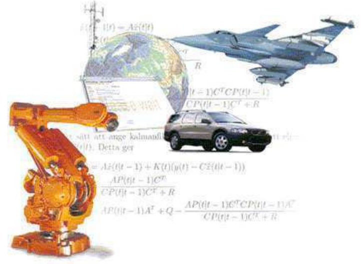
Machine Learning, Lecture 7
MCMC and Sampling Methods



Thomas Schön

Division of Automatic Control
Linköping University
Linköping, Sweden.

Email: schon@isy.liu.se,
Phone: 013 - 281373,
`www.control.isy.liu.se/~schon/`

About the Exam (I/II)

- If you have followed the course and completed the exercises you will not be surprised when you see the exam.
- You will learn new things during the exam.

Practicalities:

- Time frame: 2 days (48h), somewhere during week 34.
- Within 48 hours after you have collected the exam, you put your solutions in an envelope (seal it) and hand it in.

About the Exam (II/II)

As usual the graduate exam honor code applies. This means,

- The course books, other books and MATLAB are all allowed aids.
- Internet services such as email, web browsers and other communication with the surrounding world concerning the exam is NOT allowed.
- You are NOT allowed to actively search for the solutions in books, papers, the Internet or anywhere else.
- You are NOT allowed to talk to others (save for the responsible teachers) about the exam at all.
- If anything is unclear concerning what is allowed and not, just ask any of the responsible teachers.
- You are not allowed to look at exams from earlier version of the course (obviously hard in this course anyway...).

Outline of Lecture 7

- Summary of lecture 6
- Motivation for Monte Carlo methods
- Basic Sampling Methods
  - Transformation Methods
  - Rejection Sampling
  - Importance Sampling
- Markov Chain Monte Carlo (MCMC)
  - General Properties
  - Metropolis-Hastings Algorithm
  - Gibbs Sampling

(Chapter 11)

Summary of Lecture 6 (I/III)

In boosting we train a sequence of M models $y_m(x)$, where the error function used to train a certain model depends on the performance of the previous models. The models are then combined to produce the resulting classifier (for the two class problem) according to

$$Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$$

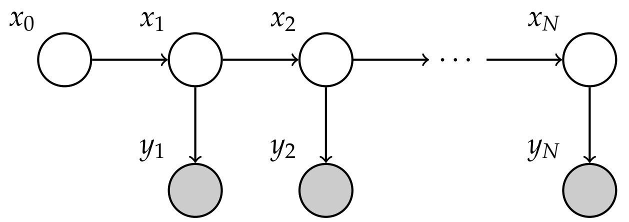
We saw that the AdaBoost algorithm can be interpreted as a sequential minimization of an exponential cost function.

Graphical Models: A graphical description of a probabilistic model where variables are represented by nodes and the relationships between variables correspond to edges.

Summary of Lecture 6 (II/III)

We started introducing some basic concepts for probabilistic graphical models $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ consisting of

1. a set of nodes $\mathcal{V}$ (a.k.a. vertices) representing the random variables and
2. a set of links $\mathcal{L}$ (a.k.a. edges or arcs) containing elements $(i, j) \in \mathcal{L}$ connecting a pair of nodes $(i, j) \in \mathcal{V}$ and thereby encoding the probabilistic relations between nodes.



Summary of Lecture 6 (III/III)

The set of parents to node j is defined as

$$\mathcal{P}(j) \triangleq \{i \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$$

The directed graph describes how the joint distribution $p(x)$ factors into a product of factors $p(x_i \mid x_{\mathcal{P}(i)})$ only depending on a subset of the variables,

$$p(x_{\mathcal{V}}) = \prod_{i \in \mathcal{V}} p(x_i \mid x_{\mathcal{P}(i)}).$$

Hence, for the state-space model on the previous slide, we have

$$p(X, Y) = p(x_0) \prod_{t=1}^{N} p(x_t \mid x_{t-1}) \prod_{t=1}^{N} p(y_t \mid x_t)$$

Motivation for Monte Carlo (I/III)

Probabilistic inference obviously depends on probability density functions $p(x)$.

