Bayesian Inference and Signal Processing: Building a Signal Interpolator Using a Gibbs Sampler

3rd May 2012

1 Outline

This project aims to investigate the methods used in Bayesian inference in the study of signal processing. The Bayesian approach exists in contrast to *frequentist* approach in the different way that the unknown parameters which describe a population are considered. A frequentist considers model parameters as fixed (unknown) quantities, where as the Bayesian statistician considers the unknown parameters as having a probability distribution associated with them, as well as a prior probability on the parameters *before* any data is taken. Using Bayes theorem the posterior probability density of the model parameters (given the data) is derived [2]. These methods have significant computational complexity and hence the feasibility of using Bayesian models has increased in recent times [1].

The purpose of this project is to develop a Gibb's sampler which is capable of restoring missing data from an signal representing an noisy audio signal.

1.1 Procedure

The investigation is divided up into 5 sections

- A simple, single dimension, Gibb's sampler was first developed to model a noisy straight line, where m, c, σ are determined.
- This sampler is then extended to multiple dimensions to interpolate samples from sine wave data both with an without noise.
- The efficacy of the sampler with different signal types and SNR's was investigated.
- The EM algorithm is implemented and a comparison made between the two algorithms.

2 Theoretical Background

2.1 Bayesian Inference

In Bayesian statistics the objective is to determine underlying parameters, θ given some observations $\boldsymbol{d}$, a model I_k [1] and some prior belief about θ ; $p(\theta|I_k)$. Using Bayes rule

$$p(\theta|d, I_k) = \frac{p(\theta|I_k)p(\boldsymbol{d}|\theta, I_k)}{p(\boldsymbol{d}|I_k)} \tag{1}$$

Using Bayes rule it is clear that

$$p(\theta|\boldsymbol{d}, I_k) \propto p(\theta|I_k)p(\boldsymbol{d}|\theta, I_k) \tag{2}$$

The normalising factor $p(\boldsymbol{d}|I_k)$ (the *Bayesian evidence* is relevant to model selection, but the proportional form is of more use when calculating parameter estimations to a given model. The *joint posterior density*, $p(\theta|\boldsymbol{d}, I_k)$ is the quantity of interest when determining the values of parameters, summarising the state of knowledge of the parameters *after* the data is known [1]. This posterior distribution can be used in order to generate an estimation of the parameter value in a number of different ways

1. An expected value: $\mathbb{E} = \int_X iXip(xi|\boldsymbol{d}d\xi$
2. A maximum a-posteriori (MAP) estimate : $\hat{\theta}_{\mathrm{ML}}(x) = \arg, \max_{\theta} f(x|\theta)$

Evaluating distribution $f(\theta|\boldsymbol{d})$ may be a complex to evaluate analytically, so instead *Monte Carlo* algorithms can be use as a much simpler way of approximating the properties of a distribution.

[1] Bayesian statistics can be used to determine the properties of the model as well (e.g. the order of an AR model)

2.2 Monte Carlo Integration

The principle of Monte Carlo sampling is that one can generate parameters of a distribution whose pdf is not easy to determine (ie. the integration is difficult to calculate) simply be sampling from it. Approximating the expectation of a pdf is a simple example. A set of N samples $x^t, t = 1, \ldots, N$ is drawn independently from a distribution p(x) then the expectation is approximated by the finite sum [3]

$$E[p(x)] \approx \frac{1}{N}\sum_{t=1}^{n} f(x^{(t)}$$

which may be much more convenient than calculating

$$E(x) = \int_X xp(x)dx$$

By the same principle other properties may be deduced; variance, kurtosis etc. This method relies on the ability of the algorithm to sample from a particular arbitrary distribution which may rely on techniques of sampling from an arbitrary distribution using a standard distribution (such as uniform, beta, guassian etc.).

2.3 Generation of random variates

Three methods for generating the samples are used in this work and are outlined here

2.3.1 Transformation

In the transformation method, random variables are mapped from one distribution to another, by sampling from the second distribution and inverting the process. In the *inversion method* variant of the transformation approach the following steps are taken

1. Sample u from a uniform distribution on $[0, 1]$; $u \sim \mathcal{U}[0, 1]$

2. Then calculate $\theta = F^{-1}(u)$ and hence $\theta \sim f$

This method is however only useful if F (the CDF of the desired distribution) is of closed form. A generalised, and differential, version of this rule can be used since $|p(y)dy| = |p(x)dx| \implies p(y) = p(x)\left|\frac{\partial x}{\partial y}\right|$ it is this form that is used in this work.

2.3.2 Rejection Sampling

In rejection sampling, samples are taken from a distribution where sampling is made easier and rejected with probability related to the relative probability of the sample occurring.

1. If $f(\theta)$ describes the desired sample pdf, and $g(\theta)$ the distribution we wish to sample from, then a value c is chosen such that $f(\theta) < cg(\theta)\forall\theta$ [2]

2. Sample a proposal θ from $g(\theta)$

3. Sample a uniform deviate $u \sim \mathcal{U}[0, cq(\theta)]$

4. Reject proposal if $u > p(\theta)$

.

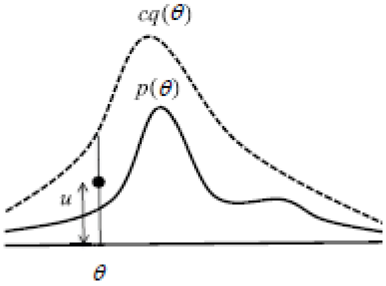

Figure 1: Illustration of rejection sampling. The particular sample here will be rejected. [3]

The efficiency of the method depends on the proportion of θ's which are rejected. Thus an appropriate $g(\theta)$ should be chosen in order that the 'envelope' matches the target distribution as closely as possible. In some cases it may be difficult to find a suitable (such as in a high-dimensional problem), in this case rejection sampling may be unsuitable.

2.3.3 Markov Chains

Where neither the transformation or rejection sampling methods are suitable, Markov chains provide an approach to generate states which represent samples from a particular distribution the A random variable θ can take a discrete state from a set Θ. A Markov

[2] This restriction puts some constraint on $g(\theta)$ in order than an appropriate c can be found

chain (by definition) is a memoryless procedure and hence

$$\begin{aligned} p(\theta_1, \theta_2, \ldots, \theta_N) = p(\theta_1) \prod_{n=2}^{N} p(\theta_1|\theta_2, \theta_3, \ldots, \theta_N) \\ = p(\theta_1) \prod_{n=2}^{N} p(\theta_t|\theta_{t-1}) \end{aligned} \quad (3)$$

Where the RHS is the *transition* probability. The method for generating a Markov Process is as follows [3]

1. Set t = 1
2. Generate an initial value of u, and set $x^(t) = u$
3. Set $t = t + 1$ and sample u from $p(\theta_t|\theta_{t-1})$
4. Set $\theta_t = u$ and repeat until $t = T$

In the early stages of a Markov process the states will be influenced by the starting value (u), before reaching a steady state. This period is known as *burn-in*. If the transition probabilities are represented in matrix form (assuming the state space is finite), where $\boldsymbol{x}_{N+1} = \boldsymbol{T}\boldsymbol{x}_N$ then the rate of convergence depends on the eigenvalue of the 2nd largest magnitude (a larger eigenvalue represents a slower convergence). For large state-spaces this calculations requires a lot of mathematical complexity. If the $\boldsymbol{T}$ (the *Markov kernel*) shows the property $\boldsymbol{x}_N = \boldsymbol{T}\boldsymbol{x}_N$ then the Markov Chain is invariant. Providing the chosen sampling technique allow an invariant distribution to exist, and for it to be identical to the joint distribution, then the technique solves the detailed balance equation, ie.

$$\boldsymbol{x}_i T_{ij} = \boldsymbol{x}_j T_{ji} \quad (4)$$

.

