Learning Subspace Conditional Embedding Operators

Gregor H. W. Gebhardt GEBHARDT@IAS.TU-DARMSTADT.DE
Department of Computer Science, Computational Learning for Autonomous Systems (CLAS),
Technische Universität Darmstadt, Darmstadt, GERMANY

Andras Kupcsik KUPCSIK@COMP.NUS.EDU.SG
Department of Computer Science, School of Computing, National University of Singapore, SINGAPORE

Gerhard Neumann NEUMANN@IAS.TU-DARMSTADT.DE
Department of Computer Science, Computational Learning for Autonomous Systems (CLAS),
Technische Universität Darmstadt, Darmstadt, GERMANY

Abstract

Estimating and predicting partially observable states of a high-dimensional and highly stochastic system is still a challenging problem in machine learning and robotics. Recently, kernel methods for nonparametric inference (Song *et al.*, 2013) have been introduced which allow belief propagation with arbitrary probability distributions. However, one of the main limiting factors is that the provided algorithms scale cubically with the number of samples in the kernel matrices. In this paper, we present an extension to these nonparametric methods for inference that uses only a subset of the samples for the state representation, while still using the full data set for learning the conditional operators. Our approach is able to significantly reduce the learning and run time of the algorithm, while maintaing or even improving the performance.

1. Introduction

Learning from partial and noisy, potentially high-dimensional data is an ubiquitous problem in machine learning and robotics. Examples of such problems are poor and incomplete sensory data of a robot or occlusions in a scene captured by a low-cost camera. In order to achieve good estimates and predictions of a system's state from these incoming data streams, we need accurate forward models of the system's dynamics. Since these models are generally hard to obtain analytically, learning them from observed data is an attractive alternative.

A well known method for state estimation and prediction is the Kalman filter (KF) (Kalman, 1960) for linear models. There are extensions for non-linear models such as the extended Kalman filter (EKF) (Julier and Uhlmann, 1997) or the unscented Kalman filter (UKF) (Wan and Van Der Merwe, 2000) which rely on local linearizations or sample-based approximations with a known model. To perform state estimation and prediction with models learned from data, Gaussian processes can be applied (Deisenroth *et al.*, 2015). Though, this method requires deterministic approximate inference techniques which are computationally expensive and, in addition, scales poorly to high-dimensional observations.

To overcome these problems, Song *et al.* (2013) recently introduced a *framework for nonparametric inference in graphical models*. This framework is based on the embedding of probability distributions into reproducing kernel Hilbert spaces (RKHS) (Smola *et al.*, 2007; Baker, 1973; Song *et al.*, 2013). With the kernel space analogs of the sum rule, the chain rule (Song *et al.*, 2013) and, as a combination of these, the Bayes' rule (Fukumizu *et al.*, 2013), it is possible to perform inference on arbitrary probability distributions. Moreover, the representation inherently allows one to learn the required models from observed time series. Yet, this framework has the severe disadvantage that the computation time scales cubically with the number of sample points used for learning.

In this paper, we propose a solution to this problem by introducing a conditional operator in a kernel subspace. While only a subset of the kernel samples is used to represent the embedded probability distribution, we make use of the whole data set to estimate the transition and observation models. Hence, our algorithm can obtain improved estimation and prediction performance, while scaling linearly with the number of samples in the training set for learning and performing inference in constant time.

Similar approaches exist for Gaussian processes (Snelson

and Ghahramani, 2006; Seeger *et al.*, 2003; Smola and Bartlett, 2001; Csató and Opper, 2002), which result in a kernel function that incorporates the full data set, yet performs the computations in a lower dimensional space spanned by a sparse reference set. However, this specific design of the kernel function restricts the set of computations available in the subspace, since kernel evaluations of new data points always require the full data set.

2. Nonparametric Inference with Hilbert Space Embeddings

In this section, we will review the embeddings of probability densities into reproducing kernel Hilbert spaces (Smola *et al.*, 2007; Song *et al.*, 2013), as well as the kernel analogs of the sum rule, the product rule (Song *et al.*, 2013), and Bayes' rule (Fukumizu *et al.*, 2013).

