How Does CONDENSATION Behave with a Finite Number of Samples?

O. King and D.A. Forsyth

Computer Science Division, U.C. Berkeley, Berkeley, CA 94720, USA
king@math.berkeley.edu, daf@cs.berkeley.edu
WWW home page: http://www.cs.berkeley.edu/~daf

Abstract. CONDENSATION *is a popular algorithm for sequential inference that resamples a sampled representation of the posterior. The algorithm is known to be asymptotically correct as the number of samples tends to infinity. However, the resampling phase involves a loss of information. The sequence of representations produced by the algorithm is a Markov chain, which is usually inhomogeneous. We show simple discrete examples where this chain is homogeneous and has absorbing states. In these examples, the representation moves to one of these states in time apparently linear in the number of samples and remains there. This phenomenon appears in the continuous case as well, where the algorithm tends to produce "clumpy" representations. In practice, this means that different runs of a tracker on the same data can give very different answers, while a particular run of the tracker will look stable. Furthermore, the state of the tracker can collapse to a single peak — which has non-zero probability of being the wrong peak — within time linear in the number of samples, and the tracker can appear to be following tight peaks in the posterior* even in the absence of any meaningful measurement. *This means that, if theoretical lower bounds on the number of samples are not available, experiments must be very carefully designed to avoid these effects.*

1 Introduction

The Bayesian philosophy is that all information about a model is captured by a *posterior* distribution obtained using Bayes' rule:

$$\text{posterior} = P(\text{world}|\text{observations}) \propto P(\text{observations}|\text{world})P(\text{world})$$

where the prior $P(\text{world})$ is the probability density of the state of the world in the absence of observations. Many examples suggest that, *when computational difficulties can be sidestepped*, the Bayesian philosophy leads to excellent and effective use of data (e.g. [2,7]. The technique has been widely used in vision.

Obtaining some representation of the world from a posterior is often referred to as *inference*. One inference technique that is quite general is to represent the posterior by drawing a large number of samples from that distribution. These samples can then be used to estimate any expectation with respect to that posterior.

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 695–709, 2000.

1.1 Condensation or "Survival of the Fittest" or Particle Filtering

The most substantial impact of sampling algorithms in vision has been the use of resampling algorithms in tracking. The best known algorithm is known as *condensation* in the vision community [1], *survival of the fittest* in the AI community [8] and *particle filtering* in the control literature [4,9]. In Condensation, one has a prior density $p(x)$ on the state of the system, a process density $p(x_t|x_{t-1})$, and an observation density $p(z|x)$. The state density at time t conditioned on the observations till then, $p(x_t|Z_t)$, is represented by a set of weighted samples $\{(s_t^{(n)}, \pi_t^{(n)}), n = 1, \ldots, N\}$. To update this representation for time $t+1$, one draws N points from the samples $s_t^{(n)}$, with replacement, with probability proportional to their weights $\pi_t^{(n)}$. These points are perturbed in accordance with the process density to get the new samples $s_{t+1}^{(n)}$. The new weights $\pi_{t+1}^{(n)}$ are computed by evaluating $p(z|x)$ for each of the new samples x in light of the new observation z. Condensation is fast and efficient, and is now quite widely applied (INSPEC produces 16 hits for the combination "condensation" and "computer vision" for the last 4 years).

Condensation can represent multi-modal distributions with its weighted samples, and can hence maintain multiple hypotheses when there is clutter or when there are other objects mimicking the target object. It can run with bounded computational resources in near real-time by maintaining the same number N of samples at each step.

There are asymptotic correctness results for Condensation essentially asserting that, for a fixed number of frames T and desired precision, there is a number N of samples so that the sampled representation at time t approximates the true density at time t to within the desired precision for $t = 1, 2, \ldots, T-1, T$ (e.g. [1]). There is little information on how large N should be[1]; in [1], examples are given of 800 frames tracked with 100 samples (p. 18) and 500 frames tracked with 1500 samples (p. 19).

In what follows, we show that iterations of the Condensation algorithm form a Markov chain, whose state space is quantized representations of a density. We show strong evidence that this Markov chain has some unpleasant properties. The process of resampling tends to make samples collapse to a single cluster, putting substantial weight on "peaky" representations. When the true density is multimodal, even if the mean computed from this clumpy density is unbiased, the movement of the clump may be slow. In turn, this means that:

- expectations computed with the representation maintained by Condensation have high variance so that different runs of the tracker can lead to very different answers (section 3.1; section 3.2) ;

[1] "Note that convergence has not been proved to be uniform in t. For a given fixed t, there is convergence as $N \to \infty$ but nothing is said about the limit $t \to \infty$. In practice this could mean that at later times t larger values of N may be required, though that could depend also on other factors such as the nature of the dynamical model." [1], p. 27

- expectations computed with the representation maintained by a particular instance of CONDENSATION have low variance so that in a particular run of the tracker, it will look stable (section 2.1; section 3.1);
- the state of the tracker may collapse to a single peak within time roughly linear in the number of samples (section 2.1; section 2.1; section 2.2; section 3.1);
- the peak to which the state collapses may bear no relationship to the dynamic model (sections 2.1 and 2.2);
- and the tracker will appear to be following tight peaks in the posterior *even in the absence of any meaningful measurement* (section 3.2).

