Unsupervised Learning

Markov Chain Monte Carlo

Zoubin Ghahramani
zoubin@gatsby.ucl.ac.uk

Gatsby Computational Neuroscience Unit, and
MSc in Intelligent Systems, Dept Computer Science
University College London

Autumn 2002

The role of integration in statistical modelling

- E step of the EM algorithm requires expected sufficient statistics:

$$E[s(X)|\theta] = \int s(h, v)\, p(h|v, \theta)\, dh$$

- Bayesian prediction:

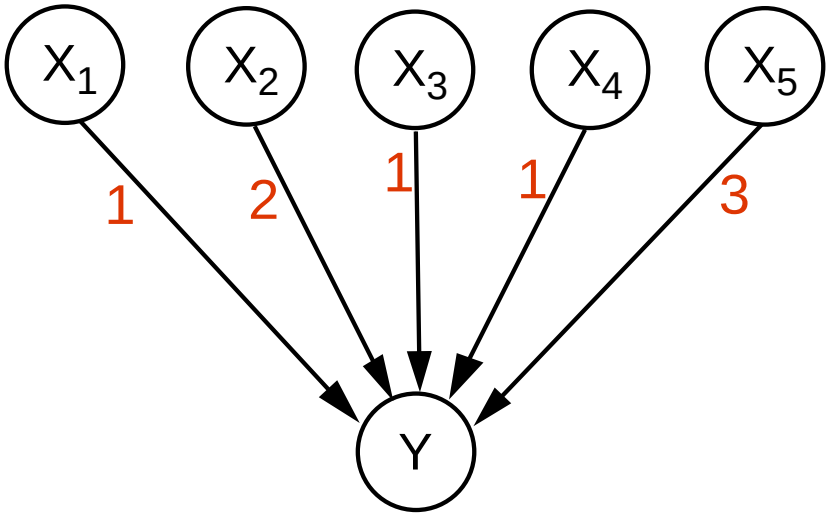
$$p(x|\mathcal{D}, m) = \int p(x|\theta, \mathcal{D}, m)\, p(\theta|\mathcal{D}, m)\, d\theta$$

- Computing model evidence for model comparison:

$$p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta, m)\, p(\theta|m)\, d\theta$$

Examples of Intractability

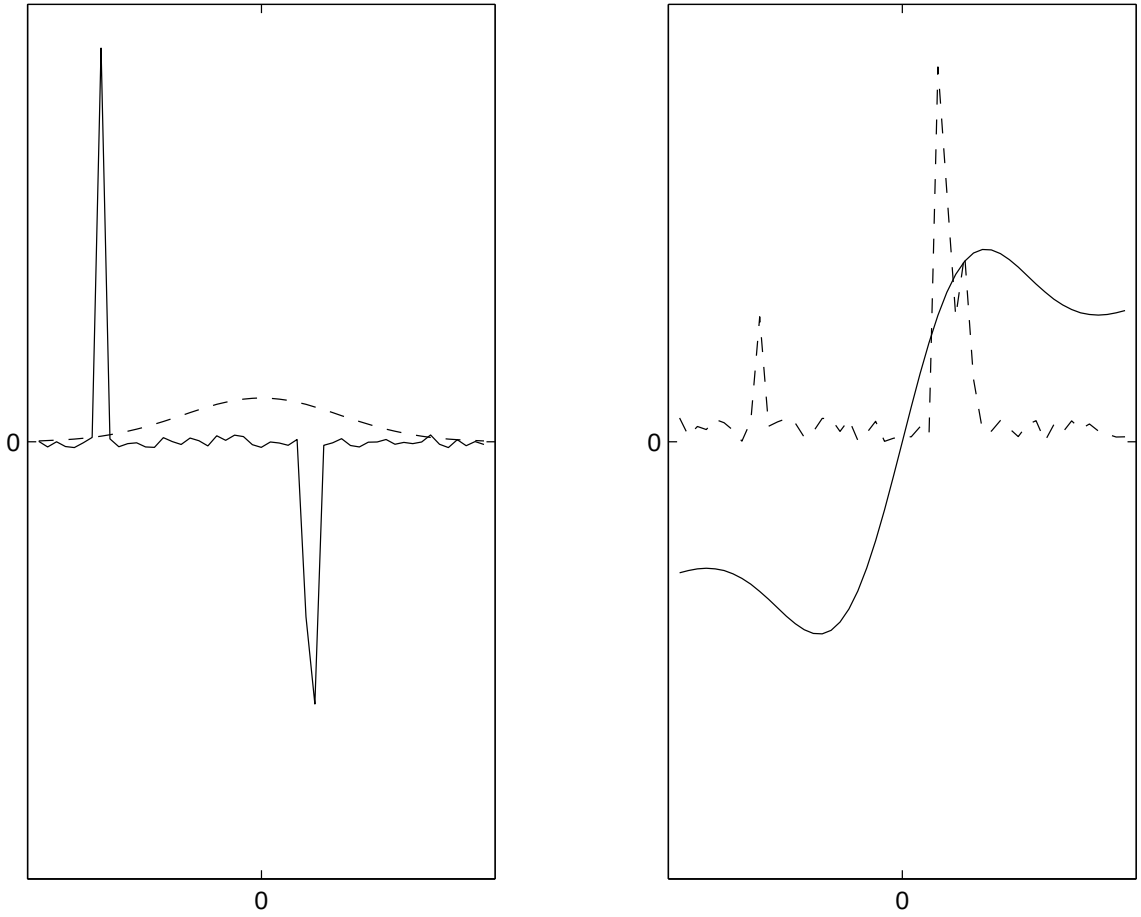
- EM for multipe cause models:



