

University of London
Imperial College of Science, Technology and Medicine
Department of Mathematics

Portfolio Optimisation with Sequential Monte Carlo

Yow Tzu Lim

Submitted in part fulfilment of the requirements for the degree of
MSc in Statistics of Imperial College, London, September 2014.

Abstract

Text of the Abstract.

Acknowledgements

I would like to express my gratitude to all who have provided support in accomplishing this thesis. I would like to thank my supervisor, Nikolas Kantas, for the opportunity given to carry out this research and also for his guidance, expertise and encouragement during the course of this research. I would like to thank the support staff, especially XYZ and ABC, who have maintained the Tesla grid servers, on which my experiments have been carried out.

I would also like to express my appreciation to my wife Ka Ki Lai for her endless support and care in my personal life during the course of this master degree study. I would like to thank God Almighty, who has given me the strength and peace to work hard on this project.

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Contributions	1
1.3 Tecnical Approach	1
1.4 Thesis organsiation	1
2 Monte Carlo Integration	3
2.1 Perfect Monte Carlo	4
2.2 Rejection sampling	5
2.3 Importance sampling	5
2.4 Markov chain Monte Carlo (MCMC)	6
2.4.1 Markov Chain	6
2.4.2 Metropolis-Hastings sampling	7

2.4.3	Gibbs sampling	8
2.5	Hidden Markov Models	8
2.5.1	Kalman Filter	10
2.6	Sequential Monte Carlo	11
2.6.1	Sequential Important Sampling (SIS)	11
2.6.2	Sequential Important Resampling (SIR)	12
2.6.3	Parameter estimation	12
2.7	Conclusion	12
3	Conclusion	13
3.1	Summary of Thesis Achievements	13
3.2	Applications	13
3.3	Future Work	13
	Bibliography	13

List of Tables

List of Figures

2.1 Hidden Markov Models 9

Chapter 1

Introduction

1.1 Motivation and Objectives

Portfolio optimisation is not a new concept.

In this project, we attempt to use SMC to solve the portfolio optimisation problem. The fundamental goal of portfolio theory is to optimally allocate resources to different investible assets. The meaning of optimality may be different to the investors. Some investors may have tax considerations; others may have different legal restrictions on investments or holding periods. Two common objectives, often contradicting, are the investment risk and return. We adopt here the Markowitz's mean-variance efficient portfolio which allow one to make the portfolio allocation by considering the trade-off between expected risk and expected return.

1.2 Contributions

Contributions here.

1.3 Tecnical Approach

1.4 Thesis organsiation

The subsequent chapters of this thesis are organised as follows:

1. Chapter 2 provides an overview on the Monte Carlo methods. It begins with a brief introduction to basic methods such as perfect Monte Carlo sampling, rejection sampling,

importance sampling. It then details two common Markov Chain Monte Carlo (MCMC) techniques, namely Metropolis-Hastings and Gibbs Sampling. Lastly, it introduces the SMC technique used in this thesis.

2. Chapter 3 presents the probl
3. Chapter 4 the formulation of the experiments in using SMC to discover the optimal policies for the given portfolio. It begins with the problem formulation.
4. Chapter 5 concludes the thesis by evaluating the degree to which the hypothesis has been justified and outlines potential work for the future.

Chapter 2

Monte Carlo Integration

The application of Monte Carlo methods in the Statistics domain is increasing popular. In the Bayesian framework, unknown parameters are assumed to be random variables and admit prior distribution that characterises the initial belief. The belief is updated with new observable evidence using Bayes theorem.

More formally, let specify the initial belief of a Bayesian model, M as the prior disitribution $p(\theta | M)$, the probability of observing an observation x given the model (likelihood) as $p(x | \theta, M)$. Using Bayesin Theorem, the updated belief (posterior distribution) $p(\theta | M)$ is given as follows:

$$\begin{aligned} p(\theta | x, M) &= \frac{p(x | \theta, M)p(\theta | M)}{p(x | M)} \\ &\propto p(x | \theta, M)p(\theta | M) \end{aligned} \tag{2.1}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior} \tag{2.2}$$

Whilst the problem formulation is elegant, there remains some subtle issues, e.g., the calculation of normalising constant, the choice of prior, etc. Refer [1] for details. The discussion only focuses on how Monte Carlo integration techniques is used to solve some of the issues involved in Bayesian framework.

