entropy

MDPI

Article

AKL-ABC: An Automatic Approximate Bayesian Computation Approach Based on Kernel Learning

Wilson González-Vanegas [1,*], Andrés Álvarez-Meza [2] and José Hernández-Muriel [1] and Álvaro Orozco-Gutiérrez [1]

[1] Automatics Research Group, Universidad Tecnológica de Pereira, Pereira 660003, Colombia; j.hernandez12@utp.edu.co (J.H.-M.); aaog@utp.edu.co (Á.O.-G.)

[2] Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales 170003, Colombia; amalvarezme@unal.edu.co

* Correspondence: wilgonzalez@utp.edu.co

Received: 3 August 2019; Accepted: 19 September 2019; Published: 24 September 2019

Abstract: Bayesian statistical inference under unknown or hard to asses likelihood functions is a very challenging task. Currently, approximate Bayesian computation (ABC) techniques have emerged as a widely used set of likelihood-free methods. A vast number of ABC-based approaches have appeared in the literature; however, they all share a hard dependence on free parameters selection, demanding expensive tuning procedures. In this paper, we introduce an automatic kernel learning-based ABC approach, termed AKL-ABC, to automatically compute posterior estimations from a weighting-based inference. To reach this goal, we propose a kernel learning stage to code similarities between simulation and parameter spaces using a centered kernel alignment (CKA) that is automated via an Information theoretic learning approach. Besides, a local neighborhood selection (LNS) algorithm is used to highlight local dependencies over simulations relying on graph theory. Attained results on synthetic and real-world datasets show our approach is a quite competitive method compared to other non-automatic state-of-the-art ABC techniques.

Keywords: approximate Bayesian computation; graph theory; kernel learning; non-linear dynamic system; statistical inference

1. Introduction

Statistical inference aims to infer a set of model parameters using measured data that comes from a system under a particular scenario [1]. Usually, this kind of task is defiance due to noise present in data after the measurement stage [2]. In this sense, two main general approaches are commonly used in the state-of-the-art to proceed with the inference [3]: (i) The frequentist approach, where inference should give right answers in repeated use under an unconditional perspective that tends to focus more on analysis than on methods (e.g., convergence rates, consistency) [4]; in a frequentist approach, procedures (rule decisions) can come from anywhere, so they do not have to be explicitly derived from a probability model. (ii) The Bayesian approach, where inferences should be made conditioned on the current data under an expert-based perspective that, from prior information about the studied phenomenon, determines a probability density function for the model parameters via the Bayes' theorem [5]; such a density, known as posterior, not only allows model checking and validation, predictive inference, and decision making but also it can tackle point and interval estimation [6].

Bayesian statistical inference tasks require to compute the likelihood function, which states how likely particular values of some statistical parameters are for a given set of observed data [1]. These approaches leverage the inclusion of a priori knowledge about the studied phenomenon into the posterior distribution. Indeed, straightforward models gather an analytic expression for the

likelihood function facilitating the evidence assessment; then, the posterior can be precisely computed. Notwithstanding, for complex models, such as those that assemble high non-linearities or stochastic behavior, the model complexity means that there is no analytical formula for the likelihood function or that it is computationally intractable and can not be evaluated in any practical amount of time, standing for a really challenging scenario to perform statistical inference using Bayesian techniques [7].

Approximate Bayesian computation (ABC) emerged like a free-likelihood method to deal with the intractability mentioned above. It was originally introduced as a solution for performing statistical inference in the field of molecular biology. The first ABC algorithm was proposed to study the demographic history of the Y chromosome [8]. However, the use of ABC techniques has influenced several research areas like systems biology [9], climate analysis [10], ecological modeling [11], nuclear imaging [12], and astronomy [13], just to mention some of them. Fundamentally, an ABC method assesses an auxiliary model with different parameter values drawn from a prior distribution to calculate simulations that are compared to the observed data [14,15]. Mainly, in the face of a large number of features and observations, different authors use statistical parameters to summarize and characterize the data [16–19]. However, the selection of proper and sufficient summary statistics could be difficult for complex models. This fact has led to the need to explore alternative approaches that rely on kernel functions to embed and compare distributions into a reproducing kernel hilbert space (RKHS) [20,21]. Nonetheless, the techniques mentioned above require the estimation of different parameters related to the similarity computation among simulations to approximate the posterior. Then, expensive tuning procedures such as grid search and cross-validation are carried out. Moreover, the user requires a vast knowledge concerning the ABC algorithm and the studied data to properly tune the free parameters, yielding to a high influence in the posterior approximation quality.

In this paper, we introduce an automatic ABC algorithm, termed automatic kernel learning ABC (AKL-ABC), which comprises a kernel learning stage based on a centered kernel alignment (CKA) technique to assess the matching between similarities defined over parameter and simulation spaces in ABC [22]. This paper is an extension of our proposal presented in [23]. Namely, here we provide a series of improvements and contributions:

- We propose a novel automatic ABC approach for computing posterior estimates avoiding any tuning procedure of free parameters. In detail, a Mahalanobis distance is optimized through a CKA-based algorithm to code the simulation and parameter space matching and an information theoretic learning (ITL)-based method to learn the kernel bandwidth. Furthermore, a graph representation is carried out to highlight local dependencies utilizing a local neighborhood selection (LNS).
- The mathematical models regarding AKL-ABC are described and enhanced (including the CKA and LNS coupling with ABC through kernel machines and graph theory).
- The experiments are expanded and explained in detail considering well-known challenging databases.
- A free parameter analysis is provided to show the performance of our AKL-ABC as an automatic approximate inference method.

Achieved results on synthetic and real-world inference problems demonstrate that our AKL-ABC is robust to substantial changes in data dynamics. Namely, the experimental results show that our automatic extension of ABC is competitive with other state-of-the-art works, and has a significant advantage concerning the automatic selection of free parameters. Additionally, a MATLAB implementation of our approach is publicly released.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 introduces the materials and methods and provides the mathematical foundations behind our proposal. Section 4 describes the experimental setup, and the obtained results are discussed in Section 5. Finally, the conclusions are outlined in Section 6.

2. Related Work

Recent progress in ABC-based inference has incorporated different research areas into the principal ABC framework to accomplish more accurate posterior estimations. While several ABC methods can be found across the literature, three main groups can be highlighted according to their main features [24]: Summary statistics-based approaches, weighting-based techniques, and regression-adjustment methods.

2.1. Summary Statistics-Based Approaches

The selection of proper and sufficient summary statistics is a crucial issue in ABC [25]. Wood [18] introduced a synthetic likelihood modeled as a multivariate normal whose mean and covariance are determined from summary statistics. The Gaussian likelihood assumption allows Markov chain Monte Carlo (MCMC) sampling through a rejection kernel. However, the unbiased estimates of the mean and covariance lead to unbiased posterior estimates; moreover, the MCMC sampler requires the tuning of free parameters. Fearnhead and Prangle [16] proved that the optimal summary statistic for inferring the model parameters, under the assumption of the quadratic loss function, is the true posterior mean given the observed data. They used this fact to model such a posterior mean as a linear model that learns summary statistics directly from data, which is further used in standard ABC. Nonetheless, their semi-automatic approach still has the problem of free parameter selection. Indirect inference, traditionally used in the context of maximum likelihood estimation (MLE), has also been introduced in the context of ABC. Gleim and Pigorsch [26] proposed to use a score vector of an auxiliary model fitted via MLE as summary statistics. Then, comparing the score of the model equipped with the observed data against scores fitted with simulations provides an idea of candidates that follows the posterior distribution. Again, this approach focuses on determining summary statistics rather than an overall automatic ABC.

2.2. Weighting-Based Techniques

Comparing summary statistics, to accept or reject candidates depends on thresholds that must be adjusted. An alternative approach introduced by Nakagome et al. [27] stands for a weighting-based inference where all posterior candidates are weighted rather than accepted or rejected. In particular, a conditional mean embedding operator sets up a mapping from summary statistics to model parameters. Such a notion of similarity (assessed in an RKHS) allows setting a probability for the input candidates. While the ABC procedure works in a completely different way, the selection of summary statistics and kernel-free parameters remains. Attempting to avoid the need for summary statistics, Park et al. [20] introduced the maximum mean discrepancy (MMD) criterion to compare probability measures. Using the kernel embedding of distributions into an RKHS, the MMD provides a measure between the distribution of simulated and observed data. It allows assigning a weight to each candidate, using a similarity kernel. The absence of summary statistics comes at the expense of a high computational load. While the authors provided faster approximations for the MMD computation, the need for free parameter selection is not eliminated.

2.3. Regression-Adjustment Methods

Different ABC techniques have been proposed for understanding regression as a fundamental concept in machine learning. Mitrovic et al. [28] modeled the functional relationship between simulations and the optimal choice of summary statistics to encode the structure of a generative model. While their flexible framework regulates the kind and amount of information extracted from data, the optimal construction of summary statistics requires multiple sets of free parameters. So, expensive and problem-dependent tuning procedures are yielded. Regarding the number of particles in ABC, Meeds and Welling [29] developed a surrogate model that works as an artificial likelihood to define an adequate amount of simulations in ABC via Gaussian process-based regression.

Though the number of required simulations to produce posterior estimations is reduced, the significant difficulty resides in the hyper-parameter tuning. Neural networks and deep learning also have been used to model the relationship between parameter values and summary statistics. Jiang et al. [30] interpreted the posterior mean as a summary statistic by connecting the full dataset to the input layer of a deep neural network (DNN). They attempted to use regularization methods for training the neural network but did not obtain significant improvement. Creel [31] also used DNN to find the posterior mean based on a subset of predefined summary statistics rather than using the full dataset. However, the tuning of a large number of free parameters is still an issue in DNN-based ABC.

