Bayesian learning and Monte Carlo methods

Bert Kappen

November 24, 2003

- Sampling methods
  - Uniform sampling
  - Importance and Rejection sampling
  - Metropolis method
  - Gibbs sampling
  - Hybrid Monte Carlo
- Illustration for perceptron learning

Perceptron

$$
\begin{aligned}
p(t=1|x,w) &= \sigma(\vec{w}\cdot\vec{x}) \\
\sigma(x) &= \frac{1}{1+\exp(-x)} \\
\vec{w}\cdot\vec{x} &= w_0 + w_1 x_1 + w_2 x_2
\end{aligned}
$$

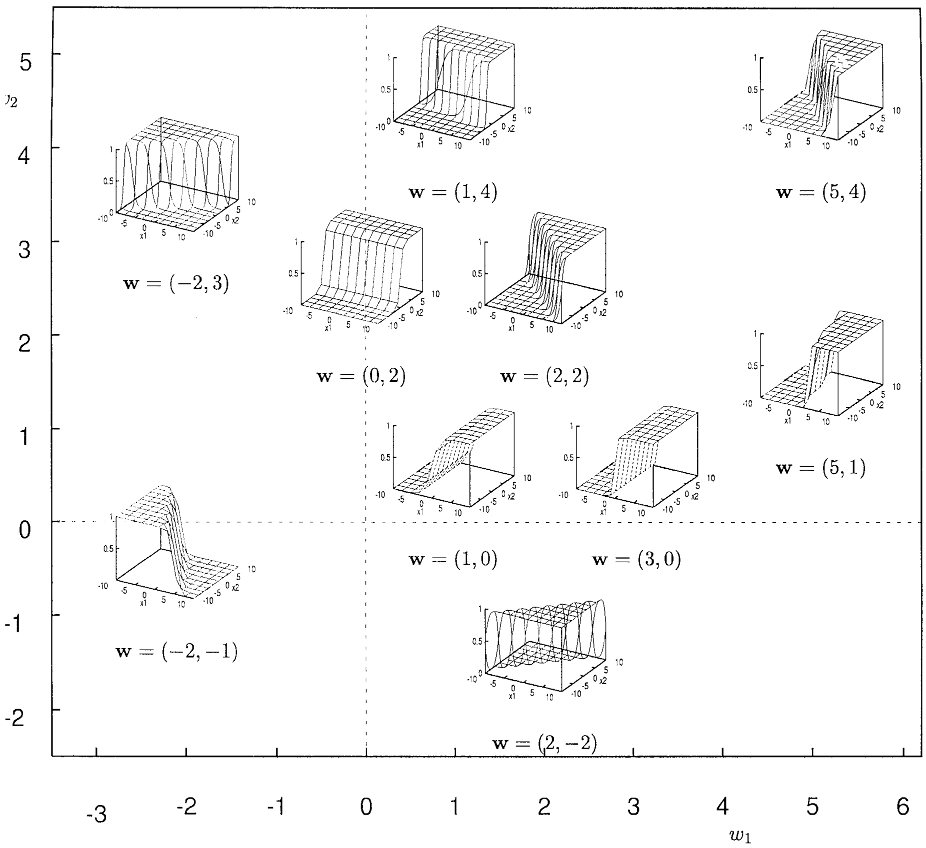


Figure 30.3. Weight space.

Bayesian learning

Data set: $\{x^\mu, t^\mu\}, \mu = 1, \ldots, P$

Probability of data point under the model: $p(t^\mu|x^\mu, w)$

Likelihood:

$$
\begin{aligned}
p(D|w) &= \prod_\mu p(t^\mu|x^\mu, w) = \exp(-G(w)) \\
G(w) &= -\sum_\mu \log(p(t^\mu|x^\mu, w))
\end{aligned}
$$

Prior:

