Kernel Methods in Approximate Bayesian Computation

Jin ZHOU

Doctor of Philosophy

Department of Statistical Science
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies,
SOKENDAI

Kernel Methods in Approximate Bayesian Computation

Jin Zhou

Doctor of Philosophy

Department of Statistical Science

School of Multidisciplinary Sciences

SOKENDAI(The Graduate University for Advanced Study)

2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Bayesian methods provide intuitive ways to evaluate randomness and probability in areas of science, engineering and economics. In Bayesian inference, uncertainty of the future event is evaluated through Bayes risk, which depends on the posterior distribution $p(y_{obs}|\theta)$ of the unknown parameter θ of the model $\mathbf{M}$ given the evidence. Posterior distribution plays a central role in Bayesian methods. It contains all the information of the parameter θ that can be used for inference, model checking and decision making. The posterior distribution $p(\theta|y_{obs})$ can be formulated with the prior distribution $p(\theta)$ which is updated through the likelihood function $p(y_{obs}|\theta)$ with the observation y_{obs}. It can be written as:

$$p(\theta|y_{obs}) = \frac{p(y_{obs}|\theta)p(\theta)}{\int_\theta p(y_{obs}|\theta)p(\theta)d\theta}.$$

The prior distribution $p(\theta)$ represents the prior beliefs of the parameters; the likelihood function $p(y_{obs}|\theta)$ is the function of parameter θ of the specific

observed data y_{obs}. The partition function $p(y_{obs}) = p(y_{obs}|\theta)p(\theta)d\theta$ is the marginal distribution of the data. Due to the lacking of information on the true distribution $p(y_{obs})$, the partition function is often unavailable and represents one of the main challenges of Bayesian analysis. Another challenge arises when the likelihood function is not of explicit analytic form or is computationally too expensive to evaluate, which calls for methods that do not need to evaluate the likelihood functions.

In the early days of Bayesian inference, due to limited computing power, the only feasible way of doing Bayesian inference is by using conjugate priors with likelihoods in the forms of exponential families. In this way, the posterior distribution falls into the same family as the prior distribution and can be explicitly analyzed. This method works fine for simple statistically models in the sense that the posterior distribution can be reasonably modeled by a parametric model. However, with the rapidly increased computing power, these limitations are no longer necessary. Significantly more complex statistical models are developed to better fit the reality.

When sampling is done with sample size of thousands or even millions, Monte Carlo integration is utilized as the numerical approximation method in many Bayesian methods to circumvent the above mentioned first challenge. Instead of directly deriving the distribution function of the posterior distribution $p(\theta|y_{obs})$, we draw samples $(\theta_1, \theta_2, ..., \theta_n)$ from the posterior distribution and then approximate it using the empirical distribution

$$p(\theta|y_{obs}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{\Theta}(\theta)$$

where δ_θ is the Dirac measure of θ such that $\delta_\Theta(\theta) = 1$ if $\theta \in \Theta$ and $\delta_\Theta(\theta) = 0$ otherwise. By the law of large number, $\frac{1}{N}\sum_{i=1}^{N} \delta_\Theta(\theta) \to p(\theta|y_{obs})$ in distribution when $N \to \infty$.

There are a large number of Monte Carlo Methods and their variants proposed for the posterior approximation including the Rejection-Acceptation method, Markov Chain Monte Carlo (MCMC) method and Sequential Monte Carlo method (SMC)[33][13][41]. In the case of Rejection-Acceptation method, each sample is generated by the random number generator independently, so the convergence applies. For Monte Carlo chains based methods, if the Mote Carlo chains are irreducible and recurrent, then the chain has the same limiting distribution from almost every starting point. In this case, the sample in the chain can be considered as drawn from the limiting distribution without care for the beginning of chain when estimating the function f. The accuracy of the approximation depends on the number of the sample drawn from the distribution and on the support of the density. These problems are about efficiency: to design an efficient algorithm that can reduce the variance of the estimation at a manageable speed.

Monte Carlo based sampling methods rely on the repeated evaluations of the likelihood function. As in the MetropolisHastings algorithm (MCMC), the samples are simulated in a sequential manner. For each time, the newly simulated sample $\theta^{(n+1)}$ is accepted by the probability

$$\frac{p(\theta^{(n+1)}|y_{obs})q(\theta^n|\theta^{(n+1)})}{p(\theta^n|y_{obs})q(\theta^{(n+1)}|\theta^n)} = \frac{p(y_{obs}|\theta^{(n+1)})p(\theta^{(n+1)})q(\theta^n|\theta^{(n+1)})}{p(y_{obs}|\theta^n)p(\theta^n)q(\theta^{(n+1)}|\theta^n)}$$

where θ^n is the current state, and $q(\theta^1|\theta^2)$ is the transitional kernel of θ, p is the prior density of the parameter θ. For each newly generated sample, the likelihood function has to be evaluated for each observation y_{obs}, and the transition kernel plays a practical role in the efficiency of the algorithm: setting a variance too large or too small will make the convergence of chain unmanageable.

As discussed above, the unavailability of the marginal distribution is circumvented by evaluating the ratio of density instead of directly calculating the density itself. Yet still, as seen in the Metropolis-Hastings method, the likelihood function has to be evaluated repeatedly in the process. For many cases, for example, if the likelihood function is computationally too demanding, or if the number of observation y_{obs} is too large, which often occurs in the big data, or if it is only known partially or lacks a functional form and is implicitly defined using a data generation program as in the population genetics, then the MCMC methods cannot be applied. For these areas, one approach is to use a different model which can be analytically analyzed; but this approach is often not optimal, as the underlying problem is often complex and calls for a complex model to describe. Another option is to use approximation; instead of using an over-simplistic model and under-fitting the data, using a complex model and making approximations is more attractive, as the discrepancy between the true model and the approximation can be measured by approximation error. This leads to a lot of interests in the so-called likelihood-free methods.

Recently, one of the likelihood-free algorithm called Approximate Bayesian Computation (ABC) becomes more and more popular among many areas [21]

[31] [43] [48] [5] [14] [2] [4]. It is introduced to make inference in the cases that the likelihood functions are intractable. Statistical models used in these cases have no explicit function forms; they are instead described by some generating programs which are designed to simulate the true underlying stochastic processes. The simulation data are used in the later inferences as samples from the desired posterior distribution. Since the program is usually much more complex than a simple function, ABC enables the domain experts to have much more expressing powers in describing their understandings of the true phenomenon and thus greatly expands the landscapes of the existing Bayesian inference algorithms.

1.1 Approximate Bayesian Computation

The fundamental difference between ABC and other Bayesian approximation methods is that the likelihood function need not to be known provided that it can be described implicitly by the generating program. No evaluations of density functions or likelihood functions are involved in the computing process. ABC does not directly sample from the posterior distribution. Instead, a sufficiently large set of parameters are generated first from some prior distribution, then the data points are simulated by the generating program using these parameters as inputs. The approximation then can be done by accepting only the data points that are close to the observation where closeness is measured by some distance measure, usually Euclidean. Then these data points are treated as i.i.d samples from the true posterior and further inferences of the parameters like posterior means and variances can be made.

This algorithm is based on the Rejection ABC.

In this section, an intuitive introduction of the Rejection-Acceptation ABC (Rejection ABC) is given [43]. The more sophisticated expansions of ABC using Monte Carlo chain based methods will be described in chapter 2 in detail.

A Rejection ABC algorithm takes a prior distribution of the parameter, the generating program and the observation as input; generating many sample pairs (θ_i, y_i) from the prior distribution and the program; then accept those that are same as the observation.

1. Inputs
   (a) Prior density function $q(\theta)$
   (b) Generating program $f(y|\theta)$ that produce y given input θ
   (c) Observation y_{obs}
2. Sampling Process
   (a) Sample independent parameters $(\theta_1, \theta_2, ..., \theta_n)$ from $q(\theta)$
   (b) Generate $(y_1, y_2, ..., y_m)$ using $f(y|\theta)$
   (c) Compare $(y_1, y_2, ..., y_m)$ with (y_{obs})
   (d) Accept y_i if $y_i = y_{obs}$
3. Outputs
   (a) A set of accepted sample with parameters θ_i as from the posterior distribution

In this strict version of rejection sample, as the generated sample strictly equals the observation, it can be considered as drawn from the true posterior distribution by a classic rejection-acceptation algorithm. To understand this, consider a classic rejection-acceptation algorithm with a target distribution function $f(x)$ and a sampling distribution $g(x)$. A sample is accepted with probability $\frac{f(x)}{Mg(x)}$, where $M > \frac{f(x)}{g(x)}$. In the case of Bayesian inference, $f(x)$ is the target posterior distribution which is proportional to $p(y|\theta)p(\theta)$ and $g(x)$ is the prior distribution. In this case, the acceptation probability is the same as the likelihood function $p(y|\theta)$ up to some constant. In the above described Rejection ABC, the output of the generating program y given θ can be seen as the realization of the likelihood function at θ; thus the sample can be seen as from the true posterior distribution.

However, to generate enough samples by $y_i = y_{obs}$ is very inefficient, or even impossible given the randomness of the generating program. In reality, a small threshold h is introduced in the distance metric function. Thus the step 3 of the sampling Process in Rejection ABC becomes: Accept y_i if $||y_i - y_{obs}|| < h$. If the distance between the generated sample and the observation is small enough, the sample is considered as from the true posterior. By introducing this threshold, an approximation error has also induced in the sampling. A large threshold induces a large acceptance rate, thus the speed of sampling. However, if the threshold is set too large, bias is introduced in the sampling and the approximation becomes less accurate. This is a central consideration in designing ABC algorithms.

There are two challenges associated with ABC. First, as only the accepted data are used in the inference, ABC algorithm becomes very inefficient when

the accepting rate gets low, which often is the case if uninformative priors are used or the dimensionality of the data is high. It is of great interests to get more efficient sampling algorithm than the simple rejection algorithm described above. To address this problem, a lot of sophisticated sampling methods have been introduced to ABC, like MCMC and Sequential Monte Carlo (SMC).

The second challenge is closely related to the first one. The simulated data come out of the simulation program are often of high dimensionalities, like gene sequence. And the acceptance is based on the distance function which suffers significantly from the curse of dimensionality. It is then a common practice to use summary statistics instead of the original data in the distance function. This approximation induces possible loss of information and can lead to biased inference. To avoid this problem, a relatively large set of original summary statistics are proposed by the domain experts first; then a dimensional reduction algorithm is applied to further reduce the dimension while preserving the information.

1.2 Contents of Thesis

In this thesis, we focus on the applications of kernel methods to the ABC to provide an automatic algorithm which can produce low dimensional summary statistics while preserving information. As described above, summary statistics play a central role in the efficiency and accuracy of the ABC methods. It is important that the dimensional reduction algorithm can achieve the lowest dimension without information loss. Although a lot of dimensional reduction

methods have been introduced to ABC already, an automatic algorithm with theoretically sound guarantees is still missing.

This thesis introduces a kernel based sufficient dimensional reduction algorithm to solve the above problem. Sufficient dimensional reduction (SDR) is a classic type of dimensional reduction algorithms that guarantees to find the sufficient lower dimensional subspace provided that the assumptions of the underlying space are met. In here sufficient means no information loss. As the assumptions of classic SDR are often too restrictive for real-world problems, we instead draw the idea from the kernel dimensional reduction method.