For now, we consider two random variables X and Y on the domains Ω_X and Ω_Y and refer to their variates as x and y, respectively. $P(X)$ is the probability distribution over the random variable X. For the filtering application, we will later consider the states as random variates y and observations as random variates x.

A *reproducing kernel Hilbert space (RKHS)* is a Hilbert space of functions $f : \Omega \to \mathbb{R}$, uniquely defined by a positive definite kernel function $k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ (Aronszajn, 1950). The kernel function implicitly defines the feature mapping ϕ, which might be infinite dimensional, and the inner product of the Hilbert space $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The elements of $\mathcal{H}$ can be reproduced by the kernel function k, i.e., $f(\cdot) = \sum_{i=1}^{m} \alpha_i k(x_i, \cdot)$.

We assume two reproducing kernel Hilbert spaces $\mathcal{H}_X$ and $\mathcal{H}_Y$ with kernel functions $k : \Omega_X \times \Omega_X \to \mathbb{R}$ and $g : \Omega_Y \times \Omega_Y \to \mathbb{R}$, respectively, where $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}_X}$ and $g(y, y') = \langle \varphi(y), \varphi(y') \rangle_{\mathcal{H}_Y}$.

Embedding of a Marginal Distribution The kernel-embedding of a marginal distribution $P(X)$ of the random variable X is the expected feature mapping (*mean map*) of its random variates (Smola *et al.*, 2007) $\mu_X := \mathbb{E}_X [\phi(X)] = \int_\Omega \phi(x) \, \mathrm{d}p(x)$. The mean map can be estimated from a finite sample set as

$$\hat{\mu}_X = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) = \frac{1}{m} \sum_{i=1}^{m} k(x_i, \cdot). \tag{1}$$

In general, given a set of feature mappings $\Phi = [\phi(x_1), \ldots, \phi(x_m)]$, any distribution $q(x)$ over the domain of X may be embedded as a linear combination of these feature mappings by $\hat{\mu}_X^q = \Phi\beta$, with a column weight vector β.

Embedding of a Joint Distribution The kernel-embedding of a joint distribution $P(X, Y)$ (Baker, 1973; Smola *et al.*, 2007) of two random variables X and Y is defined as the expected tensor product of the feature mappings $\mathcal{C}_{XY} := \mathbb{E}_{XY} [\phi(X) \otimes \varphi(Y)]$. The corresponding finite sample estimator is here

$$\hat{\mathcal{C}}_{XY} = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) \otimes \varphi(y_i) = \frac{1}{m} \Phi \Upsilon^\intercal, \tag{2}$$

with the feature matrices $\Phi = [\phi(x_1), \ldots, \phi(x_m)]$ and $\Upsilon = [\varphi(y_1), \ldots, \varphi(y_m)]$. The embedding of the joint distribution is also called the *cross-covariance operator*.

Embedding of a Conditional Distribution Similar to the embedding of a marginal distribution, the embedding of a conditional distribution $P(X|Y)$ (Song *et al.*, 2013) is defined as $\mu_{Y|x} := \mathbb{E}_{Y|x} [\varphi(Y)] = \int_{y \in \Omega_Y} \varphi(y) p(y|x) \, \mathrm{d}y$. Here, the embedding is not a single element of the RKHS but rather a family of elements. A particular element of the family is chosen by conditioning on a specific value of x. To obtain the conditional embedding for a specific value of x, Song *et al.* (2013) additionally introduced the *conditional embedding operator* $\mathcal{C}_{Y|X}$ as

$$\mu_{Y|x} = \mathcal{C}_{Y|X} \phi(x). \tag{3}$$

Based on a relation from (Fukumizu *et al.*, 2004), they obtain the conditional operator as