$$p(w) = \frac{\exp(-\alpha E_w(w))}{Z_w(\alpha)}$$

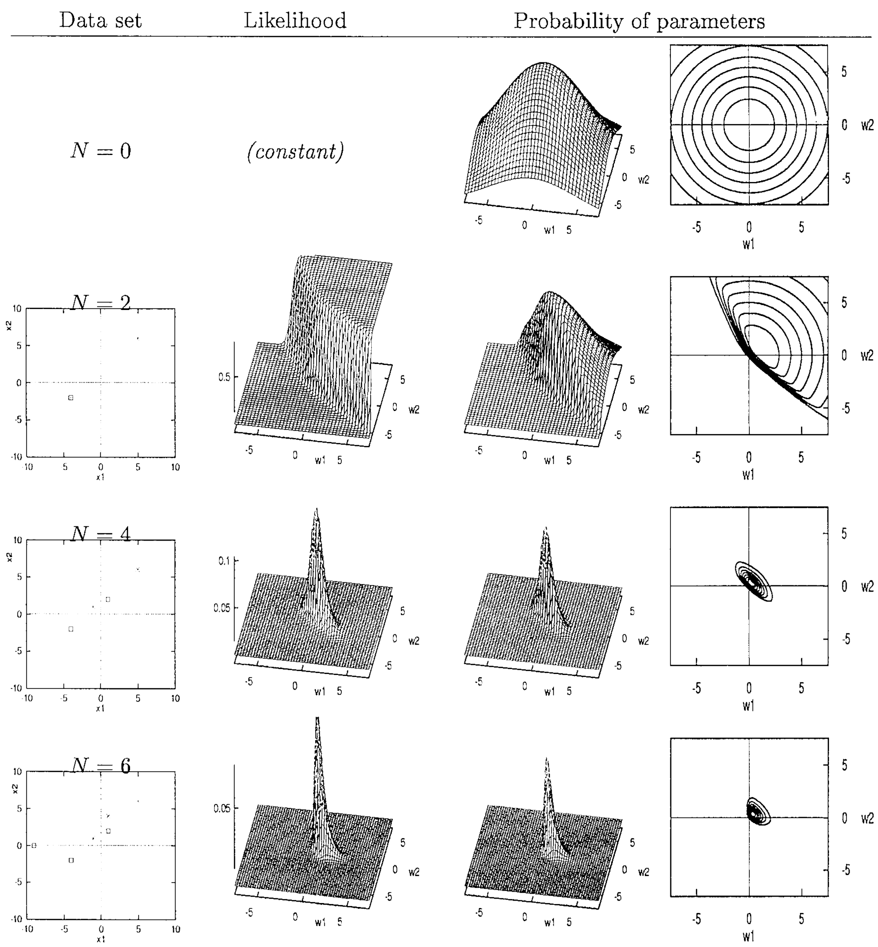
For instance,

$$E_w(w) = \sum_i w_i^2$$

makes solutions with small weights more probable.

Posterior:

$$
\begin{aligned}
p(w|D) &= \frac{p(D|w)p(w)}{p(D)} \propto \exp(-M(w)) \\
M(w) &= G(w) + \alpha E_w(w)
\end{aligned}
$$



ML versus Bayesian

Standard in neural network learning is to compute the maximum likelihood or maximum posterior solution. For new test point a

$$\begin{aligned} D &\rightarrow w_{\text{ml}} \\ p(t|a) &= p(t|a, w_{\text{ml}}) \end{aligned}$$

Bayesian approach requires integration over multiple solutions:

$$\begin{aligned} D &\rightarrow p(w|D) \\ p(t|a) &= \int dw p(w|D) p(t|a, w) = \langle p(t|a, w) \rangle_{p(w|D)} \end{aligned}$$

The problems

1. generate samples $\{x^r\}, r = 1, \ldots, R$ from $p(x)$

2. estimate

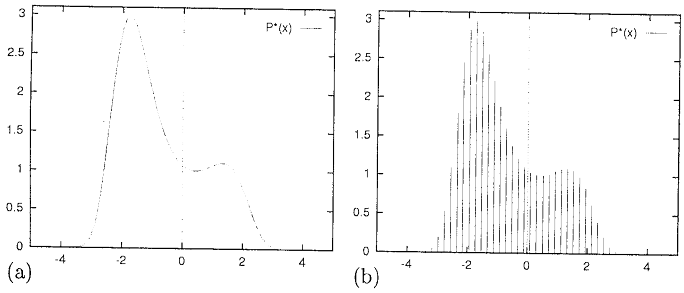
$$\Phi = \langle \phi(x) \rangle = \int d^n x p(x) \phi(x)$$

We focus on 1, since 1 solves 2:

$$\begin{aligned} \hat{\Phi} &= \frac{1}{R} \sum_r \phi(x^r) \\ \left\langle \hat{\Phi} \right\rangle &= \Phi \\ \mathrm{v}ar(\hat{\Phi}) &= \frac{\sigma^2}{R}, \quad \sigma^2 = \int d^n x p(x) (\phi(x) - \Phi)^2 \end{aligned}$$

when $\{x^r\}$ independent.

Uniform sampling



Uniform sampling:

$$\{x^r\}, r = 1, \ldots, R$$

requires a samples per dimension $\rightarrow a^n$ samples.

For learning or inference, number of parameters

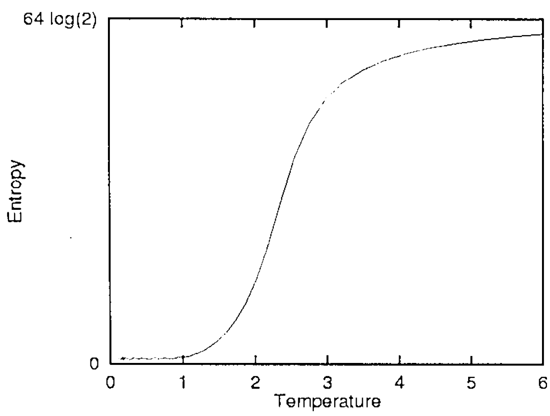
$$n = 100 - 1000.$$

Consider the Ising model

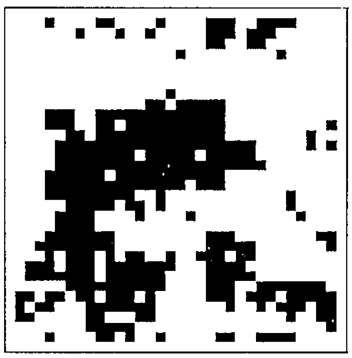
$$p(s|w) = \frac{\exp(\frac{1}{2}\sum_{ij} s_i s_j w_{ij})}{Z}$$

