8-9-2014

Research on Approximate Bayesian Computation

Jiting Xu
University of South Carolina - Columbia

Research on Approximate Bayesian Computation

by

Jiting Xu

Bachelor of Engineering
China University of Geoscience (wuhan) 2008

Master of Science
University of South Carolina 2012

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Computer Science

College of Engineering and Computing

University of South Carolina

2014

Accepted by:

Gabriel Terejanu, Director of Thesis

Jijun Tang, Reader

Jianjun Hu, Reader

Lacy Ford, Vice Provost and Dean of Graduate Studies

Acknowledgments

I would never been able to finish the thesis without the guidance of my advisor and committee members, help from my colleague and friends, and support and love from my family members.

I would like to express my deepest appreciation to my advisor, Dr. Gabriel Terejanu, for his guidance, advice, support, and caring, which make this research possible. I am so fortunate to have a such perfect advisor in my life. Without his patience, guidance, advice, and support, I cannot overcome the crisis situation and finish the thesis.

I would like to thank my committee members, Professor Tang and Professor Hu, for reading and commenting my thesis. I am grateful to Prof. Tang who always gives me his encouragement and practical advice. He is one of the best professor I have had in my life. I am grateful to Prof. Hu for his amazing and insightful data mining courses. He introduced me into the data mining field and he has been always there to help me solve all kinds of problems. He is one of the best professor I have had in my life.

I am also thankful to Prof. Manton Matthews, Prof. Srihari Nelakuditi for giving me a chance to study in Computer Science field. I really enjoy the study and life in our great department.

I would like to thank all the stuff in CSE department for their assistance.

I am also grateful to all my friends for their different kinds of assistance during my life at USC.

Most importantly, I would like to thank my family for their unconditional love

and support. In particular, I would like to thank my wife, Can Chen. She was always supporting me and encouraging me through the good times and bad.

Abstract

This thesis presents the development of a new numerical algorithm for statistical inference problems that require sampling from distributions which are intractable. We propose to develop our sampling algorithm based on a class of Monte Carlo methods, Approximate Bayesian Computation (ABC), which are specifically designed to deal with this type of likelihood-free inference. ABC has become a fundamental tool for the analysis of complex models when the likelihood function is computationally intractable or challenging to mathematically specify. The central theme of our approach is to enhance the current ABC algorithms by exploiting the structure of the mathematical models via derivative information. We introduce Progressive Correction of Gaussian Components (PCGC) as a computationally efficient algorithm for generating proposal distributions in our ABC sampler. We demonstrate on two examples that our new ABC algorithm has an acceptance rate that is one to two orders of magnitude better than the basic ABC rejection sampling.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Bayesian Computation

Bayesian computation is by far the most widely used approach to statistical inverse problems. It employs probability to represent uncertainty, which enables the use of powerful tools based on Bayes' theorem to learn about the values of model parameters that match observations. Based on Bayes' theorem, we can easily update the posterior distribution of model parameter according to the new evidence:

$$p(\theta|D) = \frac{L(D|\theta)P(\theta)}{P(D)} \tag{1.1}$$

Where θ is the model parameter; D is the new observed data; $P(\theta)$ is the prior distribution; $L(D|\theta)$ is the likelihood function; $P(D) = \sum_i^n P(D_i)P(D_i|\theta)$ is model evidence; $p(\theta|D)$ is the posterior distribution.

Therefore, we can obtain the posterior distribution of model parameter when we have new observed data based on its known prior and likelihood function. Markov Chain Monte Carlo (MCMC) has become the main computational workhorse in scientific computing for solving this type of statistical inference problems. The core of MCMC algorithm is the calculation of the acceptance rate, which can only be obtained when likelihood has a functional form that can be evaluated. Hence the likelihood needs to be prescribed, which dramatically reduces the freedom to model physical phenomena and thus the overall utility of the uncertainty quantification exercise is diminished. Consider the following general model,

$$d = f(\theta, \epsilon) \tag{1.2}$$

where $f(\cdot)$ is a mathematical model possibly constructed from physics-based principles, θ are the uncertain model parameters with a prior distribution given by $p(\theta)$, and the nuisance parameters ϵ are distributed according to $p(\epsilon)$ and represent the stochasticity in the model due to modeling errors such as missing physics. When observation D is available, MCMC algorithms sample the posterior distribution according to Bayes' theorem,

$$p(\theta|D) \propto L(D,\theta)p(\theta) \ , \tag{1.3}$$

where in general $L(D,\theta)$ is an intractable likelihood function given by the following integral,

$$L(D,\theta) = \int p(D|\theta,\epsilon)p(\epsilon)d\epsilon \ . \tag{1.4}$$

Tractable likelihood functions can only be obtained in special cases when ϵ is not embedded into the mathematical model - such as additive noise, $d = f(\theta) + \epsilon$, or multiplicative noise, $d = f(\theta)\epsilon$. As a result, in general it is challenging to use MCMC algorithms when the likelihood is computational expensive and/or impossible to calculate.

1.2 Approximate Bayesian computation

As discussed above, when the likelihood function is difficult or impossible to evaluate, researchers turned to likelihood-free methods which are called Approximate Bayesian computation (ABC). Approximate Bayesian computation[27] is a family of computational techniques in Bayesian inference that bypass the evaluation of the likelihood function[3, 22, 31]. It has been successfully used in a wide variety of fields such as genetics[30, 15], epidemiology[9, 33], population biology[28, 18, 12], and psychology [39], where the evaluation of the likelihood function is non-trivial. The basic ABC algorithm[34, 27] was introduced as a rejection sampling method, which does not require the evaluation of the likelihood function. As shown in Algorithm 1, samples

Algorithm 1 ABC rejection sampling algorithm

Require: ρ_{threh}, D, $p(\theta)$, $p(\epsilon)$
1: **for** $i = 1$ **to** N **do**
2: **repeat**
3: Sample $(\theta^*, \epsilon^*) \sim p(\theta)p(\epsilon)$
4: Simulate data D^* using the model: $D^* = f(\theta^*, \epsilon^*)$
5: Calculate discrepancy: $\rho(D^*, D)$
6: **until** $\rho(D^*, D) \leq \rho_{thresh}$
7: Accept $\theta_i \leftarrow \theta^*$
8: **end for**
9: **return** N samples $\{\theta_i\}_{i=1...N}$

of θ^* and ϵ^* are used to generate the simulated data D^* using the model in Eq.(1.2). If the distance $\rho(D, D^*)$ between the observed data D and the simulated data D^* is less than a small positive threshold ρ_{thresh}, then the proposed candidate is accepted as a sample drawn from the approximate posterior $p\Big(\theta|\rho(D, f(\theta, \epsilon)) \leq \rho_{thresh}\Big)$, which approaches the posterior distribution $p(\theta|D)$ as ρ_{thresh} tends to zero. A large ρ_{thresh} will yield a higher acceptance rate but a less accurate estimation of the posterior. On the other hand, a small ρ_{thresh} will provide an accurate estimation of the posterior at the cost of a low acceptance rate. As a result, it is not computationally feasible to apply ABC rejection sampling to computational expensive models with non-linear relationship and high dimensional parameter space. In this thesis, we propose to take advantage of this fundamental property of ABC algorithms, namely likelihood-free inference, and have an initial attempt to extend their application to predictive engineering by exploiting the structure of the model via gradient information.

1.3 Motivation

The application of the basic rejection ABC algorithm to complex model generally results in extreme low acceptance rates, which limits the use of ABC. To improve the acceptance rate of the ABC rejection sampling algorithm, a number of advanced sampling ABC variants have been derived such as ABC Markov Chain Monte Carlo

(ABC-MCMC), ABC Partial Rejection Control (ABC-PRC), ABC Population Monte Carlo (ABC-PMC), and ABC Sequential Monte Carlo (ABC-SMC). Marjoram et al.[22] introduced the ABC-MCMC by embedding the Metropolis-Hastings algorithm within the ABC. As with any MCMC method, posterior samples generated by ABC-MCMC are correlated [31], and thus the algorithm is poorly parallelizable [7]. Moreover, convergence is difficult to assess and the samples are prone to get stuck [32]. In response, Sisson et al. [31] developed the ABC-PRC which perturbs the samples using a forward kernel and a backward kernel. However, it has been shown in [2] that ABC-PRC yields a bias in the approximation of the posterior. ABC-PMC was developed by Beaumon et al.[2] to obtain an unbiased estimation[1]. It requires an adaptive Gaussian kernel function with a variance determined by the accepted samples in the previous iteration. The ABC-SMC [38] is derived from a sequential importance sampling algorithm [24] and unlike ABC-PMC the kernel function is non-adaptive and can be non-Gaussian.

In all of the above methods, the speed to which posterior estimates are obtained depends on how well one selects the proposal distributions or transition kernels. A low acceptance rate results in waisted computational effort which is unacceptable especially in the case when one deals with complex models that are computationally intensive. In this article, we propose an importance sampling approach which speeds up the finding of the proposal distribution by exploiting the gradient information. The approach is inspired by the idea of Progressive Correction of Gaussian Components(PCGC)[35, 37], which drives the proposal quickly to the high density regions of the posterior distribution. The method relies on the introduction of an artificial measurement model, which is used to iteratively move to region of interest a set of Gaussian components that will ultimately define the proposal distribution.

