Bayesian Learning of Conditional Kernel Mean Embeddings for Automatic Likelihood-Free Inference

Kelvin Hsu[12] **Fabio Ramos**[13]

[1]School of Computer Science, The University of Sydney, Australia. [2]CSIRO, Australia. [3]NVIDIA, USA.

Abstract

In likelihood-free settings where likelihood evaluations are intractable, approximate Bayesian computation (ABC) addresses the formidable inference task to discover plausible parameters of simulation programs that explain the observations. However, they demand large quantities of simulation calls. Critically, hyperparameters that determine measures of simulation discrepancy crucially balance inference accuracy and sample efficiency, yet are difficult to tune. In this paper, we present kernel embedding likelihood-free inference (KELFI), a holistic framework that automatically learns model hyperparameters to improve inference accuracy given limited simulation budget. By leveraging likelihood smoothness with conditional mean embeddings, we nonparametrically approximate likelihoods and posteriors as surrogate densities and sample from closed-form posterior mean embeddings, whose hyperparameters are learned under its approximate marginal likelihood. Our modular framework demonstrates improved accuracy and efficiency on challenging inference problems in ecology.

1 Introduction

Scientific understanding of complex phenomena are deeply reliant on the study of probabilistic generative models and their match with real world data. Often, latent and convoluted interactions result in intractable likelihood evaluations, making the setting *likelihood-free*. Instead, generative models are expressed as a stochastic forward model simulator. Inference on latent variables in this setting is particularly challenging. Approximate Bayesian computation (ABC) methods are the state-of-the-art in simulation-based Bayesian inference with intractable likelihoods (Marin et al., 2012). They infer posterior distributions of simulator parameters that aim to explain observed data. The posterior is of interest in its own right for understanding the complex phenomena, and also useful in forming predictions of future observations. They are popular due to their simplicity and applicability, and have been used extensively in the biological sciences (Beaumont, 2010; Toni et al., 2009). Nevertheless, complex models are often prohibitively expensive to simulate. Evolutionary processes of ecological systems, vibrational modes of a mechanical structure, and fluid flow across surfaces are all examples that result in formidable inference problems with demanding forward simulations. It is thus imperative for inference algorithms to perform under the constraint of limited simulation calls, posing an exceptionally challenging task.

Often, ABC methods rely on discrepancy measures between simulations and observations that are parametrized by hyperparameters such as ϵ. The resulting posterior approximation is highly sensitive to the choice of hyperparameters, yet appropriate hyperparameter tuning strategies remain to be established.

To address these issues, we present kernel embedding likelihood-free inference (KELFI), a holistic framework consisting of (1) a consistent surrogate likelihood *model* that modularizes queries from simulation calls, (2) a Bayesian *learning* objective for hyperparameters that improves inference accuracy, and (3) a posterior surrogate density and a super-sampling *inference* algorithm using its closed-form posterior mean embedding.

KELFI is based on approximating likelihoods with simulation samples using conditional mean embeddings (CMEs). CMEs encode conditional expectations empirically by leveraging smoothness within a reproducing kernel Hilbert space (RKHS) with only a small number of examples. This modularizes inference away from simulation calls. Consequently, scientists can proceed with posterior analysis after any number of simulations. Furthermore, KELFI infers both ap-

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

proximate posterior densities and samples. Critically, our learning algorithm tunes hyperparameters directly for the inference problem, including adapting ϵ to the number of simulations used. This removes the need for practitioners to arduously select hyperparameters. Finally, it can be extended to automatically learn the relevance and usefulness of each summary statistic.

2 Likelihood-Free Inference

In the likelihood-free setting, we begin with a stochastic forward model simulator which synthesizes simulations $\mathbf{x}$ given a parameter setting $\boldsymbol{\theta}$. Let $\boldsymbol{\theta} \in \vartheta$ denote a realization of the latent variable or parameter $\boldsymbol{\Theta}$, where we use upper cases to denote random variables. Let $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{D}$ denote realizations of simulation output $\mathbf{X}$ and observations $\mathbf{Y}$ respectively. We represent the simulator as $p(\mathbf{x}|\boldsymbol{\theta})$ from which we can only simulate or sample, but not query its density, making likelihood evaluations intractable, thus *likelihood-free*. To begin inference we posit a prior density $p(\boldsymbol{\theta})$ that encodes prior knowledge about plausible parameter settings to guide the inference. The goal is to infer a posterior distribution on the parameters $\boldsymbol{\theta}$ that could generate simulations $\mathbf{x}$ similar to our observations $\mathbf{y}$ by some comparison measure. This measure could be done by a standard ϵ-kernel or ABC kernel $p_\epsilon(\mathbf{y}|\mathbf{x}) = \kappa_\epsilon(\mathbf{y}, \mathbf{x})$, such as a Gaussian density $\mathcal{N}(\mathbf{y}|\mathbf{x}, \epsilon^2 I)$ (Price et al., 2017; Moreno et al., 2016).

Based on this formulation, the true full likelihood of our model can be written as follows,

$$p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) = \int_{\mathcal{X}} p_\epsilon(\mathbf{y}|\mathbf{x}) p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = \mathbb{E}[\kappa_\epsilon(\mathbf{y}, \mathbf{X}) | \boldsymbol{\Theta} = \boldsymbol{\theta}]. \tag{2.1}$$

The corresponding posterior of interest is $p_\epsilon(\boldsymbol{\theta}|\mathbf{y}) = p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) / p_\epsilon(\mathbf{y})$ where $p_\epsilon(\mathbf{y}) = \int_\vartheta p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$. Due to the presence of a non-zero ϵ, even a perfect approximation to the soft posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ will not be the exact posterior $p_{\epsilon=0}(\boldsymbol{\theta}|\mathbf{y})$ unless ϵ is annealed to zero. This is the necessary trade-off we make with limited simulations, where a non-zero ϵ is essential for tractable inference because no simulations will match the observations exactly in practice. If $\mathbf{y}$ is only available as a summary statistic, then this soft posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ that we are targeting is only an approximation to the posterior given the full data even with $\epsilon = 0$.

So far, the notation $\mathbf{x}$ and $\mathbf{y}$ denote either the full dataset or their summary statistics. This is because the summary operation can be appended to the simulator program to output summary statistics directly. In either case, we let the target posterior be $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$, so the inference problem remains structurally identical. For simplicity however, from here on $\mathbf{x}$ and $\mathbf{y}$ will denote summary statistics unless stated otherwise.

Since $p(\mathbf{x}|\boldsymbol{\theta})$ is intractable, so is the likelihood (2.1). Instead, approximations are required. Markov chain Monte Carlo (MCMC) ABC use empirical means, $p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) \approx \frac{1}{S}\sum_{s=1}^S p_\epsilon(\mathbf{y}|\mathbf{x}^{(s)}) = \frac{1}{S}\sum_{s=1}^S \kappa_\epsilon(\mathbf{y}, \mathbf{x}^{(s)})$ (Andrieu et al., 2009). Synthetic likelihood ABC (SL-ABC) and adaptive SL-ABC (ASL-ABC) alternatively use Gaussian approximations $p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) \approx \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}} + \epsilon^2 I)$ and estimate the mean and covariance from simulations (Wood, 2010). These approaches require generating S new simulations $\{\mathbf{x}^{(s)}\}_{s=1}^S$ corresponding to $\boldsymbol{\theta}$ every time the likelihood is queried.

