MONTE CARLO SMOOTHING FOR NON-LINEARLY DISTORTED SIGNALS

William Fong and Simon Godsill

Signal Processing Group, University of Cambridge,
Cambridge, CB2 1PZ, U.K.
wnwf2@eng.cam.ac.uk sjg@eng.cam.ac.uk

ABSTRACT

We develop methods for Monte Carlo filtering and smoothing for estimating an unobserved state given a non-linearly distorted signal. Due to the lengthy nature of real signals, we suggest processing the data in blocks and a block-based smoother algorithm is developed for this purpose. In particular, we describe algorithms for de-quantisation and de-clipping in detail. Both algorithms are tested with real audio data which is either heavily quantised or clipped and the results are shown.

1. INTRODUCTION

In this paper we apply Monte Carlo smoothing to non-linearly distorted signals. To fix the notation, consider the standard Markovian state-space model.

$$\begin{array}{lll} x_{t+1} & \sim & f(x_{t+1}|x_t) \quad \text{State evolution density} \\ y_{t+1} & \sim & g(y_{t+1}|x_{t+1}) \quad \text{Observation density} \end{array}$$

where $\{x_t\}$ are unobserved states of the system and $\{y_t\}$ are observations made over some time interval $t \in \{1, \ldots, T\}$. $f(.|.)$ and $g(.|.)$ are pre-specified state evolution and observation densities which may be non-Gaussian and involve non-linearity.

1.1. Particle Filter

A primary concern in many state-space inference problems is sequential estimation of the filtering distribution $p(x_t|y_{1:t})$. Updating of the filtering distribution can be achieved in principle using the standard filtering recursions

$$p(x_{t+1}|y_{1:t}) = \int p(x_t|y_{1:t}) f(x_{t+1}|x_t) dx_t$$

$$p(x_{t+1}|y_{1:t+1}) = \frac{g(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{1:t})}{p(y_{t+1}|y_{1:t})}.$$

Similarly, smoothing can be performed recursively backwards in time using the smoothing formula

$$p(x_t|y_{1:T}) = \int p(x_{t+1}|y_{1:T}) \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} dx_{t+1}.$$

In practice these filtering and smoothing computations can only be performed in closed form for linear Gaussian models using the Kalman filter-smoother and for finite state-space hidden Markov models. Here we focus on Monte Carlo particle filters [1, 2], in which the filtering distribution is approximated with an empirical distribution formed from point masses, or particles,

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \ \sum_{i=1}^{N} w_t^{(i)} = 1, w_t^{(i)} \geq 0$$

where $\delta(.)$ is the Dirac delta function and $w_t^{(i)}$ is a weight attached to particle $x_t^{(i)}$. Particles at time t can be updated efficiently to particles at time $t+1$ using importance sampling and selection methods. In this paper, we assume that a forward sweep of particle filtering using an appropriate proposal distribution for the state has already been performed on the entire dataset, generating weighted particles $\{x_t^{(i)}, w_t^{(i)}; i = 1, \ldots, N, t = 1, \ldots, T\}$ (see [1, 2] for details).

1.2. Particle Smoother

A simple and efficient method for generating realisations from the entire smoothing density $p(x_{1:T}|y_{1:T})$ using particulate approximation has been developed [3]. Sample realisations are obtained using the following factorisation

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1:T}, y_{1:T}) \quad (1)$$

where, given the particulate approximation to $p(x_t|y_{1:t})$ and using the Markovian assumptions of the model, we can write,

$$\begin{aligned} p(x_t|x_{t+1:T}, y_{1:T}) &= p(x_t|x_{t+1}, y_{1:t}) \quad (2) \\ &= \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} \\ &\propto p(x_t|y_{1:t}) f(x_{t+1}|x_t) \\ &\approx \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(x_t - x_t^{(i)}) \end{aligned}$$

with the modified weights

$$w_{t|t+1}^{(i)} = \frac{w_t^{(i)} f(x_{t+1}|x_t^{(i)})}{\sum_{j=1}^{N} w_t^{(j)} f(x_{t+1}|x_t^{(j)})}. \quad (3)$$