Markov Chains thus allow samples of a stationary distribution to be generated from an arbitrary target distribution, whilst Monte Carlo methods allow samples from a target distribution to be used to determine certain parameters of that distribution that would otherwise be difficult to obtained. The two methods are combined in Markov Chain Monte Carlo (MCMC) sampling ,and a number of algorithms exist in order to facilitate this analysis.

2.4 Gibbs sampling

The principle of Gibbs sampling is that a parameter of the model are sampled from a conditional probability, followed by the next parameter where the first parameter has been inserted into the conditional probability to form a new distribution. The process is works through each parameter, before returning to the first parameter and beginning again, thus the first iteration looks like

$$\begin{aligned} a_1^1 &\leftarrow p(a_1|a_2^0 a_3^0 \ldots a_k^0, \boldsymbol{D}) \\ a_2^1 &\leftarrow p(a_2|a_1^0 a_3^0 \ldots a_k^0, \boldsymbol{D}) \\ &\vdots \end{aligned}$$

(where a_2^1 is the first iteration of the 2nd parameter). Before returning to the first parameter and iterating through again and repeating the process.

The Gibbs sampler is only applicable for cases where it is possible to sample from the marginal densities in the model.

2.5 Linear Models

The General Linear Model (GLM) covers a group of models which assume that a signal is a linear combination of basis functions and a Gaussian noise component. The GLM is extremely useful in signal analysis since a wide variety of real-life signal phenomena follow the GLM. In the GLM the data are described by

$$d(i) = \sum_{k=1}^{M} b_k g_K(i) + e(i) \qquad \text{if } 1 \leq i \leq N \quad (5)$$

or in matrix form

$$\boldsymbol{d} = \boldsymbol{G}\boldsymbol{b} + \boldsymbol{e} \quad (6)$$

.

2.5.1 The Autoregressive Model

The Autoregressive (AR) Model is a form of random process which the output of the system is calculated from the previous outputs. The form of an order p model is

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \quad (7)$$

where $\varphi_1, \ldots, \varphi_p$ are the the model parameters, c is a constant ε_t is Guassian white noise.

AR processes are commonly used to model audio recordings [1] and an $AR(80)$ process will be used for the audio restoration Gibbs sampler in this work.

3 Investigations

3.1 Investigation 1: Deriving a straight line model using a Gibbs sampler

Single dimensioned Gibbs sampler was first set up to determine the best fit for a linear model of form

$$d_i = mx_i + c + n_i$$

where n_i represents a Gaussian white noise signal

$$n_i \sim \mathcal{N}[0, \sigma]$$

The procedure is that described in generality the Gibb's sampling section of this report. Hence for the three variables; m, c, σ that are to be determined here, the first iteration of this process is shown below

$$m_1 \leftarrow p(m|c_0, \sigma_3^0, I, \boldsymbol{D})$$

$$c_1 \leftarrow p(c|m_1, \sigma_0, I, \boldsymbol{D})$$

$$\sigma_1 \leftarrow p(\sigma|m_1, c_1, I, \boldsymbol{D})$$

The form of the marginal densities are derived in the Appendix, and described in the table below

$p(\cdot)$	*Form*	*Parameters*
$p(m\|c_0, \sigma_3^0, I, \boldsymbol{D})$	$\mathcal{N}[\mu_m, \sigma_m]$	$\mu_m = \frac{X_d - cX_1}{X_2}$ $\sigma_m = \frac{\sigma}{\sqrt{X_2}}$
$p(c\|m_1, \sigma_0, I, \boldsymbol{D})$	$\mathcal{N}[\mu_c, \sigma_c]$	$\mu_c = \frac{D_1 - mX_1}{N}$ $\sigma_c = \frac{\sigma}{\sqrt{N}}$
$p(\sigma\|m_1, c_1, I, \boldsymbol{D})$	$Ga^{-1/2}[\alpha, \beta, c]$	$\alpha = \frac{N}{2}$ $\beta = \frac{1}{2}\sum_{i=1}^{N}(d_i - mx_i -$ $c)^2$ $c = \frac{2\beta^\alpha}{\Gamma(\alpha)}$

Table 1: Parameters for the marginal probabilities in a straight line Gibb's sampler

Generating random variates sampled from the Normal distributions was trivial using Matlab, but the inverse square root distribution required use of one of the techniques described in 2.3. In this case the *rejection sampling* method was used. A Normal distribution was used to sample from and the method outlined in 2.3.1 generated the $Ga^{1/2}$ random variates. The choice of c in order to satisfy $f(\theta) < cg(\theta)\forall\theta$ was chosen using an iteration which chose the smallest value of c where this was satisfied. By using the smallest value of c, the amount of samples rejected is minimised (and hence the efficiency maximised), this was the same reason for using the Normal distribution to sample from (it is not dissimilar to the shape of the $Ga^{1/2}$). This process is shown graphically in Fig. 2 for the first iteration[3]

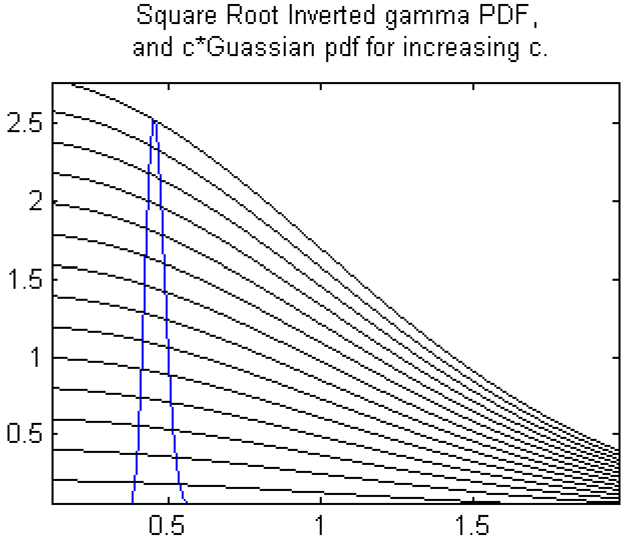

Figure 2: The $Ga^{1/2}$ distribution is shown in blue and Normal in black, with different values of k

3.1.1 Results

A data set consisting of a straight line was constructed where $m = 3$, $c = 1$ and without noise- a 'quiet' case (ie. $\mu_n = 0 = \sigma_n$). The data set consisted of 100 samples with samples 40 to 69 removed. The result after 40 iterations is shown below, indicating that qualitatively the Gibbs sampler has successfully identified the model hence correctly interpolated the missing data. The sampler was initiated with values

$$\{m_0, c_0, \sigma_0\} = \{20, 10, 4\}$$

[3] A new K is found for every instance a random $Ga^{1/2}$ variate is required, and hence once for each iteration of the Gibbs sampler, when a new sample is found for σ.

