A Mode-Hopping MCMC sampler

Cristian Sminchisescu, Max Welling and Geoffrey Hinton
Department of Computer Science, University of Toronto*
10 King's College Road, Toronto, M5S 3G5 Canada
{*crismin,welling,hinton*}*@cs.toronto.edu*

Abstract

One of the main shortcomings of Markov chain Monte Carlo samplers is their inability to mix between modes of the target distribution. In this paper we show that advance knowledge of the location of these modes can be incorporated into the MCMC sampler by introducing mode-hopping moves that satisfy detailed balance. The proposed sampling algorithm explores local mode structure through local MCMC moves (*e.g.* diffusion or Hybrid Monte Carlo) but in addition also represents the relative strengths of the different modes correctly using a set of global moves. This "mode-hopping" MCMC sampler can be viewed as a generalization of the darting method [1].

1 Introduction

It is well known that MCMC samplers have great difficulty in mixing from one mode to the other because it typically takes many steps of very low probability to make the trip. Recent improvements designed to combat random walk behavior, like Hybrid Monte Carlo and over-relaxation [14] do not solve this problem either. The question we'll answer in the paper is how to exploit knowledge of the location of the modes to design a MCMC sampler that mixes properly between them.

In this paper we'll consider two possible scenarios where this advance knowledge is present. In one example we have actively searched for high probability regions using sophisticated optimization methods [16, 23, 24, 25]. Given these local maxima, we now desire to collect unbiased samples from the underlying probability distribution. In another example we are given data-cases and aim at learning a model distribution to represent these data as accurately as possible. In this case, high probability regions should coincide with the location of large clusters of data and a clustering algorithm could be employed to identify them.

This paper is organized as follows. In section 2 we review some popular Markov chain Monte Carlo methods. Then, in section 3 we introduce the new mode-hopping sampler and some extensions. An additional proof of detailed balance appears in the appendix. Section 4 explains and illustrates an application to learning Markov random fields, while in section 5 the generalized darting method is evaluated against the spherical darting method on a "real world" vision application – estimating 3-D human body poses from 2-D image information.

Univ. of Toronto Technical Report CSRG-478, September 2003. Subm. to Machine Learning.

2 Markov Chain Monte Carlo Sampling

Imagine we are given a probability distribution $p(\mathbf{x})$ with $\mathbf{x} \in R^d$ a vector of continuous random variables. In the following we will focus on continuous variables, but the algorithm is easily extended to discrete state spaces. A very general method to sample from this distribution is provided by Markov chain Monte Carlo (MCMC) sampling. The idea is to start with an initial distribution $p_0(\mathbf{x})$ and design a set of transition probabilities that will eventually converge to the target distribution $p(\mathbf{x})$.

The most commonly known transition scheme is the Metroplis-Hastings (M-H) algorithm, where a target point is sampled from a possibly asymmetric conditional distribution $Q(\mathbf{x}_{t+1}|\mathbf{x}_t)$, where $\mathbf{x}_t$ represents the current sample. To make sure that detailed balance holds, *i.e.* $p(\mathbf{x}_t)Q(\mathbf{x}_{t+1}|\mathbf{x}_t) = p(\mathbf{x}_{t+1})Q(\mathbf{x}_t|\mathbf{x}_{t+1})$, which in turn guarantees that the target distribution remains invariant under Q, we should only accept a certain fraction of the proposed targets,

$$P_{accept} = \min\left[1, \frac{p(\mathbf{x}_{t+1})Q(\mathbf{x}_t|\mathbf{x}_{t+1})}{p(\mathbf{x}_t)Q(\mathbf{x}_{t+1}|\mathbf{x}_t)}\right] \quad (1)$$

In the most commonly used M-H algorithm, the transition distribution Q is symmetric and independent of the energy-surface at location $\mathbf{x}$. This simplifies (1) (the Q factors cancel), but leads to slow mixing due to random walk behavior. It is however not hard to incorporate local gradient information, $dE(\mathbf{x})/\mathbf{dx}$ to improve mixing speed. One could for instance bias the proposal distribution $Q(\mathbf{x}_{t+1}|\mathbf{x}_t)$ in the direction of the negative gradient $-dE(\mathbf{x})/\mathbf{dx}$ and accept using (1) [1],

$$\mathbf{x}_{\tau+1} \;=\; \mathbf{x}_\tau - \frac{\Delta\tau^2}{2}\left.\frac{dE(\mathbf{x})}{\mathbf{dx}}\right|_{\mathbf{x}=\mathbf{x}_\tau} + \Delta\tau\,\mathbf{n} \quad (2)$$

where $\mathbf{n}$ is a vector of independently chosen Gaussian variables with zero mean and unit variance, and $\Delta\tau$ is the stepsize. When the stepsize becomes infinitesimally small this is called the Langevin method and one can show that the rejection rate vanishes in this limit.

The Langevin method is a special case of a more general sampling technique called Hybrid Monte Carlo (HMC) sampling [14]. In HMC the particle is given a random initial momentum sampled from a unit-variance isotropic Gaussian density and its deterministic trajectory along the energy surface is then simulated for T time steps using Hamiltonian dynamics. If this simulation has no numerical errors the increase, ΔE, in the combined potential and kinetic energy will be zero. If ΔE is positive, the particle is returned to its initial position with a probability of $1 - \exp(-\Delta E)$. Numerical errors up to second order are eliminated by using a "leapfrog" method which uses the potential energy gradient at time τ to compute the velocity increment between time $\tau - \frac{1}{2}$ and $\tau + \frac{1}{2}$ and uses the velocity at time $\tau + \frac{1}{2}$ to compute the position increment between time τ and $\tau + 1$. The Langevin method corresponds to precisely one step of HMC (*i.e.* $T = 1$).

A host of clever MCMC samplers can be found in the literature. We refer to the excellent review [14] for more information.

3 The Mode-Hopping MCMC Algorithm

We start with reviewing the closely related "darting" algorithm described in [1]. In darting-MCMC we place spherical jump regions of equal volume at the location of the modes of the target distribution. The algorithm is based on a simple local MCMC sampler which is

[1] One can use more general biased proposal distributions, but the one defined in (2) was chosen because of its vanishing rejection rate in the limit $\Delta \rightarrow 0$

interrupted with a certain probability to check if its current location is inside one of these spheres. If so, we initiate a jump to the corresponding location in another sphere, chosen uniformly at random, where the usual Metropolis acceptance rule applies. To maintain detailed balance we decide not to move if we are located outside any of the balls. It is not hard to check that this algorithm maintains detailed balance between any two points in sampling space.

In high dimensional spaces this procedure may still lead to unacceptably high rejection rates because the modes will likely decay sharply in at least a few directions. Since these ridges of probability are likely to be uncorrelated across the modes, the proposed target location of the jump will have very low probability, resulting in almost certain rejection. In the following we will propose two important improvements over the darting method. Firstly, we allow the jump regions to have arbitrary shapes and volumes and secondly these regions may overlap. The first extension opens the possibility to align the jump regions precisely with the shape of the high probability regions of the target distribution. The second extension simplifies the design and placement of the jump regions since we don't have to worry about possible overlaps of the chosen regions.