3. Materials and Methods

In this section, we provide a brief introduction to the ABC fundamentals. First, we introduce the straightforward ABC rejection algorithm and illustrate the usage of kernel methods and Hilbert embedding in the context of ABC. Afterward, we present our automatic ABC approach based on kernel learning and graph theory.

3.1. ABC Fundamentals

The central aim of Bayesian statistical inference concerns the calculation of the posterior distribution $p(\theta|y)$ for a set of model parameters $\theta\in\Theta$ given the observed data $y\in\mathcal{X}$. In particular, the likelihood function $p(y|\theta)$ leverages the previous knowledge, as expressed in the prior distribution $\zeta(\theta)$, into the posterior via Bayes' theorem. However, when the complexity of the analyzed system leads to an intractable likelihood, neither exact nor sampled posterior $p(\theta|y)\propto p(y|\theta)\zeta(\theta)$ can be computed. ABC approaches emerged to facilitate such an inference via simulation of the likelihood through a generative model of the system $\mathcal{M}:\Theta\to\mathcal{X}$ that is statistically related to a conditional probability $p(x|\theta)$, where $x\in\mathcal{X}$ is a random variable standing for the simulated data [14]. Fundamentally, an ABC-based framework relies on the acceptance and rejection of candidates θ using their corresponding simulated samples x based on a distance function $\mathrm{d}_{\mathcal{X}}:\mathcal{X}\times\mathcal{X}\to\mathbb{R}^{+}$. The rejection in ABC is conducted by sampling multiple model parameters from $\zeta(\theta)$. The auxiliary model $\mathcal{M}$ generates simulations x that follow the conditional distribution $p(x|\theta)$; then, the subset $\{\theta : \mathrm{d}_{\mathcal{X}}(x,y)<\varepsilon\}$ (where $\varepsilon\in\mathbb{R}^{+}$ is a threshold) is accepted to follow the posterior distribution. In turn, an approximate posterior can be estimated such that:

$$\hat{p}(\theta|y;\varepsilon) \propto \hat{p}(y|\theta;\varepsilon)\zeta(\theta), \tag{1}$$

where:

$$\hat{p}(y|\theta;\varepsilon)=\int_{\mathscr{B}(y;\varepsilon)} p(x|\theta)dx, \quad \mathscr{B}(y;\varepsilon) = \{x : \mathrm{d}_{\mathcal{X}}(x,y)<\varepsilon\}. \tag{2}$$

Notice in Equation (2) how an accurate posterior relies upon an appropriate distance $\mathrm{d}_{\mathcal{X}}$ and a suitable ε-value. Still, it is often challenging to apply a distance directly on $\mathcal{X}$ when dealing with real data since it is commonly formed by a large number of observations and features. In such a case, some alternatives use a mapping $s=\vartheta(x)$ before calculating the distance, where $s\in\mathcal{S}$ is a feature space and $\vartheta:\mathcal{X}\to\mathcal{S}$ [17]. The previous setting is widely known as the straightforward ABC rejection algorithm.

Nonetheless, the use of $\vartheta(x)$ can lead to a non-sufficient feature space leaking information for complex models. As a consequence, some ABC-based inference approaches approximate the posterior $\hat{p}(y|\theta;\epsilon)$ as the convolution of the true likelihood $p(y|\theta)$ and a kernel function $\kappa:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$, which imposes a constraint to the rejection of samples as the inner product $\kappa(x,y)=\langle\phi(x),\phi(y)\rangle_{\mathcal{H}}$ in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, where $\phi:\mathcal{X}\to\mathcal{H}$ [20]. In practice, given N samples $\{x_n\sim P_{X_n}\}_{n=1}^{N}$ drawn from $p(x|\theta_n)$, with $\theta_n\sim\zeta(\theta)$, and observed data $y\sim P_Y$, a weighted sample set $W=\{\theta_n, w_n\}_{n=1}^{N}$ is calculated by fixing:

$$w_n = \frac{\kappa_G\left(\mathrm{d}_{\mathcal{H}}(P_{X_n}, P_Y);\epsilon\right)}{\sum_{n=1}^{N}\kappa_G\left(\mathrm{d}_{\mathcal{H}}(P_{X_n}, P_Y);\epsilon\right)}, \tag{3}$$

where $\kappa_G(\mathrm{d}_{\mathcal{H}}(\cdot,\cdot)\,;\epsilon)$ is a Gaussian kernel defined as:

$$\kappa_G\left(\mathrm{d}_{\mathcal{H}}(P_{X_n},P_Y);\epsilon\right)=\exp\left(\frac{-\mathrm{d}^2_{\mathcal{H}}(P_{X_n},P_Y)}{2\epsilon^2}\right), \tag{4}$$

where $\epsilon\in\mathbb{R}^+$ is the kernel bandwidth and $\mathrm{d}_{\mathcal{H}}:\mathcal{H}\mathcal{H}\to\mathbb{R}^+$ represents a distance over the Hilbert embedding-based mappings between the distributions P_{X_n} and P_Y. Figure 1 displays the main pipeline of the ABC rejection and Hilbert embedding-based ABC approaches, respectively. Finally, the set W is found as in Equation (3) to approximate $p(\theta|y)$ via the weighting-based posterior expectation as:

$$\hat{p}(\theta|y)=\sum_{n=1}^{N} w_n\kappa_G(\mathrm{d}_e(\theta,\theta_n);\sigma_\theta), \tag{5}$$

where $\mathrm{d}_e(\cdot,\cdot)$ stands for the Euclidean distance and $\sigma_\theta\in\mathbb{R}^+$ is the kernel width.

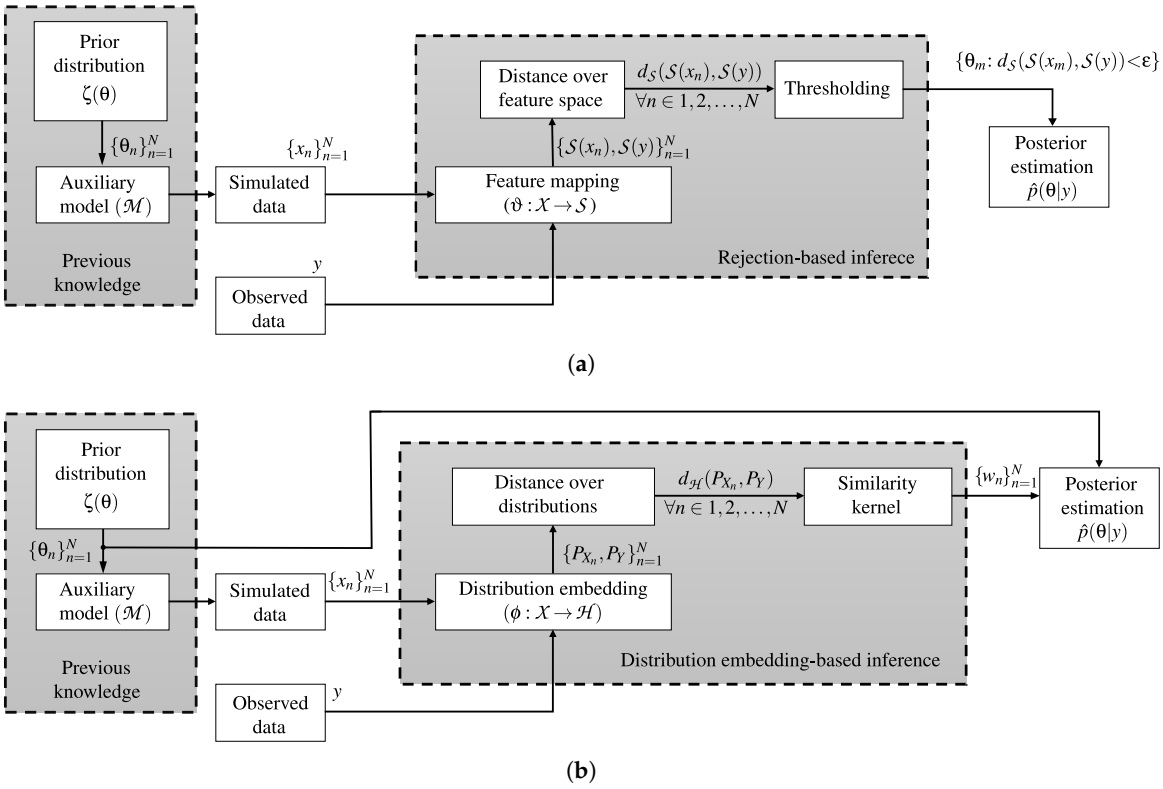


Figure 1. ABC main pipelines. (**a**) ABC rejection algorithm. (**b**) Hilbert embedding-based ABC approach.

3.2. Automatic ABC Based on Kernel Learning

Traditionally, the ABC methods do not code the information of the parameter space Θ towards the inference. Namely, the deduction of the model parameters is performed using only explicit information from simulations and observations (see how in Figure 1 there is no direct connection between $\{\theta_n\}_{n=1}^N$ and the inference stage). Consequently, we introduce a novel kernel learning-based ABC approach, termed automatic kernel learning ABC (AKL-ABC), that codes the similarities between candidates in Θ into the inference stage to obtain an automatic version of ABC. In particular, AKL-ABC comprises a kernel learning stage based on a statistical alignment to assess the matching between similarities defined over parameters and simulations. Moreover, a local neighborhood selection (LNS) algorithm is utilized to highlight local dependencies over candidates in Θ based on graph theory to enhance the kernel similarities through a pruning scheme. The main steps of our AKL-ABC are summarized in the diagram shown in Figure 2 and described below.