. For the purposes of this investigation, no considerations were made as to the prior probability of these values- they were effectively drawn from a uniform distribution. The effect of choosing different starting values is considered later.

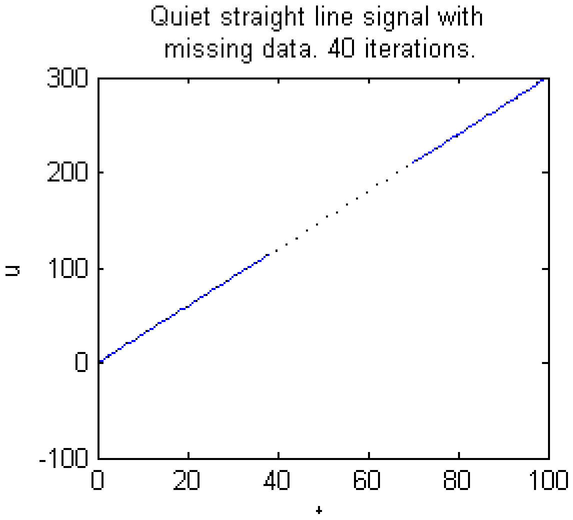

Figure 3: Gibbs sampler determining the parameters of a straight line model. The model data is shown in blue, with the Gibbs result shown in dotted black

The parameters derived after 40 iterations were

$$\{m, c, \sigma\} = \{2.9989, 1.0095, 0.4724\}$$

. The m, c values therefore closely match the parameters of the underlying model. The value for σ appears further out of range, but qualitatively the Gibbs signal does not appear especially 'noisy'. It is hypothesized that the sampler struggles to converge to parameter values close to 0 (a similarly large discrepancy is found if m or c are set to zero, possibly due to rounding error effects in Matlab.

The next investigation used the Gibb's sampler to determine the parameters of the straight line model this time with noisy data. In this case the parameters are

$$\{m, c, \sigma\} = \{0.5, 1, 1\}$$

The initial values in this test were

$$\{m_0, c_0, \sigma_0\} = \{20, 10, 4\}$$

. The chart below shows that qualitatively if appears the sampler has correctly determined the parameters of the model.

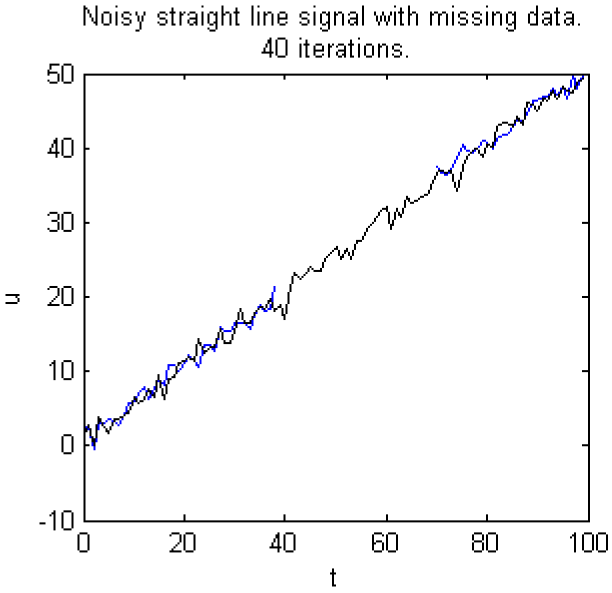

Figure 4: Gibbs sampler determining the parameters of a straight line model. The model data is shown in blue, with the Gibbs result shown in black

The numerical results were

$$\{m, c, \sigma\} = \{0.4941, 1.0347, 1.0509\}$$

, showing good agreement with the actual model. The efficiency of this sampler however was poor and would often fail during rejection sampling (a limit of 100 rejections per proposal θ was enforced, forcing the script to exit). The failure was intermittent due to the random number generator, choosing different numbers during each run. The rejection sampler in this case was very ill conditioned, the term $c = \frac{2\beta^{\alpha}}{\Gamma(\alpha)}$ in the $Ga^{1/2}$ distribution was capable of swinging by over 200 orders of magnitude depending on the value of β (which itself could swing by a few orders of magnitude). For these cases the probability of a proposal theta with a distribution containing an extreme β was extremely low. For this reason it was opted that a rejection sampler would not be used for the audio signal Gibbs sampler.

3.2 Investigation 2: Interpolation of missing audio data using a Gibbs sampler

The missing audio data problem was cast as a mutidimensional variant of the straight line interpolant. With the audio signal modeled as an AR process, the sampler determined the AR coefficients $\boldsymbol{a}$ (a vector of size p, the order of the AR process) and the unknown data points $\boldsymbol{x_u}$ (a vector of size l- the number of missing samples). Thus the first iteration of the Gibbs sampler becomes

$$\boldsymbol{x_{u,1}} \leftarrow p(\boldsymbol{x_{u,1}}|\boldsymbol{a}_0, \sigma_0, I, \boldsymbol{D})$$

$$\boldsymbol{a} \leftarrow p(\boldsymbol{a}|\boldsymbol{x_{u,1}}, \sigma_0, I, \boldsymbol{D})$$

$$\sigma_1 \leftarrow p(\sigma_1|\boldsymbol{a}_1, \boldsymbol{x_{u,1}}, I, \boldsymbol{D})$$

The pdfs for these marginal densities are derived in the Appendix,and summarised in the table below.