First consider the case when the regions are non-overlapping but of different volumes. Like in the darting method we could consider a one-to-one mapping between points in the different regions, or we could choose to sample the target point uniformly inside the new region. Because the latter is somewhat simpler conceptually, we'll use uniform sampling in this section. The deterministic case will be treated in the next section. Also, to simplify the discussion we'll first consider the case where the underlying target distribution is uniform, *i.e.* has equal probability everywhere. Due to the difference in volumes, particles are more likely to be inside a large region than in small ones. Thus, there will be a larger flow of particles going from the bigger regions towards the smaller ones violating detailed balance. To correct for it we could reject a fraction of the proposed jumps from larger towards smaller regions. There is however a smarter solution, that picks the target region proportional to its volume,

$$P_i = \frac{V_i}{\sum_j V_j} \tag{3}$$

If we view the jumps between the various regions as a (separate) Markov chain, this method samples directly from the equilibrium distribution while a rejection method would require a certain mixing time to reach equilibrium. Clearly, if the underlying distribution is not uniform, we need the Metropolis acceptance rule between the jump point and its image in the target region,

$$P_{accept} = \min\left[1, \frac{P(\mathbf{t})}{P(\mathbf{x})}\right] \tag{4}$$

where $\mathbf{t}$ is the target point and $\mathbf{x}$ is the exit point.

Now, let's see what happens if two regions happen to overlap. Again, we first consider sampling the target point uniformly in the new region, and consider a target distribution which is uniform. Consider two regions which partly overlap. Due to the fact that we use the probability P_i (3), each volume element $\mathbf{dx}$ inside the regions has equal probability of being chosen. However, points located in the intersection will be a target twice as often as points outside the intersection. To compensate, *i.e.* to maintain detailed balance, we need to reject half of the proposed jumps into the intersection. In general, we check the number of regions that contain the exit point, $n(\mathbf{x})$, and similarly for the target point, $n(\mathbf{t})$. The appropriate fraction of moves that is to be accepted in order to maintain detailed balance is $\min[1, n(\mathbf{x})/n(\mathbf{t})]$. Combining this with the Metropolis acceptance probability 4 we find,

$$P_{accept} = \min\left[1, \frac{n(\mathbf{x})P(\mathbf{t})}{n(\mathbf{t})P(\mathbf{x})}\right] \tag{5}$$

Generalized Darted MCMC Sampler

Repeat until convergence

```
1. Draw a sample $u_1 \sim U[0,1]$.
2. if $u_1 > P_{check}$:
   perform one step of a local MCMC sampler.
3. if $u_1 < P_{check}$
   (a) Identify the number of regions $n(\mathbf{x})$ that contain
       the current sample.
   (b) if $n(\mathbf{x}) = 0$
       do nothing.
   (c) if $n(\mathbf{x}) > 0$
        i. Sample a new region according to $P_i$ (3).
       ii. Propose a location inside the new region (either
           deterministically or uniformly at random).
      iii. Identify the number of regions $n(\mathbf{t})$ that contain
           the proposed sample.
       iv. Draw a sample $u_2 \sim U[0,1]$.
        v. if $u_2 > P_{accept}$ (5)
           reject move.
       vi. if $u_2 < P_{accept}$ (5)
           accept move
```

Figure 1: The steps of our generalized darting sampler.

Putting everything together, we define the mode-hopping MCMC sampler explained in figure 1.

3.1 Elliptical Regions with Deterministic Moves

In the previous section we have uniformly sampled the proposed new location of the particle inside the target region. This is a very flexible method for which it is easy to prove detailed balance. However, a deterministic transformation can be tuned to map between points of roughly equal probability which is expected to improve the acceptance rate. Consider for instance the case that the energy surfaces near the regions is exactly quadratic and have the same height (*i.e.* their centers have equal probability). We can now define a transformation between ellipses that maps between points of equal probability resulting in a vanishing rejection rate. This is obviously not the case when we use uniform sampling.

We first consider the case of non-overlapping elliptical regions. Ellipses seem a natural choice, but the algorithm presented here is by no means restricted to it. For instance, the method is readily generalized to the use of rectangles as basic shapes. We'll parameterize an ellipse by a mean $\boldsymbol{\mu}$, a covariance $\boldsymbol{\Sigma}$ and a scale α, *i.e.* the ellipse is defined to be the equiprobability contour that is α standard deviations away from the mean. We will also need the eigenvalue decomposition of the covariance, $\boldsymbol{\Sigma} = \mathbf{U}\mathbf{S}\mathbf{U}^T$, where $\mathbf{S}$ is a diagonal matrix containing the eigenvalues denoted by $\{\sigma_i\}$. A deterministic transformation between two ellipses $1 \rightarrow 2$ is given by,

$$\mathbf{x}_2 = \boldsymbol{\mu}_2 - \mathbf{U}_2\mathbf{S}_2^{1/2}\mathbf{S}_1^{-1/2}\mathbf{U}_1^T(\mathbf{x}_1 - \boldsymbol{\mu}_1) \tag{6}$$

We note that this transformation would not leave a point invariant if we chose the second ellipse to be equal the first one, but mirrors it in the origin. Even though the transformation above is one-to-one, it does change the volume element $\mathbf{dx}$, implying that we need to take the Jacobian of the transformation into consideration. The intuitive reason for this is the same as in the previous section: more particles will be located in the larger ellipses resulting in more jumps to smaller ellipses than back, violating detailed balance. To compensate we sample the target ellipse again proportional to its volume, *i.e.* using (3), where

$$V_{ellipse} = \frac{\pi^{\frac{d}{2}} \alpha^d \prod_{i=1}^{d} \sigma_i}{\Gamma(1+\frac{d}{2})} \tag{7}$$

where $\Gamma(x)$ is the gamma-function with $\Gamma(x+1) = x\Gamma(x)$, $\Gamma(1) = 1$ and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.

We will now discuss how this algorithm can be generalized in case the ellipses overlap. Consider again two ellipses which partly overlap and a uniform target density. Consider a point that is located inside both ellipses, *i.e.* in the overlap (point 1). To apply the deterministic mapping, we first need to choose one of the two ellipses as a basis for the transformation. Unfortunately, an arbitrary rule such as the ellipse on top of the stack, or the one with the largest volume will result in a violation of detailed balance. Thus, we propose to pick the ellipse at random with equal probability. Now consider the image point under the mapping (point 2), choosing either the same ellipse (resulting in mirroring the point at the origin) or choosing the other ellipse. Assume point 2 is not located in the overlap. The probability of moving from $1 \to 2$ is $\frac{1}{4}$; a factor $\frac{1}{2}$ coming from the fact that we first choose with equal probability which ellipse will be used to define the transformation, and another factor $\frac{1}{2}$ because we sample the target ellipse using (3). However, in the other direction $2 \to 1$ the probability is $\frac{1}{2}$. Note that unlike the case of uniformly sampling a target point (see previous section) the probability of going from $2 \to 1$ is not doubled[2]. Thus, to rescue detailed balance we need to accept only half of the proposed moves from $2 \to 1$, or more generally $\min[1, n(\mathbf{x})/n(\mathbf{t})]$ with $n(\cdot)$ the number of ellipses containing a point. Combining this with the usual Metropolis acceptance rule applicable to general target densities, we arrive precisely at the rule in (5).