Not only are synthetic likelihoods parametric Gaussian approximations, they also approximate separately at each $\boldsymbol{\theta}$. Instead, surrogate likelihood approaches like KELFI use consistent nonparametric approximations so that (1) only one new simulation is required at each new parameter $\boldsymbol{\theta}$ and (2) likelihood queries do not need to be at parameters where simulations are available.

3 Kernel Embedding Likelihood-Free Inference

We present KELFI in three stages. In the model stage, we build a surrogate likelihood model by leveraging smoothness properties of CMEs. In the learning stage, we derive a differentiable marginal surrogate likelihood to drive hyperparameters learning. In the inference stage, we propose an algorithm to sample from the resulting mean embedding of the surrogate posterior.

When the prior is an anisotropic Gaussian $p(\boldsymbol{\theta}) = \prod_{d=1}^D \mathcal{N}(\theta_d|\mu_d, \sigma_d^2)$, closed form solutions for KELFI exists. We will present this setting since, for many common continuous priors, the likelihood-free inference (LFI) problem can be transformed into an equivalent problem that involves a Gaussian prior. See appendix F for more detail. When this is not possible or preferred, KELFI can be approximated arbitrarily well by using arbitrarily many prior samples.

3.1 Conditional Mean Embeddings

We begin with an overview of CMEs in the context of KELFI. Kernel mean embeddings (KMEs) are an arsenal of techniques used to represent distributions in a RKHS (Muandet et al., 2017). The key object is the mean embedding of a distribution $X \sim \mathbb{P}$ under a positive definite kernel k via $\mu_X := \int_{\mathcal{X}} k(x, \cdot) d\mathbb{P}(x) = \int_{\mathcal{X}} k(x, \cdot) p(x) dx \in \mathcal{H}_k$, where the last equality assumes a density p for $\mathbb{P}$ exists and $\mathcal{H}_k$ denotes the RKHS of k. They encode distributions in the sense that function expectations can be written as $\mathbb{E}[f(X)] = \langle \mu_X, f \rangle_{\mathcal{H}_k}$ if $f \in \mathcal{H}_k$. When μ_X can only be estimated empirically in some form denoted as $\hat{\mu}_X$, the expectation can be approximated by $\mathbb{E}[f(X)] \approx \langle \hat{\mu}_X, f \rangle_{\mathcal{H}_k}$.

CMEs are KMEs that encode conditional distributions. We specifically focus on their empirical estimates as we assume we only have the resource to obtain m sets of simulation data due to budget constraints. This results in joint samples $\{\boldsymbol{\theta}_j, \mathbf{x}_j\}_{j=1}^m$ from $p(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ by sampling from a proposal prior π for $\boldsymbol{\theta}_j \sim \pi(\boldsymbol{\theta})$ and simulating $\mathbf{x}_j \sim p(\mathbf{x}|\boldsymbol{\theta}_j)$ at each $\boldsymbol{\theta}_j$. Note these samples are not necessarily from the original joint distribution $p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ if $\pi \neq p_{\boldsymbol{\Theta}}$.

We define positive definite and characteristic kernels (Sriperumbudur et al., 2010) $k : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ and $\ell : \vartheta \times \vartheta \to \mathbb{R}$. When relevant, we denote the hyperparameters of k and ℓ with α and β, and refer to them as $k_\alpha = k(\cdot, \cdot; \alpha)$ and $\ell_\beta = \ell(\cdot, \cdot; \beta)$. An useful example of such a kernel is an anisotropic Gaussian kernel $\ell(\boldsymbol{\theta}, \boldsymbol{\theta}'; \boldsymbol{\beta}) = \exp\big(-\frac{1}{2}\sum_{d=1}^D (\theta_d - \theta'_d)^2/\beta_d^2\big)$ whose hyperparameters are length scales $\boldsymbol{\beta} = \{\beta_d\}_{d=1}^D$ for each dimension $d \in [D] := \{1, \dots, D\}$, and similarly for k.

For any function $f \in \mathcal{H}_k$, we construct an approximation to $\mathbb{E}[f(\mathbf{X})|\boldsymbol{\Theta} = \boldsymbol{\theta}]$ by the inner product $\langle f, \hat{\mu}_{\mathbf{X}|\boldsymbol{\Theta}=\boldsymbol{\theta}} \rangle_{\mathcal{H}_k}$ with an empirical CME $\hat{\mu}_{\mathbf{X}|\boldsymbol{\Theta}=\boldsymbol{\theta}}$. Importantly, $\hat{\mu}_{\mathbf{X}|\boldsymbol{\Theta}=\boldsymbol{\theta}}$ is estimated from the *joint* samples $\{\boldsymbol{\theta}_j, \mathbf{x}_j\}_{j=1}^m$, even though it is encoding the corresponding conditional distribution $p(\mathbf{x}|\boldsymbol{\theta})$. This approximation admits the following form (Song et al., 2009),

$$\mathbb{E}[f(\mathbf{X})|\boldsymbol{\Theta} = \boldsymbol{\theta}] \approx \mathbf{f}^T (L + m\lambda I)^{-1} \boldsymbol{\ell}(\boldsymbol{\theta}), \tag{3.1}$$

where $\mathbf{f} := \{f(\mathbf{x}_j)\}_{j=1}^m$, $L := \{\ell(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)\}_{i,j=1}^m$, $\boldsymbol{\ell}(\boldsymbol{\theta}) := \{\ell(\boldsymbol{\theta}_j, \boldsymbol{\theta})\}_{j=1}^m$, and $\lambda \geq 0$ is a regularization parameter. This approximation is known to converge at $O_p(m^{-\frac{1}{4}})$ if λ is chosen to decay at $O_p(m^{-\frac{1}{2}})$ or better under appropriate assumptions on $p(\mathbf{x}|\boldsymbol{\theta})$ (Song et al., 2013).

