Approximate Inference

Max Welling

California Institute of Technology 136-93
Pasadena, CA 91125
welling@vision.caltech.edu

1 Introduction

Until now we have studied models that were all analytically tractible. It might have occurred to you that this basically implies using a Gaussian density for continuous variables, or using discrete random variables, usually distributed according to a multinomial density. The reason is that in order to calculate the E-step in the EM algorithm we need to integrate or sum over the hidden states. For some models summing over discrete states is computationally feasible. For other models the number of discrete hidden states is simply too large. In the case of continuous variables, integrations over Gaussians is one of the few that we know how to do analytically[1]. We have also seen examples where we combined normal random variables and discrete ones (i.e. MoG, MoE, HMM). In addition a Gaussian also allows a simple M-step, since we are really solving a weighted least squares problem. To go beyond these tractible models we would need alternative iterative optimization methods for the M-step. Here we could use basically any technique available in the optimization literature. The more serious problem occurs when we need to compute integrals over complicated distributions. In the E-step we actually need two of those. The first one occurs in the calculation of the posterior,

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})\ p(\mathbf{y})}{\int d\mathbf{y}'\ p(\mathbf{x}|\mathbf{y}')\ p(\mathbf{y}')} \tag{1}$$

and the second in the calculation of Q where we need to integrate the logarithm of the joint density over this posterior.

$$Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \int d\mathbf{y}\ p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}_{t-1})\ \log[p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}_t)] \tag{2}$$

Notice by the way that we faced the exact same problem when doing Bayesian estimation where we needed to calculate the posterior over the parameters $\boldsymbol{\theta}$, given the data,

$$p(\boldsymbol{\theta}|\mathbf{d}) = \frac{p(\mathbf{d}|\boldsymbol{\theta})\ p(\boldsymbol{\theta})}{\int d\boldsymbol{\theta}'\ p(\mathbf{d}|\boldsymbol{\theta}')\ p(\boldsymbol{\theta}')} \tag{3}$$

and then, in order to calculate the probability of a new data point, we needed to integrate over this posterior,

$$p(\mathbf{x}|\mathbf{d}) = \int d\boldsymbol{\theta}\ p(\mathbf{x}|\boldsymbol{\theta})\ p(\boldsymbol{\theta}|\mathbf{d}) \tag{4}$$

The topic of this lecture will be how to approximate these integrals, or how to estimate the parameters of a more complicated distribution approximately.

2 Variational EM

In the EM class we have already seen that we do not have to insert the posterior density $p(\mathbf{y}|\mathbf{x})$ for $q(\mathbf{y})$ in Q. As an alternative we could use a simple parametrized model $q(\mathbf{y}|\boldsymbol{\varphi})$. In the E-step we can now maximize, $F = Q + H$ (see EM lecture for definitions) over the parameters $\boldsymbol{\varphi}$. Note that we need to include the entropy term H in this case, since it also dependes on $q(\mathbf{y}|\boldsymbol{\varphi})$. This procedure obviously only makes sense if we use a model that is simpler than the full posterior, i.e. we can integrate over it, or sample from it (as we will see later), but one that is still able to capture important aspects of it. It is therefore very likely that we can not find the ML estimates of the parameters. The good news is that we still have a stable algorithm that maximizes a lower bound on the likelihood $F = L - KL$ (again see EM lecture for definitions). So maximizing F boils down to maximizing L (what we want) and at the same time minimizing KL which was the Kullback-Leibler distance between the posterior density $p(\mathbf{y}|\mathbf{x})$ and the model $q(\mathbf{y}|\boldsymbol{\varphi})$. Summarizing, the approximate algorithm thus is,

[1] It is note true that this computational convenience is the only reason for Gaussian distributions. The central limit theorem guarantees that random variables which are sums over many independent random variables converge to a Gaussian distribution (under some mild conditions). Isn't this a fortunate coincidence!

E-step Maximize or improve F over $\boldsymbol{\varphi}$.

M-step Maximize or improve F over $\boldsymbol{\theta}$.

3 Laplace Approximation

If we think that the posterior has only one mode, then it might be reasonable to approximate the full posterior by a Gaussian (Gaussians again!). The way to proceed is as follows. Assuming that the pdf is nowhere zero we can write,

$$p(\mathbf{y}|\mathbf{x}) = \exp \log[p(\mathbf{y}|\mathbf{x})] \tag{5}$$

Now we will approximate the log-term by a Taylor expansion around its "maximum a posteriori" value (MAP), i.e. the value of $\mathbf{y}$ that maximizes $p(\mathbf{y}|\mathbf{x})$.

