6.867 Fall 2017
Sampling

MCMC, Gibbs, MH

Lecture 19/24: 16th Nov, 2017

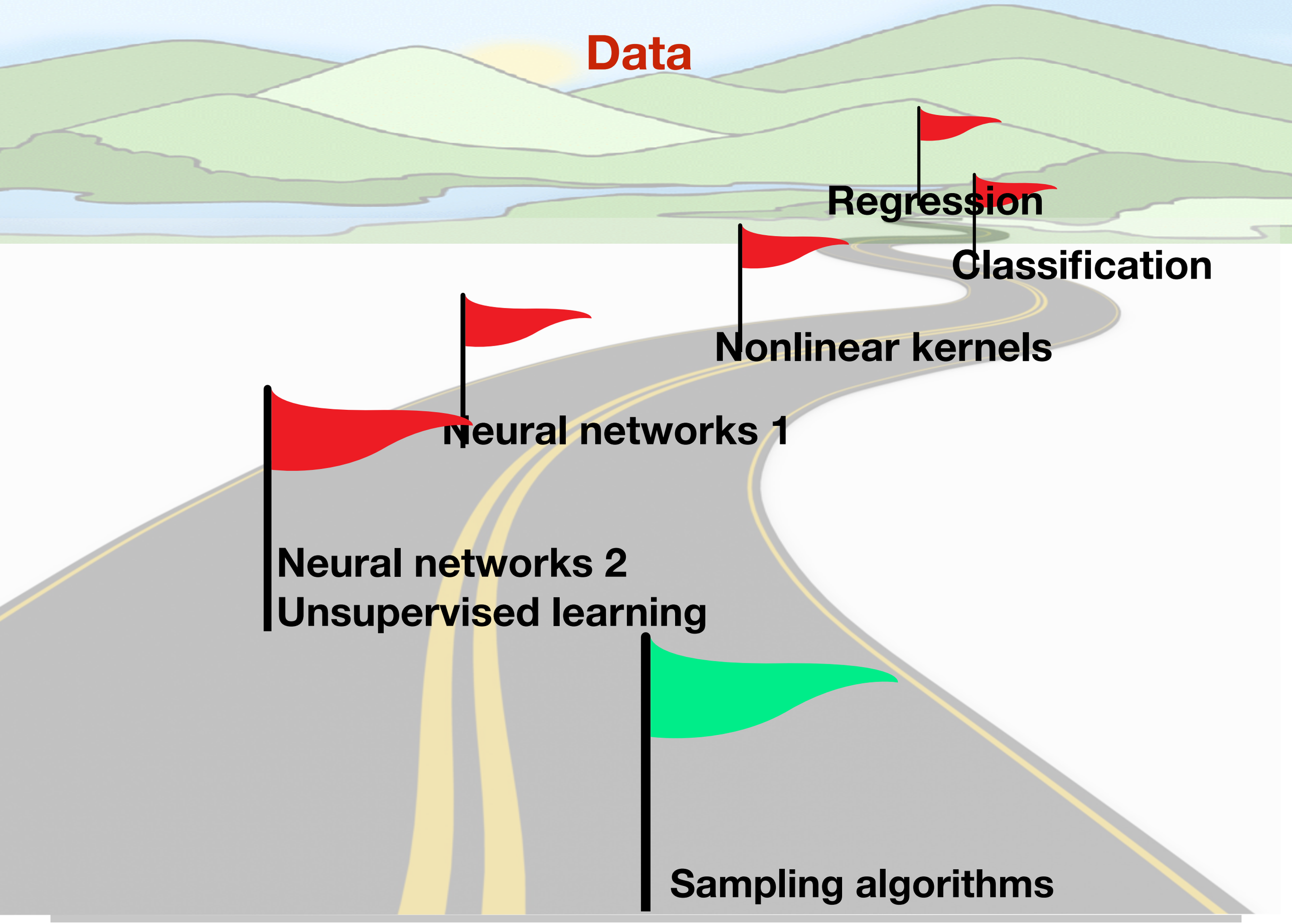
Admin

HW3 Reviews due 11/19

Project Milestone 11/21

Exam 2: covers material up to today (11/16)

Holding office hrs: 11/20, 3pm-4pm, 32-D580
(additional timings will be shared via stellar)


Data
Regression
Classification
Nonlinear kernels
Neural networks 1
Neural networks 2
Unsupervised learning
Sampling algorithms

Outline

- Why care about sampling?
- Basic concepts
- Gibbs sampling (MCMC)
- Metropolis Hastings (MCMC)
- Some theory

Why?