$$Y = X_1 + 2\,X_2 + X_3 + X_4 + 3\,X_5$$

- Bayesian prediction/model evidence for Mixture of Gaussians: exact computations are exponential in number of data points

The integration problem



We need to compute integrals of the form

$$\int F(x)\ p(x)dx,$$

where $F(x)$ is some function of a random variable X which has probability density $p(x)$.

Three typical difficulties:
left panel: full line is some complicated function, dashed is density;
right panel: full line is some function and dashed is complicated density;
not shown: integral in very high dimensions

Simple Monte Carlo

Simple Monte Carlo:

$$\int F(x)p(x)dx \simeq \frac{1}{T}\sum_{t=1}^{T} F(x^{(t)}),$$

where $x^{(t)}$ are (independent) samples from $p(x)$.

Attraction: unbiased; variance goes as $1/T$, independent of dimension!

Problem: it may be difficult to obtain the samples from $p(x)$.

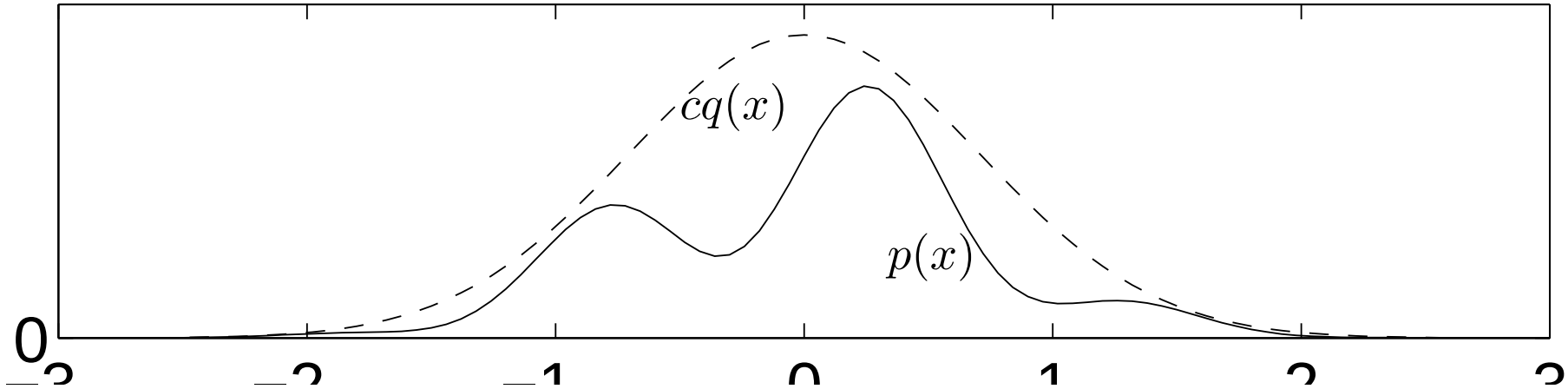
Alternatives:
Analytical approximation: expansion around the mode (i.e. Gaussian).
Numerical methods?

Rejection Sampling

Find a distribution $q(x)$ and a constant c such that $\forall x,\ p(x) \leq cq(x)$.

Sample x^* from $q(x)$ and accept x^* with probability $p(x^*)/(cq(x^*))$.

Use accepted points in simple Monte Carlo.