$$\log[p(\mathbf{y}|\mathbf{x})] = \log[p(\mathbf{y}_{MAP}|\mathbf{x})] + \frac{1}{2}(\mathbf{y}-\mathbf{y}_{MAP})^T \nabla\nabla \log[p(\mathbf{y}_{MAP}|\mathbf{x})]\ (\mathbf{y}-\mathbf{y}_{MAP}) \tag{6}$$

where

$$\nabla\nabla_{ij} = \frac{\partial^2}{\partial \mathbf{y}_i \partial \mathbf{y}_j} \tag{7}$$

Notice that the linear term disappears from the Taylor series since we have at the maximum that

$$\nabla \log[p(\mathbf{y}_{MAP}|\mathbf{x})] = 0 \tag{8}$$

Raising this again to the exponent and multiplying it with a normalization constant to make it a pdf, we find,

$$p(\mathbf{y}|\mathbf{x}) \approx \frac{p(\mathbf{y}_{MAP}|\mathbf{x})}{Z} \exp[-\frac{1}{2}(\mathbf{y}-\mathbf{y}_{MAP})^T(-\nabla\nabla \log[p(\mathbf{y}_{MAP}|\mathbf{x})])\ (\mathbf{y}-\mathbf{y}_{MAP})] \tag{9}$$

This is then our Gaussian approximation, with mean $\mu = \mathbf{y}_{MAP}$ and covariance $\boldsymbol{\Sigma} = (-\nabla\nabla \log[p(\mathbf{y}_{MAP}|\mathbf{x})])^{-1}$. We also readily identify the normalization constant to be

$$Z = \frac{\sqrt{\det(-\nabla\nabla \log[p(\mathbf{y}_{MAP}|\mathbf{x})])}}{p(\mathbf{y}_{MAP}|\mathbf{x})\ (2\pi)^{\frac{1}{2}D}} \tag{10}$$

Usually, we are now able to perform the remaining integration over the posterior since we have approximated it by a simple Gaussian.

4 sampling

Yet another powerful technique to compute averages over complicated probability densities is to sample from them and approximate,

$$\mathbf{E}[\mathbf{f}(\mathbf{y})] = \int d\mathbf{y}\ p(\mathbf{y})\mathbf{f}(\mathbf{y}) \approx \frac{1}{N}\sum_{n=1}^{N} \mathbf{f}(\mathbf{y}_n) \tag{11}$$

To see why this is a reasonable estimate let us first check that the estimate is unbiased,

$$\mathbf{E}\left[\frac{1}{N}\sum_{n=1}^{N} \mathbf{f}(\mathbf{y}_n)\right] = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}[\mathbf{f}(\mathbf{y}_n)] = \mathbf{E}[\mathbf{f}(\mathbf{y})] \tag{12}$$

Then, we look at the variance of the estimator,

$$\begin{aligned} & \mathbf{E}\left[\left(\mathbf{E}[\mathbf{f}(\mathbf{y})] - \frac{1}{N}\sum_{n=1}^{N} \mathbf{f}(\mathbf{y}_n)\right)^2\right] \\ = \ & \mathbf{E}[\mathbf{f}(\mathbf{y})]^2 - 2\mathbf{E}[\mathbf{f}(\mathbf{y})]\frac{1}{N}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)] + \frac{1}{N^2}\sum_{n,m} \mathbf{E}[\mathbf{f}(\mathbf{y}_n)\mathbf{f}(\mathbf{y}_m)] \end{aligned}$$

$$
\begin{aligned}
&= -\mathbf{E}[\mathbf{f}(\mathbf{y})]^2 + \frac{1}{N^2}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)^2] + \frac{1}{N^2}\sum_{n\neq m} \mathbf{E}[\mathbf{f}(\mathbf{y}_n)]\mathbf{E}[\mathbf{f}(\mathbf{y}_m)] \\
&= -\mathbf{E}[\mathbf{f}(\mathbf{y})]^2 + \frac{1}{N^2}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)^2] + \frac{1}{N^2}\sum_{n,m} \mathbf{E}[\mathbf{f}(\mathbf{y}_n)]\mathbf{E}[\mathbf{f}(\mathbf{y}_m)] - \frac{1}{N^2}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)]^2 \\
&= \frac{1}{N^2}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)^2] - \frac{1}{N^2}\sum_n \mathbf{E}[\mathbf{f}(\mathbf{y}_n)]^2 \\
&= \frac{1}{N}\mathbf{E}\left[(\mathbf{f}(\mathbf{y}) - \mathbf{E}[\mathbf{f}(\mathbf{y})])^2\right] = \frac{1}{N}\mathbf{Var}[\mathbf{f}(\mathbf{y})] \qquad (13)
\end{aligned}
$$

The most important conclusion from this is that the variance scales as $\frac{1}{N}$, independent of the dimension of $\mathbf{y}$. We can now also argue that for many problems it does not make much sense to know the estimate much more accurate than the intrinsic uncertainty of the problem ($\mathbf{Var}[\mathbf{f}(\mathbf{y})]$). Therefore, using around 10 samples will provide us with an uncertainty smaller than $\frac{1}{3}\sqrt{\mathbf{Var}[\mathbf{f}(\mathbf{y})]}$, i.e. more than three times as accurate as the intrinsic uncertainty.

