DD2447 - Statistical Methods in Applied Computer Science, Fall 2018
Assignment 2

Alexandra Hotti
Sebastian Ståhl

December 2018

1 2.1 Gibbs sampler for the magic word

1.1 Question 1 Question 2

The generative model, Algorithm 1, was used to generate N sequences of length $M, s^1, ..., s^N$ where $s^n = s_1^n, ..., s_M^N$. All sequences are over the alphabet $[K]$. Each sequence has a "magic" word of length W hidden in it and the rest is called background.

Algorithm 1 used to generate data

1: **for** n = 1:N **do**
2: $r_n \sim [M - w + 1]$
3: The j:th position in the mw is sampled from $q_j(x) = Cat(x|\theta_j)$, where θ_j has a $Dir(\theta_j|\alpha)$ prior.
4: All the background positions are sampled from: $q(x) = Cat(x|\theta)$, where θ has a $Dir(\theta|\alpha')$

We are using a Gibbs sampler which can estimate the posterior over start positions after having observed $s^1, ..., s^N$. The sampler is collapsed and the hyperparameters α and α' are known. The states are vectors of startpositions $R = (r_1, ..., r_N)$. We are interested in the posterior $p(r_1, ...r_N|D)$ where D is a set of sequences $s^1, ..., s^N$ which we generate with 1 and r_n is the start position of the magic word in the n:th sequence s^n. We use the Dirchlet-categorical distributions for the j:th position of the magic words, the marginal likelihood is given by (1)

$$p(D_j|R) = [\Gamma(\sum_k \alpha_k)/\Gamma(N + \sum_k \alpha_k)] \prod_k \Gamma(N_k^j + \alpha_k)/\Gamma(\alpha_k) \tag{1}$$

N_k^j is the count of the symbol k in the j:th column of the magic words, induced by R.

The marginal likelihood for the background positions is given by (2)

$$p(D_B|R) = [\Gamma(\sum_k \alpha'_k)/\Gamma(B + \sum_k \alpha'_k)] \prod_k \Gamma(B_k + \alpha'_k)/\Gamma(\alpha'_k) \tag{2}$$

To avoid underflow we log the marginal likelihoods and the formulas becomes the following:

$$p(D_j|R) = [\log(\Gamma(\sum_k \alpha_k)) - \log(\Gamma(N + \sum_k \alpha_k))] + \sum_k \log(\Gamma(N_k^j + \alpha_k)) - \log(\Gamma(\alpha_k)) \tag{3}$$

$$p(D_B|R) = [\log(\Gamma(\sum_k \alpha'_k)) - \log(\Gamma(B + \sum_k \alpha'_k))] + \sum_k \log(\Gamma(B_k + \alpha'_k)) - \log(\Gamma(\alpha_k)) \tag{4}$$

These marginal likelihoods are then used in the formula for the posterior probabilities which is defined as follows:

$$p(r_n|R_{-n}, D) = r(R_{-n} \cup r_n, D)/(p(R_{-n}, D) \propto p(R_{-n} \cup r_n, D) \propto p(D|R_{-n} \cup r_n) = p(D_B|R_{-n} \cup r_n) \prod_j p(D_j|R_{-n} \cup r_n) \tag{5}$$

And since we use log marginal likelihoods we log this expression as well which becomes the following:

$$p(r_n|R_{-n}, D) = \log(p(D_B|R_{-n} \cup r_n) + \sum_j p(D_j|R_{-n} \cup r_n) \tag{6}$$

This is how we estimate the posterior.

Results:
Provided is the results from using $\alpha = [1, 1, 1, 1]$, $\alpha' = [7, 13, 1, 5], seed = 123, N = 20, M = 10, K = 4, W = 5$. The Gibbs sampler was run for 1000 iterations. The accuracy and the histograms where made with a lag of 2 to reduce correlation and a burn in of 100 iterations:

First here are 3 histograms for 3 different start positions and accuracy for 4 different positions. Due to the page limitation of 2 pages we did not have enough room to display values for all 20 sequences.

Figure 1: Histograms for 3 start positions for Chain 1

Accuracy:

	r0	r1	r2	r3	r4
Accuracy	0.99	0.94	0.95	0.96	0.89

Convergence: To access convergence we ran the Gibbs sampler using 2 additional chains on the same data, but with other start positions. Again using a lag of 2 and a burn in of 100. We could see that all three chains converge to the same values. The seeds used for chain 2 was 321 and the seed for chain 3 was 99. The histograms all converge to the same correct start positions in all plots. They also become very similar, but not identical. Compare for instance the plot for r0 for the third and the second chain.