$s_i = \pm 1, i = 1, \ldots, n$. This distribution is intractable to compute, due to the normalisation

$$Z = \sum_{s_1} \ldots \sum_{s_n} \exp(\frac{1}{2}\sum_{ij} s_i s_j w_{ij})$$



(a)


(b)

Total number of states is 2^n, but most probability is concentrated in the so-called typical set T. Its size is approximately given by

$$|T| = 2^{H(p)}, \quad H(p) = -\sum_x p(x) \log(p(x))$$

Thus, the probability to hit the typical set is

$$p = \frac{2^H}{2^n}$$

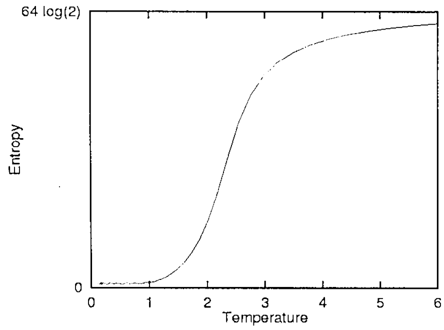
If one draws R samples uniform, the expected number of hits to the typical set is

$$R_{\text{hit}} = R\frac{2^H}{2^n}$$

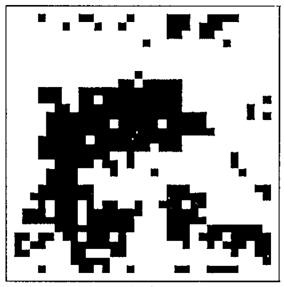
To ensure that $R_{\text{hit}} \gg 1$ one thus finds

$$R \gg 2^{n-H}$$

What is H?



(a)


(b)

- For high temperature (noise) $H \approx n$. Uniform sampling feasible
- For low temperature $H \ll n \Rightarrow R = \mathcal{O}(2^n)$

Uniform sampling only works for uniform distributions.

Better than uniform

Typically, $p(x)$ can be easily computed, up to a constant:

$$p(x) = \frac{p^*(x)}{Z}$$

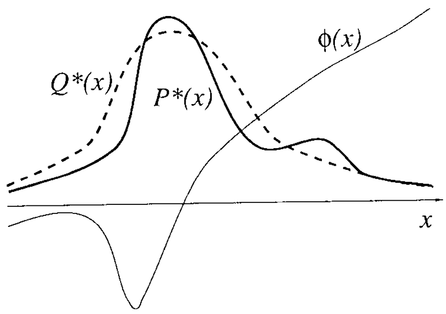
For instance

$$\begin{aligned}
p(w|D) &= \frac{p(D|w)p(w)}{p(D)} \propto \exp(-M(w)) \\
M(w) &= G(w) + \alpha E_w(w) \\
p(D) &= \int dw p(D|w)p(w)p(D)
\end{aligned}$$

Sample from another distribution $q(x)$ Often one can propose a sample density that is 1) better than uniform and 2) easy to sample from. For instance, a (spherical) Gaussian:

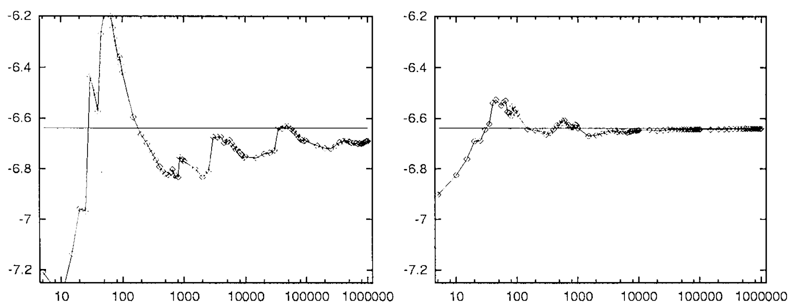
$$q(x) \propto \exp(-\sum_i x_i^2/2)$$

Importance sampling



Sample $\{x^r\}$ from $q(x)$ and compute

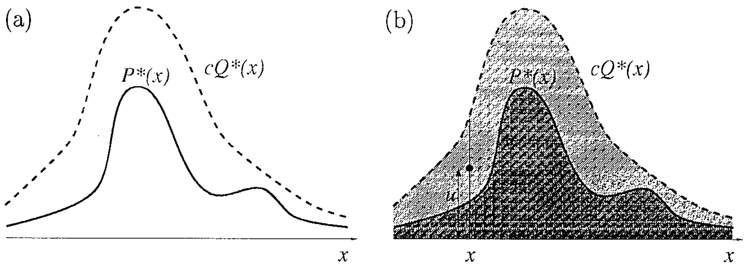
$$
\begin{aligned}
w_r &= \frac{p^*(x^r)}{q(x^r)} \\
\hat{\Phi} &= \frac{\sum_r w_r \phi(x^r)}{\sum_r w_r} \to \int dx \phi(x) p(x)
\end{aligned}
$$