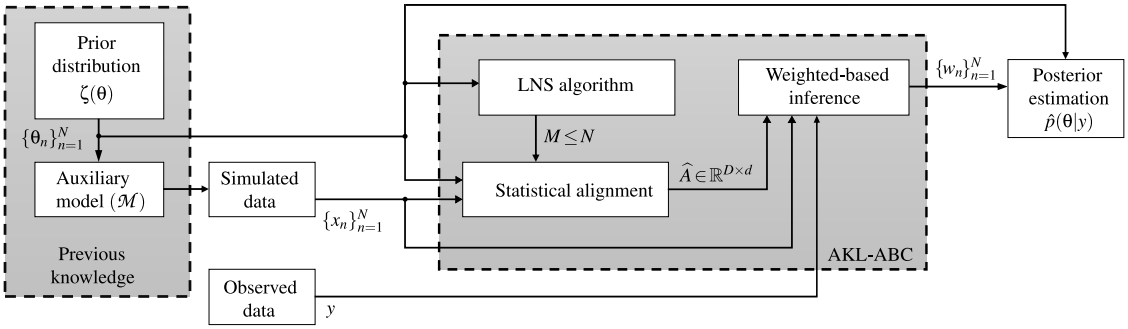


Figure 2. Sketch for the proposed AKL-ABC approach.

3.2.1. Kernel Learning in the Context of ABC

The estimation of the ABC-based posterior as in Equations (3) and (5) requires tuning of the ϵ value. To avoid its tuning, we introduce a statistical alignment approach for enhancing the inference task. The purpose behind this procedure is to include the information contained in the candidates $\{\theta_n\}_{n=1}^N$ to improve the comparison stage carried out over simulations and observations. Let $\Psi=\{\theta_n, x_n\}_{n=1}^N$ be the set of N candidates $\theta_n \in \mathbb{R}^P \sim \zeta(\theta)$ and their corresponding simulations $x_n \in \mathbb{R}^Q \sim p(x|\theta)$. Further, let $\kappa_\theta : \Theta \times \Theta \to \mathbb{R}^+$ be a similarity measure between candidates in Θ that defines the kernel matrix $\mathbf{K}_{\boldsymbol{\theta}} \in \mathbb{R}^{N\times N}$ holding elements:

$$k_\theta(\theta_n, \theta_{n'}) = \begin{cases} \exp(-\mathrm{d}^2_\Theta(\theta_n, \theta_{n'})) & \theta_n \in \Omega_{n'} \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\Omega_{n'}$ is a set holding the M-nearest neighbors of $\theta_{n'}$ in the sense of the distance $\mathrm{d}_\Theta : \Theta\times\Theta \to \mathbb{R}^+$. In this paper, to avoid large variations among components of θ_n we rely on the Mahalanobis distance as follows:

$$\mathrm{d}^2_\Theta(\theta_n, \theta_{n'}) = (\theta_n - \theta_{n'})^\top \boldsymbol{\Sigma}_\Theta^{-1} (\theta_n - \theta_{n'}) \tag{7}$$

where $\boldsymbol{\Sigma}_\Theta \in \mathbb{R}^{P\times P}$ is the sample covariance matrix. Concerning the feature space $\mathcal{S}$, we assess the similarity via the Gaussian kernel function $k_s : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$, $k_s(\vartheta(x_n), \vartheta(x_{n'})) = \exp(-\mathrm{d}^2_\mathcal{S}(\vartheta(x_n), \vartheta(x_{n'}))/2\gamma^2)$, to build the matrix $\mathbf{K}_s \in \mathbb{R}^{N\times N}$, where $\mathrm{d}^2_\mathcal{S} : \mathcal{S}\mathcal{S} \to \mathbb{R}^+$ and $\vartheta : \mathcal{X} \to \mathcal{S}$ is a feature mapping. Here, to perform the pairwise comparison between simulations in $\mathcal{S}$ we use also the Mahalanobis distance of the form [32]:

$$\mathrm{d}^2_\mathcal{S}(\vartheta(x_n), \vartheta(x_{n'})) = (\vartheta(x_n) - \vartheta(x_{n'}))^\top \boldsymbol{A}\boldsymbol{A}^\top (\vartheta(x_n) - \vartheta(x_{n'})), \tag{8}$$

where $\boldsymbol{\Sigma}_\mathcal{S}^{-1} = \boldsymbol{A}\boldsymbol{A}^\top$ stands for the inverse covariance matrix of $\vartheta(x_n) \in \mathbb{R}^D$ and $\boldsymbol{A} \in \mathbb{R}^{D\times d}$ ($d \leq D$). In this sense, we use the information respecting the similarities among candidates in Θ, represented via $\mathbf{K}_{\boldsymbol{\theta}}$, to state a notion of similarity between the simulations and a target observation in $\mathcal{S}$, represented via $\mathbf{K}_s(\boldsymbol{A}, \gamma)$. Therefore, we use a CKA-based measure between the kernel matrices as follows [22]:

$$\hat{\rho}(\mathbf{K}_{\boldsymbol{\theta}}, \mathbf{K}_s(\boldsymbol{A}, \gamma)) = \frac{\langle \overline{\mathbf{K}}_{\boldsymbol{\theta}}, \overline{\mathbf{K}}_s \rangle_\mathrm{F}}{\sqrt{\langle \overline{\mathbf{K}}_{\boldsymbol{\theta}}, \overline{\mathbf{K}}_{\boldsymbol{\theta}} \rangle_\mathrm{F} \langle \overline{\mathbf{K}}_s, \overline{\mathbf{K}}_s \rangle_\mathrm{F}}}, \tag{9}$$

where $\overline{\mathbf{K}}$ stands for the centered kernel as $\overline{\mathbf{K}} = \tilde{\boldsymbol{I}}\mathbf{K}\tilde{\boldsymbol{I}}$, being $\tilde{\boldsymbol{I}} = \boldsymbol{I} - \mathbf{1}^\top\mathbf{1}/N$ the empirical centering matrix, $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ is the identity matrix, and $\mathbf{1} \in \mathbb{R}^N$ is the all-ones vector. Moreover, The notation $\langle \cdot, \cdot \rangle_\mathrm{F}$ represents the matrix-based Frobenius norm. In Equation (9), $\hat{\rho}(\cdot, \cdot)$ is a data-driven estimator that aims to quantify the similarity between the parameter space and the feature space.

To find the projection matrix $\boldsymbol{A}$, we consider the following optimization problem:

$$\hat{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}, \gamma} \quad \log\left(\hat{\rho}(\mathbf{K}_{\boldsymbol{\theta}}, \mathbf{K}_s(\boldsymbol{A}, \gamma))\right), \tag{10}$$

where the logarithm function is used for mathematical convenience. Estimation of $\hat{\rho}$ in Equation (10) relies on the explicit objective [33]:

$$\hat{\rho}(\mathbf{K}_{\boldsymbol{\theta}}, \mathbf{K}_{\boldsymbol{s}}(\boldsymbol{A}, \gamma)) = \log\left(\operatorname{tr}\left(\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{\theta}}\tilde{\boldsymbol{I}}\right)\right) - 0.5\log\left(\operatorname{tr}\left(\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\right)\right) + \rho_0, \tag{11}$$

where $\rho_0 \in \mathbb{R}$ is a constant in $\boldsymbol{A}$. In this regard, given an initial guess $\boldsymbol{A}^0$ (calculated, for instance, using the well-known principal component analysis (PCA) algorithm), the projection matrix $\boldsymbol{A}$ is updated according to the following gradient-descent rule:

$$\boldsymbol{A}^{t+1} = \boldsymbol{A}^t - \mu_{\boldsymbol{A}}^t \nabla_{\boldsymbol{A}}\left[\hat{\rho}(\mathbf{K}_{\boldsymbol{\theta}}, \mathbf{K}_{\boldsymbol{s}}(\boldsymbol{A}^t, \gamma^t))\right], \tag{12}$$

where $\mu_{\boldsymbol{A}}^t \in \mathbb{R}^+$ is the step size of the learning rule and $\nabla_{\boldsymbol{A}}\left[\hat{\rho}(\cdot,\cdot)\right]$ stands for the gradient with respect to $\boldsymbol{A}$ of the objective function (11), defined as follows:

$$\nabla_{\boldsymbol{A}}\left[\hat{\rho}(\mathbf{K}_{\boldsymbol{\theta}}, \mathbf{K}_{\boldsymbol{s}}(\boldsymbol{A}^t, \gamma^t))\right] = -4\boldsymbol{V}^\top(\Delta \circ \mathbf{K}_{\boldsymbol{s}}) - \operatorname{diag}\left(\mathbf{1}^\top(\Delta \circ \mathbf{K}_{\boldsymbol{s}})\right)\boldsymbol{V}\boldsymbol{A}^t, \tag{13}$$

where $\boldsymbol{V} \in \mathbb{R}^{N \times D}$ is a matrix whose n-th row holds the mapped simulation $\vartheta(x_n)$, and the notations $\operatorname{diag}(\cdot)$ and $\circ$ denote the diagonal operator and the Hadamard product, respectively. Moreover, $\Delta \in \mathbb{R}^{N \times N}$ is the gradient of the objective function with respect to $\mathbf{K}_{\boldsymbol{s}}$:

$$\Delta = \frac{\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{\theta}}\tilde{\boldsymbol{I}}}{\operatorname{tr}\left(\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{\theta}}\tilde{\boldsymbol{I}}\right)} - \frac{\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}}{\operatorname{tr}\left(\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\mathbf{K}_{\boldsymbol{s}}\tilde{\boldsymbol{I}}\right)}. \tag{14}$$