We have two important problems with probabilistic inference:

1. **Computing integrals**

Examples: Bayesian Inference; Marginalization

$$\int f(x)p(x)dx \qquad p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)\,\mathrm{d}x} \qquad p(x_1) = \int p(x_1, x_2)\,\mathrm{d}x_2$$

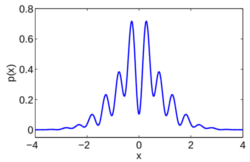
2. **Optimization**

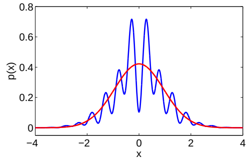
Examples: Maximum likelihood; Maximum a posteriori

$$\hat{x} = \arg\max_x p(x) \qquad \hat{x}_{\text{ML}} = \arg\max_x p(y|x) \qquad \hat{x}_{\text{MAP}} = \arg\max_x p(x|y)$$

Motivation for Monte Carlo (II/III)

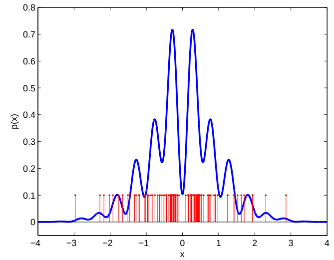
- The models of reality are becoming more and more complicated for us to be able to perform these operations exactly.
- The standard way of solving these problems is to make analytical approximations either in the model or in the solution.
  Examples:
  - Use conjugate priors to make analytical calculation of the integrals and optimization possible.
  - Variational approximations in the solutions.
- Analytical approximations change either the problem or the solution that we are trying to obtain and the effects are not always predictable.





Motivation for Monte Carlo (III/III)

- Another framework is to use numerical methods called Monte Carlo methods to solve these problems.
- With Monte Carlo, no sacrifice in the model or in the solution is made.
- The accuracy is limited only by our computational resources.



- Both integration and optimization can be done with these methods.
- If one can represent a complicated density with samples as $p(x) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x)$ any integral can be calculated by

$$\int f(x)p(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

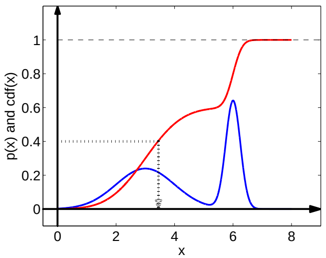
Basic Sampling Algorithms

- Most processing environments have built-in random number generators for the uniform distribution.
- Assume that we would like to generate samples from a general univariate density $p(x)$.The following scheme provides the way.

Generating samples from density $p(x)$

- Generate $u \sim \mathcal{U}(0,1)$
- Calculate $x^{(i)} = \mathrm{cdf}^{-1}(u)$

where $\mathrm{cdf}(x) = \int_{-\infty}^{x} p(x')\,\mathrm{d}x'$ is the cumulative distribution function of x.



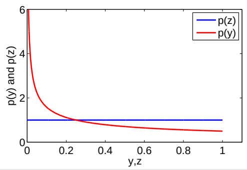
Transformation Method (I/II)

- Suppose we have a random variable $z \sim p_z(z)$ which we can obtain the samples from.
- If we transform the samples of z with an invertible function $f(\cdot)$ as $y = f(z)$, the density of the samples we obtain would be

$$p_y(y) = p_z(z)\left|\frac{dz}{dy}\right| = p_z(f^{-1}(y))\left|\frac{df^{-1}(y)}{dy}\right|$$

- The second term on the r.h.s. is absolutely necessary. Without it the r.h.s. might not even be a proper density.
- **Example:** Let $z \sim \mathcal{U}(0,1)$. Let $f(z) = z^2$. Then we have $f^{-1}(y) = \sqrt{y}$ for $0 \leq y \leq 1$.

$$p(y) = p_z(\sqrt{y})\left|-\frac{1}{2\sqrt{y}}\right| = \frac{1}{2\sqrt{y}}$$



Transformation Method (II/II)

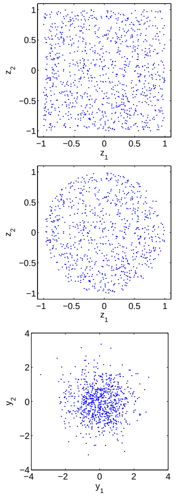
- For Gaussian random variable generation, the following transformation method is proposed.

Box-Muller Method

1. Generate $z_1, z_2 \sim \mathcal{U}(-1, 1)$.
2. If $r^2 \triangleq z_1^2 + z_2^2 > 1$, discard the samples and go to 1.
3. If $r^2 \leq 1$, calculate

$$y_1 = z_1 \sqrt{-\frac{2\ln r^2}{r^2}} \quad \text{and} \quad y_2 = z_2 \sqrt{-\frac{2\ln r^2}{r^2}}$$

for which $y_1, y_2 \sim \mathcal{N}(0, 1)$.