(a) (b)

Rejection sampling

294



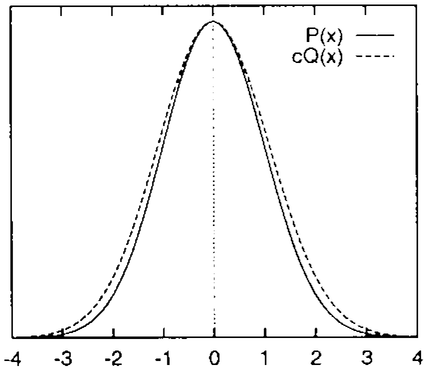
Choose, c such that for all x : $cq(x) > p^*(x)$

- generate x from $q(x)$
- generate u uniform from $[0, cq^*(x)]$
- if $u > p^*(x)$ reject x, otherwise accept x

This procedure samples $p^*(x)$ because (x, u) uniform from light grey area.

$$\hat{\Phi} \quad = \quad \sum_r \phi(x^r) \to \int dx \phi(x) p(x)$$

Rejection sampling in high dimensions



Let $p(x)$ and $q(x)$ be spherical Gaussians in n dimensions with mean 0 and $\sigma_q = 1.01\sigma_p$.

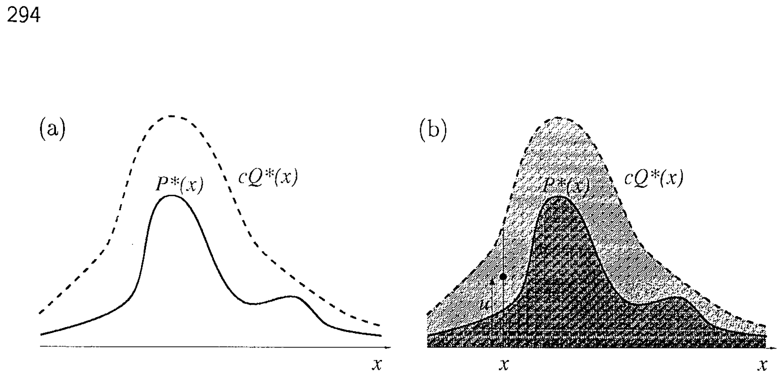
Since

$$q(0) = \left(\frac{1}{\sqrt{2\pi\sigma_q^2}}\right)^n \qquad p(0) = \left(\frac{1}{\sqrt{2\pi\sigma_p^2}}\right)^n$$

then

$$c = \frac{p(0)}{q(0)} = \left(\frac{\sigma_q}{\sigma_p}\right)^n = 1.01^n$$

With $n = 1000$ we find c=20.000.



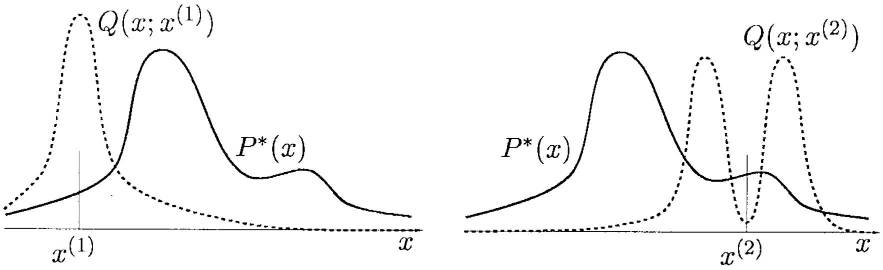
$$\text{Acceptance rate} = \frac{\text{volume } p}{\text{volume } cq} = \frac{1}{c}$$

Thus rejection sampling is inefficient in high dimensions.

A similar argument holds for importance sampling.

Metropolis algorithm

The Metropolis algorithm (1956) considers a sampling density which depends on the current sample value: $q(x|x^r)$.



