Introduction to MCMC for deep learning

Roadmap:
— Motivation: probabilistic modelling
— Monte Carlo, importance sampling
— Gibbs sampling, M–H
— Auxiliary variable methods

Iain Murray
School of Informatics, University of Edinburgh
“a probabilistic framework isn’t necessary, or even always useful. . .

. . . retro-fitting our new models to some probabilistic framework has little benefit”
Drawing model fantasies

— Insight into models
— Improve learning
— Communication
Polygonal random fields

Paskin and Thrun (2005)
Natural patch fantasies

From Osindero and Hinton (2008)
Creating training data

Microsoft Kinect  (Shotton et al., 2011)

Shallow learning: random forest applied to fantasies

Future deep learning?
Figure 1: Our unified graphical model (also known as a Bayesian network [27]), for astronomical image data. It integrates in a principled framework: large-scale cosmological models of galaxy and Milky Way formation; galaxy appearance models; spectral emission models and detailed camera, sky and telescope models. The shaded oval nodes are observed variables (i.e., their values are known) while the unshaded ones are unobserved and hence will be inferred from the raw astronomical data. The square nodes represent priors, typically informed by well-understood physics models. The arrows represent dependencies between variables in the model (and the lack thereof correspond to assumptions of independence). The conditional probability distributions within the model (which detail how a particular node depends on those variables which point to it) are not shown, but will be described in the text. The rectangles refer to replications of variables, e.g. an image will contain many stars/galaxies. The realization of this model is the ultimate goal of the project, but initial work will focus on sub-pieces of the model. This figure is best viewed in color.
Roadmap

— Probabilistic models

— **Simple Monte Carlo**
  Importance Sampling

— **Markov chain Monte Carlo (MCMC)**
  Gibbs sampling, M–H

— Auxiliary variable methods
  Swendsen–Wang, HMC
Sampling simple distributions

Use library routines for univariate distributions (and some other special cases)

This book (free online) explains how some of them work

http://luc.devroye.org/rnbookindex.html
Sampling from densities

Draw points uniformly under the curve:

Probability mass to left of point $\sim$ Uniform$[0,1]$
Rejection sampling

Sampling from $\pi(x)$ using tractable $q(x)$:

$q(x) \geq \pi^*(x), \forall x$

$\pi^*(x) = c \cdot \pi(x)$

Figure credit: Ryan P. Adams
Simple Monte Carlo

\[ \int f(x) P(x) \, dx \]

\[ \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \]

\[ x^{(s)} \sim P(x) \]

Unbiased. Variance \( \sim 1/S \)
Aside: Marginalization

Function of subset, \[ \int f(x_C)P(x_C) \, dx_C \]

Simulate all variables anyway:

\[ I \approx \frac{1}{S} \sum_{s=1}^{S} f(x_C^{(s)}), \quad x^{(s)} \sim P(x) \]
**Importance sampling**

**Rewrite integral:** expectation under simple distribution $Q$:

\[
\int f(x) \, P(x) \, dx = \int f(x) \frac{P(x)}{Q(x)} \, Q(x) \, dx,
\]

\[
\approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}) \frac{P(x^{(s)})}{Q(x^{(s)})}, \quad x^{(s)} \sim Q(x)
\]

Simple Monte Carlo applied to any integral.

Unbiased and independent of dimension?
Importance sampling (2)

Previous slide assumed we could evaluate $P(x) = P^*(x)/\mathcal{Z}_P$

\[
\int f(x) P(x) \, dx \approx \frac{\mathcal{Z}_Q}{\mathcal{Z}_P} \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}) \frac{P^*(x^{(s)})}{Q^*(x^{(s)})} w^*(s), \quad x^{(s)} \sim Q(x)
\]

\[
\approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}) \frac{w^*(s)}{\frac{1}{S} \sum_{s'} w^*(s')}
\]

