Approximate Inference using MCMC

9.520 Class 22

Ruslan Salakhutdinov
BCS and CSAIL, MIT
Plan

1. Introduction/Notation.

2. Examples of successful Bayesian models.


5. Markov chain Monte Carlo algorithms.
References/Acknowledgements

• Chris Bishop’s book: Pattern Recognition and Machine Learning, chapter 11 (many figures are borrowed from this book).


• Zoubin Ghahramani’s ICML tutorial on Bayesian Machine Learning: http://www.gatsby.ucl.ac.uk/~zoubin/ICML04-tutorial.html

• Ian Murray’s tutorial on Sampling Methods: http://www.cs.toronto.edu/~murray/teaching/
Basic Notation

\[ P(x) \quad \text{probability of } x \]
\[ P(x|\theta) \quad \text{conditional probability of } x \text{ given } \theta \]
\[ P(x, \theta) \quad \text{joint probability of } x \text{ and } \theta \]

Bayes Rule:

\[
P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}
\]

where

\[
P(x) = \int P(x, \theta)d\theta \quad \text{Marginalization}
\]

I will use probability distribution and probability density interchangeably. It should be obvious from the context.
**Inference Problem**

Given a dataset $\mathcal{D} = \{x_1, ..., x_n\}$:

Bayes Rule:

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}$$

- $P(\mathcal{D}|\theta)$ Likelihood function of $\theta$
- $P(\theta)$ Prior probability of $\theta$
- $P(\theta|\mathcal{D})$ Posterior distribution over $\theta$

Computing posterior distribution is known as the **inference** problem.

But:

$$P(\mathcal{D}) = \int P(\mathcal{D}, \theta) d\theta$$

This integral can be very high-dimensional and difficult to compute.
Prediction

\[ P(\theta|\mathcal{D}) = \frac{P(D|\theta)P(\theta)}{P(D)} \]

- \( P(D|\theta) \): Likelihood function of \( \theta \)
- \( P(\theta) \): Prior probability of \( \theta \)
- \( P(\theta|\mathcal{D}) \): Posterior distribution over \( \theta \)

**Prediction**: Given \( \mathcal{D} \), computing conditional probability of \( x^* \) requires computing the following integral:

\[
P(x^*|\mathcal{D}) = \int P(x^*|\theta, \mathcal{D})P(\theta|\mathcal{D})d\theta
\]

\[
= \mathbb{E}_{P(\theta|\mathcal{D})}[P(x^*|\theta, \mathcal{D})]
\]

which is sometimes called **predictive distribution**.
Computing predictive distribution requires posterior \( P(\theta|\mathcal{D}) \).
Model Selection

Compare model classes, e.g. $\mathcal{M}_1$ and $\mathcal{M}_2$. Need to compute posterior probabilities given $\mathcal{D}$:

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})}$$

where

$$P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|\theta, \mathcal{M})P(\theta, \mathcal{M})d\theta$$

is known as the **marginal likelihood** or **evidence**.
Computational Challenges

• Computing marginal likelihoods often requires computing very high-dimensional integrals.

• Computing posterior distributions (and hence predictive distributions) is often analytically intractable.

• In this class, we will concentrate on Markov Chain Monte Carlo (MCMC) methods for performing approximate inference.

• First, let us look at some specific examples:
  – Bayesian Probabilistic Matrix Factorization
  – Bayesian Neural Networks
  – Dirichlet Process Mixtures (last class)
We have $N$ users, $M$ movies, and integer rating values from 1 to $K$.

Let $r_{ij}$ be the rating of user $i$ for movie $j$, and $U \in \mathbb{R}^{D \times N}$, $V \in \mathbb{R}^{D \times M}$ be latent user and movie feature matrices:

$$R \approx U^\top V$$

Goal: Predict missing ratings.
Bayesian PMF

Probabilistic linear model with Gaussian observation noise. Likelihood:

\[ p(r_{ij}|u_i, v_j, \sigma^2) = \mathcal{N}(r_{ij}|u_i^\top v_j, \sigma^2) \]

Gaussian Priors over parameters:

\[
p(U|\mu_U, \Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(u_i|\mu_u, \Sigma_u),
\]

\[
p(V|\mu_V, \Lambda_V) = \prod_{i=1}^{M} \mathcal{N}(v_i|\mu_v, \Sigma_v).
\]

Conjugate Gaussian-inverse-Wishart priors on the user and movie hyperparameters \( \Theta_U = \{\mu_u, \Sigma_u\} \) and \( \Theta_V = \{\mu_v, \Sigma_v\} \).