3.2 Model: Kernel Means Likelihood

We begin by presenting our surrogate likelihood model. Since the likelihood (2.1) is an expectation under $p(\mathbf{x}|\boldsymbol{\theta})$, we propose to approximate it via an inner product with the CME of $p(\mathbf{x}|\boldsymbol{\theta})$. Specifically, if we choose k such that $\kappa_\epsilon(\mathbf{y}, \cdot) \in \mathcal{H}_k$, then $p_\epsilon(\mathbf{y}|\boldsymbol{\theta})$ can be approximated by $q(\mathbf{y}|\boldsymbol{\theta}) := \langle \kappa_\epsilon(\mathbf{y}, \cdot), \hat{\mu}_{\mathbf{X}|\boldsymbol{\Theta}=\boldsymbol{\theta}} \rangle_{\mathcal{H}_k}$. We refer to $q(\mathbf{y}|\boldsymbol{\theta})$ as the kernel means likelihood (KML). While the KML provides an asymptotically correct likelihood surrogate, for finitely many simulations it is not necessarily positive nor normalized. By using $f = \kappa_\epsilon(\mathbf{y}, \cdot)$ in (3.1) where $\boldsymbol{\kappa}_\epsilon(\mathbf{y}) := \{\kappa_\epsilon(\mathbf{y}, \mathbf{x}_j)\}_{j=1}^m$ and $\mathbf{v}(\mathbf{y}) := (L + m\lambda I)^{-1} \boldsymbol{\kappa}_\epsilon(\mathbf{y})$, the KML becomes

$$q(\mathbf{y}|\boldsymbol{\theta}) = \sum_{j=1}^m v_j(\mathbf{y}) \ell(\boldsymbol{\theta}_j, \boldsymbol{\theta}). \tag{3.2}$$

The KML converges at the same rate as the CME. See theorem A.3 for proof. It is worthwhile to note that the assumption $\ell(\boldsymbol{\theta}, \cdot) \in \text{image}(C_{\boldsymbol{\Theta}\boldsymbol{\Theta}})$ is common for CMEs, and is not as restrictive as it may first appear, as it can be relaxed through introducing the regularization hyperparameter λ (Song et al., 2013).

Theorem 3.1. *Assume $\ell(\boldsymbol{\theta}, \cdot) \in \text{image}(C_{\boldsymbol{\Theta}\boldsymbol{\Theta}})$. The kernel means likelihood (KML) $q(\mathbf{y}|\boldsymbol{\theta})$ converges to the likelihood $p_\epsilon(\mathbf{y}|\boldsymbol{\theta})$ uniformly at rate $O_p((m\lambda)^{-\frac{1}{2}} + \lambda^{\frac{1}{2}})$ as a function of $\boldsymbol{\theta} \in \vartheta$ and $\mathbf{y} \in \mathcal{Y}$.*

To satisfy $\kappa_\epsilon(\mathbf{y}, \cdot) \in \mathcal{H}_k$, we choose the standard Gaussian ϵ-kernel $\kappa_\epsilon(\mathbf{y}, \mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \epsilon^2 I)$ and let $k_\alpha = k_\epsilon$ be a Gaussian kernel with length scale $\alpha = \epsilon$. Since $\kappa_\epsilon(\mathbf{y}, \mathbf{x})$ and $k_\epsilon(\mathbf{y}, \mathbf{x})$ are scalar multiples of each other, we have that $\kappa_\epsilon(\mathbf{y}, \cdot) \in \mathcal{H}_k$. In fact, any positive definite kernel κ_ϵ can be used, since we can simply choose k_α to be its scalar multiple to form the RKHS.

When the raw data is *iid* and no sufficient summary statistics are available, we can employ a kernel on the empirical distributions of the two datasets via $\kappa_{\epsilon,\alpha}(\mathbf{y}, \mathbf{x}) \propto k_{\epsilon,\alpha}(\mathbf{y}, \mathbf{x}) = \exp\big(-\frac{1}{2\epsilon^2}\|\hat{\mu}_{\mathbf{Y}} - \hat{\mu}_{\mathbf{X}}\|_{\mathcal{H}_k}^2\big)$, where $\hat{\mu}_{\mathbf{Y}} = \frac{1}{n}\sum_{i=1}^n \bar{k}_\alpha(\mathbf{y}_i, \cdot)$, $\hat{\mu}_{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^n \bar{k}_\alpha(\mathbf{x}_i, \cdot)$ are empirical mean embeddings of the observed and simulated raw data. Here $\bar{k}$ is another kernel with hyperparameters α. This was also used in double kernel ABC (K2-ABC) (Park et al., 2016) and distribution regression ABC (DR-ABC) (Mitrovic et al., 2016) to remove the requirement of summary statistics.

3.3 Learning: Hyperparameter Learning with Marginal Kernel Means Likelihood

We now propose a hyperparameter learning algorithm for our surrogate likelihood model. The main advantage of using an approximate surrogate likelihood surrogate model is that it readily provides a marginal surrogate likelihood quantity that lends itself to a hyperparameter learning algorithm. We define the marginal kernel means likelihood (MKML) as follows,

$$q(\mathbf{y}) := \int_\vartheta q(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \sum_{j=1}^m v_j(\mathbf{y}) \mu_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_j), \tag{3.3}$$

where $\mu_{\boldsymbol{\Theta}} := \int_\vartheta \ell(\boldsymbol{\theta}, \cdot) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ is the mean embedding of $p_{\boldsymbol{\Theta}}$. If we choose ℓ to be an anisotropic Gaussian kernel with length scales $\boldsymbol{\beta} = \{\beta_d\}_{d=1}^D$, then $\mu_{\boldsymbol{\Theta}}$ is closed-form for anisotropic Gaussian priors $p(\boldsymbol{\theta}) = \prod_{d=1}^D \mathcal{N}(\theta_d|\mu_d, \sigma_d^2)$. Let $\nu_d^2 := \beta_d^2 + \sigma_d^2$, then we have

$$\mu_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \ell_{\boldsymbol{\nu}}(\boldsymbol{\theta}, \boldsymbol{\mu}) \prod_{d=1}^D \frac{\beta_d}{\nu_d}. \tag{3.4}$$

Similar to the KML, the MKML converges at the same rate as the CME. See theorem A.4 for proof.

Theorem 3.2. *Assume $\ell(\boldsymbol{\theta}, \cdot) \in \text{image}(C_{\boldsymbol{\Theta}\boldsymbol{\Theta}})$. The marginal kernel means likelihood (MKML) $q(\mathbf{y})$ converges to marginal likelihood $p_\epsilon(\mathbf{y})$ uniformly at rate $O_p((m\lambda)^{-\frac{1}{2}} + \lambda^{\frac{1}{2}})$ as a function of $\mathbf{y} \in \mathcal{Y}$.*