To provide a principled way of designing the regression function, capturing the higher order non-linearity and realizing an automatic construction of summary statistics, this thesis introduces the kernel based sufficient dimensional reduction method. This dimensional reduction method is a localized version of gradient-based kernel dimensional reduction (GKDR) [17]. GKDR estimates the projection matrix onto the sufficient subspace by extracting the eigenvectors of the kernel derivatives matrices in the reproducing kernel Hilbert spaces (RKHS). We give a brief review of this method in Chapter 3. In addition to the GKDR, in which the estimation averages over all data points to reduce variance, a localized GKDR is proposed by averaging over a small neighborhood around the observation in ABC. Each point is weighted using a distance metric measuring the difference between the simulated data and the observation. The idea is similar to the role of the distance kernel function.

Another proposal is to use different summary statistics for different pa-

rameters. Note that sufficient subspace for different parameters can be different, depending on the particular problem. In these cases, applying separated dimensional reduction procedures yield better estimations of the parameter.

Three experiments are investigated in the thesis to evaluate the proposed method against popular dimensional reduction methods. Each experiment is conducted with two sampling algorithms: Rejection ABC and Sequential-ABC (SABC). The former provides an intuitive overall comparison and the later is used to access the generated summary statistics in the extreme situations that the threshold of the distance function is pushed to as small as possible. This strategy makes the latter experiments very time consuming, but provides a useful assessment on the generated summary statistics.

1.3 Outline

In Chapter 2 we give a detailed introduction of ABC algorithms including the Rejection ABC, MCMC ABC and Sequential-ABC (SABC). Rejection ABC uses simple rejection method and is very easy to implement, it provides a baseline for analyzing other more advanced sampling method. MCMC ABC introduce a ABC version of MCMC to improve the sampling efficiency of the Rejection ABC. SABC use sequential generations of parameters and reduce the distance threshold in the meantime. It can be used to achieve a very small threshold.

In Chapter 3 we give a brief introduction to the kernel based dimensional reduction method. First the theoretical foundations of kernel methods are briefly introduce. Then a more detailed introduction is given on deriving

the GKDR. These basic details are provided for the understanding of latter chapters.

In Chapter 4 we develop the main contribution: local dimensional reduction algorithm and the separated construction of summary statistics. Discussions on hyper-parameters and computation time are included.

In Chapter 5 we investigate in detail three experiments including a Queue model, a population genetics model, and a dynamic system model. Comparison between different dimensional reduction methods is provided for each of the two sampling algorithms. In the end of this chapter, a comparison between LGKDR, GKDR and sliced inverse regression is given using Queue mode, it provides the motivation for the whole work.

In Chapter 6 we give the conclusion of the thesis and discuss the possible future directions.

Chapter 2

Approximate Bayesian Computation

As briefly discussed in Chapter 1, likelihood-free methods are gaining interests due to their ability to do inference without explicitly evaluate the likelihood function. This property makes these methods suitable for a lot of complex statistical models where explicit function forms are not available. Within the likelihood-free algorithms, ABC is a Monte Carlo method that approximates the posterior distribution by jointly generating simulated data and parameters and does the sampling based on the distance between the simulated data and the observation, without evaluating the likelihoods. ABC was first introduced in population genetics [40] [4] and then have been applied to a range of complex applications including dynamical systems [49], ecology [12], Gibbs random fields [22] and demography [5].

Bayesian inference works through updating the posterior distribution via $p(\theta|y_{obs}) \propto p(y_{obs}|\theta)p(\theta)$, where θ is the parameter of the assumed statistical

model, $p(y_{obs}|\theta)$ is the likelihood of the model and y_{obs} is the observed data point. By observing more and more data, the posterior is moving from the prior distribution to the position of maximum likelihood. Bayesian inference works very well on the situation where prior information occupies an important factor in the model and the observation is scarce which often is the case in the scientific setting. Bayesian inference relies on the evaluation of the likelihood function in the updating process and the likelihood determines the underlying statistical model.

Likelihood-free methods or ABC represent a type of methods that can be used where likelihood is intractable. This intractability includes several situations: it is computationally intractable to evaluate the likelihood function point-wise; the likelihood function cannot be expressed in analytic form or the statistical model is available but can not be solved analytically. To bypass the evaluation of the likelihood function, ABC introduces an approximation to the evaluation of the likelihood function. ABC is composed of the following components: a generating model, often written as a computer program that generates the synthesized data-set; summary statistics that transform the data to low dimensional vectors which plays a key factor in the sampling efficiency and a sampling method that employ the summary statistics and the generating model to efficiently sample from the approximate posterior distribution.

The accuracy of ABC posterior depends on sufficiency of summary statistics and Monte Carlo errors induced in the sampling. Before going into details of the three components, first we give a general introduction to the method:

Given the generating model $p(y|\theta)$ of observation y_{obs} with parameter

θ, consider summary statistics $s_{obs} = G_s(y_{obs})$ and $s = G_s(y)$, where $G_s : Y \to S$ is the mapping from the original sample space Y to low dimensional summary statistics S. The posterior distribution, $p(\theta|y_{obs})$, is approximated by $p(\theta|s_{obs})$, which is constructed as $p(\theta|y_{obs}) \approx \int p_{ABC}(\theta, s|s_{obs})ds$, with

$$p_{ABC}(\theta, s|s_{obs}) \propto p(\theta)p(s|\theta)K(\|s - s_{obs}\|/\epsilon), \tag{2.1}$$

where K is a smoothing kernel with bandwidth ϵ. In the case of Rejection ABC, K is often chosen as an indicator function $I(\|s - s_{obs}\| < \epsilon)$. If the summary statistics s are sufficient, it can be shown that $p(\theta|s_{obs})$ reduces to $p(\theta|y_{obs})$ as ϵ goes to zero[6].

As shown above, the sampling is based on the distance between the summary statistics of the simulated sample s and the observation s_{obs}. Approximation errors are induced by the distance measure and are proportional to the distance threshold ϵ. It is desirable to set ϵ as small as possible, but a small threshold will increase the simulation time. This is a trade-off between the accuracy and the efficiency (simulation time). According to recent results on asymptotic properties of ABC [16] [30], assuming that the summary statistics follow the central limit theorem, the convergence rate of ABC when accepted sample size $N \to \infty$ is depended on the behavior of $\mu = \epsilon d_N$, where ϵ is the threshold above and the d_N is defined as of the same magnitude of $eigen(\Sigma_N)$, the eigenvalues of the covariance matrix of the summary statistics as the function of N. In practice, if a specific sampling method is chosen, the threshold ϵ is constrained by the computing resources and time, thus can be accordingly determined. The design of summary statistics then remains the

most versatile and difficult part in developing an efficient ABC algorithm. To avoid the "curse of dimensionality", summary statistics should be low dimensional in addition of sufficiency.

A vast body of literature of ABC has been published. Many are devoted to reduce the sampling error by using more advanced sampling methods, from simple Rejection method[34], Markov Chain Monte Carlo(MCMC)[32] to more sophisticated methods like sequential Monte Carlo [44][49] and adaptive sequential Monte Carlo methods [35].

In the following sections, we will give more details about the components of ABC.

2.1 Rejection ABC

An elemental form of the Rejection ABC algorithm has been introduced in Chapter 1. In this section, I will give a more detailed explanation of the algorithm and its related issues [43].

The Rejection ABC algorithm introduced in Chapter 1 use the exact equal condition $d(y, y_{obs}) = 0$ to determine whether the generated sample is indeed drawn from the true posterior distribution. For discrete models with finite parameter/data space, this condition can be met but the acceptance rate, which is $y_{accepted}/y_{all}$, is very small, thus highly inefficient. For the continuous models where the probability of $d(y, y_{obs}) = 0$ is zero, this algorithm simply can not be applied.

To improve the efficiency and to apply to continuous models, a distance threshold is used in the Rejection algorithm. Instead of asking $d(y, y_{obs}) = 0$,

by setting a threshold h, the generated sample is accepted if $d(y, y_{obs}) < h$. This relaxation can significantly improve the acceptation rate and the efficiency, depending on the dimensionality of the data. The threshold introduced here is an approximation of the true posterior distribution, the bigger the threshold, the less accurate the approximation is. Formally, the approximated posterior can be written as:

$$p(\theta|y_{obs}) \propto I(||y_{obs} - y|| < h)p(y|\theta)g(\theta)$$

where I is the indication function, $p(y|\theta)$ is the data generated from the simulating program, representing the evaluation of the likelihood function of the model, $g(\theta)$ is the generation of the parameter from the prior distribution.

Setting a proper threshold for a particular model is a difficult design choice, especially if the dimensionality of the data y is high. However, for a typical model in ABC applications, the dimensionality of y can easily be a few dozens or even a few hundreds. In these cases, setting the threshold small may result in very low acceptation rate, rending the algorithm useless. However, setting the threshold too high is also a poor choice since the effect of "curse of dimensionality". Thus a second approximation is introduced to ABC, called summary statistics.

2.1.1 Summary Statistics

The synthetic data y generated from the model is often of high dimensionality. Direct comparison between the generated data y and the observation y_{obs} suffers significantly from the curse of dimensionality. The resulting sampling

efficiency is too low to be practical. Summary statistics that are sufficient are then used in the distance function. Sufficient summary statistics contain all the information for the inference theoretically, resulting in the unbiased estimation of the parameters. Thus, the comparison between the observation and the data becomes $||s_{obs} - s||$, where $s = S(y), s_{obs} = S(y_{obs})$, S is the summary statistics. In this case, if the s is sufficient that it contains all the information of y, then the approximation of posterior $p(\theta|y_{obs}) = p(\theta|s_{obs})$; otherwise, if S is not sufficient, then $p(\theta|s_{obs})$ can be understood as an approximation of $p(\theta|y_{obs})$, thus a second approximation is introduced by the usage of summary statistics. The sufficiency of summary statistics is then one of the most important factors whether an ABC algorithm is accurate.

Addition to sufficiency, low dimensionality is also an important requirement of summary statistics. Low dimensionality plays a central role in avoiding the "curse of dimensionality". It is often desirable to have a set of summary statistics that the number of the summary statistics is smaller than at least 10. However, in reality, summary statistics are rarely sufficient, especially if the dimensionality is low. And it is non-trivial to determine whether it is sufficient or not.

Traditional summary statistics such as mode, mean and quantiles are often used as summary statistics, as they are used in parametric models. But it is understandable that only use these statistics is rarely sufficient for any model that has a likelihood function more complex than exponential family distributions, not mention the models which can only be described using a simulation program and lack any functional form. For these kinds of models, since the models are often developed by the domain experts, they can

often contribute some highly informative summary statistics based on their understanding of the generating model and the underlying scientific problem. These statistics often contain a lot of information and of low dimensionality, but they are rarely sufficient.

There are two problems associated with these kinds of handpicked statistics. First, it is difficult to determine that, from which point the statistics are sufficient; or is there existed a set sufficient statistics. Second, choosing a set of appropriate summary statistics is much more difficult for complex models, the information extracted by the domain experts are limited in these cases; as the data itself become high dimensional and difficult to understand, as in the case of the gene data, it is hard to design a set a summary statistics that extract all the information contained in the data. To address this problem, a set of redundant summary statistics are often constructed as initial summary statistics. Sufficiency rather than dimensionality is the priority considerations in this process. After obtaining a large set of possible summary statistics, dimensional reduction methods are then applied to yield a set of low dimensional summary statistics while persevering the information. This approach can be understood as a detour that utilizes the techniques of dimensional reduction algorithms, which have been thoroughly studied and consist of a lot of different algorithms.