The proposed ABC-PCGC has three main steps: (1) an initial Gaussian mixture representation of the proposal distribution, (2) the progressive correction of the

individual Gaussian components, and (3) the construction of the proposal distribution and importance sampling. First, a finite Gaussian mixture model is constructed to span the variability of the prior distribution. The mean and variance of each Gaussian component is then iteratively updated using the linearized artificial measurement model until the effect of the artificial measurement noise on the posterior distribution of the proposal vanishes. The final proposal distribution is given by the mixture of the converged Gaussian components. Since we exploit gradient information and make better use of observations in constructing the proposal distribution, our algorithm explores the parametric space more efficiently using a reduce number of function evaluations, which is currently a major drawback of ABC methods. This allows us to reach acceptance rates that are two orders of magnitude better than the basic ABC rejection sampling. Furthermore, since each Gaussian component is individually updated the algorithm is easily parallelizable by construction.

1.4 Thesis Outlines

In chapter 2, I will review several ABC algorithms proposed in the literature and in chapter 3, the details of the new ABC-PCGC algorithm are presented. In chapter 4, a performance evaluation of the proposed method on two benchmark problems where we assess the acceptance rate and compare it with the ABC rejection sampling. Concluding remarks are then followed by a brief discussion on potential advancements and future work.

Chapter 2

ABC Algorithms

2.1 Introduction

As discussed in Chapter 1, the ABC basic rejection algorithm generates a single sample parameter, θ^*, from the prior distribution $p(\theta)$. Combining the model in Eq.(1.2), θ^* is used to produce the simulated data D^*. When the difference between simulated data D^* and the observed data D is equal to or sufficiently close to 0, then we consider θ^* as a sample from the posterior distribution. Repeat the above process until we have a pool of qualified parameter θ^*, then we could estimate the posterior distribution based on the accepted samples. The main obstacle that prevents ABC basic rejection method to become mainstream is the very low acceptance rate for the following reasons[21]

1. The relationship between prior distribution and posterior distribution determines the acceptance rate (computing efficiency). If the prior resembles the posterior distribution, the acceptance rate is acceptable. Otherwise, the acceptance rate is very low. Therefore, a proper prior is highly needed. As a matter of fact, it is almost impossible to get the proper prior before running the computation.

2. The dimension of parameter space plays an important role of determining the acceptance rate. If the dimension of the parameter space is low, where only few parameters in the model are needed to be estimated, then the acceptance rate is reasonable. However,in the real world, the parameter space is often very

high. It is of low possibility to obtain all "good" parameters that produce the "good" simulated data which matches the true observed data. Therefore, the acceptance rate will be extremely low in practice.

Therefore, it is critical to improve the acceptance rate by introducing new sampling techniques. As briefly discussed in Section 1.2 and 1.3, to improve the unacceptable acceptance rate, a number of advanced sampling methods has been introduced into ABC frame work, such as Markov Chain Monte Carlo(MCMC), Population Monte Carlo (PMC), and Sequential Monte Carlo(SMC). In this chapter, these approaches will be described in turn. The Population Monte Carlo (PMC) is considered as one of the most effective ABC algorithm so far[40].

2.2 ABC and MCMC

Markov Chain Monte Carlo (MCMC) sampling is a fundamental technique in Bayesian estimation[16]. MCMC has been embedded within ABC algorithms by researchers [22] about 20 years ago. MCMC sampling methods also generate the proposal from the prior distribution, and each iteration let proposed samples be tested by a small value of tolerance threshold ϵ and a probability α. Therefore, the proposed samples are approaching high density posterior distribution. The most popular MCMC sampler is Metropolis-Hastings (MH) approach[6]. Unlike the ABC basic rejection approach which obtains independent accepted particles from the posterior distribution, the MCMC-ABC sampler generates a set of dependent accepted samples from the posterior distribution. In subsequent iterations, the proposed parameter θ^* is generated from proposal distribution $q(\theta^*|\theta)$ and then produce the simulated data D^*, when the distance $\rho(D, D^*) \leq \epsilon$ and then accept the proposal with the probability α,

otherwise reject it. The probability α for MH is given by

$$\alpha = \begin{cases} min(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}) & \text{if } \rho(D, D^*) \leq \epsilon \\ 0 & \text{Otherwise} \end{cases} \tag{2.1}$$

As can be seen in the above equation, when the proposal distribution q is symmetric

Algorithm 2 ABC-MCMC

1: Initialize the proposal with θ_0
2: **for** $i = 1$ **to** N **do**
3: perturb $\theta^* \sim q(\theta^*|\theta_{i-1})$
4: Simulate D^* with the model: $D^* = f(\theta^*)$
5: Calculate discrepancy: $\rho(D^*, D)$
6: if $\rho(D^*, D) \leq \epsilon$, accept θ^* with probability: $\alpha = min(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)})$
7: if θ^* is accepted, set $\theta_i = \theta^*$. Otherwise, set $\theta_i = \theta_{i-1}$
8: **end for**

, then the probability α relies only on the prior distribution and $\rho(D, D^*)$. So, it is important to choose a proper prior in order to obtain a good acceptance rate. What is worse, if the proposal distribution $q(\theta^*|\theta_{i-1})$ is poorly chosen, the MCMC algorithm may get stuck[40]. For example, when its variance is relative small, the proposed particle can not be sufficiently perturbed. If a particle is located in a low probability posterior region, it is hard to move out of the area without introducing new information. Therefore the MCMC chains are in high danger of getting stuck. A sufficient small distance threshold, ϵ, is needed in order to filter out most of the "bad" proposals to obtain a accurate estimated posterior. Therefore, the acceptance rate of ABC-MCMC can be extremely low when the chain gets stuck. And also, MCMC chains are challenging to be parallelized[40]. Based on the analysis, we will not consider ABC-MCMC in this thesis.

2.3 ABC and Particle Filtering

Since simple MCMC sampler suffers from trapping candidates and have low acceptance rate, we move to discuss the Sequential Monte Carlo (SMC) methods[13]. Un-

like the MCMC sampler which draws a candidate θ^* one at a time, the SMC works with a pool of candidates, called particles, simultaneously by using particle filter. In each iteration, the particle is perturbed and filtered, bringing the particles closer and closer to the high density posterior region. At the begin of the SMC sampler, it initializes a pool of N candidates with equal weight from the prior distribution. Then in subsequent iterations, particles are chosen from the previous pool based on its performance (weight). A transition kernel (backward and forward) is needed for perturbing the particles. In practice, the forward and backward transition kernel are often symmetric or equal, which is designed to simplify the algorithm[31], but sometimes it is a poor choice[23, 38]. A good kernel not only maintains the original information of the previous particle but also introduce some new information to the particle, which can avoid getting stuck . Normally, a Gaussian with 0 mean and σ^2 variance is a good choice. For example, consider $\xi \sim \mathcal{N}(0, \sigma^2)$ as the kernel, the new candidate ca be obtained using $\theta^* = \theta + \xi$.

Different SMC samplers can be derived by how sampling weight are assigned and what kernel is specified. Then the subsequent section will describe three main SMC samplers: Particle Rejection Control(PRC), Population Monte Carlo(PMC), and Sequential Monte Carlo(SMC).

Particle Rejection Control sampler

The ABC Partial Rejection Control (ABC-PRC) algorithm was developed by Sisson et al.[32] as a remedy for the problems associated with ABC-MCMC discussed in the previous section. The PRC algorithm was the first particle filter algorithm embedded into ABC framework. Both a forward and backward transition kernel is needed to be specified in advance. We denote the forward kernel as a density function $q_f(.|\theta)$ and the backward kernel as $q_b(.|\theta^*)$. We use the forward transition kernel $q_f(.|\theta)$ to perturb the original particle θ to generate new particle θ^* , and then, with θ^* ,

we generate simulated data D^* based on its model and compare D^* to the observed data D by computing the distance $\rho(D, D^*)$. If the particle θ^* satisfies inspection (when the distance $\rho(D, D^*)$ is less than some sufficient small value threshold ϵ), then consider the particle as a sample from the approximate posterior domain and calculate the particle a weight which is the probability of being sampled on the subsequent iteration.If the particle can not pass inspection (if $\rho(D, D^*)$ is greater than the sufficient small value of threshold ϵ), the particle is discarded, and repeat generating new particles θ^* from θ until we obtain a particle that does pass inspection. The weight w given to the new particle $\theta^*$is

$$w = \frac{p(\theta^*)q_b(\theta|\theta^*)}{p(\theta)q_f(\theta^*|\theta)} \tag{2.2}$$

This above process is repeated until there consists of N new particles, which satisfy the requirement that $\rho(D, D^*) \leq \epsilon$. Up to now, the ABC-PRC is equivalent to the ABC basic rejection sampler (Algorithm 1) except every particle has a weight. The main difference is to repeat the above process multiple times. On consequent iterations we sample particles with probabilities based on their performance evaluated in the previous iteration. We consider the weighed particles as proposal. These weights are giving us a chance to reduce the influence of particles from the pool in low-density regions and increase the number of particles in high-density regions, which will make the best use of the particles. As a result, it will generate a pool of particle that represents a sample from the desired estimate of the posterior $p(\theta|\rho(D, D^*) \leq= \epsilon)$, with a higher efficiency compared to basic ABC rejection algorithm. Secondly, the weighting scheme reduce the probability of a chain getting stuck in a low probability region.