Hierarchical Prior.
Bayesian PMF

**Predictive distribution:** Consider predicting a rating $r^*_{ij}$ for user $i$ and query movie $j$:

$$p(r^*_{ij} | R) = \int \int p(r^*_{ij} | u_i, v_j) p(U, V, \Theta_U, \Theta_V | R) d\{U, V\} d\{\Theta_U, \Theta_V\}$$

Posterior over parameters and hyperparameters

Exact evaluation of this predictive distribution is analytically intractable.

Posterior distribution $p(U, V, \Theta_U, \Theta_V | R)$ is complicated and does not have a closed form expression.

Need to approximate.
Bayesian Neural Nets

Regression problem: Given a set of i.i.d observations \( X = \{x^n\}_{n=1}^N \) with corresponding targets \( D = \{t^n\}_{n=1}^N \).

Likelihood:
\[
p(D|X, w) = \prod_{n=1}^N \mathcal{N}(t^n|y(x^n, w), \beta^2)
\]

The mean is given by the output of the neural network:
\[
y_k(x, w) = \sum_{j=0}^M w_{kj}^2 \sigma \left( \sum_{i=0}^D w_{ji}^1 x_i \right)
\]
where \( \sigma(x) \) is the sigmoid function.

Gaussian prior over the network parameters: \( p(w) = \mathcal{N}(0, \alpha^2 I) \).
Bayesian Neural Nets

Likelihood:
\[ p(D|X, w) = \prod_{n=1}^{N} \mathcal{N}(t^n | y(x^n, w), \beta^2) \]

Gaussian prior over parameters:
\[ p(w) = \mathcal{N}(0, \alpha^2 I) \]

Posterior is analytically intractable:
\[ p(w|D, X) = \frac{p(D|w, X)p(w)}{\int p(D|w, X)p(w)dw} \]

Remark: Under certain conditions, Radford Neal (1994) showed, as the number of hidden units go to infinity, a Gaussian prior over parameters results in a Gaussian process prior for functions.
Undirected Models

\( \mathbf{x} \) is a binary random vector with \( x_i \in \{+1, -1\} \):

\[
p(\mathbf{x}) = \frac{1}{\mathcal{Z}} \exp \left( \sum_{(i,j) \in E} \theta_{ij} x_i x_j + \sum_{i \in V} \theta_i x_i \right).
\]

where \( \mathcal{Z} \) is known as partition function:

\[
\mathcal{Z} = \sum_{\mathbf{x}} \exp \left( \sum_{(i,j) \in E} \theta_{ij} x_i x_j + \sum_{i \in V} \theta_i x_i \right).
\]

If \( \mathbf{x} \) is 100-dimensional, need to sum over \( 2^{100} \) terms. The sum might decompose (e.g. junction tree). Otherwise we need to approximate.

Remark: Compare to marginal likelihood.
Monte Carlo

For most situations we will be interested in evaluating the expectation:

$$\mathbb{E}[f] = \int f(z)p(z)dz$$

We will use the following notation: $$p(z) = \frac{\tilde{p}(z)}{\mathcal{Z}}$$.

We can evaluate $$\tilde{p}(z)$$ pointwise, but cannot evaluate $$\mathcal{Z}$$.