Some of these phenomena have been noticed as a practical matter in the particle filtering literature, where they are referred to as “sample impoverishment” [3], and others have well-understand analogs in population genetics, but we present the first explanation we are aware of in the context of motion tracking.

1.2 Markov Chains

A **Markov chain** is a sequence of random variables X_k with the property that $P(X_n|X_1, \ldots, X_{n-1}) = P(X_n|X_{n-1})$. One can think of this important property as “forgetting”; the distribution for the next state of the chain depends only on the current state and not on any other past state. The chain is referred to as a **homogeneous** Markov chain if $P(X_n|X_{n-1})$ is independent of n.

If the random variables are discrete and have a countable state space, we can write a matrix $\mathcal{P}$ called the **state transition matrix** whose i, j'th element is

$$p_{ij} = P(X_n = j|X_{n-1} = i)$$

Notice that $p_{ij} \geq 0$ and

$$\sum_k p_{ik} = 1$$

because the entries of $\mathcal{P}$ are probabilities. This matrix describes the state transition process. In particular, assume that the random variable X_{n-1} has probability distribution $\mathbf{f}$. Then the random variable X_n has probability distribution $\mathbf{g}$, where

$$\mathbf{g}^T = \mathbf{f}^T \mathcal{P}$$

Now if the Markov chain is homogeneous, and X_1 has probability distribution $\mathbf{h}_1$, then X_n has probability distribution $\mathbf{h}_n$ where

$$\mathbf{h}_n^T = \mathbf{h}_{n-1}^T \mathcal{P} = (\mathbf{h}_{n-2}^T \mathcal{P})\mathcal{P} = \mathbf{h}_1^T \mathcal{P}^{n-1}$$

If the random variables are defined on a continuous domain D, and have probability density functions, then we can construct an operator analogous to the state transition matrix. In particular, we consider a function $\mathcal{P}$ on $D \times D$ with the property that

$$\mathcal{P}(u, v)dudv = P(X_n \in [v, v + dv]|X_{n-1} \in [u, u + du])$$

Notice that $\mathcal{P}(u, v) \geq 0$ for all u, v and that

$$\int_D \mathcal{P}(u, v)dv = 1$$

Now if X_{n-1} is a random variable with probability distribution function $p(X_{n-1})$, then the probability distribution function of X_n is

$$\int_D \mathcal{P}(u, v)p(u)du$$

A particularly interesting case occurs when the distribution of X_n and that of X_{n-1} are the same. This distribution is known as a **stationary distribution**. Markov chains on both discrete and continuous spaces can have stationary distributions. On a discrete space, a stationary distribution $\mathbf{p}$ has the property that

$$\mathbf{p}^T = \mathbf{p}^T \mathcal{P}$$

and on a continuous space, a stationary distribution $p(u)$ has the property that

$$\int_D \mathcal{P}(u, v)p(u)du = p(v)$$

A Markov chain is not guaranteed to have a unique stationary distribution. In particular, there may be states, known as **absorbing** states, that the chain cannot leave. In this case, for each absorbing state the chain has a stationary distribution that places all weight on that absorbing state.

2 Sample Impoverishment in the Discrete Case

If the process density in Condensation is such that the samples aren't perturbed after the resampling step, then the state space is effectively discrete, since no points which weren't in the original batch of samples will ever be introduced. We can regard the state space as a set of bins, and the samples as weighted balls placed in these bins. Each stage of the inference process (tracking, in most applications) moves these balls from bin to bin, then re-weights the balls based on the new observation. The probability that a ball will go into a particular bin at time $k + 1$ is proportional to the combined weight of the balls in that bin at time k. As a result, once a bin is empty, it can never again contain a ball. Once all the balls lie in a particular bin (which is guaranteed to happen with non-zero probability), the representation is stuck in this state.

