Probabilistic & Unsupervised Learning

Sampling Methods

Yee Whye Teh

`ywteh@gatsby.ucl.ac.uk`

Gatsby Computational Neuroscience Unit
University College London

Term 1, Autumn 2009

Integrals in Statistical Modelling

- **Parameter estimation**

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \int P(\mathcal{Y}|\theta) P(\mathcal{X}|\mathcal{Y},\theta)\, d\mathcal{Y}$$

(or using EM)

$$\theta^{\text{new}} = \underset{\theta}{\text{argmax}} \int P(\mathcal{Y}|\mathcal{X},\theta^{\text{old}}) \log P(\mathcal{X},\mathcal{Y}|\theta)\, d\mathcal{Y}$$

- **Prediction**

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

- **Model selection or weighting** (by marginal likelihood)

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

These integrals are often intractable:

- **Analytic intractability**: integrals may not have closed form in non-linear, non-Gaussian models ⇒ numerical integration.
- **Computational intractability**: Numerical integral (or sum if $\mathcal{Y}$ or θ are discrete) may be exponential in data or model size.

Examples of Intractability

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

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \int p(\theta) \prod_{i=1}^{N} \sum_{s_i} p(\mathbf{x}_i | s_i, \theta) p(s_i|\theta) \; d\theta$$

$$= \sum_{s_1} \sum_{s_2} \cdots \sum_{s_N} \int p(\theta) \prod_{i=1}^{N} p(\mathbf{x}_i | s_i, \theta) p(s_i|\theta) \; d\theta$$

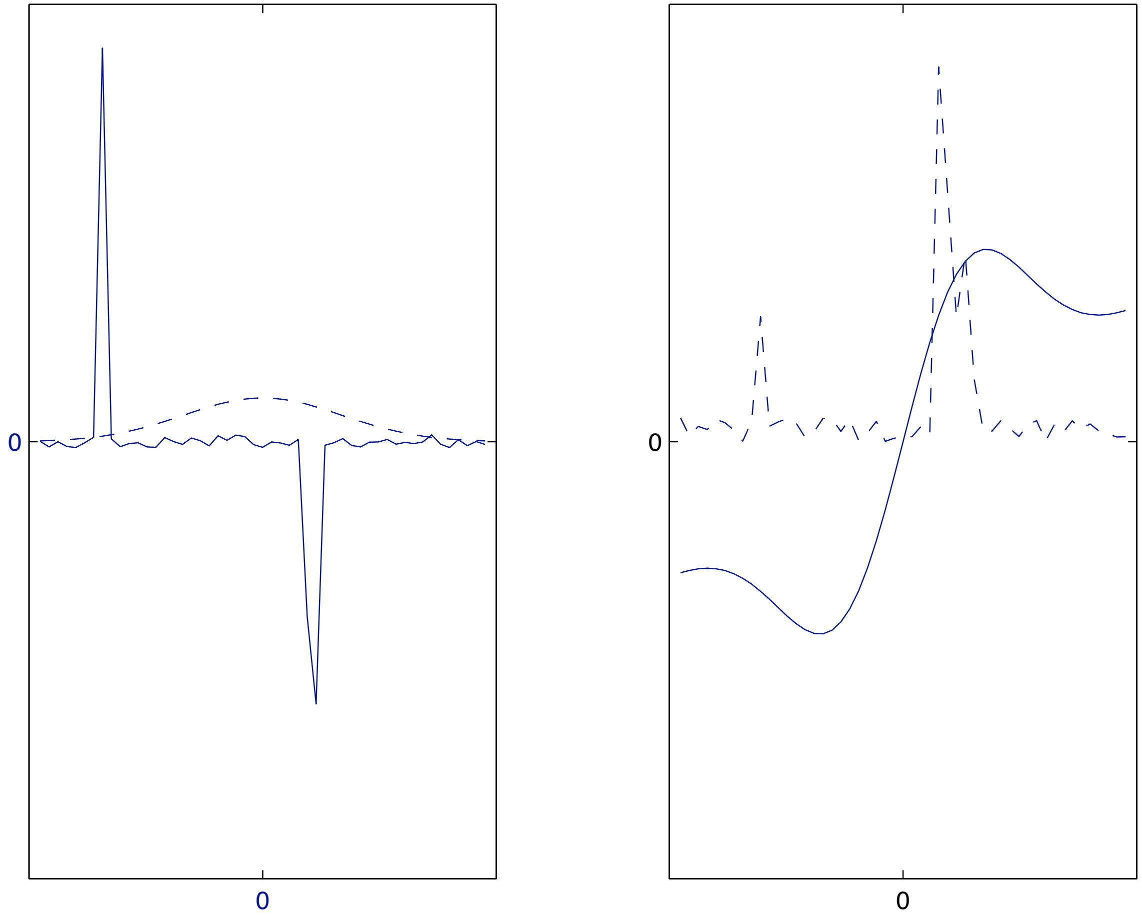
- Computing the conditional probability of a variable in a very large multiply connected directed graphical model:

$$p(X_i | X_j = a) = \sum_{\text{all settings of } X_{\backslash\{i,j\}}} p(X_i, X_{\backslash\{i,j\}}, X_j = a) / p(X_j = a)$$

- Computing the hidden state distribution in a general nonlinear dynamical system

$$p(\mathbf{y}_t | \mathbf{x}_1, \ldots, \mathbf{x}_T) \propto \int p(\mathbf{y}_t | \mathbf{y}_{t-1}) p(\mathbf{x}_t | \mathbf{y}_t) p(\mathbf{y}_{t-1} | \mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) p(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_t | \mathbf{y}_t) d\mathbf{y}_{t-1}$$

The integration problem



We commonly need to compute expected value 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: non-analytic integral (or sum) in very many dimensions

Sampling Methods

The basic idea of sampling methods is to approximate an intractable integral or sum using samples from some distribution.

Monte Carlo Methods:

- Simple Monte Carlo Sampling
- Rejection Sampling
- Importance Sampling
- . . .

Sequential Monte Carlo Methods:

- Particle Filtering
- . . .

Markov Chain Monte Carlo Methods:

- Gibbs Sampling
- Metropolis Algorithm
- Hybrid Monte Carlo
- . . .

Simple Monte Carlo Sampling

Idea: Sample from $p(x)$, average values of $F(x)$.

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 drawn from $p(x)$.