To calculate the normalisation constant in $p(x | M)$ in (2.1), we would need to be able integrate the following

$$p(x | M) = \int p(x | \theta, M)p(\theta | M)d\theta \tag{2.3}$$

analytically which is often not feasible. The Method Carlo integration methods allow one to compute an estimate of this quantity using samples generated from simulation¹.

¹Traditionally, the need of normalisation constant is often circumvented by making use of conjugate prior (the choices of certain prior distributions that yield posterior distributions from the same family) in an analytical fashion.

The need of calculate the integral of an espression also arise in the marginalisation of nuisance parameter. For example, let y be an nuisance parameter, the marginal density of x can be obtained from the joint density as follows:

$$p(x | y, M) = \int p(x, y | M) dy \quad (2.4)$$

If the integral cannot be computed analytically, Monte Carlo integration methods can be used to estimate the quantity of interest.

Another related problem is to calculate the expectation. This often arise when one attempts to summarise the posterior distribution in some way. For example, the expected value of a function of the parameter:

$$I = E[f(x)] = \int f(x)p(x)dx \quad (2.5)$$

where $f(\cdot)$ is some function of interest, $p(\cdot)$ is the probabilty density function of x . Again, this may not have an analytic solution. We may use Monte Carlo integration methods to compute an estimate for this integral.

2.1 Perfect Monte Carlo

In the simplest setting, assuming we are able to sample N independent and identically distributed (i.i.d.) points of x from $p(\cdot)$, denote these as $\{x^i\}$ where i is from $1 \dots N$, a Monte Carlo estimate of I using the the point masses of the samples is:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(x^i) \quad (2.6)$$

Informally, this estimate approximates the continous distribution with a discrete distribution with random support. It can be shown that this estimate is unbiased and converge almost surely to I as $N \rightarrow \infty$ (Law of Large number).

If the variance of $f(\cdot)$ is bounded ($\sigma_f^2 < \infty$), then the following central limit theorem holds:

$$\sqrt{N}(\hat{I} - I) \implies N(0, \sigma_f^2) \text{ as } N \rightarrow \infty \quad (2.7)$$

where \implies denotes convergence in distribution. It is important to note that the converengece rate of $\frac{1}{\sqrt{N}}$ is independent of the dimensions of x . This is in constrast with any determinstic method that has a rate that decreases a the integral dimension increases [5]. This is the main advantage of Monte Carlo integration.

2.2 Rejection sampling

However, it is not always possible to sample directly from the distribution p . Suppose we can find an instrumental distribution q that is easy to sample from such that $cq(x)$ dominates p such that $cq(x) \geq p(x) \geq 0$ for all x . then to get a random sample from p , we can first sample from q instead and accept the sample with acceptance probability $\alpha(x) = \frac{p(x)}{cq(x)}$. If the sample is rejected, repeat the process until success. The algorithm is summarised in Algorithm 1.

Algorithm 1 Rejection Sampling

```

1: function REJECTION_SAMPLING( $n$ )
2:    $r = []$ 
3:   repeat
4:     sample  $x \sim q$ 
5:     sample  $u \sim \mathcal{U}(0, 1)$ 
6:     if  $u \leq \frac{p(x)}{cq(x)}$  then
7:        $r \leftarrow [r, x]$ 
8:   until  $\text{len}(r) = n$ 

```

One could easily see that the optimal choice of proposal distribution q^* is that one minimize the area subject to the constraint that it dominates the target density p . Unfortunately, in practice a rejection sampling is only feasible in low dimension and then it becomes very ineffective as the dimension increases as lots of samples generated samples would be rejected.