To summarize, the deterministic algorithm has precisely the same structure as algorithm in fig. 1, where in the transformation (6) ellipse 1 is chosen uniformly at random from all ellipses containing point 1 and ellipse 2 is chosen using (3) with V_i given by (7).

3.2 Mode-Hopping in Discrete State Spaces

A large number of practical problems is best described by probability distributions with discrete state spaces. It is therefore of importance to discuss this case as well. Fortunately, the extension is rather straightforward, the main difference being that "volumes" are to be replaced by "number of states inside a certain distance".

In this section we'll consider the Manhattan distance, but the algorithm is by no means restricted to that choice. Consider a discrete state s in some D dimensional space, where every dimension can take one of V values, e.g. $\mathbf{s} = [0, 3, 6, 1]$ for $D = 4$ and $V = 6$. The Manhattan distance between two states $\mathbf{s}_1$ and $\mathbf{s}_2$ is the total number of changes we need to make to transform one state into the other, or,

$$\mathcal{D}(\mathbf{s}_1, \mathbf{s}_2) = \sum_{i=1}^{D} |\mathbf{s}_1^i - \mathbf{s}_2^i| \tag{8}$$

[2] The reason is that for every target ellipse the image of the point under the mapping (6) is different. However, there are circumstances, e.g. when one ellipse is completely encircled by a larger one, that isolated points have the same image for two distinct target ellipses, resulting in violation of detailed balance. Since in the continuous case this set has measure zero, we will ignore it.

First consider the situation where no points are contained in two distinct regions and the regions have the same shape. Again, we have a choice of using a deterministic transformation, mapping states one-to-one to each other. For instance, if regions are defined to be the collection of all states that are at most a distance $\mathbf{d}$ away from a reference state $\mathbf{r}$, than we can use the offset $\mathbf{s}-\mathbf{r}$ to define the mapping: $\mathbf{s}_2 \to \mathbf{r}_2 + (\mathbf{s}_1 - \mathbf{r}_1)$. Since all regions have the same number of states, we can simply pick a target region uniformly at random.

The situation is slightly more complicated if we allow for regions with different numbers of states. It is clear that one-to-one mappings are now no longer possible. If one insists on a deterministic mapping many-to-one mappings are possible, but intricate acceptance rules will need to be designed to retain detailed balance. We will therefore proceed by using a random method, where the state in the target region is picked uniformly at random from all possible states in that region. In analogy with the continuous case, in order to maintain detailed balance, we need to pick the target region according to the distribution,

$$P_i = \kappa_i / \sum_j \kappa_j \tag{9}$$

where κ_i is the number of states contained in region i.

It is also easy to generalize this to overlapping regions. The same reasoning as in section 3 leads to the conclusion that a fraction min $[1, n(\mathbf{x})/n(\mathbf{t})]$ should be accepted where $n(\cdot)$ is the number of regions that contains a point. Finally, combining this with general target densities leads to the acceptance rule (5). The resulting MCMC algorithm is now very similar to the one in fig. 1, but with a different distance measure and a probability of picking a target region given by (9).

3.3 A Further Generalization

In the previous sections we have used distance measures to define regions between which the samples could "jump". This is geometrically appealing, but unnecessary for the algorithm to function properly. More generally, we can use a set of conditions that must be satisfied in order to be able to jump between these generalized regions. In order to maintain detailed balance we should however be able to determine the total number of states which satisfy each set of conditions. The probability (9) can then be used to pick a target region and the acceptance rule (5) can be used to accept or reject a randomly picked point from that region. Overlaps are also allowed in this case.

4 Learning Random Fields

The proposed mode-hopping algorithm can only be successful if we have advance information[3] about the expected location of regions of high probability. In the following two sections we discuss examples where this is indeed the case.

In the first example we consider a situation where we want to train a Random Field (RF) model from data. The general form of a RF is given by,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{-E(\mathbf{x};\boldsymbol{\theta})} \tag{10}$$

where $\boldsymbol{\theta}$ is a set of parameters that we try to infer given the data, $E(\mathbf{x};\boldsymbol{\theta})$ is the energy and $Z(\boldsymbol{\theta})$ the normalizing constant or partition function,

$$Z(\boldsymbol{\theta}) = \int \mathbf{d}\boldsymbol{\theta}\; e^{-E(\mathbf{x};\boldsymbol{\theta})} \tag{11}$$

[3] Adapting the Markov chain to include new regions on-line would violate the Markov assumption and is therefore not guaranteed to converge to the desired probability distribution.

We use the maximum likelihood criterium to define a cost function for finding the optimal setting of these parameters,

$$\mathcal{F} = -\frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{x}_n|\boldsymbol{\theta}) = \langle E(\mathbf{x};\boldsymbol{\theta})\rangle_{data} + \log Z(\boldsymbol{\theta}) \tag{12}$$

To minimize this "free energy" (negative log-likelihood) we need to compute its gradients,

$$\frac{d\mathcal{F}}{\mathbf{d}\boldsymbol{\theta}} = \left\langle \frac{dE(\mathbf{x};\boldsymbol{\theta})}{\mathbf{d}\boldsymbol{\theta}} \right\rangle_{data} - \left\langle \frac{dE(\mathbf{x};\boldsymbol{\theta})}{\mathbf{d}\boldsymbol{\theta}} \right\rangle_{model} \tag{13}$$

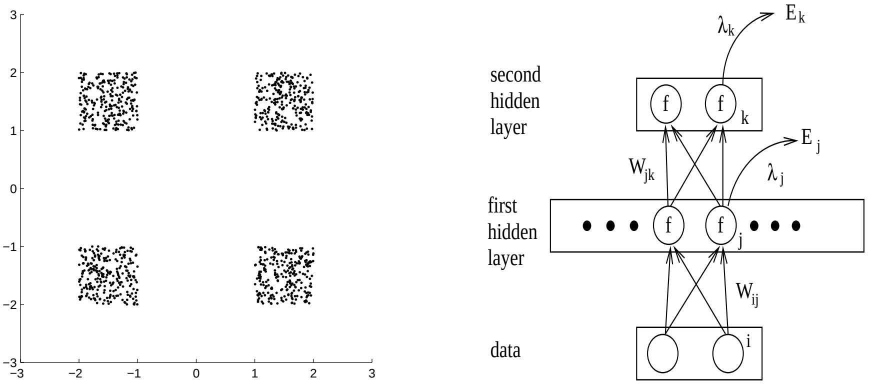
where the second term is equal to the negative derivative of the log-partition function w.r.t. $\boldsymbol{\theta}$. Note that the only difference between the two terms in (13) is the distribution which is used to average the energy derivative. In the first term we use the empirical distribution, *i.e.* we simply average over the available data-set. In the second term however we average over the model distribution as defined by the current setting of the parameters. Computing this second average analytically is typically too complicated, so approximations are needed instead. An unbiased estimate can be obtained by replacing the integral by a sample average, where the sample is to be drawn from the model $p(\mathbf{x}|\boldsymbol{\theta})$. In many cases MCMC is the only method available that can generate this sample set.