$\left[\text{For example: } x^{(t)} = G^{-1}(u^{(t)}) \text{ with } u \sim \text{Uniform}[0,1] \text{ and } G(x) = \int_{-\infty}^{x} p(x')dx'\right]$

Convergence to integral due to strong law of large numbers.

Analysis of Simple Monte Carlo

Attractions:

- unbiased:

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} F(x^{(t)})\right] = \mathbf{E}[F(x)]$$

- variance goes as $1/T$:

$$\begin{aligned}\mathbf{E}\left[\left(\frac{1}{T}\sum_{t} F(x^{(t)})\right)^2\right] - \mathbf{E}[F(x)]^2 &= \frac{1}{T^2}\left(T\mathbf{E}\left[F(x)^2\right] + (T^2 - T)\mathbf{E}[F(x)]^2\right) - \mathbf{E}[F(x)]^2 \\ &= \frac{1}{T}\left(\mathbf{E}[F(x)^2] - \mathbf{E}[F(x)]^2\right)\end{aligned}$$

Note: independent of dimension!

Problems:

- It may be difficult or impossible to obtain the samples directly from $p(x)$.
- Regions of high density $p(x)$ may not correspond to regions where $F(x)$ varies a lot (thus each evaluation might have very high variance).

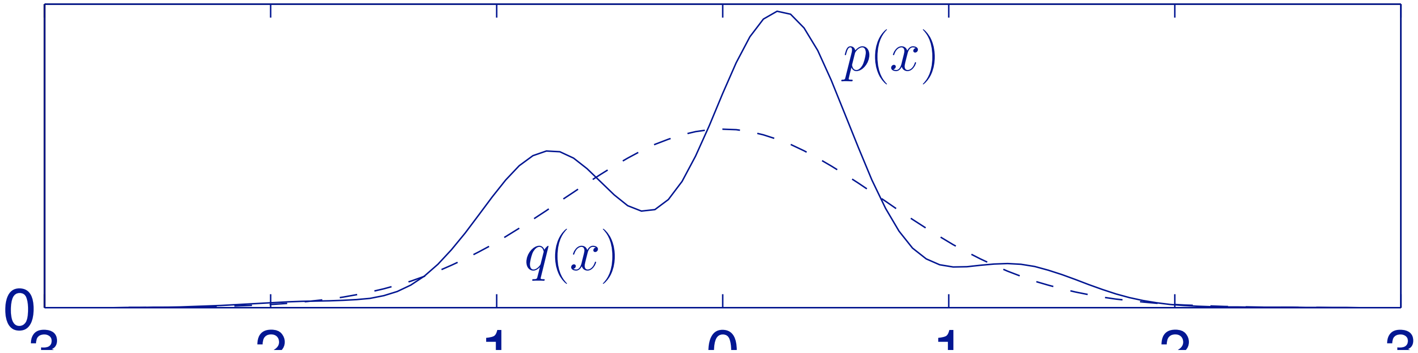
Importance Sampling

Idea: Sample from a proposal distribution $q(x)$ and weight those samples by $p(x)/q(x)$.

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



Analysis of Importance Sampling

Attraction:

- Unbiased: $\mathbf{E}_q[F(x)w(x)] = \int F(x)p(x)/q(x)q(x)dx = \mathbf{E}_p[F(x)]$.
- Variance could be smaller than simple Monte Carlo if

$$\mathbf{E}_q[(F(x)w(x))^2] - \mathbf{E}_q[F(x)w(x)]^2 \leq \mathbf{E}_p[F(x)^2] - \mathbf{E}_p[F(x)]^2$$

Optimal proposal is “posterior” $q(x) = p(x)F(x)/Z_q$ with variance 0—in fact estimate is constant: $F(x)w(x) = Z_q = \mathbf{E}_p[F(x)]$.

Problems:

- May be hard to construct $q(x)$ with small variance.
- Variance could be unbounded!

$$\mathbf{E}_q[w(x)] = \int q(x)w(x)dx = 1$$

The weights have variance $\mathbf{Var}[w(x)] = \mathbf{E}_q[w(x)^2] - 1$, with:

$$\mathbf{E}_q[(w(x)^2)] = \int \frac{p(x)^2}{q(x)^2}q(x)dx = \int \frac{p(x)^2}{q(x)}dx$$

e.g. $p(x) = \mathcal{N}(0,1), q(x) = \mathcal{N}(1,.1)$. When this happens, Monte Carlo average may be dominated by few samples; and none of the high weight samples may be found!

Analysis of Importance Sampling

Unnormalized distributions: say we only know $p(x), q(x)$ up to a constant,

$$p(x) = \tilde{p}(x)/Z_p \qquad\qquad q(x) = \tilde{q}(x)/Z_q$$

where Z_p, Z_q are unknown/too expensive to compute, but we can still sample from $q(x)$.

We can still apply importance sampling with the estimate:

$$\int F(x)p(x)dx \approx \frac{\sum_t F(x^{(t)})w(x^{(t)})}{\sum_t w(x^{(t)})} \qquad\qquad w(x) = \frac{\tilde{p}(x)}{\tilde{q}(x)}$$

But this estimate is only consistent (biased for finite T, converging to true value as $T \rightarrow \infty$).

Effective sample size of the sample may diagnose effectiveness of importance sampling. Popular estimate:

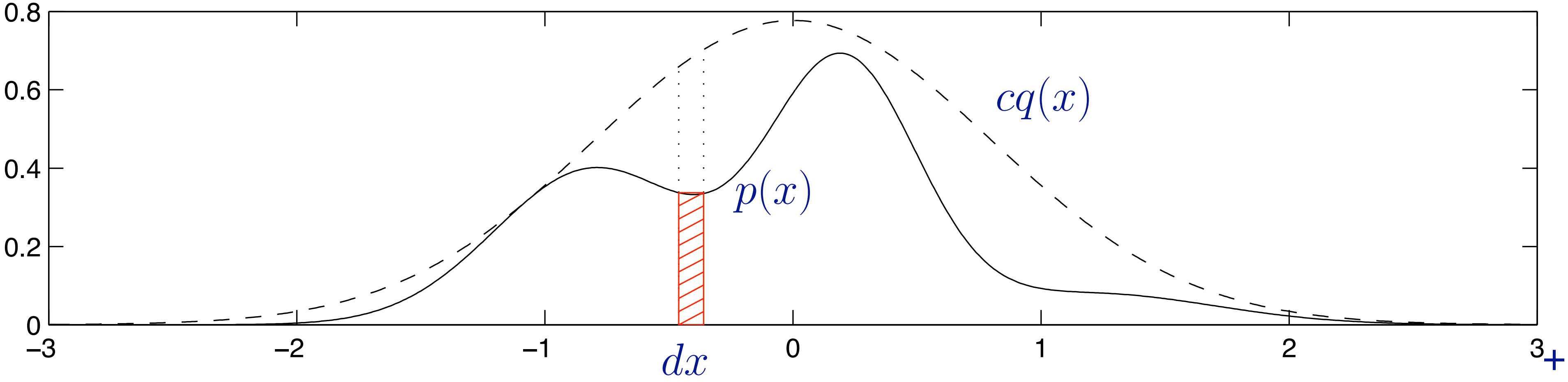
$$\left(1 + \textbf{Var}_{\text{sample}}\left[\frac{w(x)}{\textbf{E}_{\text{sample}}[w(x)]}\right]\right)^{-1} = \frac{\left(\sum_t w(x^{(t)})\right)^2}{\sum_t w(x^{(t)})^2}$$

Large effective sample size may give no indication of effectiveness (if none of high weight samples found, or if q places little mass where $F(x)$ is large).

Rejection Sampling

Idea: sample from an upper bound on $p(x)$, rejecting some samples.

- 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 as in simple Monte Carlo: $\frac{1}{T}\sum_{t=1}^{T} F(x^{(t)})$



If $y^* \sim$ Uniform$[0, cq(x^*)]$, we accept x^* if $y^* \leq p(x^*)$. Thus the probability of a point falling in the box $= q(x)dx * p(x)/cq(x) = p(x)/c$.

Proposal (x^*, y^*) is a point uniformly drawn from area under the $q(x)$ curve, accepted (x^*, y^*) from under $p(x)$ curve.

Rejection Sampling

Average acceptance probability is $1/c$.

Attraction:

- Unbiased; in fact accepted x^* is true sample from $p(x)$.
- Diagnostics easier than importance sampling: number of accepted samples is true sample size.

Problem:

- It may be difficult to find a $q(x)$ with a small $c \Rightarrow$ lots of wasted area.

  Examples:
  - Compute $p(X_i = b | X_j = a)$ in a directed graphical model: sample from $P(X)$, reject if $X_j \neq a$, averaging the indicator function $I(X_i = b)$
  - Compute **E**$(x^2 | x > 4)$ for $x \sim \mathcal{N}(0, 1)$

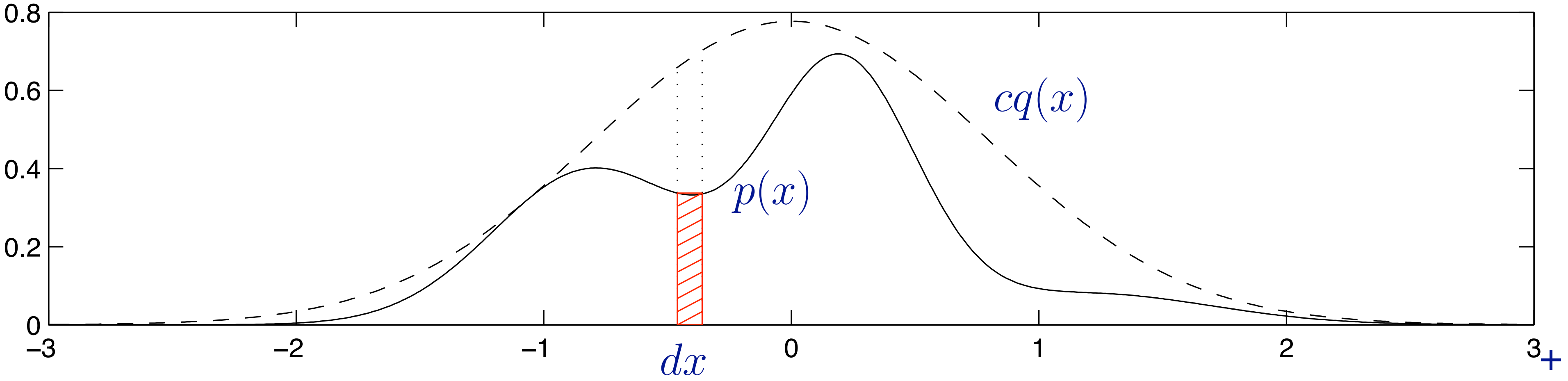
Unnormalized Distributions: say we only know $p(x), q(x)$ up to a constant,

$$p(x) = \tilde{p}(x)/Z_p \qquad\qquad q(x) = \tilde{q}(x)/Z_q$$

where Z_p, Z_q are unknown/too expensive to compute, but we can still sample from $q(x)$.

We can still apply rejection sampling if we know a c with $\tilde{p}(x) \leq c\tilde{q}(x)$. This is still unbiased!

Relationship between Importance and Rejection Sampling



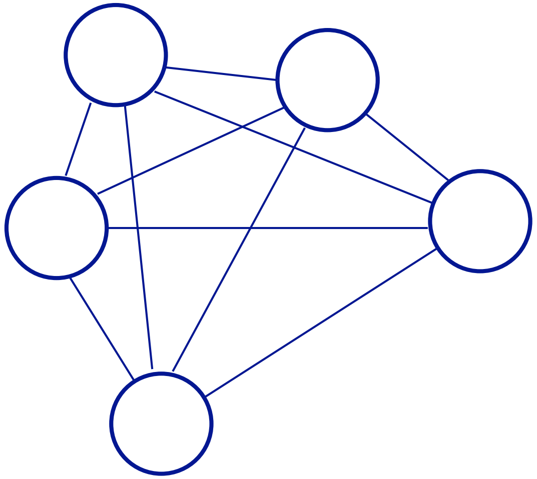
If we know a c making $q(x)$ an upper bound on $p(x)$, then importance weights are upper bounded:

$$\frac{p(x)}{q(x)} \leq c$$

So importance weights have finite variance and importance sampling is well-behaved.

Upper bound condition makes both rejection sampling work and importance sampling well-behaved.