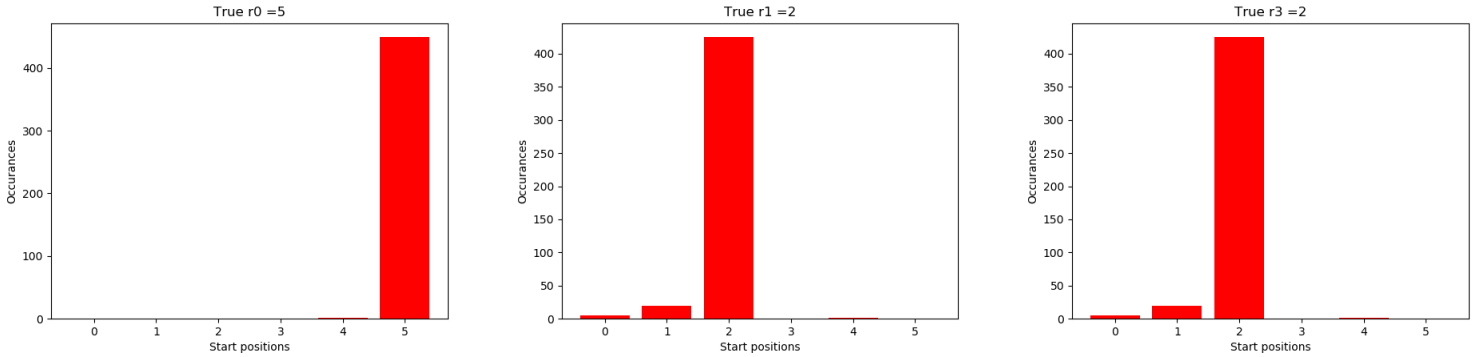


Figure 2: Histograms for 3 start positions for Chain 2

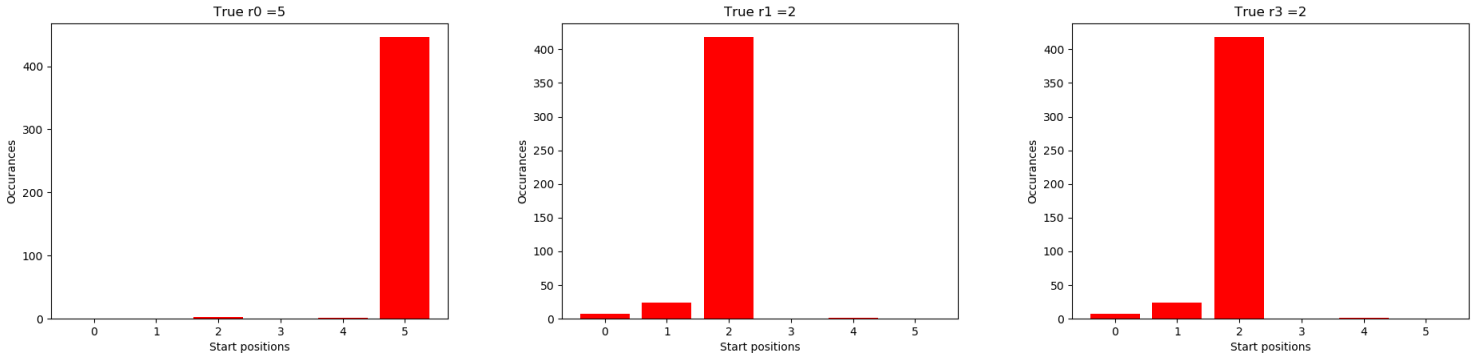


Figure 3: Histograms for 3 start positions for Chain 3

2 2.2 MCMC for the Train

Question 3 - Runtime complexity: The conditional likelihood is: $p(\mathbf{o}|G,\mathbf{x},s_1) = \prod_{t=2}^{T}\sum_{s_t} p(o_t|s_t,G,\mathbf{x})p(s_t|s_{1:t-1},G,\mathbf{x})$.
When $s_1, G, \mathbf{x}$ are given there is only a single path that yields non-zero probabilities. Therefore we do not need to sum over all states s_t instead we only consider the known non-zero path and thus we need to perform $T-1$ computation. Which yields the computational complexity: O(T)
Question 4/Question 6: We sample from the posterior $p(s_1, \mathbf{x}|G,\mathbf{o})$ using a metropolis-Hastings within Gibbs sampler.

- First we use Gibbs update for s_1 by sampling from $p(s_1|G,\mathbf{o},\mathbf{x})$. Which we can rewrite as: $p(s_1|G,\mathbf{o},\mathbf{x}) = \frac{p(s_1,\mathbf{o},\mathbf{x})}{p(\mathbf{o},\mathbf{x})} = \frac{p(\mathbf{o}|s_1,\mathbf{x})p(s_1|\mathbf{x})p(\mathbf{x})}{p(\mathbf{o}|\mathbf{x})p(\mathbf{x})} = \frac{p(\mathbf{o}|s_1,\mathbf{x})p(s_1)}{p(\mathbf{o}|\mathbf{x})}$ Where we utilize that s_1 is conditionally independent of $\mathbf{x}$. Thus the first term in the numerator simply becomes the conditional likelihood from equation (2) and the prior $p(s_1)$ is given in the assignment description as $\frac{1}{|V|}$ here $V = \frac{1}{9}$. Lastly note that the denominator is equal to the marginalization over $s1$ in the denominator, i.e. $\sum_{s_1} p(\mathbf{o}|s_1,\mathbf{x})p(s_1)$.

- Once we have the expression for the posterior we use a Gibbs sampler to update s_1. Accordingly we compute $p(\mathbf{o}|s_1,\mathbf{x})p(s_1)$ for all 9 possible start positions in the graph. Then we normalize these values to get a valid distribution and perform categorical sampling to get a new s_1 which is then used when we update the switch states.