Imagine the target distribution $p(\mathbf{x}|\boldsymbol{\theta})$ has many modes and the Markov chain is initialized in one of them. Due to the energy barriers, we do not expect the chain to mix very well between the modes which results in very poor estimates of the second term in (13). Under the assumption that the modes of the distribution are located close to clusters of data-points, a viable strategy is to start the Markov chains at various different data-points in order to have some representative samples in each mode. This strategy is indeed used in *contrastive divergence learning* [9] where a Markov chain is initiated at each data-point and run for only a few steps (*i.e.* not to equilibrium) to generate distorted reconstructions of the data-point. Those reconstructions are subsequently used in the second term of (13) to compute an estimate of the energy derivative.

Even though we have arranged to generate samples in most relevant modes of the distribution, the fact that the samples do not properly mix between the modes results poor estimates of their relative importance (*i.e.* their relative heights). The mode-hopping extension of the MCMC sampler proposed in this paper can help resolve this problem by defining regions around data clusters between which the sampler jumps. In one limit one could imagine defining a a small spherical region around every data point, or an appropriate subset of all data points. An alternative possibility is to run a clustering algorithm as a preprocessing step and to define regions corresponding to each cluster. For example, a "mixtures of Gaussians" model could be trained using the "expectation maximization" algorithm with regions corresponding to the equiprobability contours α standard deviations away from the mean. Since these regions are elliptical, the deterministic mode-hopping algorithm described in section 3.1 may be used.

4.1 A Simple Illustrative Example

Figure 2 shows some two-dimensional training data and a model that was used to model the density of the training data. The model is an unsupervised, deterministic, feedforward neural network with two hidden layers of logistic units. The parameters of the model are the weights and biases of the hidden units and one additional scale parameter per hidden unit which is used to convert the output of the hidden unit into an additive contribution to the global energy. By using backpropagation through the model, it is easy to compute the derivatives of the global energy assigned to an input vector w.r.t. the parameters and it is also easy to compute the gradient of the energy w.r.t. each component of the input vector



(a) (b)

Figure 2: a) shows a two-dimensional data distribution that has four well-separated modes. b) shows a feedforward neural network that is used to assign an energy to a two-dimensional input vector. Each hidden unit takes a weighted sum of its inputs, adds a learned bias, and puts this sum through a logistic non-linearity to produce an output that is sent to the next layer. Each hidden unit makes a contribution to the global energy that is equal to its output times a learned scale factor. There are 20 units in the first hidden layer and 3 in the top layer.

(*i.e.* the slope of the energy surface at that point in dataspace), which is needed for hybrid Monte Carlo sampling.

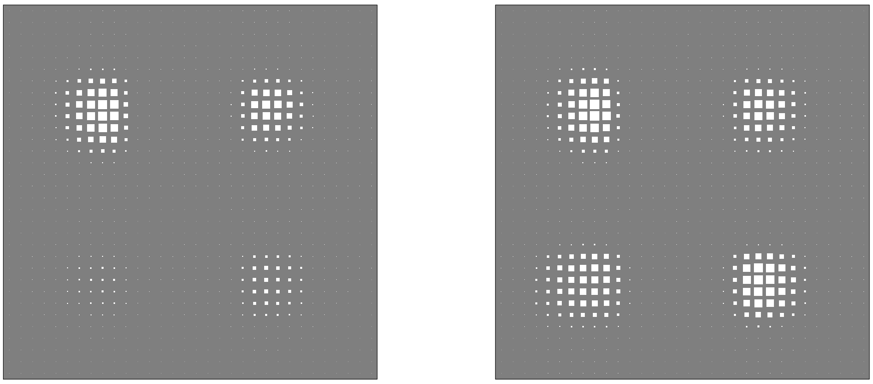
The model is trained on 1024 datapoints for 1000 parameter updates using equation 2. The reconstructions in (13) were obtained by briefly sampling ($T = 10$ time steps) from the model using HMC, where the chain for each reconstruction was initialized at the corresponding datapoint.

Figure 3a shows the probability density over the two-dimensional space that was learned by the network. Notice that the four minima are locally correct but are not all at the same height, so the model assigns much more probability mass to some minima than to others.

Figure 3b shows how the probability density is corrected by 10 parameter updates using a Markov chain that has been modified by adding an optional long-range jump at the end of each accepted trajectory. The details of the jump mechanism are slightly different than the algorithm proposed in this paper and can be found in[4] [8]. Thus, by allowing the samples to jump between the modes of the distribution, the learning procedure was able to assign the correct amount of probability mass to each mode.

Note that this example was included to illustrate and motivate the need for mode-hopping samplers in learning Markov random field models; not for testing and evaluating the generalized darting method proposed in this paper (which will be the subject of the next section). Indeed, due to the symmetry of the modes, we expect the spherical darting method of [1] to perform just as well as our generalized darting algorithm.

[4] The reason we have chosen to leave out the details of that algorithm is that it was found to behave inferior to the jump sampler proposed in this paper, but easier to implement and certainly good enough for the simple example presented here.



(a) (b)

Figure 3: (a) Shows the learned probability masses by using the HMC sampler, computed on a 32×32 grid in the dataspace. Some modes contain too little probability mass. (b) Shows the probability masses after including the long-range moves. The relative probability masses in the different modes have nicely balanced out.

5 An Application to Human Articulated Pose Estimation Using a Single Image

We explore the potential of the generalized darting method for *monocular* 3D human pose estimation given correspondences between the articulated joints of a subject in the image and the joints of a 3D articulated model. This problem is well adapted to illustrating the algorithm because the resulting energy surface is high-dimensional (around 35 parameters that defeat random sampling) and highly multimodal.[5] This is caused by body limbs that are subject to 'reflective' kinematic ambiguities (forwards *vs.* backwards slant in depth) and each such ambiguity is a local minimum (see fig. 5) under almost any problem modeling. In this context pose estimation and target tracking applications require a machinery for representing and propagating uncertainty and this requires density estimation and propagation under a static or dynamic series of observations. Particle filters [7, 10] have been one such successful approach that can accommodate multimodal distributions evolving under general dynamics and observation models with guaranteed asymptotic correctness (sampling fairness). Practically however, the sample-based representations are expensive and limited for good accuracy to 6-8 dimensions. This is partly due to the discrete search for high-probability regions that tends to be inefficient in high-dimensions[6] and prone to mode-trapping for multi-modal, ill-conditioned problems [23]. A different approach is to use a parametric mixture density representation and rely on continuous search to find the energy minima efficiently. These are used to build an approximative importance sampler. A variety of methods for locating and tracking these local minima and avoid trapping have been studied elsewhere [22, 23, 24]. However the above methods, while potentially good importance samplers for expectation calculations, will still not help with accelerating sampling from the *equilibrium distribution* and arguably not approximate the density well in the modal tails thus leading to very low correction weights. Gradient-based hybrid MCMC schemes could be effectively used instead [4]. This can significantly improve equilibrium around individual local minima basins but broken ergodicity caused by trapping still re-

[5] Monocular human pose estimation has applications for actor motion reconstruction and viewpoint synthesis from movie footage (where only one camera track is generally available) or for human-computer interaction.