In addition to the optimal learning of the matrix $\boldsymbol{A}$, the optimization problem in Equation (11) also requires the tuning of the Gaussian kernel bandwidth γ. This joint parameter estimation can be carried out optimizing one variable recursively at a time while the other variable remains unchanged. Namely, the calculation of the rule in Equation (12) is achieved under a constant γ^t-value. In turn, this kernel bandwidth is estimated (under constant $\hat{\boldsymbol{A}}$) using information theoretic learning (ITL) criteria intending to maximize the overall variability of the information potential (H_α) computed for all $\{z_n = \vartheta(x_n)^\top \hat{\boldsymbol{A}}\}_{n=1}^N$, so that all the information force magnitudes spread more widely [34]:

$$\gamma^t = \arg\max_{\gamma} \quad \operatorname{var}\{H_\alpha(z_n|\gamma^t) : n = 1, 2, \ldots, N\}; \quad \alpha \in \mathbb{R}^+. \tag{15}$$

In particular, we use the Renyi's quadratic entropy, $H_2(\cdot|\gamma^t)$, to apply the following gradient-descent update rule over the Gaussian kernel bandwidth:

$$\gamma^{t+1} = \gamma^t - \mu_\gamma^t \nabla_\gamma\left[\operatorname{var}\left\{H_2(z_n^t|\gamma^t)\right\}\right], \tag{16}$$

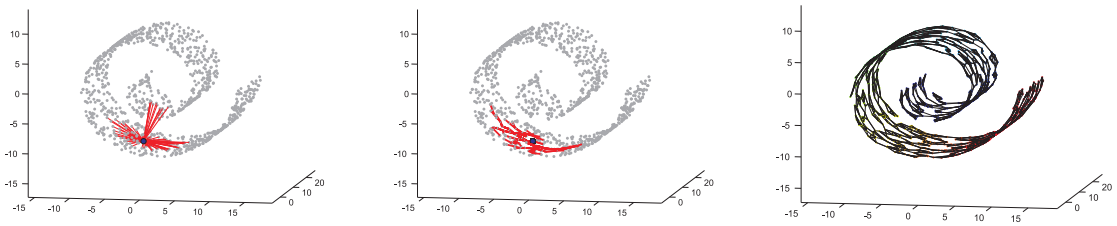
where $\mu_\gamma^t \in \mathbb{R}^+$ is the step size of the learning rule. See [34] for more details on the derivation of ∇_γ .

3.2.2. ϵ Tuning Through Nearest Neighbors Based on Graph Theory

Tuning the ϵ-value to approximate the posterior weights as in Equation (3) is a critical step. Depending on the distance output values, a particular choice of ϵ would produce a peaked posterior when just a few numbers of weights have larger values or lead to a posterior similar to the prior distribution in the limit condition when $w_n \to 1/N, \forall n = \{1, 2, \ldots, N\}$. Bearing this in mind, we use the truncated representation of Equation (6) as an alternative to avoid the influence of ϵ via the concept of neighborhood. Namely, an optimal selection of the number of nearest neighbors $M \in \mathbb{N}$ reveals the prior representative samples in the posterior distribution.

The M-value can be fixed manually after an exhaustive search based on cross-validation; however, that would hinder the automatic philosophy of this work. To avoid this issue, we use an automatic technique based on locally linear embedding (LLE) and graph theory, the local neighborhood selection

(LNS) algorithm, to facilitate the selection of the optimal number of nearest neighbors [35]. LNS aims to identify a suitable number of neighbors for each sample taking into account the structure of the dataset. Specifically, this algorithm is rooted in the idea that when a region around a point is linear and dense, the Euclidean and Geodesic distances obtain a similar set of nearest neighbors for each sample; otherwise, the Euclidean distance will detect short connections while the geodesic distance will identify the right neighbors of each sample. For a better illustration, Figure 3 shows the nearest neighbors for a particular sample in the well-known Swiss-Roll manifold (filled bullet) using both the Euclidean and the geodesic distances. Notice how the Euclidean distance selects neighbors that do not follow the structure of the manifold (Figure 3a) while the geodesic distance understands the actual structure leading to a proper selection of the nearest neighbors (Figure 3b). Figure 3c shows the completed connected graph for all points in the dataset. The LNS algorithm devoted to the estimation of the ABC posterior is described in detail in Appendix A.



(**a**) (**b**) (**c**)

Figure 3. Example of the neighborhoods found by means of Euclidean and geodesic distances. (**a**) Euclidean distance. (**b**) Geodesic distance. (**c**) Completed graph after fixing the number of neighbors using LNS.

Lastly, the ABC-based inference stage is automated via the weighted sample set $\{(\theta_n, w_n)\}_{n=1}^{N}$, where each w_n is calculated as follows:

$$w_n = \frac{\kappa_E(z, z_n)}{\sum_{n=1}^{N} \kappa_E(z, z_n)}, \tag{17}$$

where $\kappa_E : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a similarity kernel defined as:

$$\kappa_E(z, z_n) = \begin{cases} \exp(-||z - z_n||_2^2) & z_n \in \mathscr{Z} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathscr{Z}$ is a set holding the M-nearest neighbors of $z = \vartheta(y)^\top \hat{\boldsymbol{A}}$ in the sense of the Euclidean distance.

In short, Algorithm 1 shows the proposed AKL-ABC approach. Furthermore, a MATLAB implementation of the AKL-ABC is publicly available (https://github.com/WilsonGV/AKL-ABC.git).

Algorithm 1 AKL-ABC algorithm

Input: Observed data: y, prior: $\zeta(\theta)$, mapping: ϑ.
Output: Posterior estimation: $\hat{p}(\theta|y)$.
Kernel learning stage:
$\Psi^{'}=\{(\theta'_n, x'_n)\}_{n=1}^{N}$; $\theta'_n \sim \zeta(\theta)$, $x'_n \sim p(x|\theta'_n)$ ▷ Draw training data.
$M = \text{median}\{\mathbb{M}\}$ ▷ Highlight local dependencies in Θ using LNS.
$\hat{\boldsymbol{A}} = \arg\max_{\boldsymbol{A}} \log\left(\hat{\rho}(\mathbf{K}_{\boldsymbol{s}}(\boldsymbol{A}), \mathbf{K}_{\boldsymbol{\theta}})\right)$ ▷ Compute the CKA based on ϑ, M, and Ψ'.
Inference stage:
$\Psi=\{(\theta_n, x_n)\}_{n=1}^{N}$; $\theta_n \sim \zeta(\theta)$, $x_n \sim p(x|\theta_n)$ ▷ Draw simulated data.
$z=\vartheta(y)^{\mathrm{T}}\hat{\boldsymbol{A}}$ ▷ Project features of observed data
for $n = 1, \cdots, N$ **do**
    $z_n=\vartheta(x_n)^{\mathrm{T}}\hat{\boldsymbol{A}}$ ▷ Project features of simulated data
    $\tilde{w}_n = \kappa_E(z, z_n)$ ▷ Compute the n-th weight value.
end for
$w_n = \tilde{w}_n / \sum_{n=1}^{N} \tilde{w}_n$ ▷ Normalize the weights
$\hat{p}(\theta|y)=\sum_{n=1}^{N} w_n \kappa_G(\mathrm{d}_e(\theta, \theta_n); \sigma_\theta)$ ▷ Approximate the posterior.

3.3. Theoretical Aspects of AKL-ABC

One of the main contributions of AKL-ABC is the computation of automatic posterior estimates with no need for expensive procedures to select free parameters. We now address two major theoretical aspects of AKL-ABC: Learning performance and computational complexity.

To investigate the learning performance of AKL-ABC, we must analyze the CKA-based measure over matrices $\mathbf{K}_{\boldsymbol{s}}\in\mathbb{R}^{N\times N}$ and $\mathbf{K}_{\boldsymbol{\theta}}\in\mathbb{R}^{N\times N}$. The empirical alignment $\hat{\rho}(\mathbf{K}_{\boldsymbol{s}}, \mathbf{K}_{\boldsymbol{\theta}})$, as defined in Equation (9), is a statistical approximation of the true alignment $\rho(k_s, k_\theta)$ defined as [36]:

$$\rho(k_s, k_\theta) = \frac{\langle k_s, k_\theta \rangle_{\mathcal{P}}}{\langle k_s, k_s \rangle_{\mathcal{P}} \langle k_\theta, k_\theta \rangle_{\mathcal{P}}}, \tag{19}$$

where $\langle f, g \rangle_{\mathcal{P}}=\int f(a,b)g(a,b)d\mathcal{P}(a)d\mathcal{P}(b)$ based on the input distribution $\mathcal{P}$.

Theorem 1. *(Cortes et al. [22]) Assume $k_s(\cdot,\cdot)\leq\varpi$ and $k_\theta(\cdot,\cdot)\leq\varpi'$. For any $\varsigma > 0$, with probability at least $1-\varsigma$, the following inequality holds:*

$$|\rho(k_s, k_\theta) - \hat{\rho}(\boldsymbol{K}_{\boldsymbol{s}}, \boldsymbol{K}_{\boldsymbol{\theta}})| \leq 18\Lambda \left[\frac{3}{N} + 8\sqrt{\frac{\log \frac{6}{\varsigma}}{2N}} \right],$$

where $\Lambda=\max(\varpi\varpi'/\mathbb{E}[k_s^2],\ \varpi\varpi'/\mathbb{E}[k_\theta^2])$.