Random Number Generation - Perfect Sampling

Assumption: We have access to samples from the density we seek to estimate,

$$\hat{p}_N(x) = \sum_{i=1}^{N} \frac{1}{N} \delta\left(x - x^i\right)$$

We are seeking an estimate according to,

$$\hat{I}_N\left(g(x)\right) = \int g(x)\hat{p}_N(x)dx = \sum_{i=1}^{N} \frac{1}{N} g(x^i).$$

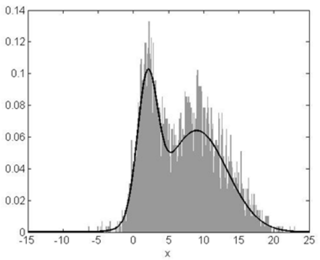
The strong law of large numbers $\lim_{N\to\infty} \hat{I}_N(g(x)) \xrightarrow{\text{a.s.}} I(g(x))$.
Central limit theorem

$$\lim_{N\to\infty} \frac{\sqrt{N}}{\sigma}\left(\hat{I}_N(g(x)) - I(g(x))\right) \xrightarrow{d} \mathcal{N}(0, 1)$$

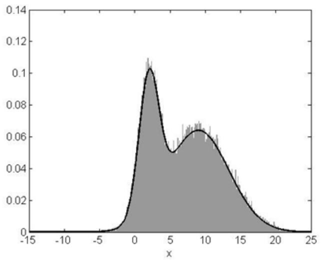
Perfect Sampling - Example and Problem

Consider sampling from the following Gaussian mixture

$$p(x) = 0.3\mathcal{N}\left(x \,|\, 2, 2\right) + 0.7\mathcal{N}\left(x \,|\, 9, 19\right)$$



5000 samples



50 000 samples

Obvious Problem: In general we are NOT able to sample from the density we are interested in!

Random Number Generation

Target density (p) - We seek samples distributed according to this density.

Proposal density (q) - This density is simple to generate samples from.

Acceptance probability (w) - Used to decide whether the sample is OK.

$$p(\tilde{x}) \propto w(\tilde{x})q(\tilde{x})$$

Three common algorithms based on this idea:

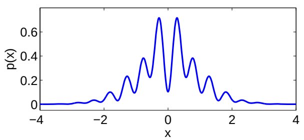
1. Rejection sampling
2. Importance sampling
3. Metropolis-Hastings

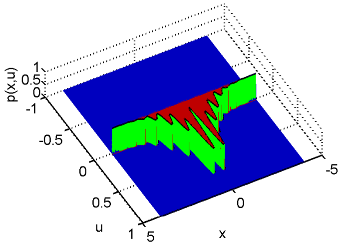
Rejection Sampling (I/IV)

- Suppose $p(x)$ is too complicated to sample directly from.
- Let us introduce a latent variable u.
- Consider the joint distribution

$$p(x,u) \triangleq p(u|x)p(x)$$

where we define
$p(u|x) \triangleq \mathcal{U}(u; 0, p(x))$



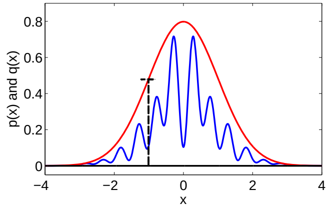


- Then

$$p(x,u) = \begin{cases} \frac{1}{p(x)}p(x) = 1, & \text{if } 0 \le u \le p(x) \\ 0, & \text{otherwise} \end{cases}$$

Rejection Sampling (II/IV)

- Hence, we must sample uniformly over the area under $p(x)$.
- For this purpose, we use a proposal density $q(x)$ that is easy to sample from such that $p(x) \le Kq(x)$ for all x.