This estimator is **consistent** but **biased**

**Exercise:** Prove that $\mathcal{Z}_P/\mathcal{Z}_Q \approx \frac{1}{S} \sum_s w^*(s)$
Rejection sampling RBMs

Product of experts:

— Draw fantasy from each expert

— If they happen to be exactly the same, accept!
Application to large problems

Approximations scale badly with dimensionality

Example: \[ P(x) = \mathcal{N}(0, \mathbb{I}), \quad Q(x) = \mathcal{N}(0, \sigma^2\mathbb{I}) \]

Rejection sampling:
Requires \( \sigma \geq 1 \). Fraction of proposals accepted = \( \sigma^{-D} \)

Importance sampling:
\[
\text{Var}[P(x)/Q(x)] = \left( \frac{\sigma^2}{2^{1/\sigma^2}} \right)^{D/2} - 1
\]
Infinite / undefined variance if \( \sigma \leq 1/\sqrt{2} \)
Unbiased positive estimators

\[ P(w) \]

\( w \)

\( 0 \quad 10 \quad 20 \quad 30 \quad 40 \quad 50 \)

\[ P(w) \]

\( w \)

\( 0 \quad 10 \quad 20 \quad 30 \quad 40 \quad 50 \)
Roadmap

— Probabilistic models

— Simple Monte Carlo
  Importance Sampling

— Markov chain Monte Carlo, MCMC
  Gibbs sampling, M–H

— Auxiliary variable methods
  Swendsen–Wang, HMC
Target distribution

\[ P(x) = \frac{1}{Z} e^{-E(x)} \]

e.g. \( x = \)
Local moves

\[ Q(x'; x) \]
**Goal:** a Markov chain,

\[ x_t \sim T(x_t \leftarrow x_{t-1}) \], such that:

\[ P(x^{(t)}) = e^{-E(x^{(t)})}/Z \] for large t.
Invariant/stationary condition

If $x^{(t-1)}$ is a sample from $P$, $x^{(t)}$ is also a sample from $P$.

$$\sum_x T(x' \leftarrow x) P(x) = P(x')$$
Ergodicity

Unique invariant distribution

if ‘forget’ starting point, $x^{(0)}$
Quick review

**MCMC:** biased random walk exploring a target dist.

Markov steps,

\[ x^{(s)} \sim T (x^{(s)} \leftarrow x^{(s-1)}) \]

MCMC gives approximate, correlated samples

\[ \mathbb{E}_P[f] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}) \]

\( T \) must leave target invariant

\( T \) must be able to get everywhere in \( K \) steps
Gibbs sampling

Pick variables in turn or randomly, and resample $P(x_i | x_{j \neq i})$

\[ T_i(x' \leftarrow x) = P(x'_i | x_{j \neq i}) \delta(x'_j - x_{j \neq i}) \]
Gibbs sampling correctness

\[ P(x) = P(x_i | x_{\backslash i}) P(x_{\backslash i}) \]

Simulate by drawing \(x_{\backslash i}\), then \(x_i | x_{\backslash i}\)

Draw \(x_{\backslash i}\): sample \(x\), throw initial \(x_i\) away
Reverse operators

If $T$ leaves $P(x)$ stationary, define a reverse operator

$$R(x \leftarrow x') = \frac{T(x' \leftarrow x) P(x)}{\sum_x T(x' \leftarrow x) P(x)} = \frac{T(x' \leftarrow x) P(x)}{P(x')}.$$  

A necessary condition: there exists $R$ such that:

$$T(x' \leftarrow x) P(x) = R(x \leftarrow x') P(x'), \quad \forall x, x'.$$

If $R = T$, known as detailed balance (not necessary)
Balance condition

\[ T(x' \leftarrow x) P(x) = R(x \leftarrow x') P(x') \]

Implies that \( P(x) \) is left invariant:

\[ \sum_{x} T(x' \leftarrow x) P(x) = P(x') \sum_{x} R(x \leftarrow x') \]
Metropolis–Hastings