2.3 Importance sampling

Instead of making a binary accept-reject decision on each sample, the main idea behinds important sampling is to weight each sample based on how well the sample from the proposal distribution q resembles the target distribution p . More formally, assume we have an instrumental distribution q that is easy to sample from which has support that includes p , we can write (2.5) as:

$$\begin{aligned}
 I &= \int f(x) \frac{p(x)}{q(x)} q(x) dx \\
 &= \int f(x) w(x) q(x) dx
 \end{aligned} \tag{2.8}$$

where $w = \frac{p(x)}{q(x)}$, which is commonly known as the importance weight. Based on this reformulation, a possible Monte Carlo estimate of I is given as follows:

$$\hat{I} = \frac{\frac{1}{N} \sum_{i=1}^N w(x^i) f(x^i)}{\frac{1}{N} \sum_{j=1}^N w^j} = \sum_{i=1}^N \tilde{w}^i f(x^i) \quad (2.9)$$

where $\tilde{w}^i = \frac{w^i}{\sum_{j=1}^N w^j}$ is the normalised importance weight. This estimate is biased as it uses the ratio of two estimator but it is still asymptotically consistent.

2.4 Markov chain Monte Carlo (MCMC)

Markov chain Monte Carlo (MCMC) are a set of algorithms that allows ones to draw random samples from the target probability distribution by constructing a ergotic Markov chain process which has its stationary distribution set to be target desired distribution.

2.4.1 Markov Chain

Conceptually, Markov chain is a stochastic process in which the past and future states of the process are independent given the current state. A Markov chain can be viewed as an *ordered* sequence of states $\{x_{1:N}\}$, in which state x_t only depends on state x_{t-1} , i.e., the state at time t is determined based on some transition distribution of the form $p(x_t | x_{t-1})$, which is independent of all previous states x_{t-2}, x_{t-3}, \dots

To ensure the Markov chain converges to a steady state (stationary distribution), the chain needs to satisfy the following properties:

1. irreducible — no matter where it starts a chain, it is possible to reach all other states in finite amount time steps.
2. aperiodic — for all the states i , the chain can return to state i at irregular time.
3. positive recurrent — for all the state i , the chain will definitely revisit the state in finite amount time stpes.

In practice, it is usually not difficult to construct the required Markov Chain, the more difficult problem is to determine how many steps it takes to converge, often known as mixing. Various visual diagnostic have been developed, yet it is often a matter of art. Refer [5] for details.

In the following sections, we present two basic MCMC algorithms: Metropolis Hastings algorithms, Gibbs algorithms and a third algorithm in which these two basic algorithms are “mixed and matched” in a hierarchical fashion to form a more advanced samplers to fit application needs.

2.4.2 Metropolis-Hastings sampling

The first major MCMC algorithm was devised by Metropolis in [4]. The algorithm constructs a Markov Chain by proposing a random walk step based a proposal distribution is symmetric, i.e., the probability of moving from state t to state $t-1$ is the same as the probability of moving from state $t-1$ to state t ($q(x_t | x_{t-1}) = q(x_{t-1} | x_t)$). The proposed step is then either accepted or rejected according to an acceptance probability, $\alpha(x) = \min \left(1, \frac{p(x_t)}{p(x_{t-1})} \right)$, where $p(x)$ is the target probability density evaluated at x .

This algorithm was generalised by Hastings [3] to allow the use of non-symmetric proposal density, with an adjustment to the acceptance probability $\alpha(x) = \min \left(1, \frac{p(x_t)q(x_t | x_{t-1})}{p(x_{t-1})q(x_{t-1} | x_t)} \right)$ and hence the name, Metropolis-Hastings sampling. This algorithm is summarised in Algorithm 2.

Algorithm 2 MetropolisHastings

```

1: function METROPOLISHASTINGS(n)
2:    $r = []$ 
3:   repeat
4:     sample  $x_t \sim q(x_{t-1})$ 
5:     sample  $u \sim \mathcal{U}(0, 1)$ 
6:     if  $u \leq \min \left( 1, \frac{p(x_t)q(x_t | x_{t-1})}{p(x_{t-1})q(x_{t-1} | x_t)} \right)$  then
7:        $x_{t-1} \leftarrow x_t$ 
8:        $r \leftarrow [r, x_{t-1}]$ 
9:   until  $\text{len}(r)=n$ 

```

It is not hard to see that the algorithm works best if the density q matches the shape as density p , which is often unknown. Assuming a Gaussian proposal density q is used, there is also an additional parameter, namely the variance σ^2 is required to be tuned. There is a trade-off to be made in the tuning process. If the σ^2 is set too be too small, the chain is likely to mix slowly (i.e., the proposed step is likely to be accepted yet each successive move is very slow and therefore the chain will converge to target distribution p at a slow rate). On the other hand, if σ^2 is too large, the proposed step is likely to be rejected as it is likely to land in a region of much lower probability density and the chain will converge slowly again. Again, refer [5] for details on convergence diagnostic.