For most probabilistic models, exact *inference is intractable*

i.e., computing P(z|x,ϕ) is hard

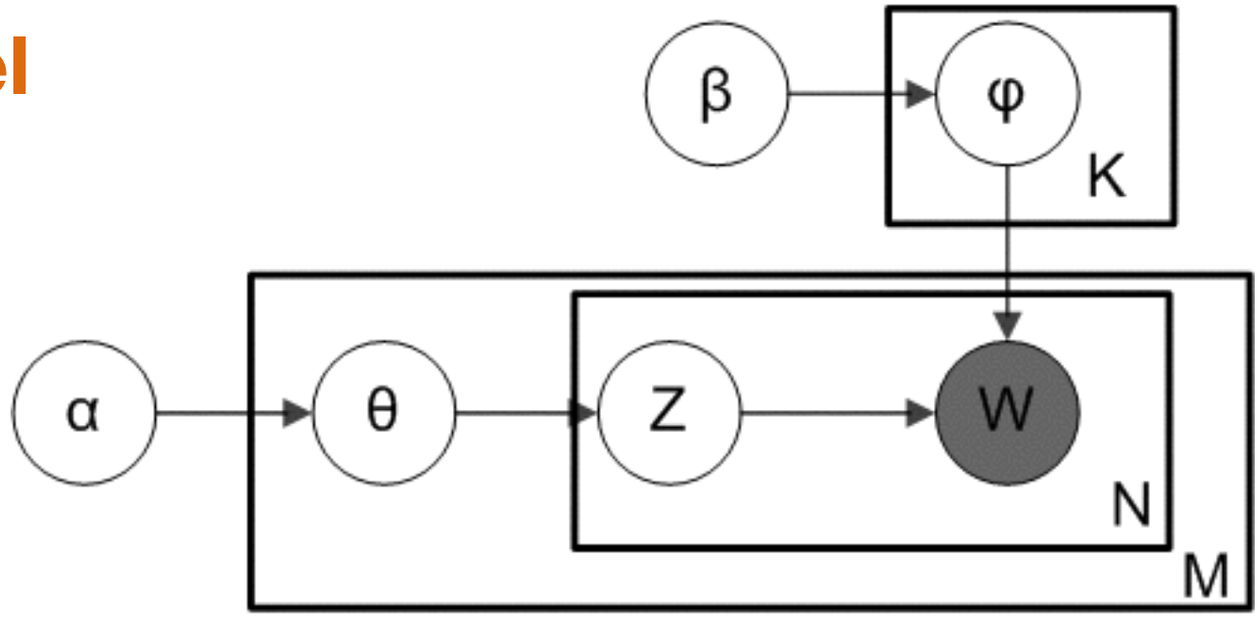
$$P(z|x,\phi) = \frac{P(x,\phi|z)\pi(z)}{P(x,\phi)}$$

Hard to compute integral

- **Inference**: $P(z|Data) \propto P(z,Data)P(z);\ P(Data)=\int P(Data,z)dz$
- **Prediction**: $P(x^*|Data)= \int P(x^*|z,Data)P(z|Data)dz = E[P(x^*|z,Data)]$
- **Model selection**: To compare models M_1 and M_2, need $P(M|Data)$

Example: Inference in LDA

LDA model



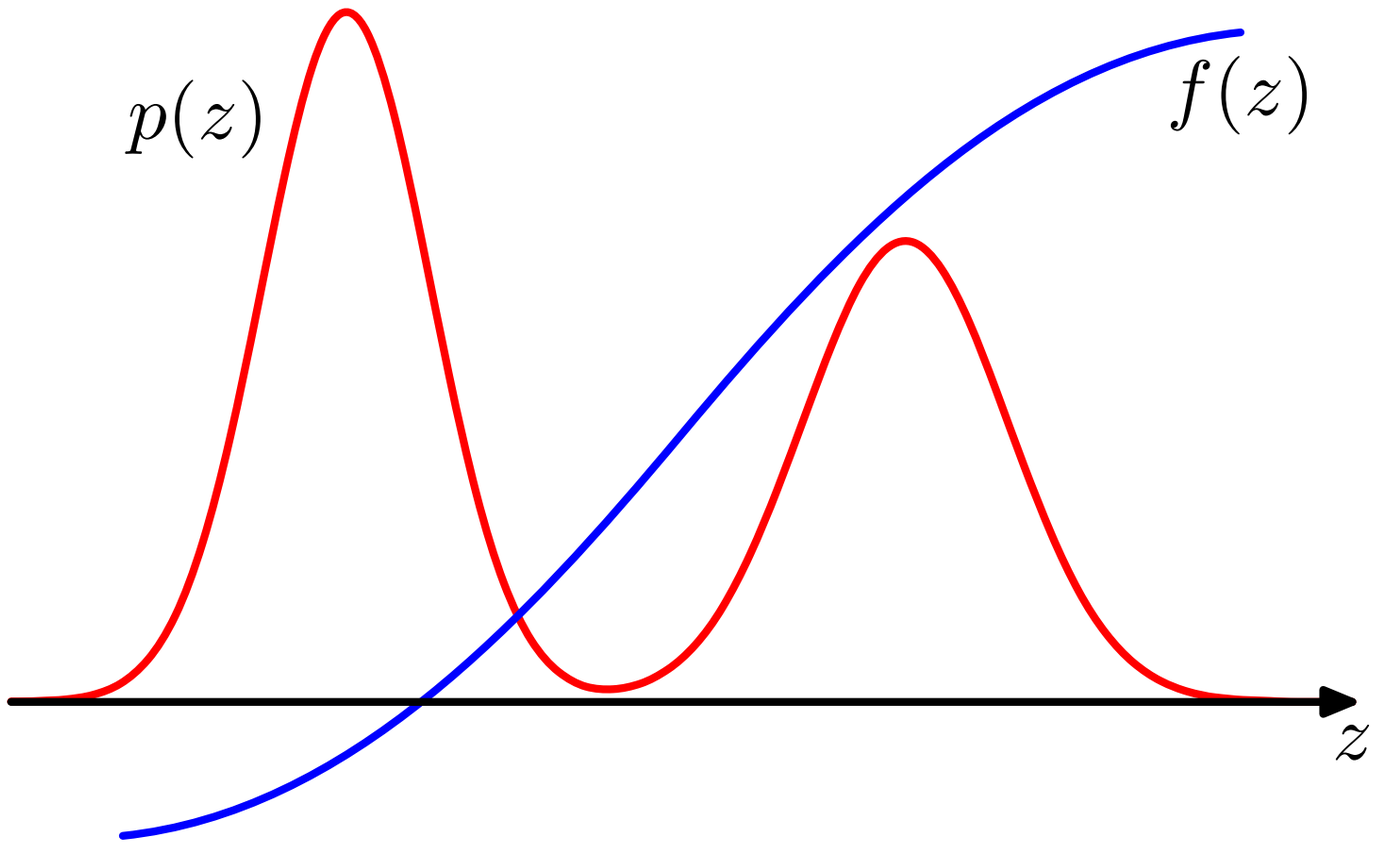
$$P(\boldsymbol{Z}, \boldsymbol{W}; \alpha, \beta) = \prod_{j=1}^{M} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^{i} + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^{i} + \alpha_i\right)} \times \prod_{i=1}^{K} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \frac{\prod_{r=1}^{V} \Gamma(n_{(\cdot),r}^{i} + \beta_r)}{\Gamma\left(\sum_{r=1}^{V} n_{(\cdot),r}^{i} + \beta_r\right)}.$$

