(x'; \mathrm{d}x)\mathrm{d}x' \quad \forall A, B$$

Proof:

$$\alpha(x; x') = \min\left(\frac{p(x')q(x'; x)}{p(x)q(x; x')}, 1\right)$$

For any x, x', at least one of $\alpha(x; x'), \alpha(x'; x)$ equals 1.

Proof:

To show symmetry take $x \neq x'$ with $\alpha(x; x') < 1$ (otherwise switch x, x' below.) Then

$$\begin{aligned} p(x)Q^{MH}(x, \mathrm{d}x')\mathrm{d}x &= p(x)q(x; x')\alpha(x; x')\mathrm{d}x\mathrm{d}x' \\ &= p(x)q(x; x')\frac{p(x')q(x'; x)}{p(x)q(x; x')}\mathrm{d}x\mathrm{d}x' \end{aligned}$$

Proof:

To show symmetry take $x \neq x'$ with $\alpha(x; x') < 1$ (otherwise switch x, x' below.) Then

$$\begin{aligned} p(x)Q^{MH}(x, \mathrm{d}x')\mathrm{d}x &= p(x)q(x; x')\alpha(x; x')\mathrm{d}x\mathrm{d}x' \\ &= p(x)q(x; x')\frac{p(x')q(x'; x)}{p(x)q(x; x')}\mathrm{d}x\mathrm{d}x' \\ &= p(x')q(x'; x) \cdot \underbrace{1}_{\alpha(x';x)} \mathrm{d}x\mathrm{d}x' \\ &= p(x')Q^{MH}(x', \mathrm{d}x)\mathrm{d}x' \end{aligned}$$

Therefore detailed balance holds and p is stationary density of the chain.

Conditional sampling / Gibbs sampling

Coordinatewise MH: To sample from $p(x_1, x_2, \ldots, x_d)$ repeat for each coordinate

- Propose new value x_i' for x_i from transition density $q_i(x_1, \ldots, x_d; \cdot)$ and accept or reject with and accept with

$$\alpha_i = \frac{p(x_1, \ldots, x_i', \ldots, x_d) q_i(x_1, \ldots, x_i', \ldots, x_d; x_i)}{p(x_1, \ldots, x_i, \ldots, x_d) q_i(x_1, \ldots, x_i, \ldots, x_d; x_i')}.$$