- Posterior distribution: $$P(\theta|D) = \frac{1}{P(D)}P(D|\theta)P(\theta)$$
- Markov random fields: $$P(z) = \frac{1}{\mathcal{Z}}\exp(-E(z))$$
Simple Monte Carlo

**General Idea:** Draw independent samples \( \{z^1, ..., z^n\} \) from distribution \( p(z) \) to approximate expectation:

\[
\mathbb{E}[f] = \int f(z) p(z) dz \approx \frac{1}{N} \sum_{n=1}^{N} f(z^n) = \hat{f}
\]

Note that \( \mathbb{E}[f] = \mathbb{E}[\hat{f}] \), so the estimator \( \hat{f} \) has correct mean (unbiased). The variance:

\[
\text{var}[\hat{f}] = \frac{1}{N} \mathbb{E}[(f - \mathbb{E}[f])^2]
\]

**Remark:** The accuracy of the estimator does not depend on dimensionality of \( z \).
Simple Monte Carlo

In general:

\[ \int f(z)p(z)dz \approx \frac{1}{N} \sum_{n=1}^{N} f(z^n), \quad z^n \sim p(z) \]

Predictive distribution:

\[ P(x^*|\mathcal{D}) = \int P(x^*|\theta, \mathcal{D})P(\theta|\mathcal{D})d\theta \]

\[ \approx \frac{1}{N} \sum_{n=1}^{N} P(x^*|\theta^n, \mathcal{D}), \quad \theta^n \sim p(\theta|\mathcal{D}) \]

**Problem:** It is hard to draw exact samples from \( p(z) \).
Basic Sampling Algorithm

How to generate samples from simple non-uniform distributions assuming we can generate samples from uniform distribution.

Define: $h(y) = \int_{-\infty}^{y} p(\hat{y}) d\hat{y}$

Sample: $z \sim U[0, 1]$.

Then: $y = h^{-1}(z)$ is a sample from $p(y)$.

**Problem:** Computing cumulative $h(y)$ is just as hard!
Rejection Sampling

Sampling from target distribution \( p(z) = \tilde{p}(z)/\mathcal{Z}_p \) is difficult. Suppose we have an easy-to-sample proposal distribution \( q(z) \), such that \( kq(z) \geq \tilde{p}(z), \forall z \).

Sample \( z_0 \) from \( q(z) \).
Sample \( u_0 \) from Uniform\([0, kq(z_0)]\)

The pair \((z_0, u_0)\) has uniform distribution under the curve of \( kq(z) \).

If \( u_0 > \tilde{p}(z_0) \), the sample is rejected.
Rejection Sampling

Probability that a sample is accepted is:

\[
p(\text{accept}) = \int \frac{\tilde{p}(z)}{kq(z)} q(z) dz = \frac{1}{k} \int \tilde{p}(z) dz
\]

The fraction of accepted samples depends on the ratio of the area under \( \tilde{p}(z) \) and \( kq(z) \).

Hard to find appropriate \( q(z) \) with optimal \( k \).

Useful technique in one or two dimensions. Typically applied as a subroutine in more advanced algorithms.
Importance Sampling

Suppose we have an easy-to-sample proposal distribution $q(z)$, such that $q(z) > 0$ if $p(z) > 0$.

The quantities $w^n = p(z^n)/q(z^n)$ are known as importance weights. Unlike rejection sampling, all samples are retained. But wait: we cannot compute $p(z)$, only $\tilde{p}(z)$.
Importance Sampling

Let our proposal be of the form \( q(z) = \tilde{q}(z) / Z_q \):

\[
E[f] = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz = \frac{Z_q}{Z_p} \int f(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\
\approx \frac{Z_q}{Z_p N} \sum_n \frac{\tilde{p}(z^n)}{\tilde{q}(z^n)}f(z^n) = \frac{Z_q}{Z_p N} \sum_n w^nf(z^n), \quad z^n \sim q(z)
\]

But we can use the same importance weights to approximate \( \frac{Z_p}{Z_q} \):

\[
\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int \tilde{p}(z)dz = \int \frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \approx \frac{1}{N} \sum_n \frac{\tilde{p}(z^n)}{\tilde{q}(z^n)} = \frac{1}{N} \sum_n w^n
\]

Hence:

\[
E[f] \approx \frac{1}{N} \sum_n \frac{w^n}{\sum_n w^n}f(z^n) \quad \text{Consistent but biased.}
\]
Problems

If our proposal distribution \( q(z) \) poorly matches our target distribution \( p(z) \) then:

- Rejection Sampling: almost always rejects
- Importance Sampling: has large, possibly infinite, variance (unreliable estimator).

For high-dimensional problems, finding good proposal distributions is very hard. What can we do?