Rejection Sampling

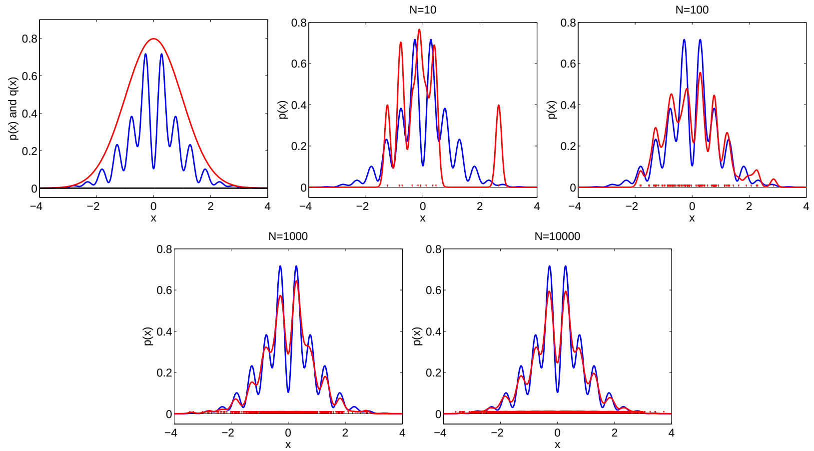
1. Sample $x^{(i)} \sim q(\cdot)$.
2. Sample $u \sim \mathcal{U}(0, Kq(x^{(i)}))$.
3. If $u \le p(x^{(i)})$ accept the sample $x^{(i)}$ as a valid sample from $p(\cdot)$. Go to 1.
4. Otherwise, discard $x^{(i)}$ and go to 1.

Rejection Sampling (III/IV)

- This procedure does not depend on the fact that $p(\cdot)$ is normalized. i.e., all $p(\cdot)$ terms can be replaced by an unnormalized version $\tilde{p}(\cdot)$ such that $p(x) = \frac{\tilde{p}(x)}{\int \tilde{p}(x')\,\mathrm{d}x'}$
- The procedure can be used with multivariate densities in the same way.
- The rejection rate is given by $1 - \frac{1}{K}$. This is the percentage of what we waste.
- Therefore, one must select K as small as possible while still satisfying $p(x) \le Kq(x)$ for all x.
- There are adaptive versions where one tries to obtain better proposals during the sampling process.
- Even the optimal K generally grows exponentially as the dimension increases.

Rejection Sampling (IV/IV)

Example: Kernel based density estimates from samples obtained with rejection sampling.



Importance Sampling (I/IV)

- So far, we have presented sampling approaches where each sample has equal contribution (importance) in the approximating particle sum as

$$p(x) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x)$$

which gave

$$\int f(x)p(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

- In importance sampling, one samples from a proposal density $x^{(i)} \sim q(\cdot)$ and uses a weighted approximation for $p(\cdot)$.

$$q(x) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x)$$

Importance Sampling (II/IV)

- We have the integral

$$\int f(x)p(x)\,\mathrm{d}x = \int f(x)\frac{p(x)}{q(x)}q(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} w^{(i)} f(x^{(i)})$$

where $w^{(i)} \triangleq \frac{p(x^{(i)})}{q(x^{(i)})}$.

- This approximation procedure is equivalent to approximating $p(\cdot)$ as

$$p(x) \approx \frac{1}{N}\sum_{i=1}^{N} w^{(i)}\delta_{x^{(i)}}(x)$$

- When the density $p(\cdot)$ is not normalized, one uses the approximation

$$p(x) \approx \frac{1}{N}\sum_{i=1}^{N} \bar{w}^{(i)}\delta_{x^{(i)}}(x) \quad \text{where} \quad \bar{w}^{(i)} = \frac{w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}$$

Importance Sampling (III/V)

Algorithm (Importance sampling)

1. *Generate* N *i.i.d. samples* $\{\tilde{x}^i\}_{i=1}^{N}$ *from the proposal density* $q(x)$ *and compute the importance weights*

$$\tilde{w}^i = p(\tilde{x}^i)/q(\tilde{x}^i), \qquad i = 1,\ldots,N.$$

2. *Form the acceptance probabilities by normalization,*

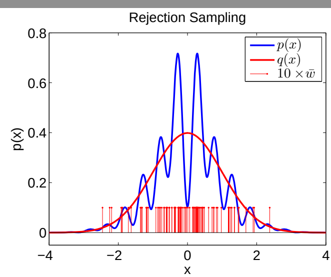
$$w^i = \tilde{w}^i / \sum_{j=1}^{N}\tilde{w}^j, \qquad i = 1,\ldots,N.$$

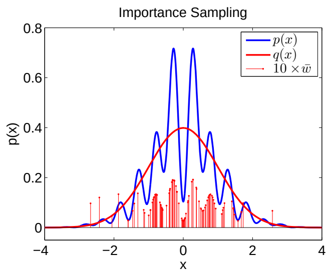
Results in the following approximation of the target density,

$$\tilde{p}(x) = \sum_{i=1}^{N} w^i \delta\left(x - \tilde{x}^i\right)$$

Importance Sampling (III/IV)