However, there are two obstacle that prevent the PRC to become mainstream. Firstly, the acceptance rate (efficiency) of the sampler relies heavily on the choices of the two kernels $q_f(\theta^*|\theta)$ and $q_b(\theta|\theta^*)$ and the prior $p(\theta)$. It is hard to choose a optimal

transition kernel[23] and proper prior distribution before computation. Secondly, the ABC-PRC produces biased estimates of the posterior [2] : the distribution estimated based on the pool of accepted weighted particles does not converge to the true posterior. Beaumont corrected for this bias using a population Monte Carlo sampling scheme.

Population Monte Carlo sampler

As shown in Algorithm 3, the Population Monte Carlo algorithm was first embedded into ABC by Cappé et al., in 2004[11]. Unlike ABC-PRC, ABC Population Monte Carlo (ABC-PMC) sampling employs a different scheme of calculating weights. In addition, the ABC-PMC algorithm specifies a single adaptive transition kernel $q(.|\theta)$ that depends on the variance of the accepted particles in the previous iteration, while both forward and backward transition kernels are required by the ABC-PRC sampler. Specifically, given the weight $w_{i,t-1}$ for particle $\theta_{i,t-1}$ on iteration t-1, the new weight $w_{i,t}$ for particle$\theta_{i,t}$ on iteration t is calculated as

$$w_{i,t} = \frac{p(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q(\theta_{j,t-1}|\theta_{i,t}, \sigma_{t-1})} \tag{2.3}$$

The variance σ_t^2 is computed as

$$\sigma_t^2 = \frac{2}{N} \sum_{i=1}^{N} (\theta_{i,t} - \sum_{j=1}^{N} \theta_{j,t}/N)^2 = 2Var(\theta_{1:N,t}) \tag{2.4}$$

One common problem with many samplers is how fast we can reach the estimated high density posterior distribution. This speed is determined by the particle acceptance rate. Poorly selected prior distributions or transition kernels lead to very low acceptance rates, which further results in a tremendous amount of computation wasted because it evaluates great amount of particles that have low chance of being selected.

Firstly, the greatest strength of the ABC-PMC is that it optimizes the acceptance probability by its weighting approach. This reason is that its weights minimize

the Kullback-Leibler (K-L) distance (also information divergence, information gain, relative entropy) between the target posterior and the proposal distribution. The K-L distance is a measurement that indicates the difference between two probability distribution.The acceptance is maximized by minimizing the K-L distance [2].

Algorithm 3 ABC-PMC

Require: Given Y and assume Y $\sim$ Model(θ), tolerance threshold θ ,and prior distribution $p(\theta)$

Iteration t = 1
for $i = 1$ **to** N **do**
  repeat
    Sample $(\theta^*) \sim q(\theta)$
    Simulate data D^* using the model: $D^* = f(\theta^*)$
    Calculate discrepancy: $\rho(D^*, D)$
  until $\rho(D^*, D) \leq \rho_{thresh}$
  $\theta_{i,1} \leftarrow \theta^*$
  $\omega_{i,1} \leftarrow \frac{1}{N}$
end for
$\sigma_1^2 \leftarrow 2 * Var(\theta_{1:N,1})$
for $t = 2$ **to** T **do**
  for $i = 1$ **to** N **do**
    repeat
      Sample (θ^*) from previous iteration : $(\theta^*) \sim \theta_{1:N}$ with wight $\omega_{i,1}$
      Perturbe θ^* : $(\theta^*) \sim \mathcal{N}(\theta^*, \sigma_{t-1}^2)$
      Simulate data D^* from θ^{**} : $X \sim Model(\theta^{**})$ Calculate discrepancy: $\rho(D^*, D)$
    until $\rho(D^*, D) \leq \rho_{thresh}$
    $\theta_{i,1} \leftarrow \theta^*$
    $\omega_{i,1} \leftarrow \frac{1}{N}$(this is weight is un-proper)
  end for
  $\sigma_1^2 \leftarrow 2 * Var(\theta_{1:N,1})$
end for

Based on above discussion, it is obvious that the ABC-basic rejection, MCMC, PRC sampler, they all rely on satisfying the condition $\rho(D, D^*) \leq \epsilon$. Therefore, the accuracy of the estimated posterior distribution and the acceptance rate (computation efficiency) is mostly determined by the selection of tolerance threshold ϵ: large value ϵ results into inaccurate posterior estimation while small value of ϵ leads to low acceptance rate. It is a trade-off between the acceptance rate and the accuracy of the

estimated posterior distribution.

Secondly, another strength of PMC samplers is that it does not rely on a single tolerance threshold ϵ and then reduce the affect of selection of ϵ. Instead it selects a series of ϵ values which decrease monotonically from a larger value to some smaller one. This engage the samples to converge slowly to a smaller feasible final value of ϵ. Hence, it highly improves the acceptance rate.

Adaptive Population Monte Carlo sampler

As discussed in the above section, the PMC does not rely on a single tolerance threshold ϵ by using a set of monotonically decreasing ϵ. It allows the samples to gradually moves to the high density posterior distribution. It is still challenging to determine how fast should the decreasing of the ϵ be. The selection of proper sequence of tolerance levels and a proper stopping criterion are becoming critical problems hindering the ABC-PMC's application, particularly in complex models[21].

1. A fast decreasing sequence of tolerance results into a fast convergence but a relatively low acceptance rate in each iteration, while a slow decreasing sequence of tolerance leads to a reasonable acceptance rate but an relatively low speed of convergence. It is crucial to weigh the pros and cons of the sequence of tolerance level. Therefore, preliminary simulation is required to make the decision, which is impractical or expensive in complex model computation.

2. Although in ABC-PMC, we do not rely on a single tolerance threshold, we approach the final threshold by degree. If the computation reaches convergence before the final threshold, it may put us at risk of repeating the rest unnecessarily. So, determining a proper stopping criterion to control the process of computation is of utmost importance. Stopping too early can decrease the accuracy of the estimation, while stopping too late will lead to a large number

of unnecessary computation time. Preliminary analysis is needed to determine the stop criterion.

To solve the shortcomings of the ABC-PMC algorithm, Lenormand et al. proposed a self-adaptive PMC algorithm[21]. The adaptive algorithm differs from the ABC-PMC algorithm of the original in four ways.

1. First, the weighted particles are divided into two portions, $1 \sim N_\alpha$ and $N_\alpha+1 \sim N$ based on its weight. In each iteration, we only update the second portion to make it satisfy the tolerance. Therefore, it not only makes the best use of the costly simulation, but also keeps the diversity of the particles.

2. Second, the tolerance values are given by the algorithm. It is the distance of α-quantile particles in the previous step. It is adaptive to choose a proper threshold based on its previous proposal.

3. Last, the stopping criterion is given by evaluating whether the particles pool has sufficiently changed during the current step. If there is no sufficient change in the current particle pool, it is considered that the particles have converged. Otherwise, we will go to the next iteration to modify the particles.

Sequential Monte Carlo sampler

The Sequential Monte Carlo (SMC) algorithms begin by generating a pool of N particles by sampling from the prior distribution $p(\theta)$ and initially have a equal weight. In the subsequent iterations, particles are chosen randomly from this pool according to particle weight. In each iteration, a transition kernel is required to modify and move the particles around in the parameter space. And its weight is computed as

$$w_{i,t} = \frac{p(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q(\theta_{j,t-1}|\theta_{i,t})} \tag{2.5}$$

Algorithm 4 Adaptive ABC-PMC

Require: Given α the proportion of particles to keep at each iteration and $p_{acc_{min}}$ the minimal acceptance rate,