Proposition 1. *Given the sample set $\{(\theta_n, w_n)\}_{n=1}^{N}$, where the weights w_n are fixed according to the AKL-ABC algorithm, the expectation $\mathbb{E}[\hat{p}(\theta|y)]=\sum_{n=1}^{N} w_n\theta_n$ tends to the expected value of the posterior distribution as $N\to\infty$.*

Proof. According to Theorem 1, the empirical CKA-based alignment used in AKL-ABC ($\hat{\rho}(\mathbf{K}_{\boldsymbol{s}}, \mathbf{K}_{\boldsymbol{\theta}})$) tends asymptotically to the true statistical alignment ($\rho(k_s, k_\theta)$) as the number of input points tend to infinity. As a consequence, $\mathbf{K}_{\boldsymbol{s}}$–$\mathbf{K}_{\boldsymbol{\theta}}$ tends to the null matrix as $N\to\infty$ meaning that the notion of similarity defined over simulations and parameters spaces is the same. Then, the neighborhood around projected features of observed data $z=\vartheta(y)^{\top}\hat{\boldsymbol{A}}$, according to Equations (17) and (18), leads to a neighborhood around the true value of the model parameters in the sense of the statistical input distribution $\mathcal{P}$. □

Remark 1. *While the LNS algorithm operates over the input set* $\{\theta_n\}_{n=1}^{N}$ *(see Appendix A), the concept of neighborhood is directly transferred from the parameter space to the simulations space. Namely, the similarity assessment is exchangeable from one space to another due to the asymptotic behavior of the statistical alignment with the number of input points.*

Finally, taking into account the main steps of AKL-ABC, we can find its computational complexity: The LNS algorithm requires $O(PN^2+\varrho N^2)$ operations, where $\varrho \leq \lceil \sqrt{N} \rceil$ takes a dynamic integer value since the LNS method determines the number of neighbors by adapting the dataset distribution (see Appendix A) [35]; furthermore, a number of $O(N^2+GN^2)$ operations is required to find the projection matrix $\hat{\boldsymbol{A}}$ based on $\mathbf{K}_{\boldsymbol{s}}$ and $\mathbf{K}_{\boldsymbol{\theta}}$, where G is the number of desired gradient-descend steps. Then, the total complexity of our AKL-ABC is $O(N^2(P+\varrho+G+1))$.

4. Experimental Setup

To evaluate the performance of our AKL-ABC approach, we analyze three different attributes: (i) Posterior quality approximation in applications that comprise both synthetic and real datasets, (ii) suitable convergence concerning the number of ABC samples (N), and (iii) adequate selection of the number of nearest neighbors (M).

4.1. Datasets and Quality Assessment

We examine the next experiments following [20]: A toy problem concerning synthetic data from a mixture model and a Bayesian inference problem for a real-world ecological dynamic system. Each experiment is described below.

Toy problem: We inspect a mixture of uniform distributions of the form:

$$p(x|\boldsymbol{\pi}) = \sum_{c=1}^{C} \pi_c \mathscr{U}(c-1, c), \tag{20}$$

where $\boldsymbol{\pi} = \{\pi_c \in \mathbb{R}^+\}_{c=1}^{C}$ stores the mixing coefficients holding $\sum_{c=1}^{C} \pi_c = 1$, and $C \in \mathbb{N}$ is the number of components. Here, the aim is to estimate the posterior $p(\boldsymbol{\pi}|y)$ for $C=5$, given synthetic observations y drawn from the mixture with true parameters (target): $\boldsymbol{\pi}^* = \{0.25, 0.04, 0.33, 0.04, 0.34\}$. Besides, as quality assessment for the toy problem we utilize the Euclidean distance as follows [20]:

$$\mathcal{E} = ||\boldsymbol{\pi}^* - \hat{\boldsymbol{\pi}}||_2, \tag{21}$$

where $\hat{\boldsymbol{\pi}}$ is the expected value of the posterior found using the weights $\{w_n\}_{n=1}^{N}$.

Real-world problem: Inference tasks over dynamic ecological systems representing chaotic and near-chaotic domains is quite a challenge [7,9]. We considered the problem of inferring the dynamics of an adult blowfly population. Particularly, Wood [18] introduced a model for describing population dynamics using a differential equation as follows:

$$N_{t+1} = PN_{t-\tau} \exp\left(-\frac{N_{t-\tau}}{N_0}\right) e_t + N_t \exp(-\delta \epsilon_t), \tag{22}$$

where N_{t+1} denotes the observation time at $t+1$, which is determined by the time-lagged observations N_t and $N_{t-\tau}$. Moreover, e_t and ϵ_t stand for the Gamma distributed noise as $e_t \sim \mathcal{G}(1/\sigma_p^2, \sigma_p^2)$ and $\epsilon_t \sim \mathcal{G}(1/\sigma_d^2, \sigma_d^2)$, respectively. Here, our aim is to estimate the posterior of the parameters $\boldsymbol{\theta} = \{P, N_0, \sigma_d, \sigma_p, \tau, \delta\}$ given observed data concerning a time series of 180 observations (available in the Supplementary Materials of [18]). For concrete testing, we adopt log-normal distributions for setting priors over $\boldsymbol{\theta}$ [29]: $\log(P) \sim \mathcal{N}(2, 2^2)$, $\log(N_0) \sim \mathcal{N}(6, 1)$, $\log(\sigma_d) \sim \mathcal{N}(-0.5, 1)$, $\log(\sigma_p) \sim \mathcal{N}(-0.5, 1)$, $\log(\tau) \sim \mathcal{N}(2.7, 1)$, $\log(\delta) \sim \mathcal{N}(-1, 0.4^2)$.

On the other hand, because in a real-world inference task there is no target value for the model parameters, the quantitative assessment must rely on the quality of predictions. Thus, the Euclidean distance is computed to measure the performance in the feature space as follows [20]:

$$\mathscr{E} = ||\vartheta(y) - \vartheta(x_n|\hat{\boldsymbol{\theta}})||_2, \tag{23}$$

where $x_n|\hat{\boldsymbol{\theta}}$ is a simulation from the model given the expected value of the posterior.

4.2. AKL-ABC Training and Method Comparison

For comparison purposes, we considered the K2-ABC approach proposed by Park et al. [20] due to its excellent performance over other methods such as: Indirect score ABC (IS-ABC) [19], semi-automatic ABC (SA-ABC) [16], kernel ABC (K-ABC) [27], and synthetic likelihood ABC (SL-ABC) [18]. We selected the previous benchmark since they belong to the two main groups of ABC algorithms considered in the scope of this paper, the ones that compute posterior estimations based on summary statistics (IS-ABC, SA-ABC, SL-ABC), and the ones that produce posterior estimates via weighting-based inferences (K-ABC, K2-ABC). In particular, K2-ABC uses a mapping of the observed and simulated data from the simulation space to an RKHS and employs a maximum mean discrepancy (MMD) criterion to construct a dissimilarity measure between distributions of observed and simulated data. While this method has outstanding results, it requires the tuning of free parameters. On the other hand, we can find the best possible performance of our AKL-ABC by running the inference stage in Algorithm 1 with $\tilde{w}_n = \kappa_E(\boldsymbol{\pi}^*, \boldsymbol{\pi}_n)$; we refer to this approach as "best".

Regarding the toy problem, we draw $N = 1000$ samples from a symmetric Dirichlet prior, $\boldsymbol{\pi} \sim$ Dirichlet($\mathbf{1}$), and then used the mixture model to form the simulated data by drawing 400 observations for each prior candidate. Moreover, we employ a histogram with ten bins as feature mapping (ϑ) in AKL-ABC and fix the Gaussian kernel bandwidths as 0.1 and 0.001 for the characteristic and similarity kernels, respectively, in the K2-ABC method. These parameters were tuned according to the cross-validation procedure planted by Park et al. [20].

For the real-world problem, we generate $N = 5000$ samples from the prior and then assess the model to form the simulated data by drawing 180 observations for each prior candidate. Besides, as feature mapping (ϑ), we selected the ten statistics used by Park et al. [20]. Furthermore, due to fluctuations produced by ϵ_t and e_t, we draw 100 simulations from the model given the expected value of the posterior and compute the boxplots of $\mathscr{E}$ for each method.

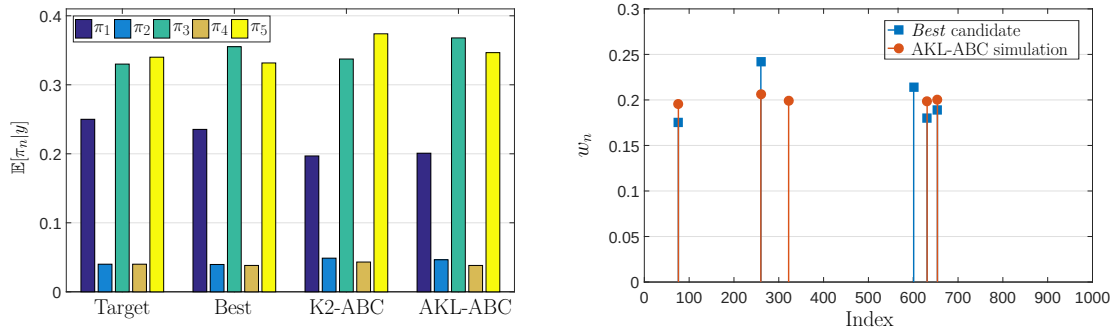
As suggested by authors in [32], we fixed the free parameters of the CKA algorithm regarding the gradient-descend optimization as follows: The adaptive step size of the learning rules are adjusted such that μ_γ^t and $\mu_{\boldsymbol{A}}^t$ decrease gradually from $1e-4$ to $1e-5$ through a maximum number of iterations empirically limited up to 100. Moreover, the initial guess for the rotation matrix $\boldsymbol{A}^0$ is computed using the well-known PCA method retaining the 95% of the variance explained (which defines the number of columns of $\boldsymbol{A}$, $d \leq D$), while γ^0 is calculated as the median of the input data Euclidean distances.