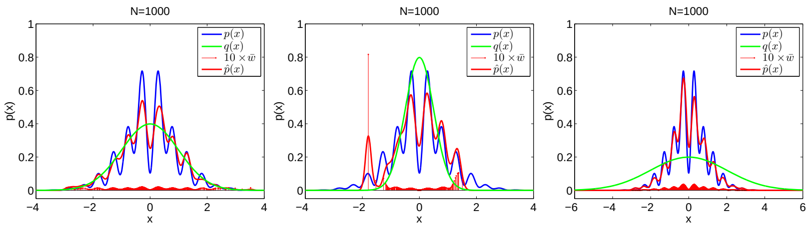
- In rejection and other types of sampling, only particles positions (denseness) carry information.
- In importance sampling, weights also carry important information.
- Note that, a large weight does not necessarily mean that the density value there is high. Particles density is still important.





Importance Sampling (IV/IV)

- Proposal selection is very important.
- Narrower proposals than the density can cause poor representation of the density in some parts of space.
- It is, in general, a good idea to choose wide proposals keeping in mind that a too wide proposal would result in too many samples with tiny weights which is a waste of computation.



Sampling Importance Resampling

$$\tilde{p}_N(x) \approx \sum_{i=1}^{N} w^i \delta(x - \tilde{x}^i)$$

We can now make use of resampling in order to generate an unweighted set of samples. This is done by drawing new samples with replacement according to,

$$\mathrm{P}\left(x^i = \tilde{x}^j\right) = w^j, \qquad j = 1, \dots, N,$$

resulting in the following unweighted approximation

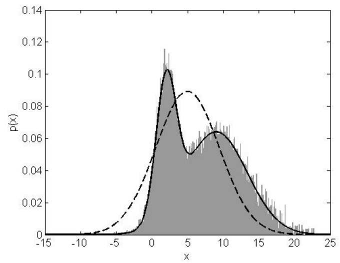
$$\hat{p}_N(x) = \sum_{i=1}^{N} \frac{1}{N} \delta(x - x^i)$$

Sampling Importance Resampling - Algorithm

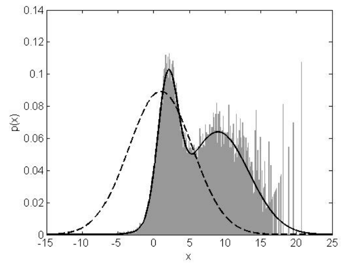
Algorithm (Sampling Importance Resampling (SIR))

1. *Generate* N *i.i.d. samples* $\{\tilde{x}^i\}_{i=1}^N$ *from the proposal density* $q(x)$ *and compute the importance weights*
$$\tilde{w}^i = p(\tilde{x}^i)/q(\tilde{x}^i), \qquad i = 1, \dots, N.$$
2. *Form the acceptance probabilities by normalization,*
$$w^i = \tilde{w}^i / \sum_{j=1}^{N} \tilde{w}^j, \qquad i = 1, \dots, N.$$
3. *For each* $i = 1, \dots, N$ *draw a new particle* x_t^i *with replacement (resample) according to,*
$$\mathrm{P}\left(x^i = \tilde{x}^j\right) = w^j, \qquad j = 1, \dots, N.$$

The Importance of a Good Importance Density



$q_1(x) = \mathcal{N}(5, 20)$



$q_2(x) = \mathcal{N}(1, 20)$

50 000 samples used in booth experiments.

Lesson learned: It is very important to be careful in selecting the importance density.

State Estimation

For a nonlinear state-space model

$$x_{t+1} \sim p(x_{t+1}|x_t)$$
$$y_t \sim p(y_t|x_t)$$

we can show that the filtering and the one-step ahead prediction densities are

$$p(x_t|y_{1:t}) = \frac{\overbrace{p(y_t|x_t)}\overbrace{p(x_t|y_{1:t-1})}}{p(y_t|y_{1:t-1})}$$
$$p(x_{t+1}|y_{1:t}) = \int \underbrace{p(x_{t+1}|x_t)}\underbrace{p(x_t|y_{1:t})}dx_t$$

The Particle Filter

Idea: Make use of SIR in order to find a first particle filter, i.e., an algorithm that provides approximations of

$$p(x_t \mid y_{1:t})$$

Recall, that we have (from the previous slide)

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

$$\underbrace{p(x_t|y_{1:t})}_{p(x_t)} \propto \underbrace{p(y_t|x_t)}_{a(x_t)}\underbrace{p(x_t|y_{1:t-1})}_{q(x_t)}$$

This implies that SIR can be used to produce estimates of $p(x_t \mid y_{1:t})$.