2.1 A Two-State Problem

Assume that we have a state space that consists of two distinct points. This could arise in tracking from the situation in figure 1. There is a stationary object ($p(x_t|x_{t-1}) = \delta_{x_{t-1}}(x_t)$) at x=1, but due to the mirrors it appears on the image plane at positions -1 and 1. Similarly, if the object were at x=-1, it would appear on the image plane at positions -1 and 1. Thus there is no way to disambiguate the positions -1 and 1 (or more generally -x and x) based on observations. We can model this by making one observation z at each stage at which we find, exactly, either the object or its reflection, and using $p(z|x) = .5\delta_x(z) + .5\delta_{-x}(z)$.

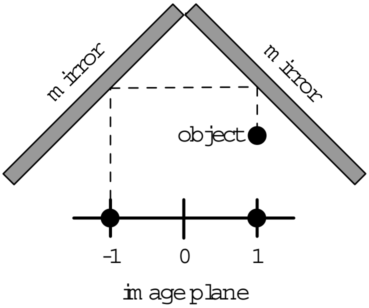

Fig. 1. Geometry for the model of section 2.1. A stationary object is reflected in a mirror, and is "tracked" — the tracker receives an exact measurement of the position of either the point or its reflection with equal probability. In that section, we show that a CONDENSATION tracker will lose track of either this object or its mirror reflection in a time linear in the number of samples maintained.

Start with a prior density $p_0(x) = .5\delta_1(x) + .5\delta_{-1}(x)$, indicating the point is equally like to be at positions -1 and 1. We should then have $p(x_t|Z_t) = p_0(x)$ for each t because the object is stationary and each observation z_t is equally consistent with the object being at -1 and 1.

We now apply a CONDENSATION tracker to estimate $p(x_t|Z_t)$. Let N be the number of samples used, and assume for convenience that N is even. First note that given our prior density and our process density, each of the sample points $s_t^{(n)}$ will be either -1 or 1. Given our observation density, each of these points will have the same weight, $\pi_t^{(n)} = 1/N$. Thus we can characterize the sampled representation at time t by a single number, X_t, which is the number of samples that are at 1 (the remaining $N - X_t$ samples are at -1). X_t represents the density

$$p(x) = \frac{X_t}{N}\delta_1(x) + \frac{N - X_t}{N}\delta_{-1}(x)$$

so the sampled representation accurately represents the true density if $X_t = N/2$ for all t.

X_t is a random variable, and the sequence $X_1, X_2, \ldots$ is a homogeneous Markov chain (because $Pr(X_t|X_1, \ldots, X_{t-1}) = Pr(X_t|X_{t-1})$). This Markov chain has the $N+1$ states $0, \ldots, N$ and, as the samples at time t are constructed by drawing N points from the samples at time $t-1$, with replacement and with equal weight, we have

$$Pr(X_t = j|X_{t-1} = i) = \binom{N}{j} \left(\frac{i}{N}\right)^j \left(\frac{N-i}{N}\right)^{N-j}$$

Losing track of an item The dynamics of the CONDENSATION algorithm guarantee that, with probability one, the sequence of X_t's for our example will eventually be the constant sequence 0 or the constant sequence N. The states 0 and N are absorbing. Since $P(i, 0) + P(i, N) > (1/2)^N$ for all i, the expected number of steps before the chain is absorbed in one of these states is easily seen to be less than 2^N. This is, however, a gross overestimate: the expected absorb time is of the order N steps, as shall be addressed presently.

The time to lose track The matrix $\mathcal{P}$ is also the transition matrix for the Wright-Fisher model of population genetics, and has been investigated since the 1920's. The samples at -1 and 1 correspond to two kinds of alleles in a population with fixed size, random mating, and non-overlapping generations. In this context the expected absorb time corresponds to the number of generations until one allele is lost entirely from the population as a result of "genetic drift."[2]

Let the vector $\mathbf{w}$ correspond to having half the samples in each mode. In [5], it is shown that the $N+1$ eigenvalues of $\mathcal{P}$ are $1, 1, (N-1)/N, (N-1)(N-2)/N^2, \ldots, (N-1)!/N^{N-1}$. Thus $\mathcal{P}$ has a basis of eigenvectors so we can write

$$\mathbf{w} = \pi + c_2\lambda_2\mathbf{v}_2 + \cdots + c_n\lambda_n\mathbf{v}_n$$