Algorithm 3.1 KELFI: Kernel Embedding Likelihood-Free Inference

```
1: Input: Data $\mathbf{y}$, simulations $\{\boldsymbol{\theta}_j, \mathbf{x}_j\}_{j=1}^m \sim p(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, query parameters $\{\boldsymbol{\theta}_r^\star\}_{r=1}^R$, KML hyperparameters
   $(\boldsymbol{\epsilon}, \boldsymbol{\beta}, \lambda)$, prior hyperparameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ or samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=1}^T$, number of samples $S$, kernel $\ell$ and $\epsilon$-kernel $\kappa$
2: Compute $\mathbf{v} \leftarrow (L + m\lambda I)^{-1}\boldsymbol{\kappa}_{\boldsymbol{\epsilon}}(\mathbf{y})$ where $L \leftarrow \{\ell_{\boldsymbol{\beta}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)\}_{i,j=1}^m$ and $\boldsymbol{\kappa}_{\boldsymbol{\epsilon}}(\mathbf{y}) \leftarrow \{\kappa_{\boldsymbol{\epsilon}}(\mathbf{y}, \mathbf{x}_j)\}_{j=1}^m$
3: Compute $q(\mathbf{y}) \leftarrow \mathbf{v}^T \boldsymbol{\mu}_{\boldsymbol{\Theta}}$ where $\boldsymbol{\mu}_{\boldsymbol{\Theta}} \leftarrow \{\mu_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_j)\}_{j=1}^m$ using (3.4) or $\boldsymbol{\mu}_{\boldsymbol{\Theta}} \leftarrow \frac{1}{T}\tilde{L}\mathbf{1}_T$ where $\tilde{L} \leftarrow \{\ell_{\boldsymbol{\beta}}(\boldsymbol{\theta}_j, \tilde{\boldsymbol{\theta}}_t)\}_{j,t=1}^{m,T}$
4: Compute $H \leftarrow \{h(\boldsymbol{\theta}_j, \boldsymbol{\theta}_r^\star)\}_{j=1,r=1}^{m,R}$ using (3.6) or $H \leftarrow \frac{1}{T}\tilde{L}\tilde{L}^\star$ where $\tilde{L}^\star = \{\ell_{\boldsymbol{\beta}}(\tilde{\boldsymbol{\theta}}_t, \boldsymbol{\theta}_r^\star)\}_{t,r=1}^{T,R}$
5: Compute posterior mean embedding $\boldsymbol{\mu} \leftarrow H^T\mathbf{v}/q(\mathbf{y}) \in \mathbb{R}^R$ and initialize $\mathbf{a} \leftarrow \mathbf{0} \in \mathbb{R}^R$
6: for $s \in \{1, \dots, S\}$ do
7:     Obtain super-sample $\hat{\boldsymbol{\theta}}_s \leftarrow \boldsymbol{\theta}_{r^\star}^\star$ where $r^\star \leftarrow \mathrm{argmax}_{r \in \{1,\dots,R\}}\, \mu_r - (a_r/s)$
8:     Update kernel sum $\mathbf{a} \leftarrow \mathbf{a} + \{\ell_{\boldsymbol{\beta}}(\boldsymbol{\theta}_r^\star, \hat{\boldsymbol{\theta}}_s)\}_{r=1}^R$
9: end for
10: Output: Posterior super-samples $\{\hat{\boldsymbol{\theta}}_s\}_{s=1}^S$
```

Consequently, the MKML $q(\mathbf{y}) = q(\mathbf{y}; \boldsymbol{\epsilon}, \boldsymbol{\beta}, \lambda)$ approximates the true marginal likelihood $p_\epsilon(\mathbf{y})$ of the inference problem defined by our likelihood-prior pair. It is a function of the hyperparameters $(\boldsymbol{\epsilon}, \boldsymbol{\beta}, \lambda)$ of the ϵ-kernel and KML model. As $p_\epsilon(\mathbf{y})$ is unavailable, we instead maximize the MKML for hyperparameter learning. Furthermore, prior hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can also be included and learned jointly. Since the map $(\boldsymbol{\epsilon}, \boldsymbol{\beta}, \lambda) \mapsto q(\mathbf{y}; \boldsymbol{\epsilon}, \boldsymbol{\beta}, \lambda)$ is differentiable, optimization can be done in an auto-differentiation environment. The learning objective to be optimized is computed in line 3 of algorithm 3.1. Each automatic gradient update has complexity dominated by $O(m^3)$ due to the Cholesky decomposition in line 2. However, since we are addressing scenarios where simulations are limited so that m is small, this optimization is relatively fast.

Importantly, if we use an anisotropic Gaussian density for the ϵ-kernel $\kappa_{\boldsymbol{\epsilon}}$ where $\boldsymbol{\epsilon} = \{\epsilon_i\}_{i=1}^n$ are the length scales corresponding to each summary statistic $\mathbf{y} = \{y_i\}_{i=1}^n$, we can perform automatic relevance determination (ARD) to learn the relevance and usefulness of each summary statistic, where a small length scale indicate high relevance for that statistic. This is because $\boldsymbol{\epsilon}$ are also the length scales of the kernel k which defines the RKHS $\mathcal{H}_k$. Since the anistropic Gaussian kernel is learned, we also refer to it as an ARD kernel. We can also learn the length scales $\boldsymbol{\beta} = \{\beta_d\}_{d=1}^D$ for the kernel $\ell_{\boldsymbol{\beta}}$ on $\boldsymbol{\theta}$, although we found that it is more useful to let $\boldsymbol{\beta} = \beta_0\boldsymbol{\sigma}$ where $\boldsymbol{\sigma} = \{\sigma_d\}_{d=1}^D$ are the standard deviations of the Gaussian prior. By doing this, we make better use of the scale differences within $\boldsymbol{\theta}$ from the prior, and let β_0 learn the overall scale that is most useful for the KML.

For general non-Gaussian kernels and priors, $\mu_{\boldsymbol{\Theta}}$ in (3.3) can be approximated using T independent prior samples $\tilde{\boldsymbol{\theta}}_t \sim p(\boldsymbol{\theta})$, $t \in [T]$, as $\tilde{\mu}_{\boldsymbol{\Theta}} = \frac{1}{T}\sum_{t=1}^T \ell(\tilde{\boldsymbol{\theta}}_t, \cdot)$.

By formulating a learning objective directly for the inference problem, KELFI provides a way to automatically tune ϵ and its own model hyperparameters.

3.4 Inference: Kernel Means Posterior and Posterior Embedding Super-Sampling

We finally present an approach for posterior inference by super-sampling directly from the equivalent posterior mean embedding defined by the KML model and the prior. Our approach begins by defining a surrogate density to approximate the posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ in analogy to the Bayes' rule, $q(\boldsymbol{\theta}|\mathbf{y}) := q(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})/q(\mathbf{y})$. We refer to $q(\boldsymbol{\theta}|\mathbf{y})$ as the kernel means posterior (KMP). Importantly, $q(\boldsymbol{\theta}|\mathbf{y})$ is unaffected even if κ_ϵ is unnormalized, so that ϵ-kernels on distributions can be readily used. The KMP has the following convergence properties. See theorem A.5 for proof.