**Arbitrary proposals** $\sim Q$:

$$Q(x'; x) P(x) \neq Q(x; x') P(x')$$

**Satisfies detailed balance** by rejecting moves:

$$T(x' \leftarrow x) = \begin{cases} Q(x'; x) \min \left(1, \frac{P(x') Q(x'; x')}{P(x) Q(x'; x)} \right) & x' \neq x \\ \ldots & x' = x \end{cases}$$

PRML, Bishop (2006)
Metropolis–Hastings

Transition operator

- Propose a move from the current state $Q(x'; x)$, e.g. $N(x, \sigma^2)$
- Accept with probability $\min\left(1, \frac{P(x')Q(x; x')}{P(x)Q(x'; x)}\right)$
- Otherwise next state in chain is a copy of current state

Notes

- Can use $P^* \propto P(x)$; normalizer cancels in acceptance ratio
- Satisfies detailed balance (shown below)
- $Q$ must be chosen so chain is ergodic

\[
P(x) \cdot T(x' \leftarrow x) = P(x) \cdot Q(x'; x) \min\left(1, \frac{P(x')Q(x; x')}{P(x)Q(x'; x)}\right) = \min\left(P(x)Q(x'; x), P(x')Q(x; x')\right)
= P(x') \cdot Q(x; x') \min\left(1, \frac{P(x)Q(x'; x)}{P(x')Q(x; x')}\right) = P(x') \cdot T(x \leftarrow x')
\]
function samples = dumb_metropolis(init, log_ptilde, iters, sigma)

D = numel(init);
samples = zeros(D, iters);

state = init;
Lp_state = log_ptilde(state);
for ss = 1:iters
    % Propose
    prop = state + sigma*randn(size(state));
    Lp_prop = log_ptilde(prop);
    if log(rand) < (Lp_prop - Lp_state)
        % Accept
        state = prop;
        Lp_state = Lp_prop;
    end
    samples(:, ss) = state(:);
end
Step-size demo

Explore $\mathcal{N}(0, 1)$ with different step sizes $\sigma$

```matlab
sigma = @(s) plot(dumb_metropolis(0, @(x)-0.5*x*x, 1e3, s));

sigma(0.1)
99.8% accepts

sigma(1)
68.4% accepts

sigma(100)
0.5% accepts
```
Generic proposals use
\[ Q(x'; x) = \mathcal{N}(x, \sigma^2) \]

\(\sigma\) large \(\rightarrow\) many rejections

\(\sigma\) small \(\rightarrow\) slow diffusion:
\[ \sim (L/\sigma)^2 \] iterations required

Adapted from MacKay (2003)
An MCMC strategy

Come up with good proposals $Q(x'; x)$

Combine transition operators:

\begin{align*}
x_1 & \sim T_A(\cdot \leftarrow x_0) \\
x_2 & \sim T_B(\cdot \leftarrow x_1) \\
x_3 & \sim T_C(\cdot \leftarrow x_2) \\
x_4 & \sim T_A(\cdot \leftarrow x_3) \\
x_5 & \sim T_B(\cdot \leftarrow x_4) \\
\ldots
\end{align*}
Roadmap

— Probabilistic models

— Simple Monte Carlo
  Importance Sampling

— Markov chain Monte Carlo (MCMC)
  Gibbs sampling, M–H

— Auxiliary variable methods
  Swendsen–Wang, HMC
The point of MCMC is to sum out variables, yet:

\[ \int f(x)P(x) \, dx = \int f(x)P(x, v) \, dx \, dv \]

\[ \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x, v \sim P(x, v) \]

We might want to introduce \( v \) if:

- \( P(x \mid v) \) and \( P(v \mid x) \) are simple \hspace{1cm} \text{(Cf RBMs, Martens and Sutskever 2010)}
- \( P(x, v) \) is otherwise easier to navigate