[6] The factored sampling method is one particular importance sampler currently used.

mains a major problem [24]. Therefore, in this work we assume the minima position and local uncertainty structure is known and concentrate on using this prior information to accelerate fair sampling. Minima structure plays an important role due to ill conditioning. In our problem this is caused by lack of observability of parameter space directions that generate motions in depth with respect to the video camera ray of sight. This results in highly non-spherical (fig. 6a shows the covariance structure of a minimum), nonaligned minima covariances which further motivates our generalization of the classical darting scheme.

5.1 Sampling and Optimization Methods for 3D Human Pose Estimation and Tracking

One line of research has concentrated on either applying scene constraints [3] or problem dependent constraints [25, 19] to regularize the search space. Deutscher *et al* [3] use multiple cameras to avoid multimodality resulting from monocular projection ambiguities and derives an annealed sampling schedule within a particle filtering framework. Their layered method is based on factored sampling at a manifold of temperatures. A related approach has been suggested by Neal [16, 17]. In contrast, Neal also includes an additional importance sampling correction designed to improve mixing. Sidenbladh *et al* [19, 20] use particle filtering and concentrate on the construction of a variety of learned dynamical models (walking as well as more general classes) in order to generate predictions for good places to look during temporal tracking. Sminchisescu & Triggs [25] use problem dependent constraints about the forward/backwards pose ambiguity in the monocular case in order to explicitly enumerate various kinematic minima (a fast way to provide the input to an algorithm like ours). Choo & Fleet [2] combine particle filtering and hybrid Monte Carlo sampling to estimate 3D human motion, using a cost function based on joint re-projection error given input from motion capture data. For our experiments, we use a similar cost function and local gradient-based sampling moves but here we propose an algorithm to better deal with broken ergodicity aspects.

Other approaches have relied on more general search methods [22, 23, 24]. Sminchisescu & Triggs [22] track articulated 3D motion from monocular images using a combination of robust constraint-consistent local optimization and ‘oversized’ covariance scaled sampling to focus samples on probable low-cost regions. Aiming to search the parameter space in a more informed way they also propose methods for discovering new local minima by deterministically or probabilistically locating saddle points (transition regions) surrounding a known minimum basin [23, 24]. In this way surrounding basins become accessible and can be progressively explored.

5.2 Problem Modeling

This section describes the humanoid visual models used in our sampling experiments. For more details see [22, 21].

Representation: The human body model is constructed from ‘skeletons’ of articulated joints controlled by angular joint parameters. It also has ‘flesh’ built from three-dimensional ellipsoids with deformation parameters (these are only used for improved surface coverage during visual tracking and not important for the pose estimation experiments here, where we only concentrate on the joint angles). A typical model has about 30-35 effectively varying joint parameters $\boldsymbol{\theta}$.

The model is used as follows. Discretized model joint positions $\mathbf{u}_i$ on local coordinate systems for each body limb are transformed into global 3D points $\mathbf{p}_i(\boldsymbol{\theta}, \mathbf{u}_i)$, then into predicted image points $\mathbf{r}_i(\boldsymbol{\theta}, \mathbf{u}_i)$ using composite nonlinear transformations $\mathbf{r}_i(\boldsymbol{\theta}, \mathbf{u}_i) = \mathbf{P}(\mathbf{p}_i(\boldsymbol{\theta}, \mathbf{u}_i)) = \mathbf{P}(\mathbf{K}(\boldsymbol{\theta}, \mathbf{u}_i))$, where $\mathbf{K}$ represents a chain of rigid transformations that map different body links through the kinematic chain to their 3D position (see fig. 4), and

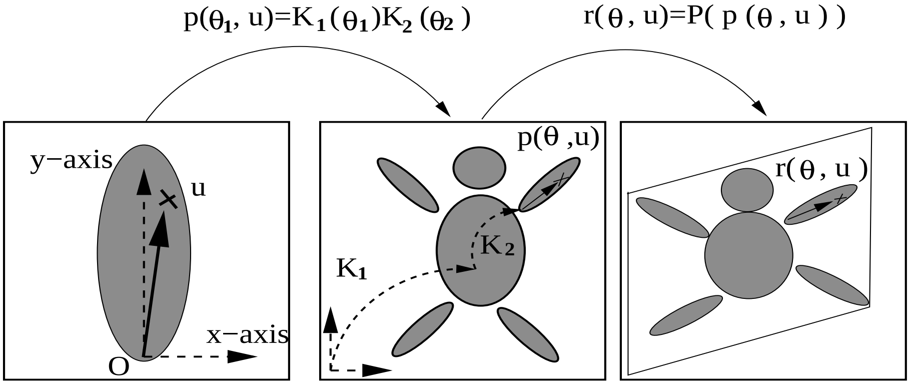


Figure 4: A simple model of a kinematic chain consisting of ellipsoidal parts. First, the feature $\mathbf{u}_i$, defined in its local coordinate frame, is mapped to a 3-D position, $\mathbf{p}_i(\boldsymbol{\theta}, \mathbf{u}_i)$, in the body model through a chain of transformations, $\mathbf{K}_i(\boldsymbol{\theta})$, between local coordinate systems. $\boldsymbol{\theta}$ are parameters that encode transformations (here rotation angles) between these reference frames. Finally, the 3-D coordinates $\mathbf{p}_i(\boldsymbol{\theta}, \mathbf{u}_i)$ are mapped into the image: $\mathbf{r}_i(\boldsymbol{\theta}, \mathbf{u}_i) = P(\mathbf{p}_i(\boldsymbol{\theta}, \mathbf{u}_i))$.

$\mathbf{P}$ represents perspective image projection. During model estimation, prediction-to-image Gaussian likelihoods are evaluated between each projected model joint $\mathbf{r}_i$ and the joint of the subject in the image $\mathbf{x}_i$ (these are the input data to the algorithm). The results are summed over all joints to produce the parameter space energy function. The cost is thus a function $e(\mathbf{x}_i|\boldsymbol{\theta})$ of the prediction errors $\Delta\mathbf{x}_i(\boldsymbol{\theta}) = \mathbf{x}_i - \mathbf{r}_i(\boldsymbol{\theta})$ between the model and the given input data. The cost gradient $\mathbf{g}_i(\boldsymbol{\theta})$ and Hessian $\mathbf{H}_i(\boldsymbol{\theta})$ are also computed and assembled over all observations. This will be necessary for second order local continuous optimization and hybrid Monte Carlo (HMC) step computations.