Theorem 3.3. *Assume $\ell(\boldsymbol{\theta}, \cdot) \in \mathrm{image}(C_{\boldsymbol{\Theta\Theta}})$ and that there exists $\delta > 0$ such that $q(\mathbf{y}) \geq \delta$ for all $m \geq M$ where $M \in \mathbb{N}_+$. The kernel means posterior (KMP) $q(\boldsymbol{\theta}|\mathbf{y})$ converges pointwise to the posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ at rate $O_p((m\lambda)^{-\frac{1}{2}} + \lambda^{\frac{1}{2}})$ as a function of $\boldsymbol{\theta} \in \vartheta$ and $\mathbf{y} \in \mathcal{Y}$. If $\sup_{\boldsymbol{\theta} \in \vartheta} p(\boldsymbol{\theta}) < \infty$ and $\sup_{\boldsymbol{\theta} \in \vartheta} p_\epsilon(\mathbf{y}|\boldsymbol{\theta}) < \infty$, then the convergence is uniform in $\boldsymbol{\theta} \in \vartheta$. If $\sup_{\mathbf{y} \in \mathcal{Y}} p_\epsilon(\boldsymbol{\theta}|\mathbf{y}) < \infty$, then the convergence is uniform in $\mathbf{y} \in \mathcal{Y}$.*

Importantly, the requirement for a $\delta > 0$ such that $q(\mathbf{y}) \geq \delta$ for all $m \geq M$ where $M \in \mathbb{N}_+$ provides an intuition for why high MKML values are favorable for learning a good approximate posterior. This requirement is an reflection on the capability of the simulator to recreate the observations $\mathbf{y}$ relative to the scale ϵ. Intuitively, the more capable the simulator $p(\mathbf{x}|\boldsymbol{\theta})$ is at generating simulations $\mathbf{x}$ that is close to $\mathbf{y}$ with respect to ϵ, the higher $p_\epsilon(\mathbf{y}) > 0$ will be relatively. Since theorem 3.2 guarantees that, for large $m > M$, $q(\mathbf{y})$ will be close to $p_\epsilon(\mathbf{y})$, we have that $q(\mathbf{y}) > 0$ for all large $m > M$ with increasing probability. In this situation, theorem 3.3 guarantees that the KMP will converge to the posterior of interest. However, consider the case when the simulator is ill-designed to recreate $\mathbf{y}$ such that the true marginal likelihood $p_\epsilon(\mathbf{y}) \approx 0$ is small. As $q(\mathbf{y})$ tends to $p_\epsilon(\mathbf{y}) \approx 0$ due to theorem 3.2, it may struggle to always stay strictly positive even for large $m > M$ since it is stochastically converging to approximately zero. In this case, convergence is difficult since the simulator was ill-designed. However, by learning ϵ through maximizing $q(\mathbf{y})$, we adapt the threshold ϵ to make $p_\epsilon(\mathbf{y})$ as high as possible, leading to a more stable posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ for the KMP to converge to.

We now define kernel means posterior embedding (KMPE), the mean embedding of the KMP, as $\tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}(\boldsymbol{\theta}^\star) := \int_\vartheta \ell(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) q(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}$. This becomes

$$\tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}(\boldsymbol{\theta}^\star) = \frac{1}{q(\mathbf{y})} \sum_{j=1}^{m} v_j(\mathbf{y}) h(\boldsymbol{\theta}_j, \boldsymbol{\theta}^\star), \tag{3.5}$$

where $h(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) := \int_\vartheta \ell(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \ell(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) p(\tilde{\boldsymbol{\theta}}) d\tilde{\boldsymbol{\theta}}$. Importantly, since the KMPE is constructed from the CME used to form the KML, it converges in RKHS norm at the same rate. See theorem A.6 for proof.

Theorem 3.4. *Assume $\ell(\boldsymbol{\theta}, \cdot) \in \mathrm{image}(C_{\boldsymbol{\Theta}\boldsymbol{\Theta}})$ and that there exists $\delta > 0$ such that $q(\mathbf{y}) \geq \delta$ for all $m \geq M$ where $M \in \mathbb{N}_+$. The kernel means posterior embedding (KMPE) $\tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}$ converges in RKHS norm to the posterior mean embedding $\mu_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}$ at rate $O_p((m\lambda)^{-\frac{1}{2}} + \lambda^{\frac{1}{2}})$.*

If we choose ℓ to be an anisotropic Gaussian kernel with length scales $\boldsymbol{\beta} = \{\beta_d\}_{d=1}^D$, h exhibits the following closed-form under anisotropic Gaussian priors,

$$h(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) = \prod_{d=1}^{D} \frac{s_d}{\sigma_d} \exp\left[-\frac{1}{2 s_d^2} \Big(a_d - b_d^2 \Big) \right], \tag{3.6}$$

where $a_d := (\theta_d^2 + \theta_d^{\star 2} + \gamma_d^2 \mu_d^2)/(2 + \gamma_d^2)$, $b_d := (\theta_d + \theta_d^\star + \gamma_d^2 \mu_d)/(2 + \gamma_d^2)$, $\gamma_d^2 := \beta_d^2 / \sigma_d^2$ and $s_d^{-2} := 2\beta_d^{-2} + \sigma_d^{-2}$. For general non-Gaussian kernels and priors, h can be approximated as $\tilde{h}(\boldsymbol{\theta}, \boldsymbol{\theta}^\star) = \frac{1}{T} \sum_{t=1}^{T} \ell(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}_t) \ell(\tilde{\boldsymbol{\theta}}_t, \boldsymbol{\theta}^\star)$.

The KMP $q(\cdot|\mathbf{y})$ is bounded and normalized but potentially non-positive. Consequently, it can be seen as a surrogate density corresponding to a signed measure. This suggests that the map $q(\cdot|\mathbf{y}) \mapsto \tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}$ is injective for characteristic kernels ℓ, analogous to mean embeddings (Sriperumbudur et al., 2011). Furthermore, as the integral (3.5) is a linear operator on $\ell(\boldsymbol{\theta}^\star, \cdot)$, the surrogate posterior mean embedding $\tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}} \in \mathcal{H}_\ell$ is in the RKHS of ℓ. With a surrogate embedding that is injective to our surrogate posterior and in the RKHS, we can apply kernel herding (Chen et al., 2010) on $\tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}$ (3.5) using kernel ℓ to obtain S super-samples $\{\hat{\boldsymbol{\theta}}_s\}_{s=1}^S$ from the surrogate density $q(\boldsymbol{\theta}|\mathbf{y})$. That is, for each $s \in [S]$, the samples are obtained by

$$\hat{\boldsymbol{\theta}}_s = \operatorname*{argmax}_{\boldsymbol{\theta} \in \vartheta} \tilde{\mu}_{\boldsymbol{\Theta}|\mathbf{Y}=\mathbf{y}}(\boldsymbol{\theta}) - \frac{1}{s} \sum_{s'=1}^{s-1} \ell(\hat{\boldsymbol{\theta}}_{s'}, \boldsymbol{\theta}). \tag{3.7}$$

The inference algorithm is presented in algorithm 3.1.

4 Related Work

The simplest ABC algorithm is arguably the rejection ABC (REJ-ABC) sampler (Pritchard et al., 1999). It posits a set of prior parameters and rejects those whose simulations do not match the observations within a fixed threshold $\epsilon > 0$ under a distance measure.

Instead of sampling from the prior, MCMC-ABC and sequential Monte Carlo ABC (SMC-ABC) sample from proposal distributions iteratively and carefully accepts or discards each proposal stochastically based on approximate likelihood ratios (Sisson et al., 2007; Marjoram et al., 2003). They can however suffer from slow mixing, where it is difficult to escape a lucky sample with a high likelihood. They also do not leverage likelihood smoothness and thus require new simulations every iteration, which are then discarded and may still not result in an accepted sample.