$p(\cdot)$	*Form*	*Parameters*
$p(\boldsymbol{x_{u,1}}\|\boldsymbol{a}_0, \sigma_0, I, \boldsymbol{D})$	$\mathcal{N}[\hat{z}, C^{-1}]$	$\boldsymbol{C}^{-1} = \boldsymbol{D}/\sigma^2$ $\hat{\boldsymbol{z}} = -\boldsymbol{D}^{-}\boldsymbol{1}\boldsymbol{B^T y}$
$p(\boldsymbol{a}\|\boldsymbol{x_{u,1}}, \sigma_0, I, \boldsymbol{D})$	$\mathcal{N}[\hat{z}, C^{-1}]$	$\boldsymbol{C}^{-1} = \boldsymbol{L^T L}/\sigma^2$ $\hat{\boldsymbol{z}} = \boldsymbol{(L^T)^{-1} L^T w}$
$p(\sigma_1\|\boldsymbol{a}_1, \boldsymbol{x_{u,1}}, I, \boldsymbol{D})$	$\chi^{-1}[k]$	$k = \frac{N-1}{2}$

Table 2: Parameters for the marginal probabilities in the audio restoration Gibb's sampler

3.2.1 Quiet Sine Wave

In the first interpolation investigation a sound wave with no noise addition (ie. 'quiet') was investigated. Thie sine wave can be modeled with a 2nd order AR process. Samples were cut out of the signal (set to 0) starting at $n = 500$ and lasting for 200 samples. The results for diffrerent numbers of iterations are shown below. The initialisations were $\boldsymbol{z_0} = \{0, 0, \ldots, 0\}$ and $\boldsymbol{a_0} = \{1, 1, \ldots, 1\}$. An order 80 AR model was use for these day.

The sampler succesfully interpolates the sine wave without any visible distortion after 50 iterations of the Gibbs sampler. From around the 30th iteration the difference in the quality of the reproduction after further iterations is limited. The magnitude of the excitation sequence tends to 0 in the gap, limited only by the resolution of the computer calculation.

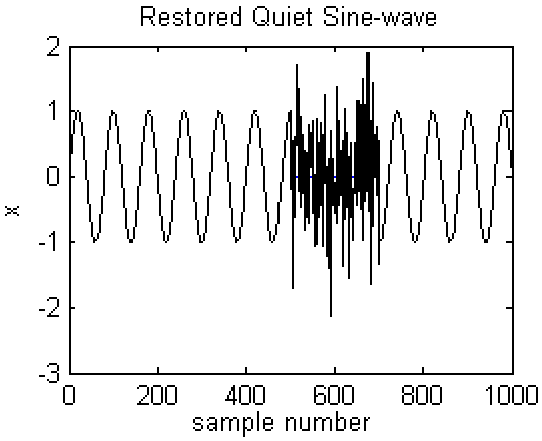

Figure 5: Interpolated signal after one iteration of the Gibbs sampler, augmented data blue, interpolation black

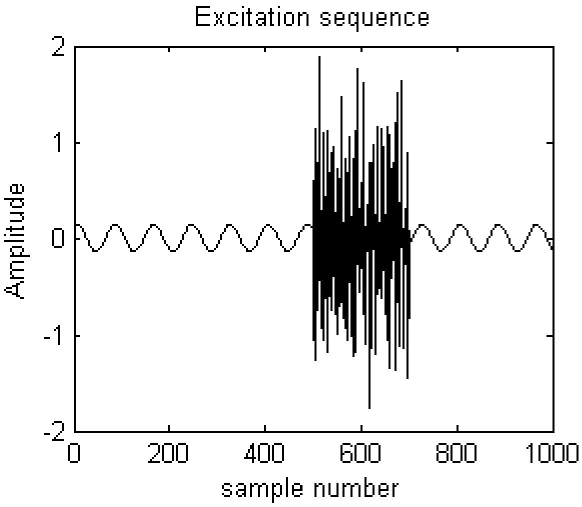

Figure 6: Excitation sequence signal after one iteration of the Gibbs sampler

For perfect restoration we would expect this value to be zero for a completely deterministic system such as this.

3.2.2 Hairy Sine Wave

A sine wave with noise addition (ie. 'hairy') was investigated. The first investigation used a white noise Guassian with $n \sim \mathcal{N}[0, 0.2]$, and thus a (power) SNR of 11 dB. The samples were set up again with 200 data points missing, starting at the 500th sample. The initilisations were $\boldsymbol{z_0} = \{0, 0, \ldots, 0\}$ and $\boldsymbol{a_0} = \{1, 1, \ldots, 1\}$.

At this SNR the interpolated values cannot be dis-

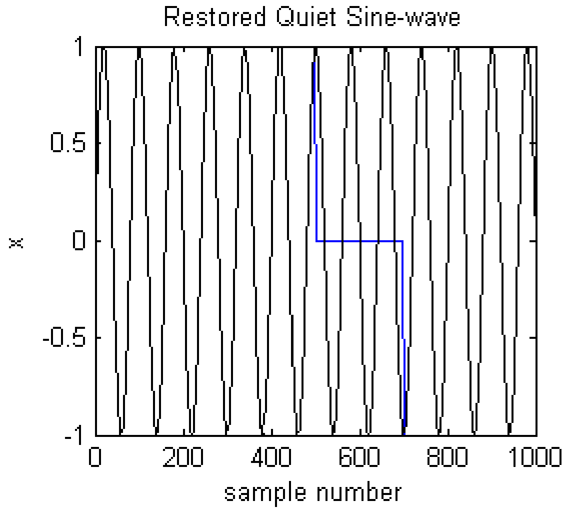

Figure 7: Interpolated signal after 50 iterations of the Gibbs sampler, augmented data blue, interpolation black

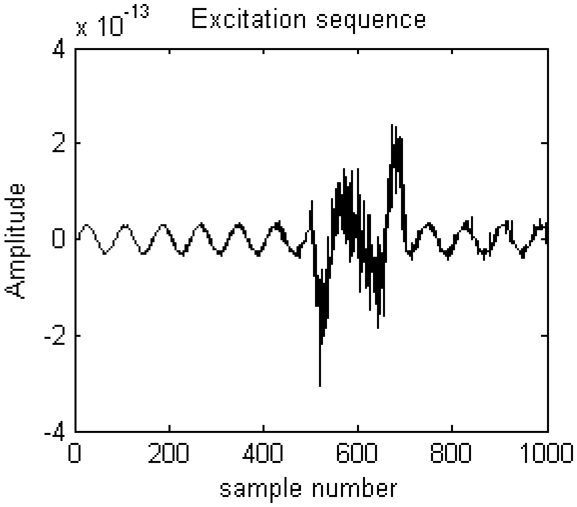

Figure 8: Excitation sequence signal after 50 iterations of the Gibbs sampler

tinguished from the rest of the sample indicating a succseful application of the interpolant.

3.2.3 Hairy Chirp

The next investigation considered the interpolation of a 'hairy chirp' signal. This wave form as a narrowing frequency, and was considered an interesting case where the frequency both changes over the period of the interpolation and does so predictably (to the observer). The initializations vectors, and missing datapoints were kept the same for this investigation, as was the order of the AR process.