2.4.3 Gibbs sampling

Gibbs sampling [2] is a special form of the MetropolisHastings sampling. The key idea in Gibbs sampling is to re-write the target multivariate distribution of interest as a product of conditional probability distributions of lesser number of parameters. A Markov Chain is then constructed to generate samples for each parameter *in turn* from the conditional distributions with an acceptance rate $\alpha(x) = 1$, i.e., accept all the samples.

More formally, suppose we would like to have obtain k samples of $\mathbf{x} = (x_1, x_2, \dots, x_n)$ from $p(x_1, \dots, x_n)$, assuming we start with an initial value \mathbf{x}^0 . For each sample ($\mathbf{x}^i : i \in \{1 \dots n\}$), sample $x_j^{(i)}$ from the conditional distribution $p(x_j | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \dots, x_n^{(i-1)})$. In other words, sample each variable from the distribution of that variable conditioned on all other variables, making use of the most recent values and updating the variable with its new value as soon as it has been sampled. The algorithm is summarised in Algorithm 3.

Algorithm 3 Gibbs

```

1: function GIBBS(n)
2:    $r = []$ 
3:   sample  $x^{(0)}$ 
4:   repeat
5:     sample  $x^{(t)} \sim p(x_j | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \dots, x_n^{(i-1)})$ 
6:      $r \leftarrow [r, x^{(t)}]$ 
7:   until len(r)=k

```

There are variations of Gibbs sampling techniques. For examples, a blocked Gibbs sampling groups more than one variables and sample from the joint distribution conditioning on other variables. Refer [5] for details.

2.5 Hidden Markov Models

Hidden Markov Models (HMMs) is a class of models that have been applied to many different problems in statistics, signal processing and engineering. Many Monte Carlo techniques have been developed to deal with inference in HMMs, one of which that we have particular interest in is the Sequential Monte Carlo (SMC). This section provides a brief background on HMMs that is essentially to understand SMC. Refer [?] for further details of inference techniques for HMMs in general.

HMMs can be viewed as a class of models that consist of two related process: an underlying Markov process, X_t , which is the target process of interest, and a observable process, Y_t , which its state can be measured and therefore provides some information about X_t . Moreover, it is assumed that these two processes also have conditional independence properties as shown