- Next we use Metropolis Hastings to update the switch states.**(Question 6)** First in a similar manner we rewrite the expression for the posterior $p(\mathbf{x}|G,\mathbf{o},s_1) = \frac{p(s_1,\mathbf{o},\mathbf{x})}{p(\mathbf{o},s_1)} = \frac{p(\mathbf{o}|s_1,\mathbf{x})p(\mathbf{x}|s_1)p(s_1)}{p(\mathbf{o}|s_1)p(s_1)} = \frac{p(\mathbf{o}|s_1,\mathbf{x})p(\mathbf{x})}{p(\mathbf{o}|s_1)}$. Where we utilize that $\mathbf{x}$ is conditionally independent of s_1. The prior of $\mathbf{x}$ is given in the description as $\frac{1}{3}$ and note that $p(\mathbf{o}|s_1) = \sum_x p(\mathbf{o}|s_1,\mathbf{x})p(x)$.

- Next we sample a new switch state, from an uniform distribution, for the first switch (0, 0) so that the first element in the vector $\mathbf{x}$ is updated. Next we want to compute the acceptance ratio between the posteriors of the new proposed **x'** and the old $\mathbf{x}$: $\frac{p(\mathbf{o}|s_1,\mathbf{x'})p(\mathbf{x'})}{p(\mathbf{o}|s_1)}\frac{p(\mathbf{o}|s_1)}{p(\mathbf{o}|s_1,\mathbf{x})p(\mathbf{x})} = \frac{p(\mathbf{o}|s_1,\mathbf{x'})}{p(\mathbf{o}|s_1,\mathbf{x})}$ which is simply the fraction between the conditional likelihood of the proposed and the old $\mathbf{x}$. The acceptance probability is given by $r = min(1, \frac{p(\mathbf{o}|s_1,\mathbf{x'})}{p(\mathbf{o}|s_1,\mathbf{x})})$. Lastly we draw a sample u from a symmetric uniform distribution, $u \sim U(0,1)$. If $u < r$ we accept the proposal, otherwise we keep our old $\mathbf{x}$. Next we use the same procedure for the 8 remaining switch states, the resulting vector is then used to update s_1 once again and we go back to the first step.

Generated data: Due to the space limitation for the report we only analyzed data for the sampled start positions. To generate G, X-truth, s-truth and o the following parameter setting was used: $n - lattice = 3$, $T = 100$ and $p = 0.1$. The number of iterations was set to 1000. Using a lag of 5 and a burn in period of 100 iterations the acceptance rate for this sampler can be found in the table below. We see that all three chains had high acceptance rates.

Question 5 -

MH within Gibbs	Chain 1	Chain 2	Chain 3
Acceptance rate	0.845	0.843	0.851

Convergence - MH within Gibbs: Our approach to examine convergence is to run multiple chains for the same parameter settings for G, X-truth and o, i.e. we analyze the same data. We then make sure that all three chains converge to the same start position despite. For all of the results below, we used a burn-in period of 100, a lag of 5 to reduce correlation and 1000 iterations. We use the same parameter settings as mentioned in Question 4. The seeds for the three different chains where: 225, 11 and 2222. Please look at figure 8a for our evidence of convergence.

Question 7 - Block size: We used a block size of 3 and 5 switch states. Where the first block consisted of the 0:th 2:nd and 4:th indices in the graph, the second block consisted of the 3:rd, 5:th and 7:th indices in the graph and the last block consisted of 1:th, 6:th and 8:th indices in the graph. These where chosen since the edges within these blocks are not connected and thus they are independent.
Question 8 - Comparing Algorithms
Accuracy & Convergence Rates: We compare the accuracy and convergence by looking at the sampled posteriors $p(s_1|G,\mathbf{o},\mathbf{x})$ for the three algorithms. Accuracy was computed as by counting the number of correctly sampled start positions divided by total number of sampled start positions using the above mentioned lag and burn-in.

Using 1000 iterations we obtain the histograms on the next page. All of the tests where run on the same data, but three other seeds where then used for sampling such that the individual rows in the histograms share the same seed. The results of an additional chain can be found in the Appendix (REF).

From the provided 6 histograms we can see that all three chains for all three algorithms converge to the true starting position for this data set, which was (2, 1).

Algorithm	Chain 1 - Accuracy	Chain 2 - Accuracy	Chain 3 - Accuracy
MH within Gibbs	0.922	0.944	0.967
Gibbs	0.967	0.944	0.972
Blocked Gibbs	0.972	0.967	0.972

However, it should be noted it is possible to pick a seed for data generation such that the train barely visits any of the states. If so the results become much poorer.