[2] "If a population is finite in size (as all populations are) and if a given pair of parents have only a small number of offspring, then even in the absence of all selective forces, the frequency of a gene will not be exactly reproduced in the next generation because of sampling error. If in a population of 1000 individuals the frequency of "a" is 0.5 in one generation, then it may by chance be 0.493 or 0.0505 in the next generation because of the chance production of a few more or less progeny of each genotype. In the second generation, there is another sampling error based on the new gene frequency, so the frequency of "a" may go from 0.0505 to 0.501 or back to 0.498. This process of random fluctuation continues generation after generation, with no force pushing the frequency back to its initial state because the population has no "genetic memory" of its state many generations ago. Each generation is an independent event. The final result of this random change in allele frequency is that the population eventually drifts to p=1 or p=0. After this point, no further change is possible; the population has become homozygous. A different population, isolated from the first, also undergoes this random genetic drift, but it may become homozygous for allele "A", whereas the first population has become homozygous for allele "a". As time goes on, isolated populations diverge from each other, each losing heterozygosity. The variation originally present within populations now appears as variation between populations." [11], p. 704

where λ_i are eigenvalues of the transition matrix $\mathcal{P}$, $\mathbf{v}_i$ are eigenvectors of that matrix, and π is a stationary distribution corresponding to a superposition of absorbing states. Now after k transitions of the chain, the probability distribution on the state is

$$\mathbf{w}^T\mathcal{P}^k = \pi + c_2\lambda_2^k\mathbf{v}_2 + \cdots + c_n\lambda_N^k\mathbf{v}_N$$

so

$$|\mathbf{w}^T\mathcal{P}^k \ - \pi| \leq C\lambda_2^k$$

where

$$C = |c_2||\mathbf{v}_2| + \cdots + |c_N||\mathbf{v}_N|$$

So, we have that after k steps

$$|\mathbf{w}^T\mathcal{P}^k - \pi| \leq C(\frac{N-1}{N})^k$$

where π is a stationary distribution (all balls in one bin) with eigenvalue one. Thus the second eigenvalue determines the asymptotic rate with which the probability that the chain has been absorbed approaches 1, and also the rate at which the variance of the representation that CONDENSATION reports collapses, as figure 2 illustrates.

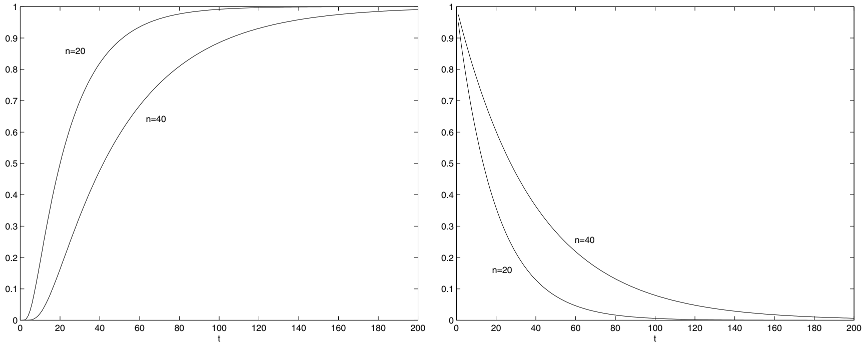

Fig. 2. On the *left*, the probability that all N samples belong to the same mode after k steps, graphed against k, assuming that the chain is started with $N/2$ samples from each mode. There are two curves, one for $N = 20$ and one for $N = 40$. On the *right*, a graph showing the variance of the posterior density estimated by CONDENSATION from the N samples after k steps. The correct variance at any stage is known to be 1. The estimated variance goes down, because the samples collapse to one mode. However, comparing these estimates from step to step would suggest that the estimate was good. There are two curves, one for $N = 20$ and one for $N = 40$. All graphs were computed numerically from powers of $\mathcal{P}$.

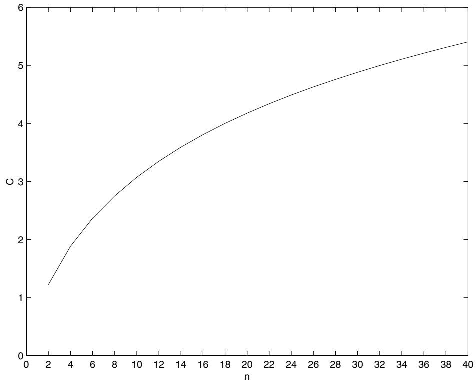

Fig. 3. We have that $|\mathbf{w}^T \mathcal{P}^k - \pi| \leq C((N-1)/N)^k$, where π is a stable distribution (all balls in one bin) with eigenvalue one. To obtain a useful finite time bound we need to know something about C. Above is a graph of C as a function of N, computed numerically; the graph suggests that C grows no faster than linearly in N.