Another branch of study include stochastic variational inference (SVI) approaches to ABC, which treats the likelihood approximation as another source of stochasticity in the stochastic gradient. This includes AV-ABC (Moreno et al., 2016), VBIL (Tran et al., 2017b), and VBSL (Ong et al., 2018). In contrast, likelihood-free variational inference (LFVI) (Tran et al., 2017a) uses density ratio estimation to approximate the variational objective, emphasizing inference on local latent variables. Nevertheless, SVI approaches posit parametric approximations that may have asymptotic bias.

Kernel-based approaches that leverage likelihood smoothness have been studied recently to reduce simulation requirements. The philosophy is that simulations of close-by parameters are informative, thus past results should not be discarded but remembered, even if this introduces model bias. Kernel ABC (K-ABC) (Nakagome et al., 2013), kernel recursive ABC (KR-ABC) (Kajihara et al., 2018), and kernel Bayes' rule (KBR) (Fukumizu et al., 2013) also employ CMEs to reduce simulation requirements. They differ to KELFI in the three aspects of model, learning, and inference. (Model) While they build posterior mean embeddings directly, KELFI builds likelihood surrogates first and make use of the full prior density to further leverage prior information before building posterior surrogates, which are then embedded into closed-form posterior mean embeddings. In contrast, the prior only appears as samples from $p(\boldsymbol{\theta})$ in K-ABC, KR-ABC, and KBR. This both limits the prior knowledge leveraged and prohibit the use of proposal prior samples. (Learning) KELFI crucially addresses hyperparameter learning in reference to the inference problem directly which was not straightforward previously. (Inference) K-ABC and KBR primarily infer posterior expectations, while KR-ABC produce point estimates. Instead, we de-

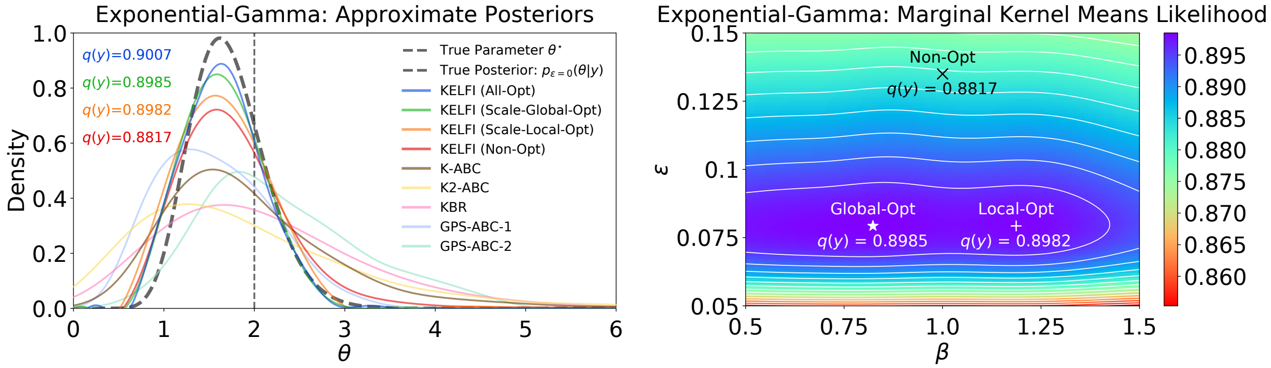

Figure 1: **(Left)** Comparison of approximate posteriors obtained from surrogate methods on the toy exponential-gamma problem. **(Right)** The corresponding MKML surface $q(\mathbf{y})$ as a function of (ϵ, β) with $\lambda = 10^{-3}\beta$.

sign a posterior sampling algorithm, which subsumes inferring posterior expectation. We further provide approximate posterior density KMP, which can both produce point estimates and quantify uncertainty.

As a consequence of theorem 3.4, the KMPE converges at rate $O_p(m^{-\frac{1}{4}})$ in RKHS norm if the regularization hyperparameter λ is chosen to decay at rate $O_p(m^{-\frac{1}{2}})$. Notably, this is faster than the convergence rate of KBR at $O_p(m^{-\frac{8}{27}\alpha})$ where $0 < \alpha \leq \frac{1}{2}$, which also requires other assumptions on the cross-covariance operators and for its two regularization hyperparameters to be decayed appropriately (Fukumizu et al., 2013).

Finally, we highlight that hyperparameter learning is a crucial aspect and differentiator of KELFI. This is especially true for learning ϵ, which tunes the critical balance between an accurate posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y}) \approx p_0(\boldsymbol{\theta}|\mathbf{y})$ with small ϵ requiring high numbers of simulation calls, or a less accurate posterior with large ϵ relaxing the number of simulations required. This has been a challenging issue to address in the ABC literature in reference to the inference problem, even though its selection is often pivotal to the performance of the algorithm.

In the Gaussian process (GP) literature, hyperparameter learning through maximum marginal likelihood plays an important role in the success of a GP regressor (GPR). GP surrogate ABC (GPS-ABC) (Meeds and Welling, 2014) and GP-accelerated ABC (GPA-ABC) (Wilkinson, 2014) model the summary statistics surface and log likelihood surface respectively via a GP surrogate. In contrast, the KML model is equivalent to placing a GP surrogate on the likelihood surface itself. This removes the assumption that summary statistics are independent and Gaussian distributed as in GPS-ABC. Importantly, while GPS-ABC and GPA-ABC apply the GP marginal likelihood to learn their surrogate hyperparameters, it cannot learn ϵ or other hyperparameters since they are not part of the surrogate. This is because both approaches maximize the marginal likelihood for the GPR problem on the their respective target surfaces, but not the marginal likelihood for the overall inference problem, thus excluding other hyperparameters in the process.

5 Experiments

The goal of the experiments is to demonstrate the inference accuracy of KELFI under limited simulation budget and the effectiveness of MKML hyperparameter learning. We begin with isotropic ϵ and anisotropic $\boldsymbol{\beta} = \beta_0 \boldsymbol{\sigma}$, and learn (ϵ, β_0) by maximizing the MKML (3.3) while keeping $\lambda = 10^{-3}\beta_0$ fixed for simplicity.

5.1 Toy Problem: Exponential-Gamma

The toy exponential-gamma problem is a standard benchmark for likelihood-free inference, since the true posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ is known and tractable even for $\epsilon = 0$.

To stress-test each method, we compare inference accuracy under very limited simulations of $m = 100$. We focus on comparing surrogate approaches, since other methods such as REJ-ABC, MCMC-ABC, SL-ABC, and ASL-ABC have reported simulation requirements several orders higher than 100 on this problem (Meeds and Welling, 2014). We use datasets of $n = 15$ for both observations and simulations, with their sample means as the summary statistic.