5. Results and Discussion

5.1. Toy Problem Results

Since this is a full controlled experiment with known parameters $\boldsymbol{\pi}^*$, we can find the best possible performance of our AKL-ABC by running the inference stage with $\tilde{w}_n = \kappa_E(\boldsymbol{\pi}^*, \boldsymbol{\pi}_n)$. The previous setting is equivalent to think that the statistical alignment found via CKA between $\mathbf{K}_{\boldsymbol{\theta}}$ and $\mathbf{K}_{\boldsymbol{s}}$ yields to $\mathbf{K}_{\boldsymbol{\theta}} = \mathbf{K}_{\boldsymbol{s}}$. Figure 4 shows the "best" performance along with K2-ABC and AKL-ABC results over the uniform mixture problem. In Figure 4a, the expected value of the posterior computed for all methods is close to the target. In particular, we obtained $\mathcal{E}_{Best} = 0.030 \pm 0.039$, $\mathcal{E}_{\text{K2-ABC}} = 0.063 \pm 0.042$, and $\mathcal{E}_{\text{AKL-ABC}} = 0.064 \pm 0.041$. Notice how the "best" approach achieves the lowest possible error providing a lower bound that the AKL-ABC could reach in the ideal case of perfect statistical alignment. These results show that our AKL-ABC is a competitive estimator to K2-ABC with a significant advantage

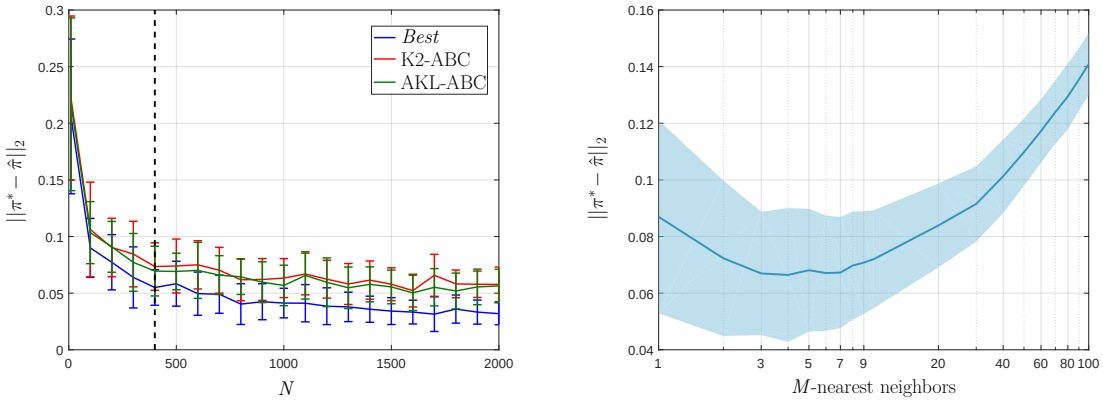
concerning the automatic selection of free parameters without requiring any tuning procedure. Besides, to provide a better understanding of the AKL-ABC effectiveness, in Figure 4b we show the weights for the five nearest neighbors (according to the LNS algorithm) used to compute the posteriors. As noted, the majority of the chosen simulations for AKL-ABC match the selected candidates using the "best", although our approach never observes the target.



(**a**) (**b**)

Figure 4. Uniform mixture model results. (**a**) Estimated mean posterior of mixing coefficients using various methods. (**b**) Weights of the first five nearest neighbors in AKL-ABC.

To observe the stability of the proposed AKL-ABC concerning the number of simulations, we select y^* to be a set of 400 synthetic observations drawn from the model (see Equation (20)) using the target $\boldsymbol{\pi}^*$. Then, we solve repeatedly the problem of inferring an approximation for $p(\boldsymbol{\theta}|y^*)$ by using each method, increasing N with a step-size of 100. Figure 5a shows the resulting $\mathcal{E}$ vs. N curve where the larger the number of simulations the lower and more confident the approximation error. While our AKL-ABC approach obtains a similar performance in comparison with the K2-ABC method in terms of stability and approximation error, our methodology overcomes the benchmark regarding its automatic philosophy that avoids any tuning procedure.



(**a**) (**b**)

Figure 5. Posterior approximation error over the uniform mixture model. (**a**) $\mathcal{E}$ vs. N curve. (**b**) $\mathcal{E}$ vs. M curve.

In turn, we compare the performance of the LNS algorithm concerning the choice of the M-nearest neighbors in AKL-ABC against a conventional grid search using y^* as observed data and $N = 400$ simulations. The $\mathcal{E}$ vs. M curve in Figure 5b proves that either small or large neighborhoods result in significant errors because, in the first case, the data structure is weakly encoded and, in the second case, the estimation is biased towards the population average. Besides, the larger the number of

neighbors, the more confident the posterior estimate, since the set of weights $\{w_n\}_{n=1}^{N}$ contains a larger number of non-zeros values. Therefore, introducing the LNS algorithm into the AKL-ABC results in an automatically computed number of neighbors ($M = 5$) lying on the minimum error region.

5.2. Real Dataset Results

Inferring the model parameters in this blowfly dataset is a very challenging task since the system dynamics can easily move from stable to chaotic regimes. The auxiliary model would produce completely different simulations in face of minimal fluctuations of the parameters [18,29]. This states an interesting scenario to test the performance and robustness of our AKL-ABC. In Figure 6, we provide the prior and the posterior approximations for each parameter, fixing σ_θ according to [37]. Notice how our proposal updates the beliefs about the model parameters leading to more concentrated posteriors. In the case of $\log(\sigma_p)$, two modes reflect different intervals with probable values for driving the noise realization associated with egg production in the blowfly population. However, there is a predominant mode that states higher probabilities for this parameter. Moreover, Figure 6g shows the closest and farthest predictions to the observed data selected from 100 realizations used to compute $\mathscr{E}$, showing that the inferred posterior lays on a stable regime. Finally, Figure 7 shows the performance of AKL-ABC compared against different ABC-based methods tested on the blowfly dataset by Park et al. [20]. As seen, our proposed method is a quite competitive approach to the benchmark. In particular, the confidence intervals of $\mathscr{E}$ are $(1.0620, 1.1232)$, $(0.9923, 1.1543)$, and $(1.8401, 2.0464)$ for the AKL-ABC, K2-ABC, and SL-ABC methods, respectively. The smaller the confidence interval, the more stable the mean posterior prediction since the model dynamic straightforwardly falls into chaotic regimes, even with minimal changes of the model parameters. Thus, the narrowest confidence interval obtained in AKL-ABC proves its capability to deal with complex dynamic data. Furthermore, AKL-ABC holds a significant advantage concerning the automatic selection of free parameters.

5.3. Computational Tractability

One of the primary considerations in ABC is computational tractability. Because there are no tuning procedures required in our AKL-ABC, the time complexity for a given dataset $\{\theta_n \in \mathbb{R}^P, \vartheta(x_n) \in \mathbb{R}^D\}_{n=1}^{N}$ is $O(N^2(P+\varrho+G+1))$. Nonetheless, in the case of other non-automatic ABC methods, the number of free parameters and the grid they define notably augment the computational burden. Namely, if the number of required operations for a given ABC method is denoted by $O(\mathscr{N})$, the final time complexity for tuning F free parameters is $O(\mathscr{N}(\prod_{i=1}^{F}\lambda_i))$, where $\lambda_i \in \mathbb{N}$ is the number of all possible values to be explored for the i-th free parameter. Notice that the thinner the exploration grid, the more prohibited the number of overall operations. For instance, the K2-ABC approach requires $O(N^2)$ operations [20]. However, the grid search needed for tuning the free parameters increases the computational complexity exponentially to $O(N^2(\lambda_1\lambda_2))$, where λ_1, λ_2 are related to grids defined over characteristic and similarity kernel widths. See that the performance of AKL-ABC depends on ϱ and G; however, in practice, the CKA and LNS algorithm have a fast convergence [32,35].