Many dimensional reduction methods have been proposed for ABC. Entropy-based subset selection [28], partial least square [52], neural network [7] and expected posterior mean [15] are a few of them. The entropy-based subset selection method works well in instances where the set of low dimensional summary statistics is a subset of the initial summary statistics, but the com-

putational complexity increases exponentially with the size of the initial summary statistics. The partial least square and neural network methods aim to capture the nonlinear relationships of the original summary statistics. In both cases, a specific form of the regression function is assumed. A comprehensive review [8] discusses the methods mentioned above and compares the performances. While the results are a mixed bag, it is reported that the expected posterior mean method (Semi-automatic ABC) [15] produces relatively better results compared to the methods mentioned above in various experiments. It is a popular choice also due to its simplicity.

Semi-automatic ABC [15] uses the estimated posterior mean as summary statistics. A pilot run of ABC is conducted to identify the regions of parameter space with non-negligible probability mass. The posterior mean is then estimated using the simulated data from that region and is used as the summary statistics in a formal run of ABC. A linear model of the form: $\theta_i = \beta^{(i)} f(\mathbf{y}) + \epsilon_i$ is used in the estimation, where $f(y)$ are the possibly non-linear transforms of the data. For each application, the features $f(y)$ are carefully designed to achieve a good estimation. In practice, a vector of powers of the data $(\mathbf{y}, \mathbf{y^2}, \mathbf{y^3}, \mathbf{y^4}, ...)$ is often used as noted in [15].

2.1.2 Distance Kernels

As discussed above, the ABC approximate estimation will converge to the true posterior if the threshold goes to 0, assuming that the summary statistics are sufficient. However, a threshold that is too small will result in a significantly lower acceptance rate and worsen sampling efficiency. Instead,

in practice, a threshold that gives a trades off between time and accuracy is preferred and a distance function is used to improve the convergence of the estimation.

Given summary statistics and the observation, a distance kernel function K with bandwidth h is used to determine the acceptance of the generated sample. Thus the posterior is then written as $p(\theta|y_{obs}) = K_h(||y - y_{obs}||)p(y|\theta)p(\theta)$. There are many distance functions used in the literature. The most simple one is the indication function. Given observation y_{obs}, summary statistics S, generated data and parameter pair (y_i, θ_i), threshold ϵ, a distance function d, the probability of acceptance δ is determined by:

$$\delta = \begin{cases} 1 & d(S(y), S(y_{obs})) < \epsilon \\ 0 & d(S(y), S(y_{obs})) \geq \epsilon \end{cases}$$

The accepted sample are then given the same weights in the estimation of the parameter as: $\theta = \frac{1}{N}\sum_{i=1}^{N}\theta_i$. The indicator function can also be viewed as a uniform kernel function. Another often used kernel function is the Gaussian function:

$$w_i = e^{-(\frac{||S(y_i)-S(y_{obs})||}{2\epsilon})^2}$$

The Gaussian distance spread the weights across the whole parameter space. when we prefer a more concentrated function, Epanechnikov kernel is often used:

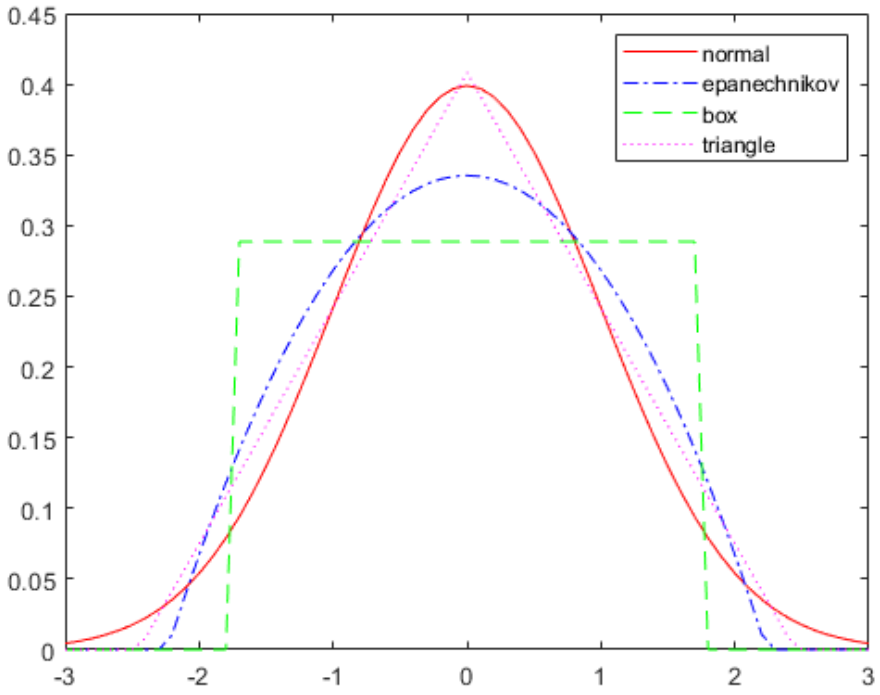

Figure 2.1: Different kernel functions typically used in ABC

$$\begin{cases} 1 - (\frac{||S(y_i) - S(y_{obs})||}{\epsilon})^2 & d(S(y), S(y_{obs})) < \epsilon \\ 0 & d(S(y), S(y_{obs})) \geq \epsilon \end{cases}$$

Depended on the application and the summary statistics, different kernel distance function can be chosen.

2.1.3 Algorithm

In this section, we combine the several tools introduced in the previous sections and introduce the final version of the Rejection ABC algorithm.

A Rejection ABC algorithm takes the prior distribution of the parameter,

the generating program and the observation as input; then it generates many pairs of (θ_i, y_i) from the prior distribution through the generating program; summary statistics and kernel functions are then used to determine how close the generated sample is to the observation; at last those samples that are close to the observation area accepted. Accepted samples are seen as drawn from the approximate posterior distribution.

1. Inputs
   (a) Prior density function $q(\theta)$
   (b) Generating program $f(y|\theta)$ that produce y given input θ
   (c) Observation y_{obs}
   (d) summary statistics S
   (e) kernel function K_h
2. Sampling Process
   (a) Sample independent parameters $(\theta_1, \theta_2, ..., \theta_n)$ from $q(\theta)$
   (b) Generate $(y_1, y_2, ..., y_m)$ using $f(y|\theta)$
   (c) Calculate summary statistics $(s_1, s_2, ..., s_n)$ by $s = S(y)$
   (d) Compare $(s_1, s_2, ..., s_m)$ with (s_{obs}) where $s_{obs} = S(y_{obs})$
   (e) Accept s_i if $K_h(||s_i - s_{obs}||) > 0$
3. Outputs
   (a) A set of accepted sample with parameters θ_i as from the posterior distribution

Here concludes the introduction of the Rejection ABC, in the next section, two advanced sampling methods are introduced to enhance the sampling efficiency.

2.2 Advanced Sampling Methods

At first, acceptance/rejection method is used in the original ABC. However, although the rejection method is conceptually simple, it is computationally very inefficient. For fast prototyping and comparison of different dimensional reduction methods, rejection method is still commonly used due to its simplicity. But for complex models which sampling efficiency is important, MCMC-ABC and Sequential ABC are more commonly used to achieve a higher sampling performance. In the following section, we give a brief introduction of these methods.

2.2.1 MCMC-ABC

Markov chain Monte Carlo (MCMC) methods are widely used in the applications where the distributions are complex. Compared to sequential Monte Carlo, MCMC methods are relatively straightforward to implement. A MCMC method samples a series of examples from a Markov chain, of which the limiting distribution is the target distribution as long as the Markov chain is irreducible. Then a sample from the chain can be considered as sampled from the target distribution. MCMC methods are extremely popular among Monte Carlo methods, its ABC version is also proposed after the Rejection ABC shows potential.

For the general MCMC method, given the existing state θ, the transitional MCMC algorithm propose a new parameter θ' from θ using a translation kernel k, θ $k(\theta, \theta') = k(\theta'|\theta)$, the newly generated sample is accepted with probability of $\min(1, \frac{p(\theta')k(\theta'|\theta)}{p(\theta)k(\theta|\theta')})$, where p is the target distribution. If the new proposal is accepted, the state of the chain become θ', otherwise it remains at θ. This conceptional simple algorithm achieves great success, and is widely used in all areas of Bayesian inference.

The ABC version of MCMC is in spirit similar to Rejection ABC, the evaluation of the likelihood is approximated by a distance-based sampling process [39] [32]. Formally, given the target distribution of the chain $p_{ABC}(\theta|s_{obs})$, the likelihood function $f(s|\theta)$, distance kernel K_h, the proposal distribution can be written as $k(\theta'|\theta)f(s'|\theta')$. And the acceptance rate α can be written as

$$\begin{aligned}\alpha &= \frac{p_{ABC}(\theta', s'|s_{obs})k(\theta'|\theta)f(s|\theta)}{p_{ABC}(\theta, s|s_{obs})k(\theta|\theta')f(s'|\theta')} \\ &= \frac{K_h(||s' - s_{obs}||)f(s'|\theta')\pi(\theta')k(\theta'|\theta)f(s|\theta)}{K_h(||s - s_{obs}||)f(s|\theta)\pi(\theta)k(\theta|\theta')f(s'|\theta')} \\ &= \frac{K_h(||s' - s_{obs}||)\pi(\theta')k(\theta'|\theta)}{K_h(||s - s_{obs}||\pi(\theta)k(\theta|\theta')}.\end{aligned}$$

where p denotes the likelihood function.

Thus the algorithm of MCMC-ABC can be written as follows:

1. Inputs
   (a) Initial state θ
   (b) Proposal distribution $k(\theta'|\theta)$

(c) Generating program $f(y|\theta')$ that produce y given input θ'

(d) Observation y_{obs}

(e) summary statistics S

(f) kernel function K_h

2. Sampling Process

(a) Sample θ' from $k(\theta'|\theta)$

(b) Generate (y) using $f(y|\theta')$

(c) Calculate summary statistics s by $s = S(y)$

(d) Compare (s) with (s_{obs}) where $s_{obs} = S(y_{obs})$

(e) Accept the generated state with probability $min(1, \alpha)$

(f) Return to step Process 1

3. Outputs

(a) A set of accepted sample with parameters θ_i as from the posterior distribution

Compared to Rejection ABC, MCMC-ABC generate a sample from a Markov chain which is more efficient than the random generation from uniform distributions. But the method suffers from "sticking" problem, which the parameter is stuck in the area where density is very low, and it becomes very difficult to jump out of this low-probability area. Also, the threshold has to be set fixed, and it is difficult to determine this parameter.

2.2.2 Sequential-ABC

Rejection ABC and MCMC-ABC both use fixed thresholds throughout the sampling process. Since the threshold controls the accuracy of the ABC approximations, it is desirable if an adaptive threshold can be used. If the threshold decreases as the sampling continues, the accuracy of the approximation can improve over time. It is also beneficial for the setting up stage of the algorithm where a relatively large threshold can be used to increase the acceptance rate. Sequential Monte Carlo [13] constructs a series of distributions in the sampling process and acts as a proper candidate for an adaptive threshold setting. In Sequential-ABC (SMC) [35] [44], a series of distributions π_i are formed with decreasing thresholds $\epsilon_i < \epsilon_{i-1} < \epsilon_{i-2}....$ The samples generated in the previous stage are reused in the later approximations with re-sampling if the weights degenerate.