A First Particle Filter

Algorithm (A first particle filter)

1. *Initialize the particles,* $\{x_0^i\}_{i=1}^N \sim p(x_0)$ *and let* $t := 1$.
2. *Predict the particles by drawing* N *i.i.d. samples,*
$$\tilde{x}_t^i \sim p(x_t|x_{t-1}^i), \qquad i = 1, \ldots, N.$$
3. *Compute the importance weights* $\{\tilde{w}_t^i\}_{i=1}^N$,
$$\tilde{w}_t^i = p(y_t|\tilde{x}_t^i), \qquad i = 1, \ldots, N.$$
and normalize $w_t^i = \tilde{w}_t^i / \sum_{j=1}^N \tilde{w}_t^j$.
4. *For each* $i = 1, \ldots, N$ *draw a new particle* x_t^i *with replacement,*
$$P(x_t^i = \tilde{x}_t^j) = w_t^j, \qquad j = 1, \ldots, N.$$
5. *Set* $t := t + 1$ *and repeat from step* 2.

IS Example – Linear System Identification (I/III)

Consider the following linear scalar state-space model

$$x_{k+1} = \theta x_k + v_k, \qquad y_k = \frac{1}{2}x_k + e_k, \qquad \begin{pmatrix} v_k \\ e_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_e^2 \end{pmatrix}\right).$$

- The initial state: $x_0 \sim \mathcal{N}(x_0; \bar{x}_0, \Sigma_0)$.
- θ with prior distribution $\theta \sim \mathcal{N}(\theta; 0, \sigma_\theta^2)$
- The identification problem is now to determine the posterior $p(\theta|y_{0:N})$ using Importance Sampling.
- As usual, note the difference in notation compared to Bishop! The observations are denoted y and the latent variables are given by x.

IS Example – Linear System Identification (II/III)

$$x_{k+1} = \theta x_k + v_k, \qquad \begin{pmatrix} v_k \\ e_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_e^2 \end{pmatrix} \right).$$
$$y_k = \frac{1}{2} x_k + e_k,$$

- We have solved this problem with both EM and VB using the latent variables $x_{0:N} \triangleq \{x_0, \dots, x_N\}$.
- The main equation for the importance sampling for this example is

$$p(\theta | y_{0:N}) \propto p(y_{0:N}|\theta) p(\theta)$$

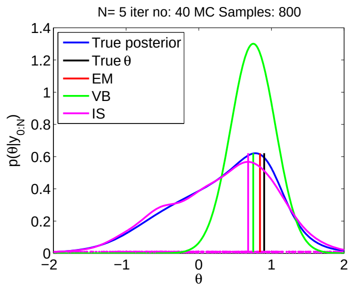
- The problem here is that we cannot normalize this density analytically since $p(y_{0:N}|\theta)$ is too complicated.
- We can still evaluate $p(y_{0:N}|\theta)$ for different values of θ.

IS Example – Linear System Identification (III/III)

- We would like to sample from $p(\theta|y_{0:N}) \propto p(y_{0:N}|\theta)p(\theta)$.
- Choose the proposal as the prior $q(\cdot) = p(\cdot)$.
- Sample $\theta^{(i)} \sim q(\cdot)$.
- Set the weights as

$$w^{(i)} = \frac{p(y_{0:N}|\theta^{(i)})p(\theta^{(i)})}{q(\theta^{(i)})} = p(y_{0:N}|\theta^{(i)})$$

- Normalize the weights $\bar{w}^{(i)} = \frac{w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}$.



- The likelihood $p(y_{0:N}|\theta^{(i)})$ required for the weights above is given by a Kalman filter as

$$p(y_{0:N}|\theta^{(i)}) = p(y_0) \prod_{i=2}^{N} p(y_i|y_{0:i-1}, \theta^{(i)}) \propto \prod_{i=2}^{N} \mathcal{N}(y_i; \hat{y}_{i|i-1}(\theta^{(i)}), S_{i|i-1}(\theta^{(i)}))$$

Markov Chain Monte Carlo

- Importance sampling is also bound to fail in high dimensions.
- This is due to the fact that, in high dimensions, the support of the density to be sampled from is only a tiny region in the overall space and for this case, it is very difficult to find a proposal without knowing the actual density.
- Markov Chain Monte Carlo (MCMC) is proposed to overcome this problem.
- Whereas in standard sampling methods, the samples are independent from each other, MCMC uses dependent samples.
- Due to the Markov property, each sample is dependent on the previous sample i.e., each $x^{(i)}$ depends on $x^{(i-1)}$.
- In general $x^{(i)}$ is generated as $x^{(i)} \sim q(x|x^{(i-1)})$ to sample from $p(x)$.