This opens an alternative route to compute the E-step. First sample from the posterior and then calculate Q by using,

$$
Q \approx \sum_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} \log[p(\mathbf{x}_n, \mathbf{y}_m|\boldsymbol{\theta}_t)] \qquad (14)
$$

In the following we will study some sampling techniques. Naive samplers however have the property that they are utterly useless in high dimensions. Markov Chain Monte Carlo (MCMC) sampling does not suffer from this (although it has problems of a different nature).

4.1 Sampling by Transforming

If the random variable $\mathbf{x} \sim p(\mathbf{x})$ is a function of another random variable $\mathbf{y} \sim q(\mathbf{y})$ from which it is easy to sample, then we can also sample easily from the first by using,

$$
\mathbf{x}_n = \mathbf{f}(\mathbf{y}_n) \qquad \mathbf{y}_n \sim q(\mathbf{y}) \qquad (15)
$$

As a simple example, consider sampling from a multivariate Gaussian and let's assume we have access to a random number generator for zero mean unit variance random numbers. Then, one can generate random numbers from a Gaussian with general mean and covariance, $\mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]$, as follows,

$$
\mathbf{x}_n = \mathbf{A}\mathbf{y}_n + \boldsymbol{\mu} \qquad \mathbf{A}\mathbf{A}^T = \boldsymbol{\Sigma}, \quad \mathbf{y} \sim \mathcal{G}_{\mathbf{y}}[0, \mathbf{I}] \qquad (16)
$$

The matrix $\mathbf{A}$ can be determined by an SVD as follows,

$$
\boldsymbol{\Sigma} = \mathbf{U}\mathbf{S}\mathbf{U}^T = \mathbf{A}\mathbf{A}^T \quad \Rightarrow \quad \mathbf{A} = \mathbf{U}\sqrt{\mathbf{S}} \qquad (17)
$$

and the square-root of the matrix $\mathbf{S}$ is defined by taking the square-root of the elements. As another useful 1D example we will assume to be able to sample from a Gamma distribution,

$$
q(y) = \frac{y^{\alpha-1}e^{-y}}{\Gamma(\alpha)} \qquad (18)
$$

where Γ is the gamma function. Then, by transforming to

$$
x = \frac{1}{w}\, y^{\frac{1}{c}}, \qquad (19)
$$

we find for $p(x)$,

$$
p(x) = \frac{wc}{\Gamma(a)}(wx)^{ac-1}e^{-(wx)^c} \qquad (20)
$$

which is a very general distribution for postive x which contains the distributions, Chi, Chi-squared, Erlang, Gamma, Exponential, Maxwell-Boltzmann, Generalized Normal, Rayleigh and Weibull.

As a more general approach (1D) we could try to find the transformation between the target density and the uniform density between 0 and 1.

$$p(x) = p(y) \left| \frac{\partial y}{\partial x} \right| \tag{21}$$

where | denotes taking the absolute value. We are after the function $x(y)$, since this would give us samples from $p(x)$ as function of the uniform samples. We therefore solve,

$$\begin{aligned} y(x) &= \int_{x'=-\infty}^{x} dx' \; p(x') = P(x) \Rightarrow \\ x(y) &= P^{-1}(y) \end{aligned} \tag{22}$$

where P denotes the cumulative distribution. One can plot P as a function of x, and plot the samples from the uniform distribution on the y-axis. From every sample we draw a horizontal line until you hit $P(x)$, at some some value of x. These is your samples from $p(x)$.

This method does not so nicely generalize to higher dimensions, and, since we need to know the inverse of the cumulative distribution, is not always very practical.

4.2 Importance Sampling

A slightly more sophisticated method is the following. Assume that we have some analytical expression for the target density *up to a normalization constant*, $f(\mathbf{x})$, with

$$p(\mathbf{x}) = \frac{1}{Z_f} f(\mathbf{x}) \tag{23}$$

Since this density is too hard to sample from we choose a simpler sample density $q(\mathbf{x})$ for which we only need to know an analytical expression for the unnormalised part $g(\mathbf{x})$, i.e. we have

$$q(\mathbf{x}) = \frac{1}{Z_g} g(\mathbf{x}) \tag{24}$$

Now let's assume we have samples $\mathbf{x}_n^q$ from this sample distribution, or equivalently from $g(\mathbf{x})$ (sampling from $q(\mathbf{x})$ and from $g(\mathbf{x})$ is the same as they only depend on the relative frequencies of occurances of $\mathbf{x}$). Can we now find an easy way to compute averages over the target density? This is done as follows,