Initialize in some random state x^1

At iteration r, sample x' from $q(x'|x^r)$ and compute

$$a = \frac{p^*(x')q(x^r|x')}{p^*(x^r)q(x'|x^r)}$$

If $a \geq 1$, accept x' as the new state: $x^{r+1} = x'$

Else, accept x' as the new state with probability a

If accept: $x^{r+1} = x'$, else $x^{r+1} = x^r$

Convergence of Metropolis algorithm

Metropolis algorithm is example of Markov process. Given two states x and x'. Define

$$a_{x'x} = \frac{p^*(x')q(x|x')}{p^*(x)q(x'|x)}, \quad a_{xx'} = \frac{1}{a_{x'x}}$$

Suppose $a_{x'x} \geq 1$. Then

Given x**,** the probability to accept x' is

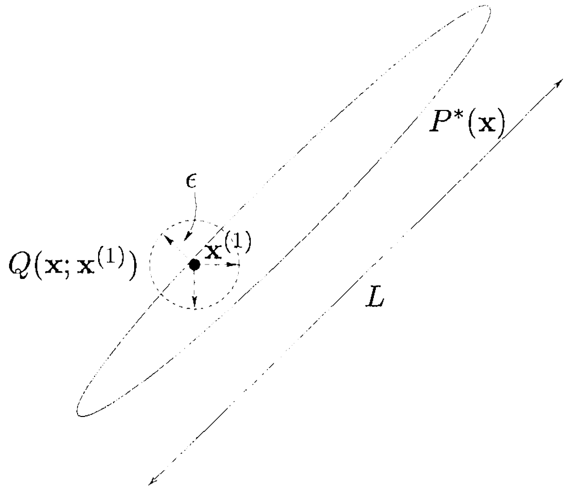
$$T(x'|x) = q(x'|x)$$

Given x'**,** the probability to accept x is

$$T(x|x') = q(x|x')a_{xx'}$$

$$\frac{T(x'|x)}{T(x|x')} = a_{x'x}\frac{q(x'|x)}{q(x|x')} = \frac{p^*(x')}{p^*(x)} = \frac{p(x')}{p(x)},$$

i.e. detailed balance. This implies that the process $T(x'|x)$ converges to $p(x)$.



When $q(x'|x)$ is Gaussian centered on x, $\frac{q(x'|x)}{q(x|x')}$ independent of x, x':

$$a_{x'x} = \frac{p^*(x')}{p^*(x)}$$

ϵ **large:**
Acceptance rate $a_{x'x}$ small.

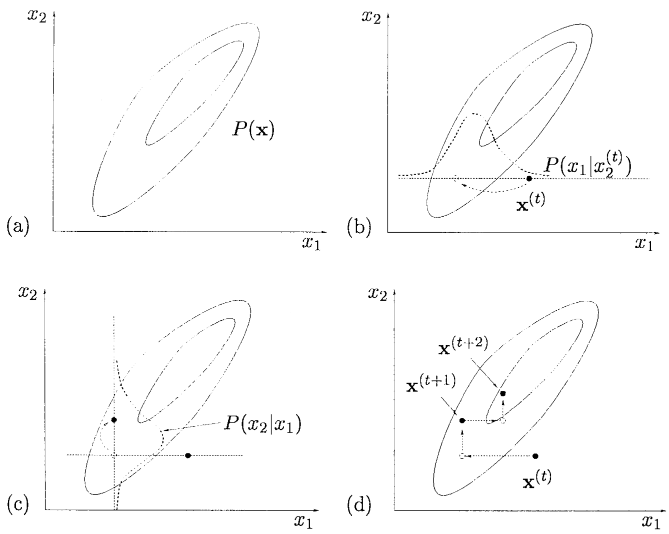
ϵ **small:**
Strong dependence on starting value
Many samples needed to sample.

Gibbs sampling

- Consider only change of one element of $(x_1, \ldots, x_n)$ at the time and define

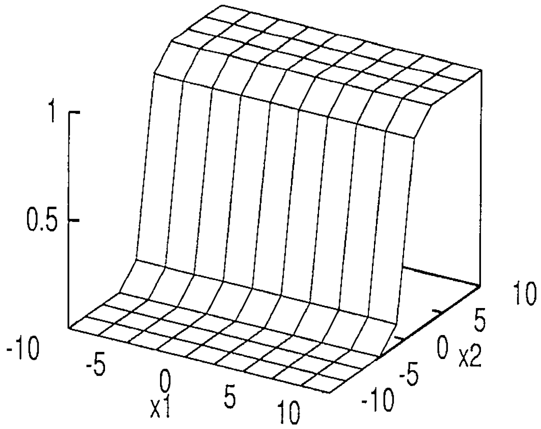
$$q(x'_i|x_1, \ldots, x_n) = p(x'_i|x_1, \ldots, x_n)$$