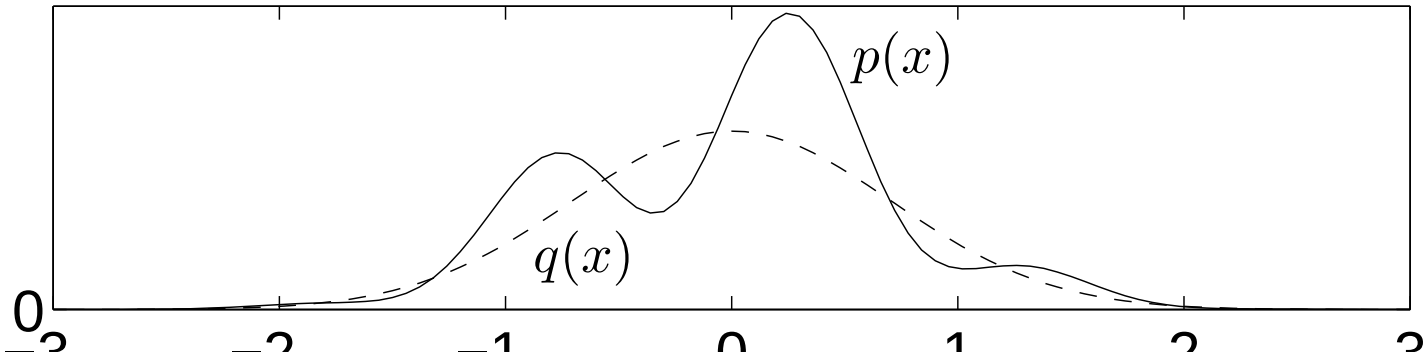
Problem: it may be difficult to find $cq(x)$ with a small c, and a $q(x)$ which is easy to sample from.

Importance Sampling

Sample $x^{(t)}$ from $q(x)$:

$$\int F(x)p(x)dx = \int F(x)\frac{p(x)}{q(x)}q(x)dx \simeq \frac{1}{T}\sum_{t=1}^{T} F(x^{(t)})\frac{p(x^{(t)})}{q(x^{(t)})},$$

where $q(x)$ is non-zero wherever $p(x)$ is; weights $w^{(t)} \equiv p(x^{(t)})/q(x^{(t)})$



Attraction: unbiased; no need for upper bound (cf rejection sampling).

Problem: it may be difficult to find a suitable $q(x)$. Monte Carlo average may be dominated by few samples (high variance); or none of the high weight samples may be found!

Analysis of Importance Sampling

Weights:

$$w^{(t)} \equiv \frac{p(x^{(t)})}{q(x^{(t)})}$$

Unbiased:

$$E_q(w) = 1$$

Variance, $V(w) = E_q(w^2) - 1$, where:

$$E_q(w^2) = \int \frac{p(x)^2}{q(x)^2} q(x) dx = \int \frac{p(x)^2}{q(x)} dx$$

What happens if $p(x) = \mathcal{N}(0, \sigma_p^2)$ and $q(x) = \mathcal{N}(0, \sigma_q^2)$?

Gibbs Sampling

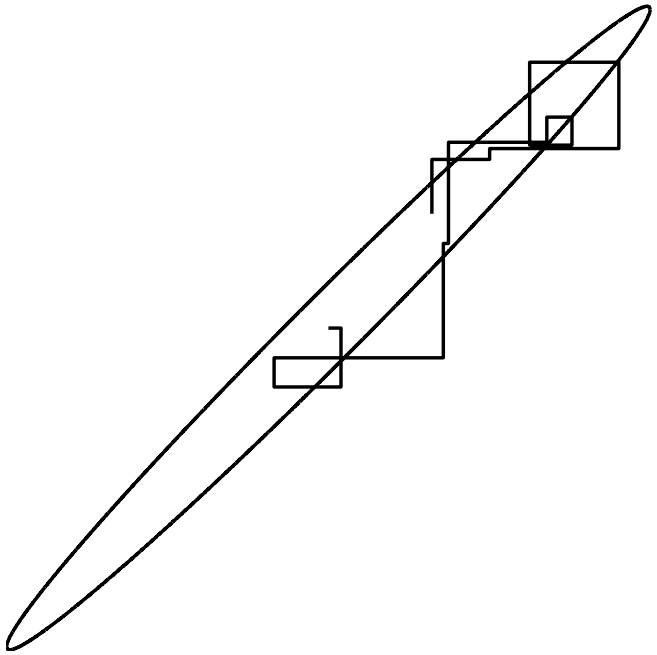
A method for sampling from a multivariate distribution, $p(\mathbf{x})$