$$\begin{aligned} \mathbf{E}[\boldsymbol{\phi}(\mathbf{x})] &= \int d\mathbf{x} \; q(\mathbf{x}) \; \frac{p(\mathbf{x})}{q(\mathbf{x})} \; \boldsymbol{\phi}(\mathbf{x}) \\ &= \int d\mathbf{x} \; q(\mathbf{x}) \; \frac{f(\mathbf{x})}{g(\mathbf{x})} \frac{Z_g}{Z_f} \; \boldsymbol{\phi}(\mathbf{x}) \\ &= \frac{Z_g}{Z_f} \int d\mathbf{x} \; q(\mathbf{x}) \; w(\mathbf{x}) \; \boldsymbol{\phi}(\mathbf{x}) \\ &\approx \frac{Z_g}{Z_f} \frac{1}{N} \sum_{n=1}^{N} w(\mathbf{x}_n^q) \; \boldsymbol{\phi}(\mathbf{x}_n^q) \\ &= \frac{\sum_{n=1}^{N} w(\mathbf{x}_n^q) \; \boldsymbol{\phi}(\mathbf{x}_n^q)}{\sum_{n=1}^{N} w(\mathbf{x}_n^q)} \end{aligned} \tag{25}$$

where we have defined weights,

$$w(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})} \tag{26}$$

and the fact that they fulfil,

$$\frac{1}{N} \sum_{n=1}^{N} w(\mathbf{x}_n^q) = \frac{Z_f}{Z_g} \tag{27}$$

The last equality is easily derived by choosing $\boldsymbol{\phi}(\mathbf{x}) = 1$ in (25).

This method will only work if the dimensionality of $\mathbf{x}$ is not very large. The problem is that the weights will be completely dominated by a few very large ones, while all the other ones are vanishingly small compared to those. This is because in high dimensions, probability densities are much more peaked, resulting in a few large values and many very small values.

4.3 Particle Filtering / Condensation

Sometimes we need to sample from a series of densities which slowly vary over time. For instance in the Kalman filter and the HMM we needed to know the density $p(\mathbf{y}_t|\mathbf{x}^t)$, i.e. the estimated state, given all the observations up till time t. This was calculated as follows,

$$p(\mathbf{y}_t|\mathbf{x}^t) = k(\mathbf{x}^t)\ p(\mathbf{x}_t|\mathbf{y}_t)\ p(\mathbf{y}_t|\mathbf{x}^{t-1}) \tag{28}$$

where $k(\mathbf{x}^t)$ is a constant independent of $\mathbf{y}_t$ and,

$$p(\mathbf{y}_t|\mathbf{x}^{t-1}) = \int d\mathbf{y}_{t-1}\ p(\mathbf{y}_t|\mathbf{y}_{t-1})\ p(\mathbf{y}_{t-1}|\mathbf{x}^{t-1}) \tag{29}$$

The first equality was interpreted as incorporating new evidence into the probability function, while the second one as propagation (or diffusion) of the probability over time, without taking a measurement into acount. We expect that these probabilities change slowly over time. If these distributions are not Gaussian or discrete, our Kalman filter/HMM equations would not be very usefull anymore. A sampling technique, called condensation can be used instead, and is in fact employed for applications like tracking. In the following we will assume that the measurement probability $p(\mathbf{x}_t|\mathbf{y}_t)$ can be any function of $\mathbf{x}_t$ and $\mathbf{y}_t$ (this function should of course be a probability in $\mathbf{x}_t$). For the transition probability $p(\mathbf{y}_t|\mathbf{y}_{t-1})$ we will allow any mapping $\mathbf{y}_t = \mathbf{f}(\mathbf{y}_{t-1})$, convolved with Gaussian noise, i.e. $\mathbf{y}_t$ is ditributed according to a full covariance Gaussian with mean $\boldsymbol{\mu} = \mathbf{f}(\mathbf{y}_{t-1})$,

$$\mathbf{y}_t \sim \mathcal{G}_{\mathbf{y}_t}[\mathbf{f}(\mathbf{y}_{t-1}), \mathbf{Q}] \tag{30}$$

The basic idea is related to importance sampling in the following way. First, let's assume we have samples from the posterior density,

$$\{\mathbf{y}_{t,n}\} \sim p(\mathbf{y}_t|\mathbf{x}^{t-1}). \tag{31}$$

To be able to compute averages over $p(\mathbf{y}_t|\mathbf{x}^t)$, we define the weights,

$$w_{t,n} = \frac{p(\mathbf{y}_{t,n}|\mathbf{x}^t)}{p(\mathbf{y}_{t,n}|\mathbf{x}^{t-1})} = k(\mathbf{x}^t)\ p(\mathbf{x}_t|\mathbf{y}_{t,n}), \tag{32}$$

where we used (28). This leads to the definition of the following normalized weights at time t,

$$\pi_{t,n} = \frac{w_{t,n}}{\sum_{n=1}^N w_{t,n}} = \frac{p(\mathbf{x}_t|\mathbf{y}_{t,n})}{\sum_{n=1}^N p(\mathbf{x}_t|\mathbf{y}_{t,n})} \tag{33}$$