Initialization:
Set $N_\alpha = |\alpha N|$
for $i = 1$ **to** N **do**
  Simulate $(\theta_i^0) \sim p(\theta)$
  Simulate data D^* using the model: $D^* = f(\theta_i^0)$
  Calculate discrepancy: $\rho_i^0(D^*, D)$
  Set $\omega_i^0 = 1$;
end for
Let $\epsilon_1 = Q_{\rho^0}(\alpha)$ be the first α-quantile of $\rho^{(0)}$.
Let $\{(\theta_i^1, \omega_i^1, \rho_i^1)\} = \{(\theta_i^0, \omega_i^0, \rho_i^0) | \rho_i^0 \leq \epsilon_1, 1 \leq i \leq N\}$
Calculate $\sigma_1^2 \leftarrow 2 * Var(\theta_{1:N_\alpha,1})$
Set p_{acc}=1
Set t =2
Iteration t = 1
repeat
  for $i = N_\alpha + 1$ **to** N **do**
    Sample $(\theta_i^*) \sim \theta_j^{t-1}$ with weight ω_j^{t-1}
    Generate $(\theta_i^{t-1} | \theta^*) \sim \mathcal{N}(theta_i^*, \sigma_{t-1}^2)$
    Simulate data D^* using the model: $D^* = f(\theta_i^{t-1})$
    Calculate discrepancy: $\rho_i^{t-1}(D^*, D)$
    $\omega_i^{t-1} \leftarrow \frac{1}{N}$(this is weight is un-proper)
  end for
  $p_{acc} = \frac{1}{N - N_\alpha} \sum_{k=N_\alpha+1}^{N} |\rho_i^{t-1}(D^*, D) \leq \epsilon_{t-1}$
  $\epsilon_t = Q_\rho^{t-1}(\alpha)$
  Let $\{(\theta_i^t, \omega_i^t, \rho_i^t)\} = \{(\theta_i^{t-1}, \omega_i^{t-1}, \rho_i^{t-1}) | \rho_i^{t-1} \leq \epsilon_1, 1 \leq i \leq N\}$
  Calculate $\sigma_t^2 \leftarrow 2 * Var(\theta_{1:N_\alpha,1})$
  Set t =t+1
until $p_{acc} > p_{acc_{min}}$

The kernel $q(.|\theta)$ is not necessary adaptive and Gaussian distribution, which differs from the classical PMC. Therefore, the ABC-SMC method is particularly powerful when the transition kernel in PMC does not have to be Gaussian.

2.4 Discussion

The Basic rejection algorithm hinges on the tolerance threshold ϵ, large values of ϵ results into inaccurate posterior estimation while small values of ϵ need long computation times due to the high rejection rate. In practice, the basic rejection algorithm will never be used. With the development of more complex algorithms, many advanced techniques, such as Markov Chain Monte Carlo, population Monte Carlo, have been introduced into ABC. However, this kind of techniques still have to handle how to find an optimal values for the tolerance threshold. In order to overcome the the trade off caused by the values of ϵ, a set of ϵ that decrease monotonically from a relative larger tolerance value to some small values, which is originated from simulated annealing. These algorithms were allowed to converge slowly on the posterior distribution. There are two drawbacks that prevent PMC to become mainstream.

1. It still relies on the ϵ at each iteration. These algorithm still have to throw away huge amount of population that cannot satisfy the tolerance threshold. Many samples were wasted.

2. As we know, fast speed of decreasing the ϵ will have less iterations and the particles are hard to pass the inspection while low speed can make the proposal easier to pass the inspection but it needs more iterations. So, we do not know the optimal speed before we run the simulation.

All in all, the PMC methods makes a big improvement to the computation efficiency. However, there are some drawbacks in this framework that limit the further development in the ABC-PMC, such as the optimal T. Then we start looking for

new ABC algorithm in other domain. The next chapter will discuss our new ABC algorithm.

Chapter 3

ABC-Progressice Correction of Gaussian Components algorithm

3.1 Introduction

ABC approaches can estimate the posterior distribution without the evaluation of the likelihood function. Therefore ABC has rapidly gained popularity over the last years, especially for the complex problems in biology [14, 26, 4, 42], ecology [19], population genetics [41, 8] and physics [10] where mostly likelihood is intractable. Unfortunately, it is well known that complex problems will always come with high-dimensional parameter spaces. When the number of parameter is increased the acceptance rate declines dramatically because it is required an huge amount of samples to be simulated in order to find a set of parameter values that satisfy the tolerance threshold, especially when we prefer to obtaining an acceptable level of accuracy for the posterior estimation. In the worst case, the acceptance rate is low such that the computation cost is beyond control. This is the well-known problem of "curse of dimensionality" [1, 5].

In the thesis, I present a new ABC algorithm, called ABC-PCGC, which utilize the idea of Progressive Correction of Gaussian Components as a means of proposal generation. PCGC [37], which has similarity with simulated annealing and progressive correction used in particle filters [25], is proposed to gradually correct a set of initial Gaussian components to cover the support of posterior distribution. Extended Kalman Filter (EKF), is a non-linear Kalman Filter, is the force to correct the Gaus-

sian components. EKF is the most widely used estimation algorithm for nonlinear system parameter estimations [20]. EKF is an extremely powerful algorithm that updates the Gaussian components based on measurement. EKF uses gradient information to guide the direction of correction, which can great improve the proposal update efficiency. In practice, because the EKF updates the Gaussian components recursively, no additional past measurement is required. Here, I merge the ABC, PCGC and EKF to improve the ABC computational efficiency. I will describe the PCGC, EKF and our algorithm in the following sections.

3.2 Progressive Correction of Gaussian Components

Terejanu et al. [35] proposed a Gaussian mixture methods to approximate the evolution of a probability density function (pdf) through a nonlinear function with a reasonable computation cost. A finite sum of Gaussian density functions is used to approximate the initial state pdf and then the mean and variance are propagated by linear theory. Each Gaussian density function has a weight which corresponds to the Gaussian kernel's importance in the whole mixture. The method has two advantages

1. The Gaussian mixture can be solved efficiently and accurately using convex optimization solvers, even if the mixture model includes many terms.

2. It decouples a complex pdf into sum of Gaussian density functions. As a consequence, it can be easily parallelized.

The Gaussian mixture model has been applied to low and moderate nonlinear systems such as uncertainty propagation through two-body system and toxic cloud transported by wind [17, 35, 36].

In this article, we use a Gaussian mixture approximation to the prior pdf and propose a "non-intrusive" way of computing a proposal for the posterior pdf. A progressive correction is designed to update the Gaussian components such that they

will reach the high density posterior region. We name this algorithm "Progressive Correction of Gaussian Components" (PCGC) and in the coming section, we apply the PCGC to ABC algorithms, which is called the ABC-PCGC algorithm.

3.3 ABC-Progressive Correction of Gaussian Components Algorithm

The overall structure of the proposed ABC-PCGC algorithm is given by the importance sampling. Given a proposal distribution over both parametric and nuisance parametric space, $q(\theta, \epsilon)$, weighted samples from the approximate posterior distribution are obtained using Algorithm 5, which describes the general form of the ABC-PCGC.

Algorithm 5 ABC-PCGC general form

Require: ρ_{threh}, D
1: Obtain proposal $q(\theta, \epsilon)$ using Algorithm 6
2: **for** $i = 1$ **to** N **do**
3: **repeat**
4: Sample $(\theta^*, \epsilon^*) \sim q(\theta, \epsilon)$
5: Simulate data D^* using the model: $D^* = f(\theta^*, \epsilon^*)$
6: Calculate discrepancy: $\rho(D^*, D)$
7: **until** $\rho(D^*, D) \leq \rho_{thresh}$
8: Accept $\theta_i \leftarrow \theta^*$
9: Calculate weight $\omega_i = \frac{p(\theta^*)p(\epsilon^*)}{q(\theta^*, \epsilon^*)}$
10: **end for**
11: Normalize weights $\omega_i^N = \frac{\omega_i}{\sum_{i=1}^{N} \omega_i}$
12: **return** N weighted samples $\{\omega_i^N, \theta_i\}_{i=1...N}$

In the followings, since we are looking for a joint proposal distribution for simplicity we denote with x the vector with components θ and ϵ,

$$x = [\theta \ \epsilon]^T \ . \tag{3.1}$$

The basis of the proposed algorithm is the construction of a joint proposal distribution using the following artificial measurement model,

$$z = \rho(D, f(x)) + w \ , \tag{3.2}$$

where w is an artificial additive measurement noise, which is normally distributed with zero mean and σ_w^2 variance. Since we are after minimizing the distance $\rho(\cdot)$, the sought proposal distribution $q(x|Z)$ can be obtain using Bayes' theorem given the artificial measurement $Z = 0$,

$$q(x|Z) \propto L(Z,x)q(x) \ , \text{ where} \tag{3.3}$$

$$L(Z,x) = \mathcal{N}(Z; \rho(D, f(x)), \sigma_w^2) \ . \tag{3.4}$$

Here, the initial proposal $q(x) \propto p(\theta)p(\epsilon)$ and $\mathcal{N}(\cdot)$ denotes a Gaussian density function. The posterior proposal $q(x|Z)$ will ensure that its marginal $q(\theta|Z)$ is positioned in the high density region of the posterior $p\Big(\theta|D)$. The posterior proposal overestimate the support of the posterior distribution due to the variance of the artificial noise $\sigma_w^2 > 0$. Ideally we would like to have $\sigma_w^2 = 0$, but this will make the newly posed statistical inverse problem also intractable. Compared with sampling from the prior distribution, sampling from this proposal will considerably improve the acceptance rate of the proposed algorithm. Furthermore, the method is designed to obtain this proposal with a reduced number of function evaluations.