- Accept: $x^{r+1} = x'$



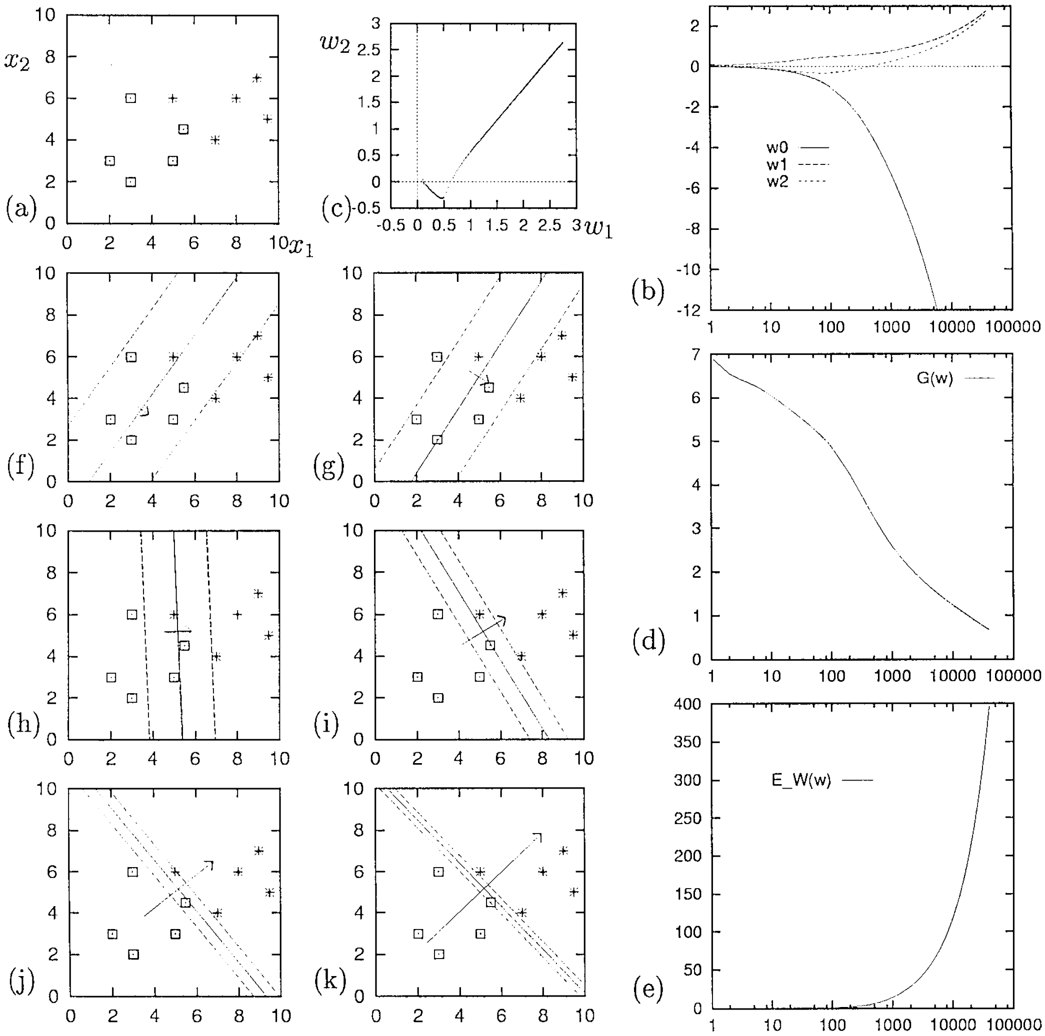
The one dimensional sampling can be done using for instance Rejection sampling.

Again the Perceptron



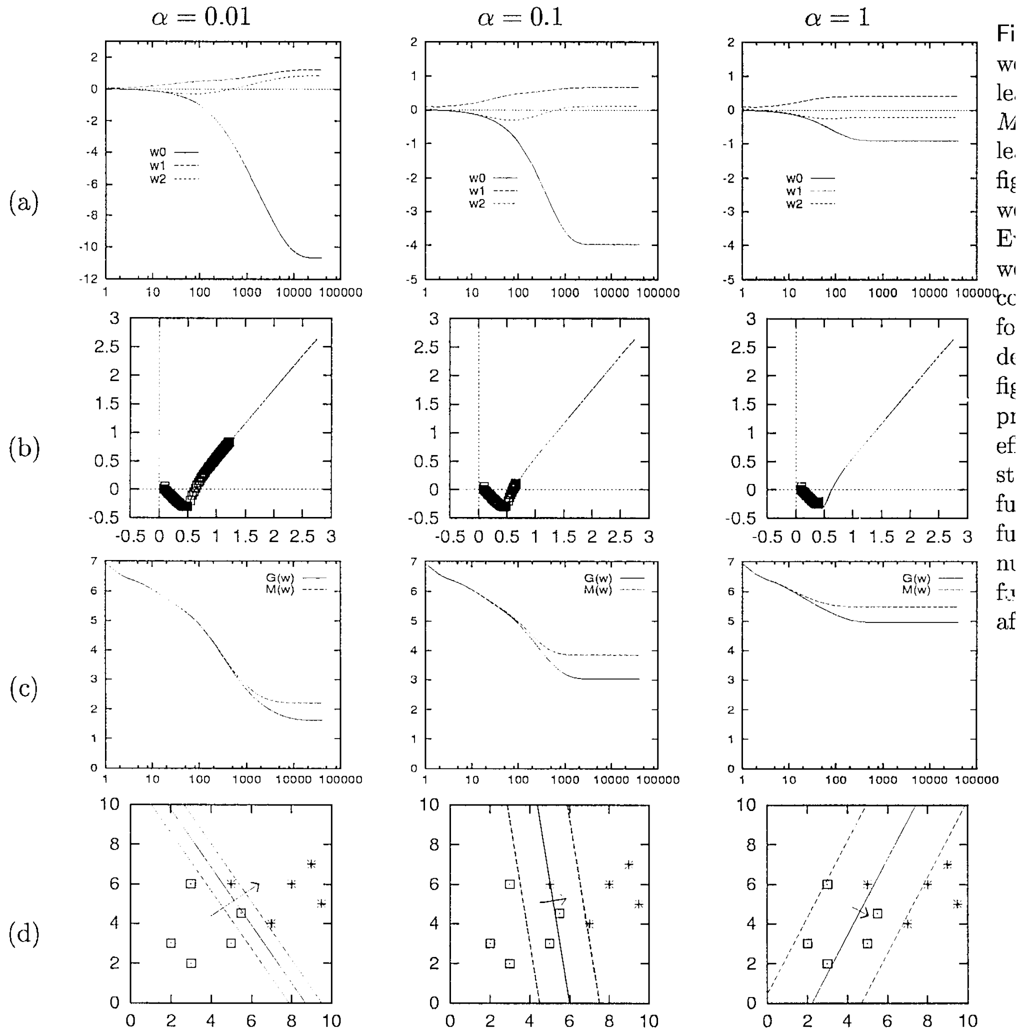
$$
\begin{aligned}
p(t=1|x,w) &= \sigma(\vec{w}\cdot\vec{x}) \\
G(w) &= -\sum_\mu \log(p(t^\mu|x^\mu,w)) \\
M(w) &= G(w) + \alpha E_w(w) \\
E_w(w) &= \sum_i w_i^2 \\
p(w|D) &= \frac{p(D|w)p(w)}{p(D)} \propto \exp(-M(w))
\end{aligned}
$$