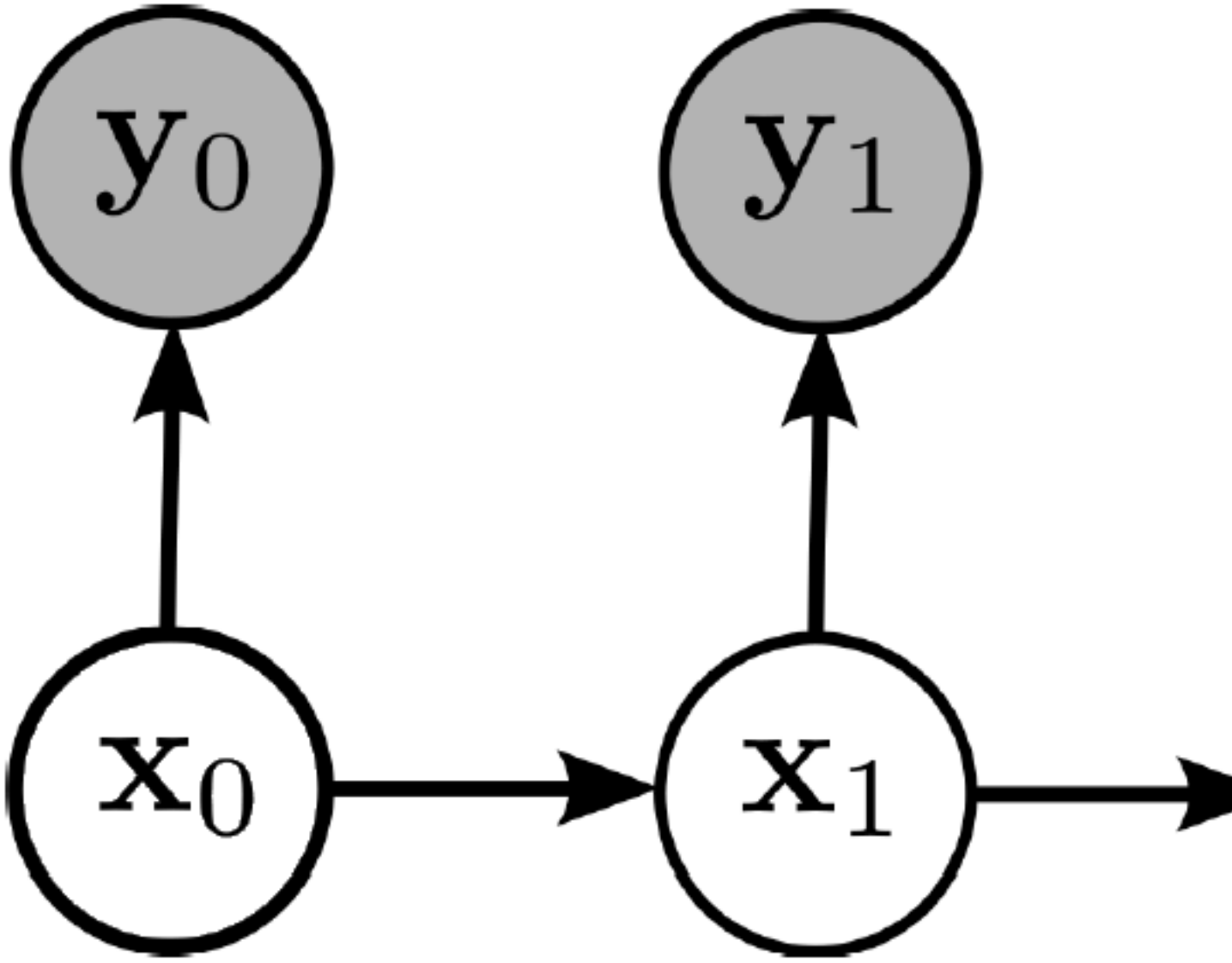


Figure 2.1: Hidden Markov Models

using the graphical model representation in Figure 2.1. More formally, these properties can be summarised as follows:

$$\begin{aligned}
 \Pr(X_n \mid X_{1:n}) &= \Pr(X_{n-1}) \\
 \Pr(Y_n \mid X_{1:n} = Y_{1:n}) &= \Pr(Y_n \mid X_{1:n}) \text{ conditional independent}
 \end{aligned}
 \tag{2.10}$$

It is worth emphasizing here that this model is designed to capture systems that evolve from one state to another over time, generating observation after each state move. The inference problem is typically about estimating the state(s) in *real-time* given the observations. This imposes an implicit requirement from the computation perspective that the estimate calculation cost should remain constant over time (i.e., the calculation cost does not increase with the increasing number of states).

Arguably, the most common inference problem in HMMs is the smoothing distribution, $p(x_{1:t} | y_{1:t})$, that is estimating the states $x_{1:t}$ based on the sequence of observations up to time t , $y_{1:t}$. Using Bayes rules, we can write the density of the distribution of interest as follows:

$$\begin{aligned} p(x_{1:t} | y_{1:t}) &\propto p(x_{1:t} | y_{1:t-1})g(y_t | x_t) \\ &\quad p(x_{1:t} | y_{1:t-1})f(x_t | x_{t-1})g(y_t | x_t) \end{aligned} \quad (2.11)$$