So far, we have converted a statistical inverse problem with intractable likelihood into an importance sampling problem where we need to solve another statistical inverse problem to obtain the posterior proposal. The new statistical inverse problem has a tractable likelihood. Nonetheless for small measurement variance σ_w^2 these type of problems are numerically challenging as the likelihood function may not be in the support of the prior proposal. We thus adopt an iterative approach based on covariance inflation to attain an overlap between likelihood and prior and at a same time to minimize the effect of the artificial measurement noise on the posterior proposal. We start out with a larger variance $\sigma_{w,large}^2$ and decompose the likelihood into K factors where the variance of each intermediate likelihood decreases with a

factor $\beta \geq 1$,

$$L(Z,x) \propto L_K(Z,x) \times \ldots \times L_2(Z,x) \times L_1(Z,x) \tag{3.5}$$

$$L_k(Z,x) = \mathcal{N}\left(Z; \rho(D, f(x)), \frac{\sigma^2_{w,large}}{\beta^{k-1}}\right) . \tag{3.6}$$

The final proposal distribution, $q(x|Z) = q_K(x|Z)$, is obtained iteratively by solving K inverse problems,

$$q_k(x|Z) \propto L_k(Z,x) q_{k-1}(x|Z) \ , \tag{3.7}$$

where $q_0(x,Z) = q(x)$ and $k = 1 \ldots K$. Note that for $\beta = 1$, $L(Z,x) \propto \mathcal{N}\Big(Z; \rho(D, f(x)), \frac{\sigma^2_{w,large}}{K}\Big)$. Thus, each additional iteration diminishes the effect of the artificial measurement noise on the posterior proposal. In the limit, as $K \to \infty$ the likelihood function approaches a Dirac delta function.

The procedure to obtain the prior proposal as well as approximating the posterior proposal is described in the following two sections.

Proposal Initialization

The key idea of constructing the prior proposal is to represent it as a finite sum of Gaussian density functions such that the mixture mean and covariance matrix matches the mean and the covariance matrix of the prior distribution. Any other decomposition scheme can be used to construct the above Gaussian mixture as long as it matches the support of the prior distribution. We begin by selecting a pool of M Gaussian components that will define the prior proposal $q(x)$ as follows,

$$q(x) = \frac{1}{M} \sum_{i=1}^{M} \mathcal{N}(x; \mu_i, \gamma\Sigma) \ . \tag{3.8}$$

Given that in this study we use Gaussian distributions as prior distributions, we adopt the splitting scheme proposed in [43] to approximate the prior Gaussian probability density function (pdf), $\mathcal{N}(x; \mu, \Sigma)$, with a mixture of two Gaussian components, $\mathcal{N}(x; \mu_1, \Sigma_1), \mathcal{N}(x; \mu_2, \Sigma_2)$. This is accomplished by splitting the prior pdf along the

jth eigen-direction as follows.

$$\mu_1 = \mu - \frac{1}{2}\sqrt{\lambda_j}V_j \tag{3.9}$$

$$\mu_2 = \mu + \frac{1}{2}\sqrt{\lambda_j}V_j \tag{3.10}$$

$$\Sigma_1 = \Sigma_2 = \Sigma - \frac{1}{4}\lambda_j V_j V_j^T \tag{3.11}$$

Here, λ_j and V_j are the jth eigenvalue and eigenvector of $\Sigma = VDV^T$ with $V = [V_1 \ \cdots \ V_n]$ and $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$. To obtain the proposal Gaussian mixture, we split along all n eigen-directions, which results in $M = 2n$ Gaussian components. One may also choose to further refine the Gaussian mixture by continue splitting each Gaussian component. This increases the number of Gaussian components exponentially if performed along all eigen-directions. Once we have the initial M Gaussian components, we gradually update their mean and covariance as described in the next section.

Progressive Correction

We are after a proposal distribution that can capture the support of the actual posterior distribution. Given the Gaussian mixture representation of the prior proposal, our goal is to find an approximate posterior proposal that also has a Gaussian mixture representation.

$$q(x|Z) \approx \frac{1}{M}\sum_{i=1}^{M}\mathcal{N}(x; \mu_i^K, \Sigma_i^K) \tag{3.12}$$

In this section, we propose to iteratively update the means and covariances of the initial Gaussian components. We take advantage of the gradient information of the distance function in Eq.(3.2), to obtain the approximate means and covariances of the Gaussian components that approximate the intermediate posteriors $q_k(x|Z)$ in Eq.(3.7). This update resembles the measurement update step in the extended

Kalman filter.

$$\mu_i^k = \mu_i^{k-1} + G_i^k[Z - f(\mu_i^{k-1})] \tag{3.13}$$

$$\Sigma_i^k = \Sigma_i^{k-1} - G_i^k P_i^{k-1} \Sigma_i^{k-1} \tag{3.14}$$

$$P_i^{k-1} = \left.\frac{\partial \rho(D, f(x))}{\partial x}\right|_{x=\mu_i^{k-1}} \tag{3.15}$$

$$G_i^k = \Sigma_i^{k-1}(P_i^{k-1})^T \left[P_i^{k-1}\Sigma_i^{k-1}(P_i^{k-1})^T + \frac{\sigma_{w,large}^2}{\beta^{k-1}}\right]^{-1} \tag{3.16}$$

Since we only have to provide an approximation of the posterior proposal we relax the condition to update also the weights of the Gaussian components. Since the means and covariances of the Gaussian components define the support of the posterior proposal, we choose not to change the weights in the mixture. By keeping them constant, we can independently update the individual Gaussian components resulting in an easily paralellizable algorithm.

The final set of Gaussian components, obtained by iterating the progressive correction, is concentrated on the high density posterior distribution. Their equally weighted mixture gives the posterior proposal in Eq.(3.12) which is used for importance sampling as described in Algorithm 5.

Note that the PCGC algorithm relies on four parameters, namely $\sigma_{w,large}^2$, β, M, and Σ, which determine its performance. Since the scope of the current paper is just to introduce the proposed algorithm and have initial feasibility studies, future work is planned on characterizing the optimal values of the PCGC parameters.

The PCGC step is summarized in Algorithm 6.

3.4 Conclusion

In this chapter, we describe how ABC-PCGC works. In general, for this algorithm, we have two steps

Algorithm 6 Progressive Correction of Gaussian Components

Require: $p(x) = \pi(\theta)\pi(\epsilon)$ - prior distribution
M - number of Gaussian components
D - observed data
β - speed factor
$\sigma^2_{w,large}$ - initial artificial noise variance
K - maximum number of correction

Proposal initialization: e.g. Eqs.(3.9)-(3.11)
$q(x) = \frac{1}{M}\sum_{i=1}^{M}\mathcal{N}(x;\mu_i^0,\Sigma_i^0)$

Update the Gaussian components:
for $i = 1$ **to** M **do**
  for $k = 1$ **to** K **do**
    Jacobian matrix:
    $P_i^{k-1} = \frac{\partial\rho(D,f(x))}{\partial x}\Big|_{x=\mu_i^{k-1}}$
    Intermediate noise variance:
    $R_k = \frac{\sigma^2_{w,large}}{\beta^{k-1}}$
    Kalman gain:
    $G_i^k = \Sigma_i^{k-1}(P_i^{k-1})^T\left[P_i^{k-1}\Sigma_i^{k-1}(P_i^{k-1})^T + R_k\right]^{-1}$
    Update the mean (noting that $Z = 0$):
    $\mu_i^k = \mu_i^{k-1} - G_i^k f(\mu_i^{k-1})$
    Update the covariance:
    $\Sigma_i^k = \Sigma_i^{k-1} - G_i^k P_i^{k-1}\Sigma_i^{k-1}$
  end for
end for

return proposal $q(x|Z) \approx \frac{1}{M}\sum_{i=1}^{M}\mathcal{N}(x;\mu_i^K,\Sigma_i^K)$

1. PCGC step-generating the proposal: We approximate the prior distribution by a finite sum of Gaussian components. Then progressively update its mean, variance to cover the support of the posterior distribution. The update process is driven by the Extended Kalman Filter, which uses the gradient information to guide the direction of particle movement. This iterative procedures will make the Gaussian components reach an high density posterior distribution. We call the updated Gaussian components as proposal, which will be used in the next step.

2. ABC step-approximating the posterior: After we have the proposal, we could put any other ABC algorithms on the top of this proposal. For previous ABC algorithm, they start the approximation from the initial prior distribution. Here, the same ABC algorithms start simulating the posterior based on the proposal generated in the above step, which will greatly improve the computing efficiency and bear the same accuracy.

In the next chapter, we will evaluate the performance of the proposed ABC-PCGC algorithm by some examples.

Chapter 4

Simulation

In the followings we compare the acceptance rate of ABC-PCGC described in Algorithm 5, against the acceptance rate of the basic ABC rejection sampling, see Algorithm 1. The evaluation is carried on two benchmark problems and simple application: (1) a simple linear function for which we can analytically compute the posterior distribution, (2) 1-dimensional toxin wind transportation problem, and (3) 2-dimensional toxin wind transportation problem.