Goal: approximate $P(Z|W, \alpha, \beta)$

Challenge: Use Bayes Rule, but how to normalize?

LDA model: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

Estimating E[f] can be hard



Difficulty of estimating E[f] = $\int f(z)p(z)dz$

Difficulty exacerbated in higher dimensions

Goals

Problem 1: Generate samples from P(z|…)

Problem 2: Compute E[f(z)] under P(z|…)

Notation: I'll just use $P(z)$ to denote $P(z \mid whatever\ else)$

Key idea: When in doubt, dilemma, difficulty, randomize!

Solving Problem 2

We focus on obtaining samples $\{z_1, z_2, \ldots, z_n\}$ from P(z)

Suppose we have i.i.d. samples $\{z_1, z_2, \ldots, z_n\} \sim P(z)$, then

$$\frac{1}{n}\sum_{i=1}^{n} f(z_i) \approx \int f(z)P(z)dz$$

and

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i)\right] = \mathbb{E}_P\left[f(z)\right]$$

The variance of above empirical estimator shrinks as σ^2/n

where σ^2 is the variance of f(z), i.e., Var[f(z)]

Solving Problem 2

The accuracy of the estimate does not depend on the dimensionality of 'z'; it just depends on σ^2/n.

So as the number of samples grows, the estimate becomes more precise. If σ^2 is small, a small number of samples will suffice to estimate the integral E[f(z)]

So have we solved high-dimensional integration? 😮

…and the answer is

$$P(z) = \frac{1}{Z_p}\tilde{P}(z)$$

Nope!

(how to normalize? Z_p typically intractable)

Generating samples from P(z)

three ways to “sample”

- Instead of P(z), use a proxy Q(z), called “proposal”
  Generate samples from Q(z), decide if they are ‘good enough’
  If **yes**, then keep, **else** reject — aka “rejection sampling”

[see Bishop 11.1.2 for more]

Say we just want to compute E[f(z)]

- Generate samples according to a proposal Q(z)
  Don’t throw (reject), but ‘weight’ samples by “importance”
  This lead to “importance sampling”

[Exercises 10; Bishop 11.1.4]

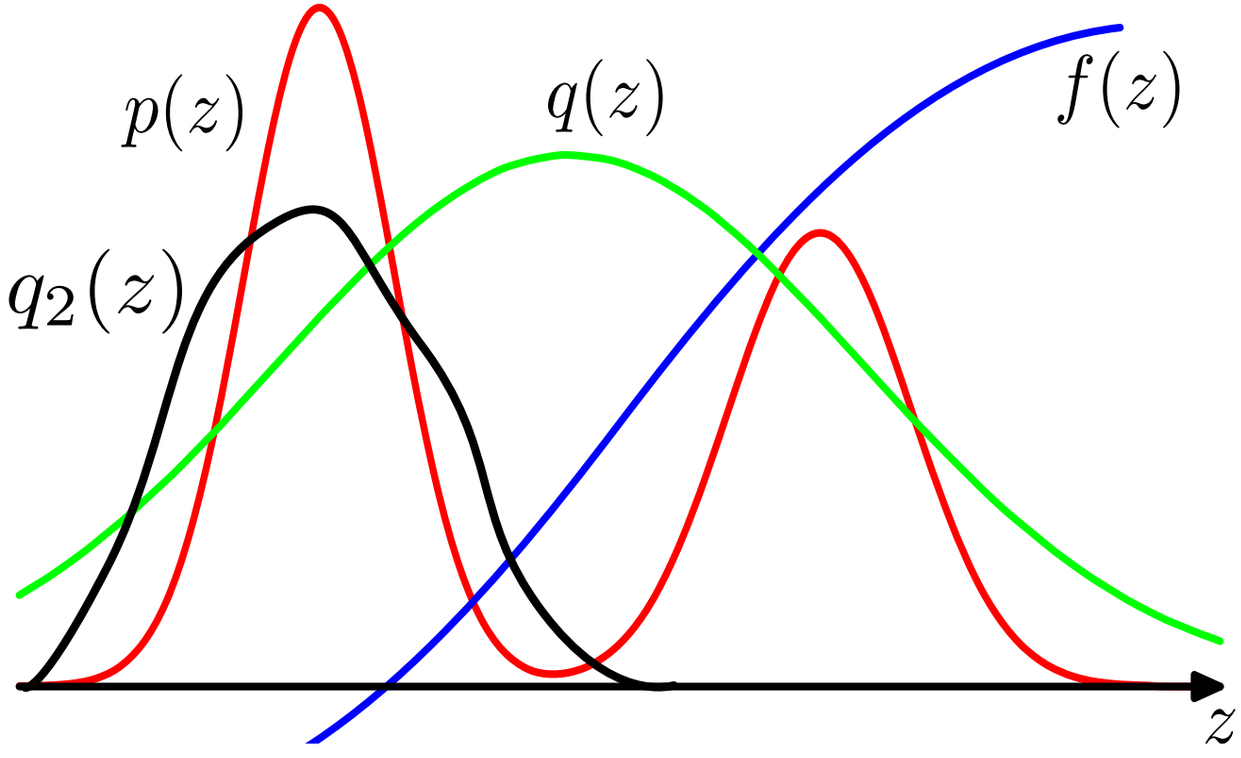
$$\begin{aligned}\mathbb{E}[f] &= \int f(\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}f(\mathbf{z}^{(l)}).\end{aligned}$$