While λ_2 tells us something about the asymptotic convergence to π, for this to be a useful finite time bound we need to know something about C. Figure 3 shows a graph of C as a function of N, computed numerically. If, as the figure suggests, C is a sublinear function of N, then since $(1-1/N)^N \leq 1/e$

$$|\mathbf{w}^T \mathcal{P}^{(aN)} - \pi| \leq C(\frac{N-1}{N})^{aN} \leq C(\frac{1}{e})^a$$

This means that an arbitrarily small probability that the chain is out of an absorbing state can be obtained in a number of steps of order $N \log C$ where $C < N$. Direct computations suggest that the factor of $\log C$ can be dispensed with entirely (figure 4).

A more natural measure of convergence may be the expected number of steps to reach an absorbing state. Although an exact formula for finite N remains elusive, it is known (e.g., [5]) by using a continuous time approximation that as the number of samples N goes to infinity, the expected time to be absorbed when starting with j samples at 1 and $N-j$ samples at -1 is asymptotically

$$-2N\{\frac{N-i}{N}\ln(\frac{N-i}{N}) + \frac{i}{N}\ln(\frac{i}{N})\}$$

In particular, when starting with $N/2$ samples at -1 and at 1, the expected time to be absorbed is asymptotically $(2\ln 2)N \approx 1.4N$. This is good approximation of the expected absorb time for small N as well, as demonstrated in figure 4.

Note that while this linear absorb time may not overly concern population geneticists, for whom the the time between the generations of interest may be many years, in Condensation there are perhaps 30 "generations" per second — one for each frame of video — so with a hundred samples, modes may be lost after only a few seconds.

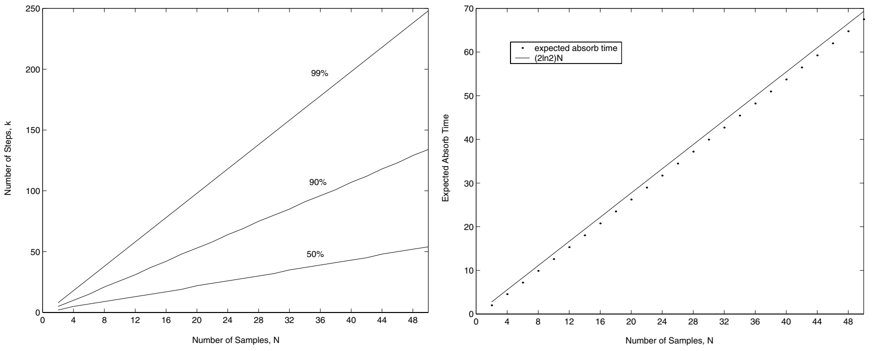

Fig. 4. The time to lose a mode is roughly linear in the number of samples. On the *left*, a graph of the number of steps required for the probability that all samples belong to the same mode to exceed 50%, 90%, and 99%, as a function of the number of samples N, obtained by evaluating $\mathbf{w}^T P^k$. On the *right*, a graph of the expected absorb time for $N = 2, 4, \ldots, 50$ samples when starting with half the samples in each mode, computed numerically. The solid line is the asymptotic expected absorb time $(2 \ln 2)N$.

2.2 Multiple States

We can consider a similar process with three, four, or in general b bins, instead of two. With b bins and N samples, the number of states in the Markov chain is $\binom{N+b-1}{b-1}$, corresponding to the number of distinct b-tuples $(i_1, \ldots, i_b)$ with nonnegative integer coordinates summing to N. The transition probability is

$$P((i_1, \ldots, i_b), (j_1, \ldots, j_b)) = \frac{1}{N^N} \binom{N}{j_1, \ldots, j_b} i_1^{j_1} \cdots i_b^{j_b}$$

The $\binom{N+b-1}{b-1}$ eigenvalues of the above P are

$$1, (N-1)/N, (N-1)(N-2)/N^2, \ldots, (N-1)!/N^{N-1}$$

with multiplicities

$$b, \binom{b}{2}, \binom{b+1}{3}, \ldots, \binom{b+N-2}{N}$$

respectively.

Since these are the same eigenvalues as for the two-bin case (though with different multiplicities) we may expect the same qualitative asymptotic behavior. Simulation results bear out this view (figure 5).

The expected time for all the samples to collapse into a single bin when starting with X_i samples from the i'th bin is asymptotically

$$-2N\{\sum_{i=1}^{b} \frac{N - X_i}{N} \ln(\frac{N - X_i}{N})\}$$

In particular, starting with N/b samples in each bin, the mean absorb time is $2N(b-1)\ln(\frac{b}{b-1})$.

For proofs, and further analysis of the multi-bin case, see [6].