4.1 Linear model

Consider the following linear model with additive noise:

$$d = f(\theta, \epsilon) = \theta + \epsilon \tag{4.1}$$

We observe data $d = 4$ and the prior distribution of $x = [\theta\ \epsilon]^T$ is given as follows:

$$p(x) = \mathcal{N}\left(x; \begin{bmatrix} 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{4.2}$$

We are starting out with $\sigma^2_{w,large} = 100$ and the distance function is given by

$$\rho(d^*, d) = (d^* - d)^2 \tag{4.3}$$

The initial variance is set to $\sigma^2_{w,large} = K\rho_{thresh}$ and the distance function is given by $\rho(d^*, d) = (d^* - d)^2$. We initialize 4 Gaussian components by splitting the prior distribution according to Eqs.(3.9)-(3.11). We then progressively update the Gaussian components using the gradient information of the distance function. As we can see

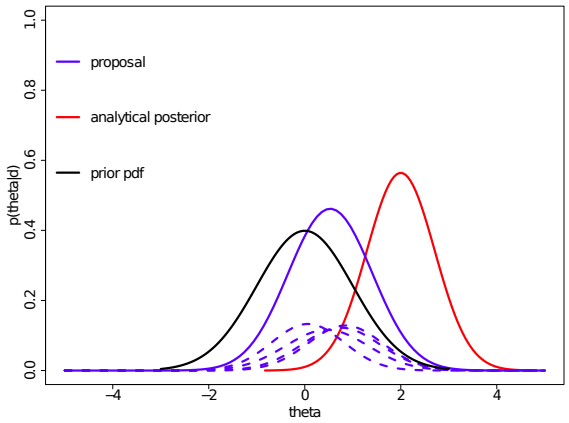

Figure 4.1: Linear example: ABC-PCGC with $\rho_{thresh} = 0.01$ and $M = 4$, k=1

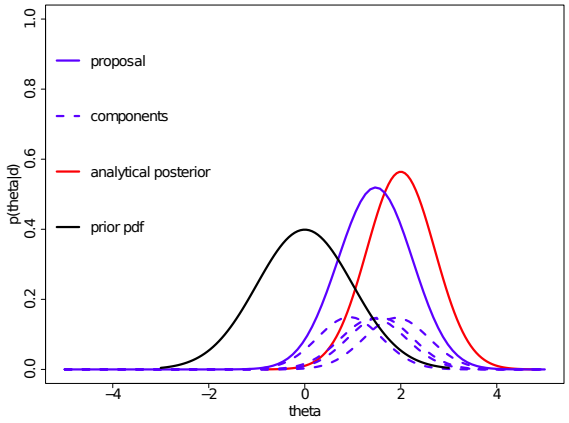

Figure 4.2: Linear example: ABC-PCGC with $\rho_{thresh} = 0.01$ and $M = 4$, k=100

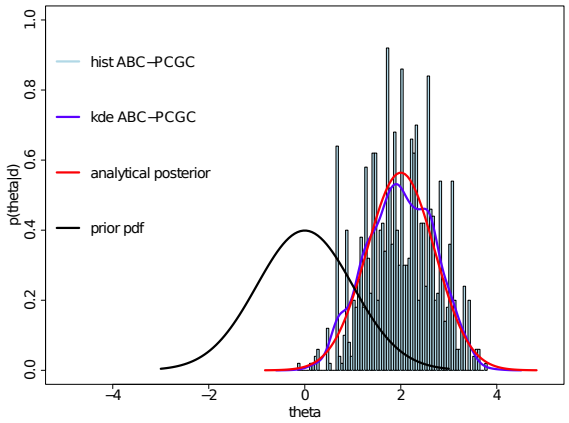

Figure 4.3: Linear example: ABC-PCGC with $\rho_{thresh} = 0.01$ and $M = 4$, k=1000

in Fig. 4.1, 4.2, 4.7, the proposal is gradually moved from the prior distribution to the high density region of the posterior distribution. For this example we have used a variance decreasing factor $\beta = 1$. The final posterior distribution of the proposal is given by the mixture of the converged Gaussian components and used to generate candidate samples as described in ABC-PCGC algorithm. We then inspect this population based on $\rho(D^*, D) < \rho_{thresh}$ as described in Algorithm 5.

We also use ABC rejection algorithm, by directly sampling from the prior distribution and inspect the samples by using the same threshold. Compared with the basic ABC rejection algorithm which directly sample from the prior, the acceptance rate of the population sampled from the proposal generated by the PCGC is increased by an order of magnitude - as shown in Table 4.1.

Table 4.1: Linear model - Acceptance Rate

ρ_{thresh}	1	0.1	0.01
ABC-Rejection	0.016	0.003	0.001
ABC-PCGC	0.296	0.051	0.006
K (ABC-PCGC)	100	1000	10000

4.2 1D Puff-based dispersion model

In this section, we apply the ABC-PCGC algorithm on a 1D puff-based dispersion model, whose bimodal posterior cannot be analytically specified. Given $c(t_k, x_s)$, the concentration reading at time t_k of a sensor located at x_s, the goal is to obtain the posterior distribution of the release location x_0 in the presence of an uncertain wind vector w.

$$c(t_k, x_s) = \frac{m}{\sqrt{2\pi p^2 (t_k w)^{2q}}} \exp\left(-\frac{1}{2} \frac{(x_0 + t_k w - x_s)^2}{p^2 (t_k w)^{2q}} \right) \tag{4.4}$$

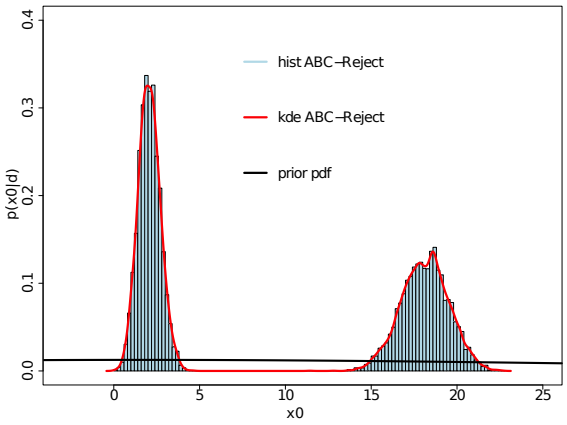

Figure 4.4: 1d Puff Source location: ABC-Rejection with $\rho_{thresh} = 1$

Here, $p = 0.8$, $q = 0.8$ are Karlsruhe-Jülich diffusion coefficient which are determined by the weather conditions [29]. The mass release is given by $m = 1000g$ and the distribution of the wind speed is assumed normal with a mean of $10m/s$ and a standard deviation of $1m/s$. The wind blows from right to left and the true release location is assumed to be at $2m$. The wind speed used to generate the synthetic data is set to $10m/s$ and the measurement time is given by $t_k = 1s$. Given the concentration of the sensor located at $x_s = 0m$ and the wind distribution, we apply ABC-PCGC to infer the original source location. The prior distribution of source location and the initial artificial noise variance is given as follows:

$$p(x_0) \sim \mathcal{N}\Big(x_0; 0, 1000\Big) \tag{4.5}$$

The initial variance is set to $\sigma^2_{w,large} = K\rho_{thresh}$ and the distance function is given by $\rho(c, c^*) = (c - c^*)^2$ where c^* is the simulated concentration. As in the previous example $\beta = 1$.

First, ABC-basic rejection algorithm is applied to obtain samples from the posterior distribution, see Fig. 4.4. The estimated posterior is a bimodal distribution and has asymmetric modes due to the wind uncertainty that appears in the denominator terms in Eq. (4.4).