Energy Function: We aim for a probabilistic interpretation and optimal estimates of the model parameters by maximizing the total probability according to Bayes rule:

$$p(\boldsymbol{\theta}|\mathbf{X}) \;\propto\; p(\mathbf{X}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}) \;=\; \exp\left(-\textstyle\sum_i e(\mathbf{x}_i|\boldsymbol{\theta})\right)\, p(\boldsymbol{\theta}) \tag{14}$$

where $e(\mathbf{x}_i|\boldsymbol{\theta})$ is the cost density associated with observation i, the sum is over all observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, and $p(\boldsymbol{\theta})$ is the prior on the model parameters. The corresponding energy surface is defined as the negative log-likelihood for the total posterior probability:

$$E(\boldsymbol{\theta}|\mathbf{X}) \;=\; -\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \;=\; \textstyle\sum_i e(\mathbf{x}_i|\boldsymbol{\theta}) + E_p(\boldsymbol{\theta}) \tag{15}$$

For the experiments we have used the classical Langevin sampler (see section 2), in combination with long-range jumps using the spherical darting method and the generalized darting method.

Observation Likelihood: In the experiments below we used a simple product of Gaussian likelihoods for each model skeletal joint (assumed independent) with cost $e(\mathbf{x}_i|\boldsymbol{\theta}) = \Delta\mathbf{x}_i^2/2\sigma^2$. Thus, the negative log-likelihood for the observations is the sum of squared model joint re-projection errors. For the case study here it provides an interesting (and difficult to handle) degree of multi-modality owing to the kinematic complexity of the human model and the large number of parameters that are unobservable in a single monocular image.

Prior Distributions: The priors accommodated in the framework include parameter stabilizers that avoid singular distributions for hard to estimate but useful modeling parameters[7],

[7] *E.g.* some of the parameters at the end of the kinematic chains that generate rotations of limbs

terms for collision avoidance between body parts, and joint angle limits.[8] During estimation, the values and gradients/Hessians of the priors are evaluated and added to the cost and gradient/Hessians contributions from the observations.

5.3 Experiments

For the experiments shown here, we have selected 4 local minima corresponding to the left forearm and left calf in a monocular side view (see fig. 5). The local minima have relative volumes of $(0.16, 0.38, 0.10, 0.36)$ and corresponding energy levels $(4.41, 6.31, 7.20, 8.29)$.

The simulation enforces joint limit constraints using reflective boundary conditions, *i.e.* by reversing the sign of the normal momentum when it hits a joint limit. We found that this gives an improved sampling acceptance rate compared to simply projecting the proposed configuration back into the constraint surface, as the latter leads to cascades of rejected moves until the momentum direction gradually swings around.

We ran the simulation for $\Delta\tau = 0.1$, using the Langevin sampler (fig. 7a), the darting method with spherical covariances (fig. 7b) and the generalized darting method with deterministic moves (fig. 7c). In fig. 7 we show a part of a larger simulation that uses a lower jump probability $P = 0.03$ to diminish the number of jumps for illustrative purposes. From fig. 7 it is easy to see that the classical sampler gets trapped in the starting mode, and wastes all of its samples exploring it repeatedly. The spherical darting method explores the minima only 2 minima based on one successful long-range jump during 600 iterations. Finally the hopped method (right) explores more minima by combining local moves with non-local jumps that are accepted more frequently. Different minima are visited using 7 jumps. This could be visually checked as after each such jump, the sampler wanders around the energy levels associated to the new local minima.

We also run a large simulation for 10^5 steps with $\Delta\tau = 0.1$. The first 200 samples were discarded in order to let the chain reach equilibrium. With this increment, the sampler mixes fast within each minimum while still preserving good acceptance rates of 94% for local moves. We also use probability $P = 0.25$ for the darted moves. For generalized darting, we estimate local covariances as the inverse Hessians at each local minimum. For optimization we use a second-order damped Newton trust region method [5] where gradient and Hessians of the energy functions are computed analytically and assembled by using the chain rule with efficient back-propagation on individual kinematic chains. The covariance volume scaling factor α was set to unity whereas for classical darting we place spheres of unit radius around each minimum. The acceptance rate in the spherical case was $a_s = 1292/24,863 = 0.052$ whereas for the generalized darting case was $a_g = 9642/25,850 = 0.388$, an important improvement. We have also run a variety of smaller experiments and have found that the results are stable if we change the volume factor α by as much as 10%.

We additionally study the performance of this sampling regime in a different experiment based on 3 runs consisting of $20,000$ simulation steps each. Here, we compute the ergodic measure [1] that gives the rate of self-averaging in equilibrium calculations. Although self-averaging is a necessary but not sufficient condition for the ergodic hypothesis to be satisfied, it is still expected to give an intuition about the rate of parameter space sampling. In here, we have chosen the parameter-space configuration as a quantity to average (alter-

around their major axis may change the energy function very little and are likely to be close to singular.

[8] Given that we estimate pose from a *single* image and thus severely loose depth information, the priors are especially useful to avoid non-admissible model configurations, particularly in a side view (fig. 5). In such cases different limbs could penetrate eachother, thus violating the body physical constraints.

natively an ergodic measure based on other property, *e.g.* the energy could be used). This it is essentially an average over pair-wise differences between average parameter-space positions for trajectories initiated in different minima during a simulation. Specifically, the average parameter space position after S moves on a trajectory *initiated* at minimum a, containing configurations $\{\boldsymbol{\theta}_i^a, i = 1..S\}$ obtained[9] during a sampling run k:

$$d_k^a(S) = \frac{1}{S}\sum_{i=1}^{S} ||\boldsymbol{\theta}_i^a|| \tag{16}$$

and the ergodic measure is defined as the average between two trajectories initiated at different minima a and b in R runs[10]:

$$e(a,b,S,R) = \frac{1}{R}\sum_{k=1}^{R}[d_k^a(S) - d_k^b(S)]^2 \tag{17}$$

For good mixing over large trajectories we expect the ergodic measure to converge towards 0. In fig. 6, we plot the ergodic measure corresponding to a classical Langevin simulation with no jumps against one using the generalized scheme for $S = 20,000$ over $R = 3$ runs. One can see that the mixing behavior of the classical sampler is not satisfactory, perhaps reflecting the average parameter-space difference between the two local minima where the sampler gets trapped and repeatedly explores. On the contrary, in the generalized darted case, it is clear that the various minima are explored and the long-range parameter space distance self-averaging effect is observed.

6 Discussion

In this paper we have discussed a new Markov chain Monte Carlo sampler, that is able to effectively jump between modes in the target distribution while maintaining detailed balance. Our method is a generalization of “darting MCMC” where the basic jump regions may have an arbitrary irregular shape and moreover are allowed to overlap. Generalizations to discrete and more general domains are also discussed.

Apart from the darting method [1], other MCMC schemes that mix between distant modes can be found in the literature[11]. In “simulated tempering” [12], a new temperature random variable is introduced that extends the sample space in such a way that at high temperatures the energy function is much smoother. The temperature itself is also sampled via random walk. At high temperatures the Markov Chain mixes much faster between distant regions, while the samples acquired at $T = 1$ are the desired samples from the target distribution. Neal extended this idea to his “tempered transitions” method [15] that uses deterministic moves between low and high temperature regimes. In [28] the “normal kernel coupler” is proposed to sample from multi-modal distributions. The idea is to simulate N Markov chains in parallel with target distribution $p(\mathbf{x}) = \prod_i p_i(\mathbf{x})$, but to use proposal distributions based on a kernel estimate of all N particles. Finally, in [26] a generalized Metropolis-Hastings MCMC sampler is proposed that has the potential of incorporating deterministic optimization algorithms to locate local maxima in the target distribution. The inclusion of deterministic long range moves in a MCMC sampler for the purpose of computing the

[9] The trajectory may well include configurations inside minima basins other than a, but in a slight abuse of notation we will identify *both* the starting minima *and* the trajectory itself with the same letter.