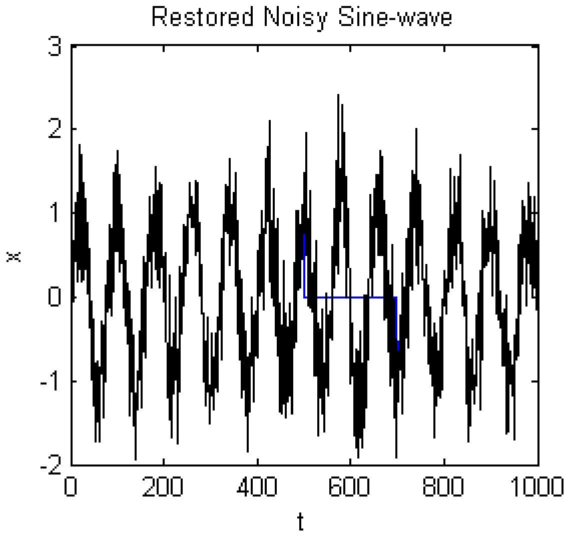

Figure 9: Interpolated hairy sine after 50 iterations of the Gibbs sampler, augmented data blue, interpolation black

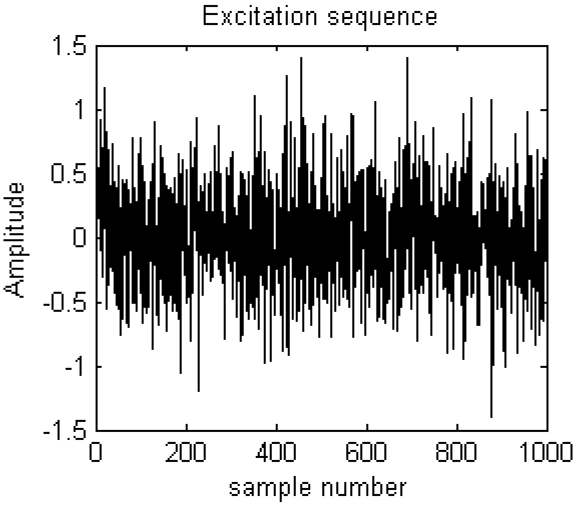

Figure 10: Excitation sequence after 50 iterations of the Gibbs sampler

It is seen that the interpolation on the hairy chirp is worse than on the pure sinusoid, even with 100 iterations of the sampler. As would be expected of a stationary interpolation of a time varying process, the algorithm is not capable of predicting the frequency shift. If the chirp signal were some part of a random audio signal it could be argued that the interpolant has found a 'reasonable' sample to fit the missing points. A similar argument is made in [**?**] regarding tuba data, the sampler obviously

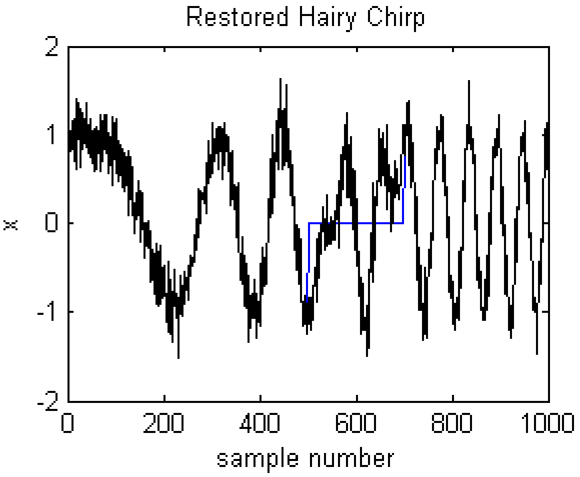

Figure 11: Interpolated hairy chirp after 100 iterations of the Gibbs sampler, augmented data blue, interpolation black

not reproduce the signal exactly, but a reasonable interpolation of what it could have been is made.

This result was also shown in [5]. It was also shown that by modelling the chirp as a time varying AR process (in that case using 5 fourier basis vectors to represent the TVAR coefficients) that the principle can be extended to such a model and hence the interpolation of the hairy chirp ran succesfully. It is beyond the scope of this work to extend the stationary model described in such a way, but the result found in [5] is shown below

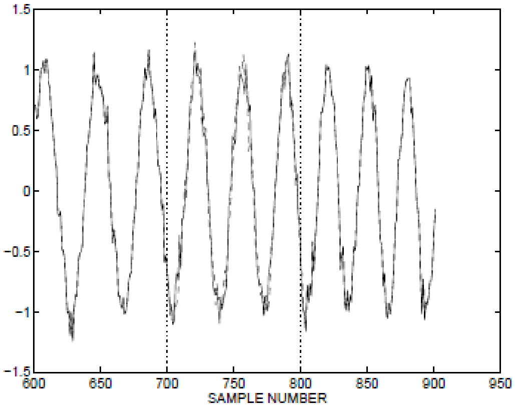

Figure 12: Interpolated hairy chirp with a Gibbs sampler using a TVAR model [5]

3.3 Investigation 3: Interpolation of missing audio data using the Expectation Maximization Algorithm

The Expectation Maximization (EM method) algorithm is an alternative to the Gibbs sampler for determining the values of both the model parameters ($\boldsymbol{\theta}$), and the missing data ($\boldsymbol{z}$), and is related to the Maximum Likelyhood (ML) method applied to the case of missing data. The EM method produces Maximum a Posteriori Probability (MAP) Estimate of both ($\boldsymbol{\theta}$), and ($\boldsymbol{z}$) by performing two operations, an *expectation step* and a *maximization step*. The method requires the calculation of the expectation using the log-likelyhoods, using the Q function, defined as

$$Q(\boldsymbol{z}, \boldsymbol{z}^*) = \int_{\Theta} \log p(\boldsymbol{z}|y, \theta) p(\theta|\boldsymbol{y}, \boldsymbol{z}^* d\theta) \tag{8}$$

The evaluation of the integral (which is derived in the appendix) yields equations for the expected values of the parameters

$$\hat{\theta} = (\boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{L})^{-}1\boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{w} \tag{9}$$

$$\hat{\phi} = (\boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{M})^{-}1\boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{v} \tag{10}$$

The maximization step then aims to maximise the expectation computed in the first step [**?**], ie. find

z_{i+1} such that $Q(\boldsymbol{z_i}, \boldsymbol{z_{i+1}})$ is maximized, thus the solution to

$$\frac{\partial Q(\boldsymbol{z_i}, \boldsymbol{z_{i+1}})}{\partial \boldsymbol{z_{i+1}}} = 0 \tag{11}$$

which as shown in the derivation is equivalent to solving the following equation

$$(\sigma^2 \boldsymbol{T} + \boldsymbol{D})\boldsymbol{z_{i+1}} = -(\sigma \boldsymbol{q} + \boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{y}) \tag{12}$$

3.3.1 Quiet Sine Wave

As with the Gibbs sampler, a sine wave without noise was used to investigate the performance of the EM algorithm. Again, the sine wave was modeled as an AR order-2 process.

By inspection of the interpolated waveform the EM algorithm approaches perfect reproduction of the sine wave signal after 3 iterations. The excitation sequence approaches zero in the gap with a magnitude of 10^{-5}. Further iterations reduce this further, 10 iterations yields a magnitude of $10^{-13)}$.