Learning in Boltzmann Machines



$$\log p(\mathbf{s}^V \mathbf{s}^H | W, \mathbf{b}) = \sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i - \log Z$$

with $Z = \sum_{\mathbf{s}} e^{\sum_{ij} W_{ij} s_i s_j - \sum_i b_i s_i}$

Generalised (gradient M-step) EM requires parameter step

$$\Delta W_{ij} \propto \frac{\partial}{\partial W_{ij}} \left\langle \log p(\mathbf{s}^V \mathbf{s}^H | W, \mathbf{b}) \right\rangle_{p(\mathbf{s}^H | \mathbf{s}^V)}$$

Write $\langle\rangle_c$ (clamped) for expectations under $p(\mathbf{s}^H | \mathbf{s}^V)$. Then

$$\nabla W_{ij} = \langle s_i s_j \rangle_c - \langle s_i s_j \rangle_u$$

with $\langle\rangle_u$ (unclamped) an expectation under the current joint distribution $p(\mathbf{s}^H, \mathbf{s}^V)$.

Learning in Boltzmann Machines

How do we find the required expectations?

- **Junction tree** is generally intractable in all but the sparsest nets (triangulation of loops makes cliques grow very large).
- **Rejection and Importance sampling** require good proposal distributions, which are difficult to come by.
- **Loopy belief propagation** fails in nets with strong correlations.
- **Mean-field methods** are possible, but approximate (and pretty inaccurate).

What is easy is conditional sampling. Given settings of all nodes in the Markov blanket of s_i can easily sample s_i. This suggests an iterative sampling algorithm:

- Choose variable settings randomly (set any clamped nodes to clamped values).
- Cycle through (unclamped) s_i, choosing $s_i \sim p(s_i|\mathbf{s}_{\setminus i})$.

After enough samples, we might expect to reach the correct distribution.

This is an example of Gibbs Sampling. Also called the heat bath or Glauber dynamics.

Markov chain Monte Carlo (MCMC) methods

Assume we are interested in drawing samples from some desired distribution $p^*(x)$.

We define a Markov chain:

$$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \dots$$

where $x_0 \sim p_0(x)$ and $T(x \rightarrow x') = p(X_t = x' | X_{t-1} = x)$ is the Markov chain transition probability from x to x'.

Then the marginal distributions are $x_t \sim p_t(x)$ with the property that:

$$p_t(x') = \sum_x p_{t-1}(x) T(x \rightarrow x')$$

We say that $p^*(x)$ is an invariant/stationary/equilibrium distribution of the Markov chain defined by T iff:

$$p^*(x') = \sum_x p^*(x) T(x \rightarrow x') \qquad\qquad \forall x'$$

Markov chain Monte Carlo (MCMC) methods

We have a Markov chain $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \ldots$ where $x_0 \sim p_0(x)$, $x_1 \sim p_1(x)$, etc, with the property that:

$$p_t(x') = \sum_x p_{t-1}(x) T(x \rightarrow x')$$

where $T(x \rightarrow x')$ is the Markov chain transition probability from x to x'.

A useful condition that implies invariance of $p^*(x)$ is detailed balance:

$$p^*(x')T(x' \rightarrow x) = p^*(x)T(x \rightarrow x')$$

We wish to find ergodic Markov chains, which converge to a unique stationary distribution regardless of the initial conditions $p_0(x)$:

$$\lim_{t \rightarrow \infty} p_t(x) = p^*(x)$$

A sufficient condition for the Markov chain to be ergodic is that

$$T^k(x \rightarrow x') > 0 \text{ for all } x \text{ and } x' \text{ where } p^*(x') > 0 \text{ and some } k.$$

That is, if the equilibrium distribution gives non-zero probability to state x', then the Markov chain should be able to reach x' from any x after some finite number of steps, k.

Gibbs Sampling

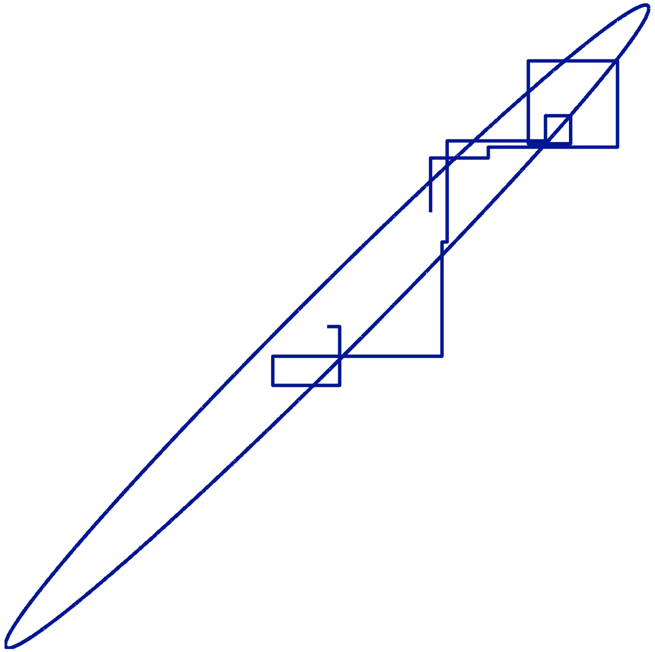
A method for sampling from a multivariate distribution, $p(\mathbf{x})$
Idea: sample from the conditional of each variable given the settings of the other variables.

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

$$p(x_i|\mathbf{x}_{\setminus i}) = p(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n)$$

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 $p(\mathbf{x})$.

Detailed balance for Gibbs sampling

We can show that Gibbs sampling has the right stationary distribution $p(\mathbf{x})$ by showing that the detailed balance condition is met.

The transition probabilities are given by:

$$T(\mathbf{x} \rightarrow \mathbf{x}') = \pi_i p(x_i'|\mathbf{x}_{\backslash i})$$

where π_i is the probability of choosing to update the ith variable (to handle rotation updates instead of random ones, we need to consider transitions due to one full sweep).