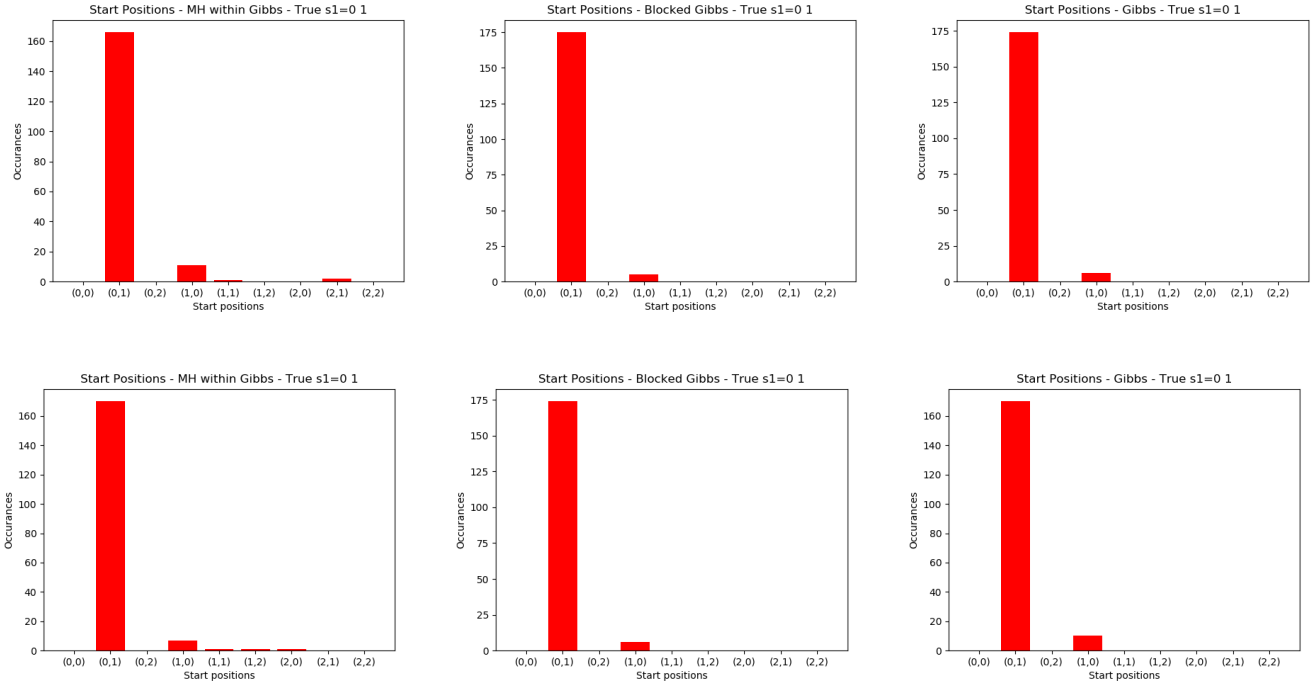


Figure 4: Histograms for sampled start positions for all three algorithms for three Chains

To examine convergence, we compare the first 100 samples, after the burn-in period with the last half of the samples. If the histogram looks the same, we have shown convergence. From the histogram we can thus conclude that all our three methods have converged.

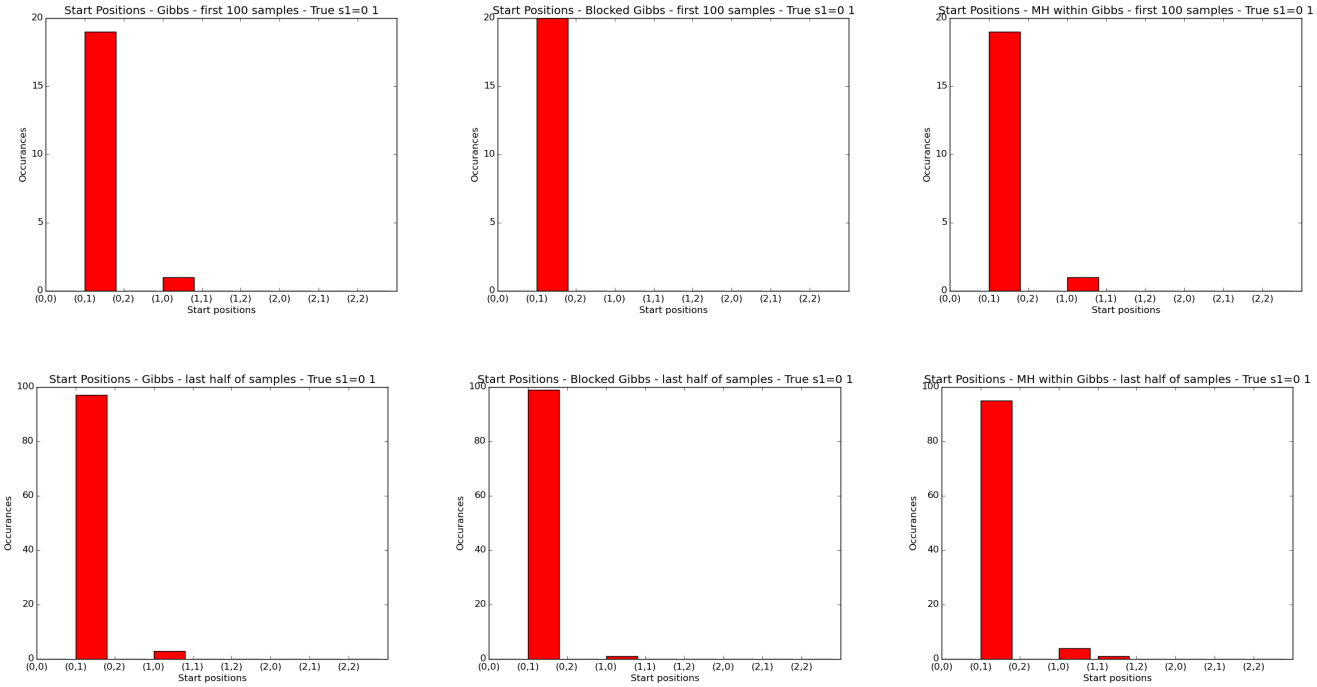


Figure 5: Histograms for sampled start positions for all three algorithms for three Chains