For GPS-ABC only we set a simulation budget of $m \leq 200$ and run it until 10000 posterior samples are generated. The hyperparameters of the GP surrogate itself are learned by maximizing the marginal likelihood of the GPR (Rasmussen and Williams, 2006). For the remaining hyperparameters that are not part of the surrogate, several configurations are compared and the results of the best two are shown, which used $m = 130$ and $m = 197$ simulations. For K-ABC, K2-ABC, and KBR, we use the median heuristic to set their length scale hyperparameters and manually

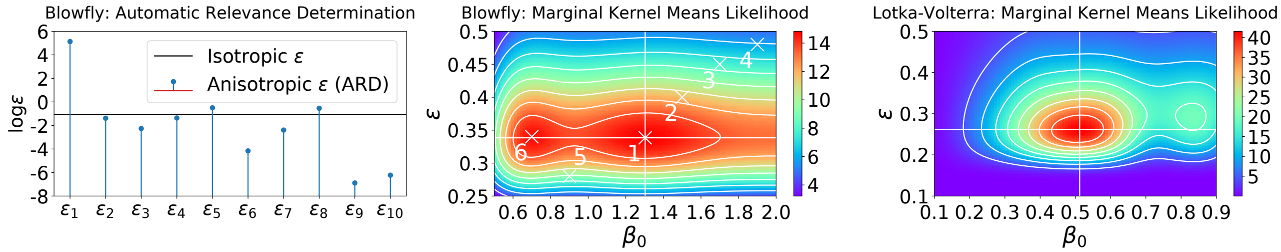

Figure 2: **(Left)** Blowfly: ARD on $\boldsymbol{\epsilon}$ for 10 summary statistics. **(Mid. & Right)** The MKML surface ($\times 10^5$) as a function of (ϵ, β_0) for fixed $\lambda = 10^{-3}\beta_0$ where $\boldsymbol{\beta} = \beta_0 \boldsymbol{\sigma}$. White intersection indicate optimum. For Blowfly, the NMSE (in %) for the indicated hyperparameter choices are: $(1) 0.72 \pm 0.02$, $(2) 1.10 \pm 0.01$, $(3) 2.07 \pm 0.01$, $(4) 2.15 \pm 0.02$, $(5) 1.11 \pm 0.02$, $(6) 1.11 \pm 0.03$. At $(\epsilon, \beta_0) = (10, 10)$ (outside the plot) the NMSE (in %) is 6.28 ± 0.03.

search for the most appropriate regularization hyperparameters. We use kernel density estimation (KDE) to visualize the posterior density from the unweighted samples of GPS-ABC and normalized weighted samples of K-ABC, K2-ABC, and KBR in fig. 1 (left).

For KELFI, we show the KMPs directly in fig. 1 (left). We first demonstrate the case when all hyperparameters $(\epsilon, \beta, \lambda)$ are learned (All-Opt). To enable visualization in 2D, we also present the case when the regularization hyperparameter λ is set to $10^{-3}\beta$ and only length scale hyperparameters (ϵ, β) are learned. In this case, we show KMPs under globally optimal (Scale-Global-Opt), locally optimal (Scale-Local-Opt), and arbitrarily chosen hyperparameters (Non-Opt). The corresponding MKML surface is shown in fig. 1 (right).

In fig. 1 we compare approximate posteriors from each algorithm against the true posterior $p_{\epsilon=0}(\boldsymbol{\theta}|\mathbf{y})$. While $\epsilon = 0$ for $p_{\epsilon=0}(\boldsymbol{\theta}|\mathbf{y})$, with only 100 simulations $\epsilon > 0$ is required for most LFI methods. Furthermore, except for K2-ABC, they only make use of summary statistics without further knowledge of the dataset size n. Consequently, most LFI methods produce approximations wider than $p_{\epsilon=0}(\boldsymbol{\theta}|\mathbf{y})$. Intuitively, there is not enough simulations and thus information to justify a more confident and peaked posterior. Nevertheless, by learning hyperparameters under the MKML, KELFI determines an appropriate scale ϵ for 100 simulations. As a result, KMPs are the closest to the true posterior $p_{\epsilon=0}(\boldsymbol{\theta}|\mathbf{y})$, with higher MKML $q(\mathbf{y})$ leading to more accurate KMPs $q(\boldsymbol{\theta}|\mathbf{y})$. This demonstrates the effectiveness of MKML as a hyperparameter learning objective for improving inference accuracy. In contrast, the two instances of GPS-ABC reveals that varying hyperparameters lead to significant changes in the resulting approximate posterior, yet without a similar objective like MKML it is unclear which one to use without ground truth. This is further emphasized by the wider posterior approximations obtained from K-ABC, K2-ABC, and KBR, which use the median heuristic to set hyperparameters. This is often sub-optimal since the heuristic makes no reference to the inference problem.

5.2 Chaotic Ecological Systems: Blowfly

The Blowfly simulator describes the complex population dynamics of adult blowflies. Across a range of parameters it exhibits chaotic behavior that have distinct discrepancies from real observations, resulting in a challenging inference problem. We follow the setup of Wood (2010). There are 6 model parameters from which the simulator generates a time series of 180 data points that is then summarized into 10 statistics as described in Meeds and Welling (2014), Moreno et al. (2016), and Park et al. (2016). We similarly place a broad diagonal Gaussian prior on log parameters.

The standard Blowfly problem has no ground truth parameters, only a set of observations. We therefore measure inference accuracy by considering mean squared errors (MSEs) between statistics generated using the posterior and the observed statistics. We normalize the MSE of each statistic by the corresponding MSE achieved under the prior, and average across the 10 statistics into a final normalized MSE (NMSE). As simulations are expensive, in fig. 3 (left) we record average NMSE against simulations used to understand inference efficiency. Each method is repeated 10 times with randomized simulations before their NMSE is averaged. Appendix D provides further details.

As new simulations become available, we relearn and update the hyperparameters for KELFI by maximizing the MKML. Figure 2 (center) shows an instance of the MKML surface used to learn the hyperparameters for KELFI when using $m = 280$ simulations. For KBR and K-ABC we update hyperparameters by the median length heuristic. For K-ABC we also report the case where the heurstic is scaled by a constant denoted with (S), which achieved significantly better accuracy and confirms that the heuristic is often sub-optimal.

Overall, the top three performers are KELFI, KBR, and GPS-ABC. Across a range of simulation calls, KELFI achieves the lowest error. It is also the only method that achieved less than 1% average NMSE

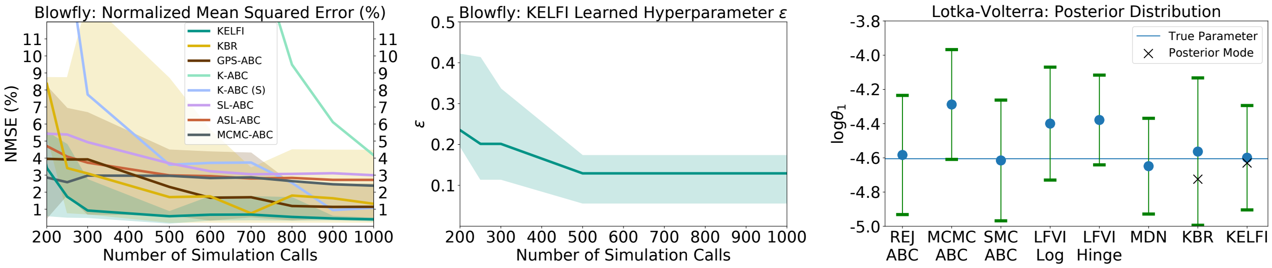

Figure 3: **(Left)** Blowfly: Average NMSE (in %) under posteriors against simulation calls. Shaded regions show NMSE variability for KELFI, KBR, and GPS-ABC. **(Mid.)** Blowfly: Learned ϵ value under maximum MKML. **(Right)** Lotka-Volterra: The middle 95% credible interval of the marginal posterior distribution of $\log \theta_1$.

within 1000 simulations and achieves this as early as 300 simulations. The most competitive methods to KELFI are KBR and GPS-ABC. For these three methods, we also show their variability from best to worst case NMSEs out of the 10 repeats to visualize their sensitivity to the stochasticity in randomized simulations. This reveals that KELFI is a stable outperformer with comparatively less variability across randomized runs.