importance weights

Generating samples from P(z)

Both rejection sampling and importance sampling suffer in high-dimensions. Rej. samp. may always reject; impt. samp. may have huge or even infinite variance

Also, it is very difficult to construct a proposal Q(z) that matches P(z); if P is multi-modal and we miss even one of the modes, the match between Q and P will be poor.



#3: The winning idea

Insight: Don't use a fixed proposal Q(z)

Instead, do something like:

1. **start** with some proposal
2. **pick** a sample according to it
3. **decide** if the sample is "good enough"
4. if not good, **update** proposal, else **accept**
5. **be clever** to ensure that this does the job!

This idea at the heart of Metropolis-Hastings (MH) method for sampling: one of the top-10 algorithms of 20th century!

In contrast to rejection samp. and impt. sampling, **not require** the varying proposal to "look like" P(z). It still works.

Simple MH sampler: Gibbs Sampler

want samples from $p(z_1, z_2, \ldots, z_n)$

Key idea: If sampling from conditionals "easy", then use cond. as proposal and always "accept"

For t = 1, …, T:

sample $z_1^{(t+1)} \sim p(z_1 | z_2^{(t)}, z_3^{(t)}, \ldots, z_n^{(t)})$ ← proposal

accept

sample $z_2^{(t+1)} \sim p(z_2 | z_1^{(t+1)}, z_3^{(t)}, \ldots, z_n^{(t)})$ ← new proposal

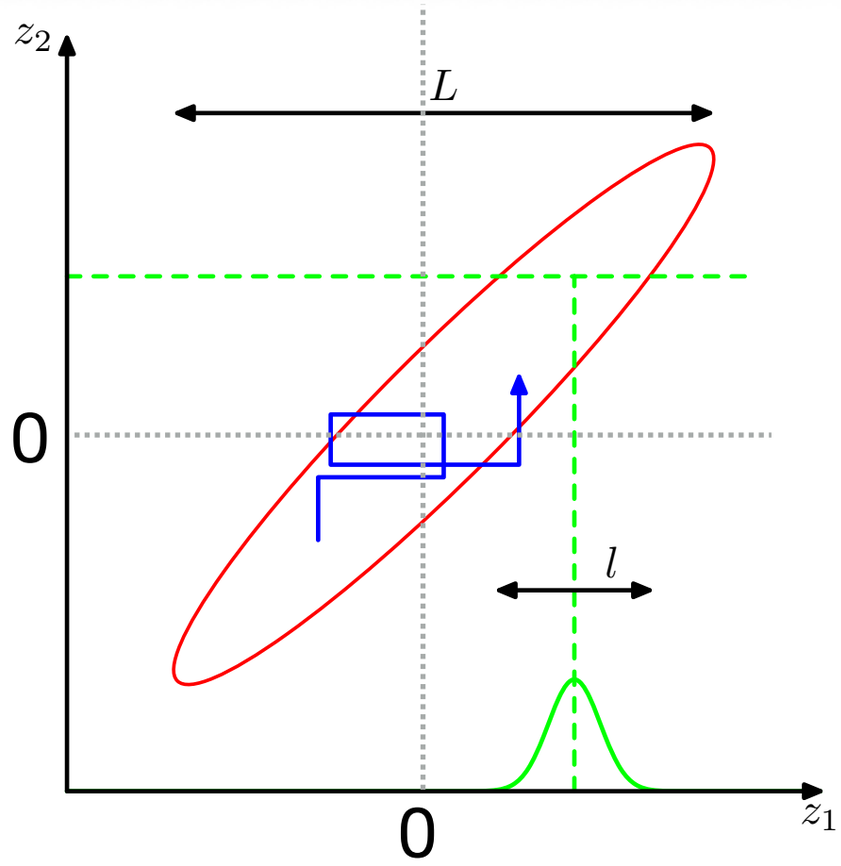
$\vdots$

sample $z_n^{(t+1)} \sim p(z_n | z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{n-1}^{(t+1)})$

Gibbs sampling: example

Illustration of Gibbs sampling by alternate updates of two variables whose distribution is a correlated Gaussian. The step size is governed by the standard deviation of the conditional distribution (green curve), and is $O(l)$, leading to slow progress in the direction of elongation of the joint distribution (red ellipse). The number of steps needed to obtain an independent sample from the distribution is $O((L/l)^2)$.