Seminal algorithm using auxiliary variables
Define a joint distribution:

\[ P(x, v) \propto e^{-E(x)} e^{-v^\top v/2} = e^{-H(x,v)} \]

Markov chain operators

- Gibbs sample velocity
- Simulate Hamiltonian dynamics
  - Conservation of energy means \( P(x, v) = P(x', v') \)
  - Metropolis acceptance probability is 1
Example / warning

Proposal:
\[ x_{t+1} = 9x_t + 1, \quad 0 < x_t < 1 \]
\[ x_{t+1} = (x_t - 1)/9, \quad 1 < x_t < 10 \]

Accept move with probability:
\[ \min \left(1, \frac{P(x')Q(x; x')}{P(x)Q(x'; x)} \right) = \min \left(1, \frac{P(x')}{P(x)} \right) \quad \text{(WRONG!)} \]
Leap-frog dynamics

a discrete approximation to Hamiltonian dynamics:

\[ v_i(t + \frac{\epsilon}{2}) = v_i(t) - \frac{\epsilon}{2} \frac{\partial E(x(t))}{\partial x_i} \]

\[ x_i(t + \epsilon) = x_i(t) + \epsilon v_i(t + \frac{\epsilon}{2}) \]

\[ p_i(t + \epsilon) = v_i(t + \frac{\epsilon}{2}) - \frac{\epsilon}{2} \frac{\partial E(x(t + \epsilon))}{\partial x_i} \]

- \( H \) is not conserved
- Transformation has unit Jacobian
- Acceptance probability becomes
  \[ \min[1, \exp(H(v, x) - H(v', x'))] \]
Hamiltonian Monte Carlo

The algorithm:

- Gibbs sample velocity $\sim \mathcal{N}(0, \mathbb{I})$
- Simulate $L$ leapfrog steps
- Accept with probability $\min[1, \exp(H(v, x) - H(v', x'))]$

The original name is **Hybrid Monte Carlo**, with reference to the “hybrid” dynamical simulation method.
Hamiltonian dynamics

Recommended reading:
MCMC using Hamiltonian dynamics, Radford M. Neal, 2011.
Handbook of Markov Chain Monte Carlo

Recent developments include:
NUTS: No U-Turn Sampler
http://arxiv.org/abs/1111.4246
Riemann manifold Hamiltonian Monte Carlo
http://www.dcs.gla.ac.uk/inference/rmhmc/
Summary of auxiliary variables

- Swendsen–Wang
- Hamiltonian (Hybrid) Monte Carlo
- Slice sampling

Some of my auxiliary representation work:

Doubly-intractable distributions

Population methods for better mixing (on parallel hardware)

Being robust to bad random number generators

Slice-sampling hierarchical latent Gaussian models
Overview

— Probabilistic models

— Simple Monte Carlo
  Importance Sampling

— Markov chain Monte Carlo (MCMC)
  Gibbs sampling, M–H

— Auxiliary variable methods
  Swendsen–Wang, HMC
Appendix slides
Finding $P(x_i = 1)$

**Method 1:** fraction of time $x_i = 1$

$$P(x_i = 1) = \sum_{x_i} \mathbb{I}(x_i = 1)P(x_i) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}(x_i^{(s)}) , \quad x_i^{(s)} \sim P(x_i)$$

**Method 2:** average of $P(x_i = 1 | x_{\setminus i})$

$$P(x_i = 1) = \sum_{x_{\setminus i}} P(x_i = 1 | x_{\setminus i})P(x_{\setminus i})$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} P(x_i = 1 | x_{\setminus i}^{(s)}) , \quad x_{\setminus i}^{(s)} \sim P(x_{\setminus i})$$

Example of “Rao-Blackwellization”.
More generally

This is easy

\[ I = \sum_{\mathbf{x}} f(x_i)P(\mathbf{x}) \approx \frac{1}{S} \sum_{s=1}^{S} f(x_i^{(s)}), \quad \mathbf{x}^{(s)} \sim P(\mathbf{x}) \]