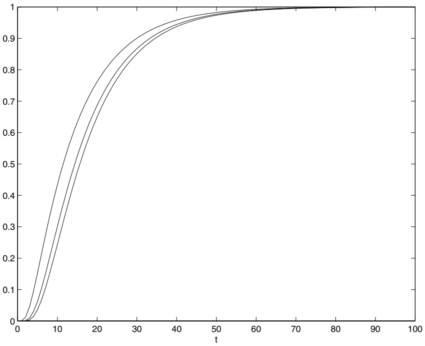

Fig. 5. When repeatedly resampling from a density with b discrete modes, the number of modes represented by the samples is nonincreasing, and the representation moves rapidly to an absorbing state where all samples lie in a single mode . The figure shows results from a chain using twelve samples, obtained by computing powers of $\mathcal{P}$. The graph shows the probability that all twelve samples are from the same mode as a function of the number of transitions k, after starting with an equal number of samples from each mode. The three curves, from left to right, are for 2,3,and 4 modes.

If we start with an equal number of samples from each mode, considerations of symmetry dictate an equal probability of all the samples ending up in either mode; more is possible: As $E(X_{k+1}|X_k) = E(X_k)$, the sequence of X_k's forms a martingale. Then by the optional stopping theorem [10], the probability that all the samples end up in a given mode is proportional to the number of samples that started in that mode. An analogous result holds when there are more than two modes, as can be seen by considering one mode in opposition with the rest of the modes combined. A consequence of this fact is that if a spurious mode can start with non-negligible fraction of the samples, it has a similarly non-negligible probability of usurping all the samples and suppressing the true mode.

3 Bad Behaviour in Continuous Spaces

The above examples were discrete to make the analysis more straightforward, but the same phenomena are visible in the continuous case. Slightly perturbing the samples will make them distinct, so that the Markov chain will no longer have an absorbing state, and the observations may vary, so that the chain will no longer be homogeneous. However, there will still be bad behaviour.

Recall that the chain's domain is sampled representations of probability densities. If a set of samples are clustered together closely, resampling these samples will tend to produce a cluster that is near the original cluster.

Two phenomena appear: Firstly, modes are lost (section 3.1), as in the discrete example; secondly, the algorithm can produce data that strongly suggest a mode is being tracked, even though this isn't happening (section 3.2). We show two examples that illustrate the effects. The examples are on a compact domain; all gaussian process densities are windowed to have support extending 2 standard deviations to either side of the mode, and particles are reflected off boundaries at -2 and 2 (-1 and 1 for the uniform observation density examples of section 3.2)

3.1 Losing Modes while Tracking Almost Stationary Points

We simulated a system with gaussian diffusion, in the same setup as in figure 1. We used the process density

$$P(x_t|x_{t-1}) \propto \exp(-(x_t - x_{t-1})^2/2(.05)^2)$$

and observation density

$$P(z|x) \propto \exp(-(x-z)^2/2\sigma^2) + \exp(-(x+z)^2/2\sigma^2)$$

The observation density represents a slightly defocussed observation which still anticipates finding the object and its mirror image. Various values of the parameter σ were employed, including $\sigma = \infty$, in which the observation density is uniform and observations are consequently completely uninformative.

The motion of an object starting at $x_0 = 1$ and undergoing a gaussian random walk was simulated; defocussed observations of this data were simulated to give a set of measurements to run the algorithm. The CONDENSATION algorithm was simulated for 50 steps on the sequence of observations using 100 samples, with diffusion and observation densities the same as those used to generate the point positions and observations, so any misbehavior is intrinsic to CONDENSATION itself, and not due to a bad estimate of the dynamics. CONDENSATION was initialized with half the samples at -1 and half at 1. As figure 6 indicates, modes are lost quite quickly. The loss of these modes results in a fall in the variance of the representation of the posterior (figure 6).

To continue the analogy with genetics, the gaussian diffusion plays a role akin to mutation, in preserving diversity among the samples. But the diffusion is small relative to the separation between the modes, so the diffusion of samples from one mode to the other would take many steps. Since the perilously small observation density puts points between the modes at a severe selective disadvantage, such a journey is highly improbable. Consequently, all the samples tend to cluster in a neighborhood of one of the modes.