[Bishop, pg. 545]



$$p(z) = \mathcal{N}(0, \Sigma)$$

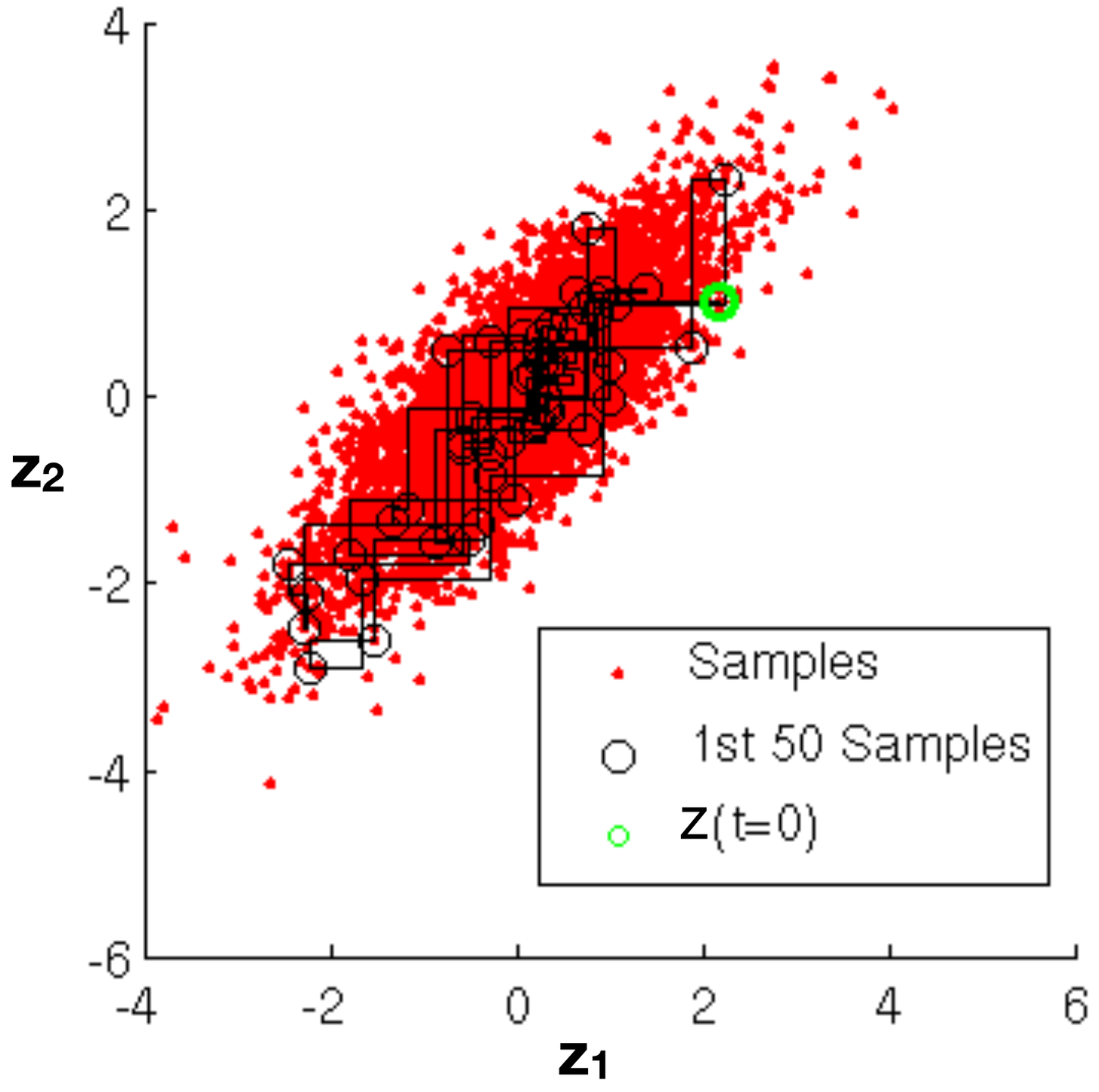
$$\Sigma = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix}$$

Exercise: Derive the conditional distributions

$$p(z_1 | z_2^{(t)})$$

$$p(z_2 | z_1^{(t+1)})$$ (these are again Gaussian)

Gibbs sampling: example



Notice the staircase due to alternating over z_1 and z_2

adapted from: https://theclevermachine.wordpress.com/2012/11/05/mcmc-the-gibbs-sampler/

High-level comments

- Gibbs samplers very popular for Bayesian methods
- Bayesian models often devised so that conditional distributions easily obtained, typically in closed form
- Broadly, though, like many MCMC techniques suffer from what is often called “slow mixing.”
- Slow mixing can be due to “random walk” nature of the Markov chain, as well as the tendency of the Markov chain to get “stuck,” only sampling a single region of having high-probability P(z)
- This behavior bad for distributions with multiple modes, heavy tails
- Great effort devoted in research to speed up MCMC methods
- No “one-size-fits-all” technique, though (complexity theory barriers)

Metropolis-Hastings Sampling

Use proposal that depends on current state $z^{(t)}$, i.e., $Q(z \mid z^{(t)})$

Can use $P^*(z)$, the un-normalized version of $P(z)$, a huge plus!