Formally, SMC sampler approximates a sequence of probability distributions $\pi_{n0 \leq n \leq T}$, which are approximated by a set of samples of size N, called particles. At time 0, a simple distribution π_0 is used such that it is easy to sample from. N random samples ${Z_0}_i^T$ are sampled from distribution π_0. Then at time n, the particles ${Z_n}_i^T$ are moved using a Markov kernel K_n which defines the probability that a sample is moving from Z_n to Z_{n+1}. During the transition of states, the weights of the particles W_n becomes smaller, indicating that it is more difficult to move the particle to the region that has a high probability measure. The Effective Sample Size (ESS) is defined as

$$ESS(W_n^{(i)}) = (\sum_{i=1}^{N}(W_n^{(i)})^2)^{-1} \tag{2.2}$$

where the weights of the particles W_n is updated by

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_n^{(i)}) L_{n-1}(Z_n^{(i)}, Z_{n-1}^{(i)})}{\pi_{n-1}(Z_{n-1}^{(i)}) K_n(Z_{n-1}^{(i)}, Z_n^{(i)})}. \tag{2.3}$$

As in [35], if an MCMC kernel of invariant distribution π_n for K_n is selected, and use the backward kernel as

$$L_{n-1}(z, z') = \frac{\pi_n(z') K_n(z', z)}{\pi_n(z)}$$

the weight update be becomes

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_{n-1}^{(i)})}{\pi_{n-1}(Z_{n-1}^{(i)})} \tag{2.4}$$

where $\pi_n(Z_n^{(i)})$ is depended on ϵ_n since the distribution is approximated by an ABC samplers which contains ϵ_n.

The ESS criterion takes values between 1 and N. It indicates that the inference based on the N weighted samples is approximately equivalent to the inference based on $ESS(W_n^{(i)})$ samples. When the value of ESS drops to very small, it is necessary to re-sample the whole particle set. As mentioned above, it is favourable to gradually reduce the threshold ϵ_n in the sequence. A simple way to selecting ϵ_n is by control the ESS over iterations by selecting the tolerance level ϵ_n such that

$$ESS(W_n^{(i)}, \epsilon_n) = \alpha ESS(W_{n-1}^{(i)}, \epsilon_{n-1}) \tag{2.5}$$

for $\alpha \in (0, 1)$, $W_n^{(i)}$ is given in 2.4 which depends on ϵ_n. The parameter α controls the speed and smoothness of the sequence. If $\alpha \approx 1$, the speed is

slow but the most of the particles will be able to move to the the next state. Instead if $\alpha \approx 0$, the sequence will move very quickly towards the target but the result will be unreliable.

The whole algorithm is summarized as follows:

1. Inputs
   (a) Initial distribution π_0
   (b) Markov transition kernels K_t
   (c) Threshold ϵ
   (d) Generating program $f(y|\theta')$ that produce y given input θ'
   (e) Observation y_{obs}
   (f) Summary statistics S
   (g) Kernel function K_h
2. Sampling Process, in step n
   (a) Sample $\theta_n^{(i)}$ from $K_t(\theta_n^{(i)}|\theta_{n-1}^{(i)})$
   (b) Determine ϵ_n to make it satisfying (2.5)
   (c) Update weights W_{n+1} by (2.4)
   (d) If $ESS(W_n^{(i)}) < N_T$, where N_T is the size threshold, resample the sample and reset the weights to $1/N$, set n to $n+1$
   (e) Return to step Process 1
3. Outputs

(a) A set of accepted sample with parameters θ_i as from the posterior distribution

When the ϵ drops to 0 or begins to drop very slowly, it will be safe to stop the sampling. And the resulting ϵ will be recorded as the final threshold. The final set of particles can be used as a sample from the target distribution. The accuracy of this approximation is depended on the final ϵ.

Chapter 3

Gradient-based Kernel Dimensional Reduction

In this chapter, a self-contained introduction to Gradient-based Kernel Dimensional Reduction (GKDR) is given before the introduction of the main method in the next chapter. Since there are many types of dimensional reduction methods exist in the literature, to give a motivation in choosing GKDR, we first introduce the idea of Sufficient Dimensional Reduction, which works well on regression problems.

3.1 Sufficient Dimensional Reduction

Sufficient Dimensional Reduction (SDR) [29] [10] aims to estimate the linear projection directions that project the explanatory variables to a lower dimensional subspace. It assumes that for a specific function, a sufficient dimensional reduction space exists such that all the information that is rel-

evant to the response variable y inside explanatory variable x are contained in the subspace B, which is called the "Sufficient Dimensional Reduction Space" (s.d.r space). Depended on the problem it tries to solve, different level of assumptions can be made. In the broadest setting, it can be written as $p(y|x) = \tilde{p}(y|B^T x)$, where p is the probability density function. In this strictest assumption, the probability distribution of y is independent of x, given $B^T x$. While this assumption may be too restrictive, in the regression problem, a relaxed assumption is sufficient. That is, given the regression function $E(y|x)$, The SDR methods are designed to find this s.d.r space. SDR is different with other dimensional reduction methods like Principle Components Analysis in the sense that it works as a supervised learning problem when estimating the projection matrix, while PCA tries to figure out the hidden structure by only look at the x. This property makes SDR suitable in the supervised learning algorithms in which conditional expectation is the focus.

More precisely, in the contest of ABC, given observation (s, θ), where $s \in \mathbb{R}^m$ are initial summary statistics and $\theta \in \mathbb{R}$ is the parameter to be estimated in a specific ABC application. Assuming that there is a d-dimensional subspace $U \subset \mathbb{R}^d$, $d < m$ such that

$$\theta \perp\!\!\!\perp s \mid B^T s, \tag{3.1}$$

where $B = (\beta_1, ..., \beta_d) \in \mathbb{R}^{m \times d}$ is the orthogonal projection matrix. The columns of B spans U and $B^T B = \mathbf{I}_d$. Condition (3.1) shows that given $B^T s$, θ is independent of the initial summary statistics s. It is then sufficient

to use d dimensional constructed vector $z = B^T s$ as the summary statistics. This subspace U is called *Sufficient Dimensional Reduction (SDR)* space [29] in classical dimensional reduction literature. While there are a tremendous amount of published works about estimating the SDR space, in this paper, we propose to use GKDR in which no strong assumption of marginal distribution or variable type is made. We give a brief review of KDR in the following sections, for further details, we refer to [17] [18] [19].

There is a vast set of literature already published in the field of SDR. In this thesis, we focus on the kernel dimensional reduction which generalizes the linear projection of SDR to incorporate non-linearity by implicitly consider all possible transformations in the estimation of the projecting matrix. Here we give a more detailed introduction to the method as it is the foundation for our latter proposal to the ABC.

3.2 Conditional Mean Embedding in Reproducing Kernel Hilbert Space

The key idea of Kernel-based methods is to implicitly map the original distributions on the training data into innite dimensional feature spaces using kernels, such that subsequent estimations or inferences of distributions can be done in the new feature space [9] [11] [51] [1]. The kernels used in this case are often characteristics, such that the function of the original space and the points in the feature space are one-to-one. It is especially useful when measuring the distance between two functions. If two functions are close in the

feature space, they are also close in the original space. This can be viewed as a generalization of the feature mapping of individual points, as used in classical kernel methods. By mapping probabilities into infinite dimensional feature spaces, we can ultimately capture all the statistical features of arbitrary distributions without explicit compute the features in the infinite dimensional spaces. This change of space is often called kernel trick. With this tool, we are able to avoid working explicitly with the infinite dimensional features, instead developing our algorithms in the forms of Gram matrices. The infinite and implicit nature of the feature spaces provides us with a rich yet ecient framework for handling arbitrary distributions and high-dimensional data. In this section, we give a brief review of kernel methods to estimate the conditional mean.

3.2.1 RKHS and Mean Embedding

A reproducing kernel Hilbert space (RKHS) $\mathbb{H}$ is a Hilbert space where all evaluation functionals are bounded. It is equivalent as specifying that the point evaluation is a continuous linear functional. As discussed above, this means that two functions f and g that are close in norm $||f - g||$ are also close in point-wise $|f(x) - g(x)|$ for all x. This property enables the usage of RKHS in statistical machine learning. It has been extensively used in Support Vector Machines (SVM) in classification problems. Recently, by embedding the distribution into RHKS, the applications of kernel methods have been expanded to many statistical problems like PCA, non-parametric Bayesian, MCMC and causal discovery [47],[46],[20],[27] [42].

RKHS is closely related to the positive-definite functions, which are also called kernel functions. For a set Ω, a positive-definite kernel is a function $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ that is symmetric such that $k(x, y) = k(y, x)$, and the Gram matrix is positive definite:

$$\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

for any $x_1, ..., x_n \in \mathbb{X}$, and $n \in N$. This positive definite kernel defines a RKHS and acts as the reproducing kernel in that space. It is known that a positive-definite kernel is uniquely associated with a Hilbert space consisting of functions such that (1)$k(x, \cdot)$ is in $\mathbb{H}$, (2) the linear hull of $k(x, \cdot)$ is dense, and (3) for any $x \in \mathbb{X}$ and $f \in \mathbb{H}$, $\langle f, k(x,)\rangle = f(x)$, this property is called reproducing property.

The kernel functions are first introduced to machine learning community by replacing an inner product $\langle x, y\rangle$ in space $\mathbb{H}$ where $x, y \in \mathbb{X}$ with a dot product and a feature map $\phi : \mathbb{X} \to \mathbb{H}$ without needing to compute ϕ directly ([23]). This is often called kernel trick. This method can be applied to any learning method where an inner product is contained. One clear advantage of using a feature mapping instead of an inner product is that the feature map introduces a nonlinear transformation into the similarity measure and should be able to capture nonlinear relationships in the data.

The idea of introducing non-linearity into the inner product is further expanded since the popularity of kernel trick enabled methods. It has been revealed that the idea of kernel mean embedding can extend the feature map to the space of probability distributions by representing each distribution as

a function

$$\mu_P(x) := \int_{\mathbb{X}} k(x, \cdot) dP(\mathbb{X}),$$

where k is a positive-definitive function. To understand this embedding, considering a RKHS $\mathbb{H}$ associated with a reproducing kernel K. For any function $f \in \mathbb{H}$, by the reproducing property, $\langle k(x, \cdot), f \rangle = f(x)$. In light of this property, we can view the kernel $k(x, \cdot)$ as a representer of x in $\mathbb{H}$. In addition, if the kernel function k is characteristic, the mapping is injective. This means that the representer function $k(x, \cdot)$ is the unique element of $\mathbb{H}$, implying that $||\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}|| = 0$ if and only if $\mathbb{P} = \mathbb{Q}$, where $\mathbb{P}$ and $\mathbb{Q}$ can be any distributions.

With the mean element defined, it is easy to extend the definition to expectation, covariance and conditional expectation.

3.2.2 Covariance Operator and Conditional Mean Embedding

Let (X, μ_X) and (Y, μ_Y) be measure spaces, and (X, Y) be a random variable on $X \times Y$ with probability distribution P. Let k_X and k_Y be measurable positive definite kernels on X and Y, respectively, with respective RKHS $\mathbb{H}_X$ and $\mathbb{H}_Y$. It is assumed that $E[k_X(X, X)]$ and $E[k_Y(Y, Y)]$ are finite. The (un-centered) cross-covariance operator $C_{YX} : \mathbb{H}_X \to \mathbb{H}_Y$ is defined as the operator such that

$$\langle g, C_{YX} f \rangle = E[f(x)g(Y)] = E[\langle f, \Phi(X) \rangle_{\mathbb{H}_X} \langle \Phi_Y(Y), g \rangle_{\mathbb{H}_Y}]$$

holds for all $f \in \mathbb{H}_X, g \in \mathbb{H}_Y$, where $\Phi_X : X \to \mathbb{H}_X$ and $\Phi_Y : Y \to \mathbb{H}_Y$ are defined by $x \to k_X(\cdot, x)$ and $y \to k_Y(\cdot, y)$, respectively. Similarly, C_{XX} denotes the operator on $\mathbb{H}_X$ that satisfies $\langle f_2, C_{XX} f_1 \rangle = E[f_2(X) f_1(X)]$ for any $f_1, f_2 \in \mathbb{H}_X$. These definitions can be viewed as extensions of the covariance matrices, as C_{YX} is the covariance of the random vectors $\Phi_X(X)$ and $\Phi_Y(Y)$ on RKHSs.