This revised particulate distribution can now be used to generate states successively in the reverse-time direction, conditioning upon future states. The algorithm proceeds as follows,

Algorithm 1 — *Sample realisations*

- Choose $\tilde{x}_T = x_T^{(i)}$ with probability $w_T^{(i)}$.
- For $t = T-1$ to 1 :
  - Calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\tilde{x}_{t+1}|x_t^{(i)})$ for each $i = 1, ..., N$;
  - Choose $\tilde{x}_t = x_t^{(i)}$ with probability $w_{t|t+1}^{(i)}$.
- $\tilde{\mathbf{x}}_{1:T} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_T)$ is an approximate realisation from $p(x_{1:T}|y_{1:T})$.

Further independent realisations are obtained by repeating this procedure as many times as required.

2. BLOCK-BASED PARTICLE SMOOTHER

For smoothing, it is necessary to store the particle histories which is not appropriate for audio signals due to the storage capacity required. We modify the general smoothing algorithm to process a lengthy dataset in blocks, which reduces the amount of stored information significantly.

A lengthy time series is divided into two non-overlapping blocks, with T_1, T_2 marking the end of each block. A filtering and smoothing analysis is performed for each block of data independently and M realisations are generated from the smoothing density

$$\tilde{x}_{1:T_1}^{(i)} \sim p(x_{1:T_1}|y_{1:T_1})$$
$$\tilde{x}_{T_1+1:T_2}^{(j)} \sim p(x_{T_1+1:T_2}|y_{1:T_2})$$

with $i, j \in \{1, \ldots, M\}$. We are interested in generating realisations from the entire smoothing density, which can be factorised as follows

$$p(x_{1:T_2}|y_{1:T_2}) = p(x_{1:T_1}|x_{T_1+1:T_2}, y_{1:T_2}) p(x_{T_1+1:T_2}|y_{1:T_2})$$

where, using the Markovian assumption, we can write,

$$\begin{aligned} & p(x_{1:T_1}|x_{T_1+1:T_2}, y_{1:T_2}) \\ &= \frac{p(x_{T_1+1}|x_{1:T_1}, y_{1:T_1}) p(x_{1:T_1}|y_{1:T_1})}{p(x_{T_1+1}|y_{1:T_1})} \\ &\propto f(x_{T_1+1}|x_{T_1}) p(x_{1:T_1}|y_{1:T_1}) \end{aligned}$$

For each realisation from block 2, $\tilde{x}_{T_1+1:T_2}^{(j)}$, we can generate a realisation from the entire smoothing sequence by regarding $\tilde{x}_{1:T_1}^{(i)}$ as particles

$$p(x_{1:T_1}|\tilde{x}_{T_1+1:T_2}^{(j)}, y_{1:T_2}) \approx \sum_{i=1}^{M} w_{T_1|T_1+1}^{(i|j)} \delta(x_{1:T_1} - \tilde{x}_{1:T_1}^{(i)})$$

with

$$w_{T_1|T_1+1}^{(i|j)} \propto f(\tilde{x}_{T_1+1}^{(j)}|\tilde{x}_{T_1}^{(i)})$$

$\tilde{x}_{1:T_1}^{(i)}$ is then joined to $\tilde{x}_{T_1+1:T_2}^{(j)}$ with probability $w_{T_1|T_1+1}^{(i|j)}$.