Markov Chain Monte Carlo

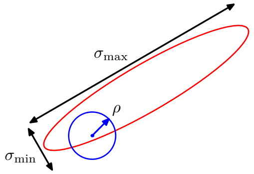
- Obviously, we want the overall behavior of the generated samples to be similar to those of $p(\cdot)$.
- MCMC methods provide a way to do this with an arbitrary proposal $q(\cdot|\cdot)$.

Metropolis Hastings Algorithm

- Generate an initial sample $x^{(1)} \sim q(\cdot)$.
- For i=2,...,
  - Sample $\bar{x} \sim q(x|x^{(i-1)})$.
  - Sample $u \sim \mathcal{U}(0,1)$.
  - Set the new sample $x^{(i)}$ as

$$x^{(i)} = \begin{cases} \bar{x}, & \text{if} \quad u < \min\left(1, \frac{p(\bar{x})}{p(x^{(i-1)})} \frac{q(x^{(i-1)}|\bar{x})}{q(\bar{x}|x^{(i-1)})}\right) \\ x^{(i-1)}, & \text{otherwise} \end{cases}.$$

M-H Example: Sampling from a Gaussian (I/II)



- Suppose the target density over $x \in \mathbb{R}^2$ is

$$p(x) = \mathcal{N}\left(x; \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}\right)$$

- Choose the proposal density $q(x|z)$ as

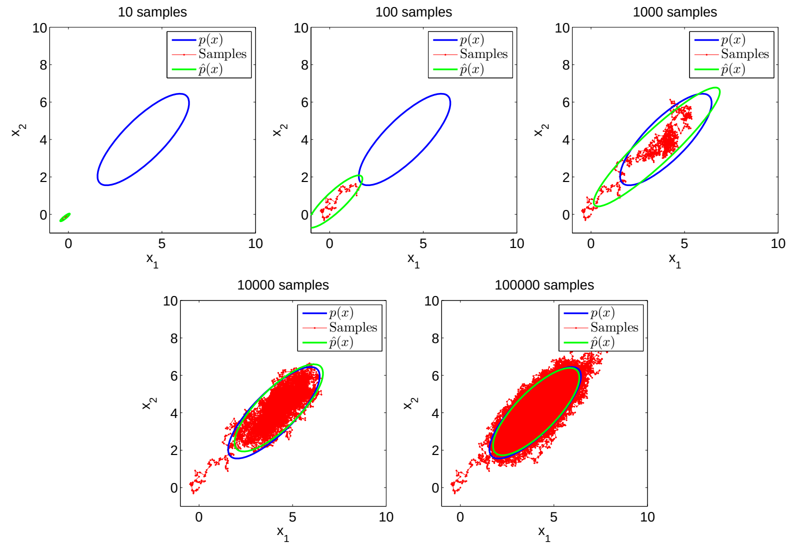
$$q(x|x^{(i-1)}) = \mathcal{N}\left(x; x^{(i-1)}, \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}\right)$$

- Noticing that $q(x|x^{(i-1)}) = q(x^{(i-1)}|x)$ for all x, we have

$$\min\left(1, \frac{p(\bar{x})}{p(x^{(i-1)})}\frac{q(x^{(i-1)}|\bar{x})}{q(\bar{x}|x^{(i-1)})}\right) = \min\left(1, \frac{p(\bar{x})}{p(x^{(i-1)})}\right)$$

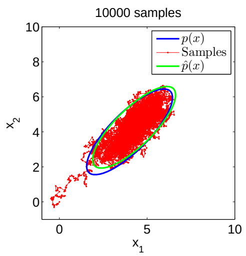
- This version of the Metropolis-Hastings algorithm is called the Metropolis algorithm.