Markov Chain Monte Carlo.
A first-order Markov chain: a series of random variables \( \{z^1, ..., z^N\} \) such that the following conditional independence property holds for \( n \in \{z^1, ..., z^{N-1}\} \):

\[
p(z^{n+1}|z^1, ..., z^n) = p(z^{n+1}|z^n)
\]

We can specify Markov chain:

- probability distribution for initial state \( p(z^1) \).
- conditional probability for subsequent states in the form of transition probabilities \( T(z^{n+1} \leftarrow z^n) \equiv p(z^{n+1}|z^n) \).

**Remark:** \( T(z^{n+1} \leftarrow z^n) \) is sometimes called a transition kernel.
Markov Chains

A marginal probability of a particular state can be computed as:

\[ p(z^{n+1}) = \sum_{z^n} T(z^{n+1} \leftarrow z^n)p(z^n) \]

A distribution \( \pi(z) \) is said to be invariant or stationary with respect to a Markov chain if each step in the chain leaves \( \pi(z) \) invariant:

\[ \pi(z) = \sum_{z'} T(z \leftarrow z')\pi(z') \]

A given Markov chain may have many stationary distributions. For example: \( T(z \leftarrow z') = I\{z = z'\} \) is the identity transformation. Then any distribution is invariant.
Detailed Balance

A sufficient (but not necessary) condition for ensuring that $\pi(z)$ is invariant is to choose a transition kernel that satisfies a **detailed balance** property:

$$\pi(z') T(z \leftarrow z') = \pi(z) T(z' \leftarrow z)$$

A transition kernel that satisfies detailed balance will leave that distribution invariant:

$$\sum_{z'} \pi(z') T(z \leftarrow z') = \sum_{z'} \pi(z) T(z' \leftarrow z)$$

$$= \pi(z) \sum_{z'} T(z' \leftarrow z) = \pi(z)$$

A Markov chain that satisfies detailed balance is said to be **reversible**.
Recap

We want to sample from target distribution $\pi(z) = \tilde{\pi}(z)/Z$ (e.g. posterior distribution).

Obtaining independent samples is difficult.

- Set up a Markov chain with transition kernel $T(z' \leftarrow z)$ that leaves our target distribution $\pi(z)$ invariant.

- If the chain is **ergodic**, i.e. it is possible to go from every state to any other state (not necessarily in one move), then the chain will converge to this unique invariant distribution $\pi(z)$.

- We obtain dependent samples drawn approximately from $\pi(z)$ by simulating a Markov chain for some time.

**Ergodicity:** There exists $K$, for any starting $z$, $T^K(z' \leftarrow z) > 0$ for all $\pi(z') > 0$. 
Metropolis-Hasting Algorithm

A Markov chain transition operator from current state $z$ to a new state $z'$ is defined as follows:

- A new 'candidate' state $z^*$ is proposed according to some proposal distribution $q(z^*|z)$, e.g. $\mathcal{N}(z, \sigma^2)$.
- A candidate state $x^*$ is accepted with probability:

$$\min \left( 1, \frac{\tilde{\pi}(z^*) q(z|z^*)}{\tilde{\pi}(z) q(z^*|z)} \right)$$

- If accepted, set $z' = z^*$. Otherwise $z' = z$, or the next state is the copy of the current state.

Note: no need to know normalizing constant $\mathcal{Z}$. 
Metropolis-Hasting Algorithm

We can show that M-H transition kernel leaves $\pi(z)$ invariant by showing that it satisfies detailed balance:

$$
\pi(z)T(z' \leftarrow z) = \pi(z)q(z'|z) \min \left(1, \frac{\pi(z')q(z|z')}{\pi(z)q(z'|z)} \right)
$$

$$
= \min \left(\pi(z)q(z'|z), \pi(z')q(z|z')\right)
$$

$$
= \pi(z') \min \left(\frac{\pi(z)}{\pi(z')} \frac{q(z'|z)}{q(z|z')}, 1 \right)
$$

$$
= \pi(z')T(z \leftarrow z')
$$

Note that whether the chain is ergodic will depend on the particulars of $\pi$ and proposal distribution $q$. 
Using Metropolis algorithm to sample from Gaussian distribution with proposal $q(z'|z) = \mathcal{N}(z, 0.04)$.

accepted (green), rejected (red).
Choice of Proposal

The specific choice of proposal can greatly affect the performance of the algorithm.