Algorithm 2 — *Block-based smoother*

- Divide the lengthy times series into non-overlapping blocks, with $T_1, \ldots, T_R$ marking the end of each block.
- Perform particle filtering and smoothing analysis on the first block, and generate $\tilde{x}_{1:T_1}^{(i)}$ for $i = 1, \ldots, M$.
- For $r = 2$ to R:
  - Using algorithm 1, generate M realisations of $\tilde{x}_{T_{r-1}+1:T_r} \sim p(x_{T_{r-1}+1:T_r}|y_{1:T_r})$.
  - For each $\tilde{x}_{T_{r-1}+1:T_r}^{(j)}$, calculate the transition probability $w_{T_{r-1}|T_{r-1}+1}^{(i|j)} \propto f(\tilde{x}_{T_{r-1}+1}^{(j)}|\tilde{x}_{T_{r-1}}^{(i)})$ for each $\tilde{x}_{T_1:T_{r-1}}^{(i)}$, $i, j = 1, \ldots, M$;
  - Choose $\tilde{x}_{1:T_{r-1}}^{(j)} = \tilde{x}_{1:T_{r-1}}^{(i)}$ with probability $w_{T_{r-1}|T_{r-1}+1}^{(i|j)}$ and set $\tilde{x}_{1:T_r}^{j} = [\tilde{x}_{1:T_{r-1}}^{(j)}, \tilde{x}_{T_{r-1}+1:T_r}^{(j)}]$.

In order to avoid degeneracy, we suggest a moderately large value of M. After processing each frame, all information regarding the weighted particles can be discarded and thus the storage capacity required is reduced significantly when $M \ll N$.

3. AUDIO MODELS

Speech signals are inherently time-varying in nature, and any realistic representation should thus involve a model whose parameters evolve over time. One such model is the time-varying autoregression (TVAR)

$$z_t = \sum_{i=1}^{p} a_{t,i} z_{t-i} + e_t \quad (4)$$

Here $a_t = [a_{t,1}, \ldots, a_{t,p}]$ is the p^{th} order AR coefficient vector and e_t is the Gaussian excitation at time t. For our simulations, a Gaussian random walk model is assumed for the log-variances $\phi_{e_t} = \log(\sigma_{e_t}^2)$

$$f(\phi_{e_t}|\phi_{e_{t-1}}, \sigma_{\phi_e}^2) = \mathcal{N}(\mu_{\phi_t}, \sigma_{\phi_e}^2) \quad (5)$$

where $\mu_{\phi_t} = \log(\alpha\sigma^2_{e_{t-1}})$ and α is a coefficient just less than 1.

For the time variation in a_t, we choose to work in the time-varying partial correlation (PARCOR) coefficient domain [3, 4]. Here each reflection coefficient ρ_t must be constrained to the interval (-1,+1) in order to ensure strict stability. The constrained PARCOR random walk model is

$$f(\rho_t|\rho_{t-1},\sigma_a^2) \propto \begin{cases} \mathcal{N}(\rho_{t-1},\sigma_a^2 I) & \text{if } \max\{|\rho_{t,i}|\} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

4. APPLICATION

We adapt the general Monte Carlo particle filter and smoother to enhance audio signal quality. We first develop a general algorithm to estimate the unobserved true state given a non-linearly distorted observation using the particle filter and smoother.

Consider a general deterministic distortion process

$$y = \mathcal{F}(z)$$

where $\{z\}$ is the original signal, $\{y\}$ is the distorted signal, and $\mathcal{F}(.)$ is the distortion function. For audio signals, non-linear distortion processes like quantisation [5] and clipping, involve a many-to-one mapping, so its inverse process

$$\mathcal{F}^{-1}(y) = \{z;\ \mathcal{F}(z) = y\}$$

is a one-to-many mapping and provides range of possible values that the input might have taken. In terms of probability distributions we have

$$p(y|z) = \delta(y - \mathcal{F}(z)) \tag{7}$$

since the mapping is deterministic.