The pair $\{\mathbf{y}_{t,n}, \pi_{t,n}\}$ is said to represent the posterior density $p(\mathbf{y}_t|\mathbf{x}^t)$, since any average over this distribution can now be computed using importance sampling,

$$\mathbf{E}[\boldsymbol{\phi}(\mathbf{y}_t)|\mathbf{x}^t] \approx \sum_{n=1}^N \pi_{t,n}\ \boldsymbol{\phi}(\mathbf{y}_{t,n}) \tag{34}$$

Now we would like to produce the equivalent tuple for time $t+1$. One possible strategy is to change the weights $\pi_{t,n}$. The disadvantage is that if the posterior has changed too much after a few time steps, than we have a few very large weights, while all the other ones vanish, i.e. the location of the samples is not in the region of large probability any more. A better strategy is to resample the sample you have, by drawing with replacement samples according to the probabilities π_n. This will have the result that samples which lie in probable regions will be drawn more often, while others will not be chosen at all. If we would compute the new weights for this new sample (which we will not yet do), they would be more uniformly distributed, i.e. the problem of a few very large weights versus a lot very small ones is softened. Notice however, that the number of different samples necessarily decreases over time, i.e. more and more copies will appear. We need another step, which again diversifies the samples, which will be the diffusion step. One could say, that we smooth the histogram we have found after resampling. The way we must smooth is dictated by (29), i.e. the natural diffusion of the dynamical system. Assuming, that the noise is Gaussian we can write,

$$\mathbf{y}_{t+1,n} = \mathbf{f}(\mathbf{y}_{t,n}) + \boldsymbol{\nu}_n \qquad \boldsymbol{\nu} = \mathcal{G}_{\boldsymbol{\nu}}[0, \mathbf{Q}] \tag{35}$$

Notice, that this step is actually a nonlinear map plus a smoothing. After this step there are no two identical copies anymore in the sample. The new, smoothed sample is thus from the distribution, $p(\mathbf{y}_{t+1}|\mathbf{x}^t)$, since

the above step precisely implements equation (29) through sampling. Finally, we need to compute the new weights $\pi_{t+1,n}$, and together with the above samples, they represent the probability density $p(\mathbf{y}_{t+1}|\mathbf{x}^{t+1})$. We are now where we started, but one time step further.

Finally, there is an easy trick to quickly perform the resampling step as follows. First comute the cumulative weights,

$$c_{t,n} = \sum_{i=1}^{n} \pi_{t,i}, \tag{36}$$

Then, by the exposition in section 4.1, we can sample a random value r uniformly between $[0, 1]$, and choose the smallest index such that $c_{t,i} \geq r$. This is then the sampled index.

4.4 Rejection Sampling

Again, we assume that we know analytical expressions for the unnormalised probability functions $f(\mathbf{x})$ and $g(\mathbf{x})$. Furthermore we will assume that we know a constant c such that for all $\mathbf{x}$,

$$c\ g(\mathbf{x}) > f(\mathbf{x}) \qquad \forall\ \mathbf{x} \tag{37}$$

Next, follow the following procedure. Generate samples from $q(\mathbf{x})$ (or equivalently $g(\mathbf{x})$). Then, for every sample, draw a uniform value u, between $[0, c\ g(\mathbf{x})]$. If the value of u is larger than $f(\mathbf{x})$ reject the sample, otherwise except it. Thus, while the $\mathbf{x}_n$ are chosen uniformly from the area under the curve $c\ g(\mathbf{x})$, the rejection rule ensures that the corrected sample is chosen uniformly from under the curve $f(\mathbf{x})$, i.e they are chosen from the target density $p(\mathbf{x})$. In high dimensional spaces we find in practice that the value for c needs to be chosen very large to ensure (37) holds true. It may also be very difficult to find such a value. The result is that there is an enormous rejection rate, rendering also this method impractical in high dimensions.

4.5 Markov Chain Monte Carlo

The conclusion from both the rejection sampling and the importance sampling was that they suffer from the "curse of dimensionality". In both cases, even in lower dimensional spaces, we also needed a proposal density which tightly matches with the target density. Both these problems can be circumvented by giving up on the idea that all samples need to be independent. Instead, we will allow the current sample from a proposal density to depend on the previously drawn sample. If a sample depends on a previous sample, but given that sample does not depend on any earlier samples, than we say that the samples form a Markov Chain (note the analogy with HMMs and Kalman Filters). The strategy will be to draw a sample from a proposal density, conditioned on the previous sample, but which *does not need to match the target density closely*, and subsequently reject a certain amount of the samples according to rules described below. In this sense it is a generalization of the rejection sampling method.