3 2.3 SMC for the stochastic volatility model

3.1 Question 10: Sequential importance sampling

Algorithm 2 Sequential Importance Sampling

for n = t:T **do**
  for n = 1:N **do**
    if $t = 1$: **then**
      $x_1^n \sim \mathcal{N}(0,\, \sigma^2)$
      $w_1^n = \alpha(x_1^n)$
    else
      $x_1^n \sim \mathcal{N}(\phi x_{t-1},\, \sigma^2)$
      $w_t^n = \alpha(x_t^n) \cdot w_{t-1}^n$

i, Choice of proposal: Our choice of proposal is our prior distribution f, i.e. $q_t = f(x_t|x_{t-a})$. We do this due to the simplicity.

i, Weight update function: Which gives us the following weight function:

$$w(x_{1:t}) = \frac{\gamma(x_{1:t})}{V_t(x_{1:t}} = \frac{\gamma_{t-1}(x_{1:t-1}}{V_t(x_{1:t-1})} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q_t(x_t|x_{1:t-1})} = \tag{7}$$

$$\frac{\gamma_{t-1}(x_{1:t-1}}{V_t(x_{1:t-1})} g(y_t|x_t) = w(x_{1:t-1})\alpha(x_{1:t-1,x_t}) \tag{8}$$

~~Thus, the the weight update simply becomes $\alpha(x_{1:t-1,xt}) = g(y_t|x_t)$.~~

i, Point estimate formula: The point estimate of x_T is given by $\hat{x}_T = \sum_{n=1}^{N} \overline{w}_T^n \cdot x_T^n$. Where N is the number of particles, T is the last time step and $\overline{w}_T^n$ denotes the normalized weight of particle n at time step T.

ii, Compute the point estimate: Our point estimate for x_T when using sequential importance sampling was $\hat{x}_T = -4.9086988809$
and $x_{truth,T} = -7.53893550613$

iii, Variance: The empirical variance of the normalized weights $\overline{w}_T$ at the last time step T, using 100 particles, was: 0.0099

ii, MSE: The mean squared error of the estimate to the truth was computed as: $E[f(X_T)] \sim \sum_{k=1}^{K} \bar{w}_k (x_T^k - x_T^*)^2$. How the MSE changes when we increase the number of particles is shown in the table below.

iii, Histograms:

Number of particles	MSE
100	7.87149151459
200	13.9298931244
300	5.03814615936
400	6.49075743086
500	6.14477598376
600	6.96888782863
700	0.635545924397
800	3.87148894897

Table 1: MSE for SIS

# of particles	MSE
100	0.744468828346
200	0.602709588287
300	0.690842640775
400	0.749419235159
500	0.701535471827
600	0.744444377499
700	0.718056808442
800	0.575885049097

Algorithm 3 Sequential Importance Sampling with resampling

for n = t:T **do**
  for n = 1:N **do**
    if $t = 1$: **then**
      $x_1^n \sim \mathcal{N}(0, \sigma^2)$
      $w_1^n = \alpha(x_1^n)$
      $\overline{w}_1^n = w_1^n / \sum_{k'=1} w_1^{k'}$
    else
      $j \sim Multinomial(\overline{w}_{t-1}^1, ..., \overline{w}_{t-1}^N)$
      $x_1^n \sim \mathcal{N}(\phi x_{t-1}, \sigma^2)$
      $\mathbf{x}^n = (x_{1:t-1}^j, x_t^n)$
      $w_t^n = w_{t-1}^n \cdot \alpha(x_t^n)$
      $\overline{w}_t^n = w_t^n / \sum_{k'=1} w_t^{k'}$