Proposal distribution:
\[ q(z' | z) = \mathcal{N}(z, \rho^2). \]

- \( \rho \) large - many rejections
- \( \rho \) small - chain moves too slowly
Gibbs Sampler

Consider sampling from \( p(z_1, \ldots, z_N) \).

\[
\text{Initialize } z_i, \ i = 1, \ldots, N
\]
\[
\text{For } t=1, \ldots, T
\]
\[
\text{Sample } z_1^{t+1} \sim p(z_1 | z_2^t, \ldots, z_N^t)
\]
\[
\text{Sample } z_2^{t+1} \sim p(z_2 | z_1^{t+1}, x_3, \ldots, z_N^t)
\]
\[
\text{\ldots}
\]
\[
\text{Sample } z_N^{t+1} \sim p(z_N | z_1^{t+1}, \ldots, z_{N-1}^{t+1})
\]

Gibbs sampler is a particular instance of M-H algorithm with proposals \( p(z_n | z_i \neq n) \) → accept with probability 1. Apply a series (component-wise) of these operators.
Gibbs Sampler

Applicability of the Gibbs sampler depends on how easy it is to sample from conditional probabilities $p(z_n|z_i\neq n)$.

- For discrete random variables with a few discrete settings:
  \[
p(z_n|z_i\neq n) = \frac{p(z_n, z_i\neq n)}{\sum_{z_n} p(z_n, z_i\neq n)}
  \]
  The sum can be computed analytically.

- For continuous random variables:
  \[
p(z_n|z_i\neq n) = \frac{p(z_n, z_i\neq n)}{\int p(z_n, z_i\neq n)dz_n}
  \]
  The integral is univariate and is often analytically tractable or amenable to standard sampling methods.
**Bayesian PMF**

**Remember predictive distribution?**: Consider predicting a rating $r^*_{ij}$ for user $i$ and query movie $j$:

$$p(r^*_{ij} | R) = \int \int p(r^*_{ij} | u_i, v_j) p(U, V, \Theta_U, \Theta_V | R) d\{U, V\} d\{\Theta_U, \Theta_V\}$$

**Posterior over parameters and hyperparameters**

Use Monte Carlo approximation:

$$p(r^*_{ij} | R) \approx \frac{1}{N} \sum_{n=1}^{N} p(r^*_{ij} | u^{(n)}_i, v^{(n)}_j).$$

The samples $(u^{n}_i, v^{n}_j)$ are generated by running a Gibbs sampler, whose stationary distribution is the posterior distribution of interest.
Bayesian PMF

Monte Carlo approximation:

\[ p(r_{ij}^* | R) \approx \frac{1}{N} \sum_{n=1}^{N} p(r_{ij}^* | u_i^{(n)}, v_j^{(n)}) \].

The conditional distributions over the user and movie feature vectors are Gaussians \( \rightarrow \) easy to sample from:

\[
p(u_i | R, V, \Theta_U, \alpha) = \mathcal{N}(u_i | \mu_i^*, \Sigma_i^*)
\]
\[
p(v_j | R, U, \Theta_U, \alpha) = \mathcal{N}(v_j | \mu_j^*, \Sigma_j^*)
\]

The conditional distributions over hyperparameters also have closed form distributions \( \rightarrow \) easy to sample from.

Netflix dataset – Bayesian PMF can handle over 100 million ratings.
Main problems of MCMC:

- Hard to diagnose convergence (burning in).
- Sampling from isolated modes.

More advanced MCMC methods for sampling in distributions with isolated modes:

- Parallel tempering
- Simulated tempering
- Tempered transitions

Hamiltonian Monte Carlo methods (make use of gradient information).

Nested Sampling, Coupling from the Past, many others.
Deterministic Methods

- Laplace Approximation

- Bayesian Information Criterion (BIC)

- Variational Methods: Mean-Field, Loopy Belief Propagation along with various adaptations.

- Expectation Propagation.

- ...