Pattern Recognition 44 (2011) 2738–2748

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Generalized darting Monte Carlo

Cristian Sminchisescu [a,b,*], Max Welling [c]

[a] Institute for Mathematics (IMAR), 21 Calea Grivitei Street, 010702 Bucharest, Romania
[b] University of Bonn, Germany
[c] University of California, Irvine, United States

ARTICLE INFO

Article history:
Received 22 September 2008
Received in revised form
18 November 2010
Accepted 5 February 2011
Available online 17 February 2011

Keywords:
Markov chain Monte Carlo
Markov random fields
Darting
Constrained optimization
3D reconstruction
Human tracking

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]. We illustrate the method on learning Markov random fields and evaluate it against the spherical darting algorithm on a 'real world' vision application of inferring 3D human body pose distributions from 2D image information.

1. Introduction

It is well known that MCMC samplers have difficulty in mixing from one mode to the other because it typically takes many steps of very low probability to make the trip [2,3]. Recent improvements designed to combat random walk behavior, like Hybrid Monte Carlo and over-relaxation [4,2] do not solve this problem when modes are separated by high energy barriers. In this paper we show how to exploit knowledge of the location of the modes to design a MCMC sampler that mixes properly between them.

We 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 [5,6]. 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, the data itself is representative of the low energy mode of a well fitted model.

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 and an auxiliary variable formulation of the method appear 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 – learning human models and estimating 3D human body poses from 2D image information.

2. Markov chain Monte Carlo sampling

Imagine we are given a probability distribution $p(\mathbf{x})$ with $\mathbf{x} \in \mathcal{X} \subset R^d$ 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 one proposed in 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:

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

* Corresponding author.
E-mail addresses: Cristian.Sminchisescu@ins.uni-bonn.de (C. Sminchisescu), Welling@ics.uci.edu (M. Welling).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.02.006

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, $dE(\mathbf{x})/d\mathbf{x}$, 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 $-dE(\mathbf{x})/d\mathbf{x}$ and accept using (1):

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

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.[1]

The Langevin method is a special case of a more general sampling technique called Hybrid Monte Carlo (HMC) sampling [4,2,7]. 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 [2] 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 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 will use uniform sampling in this section. The deterministic case will be treated in the next section. Also, to simplify the discussion we will 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 us 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

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

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

[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 $\Delta\to 0$.