The maximum likelihood solution



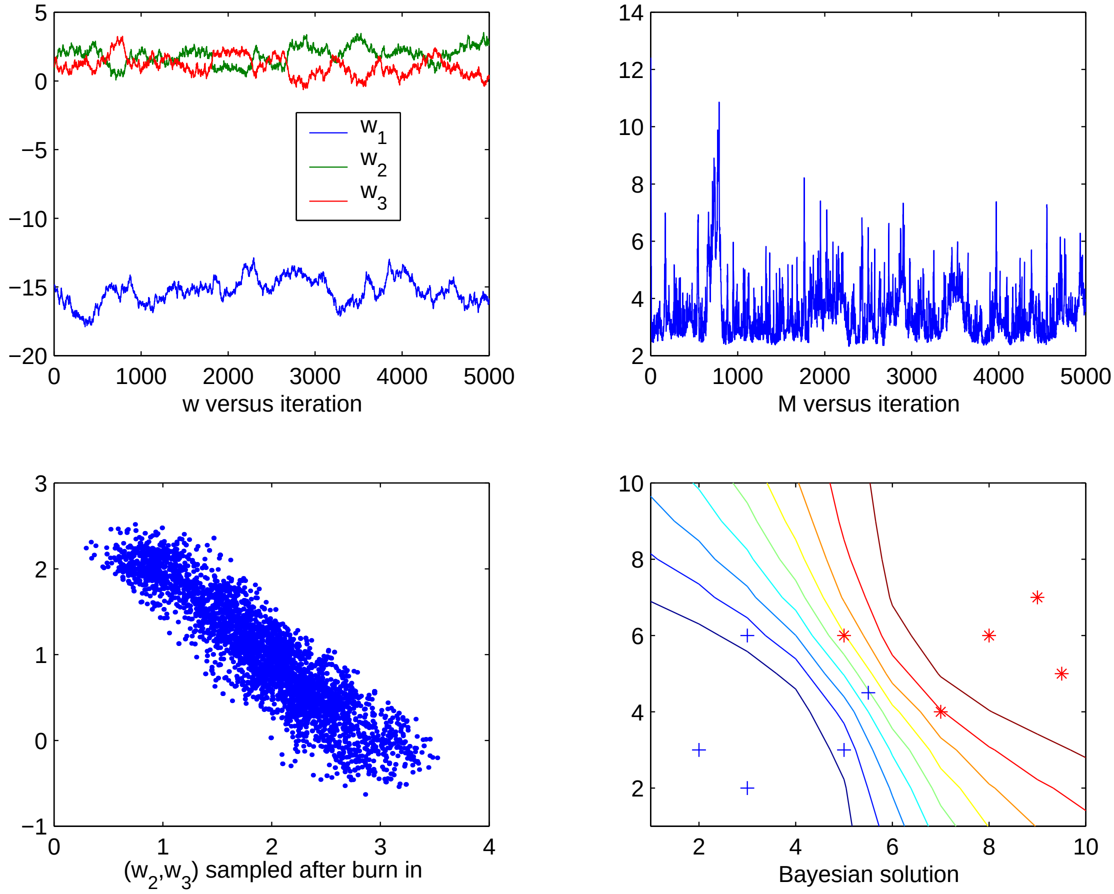
Minimizing the cost function $G(w)$ using gradient descent yields solutions with larger and larger weights.