We proceed to demonstrate and emphasize the effectiveness and suitability of MKML as a hyperparameter learning objective, using the case with 280 simulations as an example. Figure 2 (center) illustrates that hyperparameters with a higher MKML (3.3) result in lower NMSE consistently. Notably, even with suboptimal hyperparameter choices, KELFI still achieves competitive average NMSE scores of less than 2.2%. At 280 simulations, the next best average NMSE score is almost 3% by MCMC-ABC as shown in fig. 3 (left).

Figure 3 (center) suggests that learning the scale ϵ under MKML reveals an automatic decay schedule which does not have to be set a-priori. As ϵ controls the scale within which discrepancies between simulations and observations are measured, it is expected that this scale decays as more simulation data is available. Without the MKML, both the initialization of ϵ and its decay schedule are not straight forward to determine.

In fig. 2 (left), we show that we can perform ARD on the ABC ϵ-kernel $\kappa_{\boldsymbol{\epsilon}}$, and hence the kernel $k_{\boldsymbol{\epsilon}}$, by using a different ϵ_i for each of the 10 statistics. We do this by initializing each ϵ_i to the isotropic solution in fig. 2 (center) and further optimize the MKML to learn all ϵ_i jointly. In particular, the first summary statistic describes the average log population numbers nears its troughs (first quartile), and is determined to be comparatively irrelevant (high ϵ_i). Meanwhile, the last two statistics describe the number of peaks at two thresholds, and are determined to be comparatively relevant (low ϵ_i). This agrees with the intuition that Blowfly population dynamics are highly characterized by its peaks, instead of i its troughs (Wood, 2010).

5.3 Predator-Prey Dynamics: Lotka-Volterra

The Lotka-Volterra simulator describes the time evolution of the populations within a predator-prey system. Only for a small set of parameters does the model simulate a realistic scenario with oscillatory behavior, making the inference task formidably challenging. We follow the exact setup as described in Papamakarios and Murray (2016). There are 4 parameters and 9 normalized summary statistics. We place the same uniform prior on the log parameters and use the same ground truth parameters. After performing inference on all four parameters, we show in fig. 3 (right) the marginal posterior distribution for $\log \theta_1$.

KELFI achieves competitive performance using only 2500 simulations, with both posterior mean and mode close to the true value. The MKML for hyperparameter learning is shown in fig. 2 (right). Posterior mode is obtained by maximizing the KMP. Meanwhile, the three ABC methods used up to 100000 simulations. While confident, LFVI (Tran et al., 2017a) tends to have a biased posterior mean. For direct comparison, both KELFI and mixture density network (MDN) (Papamakarios and Murray, 2016) use the original prior as the proposal prior. KELFI achieves slightly higher accuracy than MDN which used 10000 simulations, 4 times that used for KELFI. Finally, we also similarly use 2500 simulations for KBR. With the same number of simulations, KELFI achieves higher accuracy in both mean and mode with higher confidence.

6 Conclusion

KELFI provides a holistic framework for automatic likelihood-free inference. It is a stable outperformer compared to state-of-the-art methods, while producing interpretable automatic relevance determination of summary statistics and automatic decay schedules for ϵ. By optimizing an approximate Bayesian marginal likelihood, it automatically learns and adapts hyperparameters including the ϵ-kernel to improve inference accuracy when limited simulations are available.

References

Andrieu, C., Roberts, G. O., et al. (2009). The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725.

Beaumont, M. A. (2010). Approximate bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406.

Carmeli, C., De Vito, E., Toigo, A., and Umanitá, V. (2010). Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61.

Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *The Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 109–116. AUAI Press.

Flaxman, S., Sejdinovic, D., Cunningham, J. P., and Filippi, S. (2016). Bayesian learning of kernel embeddings. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 182–191. AUAI Press.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99.

Fukumizu, K., Song, L., and Gretton, A. (2013). Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14(1):3753–3783.

Grünewälder, S., Lever, G., Baldassarre, L., Patterson, S., Gretton, A., and Pontil, M. (2012). Conditional mean embeddings as regressors. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, volume 2, pages 1823–1830.

Kajihara, T., Kanagawa, M., Yamazaki, K., and Fukumizu, K. (2018). Kernel recursive abc: Point estimation with intractable likelihood. In *International Conference on Machine Learning*, pages 2405–2414.

Kanagawa, M., Nishiyama, Y., Gretton, A., and Fukumizu, K. (2016). Filtering with state-observation examples via kernel monte carlo filter. *Neural computation*, 28(2):382–444.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.

Meeds, E. and Welling, M. (2014). Gps-abc: Gaussian process surrogate approximate bayesian computation. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 593–602. AUAI Press.

Mitrovic, J., Sejdinovic, D., and Teh, Y. W. (2016). DR-ABC: Approximate Bayesian Computation with kernel-based distribution regression.

Moreno, A., Adel, T., Meeds, E., Rehg, J. M., and Welling, M. (2016). Automatic variational abc. *stat*, 1050:28.

Muandet, K., Fukumizu, K., Sriperumbudur, B., Schölkopf, B., et al. (2017). Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141.

Nakagome, S., Fukumizu, K., and Mano, S. (2013). Kernel approximate Bayesian computation in population genetic inferences. *Statistical applications in genetics and molecular biology*, 12(6):667–678.

Ong, V. M., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. (2018). Variational Bayes with synthetic likelihood. *Statistics and Computing*, 28(4):971–988.

Papamakarios, G. and Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pages 1028–1036.

Park, M., Jitkrittum, W., and Sejdinovic, D. (2016). K2-ABC: Approximate Bayesian computation with kernel embeddings.

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2017). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, pages 1–11.

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

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

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

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968. ACM.

Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. (2010). On the relation between universality, characteristic kernels and RKHS embedding of measures. In *AISTATS*, pages 773–780.

Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. (2011). Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(Jul):2389–2410.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202.

Tran, D., Ranganath, R., and Blei, D. (2017a). Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533.

Tran, M.-N., Nott, D. J., and Kohn, R. (2017b). Variational Bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882.

Wilkinson, R. (2014). Accelerating abc methods using gaussian processes. In *Artificial Intelligence and Statistics*, pages 1015–1023.

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