We can also write the operators in integral expressions. With $g(y) = k_Y(\cdot, y)$, the reproducing property can be rewritten as

$$(C_{YX} f)(y) = \int k_Y(y, \tilde{y}) f(\tilde{x}) dP(\tilde{x}, \tilde{y})$$

and

$$(C_{XX} f)(x) = \int k_X(x, \tilde{x}) f(\tilde{x}) dP_X(\tilde{x}),$$

where P_X is the marginal distribution of X. These equations show the explicit expressions of C_{YX} and C_{XX} as integral operators.

Empirical estimation of the covariance operators is straightforward with the reproducing property of the RKHSs. Given i.i.d. sample $(X_1, Y_1), ..., (X_n, Y_n)$, the covariance operator is estimated by the empirical covariance operator

$$\tilde{C}_{YX}^{(n)} f = \frac{1}{n} \sum_{i=1}^{n} k_Y(\cdot, Y_i) \langle k_X(\cdot, X_i), f \rangle_{\mathbb{H}_X} = \frac{1}{n} \sum_{i=1}^{n} f(X_i) k_Y(\cdot, Y_i)$$

The estimator $\tilde{C}_{XX}^{(n)}$ can be written accordingly. It is known that these estimators are consistent in the Hilbert-Schmidt norm.

The crucial result that forms the foundation for estimation of conditional

mean embedding is the following result. If $E[g(Y)|X = \cdot] \in \mathbb{H}_X$ holds for g, then

$$C_{XX}E[g(Y)|X = \cdot] = C_{XY}g.$$

If C_{XX} is injective, the above relation can be expressed as

$$E[g(Y)|X = \cdot] = C_{XX}^{-1}C_{XY}g. \tag{3.2}$$

These covariance operators are fundamental elements in building the kernel version methods of the classical statistical methods like PCA and CCA. And in the next section, it will be used to estimate the gradients of the conditional mean embedding.

3.3 Kernel Dimensional Reduction

Let $B = (\beta_1, ..., \beta_d) \in \mathbb{R}^{m\times d}$ be the projection matrix to be estimated, and $z = B^T s$. We assume (3.1) is true and $p(\theta|s) = \tilde{p}(\theta|z)$. The gradient of the regression function is denoted by ∇_s as

$$\nabla_s = \frac{\partial E(\theta|s)}{\partial s} = \frac{\partial E(\theta|z)}{\partial s} = B\frac{\partial E(\theta|z)}{\partial z} \tag{3.3}$$

which shows that the gradients are contained in the SDR space. Given the following estimator $M = E[\nabla_s \nabla_s^T] = BAB^T$, where $A_{ij} = E[E(\theta|\beta_i^T s)E(\theta|\beta_j^T s)]$, $i, j = 1, ..., d$. The projection directions β lie in the subspace spanned by the eigenvectors of M. It is then possible to estimate the projection directions using eigenvalue decomposition. In GKDR, the matrix M is estimated by

the kernel method described below.

Let Ω be an non-empty set, a real valued kernel $k : \Omega \times \Omega \to \mathbb{R}$ is called positive definite if $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$ for any $x_i \in \Omega$ and $c_i \in \mathbb{R}$. Given a positive definite kernel k, there exists a unique reproducing kernel Hilbert space (RKHS) $\mathbb{H}$ associated with it such that: (1)$k(\cdot, x)$ spans $\mathbb{H}$; (2)$\mathbb{H}$ has the *reproducing property* [1]: for all $x \in \Omega$ and $f \in \mathbb{H}$, $\langle f, k(\cdot, x)\rangle = f(x)$.

Given training sample $(s_1, \theta_1), ..., (s_n, \theta_n)$, let $k_S(s_i, s_j) = exp(-||s_i - s_j||^2/\sigma_S^2)$ and $k_\Theta(\theta_i, \theta_j) = exp(-||\theta_i - \theta_j||^2/\sigma_\Theta^2)$ be Gaussian kernels defined on $\mathbb{R}^m$ and $\mathbb{R}$, associated with RKHS $\mathbb{H}_S$ and $\mathbb{H}_\Theta$, respectively. With assumptions of boundedness of the conditional expectation $E(\theta|S = s)$ and the average gradient functional with respect to z, the functional can be estimated using cross-covariance operators defined in RKHS and the consistency of their empirical estimators are guaranteed [19].

To derive the estimator, first assume that, for any $g \in \mathbb{H}_\Theta$ there exists a function $\phi(z)$ on $\mathbb{R}$ such that

$$E(g(\Theta)|S) = \phi_g(B^T S). \tag{3.4}$$

In addition to the assumptions we make above about RKHS $\mathbb{H}_\Theta$ and $\mathbb{H}_\mathbb{S}$, We further make some technical assumptions that

1. $k_S(\tilde{s}, s)$ is continuously differentiable, its gradient lies in the range of C_{SS}.

2. $\phi(z)$ is differentiable with respect to z and the functional $g \to \frac{\partial \phi_g(z)}{\partial z^a}$ is continuous for any $z \in \mathbb{R}$ and $a = 1, ..., d$.

3. $E(k_\Theta(\theta, \Theta)|S = \cdot) \in \mathbb{H}_\mathbb{S}$

With the above assumptions, there exists $\Phi_a(z) \in \mathbb{H}_\mathbb{S}$ such that

$$\langle g, \Phi_a(z) \rangle = \frac{\partial \phi_g(z)}{\partial z^a}$$

$\Phi_a(z)$ is the derivative of $z \to E[k_\Theta(\cdot, \Theta)|S = z]$. By the above assumption of SDR, the derivative of the original function g and can be written as

$$\frac{\partial E[g(\Theta)|S = s]}{\partial s^i} = \frac{\phi_g(B^T s)}{\partial s^i} = \sum_{a=1}^{d} B_{ia} \langle g, \Phi_a(B^T S) \rangle \tag{3.5}$$

holds for any $g \in \mathbb{H}_\Theta$. This expression explicitly shows the relation of the gradient and the projection matrix B. On the other hand, by 3.2 and the assumptions above, there exist $C_{SS}^{-1}(\partial k_S(\cdot, s)/\partial x^i)$ such that for any g

$$\frac{\partial E[g(\Theta)|S = s]}{\partial s^i} = \langle g, C_{\Theta S} C_{SS}^{-1} \frac{\partial k_S(\cdot, s)}{\partial s^i} \rangle \tag{3.6}$$

Using 3.5 and 3.6, we construct a covariance matrix of average gradients as

$$M_{ij} = \langle C_{\Theta S} C_{SS}^{-1} \frac{\partial k_S(\cdot, s)}{\partial s^i}, C_{\Theta S} C_{SS}^{-1} \frac{\partial k_S(\cdot, s)}{\partial s^j} \rangle$$

its empirical estimator can be easily shown as

$$\widehat{M}_n(s_i) = \nabla \mathbf{k}_S(s_i)^T (G_S + n\epsilon_n I_n)^{-1} G_\Theta (G_S + n\epsilon_n I_n)^{-1} \nabla \mathbf{k}_S(s_i) \tag{3.7}$$

where G_S and G_Θ are Gram matrices $k_S(s_i, s_j)$ and $k_\Theta(\theta_i, \theta_j)$, respectively. $\nabla \mathbf{k}_S \in \mathbb{R}^{n \times m}$ is the derivative of the kernel $\mathbf{k}_S(\cdot, s_i)$ with respect to s_i, and ϵ_n

is a regularization coefficient. This matrix can be viewed as the straight forward extension of covariance matrix in principle component analysis (PCA); the data here are the features in RKHS representing the gradients instead of the gradients in their original real space.

The averaged estimator $\tilde{M} = 1/n \sum_{i=1}^{n} \widehat{M}_n(s_i)$ is calculated over the training sample $(s_1, \theta_1), ..., (s_n, \theta_n)$. Finally, the projection matrix B is estimated by taking d eigenvectors corresponding to the d largest eigenvalues of $\tilde{M}$ just like in PCA, where d is the dimension of the estimated subspace.

Chapter 4

Local Kernel Dimensional Reduction

To provide a principled way of designing the regression function, capturing the higher order non-linearity and realizing an automatic construction of summary statistics, we introduce the kernel based sufficient dimensional reduction method as an extension of the linear projection based Semi-automatic ABC. This dimensional reduction method is a localized version of gradient based kernel dimensional reduction (GKDR) [17]. GKDR estimates the projection matrix onto the sufficient subspace by extracting the eigenvectors of the kernel derivatives matrices in the reproducing kernel Hilbert spaces (RKHS). We give a brief review of this method in Section 2. In addition to the GKDR, in which the estimation averages over all data points to reduce variance, a localized GKDR is proposed by averaging over a small neighborhood around the observation in ABC. Each point is weighted using a distance metric measuring the difference between the simulated data and the

observation. The idea is similar to the role of the distance kernel function in (2.1). Another proposal is to use different summary statistics for different parameters. Note that sufficient subspace for different parameters can be different, depending on the particular problem. In these cases, applying separated dimensional reduction procedures yield better estimations of the parameter.

The proposed method gives competitive results in comparison with Semi-automatic ABC[15] when using simple rejection sampling. Substantial improvements are reported in the sequential Monte Carlo cases, where threshold ϵ are pushed to as small as possible to isolate the performance of summary statistics from the Monte Carlo errors.

As discussed above, the estimator $\tilde{M}$ is obtained by averaging over the training sample s_i. When applied to ABC, since only one observation sample is available, we propose to generate a set of training data using the generating model and introduce a weighting mechanism to concentrate on the local region around the observation and avoid regions with low probability density.

Given simulated data $X_1, ..., X_N$ and a weighted kernel $K_w : \mathbb{R}^m \to \mathbb{R}$, we propose the local GKDR estimator

$$\tilde{M} = \frac{1}{N} \sum_{i=1}^{N} K_w(X_i) \widehat{M}(X_i) \tag{4.1}$$

where $\widehat{M}$ is $m \times m$ matrix and $K_w(X_i)$ is the corresponding weight. $K_w(x)$ can be any weighting kernel. In the numerical experiments, a triweight kernel is used, which is written as

$$K_w(X_i) = (1-u^2)^3 \mathbf{1}_{u<1} \quad u = \frac{\|X_i - X_{obs}\|^2}{\|X_{th} - X_{obs}\|^2}$$

where $\mathbf{1}_{u<1}$ is the indicator function, and X_{th} is the threshold value which determines the bandwidth. The normalization term of the triweight kernel is omitted since it does not change the eigenvectors we are estimating. The bandwidth determined by X_{th} is chosen by empirical experiments and will be described in 4.2. The triweight kernel is chosen for its concentration in the central area than other "bell-shaped" kernels and works well in our experiments. Other distance metrics could be used instead of squared distance.

Description of LGKDR algorithm are given in Algorithms 1. Procedure **GenerateSample** is the algorithm to generate sample with parameter as input. Procedure **LGKDR** is the algorithm to calculate matrix $M(X_i)$ as given in (3.7) and (4.1).