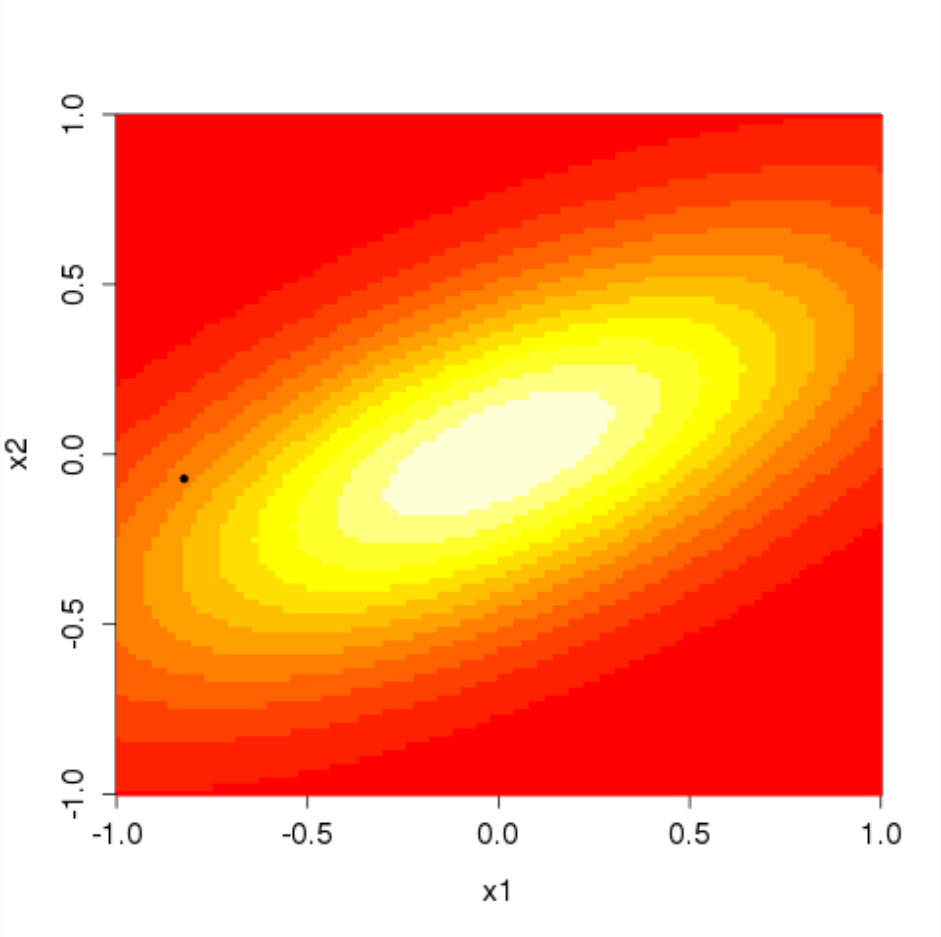
Conditional sampling / Gibbs sampling

Special case: Gibbs sampler, where one samples from the conditional densities

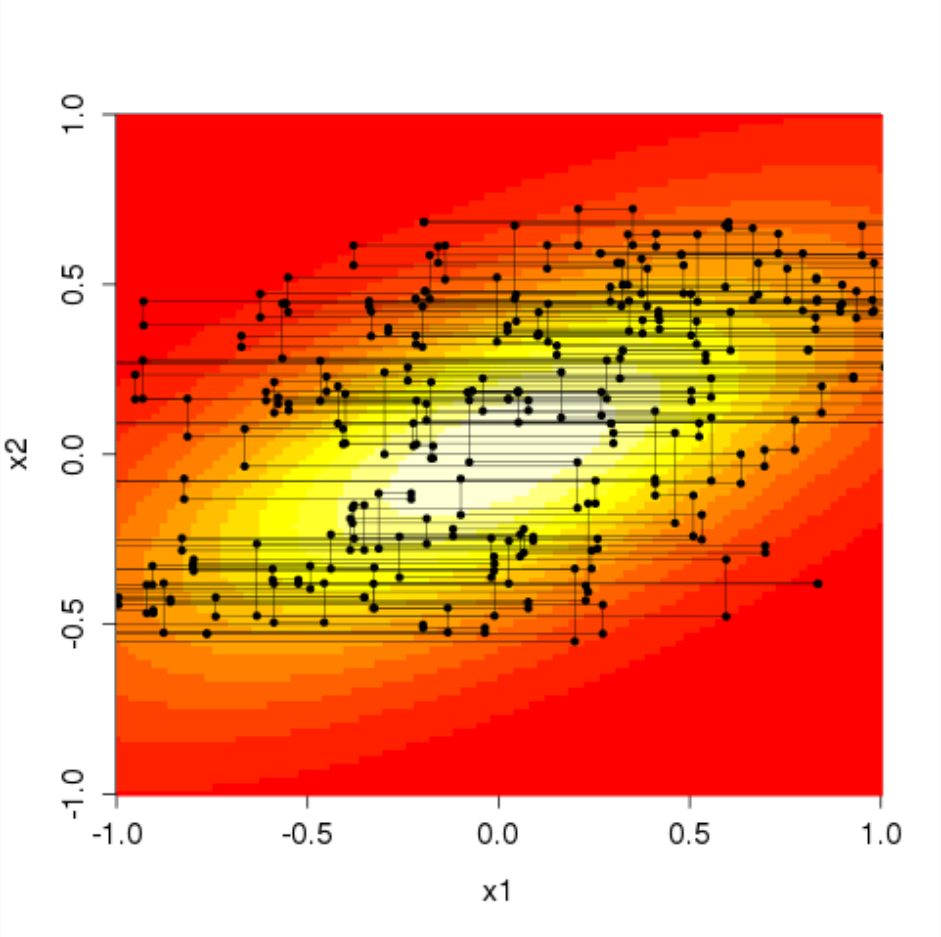
$$q_i(x_1, \ldots, x_d; x_i) = p(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d),$$

respective. Then all proposals are accepted ($\alpha_i = 1$).