Firstly, we mention that the sample estimate of the average in (11) is still unbiased (without proof) and that the total variance (13) will now be multiplied by a factor depending the correlations between the samples, but not on the dimensionality of the problem. This implies that even with dependent data we can still find an unbiased estimate of an average, albeit with a larger variance.

What are the ingredients for such a Markov Chain sampler. Most importantly, we need a proposal density $S(\mathbf{x}_n|\mathbf{x}_{n-1})$ from which to sample. Then, we need a rule for rejecting or accepting samples from that density, $A(\mathbf{x}_n, \mathbf{x}_{n-1})$. Together, they must ensure that eventually, the samples will be drawn from the target distribution $p(\mathbf{x})$. The idea is now to start with random initial distribution $q(\mathbf{x}_1)$ and draw the first sample. Obviously, if $q(\mathbf{x}) \neq p(\mathbf{x})$ the sample is not drawn from $p(\mathbf{x})$. Next, we propose a new sample from $S(\mathbf{x}_2|\mathbf{x}_1)$ and accept it with a probability $A(\mathbf{x}_2, \mathbf{x}_1)$. The total transition probabability for some $\mathbf{x}_2 \neq \mathbf{x}_1$ is therefore,

$$T(\mathbf{x}_2|\mathbf{x}_1) = S(\mathbf{x}_2|\mathbf{x}_1)\ A(\mathbf{x}_2, \mathbf{x}_1) \qquad \mathbf{x}_2 \neq \mathbf{x}_1 \tag{38}$$

If the new state is rejected, we simply keep the old sample, which now appears twice in our sample-set. We can now ask the question, what is the marginal distribution from which we picked $\mathbf{x}_2$? This given by,

$$\tilde{q}(\mathbf{x}_2) = \int d\mathbf{x}_1\ T(\mathbf{x}_2|\mathbf{x}_1)\ q(\mathbf{x}_1) \tag{39}$$

Note that this is not equal to $q(\mathbf{x}_1)$ in general. Therefore during the execution of the Markov chain the sampling distribution changes. What we need is that after some time the distribution becomes equal to the target distribution. There are two important issues. Firstly, we need to make sure that the chain has only one unique stable (or equilibrium) distribution and that it will always be reached if the chain is run long

enough. This property is called ergodicity and will not pursue it here. Then, we need to make sure that this unique stable distribution is the target distribution. This can be expressed through,

$$p(\mathbf{x}_n) = \int d\mathbf{x}_{n-1}\ T(\mathbf{x}_n|\mathbf{x}_{n-1})\ p(\mathbf{x}_{n-1}) \tag{40}$$

Usually, a stronger condition can be checked with less effort, which is called detailed balance,

$$T(\mathbf{x}_n|\mathbf{x}_{n-1})\ p(\mathbf{x}_{n-1}) = T(\mathbf{x}_{n-1}|\mathbf{x}_n)\ p(\mathbf{x}_n) \tag{41}$$

To see that detailed balance implies (40) we integrate the right hand site over $\mathbf{x}_{n-1}$ using $\int d\mathbf{x}_{n-1}\ T(\mathbf{x}_{n-1}|\mathbf{x}_n) = 1$. A chain satisfying detailed balance is said to be reversible for obvious reasons.

At this stage we can already draw one important conclusion. Even if we have chosen transition probabilities, such that the equilibrium distribution is the target distribution, in the initial stages of the Markov chain, we will not be sampling from this target distribution. In other words, we need to give it some time to reach the equilibrium distribution (burn in period). It is often hard to tell whether we have reached equilibrium in practice.

Next, let us assume that the proposal distribution $S(\mathbf{x}_n|\mathbf{x}_{n-1})$ is symmetric under exchanging $\mathbf{x}_n$ with $\mathbf{x}_{n-1}$. In this case a valid acceptance function $A(\mathbf{x}_n, \mathbf{x}_{n-1})$ is given by,

$$A(\mathbf{x}_n, \mathbf{x}_{n-1}) = \mathbf{min}\left[1, \frac{f(\mathbf{x}_n)}{f(\mathbf{x}_{n-1})}\right] \tag{42}$$

where we have defined again $p(\mathbf{x}) = \frac{f(\mathbf{x})}{Z_f}$. In words the above says, if the probability of the new sample is larger than the probability of the old sample, always accept. Otherwise, accept with probability $\frac{f(\mathbf{x}_n)}{f(\mathbf{x}_{n-1})}$. It is now easy to show that also this rule satisfies detailed balance. For $\mathbf{x}_n = \mathbf{x}_{n-1}$ this is trivial since the new sample is then equal to the old sample. For $\mathbf{x}_n \neq \mathbf{x}_{n-1}$ we have,