The maximum posterior solution



Minimizing the cost function $M(w)$ yields more regular solutions for larger α.

The full Bayesian solution



$$\alpha = 0.01, q(w'|w) = \mathcal{N}(w'|w, \sigma), \sigma = 0.1$$

$$p(t|x) = \int dw p(t|x, w)p(w|D) \approx \frac{1}{R} \sum_r p(t|x, w^r)$$

The Hybrid Monte Carlo Method

Let

$$P(q) = \frac{e^{-E(q)}}{Z}$$

with E and its gradient $\frac{\partial E}{\partial q_i}$ easy to compute.

Gradient information reduces random walk behaviour in Metropolis method.

Double the state space by introducing for each q_i a momentum p_i Define the Hamiltonian

$$H(p,q) = E(q) + \frac{1}{2}\sum_i p_i^2$$

The Hamiltonian dynamics:

$$\frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial q_i} \qquad \frac{\partial q_i}{\partial t} = \frac{\partial H}{\partial p_i}$$

leaves H invariant.

Run Metropolis and Hamilton dynamics in (p,q) space.

Pseudo code

Choose initial q_1.

Loop:

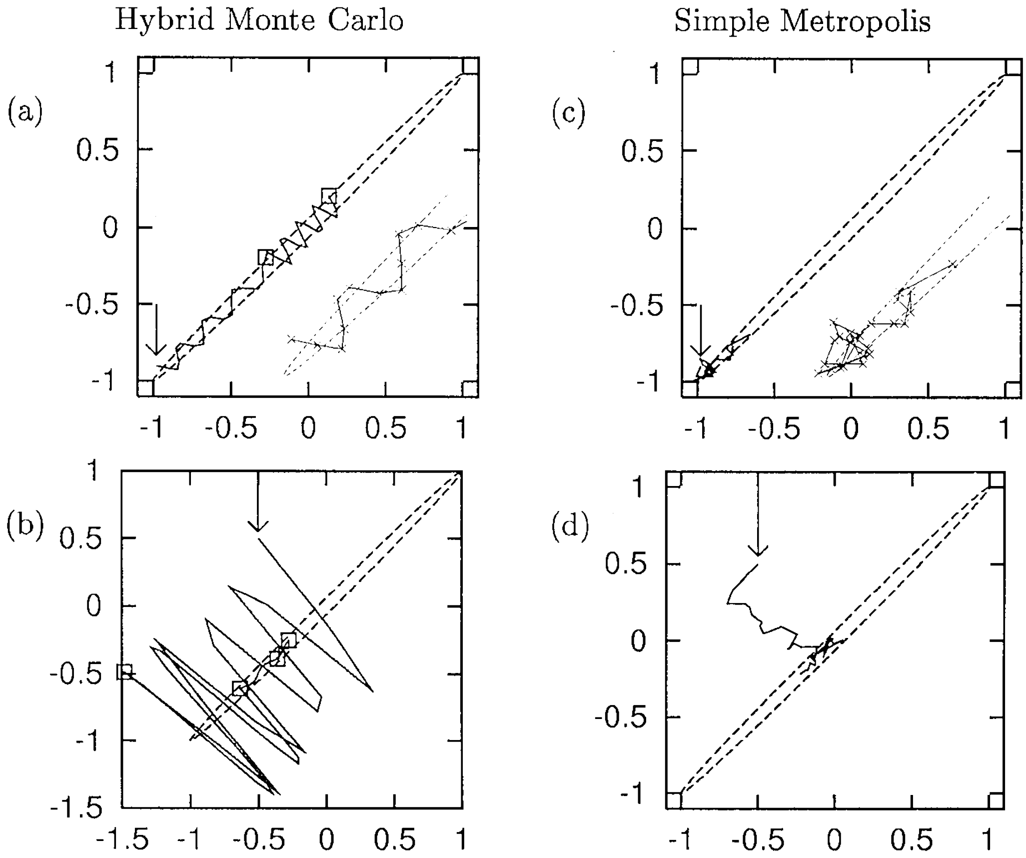
1. choose p_1 from $\mathcal{N}(0,1)$, giving (q_1, p_1)

2. run Hamilton dynamics, giving (q_2, p_2)

3. Metropolis step: accept (q_2, p_2) as new state with probability
$$\min\left(1, \frac{e^{-H(q_2,p_2)}}{e^{-H(q_1,p_1)}}\right)$$

4. On rejection, $(q_2, p_2) = (q_1, p_1)$

Comparison of HMC and Metropolis



Two dimensional elongated Gaussian distribution. a-b) Hybrid Monte Carlo method c-d) Metropolis method.