3.2 Gaining Modes without Measurements

The state space is now the interval $[-1, 1]$; we supply a small diffusion process, and use a uniform observation density — i.e. there is no information at all about

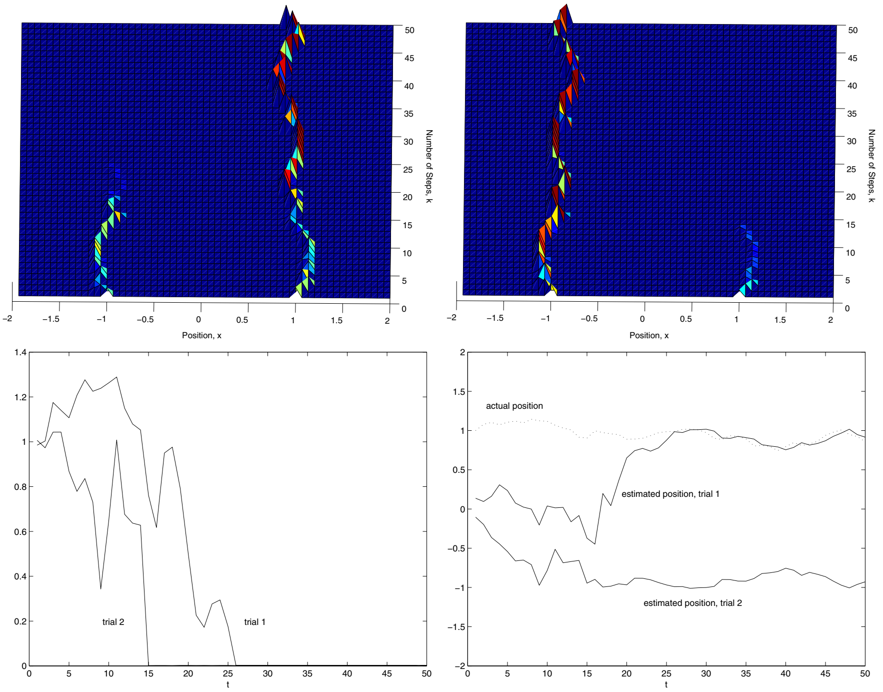

Fig. 6. On the top line, we show the Condensation representation of the prior density for time $k+1$ for the example of section 3.1, where the posterior has two modes, plotted for two sample runs, using 100 samples. In the first run the mode near -1 is lost after around 25 steps, and in the second run the mode near 1 is lost after around 15 steps. The *bottom left* figure shows the variance computed for the representation at each time step for the two runs; this variance declines because the samples collapse to one or another of the modes. The *bottom right* figure graphs the means computed from the weighted samples at each time step. Notice that for a single run, the mean estimate looks rather good (it hardly changes from stage to stage) but across runs, it looks bad (section 4).

where objects are. In this case, if the samples representing the posterior at step k are close together, then they will almost certainly be close together at step $k+1$ (because the probability of a sample moving a long distance with a small diffusion process is very small). This suggests (but does not prove) that in this case the Condensation algorithm yields an Markov chain whose stationary distribution has most of its weight on "clumped" representations.

The effect is very noticeable in a simulation. The Condensation algorithm was simulated for 100 steps using 20 samples, with diffusion given by $P(x_t|x_{t-1}) \propto \exp(-(x_t - x_{t-1})^2/2(.05)^2)$, observation density $P(z|x)$ uniform on $[-1,1]$, and initial distribution uniform on $[-1,1]$, yielding the results of figure 7. Notice that quite tight clumps of samples appear (in different places in each run) suggesting quite falsely that the tracker is actually tracking something.

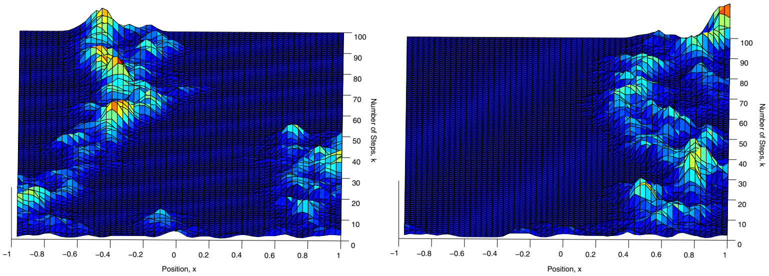

Fig. 7. The CONDENSATION algorithm was simulated for 100 steps using 20 samples, on the problem described in section 3.2, where no information is available about the position of an object and so the posterior should be uniform at every step. The plots above show the density estimated from the samples at each stage t, in two trial runs. Note that the probability quickly becomes concentrated, though it should remain uniformly distributed, and that the region in which it becomes concentrated differs from run to run; this effect makes CONDENSATION experiments very difficult to evaluate (section 4).

4 Discussion

While a representation of a posterior may use N samples, these samples may not be very effectively deployed (for example, if all are clumped close together). The resampling phase in CONDENSATION guarantees that there is a non-zero probability of losing modes and a very small probability of regaining them. This means that the effective number of samples goes down with each resampling step. Generally, this effect is governed by the dynamics of the Markov chain; unusually, one wishes the chain *not* to burn in (and perhaps reach an absorbing state). This means that to be able to use CONDENSATION with a finite number of samples and a guarantee of the quality of the results, one must be able to bound the second eigenvalue of the Markov chain *below*. This is sometimes as difficult as bounding it *above*, which is required to guarantee good behaviour from Markov chain Monte Carlo (e.g. [12]).