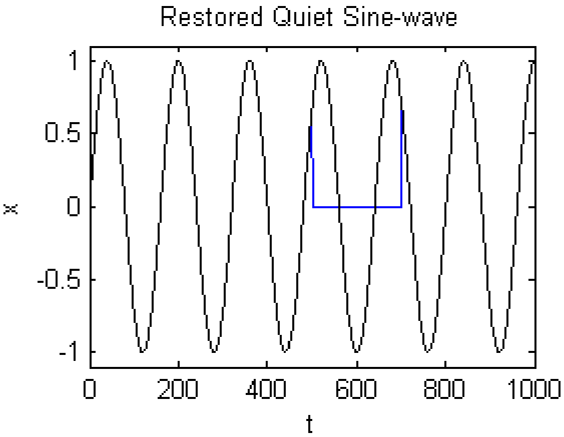

Figure 13: Interpolated quiet sine after 3 iterations of the EM algorithm, augmented data blue, interpolation black

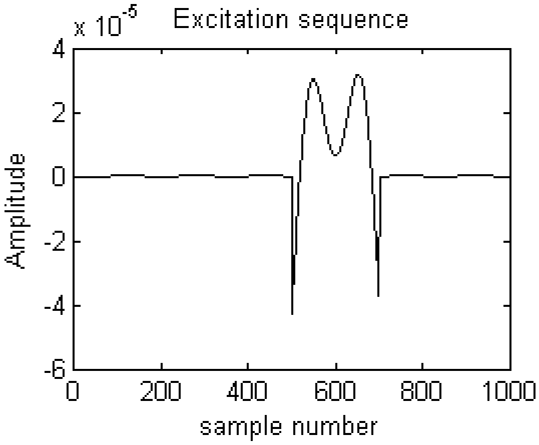

Figure 14: Excitation sequence of the EM algorithm

3.3.2 Hairy Sine Wave

Fourteen iterations on the EM algorithm were performed to interpolate the same hairy sine wave that the Gibbs sampler interpolated. The hairy sine wave was again modeled as an AR order-80 process with the same white noise addition as before.

The EM algorithm shows an implausible restoration where the interpolated sine wave appears to have undergone a hair cut, indicating that some form of noise filtering is taking place. This shows the same result as has been generated in [1].

4 Discussion

We have seen the succesful application of both the Gibbs sampler and EM algorithms to the taks of missing data interpolation. Both methods successfully interpolated the quiet sine wave by modelling the signal as an order 2 AR process. The magintude of the excitation energy could be reduced to a minimum of around 10^{-13} in both cases, limited by the finite precision of the computer. In the case of the noisy or hairy sine wave (AR-80 process), the performances of the EM algorithm and the Gibbs sampler differ significantly. The EM algorithm can be seen to be performing some form of noise filtering, the interpolated is not a realisitic interpolation of the signal, wheras the observer can not distinguish between interpolated and real data points for the Gibbs sampler after 3 iterations. It was also shown that the Gibbs sampler was capable of creating 'realistic' in-

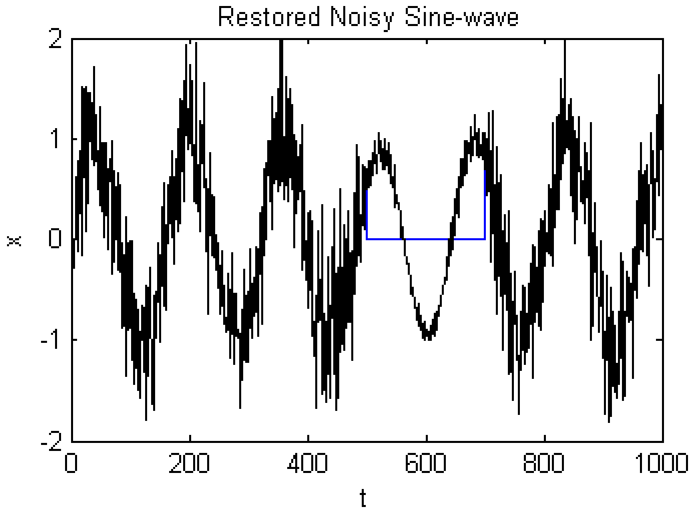

Figure 15: Interpolated noisy sine after 3 iterations of the EM algorithm, augmented data blue, interpolation black

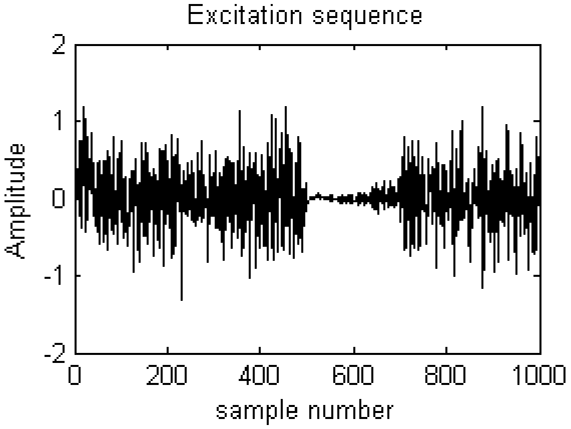

Figure 16: Excitation sequence of the EM algorithm

terpolation for the time varying chirp signal. The interpolation was not an accurate representation of the signal's actual waveform, and it has been shown in [5] that by simulating a TVAR process perfect reproduction of the chirp signal is possible.
One interesting way of considering the differences between the performances of the Gibbs and EM algorithms is to consider the analogy with thermodynamics.

4.1 Thermodynamic analogy

In statistical thermodynamics the *boltzmann* distribution is used to describe the distribution of states at energy E_i, whilst at a temperature T

$$p_B(E) = \exp\left(\frac{-E_i}{kT}\right) \tag{13}$$

The analogy is drawn in that nature seeks to minimise energy, whereas in the estimation process one seeks to maximise probability. Therefore an equivalent substitution can be made

$$E(\omega) = -\log[p(\omega)]$$

In both the case of the EM and Gibbs algorithms $\log[p(\omega)] = p(\boldsymbol{z}|\boldsymbol{y})$. However, where as the EM algorithm samples at $kT = 0$ the Gibbs sampler produces interpolant at $kT = 1$ (ie. ambient). This gives us an interpretation of the differing performances of the Gibbs and EM algoirthms, the Gibbs sampler produces an interpolant *at the same temperature* as the observed data.

A Derivations

A.1 Model Formulation

Here the basic setup for the AR process with missing data is described and the formulation with results used in both the Gibbs sampler and EM algorithm.

We first consider an augented data vector $\boldsymbol{x}$ which is comprised of the observed data $\mathbf{y}$ and the missing data vector $\mathbf{v}$. Since the first p samples of an AR(p) process depend on values of variables that have not been observed, in some instances it makes more sense to omit the first p values, the vector $\mathbf{w}$ is in this form.