where $f(x_t | x_{t-1})$ is the transition density and $g(y_t | x_t)$ is the likelihood. This recursion is often re-written into two separate steps: prediction (the estimation of distribution of t states given only $t - 1$ states) and update steps (the correction of the predicted distribution taking into account the new observation as follows:

$$\begin{aligned} p(x_{1:t} | y_{1:t-1}) &= p(x_{1:t-1} | y_{1:t-1})f(x_t | x_{t-1}) \\ p(x_{1:t} | y_{1:t}) &= \frac{p(x_{1:t} | y_{1:t-1})g(y_t | x_t)}{\int p(x_{1:t} | y_{1:t-1})g(y_t | x_t)dx_{1:t}} \end{aligned} \quad (2.12)$$

Moreover, the estimate of any smoothing distribution $p(x_{j:k} | y_{1:t})$ where $j \leq k \leq l$ can be obtained by integrating out x that are not interested in as follows:

$$p(x_{j:l} | y_{1:t}) = \int p(x_{1:t} | y_{1:t})dx_{1:j,l+1:t} \quad (2.13)$$

One particular smoothing distribution of interest is the final marginal distribution $p(x_t | y_{1:t})$, which is often referred to as the filtering distribution.

Another distribution of interest is the prediction distribution, that is the estimation of the distribution of any unseen *future* state based on the sequence of observations up to time. Referring to (2.13), if we let $j = 1$ and $l \geq n$ in , we have:

$$p(x_{j:l} | y_{1:t}) = p(x_{j:t} | y_{1:t}) \prod_{i=t+1}^k f(x_i | x_{i-1}) \quad (2.14)$$

Therefore, any prediction density can be obtained by integrating out the variables of not interest from the above equation.

The problem of distribution estimation as discussed appears to be simple, but the problem is in fact far from being resolved in practice. The integral appear in the above equations are often intractable and can only be estimated.

2.5.1 Kalman Filter

In a special case setting, in which the transition distribution of $X_t | X_{t-1}$ and the likelihood are Gaussian distributions centred at a point of a linear combination of the conditional variables

of the following form:

$$\begin{aligned} f_n(x_n \mid x_{n-1}, u_n) &= N(A_n x_{n-1} + F_n(u_n), B_n B_n^T) \\ g_n(y + n \mid x_n, u_n) &= N(C_n x_n - 1, D_n D_n^T) \end{aligned} \quad (2.15)$$

Using the properties of Gaussian distribution, the integral can be resolved analytically. This leads the widely used *Kalman Filter* [?], which has the following recursion forms:

$$1 + 1 \quad (2.16)$$

There are various extensions to this approach. For example, the Extended Kalman Filtre (EKF) which uses Taylor Series expansion to linearise at the conditional variables locally, Unscentend Kalman Filter, etc.. Refer [?] for further details on Kalman Filter.

2.6 Sequential Monte Carlo

As mentioned, there is however no analytical solution for this estimation problem in a general settings for this model. A common technique to dealing with this problem is Sequential Monte Carlo (SMC). Conceptually, the distribution of a state at any point of time is represented using a set of weighted samples (a.k.a. particles). These particles are propagate from one state to another through time by proper re-weighting and re-sampling as necessary to track the changes of the distribution. To estimate the smoothing/filtering/prediction distribution, the integrals are performed on these samples using Monte Carlo technique in a recursion fashion, i.e., the estimation problem at step t is calculated with the estimate obtained at time $t - 1$, hence the name.

2.6.1 Sequential Important Sampling (SIS)

Recalled that importance sampling allows one to calculation the expectation with respect to a distribution π , $E_\pi[f(X)]$ using samples from a different proposal distribution q as $E_q[\pi(X)f(X)/q(X)]$.

Sequential importace sampling extends this idea further.

I The basis of sequential important sampling technique

2.6.2 Sequential Important Resampling (SIR)

2.6.3 Parameter estimation

2.7 Conclusion

In this chapter,

Chapter 3

Conclusion

3.1 Summary of Thesis Achievements

Summary.

3.2 Applications

Applications.

3.3 Future Work

Future Work.

Bibliography

- [1] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- [2] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, November 1984.
- [3] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.
- [4] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [5] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.