Nothing here should be read as a suggestion that CONDENSATION not be used, just that, like other sampling algorithms, it should be used very carefully. Generally, bounds will not be available, so that the algorithm's usefulness depends on designing experiments to take into account the possible effects of sample impoverishment.

Sample impoverishment makes experiments difficult to evaluate, because very poor estimates of a posterior may look like very good estimates. Representations of a probability distribution are mainly used to compute expectations (e.g. the center of gravity of a tracked object, etc.) as a weighted sum over samples. A standard technique for checking the quality of an estimate of an expectation is to look at the variance of these estimates. Now assume that we are tracking a stationary object with CONDENSATION; the estimate of the object's center of

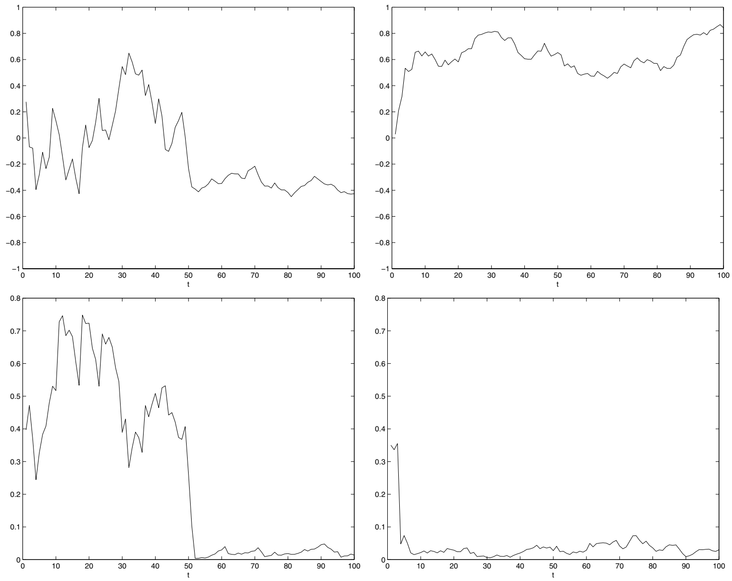

Fig. 8. The top two plots below show the mean of the samples at each time step for the two distinct trial runs for the problem of section 3.2. (Since the density should be uniform at each stage, the mean should be 0.) The bottom two plots below show the variance of the samples at each time step for the two trial runs. (Since the density should be uniform at each stage, the variance should be 2/3.) The tight clusters and slow drift of the clusters makes it look like there's an object being tracked even though we have no idea whatsoever where the object is.

gravity from frame to frame may have low variance, while the actual variance — which is obtained by looking at the estimates obtained by starting the algorithm in different places — is high (c.f. the last three sentences of the second footnote). This effect appears in both the discrete (figure 2) and continuous cases (figures 6; 8).

Another approach to combating this effect is to use fewer resampling steps (as in the SIS/SIR algorithm of [9], where an estimate of the effective number of samples is used); this probably involves using a more heavily constrained dynamical model so that fewer resampling steps are required. Finally, one might generate new samples occasionally.

5 Acknowledgements

Thanks to the referees for their helpful suggestions.

References

1. A. Blake and M. Isard. Condensation - conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.
2. B.P. Carlin and T.A. Louis. *Bayes and empirical Bayes methods for data analysis.* Chapman and Hall, 1996.
3. J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for non-linear problems. *IEEE Proc. Radar, Sonar and Navigation*, 146(1):2–7, 1999.
4. A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Cambridge University, 1998. CUED/F-INFENG/TR310.
5. W.J. Ewens. *Population Genetics.* Methuen, 1969.
6. W.J. Ewens. *Mathematical Population Genetics.* Springer, 1979.
7. A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis.* Chapman and Hall, 1995.
8. K. Kanazawa, D. Koller, and S. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc Uncertainty in AI*, 1995.
9. J.S. Liu and R. Chen. Sequential monte-carlo methods for dynamic systems. Technical report, Stanford University, 1999. preprint.
10. J.R. Norris. *Markov Chains.* Cambridge University Press, 1997.
11. D.T. Suzuki, A.J.F. Griffiths, J.H. Miller, and R.C. Lewontin. *An Introduction to Genetic Analysis.* W.H. Freeman, 1989.
12. L. Tierney. Introduction to general state-space markov chain theory. In W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors, *Markov chain Monte Carlo in practice.* Chapman and Hall, 1996.