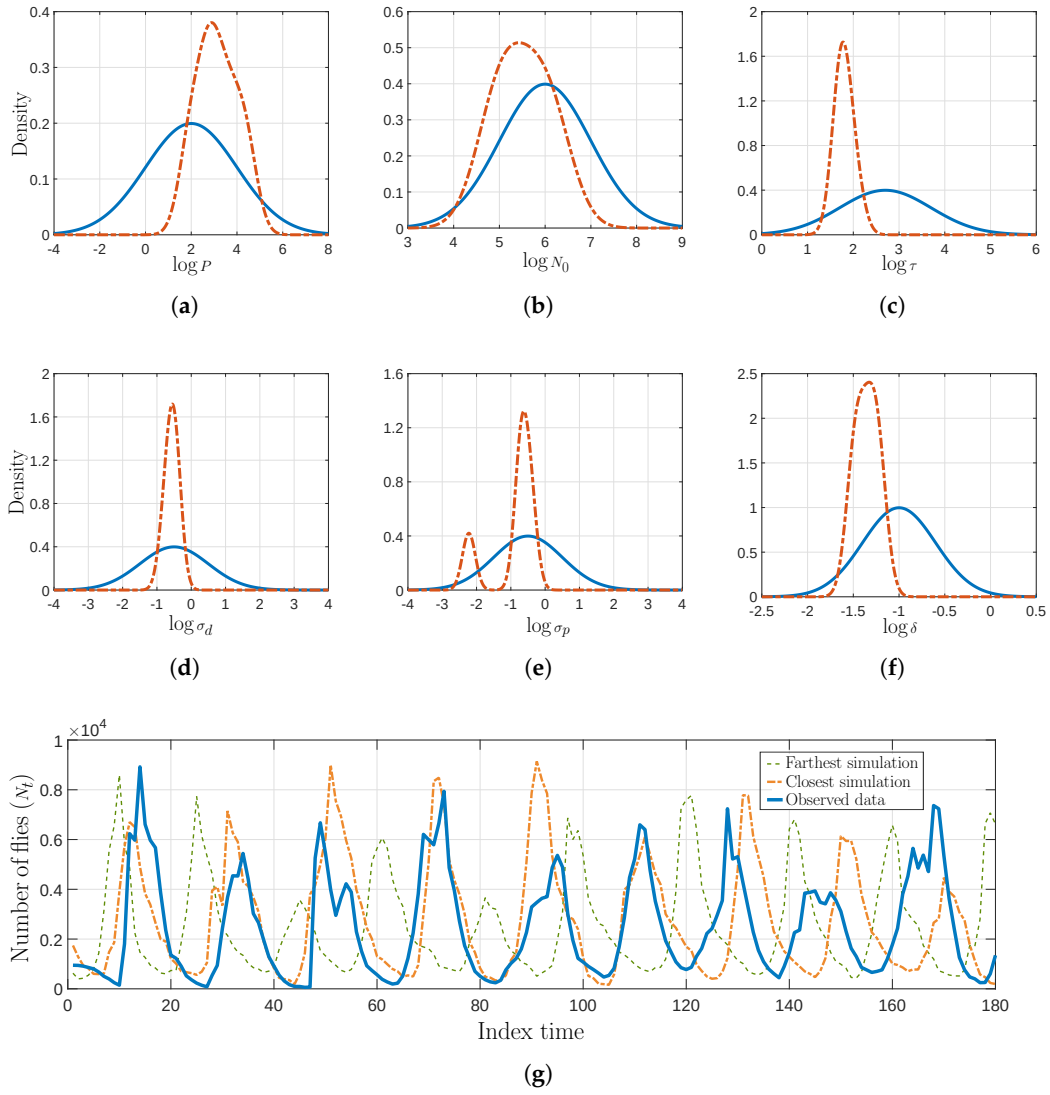


Figure 6. Non-linear ecological dynamic system results. (**a**–**f**) Prior distribution (solid line) and AKL-ABC-based posterior estimation (dashed line) of model parameters in the log-space. (**g**) Some predictions from the model using the expected value of the parameters found via AKL-ABC.

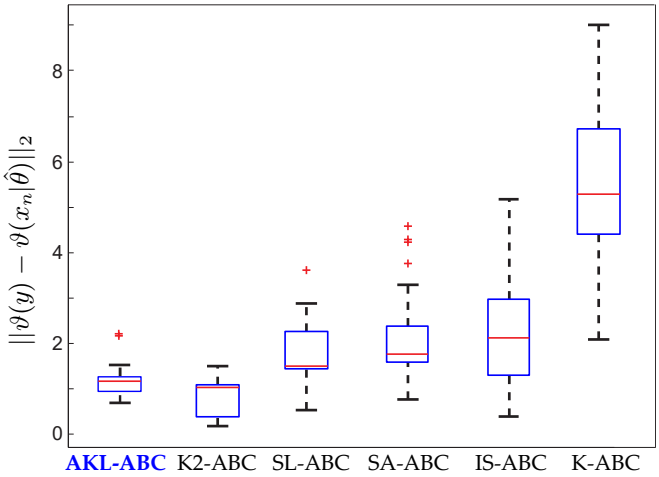


Figure 7. Performance of our AKL-ABC against the different ABC methods tested over the blowfly dataset by Park et al. [20].

6. Conclusions

In this paper, we focus on the problem of automatically performing Bayesian statistical inference under the intractability of the likelihood function. In particular, we propose an automatic enhancement of the well-known ABC algorithm devoted to approximate Bayesian inference called AKL-ABC. In particular, we include a metric learning approach based on a CKA framework to quantify the statistical alignment between parameter and simulation spaces in ABC. Then, a Mahalanobis distance is learned through CKA, and a graph representation based on a local neighborhood selection algorithm is employed to reveal local relationships among parameter and simulation samples. Notably, AKL-ABC has an advantage over other ABC approaches: The statistical alignment over parameter and simulations spaces and the concept of neighborhood introduce additional information in the inference procedure such that the overall ABC framework does not require the tuning of any free parameters. Attained results on a synthetic dataset and a real-world ecological system show the introduced AKL-ABC is robust to substantial changes in data dynamics and produces quite competitive posterior approximation compared to other non-automatic state-of-the-art ABC methods.

Future work includes the extension of AKL-ABC for high-dimensional problems where a large number of observations could be prohibited, via a possibly global neighborhood selection approach that supports a faster computation of the number of neighbors (M) required in AKL-ABC, taking into account comprehensive features rather than local properties of the input data. Moreover, the inclusion of other dissimilarity measures besides the Mahalanobis distance, coupled with the neighborhood-based philosophy of AKL-ABC, is also a potential line of research to deal with applications that gather complex and noisy data. Lastly, the computational burden would be enhanced based on stochastic gradient approaches [38].

Author Contributions: Conceptualization, W.G.V. and A.M.Á.-M.; software, W.G.V., A.M.Á.-M. and J.A.H.-M.; writing—original draft preparation, W.G.V. and J.A.H.-M.; formal analysis, all authors; writing—review and editing, all authors; project administration, Á.A.O.-G. All authors have read and approved the final manuscript.

Funding: Research under grants provided by the project with code: 1110-745-58696, funded by Colciencias, Colombia. Moreover, author J.H. was supported by Colciencias under the 775 agreement, "*Jóvenes investigadores e innovadores por la paz 2017*".

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following are the symbols used in this manuscript:

Symbol	Description
Θ	Parameter space
$\mathcal{X}$	Simulations space
$\mathcal{S}$	Feature space
$\mathcal{H}$	Reproducing Kernel Hilbert Space (RKHS)
θ_n	n-th prior sample
x_n	n-th simulation
y	Observed data
w_n	Representation weight associated to the n-th prior sample
z_n	Projection of the n-th simulation
Ω_n	Set of the M-nearest neighbors of θ_n according to the LNS algorithm
$\mathscr{Z}$	Set of the M-nearest neighbors of projected features of the observed data
$\mathcal{M}$	Auxiliary model
ϑ	Feature mapping

ϕ	Mapping function associated to the RHKS $\mathcal{H}$
$\kappa(\cdot,\cdot)$	Kernel function
κ_G	Gaussian kernel
κ_E	Similarity kernel
$\hat{\rho}(\cdot,\cdot)$	Statistical alignment between two kernel matrices
H_α	α-order Information Potencial
d_Θ	Distance between prior samples
$\mathrm{d}_\mathcal{X}$	Distance between simulations
$\mathrm{d}_\mathcal{S}$	Distance between features of simulations
$\mathrm{d}_\mathcal{H}$	Distance between distributions in an RKHS
d_e	Euclidean distance
$\zeta(\theta)$	Prior distribution
$p(\theta\|y)$	Posterior distribution
$p(y\|\theta)$	Likelihood function
$p(x\|\theta)$	Conditional distribution associated to simulations
P_{X_n}	Distribution of the n-th simulation
P_Y	Distribution of the observed data
$\mathbf{K}_{\boldsymbol{\theta}}$	Kernel matrix over prior samples
$\mathbf{K}_{\boldsymbol{\mathcal{S}}}$	Kernel matrix over features of simulations
$\boldsymbol{A}$	Projection matrix for the Centered Kernel Alignment (CKA)
$\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$	Sample covariance matrix of prior samples
$\boldsymbol{V}$	Feature matrix of simulations
ε	Threshold for rejection ABC
$\epsilon, \gamma, \sigma_\theta$	Gaussian kernel bandwidths
M	M-nearest neighbors according to the LNS algorithm
ϱ	number of iterations performed by the LNS algorithm
G	number of iterations required by the gradient-descent method in AKL-ABC
$\mu_{\boldsymbol{A}}, \mu_\gamma$	Step sizes for gradient descent rules

Appendix A. Local Neighborhood Selection (LNS) Algorithm

Given an input dataset $\Theta=\{\theta_n\}_{n=1}^N$, the following are the main steps of the LNS algorithm [35]:

1. Compute the Euclidean distance d_e for all points in Θ.
2. Construct the minimal connected neighborhood graph $\mathcal{G}$ of the given dataset Θ by the M-nearest neighbors method (MNN) fixing the smallest neighborhood size $m_{min}=1$. Check the full connectivity of the graph by using the Breadth-first search (BFS) [39]. If the graph is not full connected, update $m_{min}=m_{min}+1$ and start again this step.
3. Compute the geodesic distance d_g over $\mathcal{G}$ by using the Dijistra's algorithm [39].
4. Define $m_{max}=N^2/(m_{min}N_\mathcal{G})$, where $N_\mathcal{G}$ is the number of edges in $\mathcal{G}$ and N the number of samples in Θ.
5. Set the vector $\mathbb{M}_i=[m_{min}+1,...,m_{max}]$, with $\mathbb{M}_i\in\mathbb{R}^B$. The vector $\mathbb{M}_i$ contains the possible values of m for each θ_i.
6. For each θ_i define the sets $\mathcal{T}_{\mathrm{d}_e}^{(b)}$ and $\mathcal{T}_{\mathrm{d}_g}^{(b)}$, with $b=\{1,...,B\}$. Each element in $\mathcal{T}_{\mathrm{d}_e}^{(b)}$ and $\mathcal{T}_{\mathrm{d}_g}^{(b)}$ corresponds to the m_{i_b} nearest neighbors θ_j of θ_i ($j=\{1,...,m_{i_b}\}$) according to the Euclidean and geodesic distances, respectively.
7. Calculate the linearity conservation matrix $\mathbf{L}\in\mathbb{R}^{N\times B}$, which analyses the similarity of the neighborhoods obtained by d_e and d_g, taking into account the patch size. Each element of $\mathbf{L}$ can be computed as, $\ell_{ib}=|\{\overline{\mathcal{T}_{\mathrm{d}_e}^{(b)}\cap\mathcal{T}_{\mathrm{d}_g}^{(b)}}\}|/m_{i_b}$, where $|\cdot|$ calculates the cardinality of a set and $\overline{\{\cdot\}}$ the complement.
8. Initially, for each θ_i define the set $\mathscr{M}_o=\emptyset$. Verify the equality $\ell_{ib}=\min\{\boldsymbol{v}_i\}$, where $\boldsymbol{v}_i\in\mathbb{R}^B$ is the i-th row vector of $\mathbf{L}$. If the equality is fulfilled, then update $\mathscr{M}_o=\mathscr{M}_o\cup m_{i_b}$.
9. Define m_i for each θ_i as $m_i=\max\{\mathscr{M}_o\}$.