Repeatedly:
1) pick i (either at random or in turn)
2) replace x_i by a sample from the conditional distribution

$$p(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n)$$

Gibbs sampling is feasible if it is easy to sample from the conditional probabilities.
This creates a Markov chain

$$\mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)} \rightarrow \mathbf{x}^{(3)} \rightarrow \ldots$$

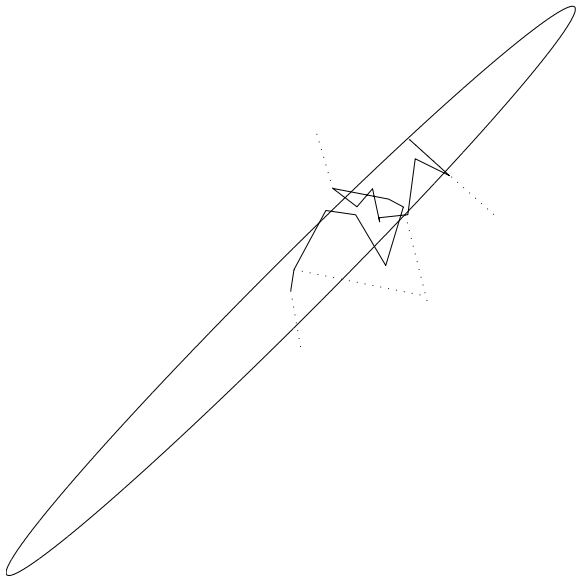


Example: 20 (half-) iterations of Gibbs sampling on a bivariate Gaussian

Under some (mild) conditions, the **equilibium distribution**, i.e. $p(\mathbf{x}^{(\infty)})$, of this Markov chain is $p(\mathbf{x})$
Terminology: This is a **Markov chain Monte Carlo (MCMC)** method

The Metropolis algorithm

Based on the current state $\mathbf{x}$, propose a new state $\mathbf{x}^*$ using a proposal distribution $S(\mathbf{x}, \mathbf{x}^*)$.
Accept the new state with probability $\min(1, p(\mathbf{x}^*)/p(\mathbf{x}))$; otherwise retain the old state.



Example: 20 iterations of global metropolis sampling from bivariate Gaussian; rejected proposals are dotted.

Local vs global proposal distributions.
Proof of correctness relies on symmetry of $S(\mathbf{x}, \mathbf{x}^*)$ and detailed balance.
Note, we need only to compute ratios of probabilities (no normalizing constants).

Hybrid Monte Carlo: overview

Typical distance traveled by a random walk in n steps is proportional to $\sqrt{n}$

Fundamental requirement: seek regions of high probability while **avoiding random walk behavior**

Hybrid Monte Carlo: We introduce a fictitious physical system where model parameters correspond to positions q; the position variables are augmented with momentum variables p to help avoid random walks.

We simulate the dynamical system in such a way that the distributions of positions, $p(q)$, corresponds to the desired distribution $p(x)$.
For estimation purposes we ignore the momentum variables p.

Dynamical system

In the physical system, positions q correspond to random variables x and are augmented by momentum variables p:

$$p(p,q) \propto \exp(-H(q,p)) \qquad H(p,q) = E(q) + K(p)$$
$$E(q) = -\log(p(q)) \qquad K(p) = \tfrac{1}{2}\textstyle\sum_i p_i^2/m_i$$

The physical system evolves at constant energy H according to Hamiltonian dynamics:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} = \frac{p_i}{m_i} \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} = -\frac{\partial E}{\partial q_i}.$$

Dynamical system: leapfrog simulation

The system is simulated approximately via a series of L discrete leapfrog steps of size ϵ:

$$\begin{aligned}
\hat{p}_i(t+\frac{\epsilon}{2}) &= \hat{p}_i(t) - \frac{\epsilon}{2}\frac{\partial E(\hat{q}(t))}{\partial q_i} \\
\hat{q}_i(t+\epsilon) &= \hat{q}_i(t) + \epsilon\frac{\hat{p}_i(t+\frac{\epsilon}{2})}{m_i} \\
\hat{p}_i(t+\epsilon) &= \hat{p}_i(t+\frac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial E(\hat{q}(t+\epsilon))}{\partial q_i}
\end{aligned}$$

Properties of the dynamical system

Hamiltonian dynamics has the following important properties:

1) preserves total energy, H,
2) is reversible in time
3) preserves phase space volumes (Liouville's theorem)

The leapfrog discretization only approximately preserves the total energy H, and

1) is reversible in time
2) preserves phase space volume

The dynamical system is simulated using the leapfrog discretization and the new state is used as a proposal in the Metropolis algorithm to eliminate the bias caused by the leapfrog approximation

Stochastic dynamics

Changes in the total energy, H, are introduced by interleaving the deterministic leapfrog transitions with stochastic updates of the momentum variables.