Exploring a bivariate Gaussian



Exploring a bivariate Gaussian



Data augmentation

A typical situation: There is a natural Bayesian model for the internal state of a system

$$X \sim \pi(x, \theta) = f_\theta(x)\pi(\theta).$$

But the state is only observed indirectly and with error

$$Y = h(X) + \epsilon$$

with $h\colon \mathbb{R}^m \to \mathbb{R}^n$, $n < m$ and observation error ϵ. Denote the density of Y given $X = x$ by f_x.

Use Gibbs or coordinatewise MH with coordinates θ and x to sample from

$$\pi(x, \theta \mid Y = y) \propto f_x(y) f_\theta(x) \pi(\theta).$$

Data augmentation

Iteratively:

Imputation: Propose new value for x

$$x' \sim q_x(x, \theta; \cdot)$$

and accept with

$$\alpha_x = \frac{\pi(x', \theta \mid Y = y) q_x(x', \theta; x)}{\pi(x, \theta \mid Y = y) q_x(x, \theta; x')}.$$

Inference: Propose new value for θ

$$\theta' \sim q_\theta(x, \theta; \cdot)$$

and accept with

$$\alpha_\theta = \frac{\pi(x, \theta' \mid Y = y) q_\theta(x, \theta'; \theta)}{\pi(x, \theta \mid Y = y) q_\theta(x, \theta; \theta')}.$$

Earlier example

Hierarchical Bayesian model

$$
\begin{aligned}
s^2 &\sim \text{InverseGamma}(2, 3) \\
m &\sim \text{N}(0, s^2) \\
x &\sim \text{N}(m, s^2) \\
y &\sim \text{N}(m, s^2)
\end{aligned}
$$

Stochastic gradient descent

Outline

- Gradient descent
- Automatic differentiation / back propagation
- Stochastic gradient descent
- Applications in estimation, regression, minimising a loss function

Derivatives (notation and recap)

Multiple chain rule

$$\frac{\partial}{\partial x} z(w(v(x)) = \frac{\partial z}{\partial w} \frac{\partial w}{\partial v} \frac{\partial v}{\partial x}$$

Derivatives (notation and recap)

Multiple chain rule

$$\frac{\partial}{\partial x} z(w(v(x)) = \frac{\partial z}{\partial w} \frac{\partial w}{\partial v} \frac{\partial v}{\partial x}$$

Total derivative

$$\frac{\partial}{\partial x} z(w_1(x), \ldots, w_k(x)) = \frac{\partial z}{\partial w_1} \frac{\partial w_1}{\partial x} + \cdots + \frac{\partial z}{\partial w_k} \frac{\partial w_k}{\partial x}$$

Gradients and derivatives

Differentiable function

$$F \colon \mathbb{R}^d \to \mathbb{R}.$$

The gradient is the vector of derivatives

$$\nabla F(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} F(x) \\ \dots \\ \frac{\partial}{\partial x_d} F(x) \end{pmatrix}.$$

Gradients and derivatives

Differentiable function

$$F \colon \mathbb{R}^d \to \mathbb{R}.$$

The gradient is the vector of derivatives

$$\nabla F(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} F(x) \\ \dots \\ \frac{\partial}{\partial x_d} F(x) \end{pmatrix}.$$