Figure 7: sample

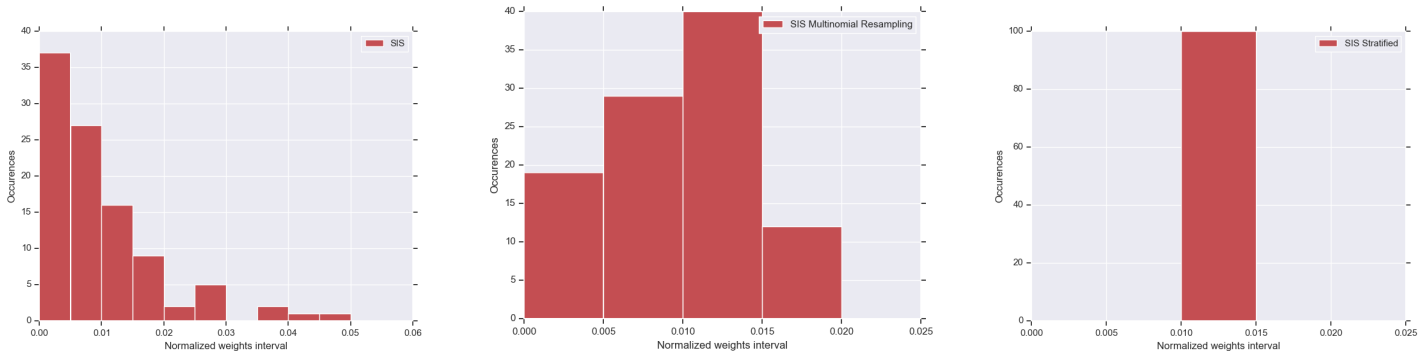


Figure 6: Histograms for the three different SIS models

3.2 Question 11: SIS with resampling

ii, Compute the point estimate: Our point estimate for x_T when using SIS with resampling was $\hat{x}_T = -7.16864915882$ and $x_{truth,T} = -7.53893550613$

iii, Variance: The empirical variance of the normalized weights $\overline{w}_T$ at the last time step T, using 100 particles, was: $2.12248111609e-05$

ii, MSE: The mean squared error of the estimate to the truth was computed as: $E[f(X_T)] \sim \sum_{k=1}^{K} \bar{w}_k (x_T^k - x_T^*)^2$. How the MSE changes when we increase the number of particles is shown in the table below.

3.3 Question 12: SIS with Stratified Resampling

Algorithm 4 Sequential Importance Sampling with Stratified Resampling

$\mathbb{E}[N_n^{(i)}] = NW_n^{(i)}$ but $\mathbb{V}[N_n^{(i)}] < NW_n^{(i)}(1 - W_m^{(i)})$
Select $U_1 \sim \mathcal{U}[0, \frac{1}{N}]$
for i = 2:N **do**
    $U_i = U_1 + \frac{i-1}{N} = U_{i-1} + \frac{1}{N}$
    $N_n^{(i)} = \#\{U_j : \sum_{m=1}^{i-1} W_n^{(m)} \leq U_j \leq \sum_{m=1}^{i} W_n^{(m)}\}$
    where $\sum_{m=1}^{0} = 0$

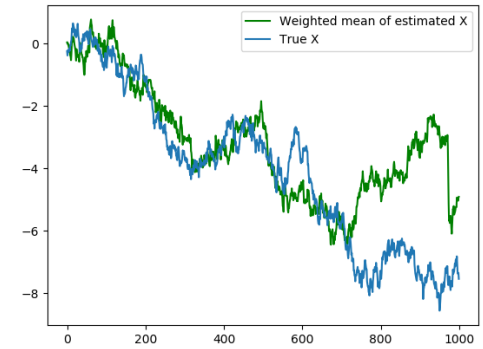
ii, Compute the point estimate: Our point estimate for x_T when using SIS with stratified resampling was $\hat{x}_T = -6.99118118081$ and $x_{truth,T} = -7.53893550613$

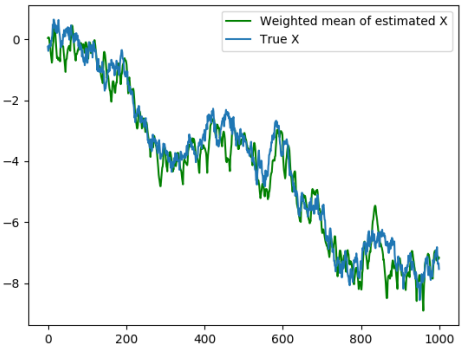
iii, Variance: The empirical variance of the normalized weights $\overline{w}_T$ at the last time step T, using 100 particles, was: 3.00926553811e-36

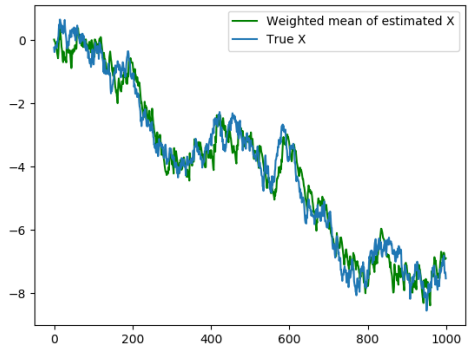
ii, MSE: The mean squared error of the estimate to the truth was computed as: $E[f(X_T)] \sim \sum_{k=1}^{K} \bar{w}_k (x_T^k - x_T^*)^2$. How the MSE changes when we increase the number of particles is shown in the tablesd below.

3.4 Question 13

Below the plots show the weighted mean of the latent variables X at each time step . We can see that the sequential importance samplers that uses re sampling perform better that the normal sequential importance sampler. Which is probably since unimportant particles with low weights do not get resampled.