The following proposal distribution is used for generating state realisations in the Monte Carlo filter

$$q(z_t, a_t|y_{1:t}, z_{t-p:t-1}, a_{t-1}) = p(z_t|y_{1:t}, z_{t-p:t-1}, a_t)p(a_t|a_{t-1})$$

Expanding $p(z_t|y_{1:t}, z_{t-p:t-1}, a_t)$, we get

$$p(z_t|y_{1:t}, z_{t-p:t-1}, a_t) \propto p(y_t|z_t)p(z_t|z_{t-p:t-1}, a_t) \tag{8}$$

As $p(y_t|z_t)$ is a delta function which is non-zero only for $z_t \in \mathcal{F}^{-1}(y_t)$ and since $p(z_t|z_{t-p:t-1}, a_t)$ is a Gaussian density function (equation 4), $p(z_t|y_{1:t}, z_{t-p:t-1}, a_t)$ is a truncated Gaussian distribution.

The importance weight associated with each particle is then calculated as follows

$$\begin{aligned} w_t &\propto p(y_t|a_t, z_{t-p:t-1}) \\ &= \int p(y_t|z_t)p(z_t|a_t, z_{t-p:t-1})dz_t \\ &= \int_{z_t\in\mathcal{F}^{-1}(y_t)} p(z_t|a_t, z_{t-p:t-1})dz_t \end{aligned} \tag{9}$$

In this calculation, the current state z_t is marginalised in order to improve the statistical stability of our filter.

4.1. de-Quantisation

For digital signal processing, a signal is represented as discrete in both time (due to sampling) and value (due to quantisation). The quantisation process, $\mathcal{Q}$, rounds the signal to the nearest predefined step. The feasible region for the inverse process is thus

$$y_t - \frac{\Delta}{2} < z_t \le y_t + \frac{\Delta}{2}$$

and hence equation 9 becomes

$$w_t \propto \int_{y_t-\frac{\Delta}{2}}^{y_t+\frac{\Delta}{2}} p(z_t|a_t, z_{t-p:t-1})dz_t$$

$$equation = \Phi\left(y_t + \frac{\Delta}{2}; \mu_{zt}, \sigma^2_{e_t}\right) - \Phi\left(y_t - \frac{\Delta}{2}; \mu_{zt}, \sigma^2_{e_t}\right)$$

where Φ is the Gaussian cumulative density function and $\mu_{zt} = \sum_{i=1}^{p} a_{i,t}z_{t-i}$.

The most obvious method for generating samples of z_t from the truncated Gaussian (equation 8) is to draw samples from the full distribution and reject those lying outside the bounds. The acceptance rate is unacceptably small if the feasible region lies far in the tails of the distribution. In this case, we suggest sampling the region using rejection sampling with the proposal distribution being a trapezium.

4.2. de-Clipping

Clipping occurs when an input signal goes beyond the dynamic range of a system. Those parts of the signal which exceed the limit will be clipped to the clipping threshold, $\pm\tau$. The feasible region for the inverse clipping process is

$$\begin{aligned} z_t &\ge \tau && \text{for } y_t \ge \tau \\ z_t &= y_t && \text{for } -\tau < y_t < \tau \\ z_t &\le -\tau && \text{for } y_t \le -\tau \end{aligned}$$

In this case equation 9 becomes

$$w_t \propto \begin{cases} 1 - \Phi\left(\tau; \mu_{zt}, \sigma^2_{e_t}\right) & \text{for } y_t \ge \tau \\ p(z_t = y_t|a_t, z_{t-p:t-1}) & \text{for } -\tau < y_t < \tau \\ \Phi\left(-\tau; \mu_{zt}, \sigma^2_{e_t}\right) & \text{for } y_t \le -\tau \end{cases}$$

where $\mu_{zt} = \sum_{i=1}^{p} a_{i,t}z_{t-i}$.

5. RESULTS

A section of speech data representing the word "reward" is used to test the de-quantisation and de-clipping algorithms.

In both cases, the suggested algorithms are found to be effective. Audible improvements in audio quality and noticeable improvement in signal-to-noise ratio are observed.

The data specifications are as follows, the TV-PARCOR model order is $p = 6$ and the fixed hyperparameters used are $\sigma_a^2 = 0.001$, $\alpha = 0.955$ and $\sigma_{\phi_e}^2 = 0.01$.