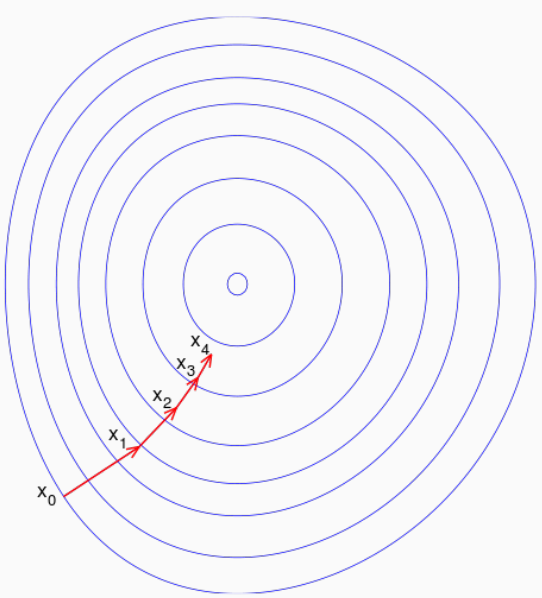
The vector

$$-\nabla F(x)$$

points in the direction of steepest descent.

Gradient descent

Objective: Get close to a (local) minimum $x^\star$ of $F(x)$.



$x_{i+1} = x_i - \gamma \nabla F(x_i)$.
Move in direction of steepest descent.

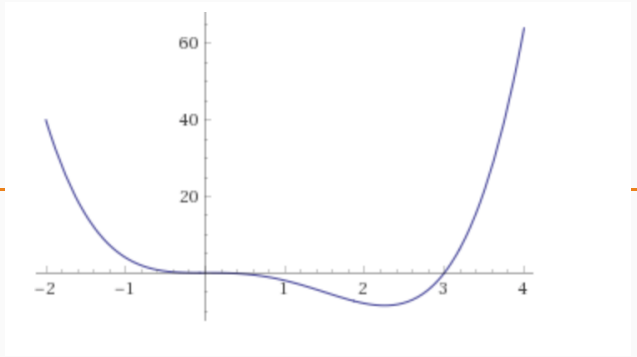
γ is called the learning rate.

Learning rate

γ too small: moving slowly with small steps.

γ too large: overshooting.

Julia example



$f(x) = x^4 - 3x^3 + 2$,
with derivative $f'(x) = 4x^3 - 9x^2$

```
x = 6.0
γ = 0.01
precision = 0.00001

∇f(x) = 4x^3 - 9x^2

while true
    global x
    step = γ * ∇f(x)
    if abs(step) < precision
        break
    end
    x = x - step
end

println("The local minimum occurs at $x")
```

Result: $x = 2.24996$. True optimum: $2.25 = \frac{9}{4}$.

Automatic differentiation

Finite differences

$$f'(x) \approx \frac{f(x + he_i) - f(x)}{h}, \quad \text{h small.}$$

This is inefficient - it requires d evaluations of $F : \mathbb{R}^d \to \mathbb{R}$ to compute the gradient

$$\nabla F(x).$$

Finite differences

$$f'(x) \approx \frac{f(x + he_i) - f(x)}{h}, \quad \text{h small.}$$

This is inefficient - it requires d evaluations of $F : \mathbb{R}^d \to \mathbb{R}$ to compute the gradient

$$\nabla F(x).$$

It is also only approximate and h must be balanced between floating-point precision and mathematical precision.

Automatic differentiation

Task: Compute gradient of

$$z = x_1 x_2 + \sin(x_1)$$

in $x_1 = 2, x_3 = 3$.

Automatic differentiation

Task: Compute gradient of

$$z = x_1 x_2 + \sin(x_1)$$

in $x_1 = 2, x_3 = 3$.

Static single assignment form (SSA):

$$x_1 = 2$$
$$x_2 = 3$$
$$w_1 = x_1 x_2$$
$$w_2 = \sin(x_1)$$
$$z = w_1 + w_2$$

SSA form in Julia code

```
julia> f(x1,x2) = x1*x2 + sin(x1)
f (generic function with 1 method)

julia> @code_lowered f(2.0,3.0)
CodeInfo(
1 1 ─ %1 = x1 * x2
  │   %2 = (Main.sin)(x1)
  │   %3 = %1 + %2
  └──      return %3
)
```

Back propagation

Gradient of $z = x_1 x_2 + \sin(x_1)$ in $x_1 = 2$,$x_3 = 3$.