$$\begin{aligned}
T(\mathbf{x}_n|\mathbf{x}_{n-1})\ p(\mathbf{x}_{n-1}) &= S(\mathbf{x}_n|\mathbf{x}_{n-1})\ \mathbf{min}\left[1, \frac{f(\mathbf{x}_n)}{f(\mathbf{x}_{n-1})}\right]\ p(\mathbf{x}_{n-1}) \\
&= S(\mathbf{x}_n|\mathbf{x}_{n-1})\ \mathbf{min}\left[1, \frac{p(\mathbf{x}_n)}{p(\mathbf{x}_{n-1})}\right]\ p(\mathbf{x}_{n-1}) \\
&= S(\mathbf{x}_n|\mathbf{x}_{n-1})\ \mathbf{min}\left[p(\mathbf{x}_{n-1}), p(\mathbf{x}_n)\right] \\
&= S(\mathbf{x}_{n-1}|\mathbf{x}_n)\ \mathbf{min}\left[p(\mathbf{x}_n), p(\mathbf{x}_{n-1})\right] \\
&= T(\mathbf{x}_{n-1}|\mathbf{x}_n)\ p(\mathbf{x}_n)
\end{aligned} \tag{43}$$

The above algorithm is called Metropolis algorithm, due to the particular acceptance function. A popular alternative acceptance function is the Boltzmann acceptance function.

$$A(\mathbf{x}_n|\mathbf{x}_{n-1}) = \frac{p(\mathbf{x}_n)}{p(\mathbf{x}_n) + p(\mathbf{x}_{n-1})} \tag{44}$$

which basically says, propose a new sample, and then chose among the old and the new sample randomly according to their relative probabilities. You can easily check for yourself that this rule satisfies detailed balance.

The acceptance rule can also be easily generalized for non-symmetric sample densities as follows,

$$A(\mathbf{x}_n, \mathbf{x}_{n-1}) = \mathbf{min}\left[1, \frac{f(\mathbf{x}_n)\ S(\mathbf{x}_{n-1}|\mathbf{x}_n)}{f(\mathbf{x}_{n-1})\ S(\mathbf{x}_n|\mathbf{x}_{n-1})}\right] \tag{45}$$

It is an easy matter to prove the detailed balance property again. The MCMC algorithm with this property is called Metroplis-Hastings sampler.

It is sometimes also very useful to draw samples from one dimension at a time, keeping all the other dimensions fixed, i.e.

$$S(\mathbf{x}_n|\mathbf{x}_{n-1}) = S_k(x_{n,k}|\mathbf{x}_{n-1})\ \prod_{i\neq k} \delta(x_{n,i} - x_{n-1,i}) \tag{46}$$

Note, that for sample n we thus replace only the dimension k with its new updated value, while the rest remain at their old value. For the next sample $n+1$, we choose a different dimension i (possibly also at

random) and sample again, now conditioned on the sample with the k'th dimension updated. The acceptance function will be,

$$A_k(x_{n,k}, \mathbf{x}_{n-1}) = \mathbf{min}\left[1, \frac{f(\mathbf{x}_n)\ S_k(x_{k,n-1}|\mathbf{x}_n)}{f(\mathbf{x}_{n-1})\ S_k(x_{k,n}|\mathbf{x}_{n-1})}\right] \tag{47}$$

To proof detailed balance, we need to proof detailed balance for all the sub-samplers. For $\mathbf{x}_n = \mathbf{x}_{n-1}$ this is again trivial. For any value $x_{i,n} \neq x_{i,n-1}, \quad i \neq k$ the transition probability is zero, which therefore also satisfies detailed balance. Finally then, we need to proof the detailed balance for $x_{k,n} \neq x_{k,n-1}$, which goes analoguesly as in the previous cases.

Finally, we would like to know what the explicit form of the proposal density $S(\mathbf{x}_n|\mathbf{x}_{n-1})$ is? One of the simplest choices is a isotropic Gaussian with a mean equal to the previous sample value. Remains the question as to which variance we should choose. If we choose it too large, we will have a very high rejection rate in high dimenions. High rejection rates also imply highly dependent samples since we end up with a long list of the same sample, before another one is accepted (remember if a sample is rejected we duplicate the old one). The peaky shape of distributions in high dimensions was indeed the reason why rejection sampling wasn't working in high dimensions in the first place. We therefore see that MCMC is protected against this high rejection rate exactly by ensuring that the next sample is still close to the old sample and therefore likely to be in a region of space where $p(\mathbf{x})$ is large. But choosing the variance too small is also not good, since we need a lot of time to move from one region of space to another, and therefore we need a lot a time to sample from all regions of x-space where $p(\mathbf{x})$ has a significant value. If we call σ the variance of the proposal Gaussian, L a measure of the length scale of the target density, and α the total acceptance rate, then we need approximately,

$$T = \frac{1}{\alpha}\left(\frac{L}{\sigma}\right)^2 \tag{48}$$

iterations to traverse from one side of the target density to the other side[2]. We therefore have a trade-off between chosing a large and a small variance for the proposal density. Heuristically, one may tune the system such that it has a reasonable acceptance rate, let's say, one out of three (one cannot adapt the variance while actually gathering the samples though!). There have been developed methods to improve this slow traversion of state space which are beyond the scope of this class (i.e. Hybrid Monte Carlo, overrelaxation).