Initialize a starting guess $Q(z)$; $t=0$

Repeat until "burn in" or some stopping criteria met

Draw a tentative sample z' from $Q(z|z^{(t)})$

Accept or **reject** z' with probability

$$A(z'|z) = \min\left(1, \frac{P^*(z')Q(z^{(t)}|z')}{P^*(z^{(t)})Q(z'|z^{(t)})}\right)$$

If z' is accepted

set $z^{(t+1)}=z'$

else

$z^{(t+1)}=z^{(t)}$

observe: generated samples are dependent

Metropolis-Hastings

$$A(z'|z) = \min\left(1, \frac{P^*(z')Q(z^{(t)}|z')}{P^*(z^{(t)})Q(z'|z^{(t)})}\right)$$

Notice, we only need ratios: $P^*(z')/P^*(z^{(t)})$ and $Q(z^{(t)}|z')/Q(z'|z^{(t)})$

Thus, no knowledge of normalizing constants required!

Example:

[see example at: *http://people.csail.mit.edu/dsontag/courses/pgm13/slides/lecture9.pdf*]

Exercise: Show that Gibbs sampling is a special case of MH!

Informally: if $Q(z'|z)>0$ for all z,z', as $t\to\infty$, distribution of $z(t)$ tends to $P^*(z)/Z_p$

But, how fast?

Metropolis-Hastings intuition

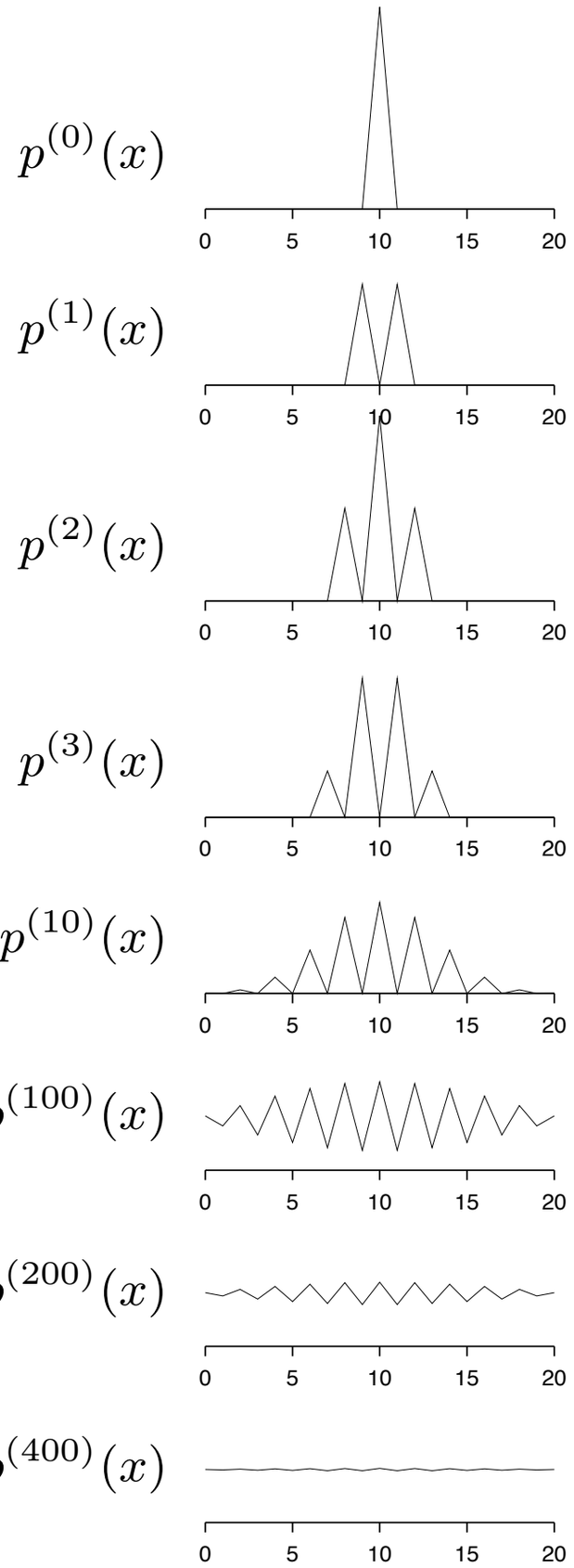
Exercise: Consider a 1D random walk. At each step, we move either left or right with equal probability. Show that if we take steps of size **ε** then after T such steps, we are likely to have moved only a distance of $\sqrt{T\varepsilon}$

$$P(x) = \begin{cases} 1/21 & x \in \{0, 1, 2, \ldots, 20\} \\ 0 & \text{otherwise.} \end{cases}$$

Proposal

$$Q(x'; x) = \begin{cases} 1/2 & x' = x \pm 1 \\ 0 & \text{otherwise.} \end{cases}$$

Essentially, at each step we explore with a step of 1. To cover 20 “states”, we require ~ 20^2 = 400 steps