Since the dimensional reduction procedure is done before the sampling, it works as a preprocessing unit to the main ABC sampling procedure. It can be embodied in any ABC algorithm using different sampling algorithms. In this paper, the rejection sampling method is firstly employed for its simplicity and low computation complexity as a baseline. Further results on Sequential Monte Carlo ABC are also reported to illustrate the advantage of the purposed method. In these experiments, the distance thresholds are pushed to as small as possible to suppress the Monte Carlo errors and isolate the effects of summary statistics alone.

input : weighting kernel K_w, procedure `GenerateSample`, prior distribution D_{prior}, number of an accepted sample N, process `LGKDR`
output: projection matrix B

training sample generation;
while $i \leq N$ **do**
  draw $\theta_i \leftarrow D_{prior}$;
  $X_i \leftarrow$ `GenerateSample`(θ_i);
  $w(i) \leftarrow K_w(X_i)$;
  if $w \leq 1$ **then**
    $i \leftarrow i + 1$
  end
end
calculate B;
for $j \leftarrow 1$ **to** N **do**
  $M \leftarrow M +$ `LGKDR`$(w(j). * X_j)$
end
$M_{ave} \leftarrow M./N$;
$B \leftarrow eigen(M_{ave})$;

Algorithm 1: LGKDR

input : projection matrix B, distance kernel K_d, bandwidth ϵ, number of sample N_{ABC} and observation X_{ob}
output: set of parameters $\{\theta(j)\}$

$j \leftarrow 1$;
for $i \leftarrow 1$ *to* N_{ABC} **do**
  draw $\theta_i \leftarrow D_{prior}$;
  $X_i \leftarrow$ `GenerateSample`(θ_i);
  if $K_d(B^T X_i, B^T X_{ob}) < \epsilon$ **then**
    $\theta(j) \leftarrow \theta_i$;
    $j \leftarrow j + 1$;
  end
end

Algorithm 2: Rejection ABC

input : projection matrix B, distance kernel K_d, target threshold ϵ_t, number of particle N_{ABC}, effective sample size threshold ess_t
output: set of parameters $\{\theta(j)\}$

for $i \leftarrow 1$ *to* N_{ABC} **do**
  draw $\theta_i \leftarrow D_{prior}$;
  $X_i \leftarrow$ `GenerateSample`(θ_i);
end
$\epsilon \leftarrow Maximum(K_d(B^T X, B^T X_{obs}))$;
while $\epsilon \geq \epsilon_t$ **do**
  decrease ϵ;
  for $i \leftarrow 1$ *to* N_{abc} **do**
    if $K_d(B^T X_i, B^T X_{abc}) \leq \epsilon$ **then**
      $X_{partical} \leftarrow X_i$;
      $\theta_{particle} \leftarrow \theta_i$;
      calculate weight W_i;
    end
  end
  `MoveParticle` $(X_{particle})$;
  if $\sum_{i=0}^{N_{abc}} W_i \leq ess_t$ **then**
    $X \leftarrow$`Resample` $(X_{particle})$;
  end
end

MoveParticle;
for X_j *in* $X_{particle}$ **do**
  $\theta_{new} \leftarrow Normal(\theta_j, std(\theta_{partical}))$;
  $X_j \leftarrow$ `GenerateSample`(θ_{new});
  update weight W_j;
end

Resample **for** $i \leftarrow 1$ *to* N_{abc} **do**
  copy X_i $N_{abc}W_i$ times;
  if $W_i = 0$ **then**
    discard X_i
  end
  re-weight W_i to $1/N_{abc}$
end

Algorithm 3: Sequential-ABC

4.1 Separated Dimensional Reduction

In some problems, not all summary statistics are necessary for every parameter. For example, in the M/G/1 Queue model, the parameter θ_3 that controls the distribution of the inter-arrival time are not related to the parameters θ_1 and θ_2, which jointly determine the distribution of the service time. It can be expected that using different sets of summary statistics for θ_3 with smaller dimensionality would improve the sampling efficiency. To do that, the information that is unrelated to the particular parameter is dropped in the dimensional reduction in exchange of lower dimensionality. The experiments show that better results can be achieved using these settings.

More precisely, LGKDR incorporates information of θ in the calculation of gradient matrix $\tilde{M}$. If θ is a vector, the relation of different elements of θ are contained in the gram matrix G_θ as in (3.7). Separate estimations concentrate on the information of the specific parameter rather than the whole vector. As shown in the experiments in Chapter 5, it can construct significantly more informative summary statistics in some problems by means of reducing estimation error.

For Semi-automatic ABC [15], the summary statistic for each parameter is the estimated posterior mean, thus naturally separated. However, if these one dimensional vectors are used for each parameter separately, the results are not very good. For best subset selection methods [52][38], summary statistics are chosen as the best subset of the original summary statistics using mutual information or sufficiency criterion. It can also be extended to a separated selection procedure. In LGKDR, we simply construct summary

statistics by using only the particular parameter as the response variable.

4.2 Discussion on Hyper Parameters

In this section, we discuss the parameters for LGKDR. Parameters for the ABC sampling will be discussed in the experiments section.

First, the bandwidth of the weighting kernel affects the accuracy of LGKDR. By selecting a large bandwidth, the weights of directions spread out a larger region around the observation points. A small bandwidth concentrates the weights on the directions estimated close to the observation sample. In our experiments, a bandwidth corresponding to an acceptance rate of approximately 10% gives a good result and is used throughout the experiments. The same parameter is set for the Semi-automatic ABC as well for the similar purpose. A more principled method for choosing bandwidth, like cross-validation, could be applied to select the acceptance rate if the corresponding computation complexity is affordable.

The bandwidth of the Gaussian kernels σ_S, σ_Θ and the regularization parameter ϵ_n are crucial to all kernel based methods. The first two determine the function spaces associated with the positive definite kernels and the latter affects the convergence rate (see [45]). In this paper, cross-validation is adapted to select the proper parameters. In the cross-validation, for each set of candidate parameters, the summary statistics are constructed using a simulated observation θ_{obs}, s_{obs}, a training set $(\theta_{training}, S_{training})$ and a test set $(\theta_{test}, S_{test})$. A small pilot run of rejection ABC is performed and the estimation of parameters are calculated by kNN regression of θ_{test} with the

S_{test}. K is set to 5 in all cases. The parameters that yield the smallest least error between the θ_{test} and θ_{obs} are chosen. The final summary statistics are then constructed and passed to the formal run of ABC.

4.3 Computational Complexity

Computational complexity is an important concern of ABC methods. LGKDR requires matrix inversion, solving eigenvalue problems and the cross-validation procedure. In this paper, the training sample size is fixed to 2×10^3 and 10^4 for LGKDR and Semi-automatic ABC, respectively. Under this setting, the total computational time of LGKDR is about 10 times over the linear regression. We believe that it is a necessary price to pay if the non-linearity between the summary statistics are strong. Being unable to capture this information in dimensional reduction step will induce a poor sampling performance and a biased estimation. Also, although the cross-validation procedure takes the majority of computation time in LGKDR, it needs to be performed only once for each problem. Once the parameters are chosen, the computation complexity of LGKDR is comparable to the linear-type algorithms. Overall the computational complexity depends on both the dimensional reduction step and the sampling step. For complex models like population genetics, sampling is significantly more time consuming than the dimensional reduction procedure.

Chapter 5

Experiments

In this section, we investigate three problems to demonstrate the performance of LGKDR. Our method is compared to the classical ABC using initial summary statistics and the Semi-automatic ABC [15] using estimated posterior means. In the first problem, we discuss a population genetics model, which was investigated in many ABC literature. We adopt the initial summary statistics used in [36], and rejection ABC is used as the sampling algorithm. In the second problem, an M/G/1 stochastic queue model which was used in [7] and [15] are discussed. While the model is very simple, the likelihood function could not be trivially computed. In the last experiment, we explore the Ricker model as discussed in [53] and [15]. The latter two problems are investigated by both Rejection ABC and sequential ABC method (SMC ABC) [35], the first problem is omitted from SMC ABC because it involves repeated calling an outside program for simulation and is too time-consuming for SMC ABC.

5.1 Implementation Details

The Rejection ABC is described in Algorithm 2 and the SMC ABC is shown in Algorithm 3. The hyper-parameters used in LGKDR is set as discussed in section 4.2. We use a modified code from [35] and R package "Easyabc" [26] in our SMC implementation and would like to thank the corresponding authors. Gaussian kernels are used in all the LGKDR algorithms. The detailed specifications of Semi-automatic ABC will be described in each experiment.

For evaluation of the experiments conducted using rejection ABC, a set of parameters θ^j where $j \in 1, ..., N_{obs}$ and the corresponding observation sample Y_{obs}^j are simulated from the prior and the conditional probability $p(Y|\theta)$, respectively, and are used as the observations. For each experiment, we fix the total number of simulations N and the number of accepted sample N_{acc}. The sample used for rejection are then generated and fixed for all three methods. Using this setting, although the randomness of the simulation program is contained in the sample, yet the sample used for each method is same and fixed, we can ignore the randomness in the simulation program and compare the methods more fairly. Also, by accurately determine the acceptance rate, which is the most influential parameter for the estimation accuracies. The Mean squared error (MSE) over the accepted parameters $\hat{\theta}_i^j$ and observation θ^j are defined as

$$MSE_j = \frac{1}{N_{acc}} \left(\sum_{i=1}^{N_{acc}} (\theta^j - \hat{\theta}_i^j)^2 \right).$$

The Averaged Mean Square Error (AMSE) is then computed as the average over MSE_j of each observation pair (θ^j, Y_{obs}^j) as

$$AMSE = \frac{1}{N_{obs}} \sum_{j=1}^{N_{obs}} MSE_j. \tag{5.1}$$

It is used as the benchmark for Rejection ABC. Because of the difference of computation complexity, for the fairness of comparison, the acceptance rates are set differently. For LGKDR, the acceptance rate is set to 1%; while for Semi-automatic ABC and original ABC, the acceptance rates are set to 0.1%. The training sample and simulated sample are generated from the same prior and remain fixed.

For SMC ABC, to get to as small tolerance as possible, the simulation time is different for a different method. AMSE is used as the benchmark for the accuracy of the queue model. In the case of the Ricker model, due to the extremely long simulation time, only one observation is used and MSE is used instead in this case. Computation time are reported for both experiments.

5.2 Parameter Settings

Several parameters are necessary for running the simulations in ABC. For Rejection ABC, the total number of samples N and the accepted number of samples N_{acc} are set before the simulation as mentioned above. For Semi-automatic ABC and LGKDR, a training set needs to be simulated to calculate the projection matrix. For LGKDR, a further testing set is also generated for cross-validation purposes. The value of these parameters is reported in the corresponding experiments. The simulation time for generating these sample sets are negligible compared to the main ABC, especially in SMC

ABC. For LGKDR, another important parameter is the target dimensionality d. There are no theoretically sound methods available to determine the intrinsic dimensionality of the initial summary statistics. In practice, since the projection matrix is simply the extracted eigenvectors of the matrix M as in (4.1) ordered by the absolute value of the corresponding eigenvalues, the dimensionality is just the number of the eigenvectors been used. In our experiments, we run several Rejection ABC procedures using different B on a small fixed test set and then fix the dimensionality. Since the test set is fixed and the different projection matrices are directly accessible, this procedure is very fast. A starting point can be set by preserving 70% of the largest eigenvalues in magnitude and it usually works well. There is a large collection of literature on how to choose the number of principal components in PCA, which is similar to our problem, for example, see [50] and reference therein.