Then we have:

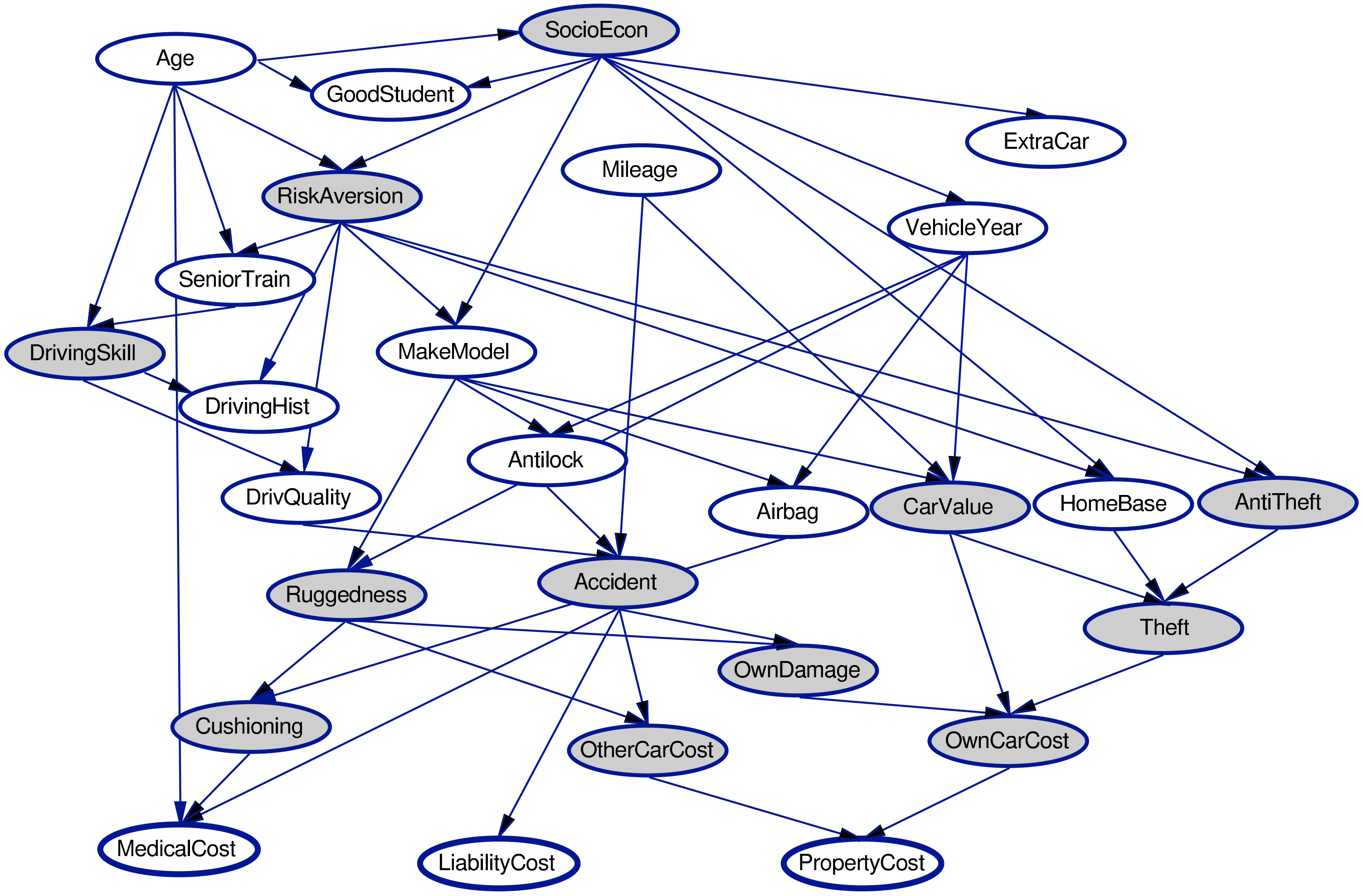
$$T(\mathbf{x} \rightarrow \mathbf{x}')p(\mathbf{x}) = \pi_i p(x_i'|\mathbf{x}_{\backslash i}) \underbrace{p(x_i|\mathbf{x}_{\backslash i})p(\mathbf{x}_{\backslash i})}_{p(\mathbf{x})}$$

and

$$T(\mathbf{x}' \rightarrow \mathbf{x})p(\mathbf{x}') = \pi_i p(x_i|\mathbf{x}'_{\backslash i}) \underbrace{p(x_i'|\mathbf{x}'_{\backslash i})p(\mathbf{x}'_{\backslash i})}_{p(\mathbf{x}')}$$

But $\mathbf{x}'_{\backslash i} = \mathbf{x}_{\backslash i}$ so detailed balance holds.

Gibbs Sampling in Graphical Models



Initialize all variables to some settings. Sample each variable conditional on other variables (equivalently, conditional on its Markov blanket).

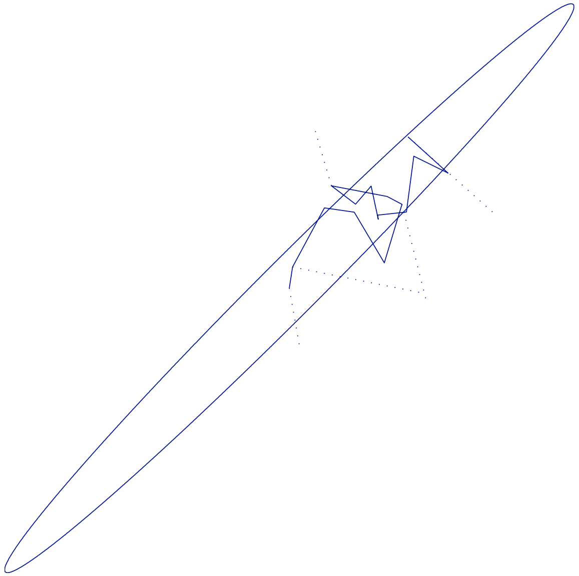
The BUGS software implements this algorithm for very general probabilistic models (but not too big ones).

The Metropolis-Hastings algorithm

Gibbs sampling can be slow ($p(x_i)$ may be well determined by $\mathbf{x}_{\backslash i}$), and conditionals may be intractable. Global transition might be better.

Idea: Propose a change to current state; accept or reject.
(A kind of rejection sampling)

Each step: Starting from the current state **x**,



1. Propose a new state $\mathbf{x}'$ using a proposal distribution
   $$S(\mathbf{x}'|\mathbf{x}) = S(\mathbf{x} \to \mathbf{x}').$$
2. Accept the new state with probability:
   $$\min\left(1, p(\mathbf{x}')S(\mathbf{x}' \to \mathbf{x})/p(\mathbf{x})S(\mathbf{x} \to \mathbf{x}')\right);$$
3. Otherwise **retain the old state**.