Static single assignment form (SSA):

$$w_1 = x_1 x_2 = 3 \cdot 2$$

$$w_2 = \sin(x_1) = \sin(2)$$

$$z = w_1 + w_2$$

Back propagation

Gradient of $z = x_1 x_2 + \sin(x_1)$ in $x_1 = 2$,$x_3 = 3$.

Static single assignment form (SSA):

$$w_1 = x_1 x_2 = 3 \cdot 2 \qquad \frac{\partial w_1}{\partial x_1} = 3, \frac{\partial w_1}{\partial x_2} = 2$$

$$w_2 = \sin(x_1) = \sin(2) \qquad \frac{\partial w_2}{\partial x_1} = \cos(2)$$

$$z = w_1 + w_2 \qquad \frac{\partial z}{\partial w_1} = 1, \frac{\partial z}{\partial w_2} = 1$$

Back propagation

Gradient of $z = x_1 x_2 + \sin(x_1)$ in $x_1 = 2, x_3 = 3$.

Static single assignment form (SSA):

$$w_1 = x_1 x_2 = 3 \cdot 2 \qquad \frac{\partial w_1}{\partial x_1} = 3, \frac{\partial w_1}{\partial x_2} = 2$$

$$w_2 = \sin(x_1) = \sin(2) \qquad \frac{\partial w_2}{\partial x_1} = \cos(2)$$

$$z = w_1 + w_2 \qquad \frac{\partial z}{\partial w_1} = 1, \frac{\partial z}{\partial w_2} = 1$$

Gradient: $$\frac{\partial z}{\partial x_1} = \frac{\partial z}{\partial w_1}\frac{\partial w_1}{\partial x_1} + \frac{\partial z}{\partial w_2}\frac{\partial w_2}{\partial x_1} = 3 + \cos(2)$$

(Combine all paths from x_1 to z.)

Back propagation

Gradient of $z = x_1 x_2 + \sin(x_1)$ in $x_1 = 2, x_3 = 3$.

Static single assignment form (SSA):

$$w_1 = x_1 x_2 = 3 \cdot 2 \qquad \frac{\partial w_1}{\partial x_1} = 3, \frac{\partial w_1}{\partial x_2} = 2$$

$$w_2 = \sin(x_1) = \sin(2) \qquad \frac{\partial w_2}{\partial x_1} = \cos(2)$$

$$z = w_1 + w_2 \qquad \frac{\partial z}{\partial w_1} = 1, \frac{\partial z}{\partial w_2} = 1$$

Gradient: $$\frac{\partial z}{\partial x_1} = \frac{\partial z}{\partial w_1}\frac{\partial w_1}{\partial x_1} + \frac{\partial z}{\partial w_2}\frac{\partial w_2}{\partial x_1} = 3 + \cos(2)$$

(Combine all paths from x_1 to z.)

Similar for $\frac{\partial z}{\partial x_2}$.

Automatic derivatives in Julia

```
julia> using Zygote
```

Automatic derivatives in Julia

```
julia> using Zygote
```

```
julia> f(x1,x2) = x1*x2 + sin(x1)
f (generic function with 2 methods)

julia> gradient(f, 2, 3)
(2.583853163452857, 2)
```

Automatic derivatives are exact:

```
julia> gradient(f, 2, 3) .- (3.0 + cos(2.0), 2.0)
(0.0, 0.0)
```

Maximum likelihood method

If $X_1, \ldots, X_n$ are independent observations from density $p_\theta(x)$ with parameter $\theta \in \mathbb{R}^d$, then the negative log-likelihood is

$$F(\theta) = -\ell(\theta; x_1, \ldots, x_n) = -\log \prod_{i=1}^{n} p_\theta(x_i) = -\sum_{i=1}^{n} \log p_\theta(x_i)$$

and

$$\hat{\theta} = \operatorname*{argmin}_\theta F(\theta)$$

is the maximum likelihood estimator (assuming a unique global minimum).