Credit: MacKay, 2005

Metropolis-Hastings intuition

Rule of thumb:

If the largest length scale of the space (in multiple dimensions) of probable states is L, an MH method whose proposal generates a random walk with step size ε, must be run for at least $O((L/\varepsilon)^2)$ iterations to obtain an independent sample (assuming that essentially, each step is accepted; otherwise even more steps)

If P(z) has several widely separated modes, then it will take much longer to get indp. samples

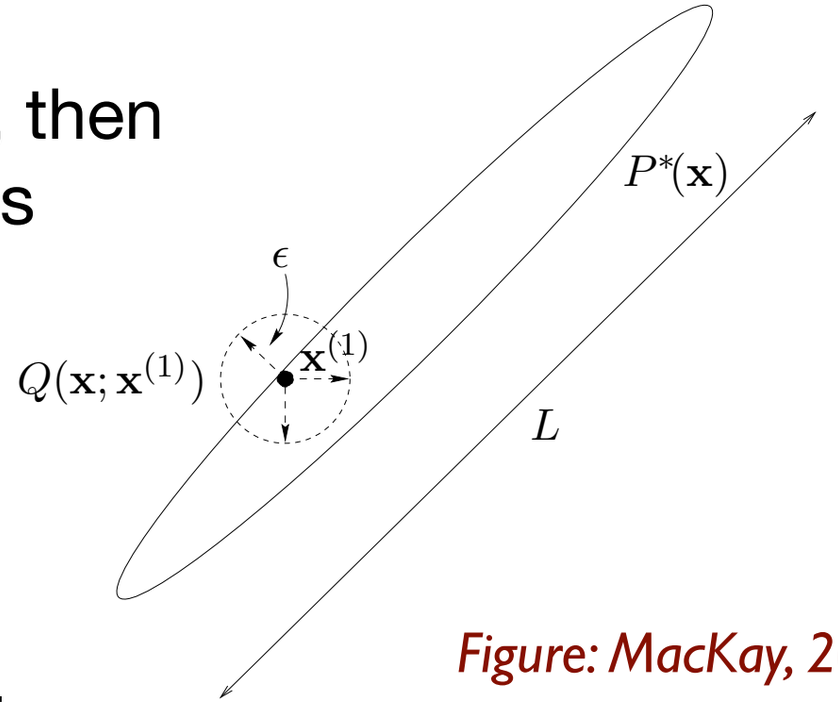


Figure: MacKay, 2005

small length scale: to avoid too many rejections (large step may easily take us into a low probability area, and hence cause rejection); **overly small steps**: random walk, also bad.

#onlyatMIT

Some theory

Life is gray, but the golden tree of theory is always green.

—Szerb on Goethe (*Journey by Moonlight*).

Markov Chains

Aim: analyze convergence to target distribution P(z)

- To understand why MH (and thus Gibbs) work, we recall Markov Chains
- Recall a Markov Chain is a sequence of rvs $(z^{(1)}, z^{(2)},\ldots,z^{(t)})$ such that $P(z^{(t+1)}=z|z^{(1)},\ldots,z^{(t)}) = P(z^{(t+1)}=z|z^{(t)})$
- **Transition kernel:** $T(z'|z) = P(z'|z)$
  (this is a matrix for discrete states, operator for continuous states)

This is a homogenous transition kernel (fixed with time 't')

Markov Chains for MH

Key idea: Think in terms of distributions over states. Then, moving through state space, amounts to transiting between distributions.

Thus, let $p^{(t)}(z)$ be distribution over states at time t

Transition kernel $T(z'|z)$

Update $$p^{(t+1)}(z') = \int T(z'|z)p^{(t)}(z)dz$$

$$P(x) = \begin{cases} 1/21 & x \in \{0, 1, 2, \ldots, 20\} \\ 0 & \text{otherwise.} \end{cases}$$