5.3 Population Genetics

Analysis of population genetics is often based on the coalescent model[24]. A constant population model is used in simple situations, where the population is assumed unchanged across generations. The parameter of interests, in this case, is the scaled mutation rate θ, which controls the probability of mutation between each generation. The detailed introduction of coalescent models can be found in [37]. Various studies [3] [32] [44] have been conducted in population genetics following different sampling algorithms. In this study, we adopt the setting of kernel ABC [36] and compare the performance with

ABC and Semi-automatic ABC.

100 chromosomes are sampled from a constant population ($N = 10000$). The summary statistics are defined using the spectrum of the numbers of segregating sites, $\mathbf{s}_{sfs}$, which is a coarse-grained spectrum consisting of 7 bins based on the Sturges formula $(1 + log_2 S_{seg})$. The frequencies were binned as follows: $0 - 8\%, 8 - 16\%, 16 - 24\%, 24 - 32\%, 32 - 40\%, 40 - 48\%$ and $48 - 100\%$, we use the uniform distribution $\theta \sim [0, 30]$ in this study rather than the log-normal distribution in [36]. As ABC is often used for exploratory researches, we believe that the performance based on an uninformative prior is important for evaluating summary statistics. The program package ms is used to generate the sample, which is of common choice in the literature of coalescent model [25].

We test 3 typical scaled mutation rates $5, 8$ and 10 rather than random draws from the prior. The results are averaged over 3 tests. A total number of 10^6 sample is generated; 10^5 sample is generated as the training sample for LGKDR and Semi-automatic ABC. Different acceptance rates are set for different methods as discussed above. We use $\mathbf{s}_{sfs}$ as the summary statistics for both Semi-automatic ABC and LGKDR. Local linear regression is used as the regression function for the former. In LGKDR, the dimension is set to 2.

As shown in Table 5.1, the performance of both LGKDR and Semi-automatic ABC improve over original ABC method. LGKDR and Semi-automatic ABC achieve very similar results suggesting that the linear construction of summary statistics are sufficient for this particular experiment.

Table 5.1: AMSE, Coalescent Model.

Method	mutation rate θ
ABC	1.94
Semi-automatic ABC	1.62
LGKDR	1.66

5.4 M/G/1 Queue Model

The M/G/1 model is a stochastic queuing model that follows the first-come-first-serve principle. The arrival of customers follows a Poisson process with intensity parameter λ. The service time for each customer follows an arbitrary distribution with fixed mean (G), and there is a single server (1). This model has an intractable likelihood function because of its iterative nature. However, a simulation model with parameter $(\theta_1, \theta_2, \theta_3)$ can be easily implemented to simulate the model. It has been analyzed by ABC using various different dimensional reduction methods as in [15] and [7], with the comparison to the indirect inference method. We only compare our method with Semi-automatic ABC, since it produces substantially better results than the other methods mentioned above.

The generative model of the M/G/1 model is specified by

$$Y_n = \begin{cases} U_n & \text{if } \sum_{i=1}^{n} W_i \leq \sum_{i=1}^{n-1} Y_i \\ U_n + \sum_{i=1}^{n} W_i - \sum_{i=1}^{n-1} Y_i & \text{if } \sum_{i=1}^{n} W_i > \sum_{i=1}^{n-1} Y_i \end{cases}$$

where Y_n is the inter-departure time between the nth and $n-1$th customer, U_n is the service time for the nth customer, and W_i is the inter-arrival

time between the nth and $n-1$th customer. The process is initialized with $Y_1 = U_1$. The service time is uniformly distributed in interval $[\theta_1, \theta_2]$. The inter-arrival time follows an exponential distribution with rate θ_3. These configurations stay the same as [7] and [15]. We set uninformative uniform priors for $\theta_1, \theta_2 - \theta_1$ and θ_3 as $[1, 10]^2 \times [1, 1/3]$.

For the rejection ABC, we simulate a set of 30 pairs of $(\theta_1, \theta_2, \theta_3)$ but avoid boundary values. They are used as the true parameters to be estimated. The total number of 10^6 samples are generated. The posterior mean is estimated using the empirical mean of the accepted samples. The simulated samples are fixed across different methods for comparison.

we use the quantiles of the sorted inter-departure time Y_n as the exploration variable of the regression model $f(y)$ as in [15]. The powers of the variables are not included as no significant improvements are reported. A pilot ABC procedure is conducted using a fixed training sample set of size 10^4. Local linear regression is used rather than a simple linear regression for better results. For LGKDR, we use the same quantiles as initial summary statistics for dimensional reduction as in Semi-automatic ABC. The number of accepted training sample is 2×10^3 in for the LGKDR. The dimension is manually set to 4, as small as the performance is not degraded.

The experimental results of Rejection ABC are shown in Table-5.2. "LGKDR" refers to the LGKDR that does not use separated estimation. "focus 1" denotes the separated dimensional reduction for parameter θ_1, and the following rows are of similar form. Compared to ABC, "Semi-automatic ABC" gives a substantial improvement on the estimation of θ_1; the other parameters show similar or slightly worse results. LGKDR method improves over ABC

on θ_1 and θ_2, but the estimation of θ_1 is not as good as in Semi-automatic ABC. However, after applying separated estimation, θ_1 presents a substantial improvement compared to Semi-automatic ABC. Separated estimations for θ_2 and θ_3 give no improvements. It suggests that the sufficient dimensional reduction subspace for θ_1 is different from the others and a separated estimation of θ_1 is necessary.

For SMC ABC, a set of 10 pairs of parameters are generated, and the results on SMC and LGKDR are reported. Other settings are the same as the rejection ABC. We omit the results of using Semi-automatic ABC since the sequential chain did not converge properly using these summary statistics and the induced errors were too large to be meaningful. In SMC ABC, two experiments are reported: SMC ABC1 and SMC ABC2. The number of particles is set to 2×10^4 and 10^5, respectively. In LGKDR, the number of particles are set to 2×10^4 and the training sample size for the calculation of projection matrix is 2×10^3, accepted from a training set of size 4×10^4. The dimensionality is set to 5. Cross-validation is conducted using a test set of size 2×10^4.

Results of SMC ABC are shown in Table-5.3. AMSEs are reported. The simulation time is shown as well. The computational time of constructing LGKDR summary statistics is included in the total simulation time and is listed in the bracket. The results show that LGKDR gives better results of parameter θ_1 and θ_2, using less time compared to SMC ABC with set $E2$. The estimation of θ_3 is worse but the difference is small (0.005). Focusing on θ_3 produces an estimation as good as in SMC ABC.

Table 5.2: AMSE, Queue Model, Rejection ABC

Method	θ_1	θ_2	θ_3
ABC	0.2584	0.5113	0.0019
Semi-automatic ABC	0.0112	0.5279	0.0024
LGKDR	0.0623	0.2259	0.0023
LGKDR(focus 1)	0.0082	5.0656	0.0031
LGKDR(focus 2)	0.3942	0.2514	0.0020
LGKDR(focus 3)	0.2229	3.4958	0.0020

Focus means using only that particular parameter as responce variable.

Table 5.3: AMSE, Queue Model, SMC ABC

Method	θ_1	θ_2	θ_3	Total time
SMC ABC 1	0.0404	0.4928	0.0139	9.6e+03
SMC ABC 2	0.0429	0.1964	0.0054	3.3e+04
LGKDR	0.0235	0.1605	0.0110	2.0e+04 (7.78e+3)
LGKDR(focus 3)	0.4854	0.1383	0.0059	2.1e+04 (7.85e+3)

The simulation time of SMC ABC 1 is set to 1e+04 to provide a baseline performance. In SMC ABC 2, the simulation is continued until the bandwidth is no longer changing.

5.5 Ricker Model

Chaotic ecological dynamical systems are difficult for inference due to its dynamic nature and the noises presented in both the observations and the process. Wood [53] addresses this problem using a synthetic likelihood inference method. Fearnhead [15] tackles the same problem with a similar setting using the Semi-automatic ABC and reports a substantial improvement over other methods. In this experiment, we adopt the same setting and apply LGKDR with various configurations.

A prototypic ecological model with Richer map is used as the generating model in this experiment. A time course of a population N_t is described by

$$N_{t+1} = rN_t e^{-N_t + e_t} \tag{5.2}$$

where e_t is the independent noise term with variance σ_e^2, and r is the growth rate parameter controlling the model dynamics. A Poisson observation y is made with mean ϕN_t. The parameters to infer are $\theta = (\log(r), \sigma_e^2, \phi)$. The initial state is $N_0 = 1$ and observations are $y_{51}, y_{52}, \cdots, y_{100}$.

The original summary statistics used by Wood [53] are the observation mean $\bar{y}$, auto-covariances up to lag 5, coefficients of a cubic regression of the ordered difference $y_t - y_{t-1}$ on the observation sample, estimated coefficients for the model $y_{t+1}^{0.3} = \beta_1 y_t^{0.3} + \beta_2 y_t^{0,6} + \epsilon_t$ and the number of zero observations $\sum_{t=51}^{100} \mathbf{1}(y_t = 0)$. This set is denoted as E0 as in [15]. Additional two sets of summary statistics are defined for Semi-automatic ABC. The smaller E1 contains E0 and $\sum_{t=51}^{100} \mathbf{1}(y_t = j)$ for $1 \leq j \leq 4$, logarithm of sample variance, $\log(\sum_{t=51}^{100} y_t^j)$ for $2 \leq j \leq 6$ and auto-correlation to lag 5. Set E2 further

includes time-ordered observation y_t, magnitude-ordered observation $y_{(t)}$, y_t^2, $y_{(t)}^2$, $\{\log(1+y_t)\}$, $\{\log(1+y_{(t)})\}$, time difference Δy_t and magnitude difference $\Delta y_{(t)}$. Additional statistics are added to explicitly explore the non-linear relationships of the original summary statistics and are carefully designed.

In Rejection ABC, we use set E0 for ABC without dimensional reduction since the dimension of the larger sets induces severely decreased performance. Sets E1 and E2 are used for Semi-automatic ABC as in [15]. In LGKDR, we tested sets E0 and E1 in different experiments. The result on E2 is omitted as the result is similar with using the smaller set of statistics, indicating that manually designed non-linear features are unnecessary for LGKDR. The sufficient dimension is set to 5; a smaller value induces substantial worse results. We simulated a set of 30 parameters, a fixed simulated sample of size 10^7 for all the methods and a training sample of size 10^6, a test sample of size 10^5 for LGKDR and Semi-automatic ABC. The values of $\log(r)$ and ϕ are fixed as in [15], and $\log(\sigma_e)$ are drawn from an uninformative uniform distribution on $[log(0.1), 0]$.

The results are shown in Table 5.4. The performance of Semi-automatic ABC using the bigger set E2 is similar to ABC but is substantially worsen with set E1, suggesting that the non-linear information are essential for an accurate estimation in this model. These features are needed to be explicitly designed and incorporated into the regression function for Semi-automatic ABC. LGKDR using summary statistics set E0 gives similar results compared with ABC. Using larger set E1, the accuracy of $\log(r)$ is slightly worse than using set E0, but the accuracy of σ_e and ϕ present substantial improvements. The additional gains of separate constructions of summary statistics

in this model are mixed for a different parameter, $\log(r)$ and ϕ show very small improvements but σ_e gets improvements in both cases. Overall, We recommend using separate constructions for the potential improvements if the additional computational costs are affordable.