4 2.4 Stochastic volatility unknown parameters I

The assignment is to use Particle Marginal Metropolis Hastings (PMMH) to jointly infer the model parameters θ and the latent variables $X_{1:T}$ of a Stochastic Volatility Model. Where the latent variables denotes the underlying volatility of some financial asset and the observed variables $Y_{1:T}$ denotes the scaled log-returns from this asset. In the assignment we are given a stochastic volatility model where two of the variance parameters β and σ are unknown. The stochastic volatility model is:

$$\begin{aligned} X &\sim \mathcal{N}(x_1|0,\sigma^2) \\ X_t|(X_{t-1}=x_{t-1}) &\sim \mathcal{N}(x_t|\phi x_{t-1},\sigma^2) \quad t=1,...T \\ Y_t|(X_t=x_t) &\sim \mathcal{N}(y_t|0,\beta^2 exp(x_t)) \quad t=1,...T \end{aligned}$$

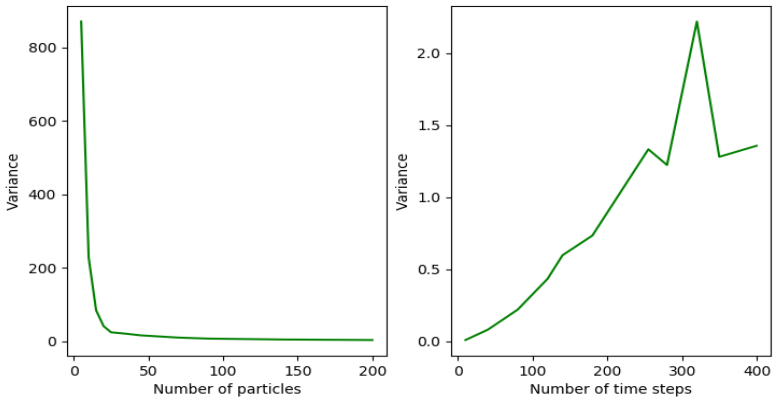
- Initialization:
  - $\theta(0)$ is initialized by giving β and σ arbitrary initial values and by setting ϕ equal to its fixed value 1.
  - $\{Z^*,\bar{w},x^k\}_{k=1}^K \leftarrow SMC(\theta(0))$ we first infer the hidden states using SMC with multinomial resampling and estimate the marginal likelihood $p_\theta(y_{1:T})$ (see equation 4 in the assignment descripiton) .
  - $k^{'} \sim \bar{w}^k$ We sample a trajectory with multinoimail resampling, again using the normalized weights and set: $\mathbf{x}(0) = \mathbf{x}^{k^{'}}$ and $\hat{Z}(0) = Z^*$.
- For n = 1,...,numIter:
  - $\theta^*|\theta(n-1) \sim q_\theta(\cdot|\theta(n-1))$ We then propose a new theta by using a Gaussian random walk on the θ space as our proposal distribution for β and σ .
  - $\{Z^*,\bar{w},x^k\}_{k=1}^K \leftarrow SMC(\theta^*)$
  - $k^{'} \sim \bar{w}^k$
  - Then we compute the acceptance ratio as: $\alpha = min(1, \frac{\hat{Z}^*}{\hat{Z}(n-1)} \frac{p(\theta^*)}{p(\theta(n-1))})$ Note that since we are using a symmetric proposal $q_\theta(\theta^*|\theta(n-1)) = q_\theta(\theta(n-1)|\theta^*)$ and thus these terms cancel. Also, the prior distributions for $sigma^2$ and $beta^2$ are given in the assignment as inverse Gamma priors with parameters $a = 0.01, b = 0.01$.
  - With probability α we set: $(\theta(n), x(n), \hat{Z}) = (\theta^*, x^{k^{'}}, \hat{Z}^*)$ Otherwise set: $(\theta(n), x(n), \hat{Z}) = (\theta(n-1), x(n-1), \hat{Z}(n-1))$

4.1 Question 14

Below is the mean of 10 log likelihood estimates for different parameter settings of σ and β we can see that the maximum likelihood estimate of $\beta = 1.75$ and $\sigma = 1.5$ is far of from the ground truth.

	$\sigma=0.25$	$\sigma=0.5$	$\sigma=0.75$	$\sigma=1$	$\sigma=1.25$	$\sigma=1.5$	$\sigma=1.75$
$\beta=0.25$	-93.498	-93.892	-93.288	-94.628	-94.84	-97.75	-98.31
$\beta=0.5$	-96.378	-95.896	-95.998	-97.129	-97.39	-100.586	-98.73
$\beta=0.75$	-100.05	-100.13	-100.19	-100.33	-103.25	-100.25	-101.9
$\beta=1$	-102.48	-102.96	-101.97	-104.408	-105.47	-104.04	-105.79
$\beta=1.25$	-107.61	-107.578	-107.044	-108.15	-106.74	-108.55	-108.18
$\beta=1.5$	-112.26	-108.8	-112.5	-112.018	-111.782	-111.231	-111.50
$\beta=1.75$	-115.08	-115.65	-114.82	-115.33	-116.31	-116.55	-114.66

4.2 Question 15



The variance decreases exponentially as we increase the number of particles in each time steps. While the variance increases as we increase the number of time steps T.