[10] Note that there are two *different* simulations for each run k, one for a and another for b.

[11] We do not claim that the listed methods are an exhaustive overview of the literature. We apologize beforehand if important contributions have been omitted and would be grateful if they were being brought to our attention.

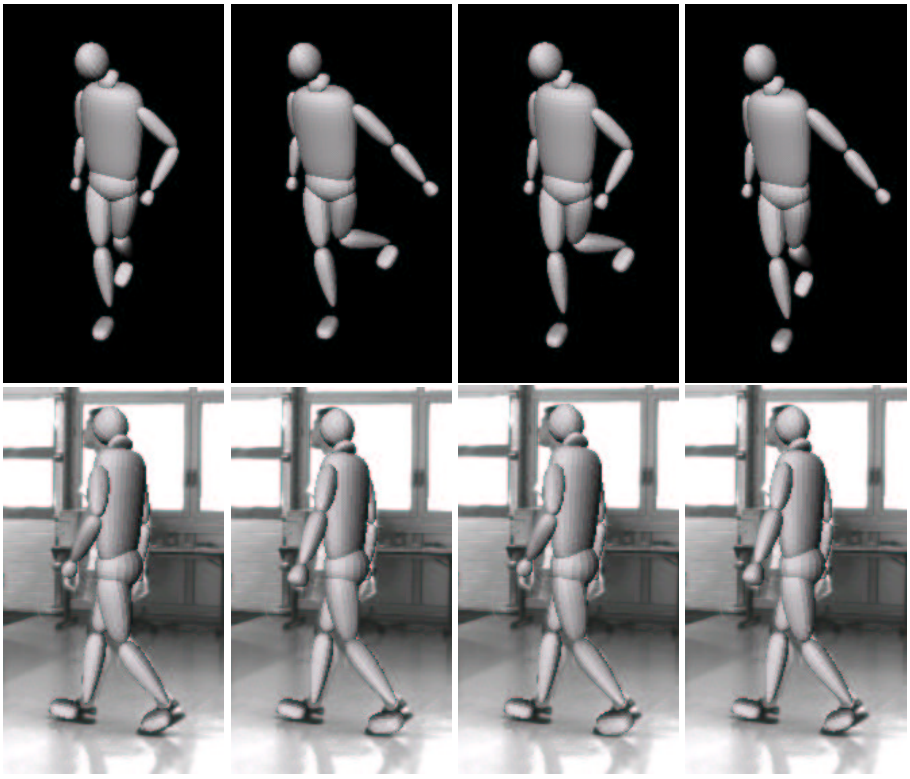


Figure 5: Four local minima of the energy function in (15), defined over 35 parameters (the human joint angles). Note the very different 3D poses and very similar image projections and thus similar cost.

free energy of a physical system can also be found in the physics and chemistry literature [27, 1, 18, 13, 11].

The main advantage of the proposed generalized darting method is that one can tune the shape of the jump regions to match the shape of the high probability regions of the target distributions. This should help to achieve an improved acceptance probability of attempted jumps between regions. Note however that we do not claim that our method is superior to all earlier schemes under all circumstances. In fact, we have only compared our method with the classical darting method and shown improved acceptance rates. No doubt, the various methods described above will have different properties for different target distributions, or in the presence of different amounts of prior knowledge about the target distribution. We have made no further attempts to explore these issues in this paper.

In the absence of sufficient prior knowledge of the position and shape of the modes of the target distribution, the darting framework may suffer from unacceptably high rejection rates for long range jumps. This problem will almost certainly be aggravated in high dimensions. The possibility to change the location and shape of the regions adaptively would be advantageous but difficult without violating the Markovian property of the chain. Clever ways around this obstacle do exist in the literature [6] but further research will be required to find out if they can be applied to the proposed generalized darting method.

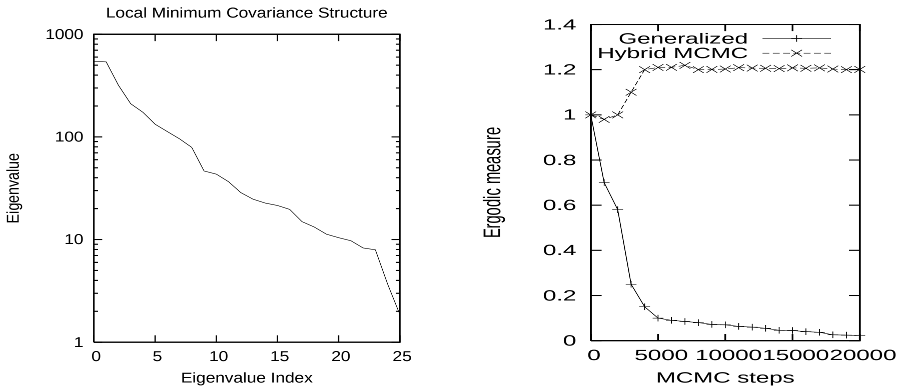


Figure 6: (a, left) Top 25 eigenvalues of the covariance matrix (corresponding to a 35 parameter model) for a local minimum shows the typical ill-conditioning of the monocular human pose estimation. (b, right) The ergodic measure compared for a classical Langevin gradient-based sampling scheme and the generalized darting method. The classical sampler doesn't mix well because the long-range energy difference between trajectories reflects the memory of the minima where they were initiated. The generalized method mixes much better and explores various minima so the parameter-space difference over long-term trajectories tends to zero (see text).

A Proof of Detailed Balance

To proof detailed balance between any pair of points in the sample space, we consider the following three possibilitites:

1. Both points are located outside any of the jump-regions.
2. One of the two points is located inside one or more jump-regions while the other one is located outside any of the regions.
3. Both points are located in one or more of the regions.

1: When both points are located outside any of the jump-regions detailed balance follows because of the Markov chain for the local moves is assumed to respect detailed balance. With probability P_{check} this Markov chain is interrupted to check if the particle is located inside a jump-region. But since both points under consideration are assumed to be located outside any jump-region this interruption will be symmetric and does not destroy detailed balance.

2: The particle located outside any jump-region follows its local dynamics (i.e. it is not interrupted) with probability $1 - P_{check}$. The particle inside one or more regions will also follow its local dynamics (i.e. it will not attempt a jump) with probability $1 - P_{check}$. With probability P_{check} the sampler decides to perform a check. But in that case the particle outside any region will stay put while the particle inside one or more regions will attempt a jump and will therefore never end up ouside the set of all regions. Thus detailed balance again holds.

3: We will prove the case of two points in possibly overlapping regions, where the jump points are sampled uniformly at random inside a target region. The prove for the deterministic case goes along similar lines (see section 3.1).