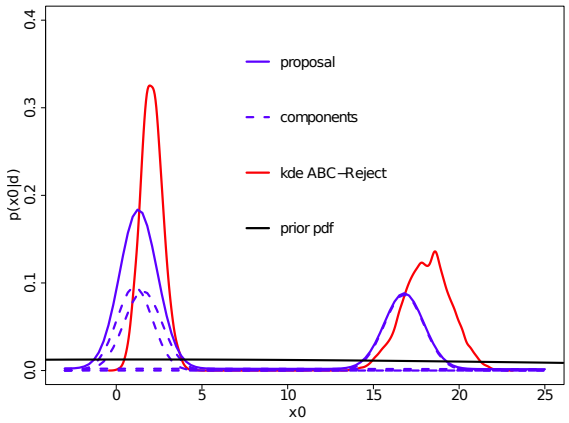

Figure 4.5: 1d Puff Source location: ABC-PCGC with $\rho_{thresh} = 1, k = 1$

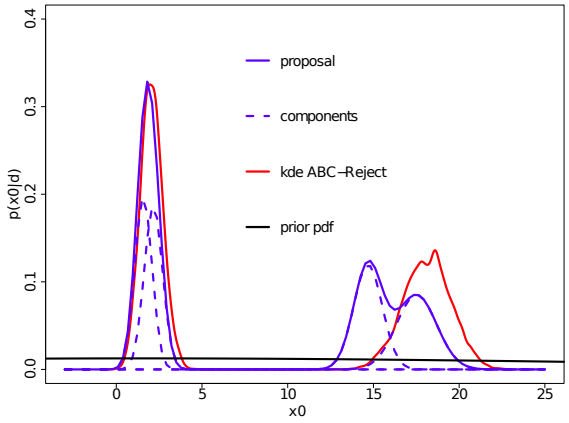

Figure 4.6: 1d Puff Source location: ABC-PCGC with $\rho_{thresh} = 1$ k=100

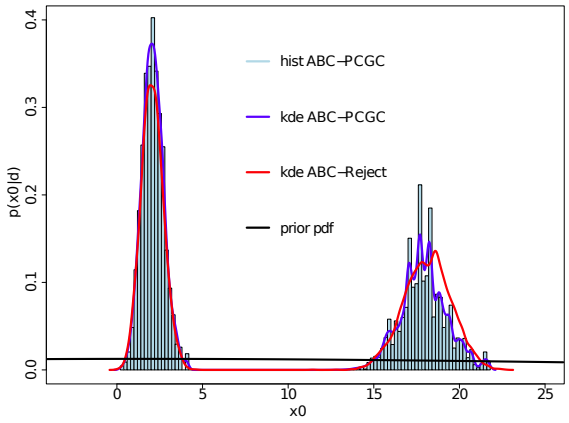

Figure 4.7: 1d Puff Source location: ABC-PCGC with $\rho_{thresh} = 1$, k=1000

Next, we apply the ABC-PCGC algorithm to update the initial prior distribution. Based on the same splitting scheme given in Section 3.3 we obtain 4 Gaussian components to approximate the initial prior distribution of source location. As can be seen from Fig. 4.5, 4.6 the proposal is moving to the target posterior region and the final samples from the posterior are presented in Fig. 4.7. We repeat the simulation with different threshold values and compare the acceptance rate between the ABC-basic rejection and ABC Basic rejection with PCGC, see Table 4.3.

Table 4.2: 1d Puff model - Acceptance Rate

ρ_{thresh}	100	10	1
ABC-Rejection	0.06	0.02	0.006
ABC-PCGC	0.70	0.46	0.1
K (ABC-PCGC)	100	1000	10000

The ABC-PCGC has one to two orders of magnitude higher acceptance rate than acceptance rate of the ABC basic rejection algorithm.

4.3 2D Puff-based dispersion model

In this section, we extend the 1D puff-based model to 2D model. The concentration $c(t_k, x_s)$ is given in equation 4.4. The only difference is that we have 2-dimensional space for the puff location and wind condition. The rest parameter will keep the same. $p = 0.8$, $q = 0.8$. The mass release is given by $m = 1000g$, the distribution of the wind speed is assumed normal with a mean of 2m/s and a standard deviation of 0.1m/s, and the wind distribution of wind direction is assumed normal with a mean $\frac{\pi}{4}$ and a stand deviation of 0.01π. The true release location is assumed to be at $x_s = (2, 2)$ and the measurement time is given by $t_k = 1s$. The sensor location is given at $x_s = (0, 0)$. We apply ABC-PCGC to infer the original source location. The

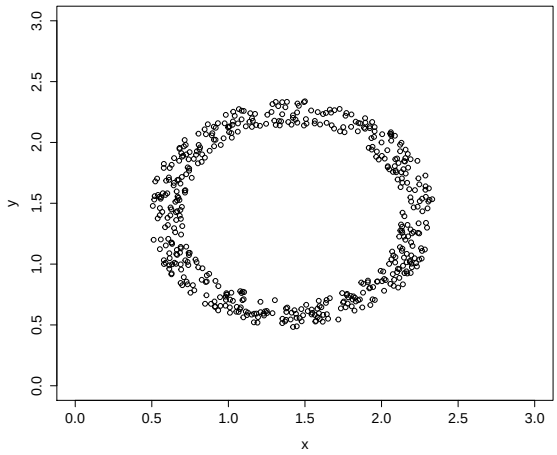

Figure 4.8: 2d Puff Source location: ABC-Basic Rejection with $\rho_{thresh} = 10$

prior distribution of the source location is given as follows:

$$p(x) = \mathcal{N}\left(x; \begin{bmatrix} 0 & 0 \end{bmatrix}, \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}\right) \tag{4.6}$$

The initial variance is set to $\sigma^2_{w,large} = K\rho_{thresh}$ and the distance function is given by $\rho(c, c^*)^2$ where c^* is the simulated concentration. As in the previous example $\beta = 1$.

First, ABC-basic rejection algorithm is applied to obtain samples from the posterior distribution, see Fig. 4.8. The estimated posterior is a circular ring shape distribution.

Next, we apply the ABC-PCGC algorithm to update the initial prior distribution. Based on the same splitting scheme given in Section 3.3, since the dimensionality of the source of the joint between source location and wind uncertainty is four dimensional, this Gaussian is approximated using 8 Gaussian components by splitting along all directions. First, we split the initial prior into 8 sub-Gaussian distribution and again split each sub-Gaussian components into another 8 components, then we obtain 64 Gaussian components to approximate the initial prior distribution of source location. As can be seen from Figure 4.9, the PCGC estimated posterior distribution can only cover partial solution of the ABC basic rejection. Next, we increase to use 521

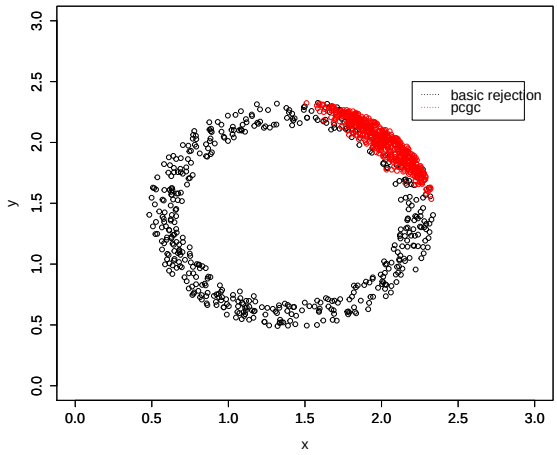

Figure 4.9: 2d Puff Source location: ABC-PCGC with $\rho_{thresh} = 10$, M=64

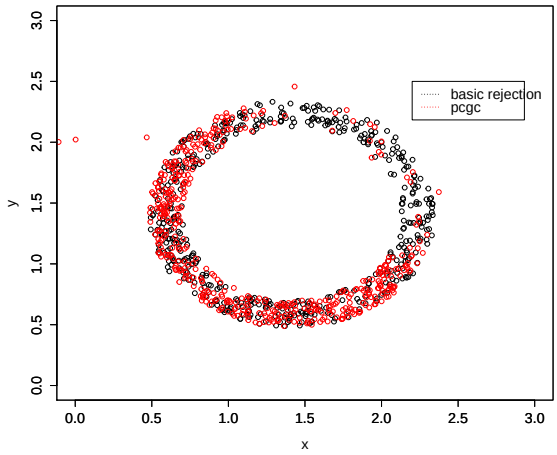

Figure 4.10: 2d Puff Source location: ABC-PCGC with $\rho_{thresh} = 10$, M=512

Gaussian components to represent the initial prior, the comparison can be seen from Figure 4.10. We repeat the simulation with different threshold values and compare the acceptance rate between the ABC-basic rejection and ABC Basic rejection with PCGC, see Table 4.3. The ABC-PCGC has one order of magnitude higher acceptance rate than acceptance rate of the ABC basic rejection algorithm. And the acceptance rate of PCGC with 512 Gaussian components is lower than the acceptance rate of PCGC with 64 Gaussian components, while high number of Gaussian components

simulation has a better estimation, see Fig. 4.9,4.10.

Table 4.3: 2d Puff model: Acceptance Rate

ρ_{thresh}	100	10	1
ABC-Rejection	0.04	0.01	0.004
ABC-PCGC M=64	0.22	0.16	0.029
ABC-PCGC M=512	0.11	0.09	0.019
K (ABC-PCGC)	100	1000	10000

Chapter 5

Conclusion and Future Works

5.1 Conclusion

Compared to the traditional Monte Carlo sampling, a new sampling algorithm is proposed to solve statistical inverse problems with intractable likelihood functions. The main contributions in the construction of the ABC-PCGC is the use of gradient information to construct proposal distributions that are constrained by the data using a reduce number of function evaluations. Hence, the new sampling strategy makes better use of observations, which is currently a major drawback of ABC methods.

The other contribution of the thesis is summarized as follows:

1. Instead of directly sampling from the prior distribution, we constructed a joint proposal distribution by introducing artificial noise. We convert a statistical inverse problem with intractable likelihood into an importance sampling problem. Sampling from this proposal will considerably improve the acceptance rate of the ABC-PCGC.

2. We approximate the prior with a finite sum of Gaussian density functions such that the mixture mean and covariance matrix matches the mean and covariance matrix of the prior distribution.

3. Once we obtain the prior proposal, then we could independently update each Gaussian components by iterating the progressive correction. Then the algorithm can be easily parallelized.