In the literature it is also recommended to use a proposal sample density with longer tails then the normal density (e.g. a Cauchy density), which has the advantage that larger steps in state space are more often tried out.

Another pitfall for MCMC methods is that the target density may be multimodal. If the modes are seperated by regions of very low probabilities, than it becomes increasingly difficult to jump from one region to another. What we need is a method that at least settles in the region where most of the probability mass is present (ideally we want a method which visits all regions of course), since this will give the dominant contribution to the sample average that we want to calculate. A method which improves ones chances of ending up in important regions of x-space (i.e. that doesn't get trapped on small islands of likelihood), is given by *simulated annealing*. We first define,

$$p(\mathbf{x}) = \frac{1}{Z_f}\ f(\mathbf{x})^\alpha \tag{49}$$

where α fullfills the role of inverse temprature (compare with deterministic annealing in vector quantization!). If $\alpha = 1$ we are sampling from the correct density. For lower α (higher temperature), the modified density looks more uniform, with less isolated islands. If we start our sampling at a high temperature and slowly lower it (annealing) until $\alpha = 1$ is reached, we decrease our chances of ending up on an insignificant island[3].

4.6 Gibbs Sampling

Gibbs sampling is a special case of the general MCMC method, where we sample one dimension at a time. The way we decide to do it however, ensures that the new sample will always be accepted! The procedure is as follows.

- Determine the conditional density $p(x_1|x_2, ..., x_d)$ and draw a sample.

[2] This equation can be derived using the fact that it is a random walk

[3] This technique can also be used to find the minimum of a function without ending up in a local minimum. First write $f(\mathbf{x})^\alpha = e^{-\alpha E(\mathbf{x})}$ where $E(\mathbf{x})$ is the energy function, the minimum of which you want to find. Start your sampling at low α and slowly increase until $\alpha = \infty$ (not 1!). At $T = 0$, the only admissible state to sample from is the ground state of the system, i.e. the minimum of $E(\mathbf{x})$. The annealing scheme is again reducing ones chances of getting stuck in local minima.

- determine the conditional density $p(x_2|x_1, x_3, x_4, ..., x_d)$ and draw a sample. The value of the conditioning variable x_1 is the one just drawn in the previous iteration.
- etcetera until you have reached x_d. Then start all over again.

It is easy to prove detailed balance again,

$$\begin{aligned} T(x'_k|\mathbf{x})\ p(\mathbf{x}) &= p(x'_k|x_{i\neq k})\ p(\mathbf{x}) \\ &= \frac{p(x'_k, x_{i\neq k})}{p(x_{i\neq k})}\ p(x_k|x_{i\neq k})\ p(x_{i\neq k}) \\ &= p(x_k|x_{i\neq k})\ p(x'_k, x_{i\neq k}) \\ &= p(x_k|x'_{i\neq k})\ p(x'_k, x'_{i\neq k}) \\ &= T(x_k|\mathbf{x}')\ p(\mathbf{x}') \end{aligned} \tag{50}$$

where $'$ denotes new sample, and we used that $x'_{i\neq k} = x_{i\neq k}$. Proving that it is an ergodic Markov Chain is in general more involved and case dependent. As mentioned, Gibbs sampling may also be interpreted as an instance of MCMC sampling where the acceptance rate is simply one. This is easily checked by computing the acceptance function for non symmetric MCMC where we sample from one dimension at a time (47),

$$\begin{aligned} A_k(x'_k, \mathbf{x}) &= \mathbf{min}\left[1, \frac{p(\mathbf{x}')\ S_k(x_k|\mathbf{x}')}{p(\mathbf{x})\ S_k(x'_k|\mathbf{x})}\right] \\ &= \mathbf{min}\left[1, \frac{p(x'_k|x'_{i\neq k})\ p(x'_{i\neq k})\ p(x_k|x'_{i\neq k})}{p(x_k|x_{i\neq k})\ p(x_{i\neq k})\ p(x'_k|x_{i\neq k})}\right] \\ &= \mathbf{min}\left[1, 1\right] = 1 \end{aligned} \tag{51}$$

where we used again $x'_{i\neq k} = x_{i\neq k}$. The good thing about Gibbs sampling is of course that we don't have to tune any parameters, and that we have a zero rejection rate. The method does require of course that the conditional probabilities are easy to calculate and to sample from.