10. Smooth m_i to obtain similar properties in surrounding neighborhoods according to $m_i = (m_i + M_{\mathcal{O}}\mathbf{1}) \,/\, (m_i+1)$, where $M_{\mathcal{O}} \in \mathbb{R}^{m_i}$ is a vector with the sizes of the neighborhoods of each element in $\mathcal{O}$ (set with the first m_i nearest neighbors θ_j of θ_i according to the Euclidean distance, $j = \{1, .., m_i\}$).
11. Store all the values m_i into the vector $\mathbb{M}$.
12. Remove the outliers in $\mathbb{M}$ (see [40]), and replace them by the average of the elements in $\mathbb{M}$, which were not identified as outliers.
13. Each element in $\mathbb{M}$ contains the number of nearest neighbors m_i for each θ_i.

Subsequently, a global representation of the manifold is accomplished by defining the M-value as $M = \text{median}\{\mathbb{M}\}$ to avoid possible irregular neighborhood sizes.

References

1. Wasserman, L. Models, Statistical Inference and Learning. In *All of Statistics: A Concise Course in Statistical Inference*; Springer: New York, NY, USA, 2004; pp. 87–96. [CrossRef]
2. Thijssen, J. *A Concise Introduction to Statistical Inference*; Chapman and Hall/CRC: London, UK, 2016.
3. Casella, G.; Berger, R.L. *Statistical Inference*; Duxbury: Pacific Grove, CA, USA, 2002; Volume 2.
4. Bickel, P.; Klaassen, C.; Ritov, Y.; Wellner, J. *Efficient and Adaptive Estimation for Semiparametric Models*; Johns Hopkins Series in the Mathematical Sciences; Springer: New York, NY, USA, 1998.
5. Box, G.E.; Tiao, G.C. *Bayesian Inference in Statistical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 40.
6. Meeker, W.Q.; Hahn, G.J.; Escobar, L.A. *Statistical Intervals: A Guide for Practitioners and Researchers*; John Wiley & Sons: Hoboken, NJ, USA, 2017; Volume 541.
7. Toni, T.; Welch, D.; Strelkowa, N.; Ipsen, A.; Stumpf, M. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface* **2009**, *6*, 187–202. [CrossRef] [PubMed]
8. Pritchard, J.K.; Seielstad, M.T.; Perez-Lezaun, A.; Feldman, M.W. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Mol. Biol. Evol.* **1999**, *16*, 1791–1798. [CrossRef] [PubMed]
9. Liepe, J.; Kirk, P.; Filippi, S.; Toni, T.; Barnes, C.; Stumpf, M. A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. *Nat. Protoc.* **2014**, *9*, 439–456. [CrossRef] [PubMed]
10. Holden, P.B.; Edwards, N.R.; Hensman, J.; Wilkinson, R.D. ABC for Climate: Dealing with Expensive Simulators. In *Handbook of Approximate Bayesian Computation*; CRC Press: Boca Raton, FL, USA, 2018; Chapter 19. [CrossRef]
11. Fasiolo, M.; Wood, S.N. Approximate methods for dynamic ecological models. *arXiv* **2015**, arXiv:1511.02644.
12. Fan, Y.; Meikle, S.R.; Angelis, G.; Sitek, A. ABC in nuclear imaging. In *Handbook of Approximate Bayesian Computation*; CRC Press: Boca Raton, FL, USA, 2018; Chapter 25. [CrossRef]
13. Wawrzynczak, A.; Kopka, P. Approximate Bayesian Computation for Estimating Parameters of Data-Consistent Forbush Decrease Model. *Entropy* **2018**, *20*, 622. [CrossRef]
14. Turner, B.M.; Zandt, T.V. A tutorial on approximate Bayesian computation. *J. Math. Psychol.* **2012**, *56*, 69–85. [CrossRef]
15. Hainy, M.; Müller, W.G.; Wynn, H.P. Learning Functions and Approximate Bayesian Computation Design: ABCD. *Entropy* **2014**, *16*, 4353–4374. [CrossRef]
16. Fearnhead, P.; Prangle, D. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation [with Discussion]. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2012**, *74*, 419–474. [CrossRef]
17. Joyce, P.; Marjoram, P. Approximately sufficient statistics and bayesian computation. *Stat. Appl. Genet. Mol. Biol.* **2008**, *7*. [CrossRef]
18. Wood, S. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* **2010**, *466*, 1102–1104. [CrossRef]
19. Gleim, A.; Pigorsch, C. Approximate Bayesian Computation with Indirect Summary Statistics. Available online: http://ect-pigorsch.mee.uni-bonn.de/data/research/papers (accessed on 10 July 2019).

20. Park, M.; Jitkrittum, W.; Sejdinovic, D. K2-ABC: Approximate Bayesian Computation with Kernel Embeddings. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; Gretton, A., Robert, C.C., Eds.; PMLR: Cadiz, Spain, 2016; Volume 51, pp. 398–407.
21. González-Vanegas, W.; Alvarez-Meza, A.; Orozco-Gutierrez, Á. Sparse Hilbert Embedding-Based Statistical Inference of Stochastic Ecological Systems. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*; Mendoza, M., Velastín, S., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 255–262.
22. Cortes, C.; Mohri, M.; Rostamizadeh, A. Algorithms for Learning Kernels Based on Centered Alignment. *J. Mach. Learn. Res.* **2012**, *13*, 795–828.
23. González-Vanegas, W.; Álvarez-Meza, A.; Orozco-Gutiérrez, A. An Automatic Approximate Bayesian Computation Approach Using Metric Learning. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*; Vera-Rodriguez, R., Fierrez, J.; Morales, A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 12–19.
24. Beaumont, M.A. Approximate Bayesian Computation. *Annu. Rev. Stat. Its Appl.* **2019**, *6*, 379–403. [CrossRef]
25. Prangle, D. Summary statistics in approximate Bayesian computation. In *Handbook of Approximate Bayesian Computation*; CRC Press: Boca Raton, FL, USA, 2018; Chapter 5. [CrossRef]
26. Pigorsch, E.G.C. Approximate Bayesian Computation with Indirect Summary Statistics; Technical Report. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.665.5503 (accessed on 19 September 2019).
27. Nakagome, S.; Fukumizu, K.; Mano, S. Kernel approximate Bayesian computation in population genetic inferences. *Stat. Appl. Genet. Mol. Biol.* **2013**, *12*, 667–678. [CrossRef] [PubMed]
28. Mitrovic, J.; Sejdinovic, D.; Teh, Y.W. DR-ABC: Approximate Bayesian Computation with Kernel-Based Distribution Regression. In *Machine Learning Research, Proceedings of the 33rd International Conference on Machine Learning*; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, New York, NY, USA, 2016; Volume 48, pp. 1482–1491.
29. Meeds, E.; Welling, M. GPS-ABC: Gaussian Process Surrogate Approximate Bayesian Computation. *arXiv* **2014**, arXiv:1401.2838.
30. Jiang, B.; Wu, T.Y.; Zheng, C.; Wong, W.H. Learning summary statistic for approximate bayesian computation via deep neural network. *Stat. Sin.* **2017**, *27*, 1595–1618.
31. Creel, M. Neural nets for indirect inference. *Econom. Stat.* **2017**, *2*, 36–49. [CrossRef]
32. Alvarez-Meza, A.M.; Orozco-Gutierrez, A.; Castellanos-Dominguez, G. Kernel-Based Relevance Analysis with Enhanced Interpretability for Detection of Brain Activity Patterns. *Front. Neurosci.* **2017**, *11*, 550. doi:10.3389/fnins.2017.00550. [CrossRef]
33. Brockmeier, A.J.; Choi, J.S.; Kriminger, E.G.; Francis, J.T.; Principe, J.C. Neural Decoding with Kernel-Based Metric Learning. *Neural Comput.* **2014**, *26*, 1080–1107. [CrossRef]
34. Álvarez-Meza, A.M.; Cárdenas-Peña, D.; Castellanos-Dominguez, G. Unsupervised Kernel Function Building Using Maximization of Information Potential Variability. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*; Bayro-Corrochano, E., Hancock, E., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 335–342.
35. Álvarez Meza, A.; Valencia-Aguirre, J.; Daza-Santacoloma, G.; Castellanos-Domínguez, G. Global and local choice of the number of nearest neighbors in locally linear embedding. *Pattern Recognit. Lett.* **2011**, *32*, 2171–2177. [CrossRef]
36. Cristianini, N.; Shawe-Taylor, J.; Elisseeff, A.; Kandola, J.S. On Kernel-Target Alignment. In *Advances in Neural Information Processing Systems 14*; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2002; pp. 367–373.
37. Shimazaki, H.; Shinomoto, S. Kernel bandwidth optimization in spike rate estimation. *J. Comput. Neurosci.* **2010**, *29*, 171–182. [CrossRef]
38. Liu, L.; Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Han, J. On the Variance of the Adaptive Learning Rate and Beyond. *arXiv* **2019**, arXiv:1908.03265.

39. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Series in Artificial Intelligence; Prentice Hall: Upper Saddle River, NJ, USA, 2010.
40. Rencher, A.C. *Methods of Multivariate Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2003; Volume 492.