The preliminary numerical results provide evidence that the performance of the proposed algorithm is superior to the ABC rejection sampling.

5.2 Future Works

However, more work needs to be done to asses its performance on additional benchmark problems as well as comparison with more advanced ABC algorithms. Future studies are also planned to characterized the parameterization of the proposed ABC-PCGC algorithm.

1. The PCGC algorithm relies on four parameters, namely $\sigma^2_{w,large}$, β, M, and Σ, which determine its performance. Since the scope of the current thesis is just to introduce the proposed algorithm and have initial feasibility studies, future work is planned on characterizing the optimal values of the PCGC parameters.

2. Other prior approximation method can be used to refine or improve the prior proposal initialization.

3. The new PCGC algorithm should be applied into more complex real-world problem to test its performance.

Even though there is a lot of further research is needed, we develop a relative new idea of sampling in ABC framework.

Bibliography

[1] M. A. Beaumont, *Approximate bayesian computation in evolution and ecology*, Annual Review of Ecology, Evolution, and Systematics **41** (2010), 379–406.

[2] M. A. Beaumont, J. M. Cornuet, J. M. Marin, and C. P. Robert, *Adaptive approximate bayesian computation*, Biometrika **asp052** (2009), 1–8.

[3] M. A. Beaumont, W. Zhang, and D J Balding, *Approximate bayesian computatation in population genetics*, Genetics **162** (2002), 2025–2035.

[4] C. Becquet and M. Przeworski, *A new approach to estimate parameters of speciation models with application to apes*, Genome Research **17** (2007), 1505–1519.

[5] R. Bellman (ed.), *Adaptive control processes: A guided tour*, Princeton University Press, 1961.

[6] B. A. Berg (ed.), *Markov chain monte carlo simulations and their statistical analysis*, World Scientific, Singapore, 2004.

[7] G. Bertorelle, A. Benazzo, and S. Mona, *Bc as a flexible framework to estimate demography over space and time: some cons, many pros*, Molecular Ecology **19** (2009), 2609–2625.

[8] M. G. B. Blum and M. Jakobsson, *Deep divergences of human gene trees and models of human origins*, Molecular Biology and Evolution **28** (2010), 889–698.

[9] M. G. B. Blum and V. C. Tran, *Hiv with contact tracing: a case study in approximate bayesian computation*, Biostatistics **6** (2010), 644–660.

[10] P. Bortot, S. G. Coles, and S. A. Sisson, *Inference for stereoloical extremes*, Journal of the American Statistical Association **102** (2007), 84–92.

[11] O. CappÃĭ, A. Guillin, J. M. Marin, and C. P. Robert, *Population monte carlo*, Journal of Computational and Graphical Statistics **13** (2004), 907–929.

[12] J. M. Cornuet, F. Santos, M. A. Beaumont, C. P. Robert, J. M. Marin, D. J. Balding, T Guillemaud, and A Estoup, *Inferring population history with diy abc: a user-friendly approach to approximate bayesian computation*, Bioinformatics **24** (2008), 2713–2719.

[13] A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential monte-carlo methods in practice*, Springer-Verlag, apr, 2001.

[14] A. Estoup, M. A. Beaumont, F. Sennedot, C. Moritz, and Cornuet, *Genetic analysis of complex demographic scenarios: spatially expanding populations of the cane toad, bufo marinus*, Evolution **58** (2004), 2021–2036.

[15] M. Foll, M. A. Beaumont, and O. Gaggiotti, *An approximate bayesian computation approach to overcome biases that arise when using amplified fragment length polymorphism markers to study population structure*, Genetics **179** (2008), 927–939.

[16] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin Chapman (eds.), *Bayesian data analysise*, Springer-Verlag, apr, 2004.

[17] D. Giza and P. Singla, *An approach for nonlinear uncertainty propagation: Application to orbital mechanics*, In AIAA Guidance Navigation and Control Conference (2009).

[18] G Hamilton, M Currat, N Ray, G Heckel, M A Beaumont, and L Excoffier, *Bayesian estimation of recent migration rates after a spatial expansion*, Genetics **170** (2005), 409–417.

[19] F. Jabot and J. Chave, *Inferring the parameters of the neutral theory of biodiversity using phylogenetic information and implications for tropical forests*, Ecology Letters **12** (2009), 239–248.

[20] S.J. Julier and Uhlmann J.K., *Unscented filtering and nonlinear estimation*, Proceedings of the IEEE (2004), 401–422.

[21] M. Lenormand, F. Jabot, and G. Deffuant, *Adaptive approximate bayesian computation for complex models*, 8th World Congress in Probability and Statistics, 09/07/2012-14/07/2012, Istanbul, TUR, 2012, p. 11.

[22] P. Marjoram, J. Molitor, V. Plagnol, , and S. Tavare, *Markov chain monte carlo without likelihoods*, Proceedings of the National Academy of Sciences of the United States **100** (2003), 324–328.

[23] D. Moral, P. Doucet, and A. Jasra, *Sequential monte carlo samplers*, Journal of the Royal Statistical Society **68** (2006), 411–432.

[24] ———, *An adaptive sequential monte carlo method for approximate bayesian computation technical report*, Statistics and Computing **22** (2012), 1009–1020.

[25] C. Musso, N. Oudjane, , and F. Legland, *Improving regularized particle filters. in a. doucet and n. de freitas and and n. gordon sequential monte carlo methods in practice*, Statistics for Engineering and Information Science **12** (2001).

[26] V. Plagnol and S. Tavare (eds.), *Approximate bayesian computation and mcmc*, Springer-Verlag, 2004.

[27] J. K. Pritchard, M. T. Seielstad, and A. Perez Lezaun, *A population growth of human y chromosomes: a study of y chromosome microsatellites*, Molecular Biology and Evolution **16** (1999), 1791–1798.

[28] O. Ratamann, O. Jorgensen, T. Hinkley, M. Stumpf, S. Richardson, and Cwiuf, *Using likelihood-free inference to compare evolutionary dynamics of the protein networks of h. pylori and p. falciparum*, PLos Computational Biology **3** (2007), 2266–2276.

[29] K.V.U. Reddy, Y.Cheng, T.Singh, and P.D.Scott, *Data assimilation in variable dimension dispersion models using particle filters*, The 10th International Conference on Information Fusion (2007).

[30] K D SIEGMUND, P MARJORAM, and D SHIBATA, *Modeling dna methylation in a population of cancer cells*, Statistical application Genetics **7** (2008), 18.

[31] S. Sisson, Y. Fan, and M. M. Tanaka, *Sequential monte carlo without likelihoods*, Proceedings of the National Academy of Sciences of the United States **104** (2007), 1760–1765.

[32] M. Sunnaker, A. G. Busetto, E. N., J. Corander, and M. Foll, *Approximate bayesian computation*, PLOS Computational Biology **9** (2013), 1–9.

[33] M M Tanaka, A R Francis, F Luciani, and S A Sisson, *Using approximate bayesian computation to estimate tuberculosis transmission parameters from genotype data*, Genetics **173** (2006), 1511–1520.

[34] S Tavare, D J Balding, R C Griffiths, and P Donnelly, *Inferring coalescence times from dna sequence data*, Genetics **145** (1997), 505–518.

[35] G. Terejanu, P. Singla, T. Singh, , and P. D. Scott, *Uncertainty propagation for nonlinear dynamical systems using gaussian mixture models*, Journal of Guidance, Control, and Dynamics **31** (2008), 1623–1633.

[36] _____, *Decision based uncertainty propagation using adaptive gaussian mixtures*, In The 12th International Conference on Information Fusion (2009).

[37] _____, *A decision-centric framework for density forecasting*, Journal of advances in information fusion **5** (2010), 73–87.

[38] T Toni, D Welch, N Strelkowa, A Ipsen, and M P Stumpf, *Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems*, Journal of the Royal Society Interface **6** (2009), 187–202.

[39] B M Turner, *Likelihood-free bayesian modeling*, Ph.D. thesis, Ohio State University, Columbus, 2011.

[40] Brandon M Turner and Trisha Van Zandt, *A tutorial on approximate bayesian computation*, Journal of Mathematical Psychology **56** (2012), no. 2, 69–85.

[41] P. Verdu, F. Austerlitz, A. Estoup, R. Vitalis, M. Georges, S. Thery, A. L. Froment, S. Le Bomin, A. Gessain, J.M. Hombert, L. Van der Veen, L. Quintana-Murci, S. Bahuchet, and E. Heyer, *Origins and genetic diversity of pygmy hunter-gatherers from western central africa*, Current Biology **19** (2009), 312–318.

[42] R.D. Wilkinson, M. Steiper, C. Soligo, R. Martin, Z. Yang, and S. Tavare, *Dating primate divergences through an integrated analysis of palaeontological and molecular data*, Systematic Biology **60** (2010), 16–31.

[43] Z.H. Zhang, C. Chen, J. Sun, and K. L. Chan, *Em algorithms for gaussian mixtures with split-and-merge operation*, Pattern Recognition **36** (2003), 1973–1983.