In an AR(p) process each data point is dependent on the previous p values and hence we have

$$x_i = \sum_{j=1}^{p} \theta_i x_{i-j} + e_i$$

The excitation sequence can be written in the form

$$\mathbf{e} = \mathbf{w} - \mathbf{L}\theta$$

and also

$$\mathbf{e} = \mathbf{Kx}$$

where $\mathbf{L}$ is a band diagonal Toeplitz matrix. $\mathbf{e}$ is the excitation energy, and is by assumption a independently identically distributed (i.i.d.) Guassian white noise process

Next we define the likelyhood of the augmented data

$$p(\mathbf{e}) = p(\mathbf{w}|\theta, \sigma)$$

then from the assumptions of the excitation process

$$p(\mathbf{w}|\theta,\sigma) = (2\pi\sigma^2)^{-(n-p)/2} \exp\left(-\frac{\mathbf{e^T e}}{2\sigma^2}\right)$$

Using our two equations for $\mathbf{e}$ we can derive two further equations as a quadratic in θ

$$\mathbf{e^T e} = \mathbf{w^T w} - 2\mathbf{w^T L}\theta + \theta^\mathbf{T} \mathbf{L^T L}\theta$$

and

$$\mathbf{e^T e} = \mathbf{y^T y} - 2\mathbf{y^B z} + \mathbf{z^T D^T D z}$$

Those matrices were derived by considering the dimensions of

$$\mathbf{K^T K} = \begin{pmatrix} \mathbf{A_{11}} & \mathbf{B_1} & \mathbf{A_{12}} \\ \mathbf{B_1^T} & \mathbf{D} & \mathbf{B_2^T} \\ \mathbf{A_{21}} & \mathbf{B_2} & \mathbf{A_{22}} \end{pmatrix}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} \\ & \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B_1} \\ \mathbf{B_2} \end{pmatrix}$$

$$\mathbf{w} = \begin{pmatrix} \mathbf{y_1} \\ \mathbf{z} \\ \mathbf{y_2} \end{pmatrix}$$

and

$$\mathbf{y} = \begin{pmatrix} \mathbf{y_1} \\ \mathbf{y_2} \end{pmatrix}$$

A.2 Gibbs Sampler

A.2.1 Straight line algorithm

For the straight line gibbs sampler we need to determine the conditional densities for each of c, σ, m required for our interpolation of the straight line.

We start with the likelyhood function for Gaussian white noise

$$p(\mathbf{D}|m,c,\sigma,I) = \frac{1}{(2pi\sigma^2)^{N/2}} \times \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i - mx_i - C)^2\right]$$

$$= (2\pi\sigma^2)^{-N/2} \times e^U$$

where

$$y = \left[-\frac{1}{2\sigma^2}(D_2 - 2cD_1 - 2MX_d + 2mcX_1 + m^2X_2 + NC^2)\right]$$

and

$D_1 = \sum_{i=1}^N d_i$ $D_2 = \sum_{i=1}^N d_i^2$, $X_D = \sum_{i=1}^N x_i d_i$, $X_1 = \sum_{i=1}^N x_i$ $X_2 = \sum_{i=1}^N x_i^2$ Then applying bayes thereom we have

$$p(m,c,\sigma|\mathbf{D},I) = \frac{p(\mathbf{D}|m,c,\sigma,I)p(m|I)p(c|I)p(\sigma|I)}{p(\mathbf{D}|I)}$$

Given no information we can assume that the priors are uniform

$$p(m|I) = k_1$$

$$p(c|I) = k_2$$

$$p(\sigma|I) = \frac{k_3}{\sigma}$$

and $p(\mathbf{D}|I)$ is constant for a fixed model. So

$$p(m,c,\sigma|\mathbf{D},I) \propto \sigma^{-1} p(\mathbf{D}|m,c,\sigma,I)$$

so using Bayes we have

$$p(m|c,\sigma,\mathbf{D},I) \propto \exp\left[\frac{-1}{2\sigma^2}(m^2X_2 - 2m(X_d - cX_1))\right]$$

Thefore we can sample m from a Gaussian with mean $\frac{X_d - cX_1}{X_2}$ and variance $\frac{\sigma}{\sqrt{X_2}}$.

A similar derivation for the intercept shows we sample from another Gaussian with mean

$$\frac{D_1 - mX_1}{N}$$

and variance

$$\frac{\sigma}{\sqrt{N}}$$

.

The standard deviation however is sampled from

$$p(\sigma|m,c,\mathbf{D},I) \propto \sigma^{-(N+1)} e^{(} - A)$$

where

$$A = \frac{1}{2\sigma^2}(m^2X_2 - 2m(X_D - cX_1) + Nc^2 + D_2 - 2cD_1)$$

This in the form of a 'Square root inverted gamma' ($Ga^{-1/2}$) distribution

$$p(x) = cx^{-(2\alpha+1)} \exp\left[\frac{-\beta}{x^2}\right]$$

where $\alpha = \frac{N}{2}$, $\beta = \frac{1}{2}\sum_{i=1} N(d_i - mx_i - c)^2$ and $c = \frac{2\beta^\alpha}{\gamma(\alpha)}$

A.2.2 Gibbs sampler for an Autoregressive Function

Deriving the conditional densities fo the Autoregressive function progresses in a similar manner to that for the straight line interpolator. In this instance we need to derive the conditional densities for the missing data vector $\mathbf{z}$, the AR model parameters θ and the noise standard deviation.

Using the results derived for the AR model we start with

$$p(\mathbf{z},\theta,\sigma|\mathbf{y})\sigma^{-N}e^U$$

where

$$U = \frac{-1}{2\sigma^2}(\mathbf{y^TAy} + 2\mathbf{y^TBz} + \mathbf{z^TDz})$$

integrating out the missing data yields

$$p(\theta,\sigma|\mathbf{y}) \propto \sigma^{-(N-M)}e^U$$

using Bayes theorem we derive the coniditional probability in the form

$$p(\mathbf{z}|\theta,\sigma,\mathbf{y}) \propto e^U$$

with

$$U = \frac{-1}{2\sigma^2}(\mathbf{z} - \hat{\mathbf{z}})^T C^{-1}(\mathbf{z} - \hat{\mathbf{z}})$$

Thus we need to sample from a multivariate Gaussian with inverse covariance matrix

$$C^{-1} = \frac{\mathbf{D}}{\sigma^2}$$

and

$$\hat{\mathbf{z}} = -\mathbf{D^{-1}B^Ty}$$

The same method is applied to yield the autoregressive parameters, starting with

$$p(\mathbf{z}, \theta, \sigma|\mathbf{y})\sigma^{-N}e^U$$

where

$$U = \frac{-1}{2\sigma^2}(\mathbf{w^Tw} - 2\mathbf{w^TL}\theta + \theta^T\mathbf{L^TL}\theta)$$

again, integrating out the autoregressive parameters gives

$$p(\theta|\mathbf{z}, \sigma, \mathbf{y}) \propto e^{-Q/2\sigma^2}$$

with

$$Q = \mathbf{w^TL}(\mathbf{L^TL})^{\mathbf{-1}}\mathbf{L^tw} - 2\mathbf{w^TL}\theta + \theta^T\mathbf{L^TL}\theta$$