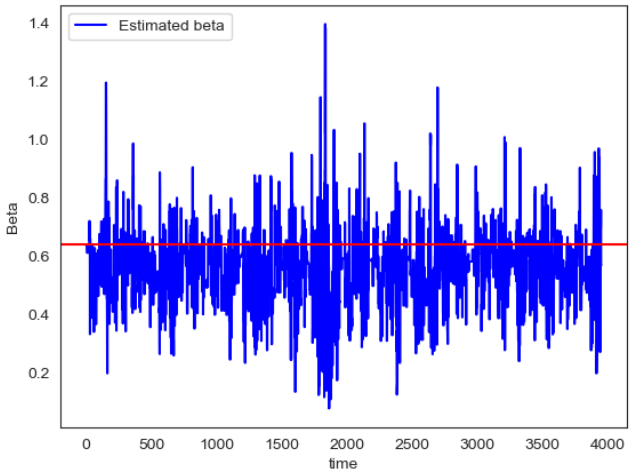
4.3 Question 16

NOTE: Unfortunately we mistakenly used $T = 1000$ instead of $T = 100$as stated in the assignment description. However, we did not notice this until right before the assignment was due. **i.Proposal distribution** As previously mentioned we use a Gaussian random walk proposal distribution for the variance parameters β and σ. The parameter proposal is a normal multivariate Gaussian distribution centered around the parameters from the previous iteration $\theta(n-1)$, with covariance matrix $\sum = I_2 \cdot \epsilon^2$ where I_2 is the identity matrix and the covariance thus is a matrix with the ϵ values along its diagonal. Also $\epsilon > 0$ and is the step size of the random walk.

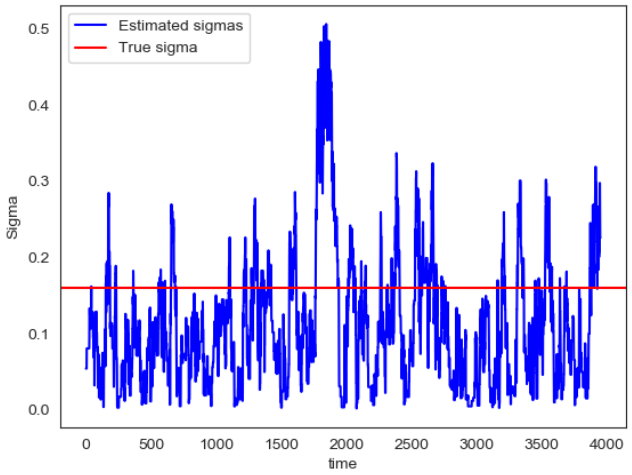
i.Acceptance probability The acceptance probability was computed as:

$\alpha = min(1, \frac{\hat{Z}^*}{\hat{Z}(n-1)} \frac{p(\theta^*)}{p(\theta(n-1))})$

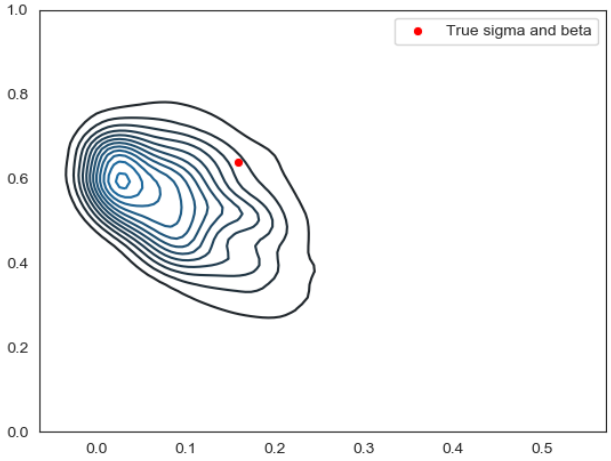
Since we are using a symmetric proposal $q_\theta(\theta^*|\theta(n-1)) = q_\theta(\theta(n-1)|\theta^*)$ and thus these terms cancel. Also, the prior distributions for $sigma^2$ and $beta^2$ are given in the assignment as inverse Gamma priors with parameters $a = 0.01, b = 0.01$.



(d) Traceplot for the inferred β



(e) Traceplot for the inferred σ



(f) Density plot for σ and β

Appendix:

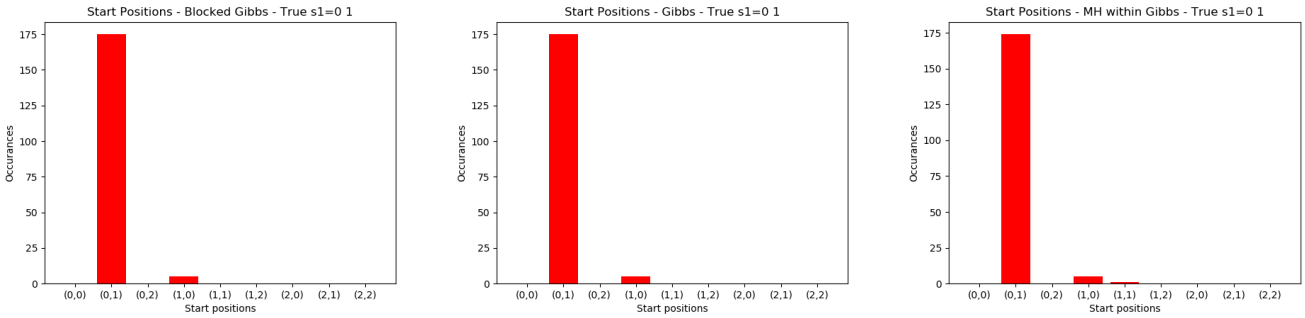


Figure 8: Histograms for sampled start positions for all three algorithms for one Chain not included in section 2.2

References