With probability $1-P_{check}$ we follow the local dynamics of the Markov chain which fullfils detailed balance by assumption. With probability P_{check} we initiate a jump to some other

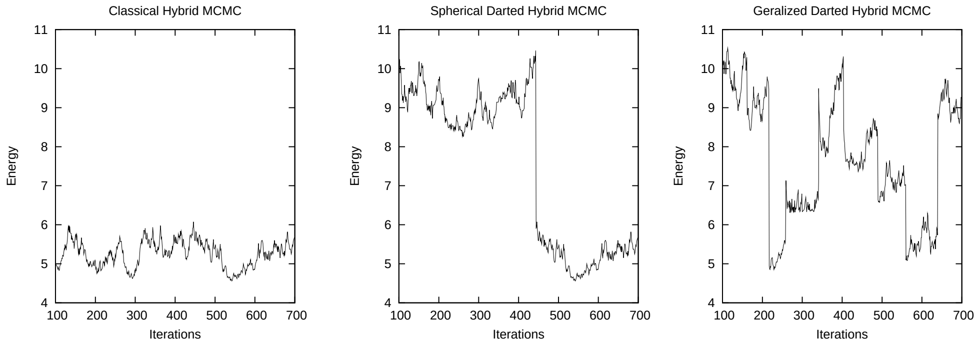


Figure 7: Classical hybrid Monte Carlo (left) gets trapped in the starting minimum. Spherical darting explores the minima more thoroughly and one can see that only 2 minima are visited during 600 iterations (only 1 successful jump). Finally the generalized darting method (right) explores the different minima by combining local moves with non-local jumps that are accepted more frequently. 8 local minima are visited via 7 jumps (note that after each jump the sampler explores its new local minimum for a while before the next jump).

point in some other region. Define A to be the set of regions that contain point 1 and B the set of regions that contain point 2. We now have,

$$p(\mathbf{x}_1)\ P(\mathbf{x}_1 \to \mathbf{x}_2) = p(\mathbf{x}_1)\ P_{check} \sum_{i \subset B} \frac{P_i}{V_i}\ P_{accept:1 \to 2} \tag{18}$$

$$= p(\mathbf{x}_1)\ P_{check}\ \frac{n(\mathbf{x}_2)}{\sum_j V_j}\ \min\left[1, \frac{p(\mathbf{x}_2) n(\mathbf{x}_1)}{p(\mathbf{x}_1) n(\mathbf{x}_2)}\right] \tag{19}$$

$$= P_{check}\ \frac{1}{\sum_j V_j}\ \min\left[p(\mathbf{x}_1) n(\mathbf{x}_2), p(\mathbf{x}_2) n(\mathbf{x}_1)\right] \tag{20}$$

$$= p(\mathbf{x}_2)\ P_{check}\ \frac{n(\mathbf{x}_1)}{\sum_j V_j}\ \min\left[1, \frac{p(\mathbf{x}_1) n(\mathbf{x}_2)}{p(\mathbf{x}_2) n(\mathbf{x}_1)}\right] \tag{21}$$

$$= p(\mathbf{x}_2)\ P_{check} \sum_{i \subset A} \frac{P_i}{V_i}\ P_{accept:2 \to 1} \tag{22}$$

$$= p(\mathbf{x}_2)\ P(\mathbf{x}_2 \to \mathbf{x}_1) \tag{23}$$

where P_i (equation 3) is the probability of jumping to region i and the factor $1/V_i$ is included because the target point is sampled uniformly at random inside this region. Thus, once again establish detailed balance.

Acknowledgments We thank Radford Neal and Yee-Whye Teh for helpful advice.

References

[1] I. Andricioaiei, J. Straub, and A. Voter. Smart Darting MonteCarlo. *J. Chem. Phys.*, 114(16), 2001.

[2] K. Choo and D. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *IEEE International Conference on Computer Vision*, 2001.

[3] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.

[4] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

[5] R. Fletcher. Practical Methods of Optimization. In *John Wiley*, 1987.

[6] Walter R. Gilks, Gareth O. Roberts, and Sujit K. Sahu. Adaptive Markov Chain Monte Carlo Through Regeneration. *Journal of the American Statistical Association*, 93(443):1045–1054, 1998.

[7] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Non-linear/Non-Gaussian State Estimation. *IEE Proc. F*, 1993.

[8] G. Hinton and M. Welling. Wormholes Improve Contrastive Divergence. In *Advances in Neural Information Processing Systems*, 2003.

[9] G.E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14:1771–1800, 2002.

[10] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 1998.

[11] C. Jarzynski. Targeted Free Energy Perturbation. Technical Report LAUR-01-2157, Los Alamos National Laboratory, 2001.

[12] E. Marinari and G. Parisi. Simulated Tampering: A New Monte Carlo Scheme. *Europhysics Letters*, 19(6), 1992.

[13] M. Miller and W. Reinhardt. Efficient Free Energy Calculations by Variationally Optimized Metric Scaling. *J. Chem. Phys.*, 113(17), 2000.

[14] R. Neal. Probabilistic Inference Using Markov Chain Monte Carlo. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[15] R. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366, 1996.

[16] R Neal. Annealed Importance Sampling. Technical Report 9805, Department of Statistics, University of Toronto, 1998.

[17] R. Neal. Annealed Importance Sampling. *Statistics and Computing*, 11:125–139, 2001.

[18] H. Senderowitz, F. Guarnieri, and W. Still. A Smart Monte Carlo Technique for Free Energy Simulations of Multiconformal Molecules. Direct Calculation of the Conformational Population of Organic Molecules. *J. Am. Chem. Society*, 117, 1995.

[19] H. Sidenbladh, M. Black, and D. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *European Conference on Computer Vision*, 2000.

[20] H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, 2002.

[21] C. Sminchisescu. *Estimation Algorithms for Ambiguous Visual Models–Three-Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*. PhD thesis, Institute National Politechnique de Grenoble (INRIA), July 2002.

[22] C. Sminchisescu and B. Triggs. Covariance-Scaled Sampling for Monocular 3D Body Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 447–454, Hawaii, 2001.

[23] C. Sminchisescu and B. Triggs. Building Roadmaps of Local Minima of Visual Models. In *European Conference on Computer Vision*, volume 1, pages 566–582, Copenhagen, 2002.

[24] C. Sminchisescu and B. Triggs. Hyperdynamics Importance Sampling. In *European Conference on Computer Vision*, volume 1, pages 769–783, Copenhagen, 2002.

[25] C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[26] H. Tjelmeland and B. K. Hegstad. Mode Jumping Proposals in MCMC. Technical report, Norwegian University of Science and Technology, Trondheim, Norway, 1999. Preprint Statistics No.1/1999.

[27] A. Voter. A Monte Carlo Method for Determining Free-Energy Differences and Transition State Theory Rate Constants. *J. Chem. Phys.*, 82(4), 1985.

[28] G. Warnes. The Normal Kernel Coupler: An adaptive Markov Chain Monte Carlo Method for Efficiently Sampling from Multi-modal Distributions.