Proposal

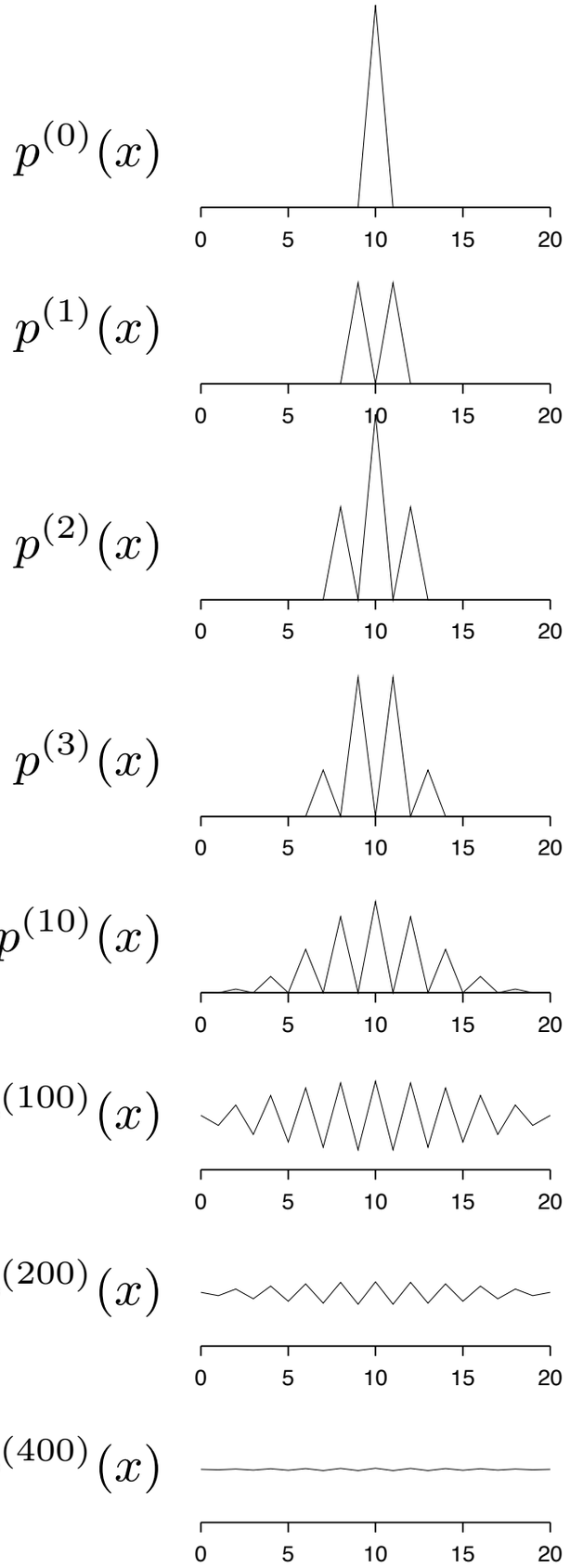
$$Q(x'; x) = \begin{cases} 1/2 & x' = x \pm 1 \\ 0 & \text{otherwise.} \end{cases}$$

$p^0(x)$=[….,1,….]

$p^{(t+1)}(x)$=T$p^{(t)}(x)$

T=

$$\begin{bmatrix}
1/2 & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & . & 1/2 & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & 1/2 & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & 1/2 & 1/2
\end{bmatrix}$$



Credit: MacKay, 2005

Markov Chain MC

Initial state: $p^{(0)}(z)$

Transition probability: $T(z'|z)$

Update $p^{(t+1)}(z') = \int T(z'|z)p^{(t)}(z)dz$

Required conditions: (for above process to converge to target)

1. Target distribution is a stationary / invariant under T(z'|z)

$$\pi(z') = \int T(z'|z)\pi(z)dz$$

2. Chain is "*ergodic*", that is

$$p^{(t)}(z) \to \pi(z), \text{ as } t \to \infty, \text{for any } p^{(0)}(z)$$

MH convergence analysis

Ergodicity can break down if chain is **reducible** or **periodic**

Reducible: state space contains two or more subsets of states **not** reachable from each other; chain ends up having many invariant distribs.

Exercise: What does this imply for eigenvalues of transition matrix?

Periodic: For some initial conditions, $p^{(t)}(z)$ doesn't tend to an invariant distribution, but instead to a periodic limit cycle. Transition matrix of such a chain has more than one eigenvalue of magnitude =1

MH convergence analysis

Claim: Assuming ergodicity, target distribution P(z) is an invariant distr. for chain defined by MH.

Proof:

Recall we draw according to Q(z'|z) and accept as per A(z'|z)

$$A(z'|z) = \min\left(1, \frac{P(z')Q(z|z')}{P(z)Q(z'|z)}\right)$$

Hence transition kernel is $T(z'|z) = A(z'|z)Q(z'|z)$

Notice that if A(z'|z) < 1, then A(z|z') = 1 (**Why?**)

MH convergence analysis

Now suppose A(z'|z) < 1, and thus A(z|z')=1. Thus,

$$A(z'|z) = \frac{P(z')Q(z|z')}{P(z)Q(z'|z)}$$

$$Q(z'|z)A(z'|z)P(z) = Q(z|z')P(z')$$

$$Q(z'|z)A(z'|z)P(z) = Q(z|z')A(z|z')P(z')$$

$$T(z'|z)P(z) = T(z|z')P(z')$$

so-called "*detailed balance*" condition

And once we have detailed balance, stationarity follows easily!

MH convergence analysis

$$T(z'|z)P(z) = T(z|z')P(z')$$

$$\sum_z T(z'|z)P(z) = \sum_z T(z|z')P(z')$$

$$\sum_z T(z'|z)P(z) = P(z')\sum_z T(z|z')$$

$$\sum T(z'|z)P(z) = P(z')$$

This is the desired stationarity

(can replace sum by integrals as necessary)

Low brow view: MH is like running the "power method" to compute the dominant eigenvector (aka stationary distr.) of the transition matrix. Because of irreducibility, aperiodicity, and since transition matrix is non-negative, iterating with it will converge to its dominant eigenvector.

Further explorations

- How fast does the chain mix?
- Detecting convergence / burn in
- Methods for speeding up mixing
- Hybrid Monte Carlo / **Hamiltonian Monte Carlo** (uses gradient info as well as momentum!)

Explore: Stochastic gradient versions of Langevin MC, HMC, etc.

Explore: What about blending MCMC with Variational Inference?

References

Chapter 29-32 of David MacKay's book:
"Information Theory, Inference, and Learning Algorithms"

Andrieu, de Freitas, Doucet, Jordan.
"*Introduction to MCMC for Machine Learning*"

Michael Betancourt
"*A Conceptual introduction to Hamiltonian Monte Carlo*"

Christopher Bishop
Chapter 11 of textbook