Since the distribution of the momenta are independent Gaussian, this is easily done using Gibbs sampling with persistence:

$$p_i = \alpha p_i + \sqrt{1 - \alpha^2}\varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, 1)$ and the persistence $0 \leq \alpha < 1$.

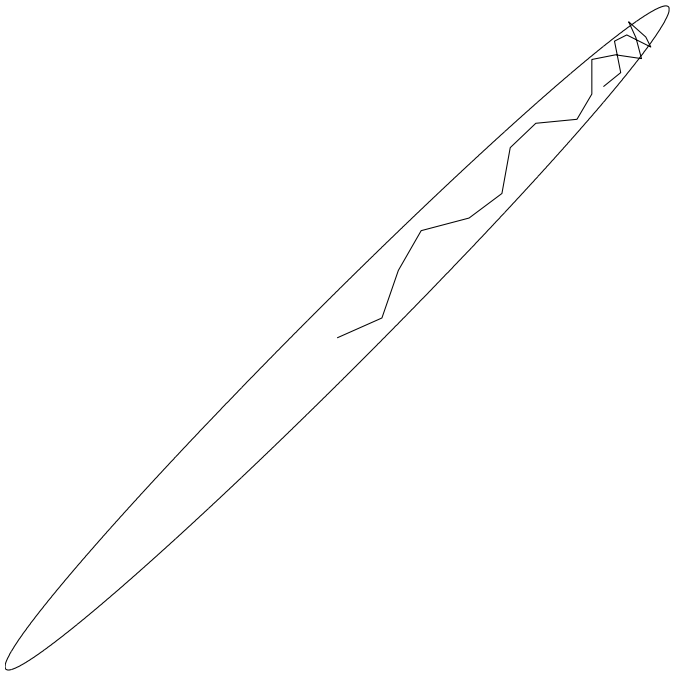
Persistence is added to further suppress random walks

Hybrid Monte Carlo algorithm

1) A new state is proposed by deterministically simulating a trajectory with L discrete steps from (q, p) to (q^*, p^*). The new state is accepted by the Metropolis algorithm with probability:

$$\min(1, \exp(-(H(p^*, q^*) - H(p, q)))),$$

otherwise the state remains the same and momentum is negated.
2) Stochastically update the momenta using Gibbs sampling with persistence



Example: $L = 20$ leapfrog iterations when sampling from bivariate Gaussian

An example: Bayesian inference in neural network models

Training set: $D = \{(x^{(c)}, y^{(c)})\},\ c = 1 \ldots n$.
Single hidden layer net (with direct connections from inputs to output):

$$f(x) = \sum_i W_i^{ho} h_i(x) + \sum_j W_j^{io} x_j + b$$

$$h_i(x) = \tanh(\sum_j W_{ji}^{ih} x_j + b_j)$$

model parameters (W's and b's) are called θ.
Likelihood (really $p(y^{(1)}, \ldots, y^{(n)} | x^{(1)}, \ldots, x^{(n)}, \theta)$):

$$p(D|\theta) \propto \prod_{c=1}^{n} \exp(-(f(x^{(c)}) - y^{(c)})^2 / 2\sigma^2).$$

The maximum likelihood approach minimizes the cost function $-\log(p(D|\theta))$.
Task: predict $y^{(n+1)}$ given a new input $x^{(n+1)}$ and the training examples D.

Using Bayes' rule

Bayes' rule: posterior $\propto$ likelihood $\times$ prior.
Using a hierarchical prior, which is specified in terms of "hyperparameters" γ:

$$p(\theta, \gamma|D) \propto p(D|\theta)p(\theta|\gamma)p(\gamma)$$

The predictive distribution is obtained by integration:

$$p(y^{(n+1)}|x^{(n+1)}, D) = \int p(y^{(n+1)}|x^{(n+1)}, \theta, \gamma)p(\theta, \gamma|D)d\theta d\gamma$$

Mean of predictive distribution (for squared error loss):

$$\hat{y}^{(n+1)} = \int f(x^{(n+1)}, \theta)p(\theta, \gamma|D)d\theta d\gamma$$

Computational task: do the integrals over the posterior distribution.