But this might be better

\[ I = \sum_{\mathbf{x}} f(x_i)P(x_i|\mathbf{x}_{\setminus i})P(\mathbf{x}_{\setminus i}) = \sum_{\mathbf{x}_{\setminus i}} \left( \sum_{x_i} f(x_i)P(x_i|\mathbf{x}_{\setminus i}) \right) P(\mathbf{x}_{\setminus i}) \]

\[ \approx \frac{1}{S} \sum_{s=1}^{S} \left( \sum_{x_i} f(x_i)P(x_i|\mathbf{x}_{\setminus i}^{(s)}) \right), \quad \mathbf{x}_{\setminus i}^{(s)} \sim P(\mathbf{x}_{\setminus i}) \]
How should we run MCMC?

- The samples aren’t independent. Should we thin, only keep every $K$th sample?
- Arbitrary initialization means starting iterations are bad. Should we discard a “burn-in” period?
- Maybe we should perform multiple runs?
- How do we know if we have run for long enough?
Can *thin* samples so approximately independent. But, *can use all samples.*

The simple Monte Carlo estimator is still:

— consistent
— unbiased if the chain has “burned in”

**The correct motivation to thin:**

if computing $f(x^{(s)})$ is expensive

In some special circumstances strategic thinning can help.

http://dx.doi.org/10.1016/S0167-7152(99)00142-X — Thanks to Simon Lacoste-Julien for the reference.
Empirical diagnostics

Recommendations

**Diagnostic software:** R-CODA

For opinion on thinning, multiple runs, burn in, etc.


http://www.jstor.org/stable/2246094
Slice sampling idea

Sample point uniformly under curve $\tilde{P}(x) \propto P(x)$

$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

$$p(x|u) \propto \begin{cases} 1 & \tilde{P}(x) \geq u \\ 0 & \text{otherwise} \end{cases} = \text{“Uniform on the slice”}$$
Slice sampling

Unimodal conditionals

- bracket slice
- sample uniformly within bracket
- shrink bracket if $\tilde{P}(x) < u$ (off slice)
- accept first point on the slice
Slice sampling

Multimodal conditionals

\[ \tilde{P}(x) \]

- place bracket randomly around point
- linearly step out until bracket ends are off slice
- sample on bracket, shrinking as before

Satisfies detailed balance, leaves \( p(x|u) \) invariant
Advantages of slice-sampling:

- Easy — only require $\tilde{P}(x) \propto P(x)$
- No rejections
- Tweak params not too important

There are more advanced versions. Neal (2003) contains many ideas.
General references:

Various figures and more came from (see also references therein):

Specific points:
If you do Gibbs sampling with continuous distributions this method, which I omitted for material-overload reasons, may help:

An example of picking estimators carefully:

A key reference for auxiliary variable methods is:


An early reference for parallel tempering:

Software:
Neural networks and other flexible models: http://www.cs.utoronto.ca/~radford/fbm.software.html
CODA: http://www-fis.iarc.fr/coda/

Other Monte Carlo methods:
Nested sampling is a new Monte Carlo method with some interesting properties:
Nested sampling for general Bayesian computation, John Skilling, *Bayesian Analysis*, 2006.

Approaches based on the “multi-canonical ensemble” also solve some of the problems with traditional temperature-based methods:

A good review paper:

Particle filters / Sequential Monte Carlo are famously successful in time series modeling, but are more generally applicable.
This may be a good place to start: http://www.cs.ubc.ca/~arnaud/journals.html

Exact or perfect sampling uses Markov chain simulation but suffers no initialization bias. An amazing feat when it can be performed:
Annotated bibliography of perfectly random sampling with Markov chains, David B. Wilson
http://dbwilson.com/exact/

MCMC does not apply to *doubly-intractable* distributions. For what that even means and possible solutions see:
http://www.gatsby.ucl.ac.uk/~iam23/pub/06doubly_intractable/doubly_intractable.pdf