5.1. de-Quantisation

The speech data is 4-bit quantised with no added noise. Due to the lengthy nature of the audio signal, the quantised signal is analysed in blocks of size 500 using the block-based particle smoother (algorithm 2). The number of particles, N, used to approximate the posterior distribution is 500 and prior boosting [1] is employed to improve particle quality. After a forward sweep of the particle filter, smoothing analysis is performed to generate $M = 50$ smoothed realisations of the signal.

A typical frame showing the original signal, quantised signal and the reconstructed signal is presented in Figure 1. The reconstructed signal is found by taking the average over all the smoothed realisations. Comparing the quantised signal and reconstructed signal, there is audible improvement. For the SNR, it improves from $13.7dB$ to $16.8dB$.

5.2. de-Clipping

The clipping threshold $\pm\tau$ is chosen so that about 20% of the data is clipped. A forward sweep of particle filter is applied to the entire dataset. $N = 500$ particles are used. Prior boosting is applied, to reflect the high posterior uncertainty in the clipped region. Finally, smoothing is applied to generate $M = 10$ realisations.

A typical frame showing the original signal, clipped signal and the reconstructed signal is presented in Figure 2. Compared with the clipped signal, the SNR improves from $5.7dB$ to $13.8dB$.

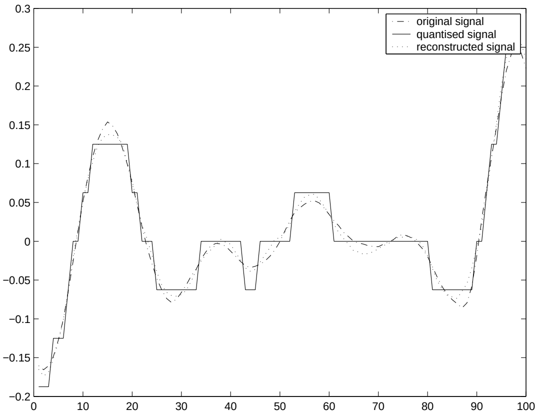

Figure 1: A typical frame showing the original, quantised and reconstructed signals

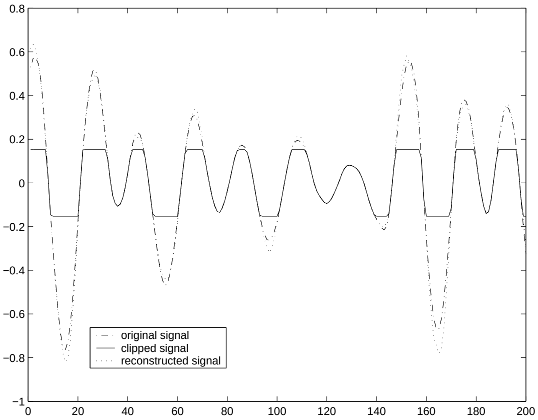

Figure 2: A typical frame showing the original, clipped and reconstructed signals

6. CONCLUSION

We develop the block-based particle smoother and adapt the Monte Carlo filtering and smoothing algorithm to estimate the unobserved state given a non-linearly distorted signal. The algorithms are tested against real audio signals and encouraging results are obtained. Further tests will be conducted and results will be published in due course.

7. REFERENCES

[1] A. Doucet, N. de Freitas and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[2] A. Doucet, S. J. Godsill and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistic and Computing*, 10:197–208, 2000.

[3] A. Doucet, S. J. Godsill and M. West. Monte Carlo filtering and smoothing with application to time-varying apectral estimation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 701–704, 2000.

[4] B. Friedlander. Lattice filters for adaptive processesing. *Proc.IEEE*, 70(8):829–867, August 1982.

[5] P. T. Troughton and S. J. Godsill. MCMC methods for restoration of quantised time series. In *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, volume 2, pages 447–451, 1999.