In SMC ABC, we use set E0 for the SMC, E1 for LGKDR and both E1 and E2 for Semi-automatic ABC. The number of particles is set to 5×10^3 for all experiments. Other parameters are the same as in Rejection ABC. Only one set of the parameter is used and the time of simulation is set to achieve a tolerance which is as small as possible. Simulation time are reported with a computational time of LGKDR included. We show several results with different settings of dimensionality in LGKDR to illustrate the influence of that hyper-parameter. For LGKDR, we achieve better results on θ_1 and θ_2 with less computation time, especially on θ_1. After focusing on θ_3, we get a comparable result on θ_3 as well with less time. Also should be noticed is the computation time of the LGKDR itself. Since the cross validation for choosing kernel parameters is only to be done once, the computation time should be averaged if multiple run of experiments are done. Another observation is that, if the dimensionality is too high, the efficiency of the SMC chain is decreased; if it is set too low, more bias is induced in the estimated posterior mean suggesting a loss of information in the constructed summary statistics. In this experiment, dimensionality 6 is chosen by counting the number of largest 70% eigenvalues in magnitude as discussed before.

The results are shown in Table-5.5. It shows that the LGKDR can achieve similar results as Semi-automatic ABC using only 1/10 of the simulation time.

Table 5.4: AMSE, Ricker Model, Rejection ABC

Method	$\log(r)$	σ_e	ϕ
ABC(E0)	0.049	0.217	0.944
Semi-automatic ABC(E2)	0.056	0.246	0.936
Semi-automatic ABC(E1)	0.082	0.279	1.387
LGKDR(E0)	0.043	0.241	0.984
LGKDR(E0,focus1)	0.043	0.221	1.221
LGKDR(E0,focus2)	0.068	0.200	1.234
LGKDR(E0,focus3)	0.047	0.211	1.007
LGKDR(E1)	0.047	0.179	0.895
LGKDR(E1,focus1)	0.048	0.220	1.38
LGKDR(E1,focus2)	0.059	0.174	2.694
LGKDR(E1,focus3)	0.054	0.292	0.829

Table 5.5: MSE, Ricker Model, SMC ABC

Method	$\log(r)$	σ_e	ϕ	Total time
ABC(E0)	0.001	0.003	0.430	4.0e+5
Semi-automatic ABC(E2)	0.002	0.020	0.013	4.3e+5
Semi-automatic ABC(E1)	0.031	0.079	0.019	1.7e+5
LGKDR(Dimensional 3)	0.024	0.131	0.779	8.6e+4
LGKDR(Dimensional 6)	0.006	0.018	0.012	4.5e+4
LGKDR(Dimensional 9)	0.001	0.040	0.250	2.8e+5

All experiments are continued until no smaller bandwidth can be reached

Table 5.6: AMSE, Queue Model, Rejection ABC

Method	θ_1	θ_2	θ_3
ABC	0.1512	0.2539	0.0020
LGKDR	0.0764	0.1949	0.0023
GKDR	0.0901	0.2194	0.0023
SIR	0.1961	0.1735	0.0021

5.6 Compare with Other SDR Dimensional Reduction Methods

In this section, we compare the results of LGKDR the other classic SDR dimensional reduction method including sliced inverse regression (SIR) and GKDR without local modifications. The same model of the Queue model from section 5.4 is used in this experiment. The total number of observation is set to 80. The results are averaged over all 80 runs of ABC and are compared using same criterion as 5.1: AMSE. The number of simulations for each run of ABC is fixed to 1×10^6. The acceptance rate is fixed to 0.3 %.

For LGKDR, the parameters of the RKHS kernels are fixed throughout all runs of ABC. The number of training sample used in GKDR and LGKDR are set to 3000. Other settings including the settings of the sampling are the same as section 5.4. For SIR, the number of slices are set to 10, with 3000 samples used as reference data. The output dimension for the all three projection methods are set to 5. The results are shown in Table 5.6.

From the result, we can see that LGKDR consistently outperforms GKDR without localization. It is a reasonable result since the training samples form GKDR are scattered in the whole parameter space and thus the directions

estimated are not as accurate as in local GKDR. For SIR, although the result on θ_2 is best among all methods, the error on θ_1 is even larger than the original ABC, making it difficult to use. All three methods show similar results on θ_3. Overall, LGKDR provides a competitive result without the need to re-adjusting the parameters for each run of sampling.

Chapter 6

Conclusions

In this thesis, we first review the basic idea of ABC and the motivations for using likelihood-free Bayesian methods. Then the basics of kernel methods are briefly reviewed to give a proper understanding of GKDR.

The main contribution of this thesis is the proposal of using LGKDR algorithm for automatically constructing summary statistics in ABC. The proposed method assumes no explicit functional forms of the regression functions nor the marginal distributions, and implicitly incorporates higher order moments up to infinity. As long as the initial summary statistics are sufficient, our method can guarantee to find a sufficient subspace with low dimensionality. While the involved computation is more expensive than the simple linear regression used in Semi-automatic ABC, the dimensional reduction is conducted as the pre-processing step and the cost may not be dominant in comparison with a computationally demanding sampling procedure during ABC. Another advantage of LGKDR is the avoidance of manually designed features; only initial summary statistics are required. With the parameter

selected by the cross-validation, construction of low dimensional summary statistics can be performed as in a black box. For complex models in which the initial summary statistics are hard to identify, LGKDR can be applied directly to the raw data and identify the sufficient subspace. We also confirm that construction of different summary statistics for different parameter improve the accuracy significantly.

Another contribution of the thesis is on the experiments of dimensional reduction method on Sequential-ABC methods. By using Sequential-ABC and keeping the acceptance distance as small as possible, we are able to suppress the influence of the Monte Carlo errors as low as possible, making computations of different dimensional reduction methods more reasonable.

For possible future directions of this work, first, the relationship of concentration rate for the training sample used in LGKDR is only empirically decided. A theoretical analysis may help better understanding this problem. Second, currently, the output dimensional of LGKDR is set based on experiments by looking at the top few largest eigen values of the matrix, a more principled way of deciding the output dimension is a good direction.

Bibliography

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–337, 1950.

[2] M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

[3] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[4] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, Dec. 2002.

[5] G. Bertorelle, A. Benazzo, and S. Mona. ABC as a flexible framework to estimate demography over space and time: some cons, many pros. *Molecular Ecology*, 19(13):2609–2625, July 2010.

[6] M. G. B. Blum. Approximate Bayesian Computation: A Nonparametric Perspective. *Journal of the American Statistical Association*, 105(491):1178–1187, 2010.

[7] M. G. B. Blum and O. Francois. Non-linear regression models for approximate bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

[8] M. G. B. Blum, M. Nunes, D. Prangle, and S. A. Sisson. A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. *Statistical Science*, 28(2):189–208, May 2013.

[9] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schlkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[10] R. D. Cook and L. Ni. Sufficient dimension reduction via inverse regression. *Journal of the American Statistical Association*, 100(470):410–428, 2005.

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[12] K. Csillry, M. G. B. Blum, O. E. Gaggiotti, and O. Franois. Approximate Bayesian computation (ABC) in practice. *Trends in Ecology and Evolution*, 25(7):410–418, 2010.

[13] P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[14] H. H. Fan and L. S. Kubatko. Estimating species trees using approximate bayesian computation. *Molecular Phylogenetics and Evolution*, 59(2):354 – 363, 2011.

[15] P. Fearnhead and D. Prangle. Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74(3):419–474, 2012.

[16] D. T. Frazier, G. M. Martin, C. P. Robert, and J. Rousseau. Asymptotic properties of approximate Bayesian computation. *Biometrika*, 105(3):593–607, 06 2018.

[17] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research*, 5(1):73–99, 2004.

[18] K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

[19] K. Fukumizu and C. Leng. Gradient-Based Kernel Dimension Reduction for Regression. *Journal of the American Statistical Association*, 109(505):359–370, 2014.

[20] K. Fukumizu, L. Song, and A. Gretton. Kernel bayes rule. In *Advances in Neural Information Processing Systems 24*, pages 1737–1745. 2011.

[21] P. J. Green, K. Łatuszyński, M. Pereyra, and C. P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25(4):835–862, Jul 2015.

[22] A. Grelaud, C. P. Robert, J.-M. Marin, F. Rodolphe, and J.-F. Taly. ABC likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis*, 4(2):317–335, June 2009.

[23] T. Hofmann, B. Schlkopf, and A. J. Smola. Kernel methods in machine learning. *Ann. Statist.*, 36(3):1171–1220, 06 2008.

[24] R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics (Oxford, England)*, 18(2):337–338, 2002.

[25] R. R. Hudson. Ms-a Program for Generating Samples Under Neutral Models. *Bioinformatics*, 18(2002):337–338, 2002.

[26] F. Jabot, T. Faure, and N. Dumoulin. Easyabc: performing efficient approximate bayesian computation sampling schemes using r. *Methods in Ecology and Evolution*, 4(7):684–687, 2013.

[27] D. Janzing, E. Sgouritsa, O. Stegle, J. Peters, and B. Schölkopf. Detecting low-complexity unobserved causes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pages 383–391, Arlington, Virginia, United States, 2011. AUAI Press.

[28] P. Joyce and P. Marjoram. Approximately sufficient statistics and bayesian computation. *Statistical applications in genetics and molecular biology*, 7(1):Article26, 2008.

[29] K.-C. Li. Sliced Inverse Regression for Dimension Reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[30] W. Li and P. Fearnhead. On the asymptotic efficiency of approximate Bayesian computation estimators. *Biometrika*, 105(2):285–299, 01 2018.

[31] J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and recent developments in approximate bayesian computation. *Systematic Biology*, 66(1):e66–e82, 2017.

[32] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci. U.S A*, 100(0027-8424):15324–15328, 2003.

[33] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[34] W. S. Moore. Inferring Phylogenies from Mtdna Variation - Mitochondrial-Gene Trees Versus Nuclear-Gene Trees. *Evolution*, 49(4):718–726, 1995.

[35] P. D. Moral, A. Doucet, and A. Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing*, 22(5):1009–1020, Sep 2012.

[36] S. Nakagome, K. Fukumizu, and S. Mano. Kernel approximate Bayesian computation in population genetic inferences. *Statistical Applications in Genetics and Molecular Biology*, 12(6):667–678, 2013.

[37] M. Nordborg. *Coalescent Theory*, pages 843–877. John Wiley & Sons, Ltd, 2008.

[38] M. Nunes and D. J. Balding. On optimal selection of summary statistics for approximate Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article34, 2010.

[39] V. Plagnol and S. Tavaré. Approximate bayesian computation and mcmc. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 99–113, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[40] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, Dec. 1999.

[41] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[42] E. Sgouritsa, D. Janzing, J. Peters, and B. Schölkopf. Identifying finite mixtures of nonparametric product distributions and causal inference of confounders. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 556–575, Arlington, Virginia, United States, 2013. AUAI Press.

[43] S. A. Sisson, Y. Fan, and M. A. Beaumont. Overview of Approximate Bayesian Computation. *ArXiv e-prints,1802.09720*, Feb. 2018.

[44] S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–1765, 2007.

[45] A. J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Netw.*, 11(4):637–649, June 1998.

[46] L. Song, K. Fukumizu, and A. Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, July 2013.

[47] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *International Conference on Machine Learning (ICML 2009)*, June 2009.

[48] 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(3):1511–1520, 2006.

[49] 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(31):187–202, 2009.

[50] S. Valle, W. Li, and S. J. Qin. Selection of the number of principal components:the variance of the reconstruction error criterion with a comparison to other methods. *Industrial & Engineering Chemistry Research*, 38(11):4389–4401, 1999.

[51] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[52] D. Wegmann, C. Leuenberger, and L. Excoffier. Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics*, 182(4):1207–1218, 2009.

[53] S. N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.