Maximum likelihood method

If $X_1, \ldots, X_n$ are independent observations from density $p_\theta(x)$ with parameter $\theta \in \mathbb{R}^d$, then the negative log-likelihood is

$$F(\theta) = -\ell(\theta; x_1, \ldots, x_n) = -\log \prod_{i=1}^{n} p_\theta(x_i) = -\sum_{i=1}^{n} \log p_\theta(x_i)$$

and

$$\hat{\theta} = \operatorname*{argmin}_\theta F(\theta)$$

is the maximum likelihood estimator (assuming a unique global minimum). Estimators that arise as minimisers of sums are called M-estimators.

Regression with quadratic loss

If $(y_1, x_1), \ldots, (y_n, x_n)$ are data points,

$$\hat{\theta} = \operatorname*{argmin}_{\theta} F_n(\theta) = \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(\theta; y_i, x_i)$$

with quadratic loss

$$L(\theta; y, x) = \|y - f(x; \theta)\|_2^2$$

e.g. with $\theta = (m, b)$ and

$$f(x; \theta) = mx + b.$$

Regularised maximised likelihood

If $x_1, \ldots, x_n$ are i.i.d from density p_θ, $\theta \in \mathbb{R}^d$ a parameter, target

$$\hat{\theta} = \operatorname*{argmin}_\theta F_n(\theta) = \operatorname*{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} L(\theta; x_i)$$

with

$$L(\theta; x_i) = -\log p(y_i, x_i; \theta) + \lambda \|\theta\|_2^2$$

Estimate the gradient

Empirical loss over the data:

$$F_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i)$$

Estimate the gradient

Empirical loss over the data:

$$F_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i)$$

Better objective - minimise expected loss:

$$F(\theta) = \mathrm{E}_X[L(\theta, X)]$$

Use gradient descent and law of large numbers (∇ linear...)

$$\nabla F(\theta) = \mathrm{E}_X[\nabla L(\theta, X)] = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i)$$

Estimate the gradient

What if we estimate gradient with just one sample? (In practise: just pick a random one)

Estimate the gradient

What if we estimate gradient with just one sample? (In practise: just pick a random one)

Unbiased estimate of gradient:

$$\mathrm{E}[\nabla L(\theta, X_i)] = \nabla F(\theta)$$

Noisy, but useful!

Stochastic gradient descent

In the maximum log likelihood example the objective is of the form

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

Often n is very large.

Stochastic gradient descent moves along the gradient of a random f_i:

$$x_{j+1} = x_j - \gamma \nabla f_m(x_j) \quad m \sim U(\{1, \ldots, n\})$$

Note that

$$\mathrm{E}[\nabla f_m(x)] = \nabla F(x).$$

"Moves on average in the right direction."

Batch gradient descent

$$x_{j+1} = x_j - \gamma \frac{1}{K} \sum_{k=1}^{K} \nabla f_{m_k}(x_j) \quad m_1, \ldots, m_K \sim U(\{1, \ldots, n\})$$

Again

$$\mathrm{E}[\frac{1}{K} \sum_{k=1}^{K} \nabla f_{m_k}(x_j)] = \nabla F(x).$$

Less noisy, but also more expensive to compute...

Online stochastic gradient descent

Example: Maximum likelihood estimation again.

Infinite stream of i.i.d. samples $X_1, \ldots, X_j, \ldots$ from $p_\theta(x)$ with parameter $\theta \in \mathbb{R}^d$,

$$\theta_{j+1} = \theta_j + \gamma \nabla \log p_{\theta_j}(X_j).$$

Doesn't require to hold all data in memory.

AD software

Julia: `http://www.juliadiff.org`, Zygote.jl and others

Matlab: `http://www.autodiff.org`

Python: `https://github.com/HIPS/autograd`

AD software

Julia: `http://www.juliadiff.org`, Zygote.jl and others

Matlab: `http://www.autodiff.org`

Python: `https://github.com/HIPS/autograd`

R? Possibly `http://tobiasmadsen.com/2017/03/31/automatic_differentiation_in_r/`

Synthesis and modern stochastic simulation