Thus we are left with another multivariate Gaussian with inverse covariance matrix

$$\mathbf{C^{-1}} = \frac{\mathbf{L^TL}}{\sigma^2}$$

and

$$\hat{\theta} = (\mathbf{L^TL})^{\mathbf{-1}}\mathbf{L^Tw}$$

Finally, the method is applied again to calculate standard deviation, starting with

$$p(\sigma, \theta, \mathbf{z}|\mathbf{y}) = \sigma^{-N}\exp\left[-\frac{\mathbf{e^Te}}{2\sigma^2}\right]$$

integrating out the standard deviation gives

$$p(\theta, \mathbf{z}|\mathbf{y}) \propto (e^Te)^{-N/2}$$

so

$$p(\sigma|\theta, \mathbf{z}, \mathbf{y}) \propto (e^Te)^{-N/2}\sigma^{-N}\exp\left[-\frac{\mathbf{e^Te}}{2\sigma^2}\right]$$

However this is in the form of an inverse chi density with mode

$$\hat{\sigma} = \left(\frac{\mathbf{e^Te}}{N}\right)^{1/2}$$

this is difficult to sample from in matlab, so instead we sample from the Gamma distribution and scale. Hence, the two steps are generate a gamma variate with $\alpha = \frac{N-1}{2}$

$$X_i \leftarrow x^{\alpha-1}\exp -x$$

and then take the reciprocal square root and scale so

$$\sigma_i = \left(\frac{\mathbf{e^Te}}{2}\right)^{0.5}\frac{1}{\sqrt{x_i}}$$

A.3 EM Algorithm

Starting from Bayes' theorem we have

$$p(\mathbf{z}|\mathbf{y}) \propto \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{p(\theta|\mathbf{y}, \mathbf{z})}$$

then taking logs of both sides we have

$$\log p(\mathbf{z}, \mathbf{y}|\theta) = p(\mathbf{z}, \mathbf{y}|\theta) - p(\theta|\mathbf{y}, \mathbf{z})$$

We can multiply both sides by $p(\theta|\mathbf{y}, \mathbf{z}^*$ and then work out the expectation with respect to θ;

$$\log p(\mathbf{z}|\mathbf{y}) = \int_\Theta \log(\mathbf{z}|\mathbf{y}, \theta)p(\theta|\mathbf{y}, \mathbf{z}^*)d\theta$$

$$- \int_\Theta \log p(\theta|\mathbf{y}, \mathbf{z})p(\theta|\mathbf{y}, \mathbf{z}^*)$$

We can define

$$Q(\mathbf{z}, \mathbf{z}^*) = \int_\Theta \log(\mathbf{z}|\mathbf{y}, \theta)p(\theta|\mathbf{y}, \mathbf{z}^*)d\theta$$

$$H(\mathbf{z}, \mathbf{z}*) = \int_\Theta \log p(\theta|\mathbf{y}, \mathbf{z})p(\theta|\mathbf{y}, \mathbf{z}^*)d\theta$$

A.3.1 Expectation

Fitzgerald shows in [1] how to solve the integral for the Q function in closed form, yielding

$$Q(\mathbf{z_{i+1}}, \mathbf{z_i}) = K(\mathbf{z_i})\left[\text{Trace}\left(\frac{(\mathbf{M^TM}^{-1})}{\mathbf{L^TL})}\right)2\right]$$

$$K(\mathbf{z_i})\left[\frac{(\mathbf{w}-\mathbf{L}\hat{\phi})^T(\mathbf{w}-\mathbf{L}\hat{\phi}}{2\sigma^2}-\frac{N}{2}\log(2\pi\sigma^2)\right]$$

$$K(\mathbf{z_i})=-\frac{(2\pi\sigma^2)^{-\frac{N}{2}}}{\sqrt{det\mathbf{M^TM}}}e^U$$

where

$$U=\left[-(\mathbf{V^Tv}-\mathbf{v^TM(M^TM)^{-1}M^Tv}2\sigma^2\right]$$

and

$$\hat{\phi}=((\mathbf{M^TM})^{-1})\mathbf{M^Tv}$$

A.4 Maximisation

In the maximisation step we aim to find a $\mathbf{z_{i+1}}$ and so

$$Q(\mathbf{z_{i+1}}z_i)$$

this is equivalent to solving

$$\frac{1}{2}\text{Trace}\left[((\mathbf{L_i^TL_i})^{-1}\frac{\partial}{\partial\mathbf{z_{i+1}}}(\mathbf{L_{i+1}}^TL_{i+1})\right]=0$$

$$+\frac{1}{2\sigma^2}\frac{\partial}{\partial\mathbf{z_{i+1}}}(\mathbf{w}-\mathbf{L_{i+1}}\hat{\phi})^T(\mathbf{w}-\mathbf{L_{i+1}}\hat{\phi})=0$$

we can also use the result found earlier

$$\mathbf{e}=\mathbf{w}-\mathbf{L_{i+1}}\hat{\phi}=\mathbf{K}(\hat{\phi})\mathbf{w}$$

Using this Fitzgerald [1] shows how the trace can be differentiated to yield the equation

$$\frac{1}{2}\text{Trace}\left[((\mathbf{L_i^TL_i})^{-1}\frac{\partial}{\partial\mathbf{z_{i+1}}}\right]=\mathbf{Tz_{i+1}}+\mathbf{q}$$

and thus

$$(\sigma^2\boldsymbol{T}+\boldsymbol{D})\boldsymbol{z_{i+1}}=-(\sigma\boldsymbol{q}+\boldsymbol{B^Ty})$$

which is equation 12 in the text.

References

[1] Fitzgerald, W. J. and O Ruanaidh, J. J. K., 1996. *Numerical Bayesian Methods Applied to Signal Processing.* New York. Springer-Verlag.

[2] Stevens, J. W., 2009. *What is Bayesian Statistics?* (What is Series). Oxford University.

[3] Steyvers, M., 2011. *Computational Statistics with Matlab* University of California.

[4] Blimes, J. A., 1998. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models* University of Berkley.

[5] Godsill, S.J., Rajan, J.J. and Rayner, P.JW., 1996 *A Bayesian approach to parameter estimation and inpterpolation of time-varying autoregressive processes using the Gibbs sampler* University of Cambridge.

[6] Fitzgerald, W. J., 16/05/2001. *An introduction to Bayesian inference applied to signal and data processing* Cambridge University Engineering Department.