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

- Metropolis algorithm was symmetric $S(\mathbf{x}'|\mathbf{x}) = S(\mathbf{x}|\mathbf{x}')$. Hastings generalised.
- **Local** (changing one or few x_i's) vs **global** (changing all **x**) proposal distributions.
- Efficiency dictated by balancing between high acceptance rate and large step size.
- May adapt $S(\mathbf{x} \to \mathbf{x}')$ to balance these, but stationarity only holds once S is fixed.
- Note, we need only to compute ratios of probabilities (no normalizing constants).

Detailed balance for Metropolis-Hastings

The transition kernel is:

$$T(\mathbf{x} \rightarrow \mathbf{x}') = S(\mathbf{x} \rightarrow \mathbf{x}') \min \left(1, \frac{p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x})}{p(\mathbf{x})S(\mathbf{x} \rightarrow \mathbf{x}')}\right)$$

with $T(\mathbf{x} \rightarrow \mathbf{x})$ the expected rejection probability.

Without loss of generality we assume $p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x}) \leq p(\mathbf{x})S(\mathbf{x} \rightarrow \mathbf{x}')$.
Then

$$\begin{aligned} p(\mathbf{x})T(\mathbf{x} \rightarrow \mathbf{x}') &= p(\mathbf{x})S(\mathbf{x} \rightarrow \mathbf{x}') \cdot \frac{p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x})}{p(\mathbf{x})S(\mathbf{x} \rightarrow \mathbf{x}')} \\ &= p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x}) \end{aligned}$$

and

$$\begin{aligned} p(\mathbf{x}')T(\mathbf{x}' \rightarrow \mathbf{x}) &= p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x}) \cdot 1 \\ &= p(\mathbf{x}')S(\mathbf{x}' \rightarrow \mathbf{x}) \end{aligned}$$

Practical MCMC

Markov chain theory guarantees that

$$\frac{1}{T}\sum_{t=1}^{T} F(x_t) \to \mathbf{E}[F(x)] \text{ as } T \to \infty.$$

But given finite computational resources we have to compromise. . .

Convergence diagnosis is hard. Usually plot various useful statistics, e.g. log probability, clusters obtained, factor loadings, and eye-ball convergence.

Control runs: initial runs of the Markov chain used to set parameters like step size etc for good convergence. These are discarded.

Burn-in: discard first samples from Markov chain before convergence.

Collecting samples: usually run Markov chain for a number of iterations between collected samples to reduce dependence between samples.

Number of runs: for the same amount of computation, we can either run one long Markov chain (best chance of convergence), lots of short chains (wasted burn-ins, but chains are independent), or in between.

Practical MCMC

Multiple transition kernels: different transition kernels have different convergence properties, it is often a good idea to use multiple kernels, but cannot have the choice between kernels be dependent on the current state.

Integrated autocorrelation time: estimate of the amount of time before $F(x_t)$'s become independent (no guarantees, probably underestimates autocorrelation time). Assume wlog $\mathbf{E}[F(x)] = 0$.

$$\mathbf{Var}\left[\frac{1}{T}\sum_{t=1}^{T} F(x_t)\right] = \mathbf{E}\left[\left(\frac{1}{T}\sum_{t=1}^{T} F(x_t)\right)^2\right] = \frac{\mathbf{Var}[F(x)]}{T}\left(1 + 2\sum_{t=1}^{T-1}\left(1 - \frac{t}{T}\right)\frac{C_t}{C_0}\right)$$

where $C_t = \mathbf{E}[F(x_i)F(x_{i+t})]$ are the autocorrelation times. The integrated autocorrelation time, the factor within parentheses, is the amount of correlated samples we need in excess of true iid samples to achieve the same variance. As $T \to \infty$, this is:

$$1 + 2\sum_{t=1}^{\infty}\frac{C_t}{C_0}$$

Annealing

Very often, need to sample from unnormalised distribution:

$$p(\mathbf{x}) = \frac{1}{Z}e^{-E(x)}$$

MCMC sampling works well in this setting but may mix slowly. For Gibbs sampling, usually possible to normalise conditional).

Often useful to introduce temperature $1/\beta$:

$$p_\beta(\mathbf{x}) = \frac{1}{Z_\beta}e^{-\beta E(x)}$$

When $\beta \rightarrow 0$ (temperature $\rightarrow \infty$) all states are equally likely: easy to sample and mix.
As $\beta \rightarrow 1$, $p_\beta \rightarrow p$.

Simulated annealing: start chain with β small and gradually increase to 1. Can be used for optimisation (or for finding global mode) by taking $\beta \rightarrow \infty$.

Annealed importance sampling: use importance sampling to correct for fact that chain need not have converged at each β. Equivalently, use annealing to construct a good proposal for importance sampling.

Hybrid Monte Carlo: overview

The typical distance traveled by a random walk in n steps is proportional to $\sqrt{n}$. We want to seek regions of high probability while avoiding random walk behavior.

Assume that we wish to sample from $p(\mathbf{x})$ while avoiding random walk behaviour. If we can compute derivatives of $p(\mathbf{x})$ with respect to $\mathbf{x}$, this is *useful information* and we should be able to use it to draw samples better.

Hybrid Monte Carlo: We think of a fictitious physical system with a particle which has position $\mathbf{x}$ and momentum $\mathbf{v}$. We will design a sampler which avoids random walks in $\mathbf{x}$ by simulating a dynamical system.

We simulate the dynamical system in such a way that the marginal distribution of positions, $p(\mathbf{x})$, ignoring the momentum variables corresponds to the desired distribution.

Hybrid Monte Carlo: the dynamical system

In the physical system, positions $\mathbf{x}$ corresponding to random variables of interest are augmented by momentum variables $\mathbf{v}$:

$$\begin{array}{rr} p(\mathbf{x},\mathbf{v}) \propto \exp(-H(\mathbf{x},\mathbf{v})) & \qquad H(\mathbf{x},\mathbf{v}) = E(\mathbf{x}) + K(\mathbf{v}) \\ E(\mathbf{x}) = -\log p(\mathbf{x}) & K(\mathbf{v}) = \frac{1}{2}\sum_i v_i^2 \end{array}$$

Importantly, note that $\int p(\mathbf{x},\mathbf{v})d\mathbf{v} = p(\mathbf{x})$, the desired distribution and $p(\mathbf{v}) = N(0,I)$.
We think of $E(\mathbf{x})$ as the potential energy of being in state $\mathbf{x}$, and $K(\mathbf{v})$ as the kinetic energy associated with momentum $\mathbf{v}$. We assume “mass” $= 1$, so momentum $=$ velocity.