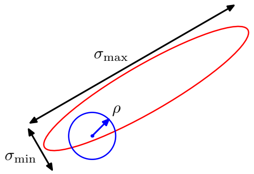
M-H Example: Sampling from a Gaussian (II/II)



M-H Some General Issues (I/II)

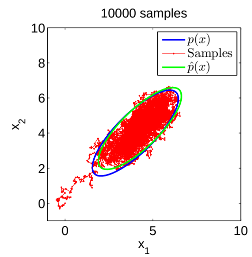
- M-H makes the samples converge to the samples of a stationary distribution which is the target distribution $p(\cdot)$.
- The time that passes before the samples starting to represent the target density is called burn-in period.
- We generally have to use only the samples obtained after the burn-in period.
- Diagnosing convergence to the target distribution with MCMC algorithms is still an active area of research.
- After the burn-in period is over, the Markov chain is said to be mixed.

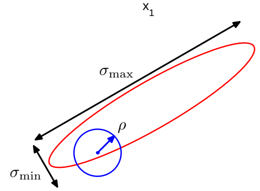




MH Some General Issues (II/II)

- Proposal selection is still an important problem.
- If the proposal is selected too narrow, then step-sizes get smaller and the burn-in period becomes longer.
- If the proposal is too wide, then the burn-in gets shorter, however, the acceptance rate is decreased significantly.





Gibbs Sampling

- Gibbs sampling is a special case of the Metropolis-Hastings algorithm where the proposal function is set to be the conditional distribution of the variables.
- It is especially useful when the dimension of the space to sample is very large e.g. images.
- Suppose, we are sampling in a two dimensional space $x = [x_1, x_2]^{\mathrm{T}}$. Then the Gibbs sampler works as follows.

Gibbs Sampler for 2D

- Sample $x^{(1)} \sim q(\cdot)$.
- For $i = 2, 3, \ldots,$
  - Sample $x_1^{(i)} \sim p(x_1 | x_2^{(i-1)})$.
  - Sample $x_2^{(i)} \sim p(x_2 | x_1^{(i)})$.
  - Set $x^{(i)} = [x_1^{(i)}, x_2^{(i)}]^{\mathrm{T}}$.

- Note that due to the special proposal, a Gibbs sampler does not have an accept-reject step as M-H.

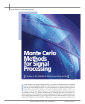
Optimization with MCMC

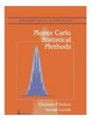
- Maximum a posteriori estimation requires

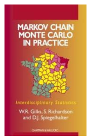
$$\hat{x}_{\text{MAP}} = \arg\max_x p(x|y) = \arg\max_x p(x, y)$$

- The densities, in general, have multiple modes and local optima.
- MCMC methods can be used to find global optimum of such densities.
- For this purpose, a time-varying target density is selected in an MCMC iteration.

References







- Spall, J.C., "Estimation via Markov chain Monte Carlo," *IEEE Control Systems Magazine*, vol.23, no.2, pp. 34- 45, Apr. 2003. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1188770&isnumber=26659
- Doucet, A.; Wang X., "Monte Carlo methods for signal processing: a review in the statistical signal processing context," *IEEE Signal Processing Magazine*, vol.22, no.6, pp. 152–170, Nov. 2005. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1550195&isnumber=33042
- Andrieu C.; De Freitas N.; Doucet A.; Jordan M. I., "An Introduction to MCMC for Machine Learning," *Machine Learning*, vol.50, pp. 5–43, 2003. http://www.cs.ubc.ca/~arnaud/andrieu_defreitas_doucet_jordan_intromontecarlomachinelearning.pdf
- Robert C. P.; Casella G., Monte Carlo Statistical Methods, Springer, 2004. Some density functions in this lecture came from this one!
- Gilks W. R.; Richardson S.; Spiegelhalter D. J., Markov Chain Monte Carlo, Chapman & Hall, 1996.
- Bishop C., Pattern Recognition and Machine Learning, Springer, 2006.
- MacKay D. J. C., Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003. (available online) http://www.inference.phy.cam.ac.uk/mackay/itila/

A Few Concepts to Summarize Lecture 11

Monte Carlo Methods: Approximate inference tools using the samples from the target densities.

Basic Sampling Methods: The sampling methods to obtain independent samples from target densities. Though quite powerful, these would give bad results with high dimensions.

MCMC: Monte Carlo methods which produce dependent samples but more robust in high dimensions.

Metropolis-Hastings Algorithm: The most well-known MCMC algorithm using arbitrary proposal densities.

Gibbs Sampler: A specific case of M-H algorithm which samples from conditionals iteratively and always accepts a new sample.