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

$$\frac{dx_i}{d\tau} = \frac{\partial H}{\partial v_i} = v_i \qquad\qquad \frac{dv_i}{d\tau} = -\frac{\partial H}{\partial x_i} = -\frac{\partial E}{\partial x_i}.$$

The first equation says derivative of position is velocity. The second equation says that the system accelerates in the direction that decreases potential energy.

Think of a ball rolling on a frictionless hilly surface.

Hybrid Monte Carlo: how to simulate the dynamical system

We can simulate the above differential equations by discretising time and running some difference equations on a computer. This introduces small (we hope) errors. (The errors we care about are errors which change the total energy—we will correct for these by occasionally rejecting moves that change the energy.)

A good way to simulate this is using leapfrog simulation. We take L discrete steps of size ϵ to simulate the system evolving for $L\epsilon$ time:

$$\begin{aligned}
\hat{v}_i(\tau + \tfrac{\epsilon}{2}) &= \hat{v}_i(\tau) - \frac{\epsilon}{2}\,\frac{\partial E(\hat{x}(\tau))}{\partial x_i} \\
\hat{x}_i(\tau + \epsilon) &= \hat{x}_i(\tau) + \epsilon \frac{\hat{v}_i(\tau + \tfrac{\epsilon}{2})}{m_i} \\
\hat{v}_i(\tau + \epsilon) &= \hat{v}_i(\tau + \tfrac{\epsilon}{2}) - \frac{\epsilon}{2}\,\frac{\partial E(\hat{x}(\tau + \epsilon))}{\partial x_i}
\end{aligned}$$

Hybrid Monte Carlo: 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 discretisation 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 discretisation and the new state is used as a proposal in the Metropolis algorithm to eliminate the bias caused by the leapfrog approximation

Hybrid Monte Carlo Algorithm

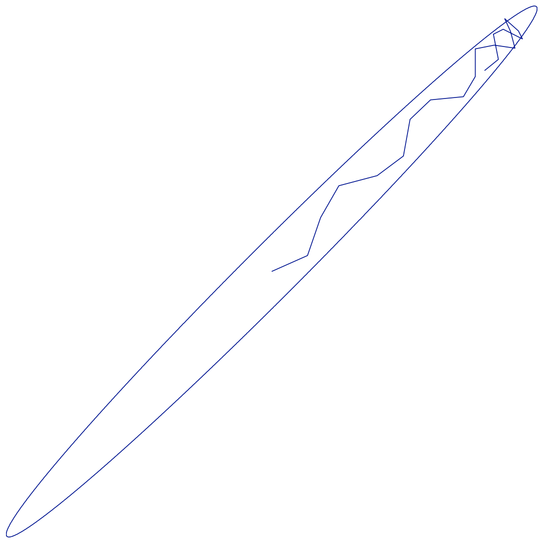
1) A new state is proposed by deterministically simulating a trajectory with L discrete steps from $(\mathbf{x}, \mathbf{v})$ to $(\mathbf{x}^*, \mathbf{v}^*)$. The new state $(\mathbf{x}^*, \mathbf{v}^*)$ is accepted with probability:

$$\min(1, \exp(-(H(\mathbf{v}^*, \mathbf{x}^*) - H(\mathbf{v}, \mathbf{x})))),$$

otherwise the state remains the same.

2) Stochastically update the momentum vector using Gibbs sampling

$$\mathbf{v} \sim p(\mathbf{v}|\mathbf{x}) = p(\mathbf{v}) = N(0, I)$$



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

Other Ideas in MCMC

- Auxiliary variables: introduce additional variables to improve convergence speed or computational cost.
- Rao-Blackwellisation or collapsing: integrating out some variables to lower variance or improve convergence.
- Exact sampling: yield **exact** samples from the equilibrium distribution of a Markov chain, making use of the idea of coupling—if two Markov chains use the same set of pseudo-random numbers, then even if they started in different states, once they transition to the same state they will stay in the same state.
- Slice sampling: to sample from $p(x)$, introduce auxiliary variable y uniform between $[0, p(x)]$ (area under $p(x)$ curve), and Gibbs sample x and y.
- Adaptive rejection sampling: during rejection sampling, if sample rejected use it to improve the proposal distribution.
- $\ldots$

Sampling - Importance Resampling (SIR)

Another (approximate) approach is to resample from the importance-weighted samples:

- Sample $\xi^{(s)} \sim q(x)$, and calculate importance weights $w^{(s)} = p(\xi^{(s)})/q(\xi^{(s)})$.
- Define $\tilde{q}(x) = \sum_{s=1}^{S} w^{(s)} \delta(x - \xi^{(s)}) \Big/ \sum_{s=1}^{S} w^{(s)}$.
- Resample $x^{(t)} \sim \tilde{q}(x)$.

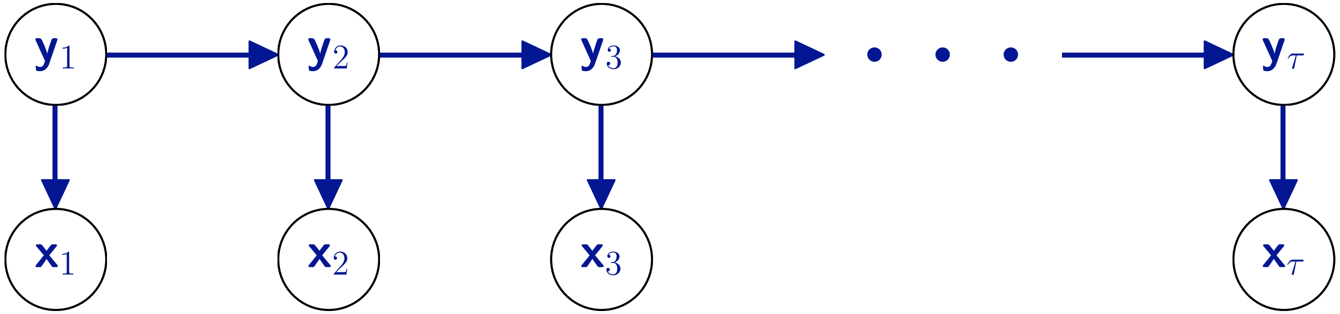
Then,

$$\begin{aligned} E_x[F(x)] &= \int dx \; F(x) \tilde{q}(x) \\ &= \int dx \; F(x) \frac{\sum_{s=1}^{S} w^{(s)} \delta(x - \xi^{(s)})}{\sum_{s=1}^{S} w^{(s)}} \\ &= \frac{\sum_{s=1}^{S} w^{(s)} F(\xi^{(s)})}{\sum_{s=1}^{S} w^{(s)}} \end{aligned}$$

but the expected value of this expression with respect to $\xi \sim q$ is only correct as $S \rightarrow \infty$.

By itself, SIR looks unattractive relative to IS due to this bias. But we sometimes really do need *samples* (*i.e.* a picture of the distribution) rather than just expectations. E.g., if propagating beliefs.

Particle Filtering



Suppose we want to compute $p(\mathbf{y}_t|\mathbf{x}_1 \ldots \mathbf{x}_t)$ in a non-linear ssm. We have

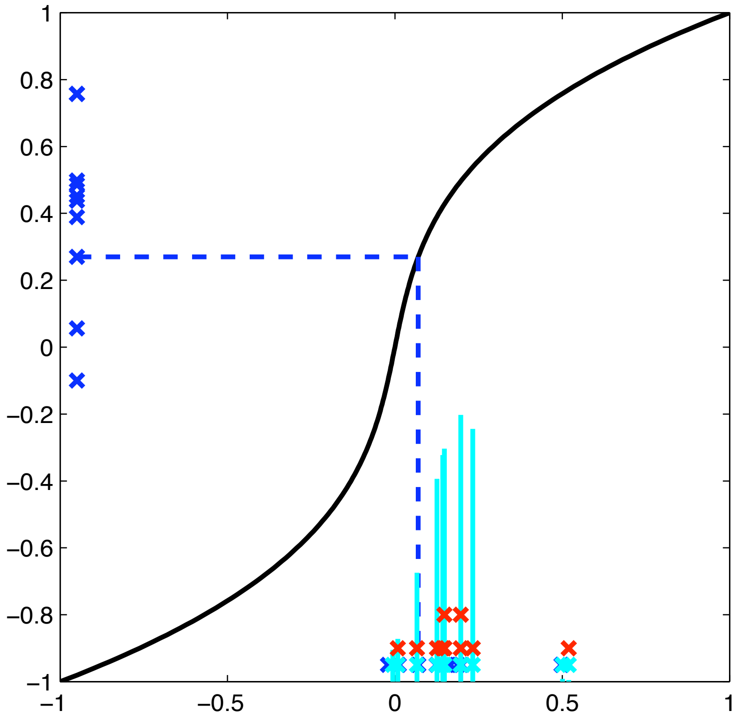
$$\begin{aligned} p(\mathbf{y}_t|\mathbf{x}_1 \ldots \mathbf{x}_t) &\propto \int p(\mathbf{y}_t \mathbf{y}_{t-1} \mathbf{x}_t|\mathbf{x}_1 \ldots \mathbf{x}_{t-1})\, d\mathbf{y}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{y}_t) p(\mathbf{y}_t|\mathbf{y}_{t-1}) p(\mathbf{y}_{t-1}|\mathbf{x}_1 \ldots \mathbf{x}_{t-1})\, d\mathbf{y}_{t-1} \end{aligned}$$

If we have samples $\mathbf{y}_{t-1}^{(s)} \sim p(\mathbf{y}_{t-1}|\mathbf{x}_1 \ldots \mathbf{x}_{t-1})$ we can recurse (approximately):

- draw $\mathbf{y}_t^{(s)} \sim p(\mathbf{y}_t|\mathbf{y}_{t-1}^{(s)})$
- calculate (unnormalised) weights $w_t^{(s)} = p(\mathbf{x}_t|\mathbf{y}_t^{(s)})$.
- resample $\mathbf{y}_t^{(s')} \sim \sum_{s=1}^{S} w_t^{(s)} \delta(\mathbf{y} - \mathbf{y}_t^{(s)}) \Big/ \sum_{s=1}^{S} w_t^{(s)}$

This is called Particle Filtering (this version, with $q = p(\mathbf{y}_t|\mathbf{y}_{t-1}^{(s)})$ is also called a “bootstrap filter” or “condensation” algorithm).

Particle Filtering



- Could avoid resampling by propagating weights. However variance in weights accumulates. Resampling helps eliminate unlikely particles.
- Can trigger resamples conditioned on variance – "stratified resampling".
- Can use better proposal (q) distributions (including $p(\mathbf{y}_t|\mathbf{x}_t\mathbf{y}_{t-1}^{(s)})$ if available).
- Particle *smoothing* is possible, but often inaccurate. Difficult to create a good proposal.
- EM learning is not easy because of smoothing problems and also obtaining joint on $(\mathbf{y}_{t-1}, \mathbf{y}_t)$.
- Widely used in engineering tracking applications, where filtering is most appropriate.
- Many variants . . .

References

- Excellent introductions to MCMC: Radford Neal (1993) Probabilistic Inference Using Markov Chain Monte Carlo Methods, Tech report CRG-TR-93-1 Toronto Computer Science; and David MacKay, Introduction to Monte Carlo Methods.
- Radford Neal (1998) Annealed Importance Sampling, Tech report 9805 Toronto Statistics, and Statistics and Computing (2001) 11:125-139.
- Radford Neal (2003) Slice Sampling, 31:705-767.
- W. R. Gilks and P. Wild (1992) Adaptive Rejection Sampling for Gibbs Sampling, Applied Statistics 41:337-348.
- James G. Propp and David B. Wilson (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Structures and Algorithms, 9:223-252.
- A. Doucet, N. de Freitas and N. Gordon (2001) Sequential Monte Carlo Methods in Practice, Springer.
- P. Del Moral, A. Doucet, A. Jasra (2006) Sequential Monte Carlo Samplers, Journal of the Royal Statistical Society B, 68:411-436.
- P. Fearnhead (1998) Sequential Monte Carlo Methods in Filter Theory, Ph.D. Thesis, Merton College, University of Oxford.
- M. Isard and A. Blake (1996) Contour tracking by stochastic propagation of conditional density, European Conference on Computer Vision.