$$\mathcal{C}_{Y|X} = \mathcal{C}_{YX} \mathcal{C}_{XX}^{-1}, \tag{4}$$

and derive its finite sample estimator as

$$\hat{\mathcal{C}}_{Y|X} = \Upsilon (\mathbf{K} + \lambda \mathbf{I}_m)^{-1} \Phi^\intercal, \tag{5}$$

with the feature matrices $\Upsilon := (\varphi(y_1), \ldots, \varphi(y_m))$ and $\Phi := (\phi(x_1), \ldots, \phi(x_m))$, the Gram matrix $\mathbf{K} = \Phi^T \Phi \in \mathbb{R}^{m \times m}$, and regularization parameter λ.

The Kernel Sum Rule Given a joint distribution $P(X, Y)$, the sum rule computes the marginal distribution $P(X)$ by integrating out variable Y. By factorizing $P(X, Y)$ into $P(X|Y)\pi(Y)$ a prior distribution π of Y can be taken into account which is in general different from the distribution $P(Y)$ observed in the training set and might additionally be represented using a different sample set $\{\tilde{y}_1, \ldots, \tilde{y}_{\tilde{m}}\}$ with weights $\boldsymbol{\alpha}$. Song *et al.* (2013) derive the kernel sum rule by embedding this factorization into the RKHS as

$$\begin{aligned} \mu_X^\pi &= \mathbb{E}_Y \mathbb{E}_{X|Y} [\phi(X)] = \mathbb{E}_Y \left[\mathcal{C}_{X|Y} \varphi(Y) \right] \\ &= \mathcal{C}_{X|Y} \mathbb{E}_Y [\varphi(Y)] = \mathcal{C}_{X|Y} \mu_Y^\pi \\ &= \Upsilon \left(\boldsymbol{G} + \lambda \boldsymbol{I}_m \right)^{-1} \tilde{\boldsymbol{G}} \boldsymbol{\alpha}, \end{aligned} \tag{6}$$

with the Gram matrices $\boldsymbol{G} = \Upsilon^T\Upsilon \in \mathbb{R}^{m\times m}$ and $\tilde{\boldsymbol{G}} = \Upsilon^T\tilde{\Upsilon} \in \mathbb{R}^{m\times\tilde{m}}$. The superscript π denotes that the mean map μ_X^π is conditioned on the prior distribution $\pi(Y)$. In addition, Song *et al.* (2013) also provide a kernel sum rule for the tensor product features as

$$\mathcal{C}_{XX}^\pi = \Upsilon \mathrm{diag}\left(\left(\boldsymbol{G} + \lambda \boldsymbol{I}_m\right)^{-1}\tilde{\boldsymbol{G}}\boldsymbol{\alpha}\right)\Upsilon^\intercal, \tag{7}$$

where $\mathcal{C}_{XX}^\pi$ is the *prior modified covariance operator*.

The Kernel Chain Rule Given a conditional distribution $P(X|Y)$ and a marginal prior distribution $\pi(Y)$, the chain rule computes the joint distribution $Q(X,Y) = P(X|Y)\pi(Y)$. By embedding the marginal probability as $\mathcal{C}_{YY}^\pi$ in a tensor product RKHS and then applying the conditional embedding operator, (Song *et al.*, 2013) derived the kernel chain rule as

$$\mathcal{C}_{XY}^\pi = \mathcal{C}_{X|Y}\mathcal{C}_{YY}^\pi = \Upsilon\left(\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\tilde{\boldsymbol{G}}\mathrm{diag}\left(\boldsymbol{\alpha}\right)\Phi^\intercal. \tag{8}$$

Alternatively, the kernel chain rule can also be applied to the mean embedding μ_Y by making use of the conditional cross-covariance operator (Song *et al.*, 2013; Fukumizu *et al.*, 2004) which results in

$$\begin{aligned}\mathcal{C}_{XY}^\pi &= \mathcal{C}_{(XY)|Y}\mu_Y^\pi \\ &= \Upsilon\mathrm{diag}\left(\left(\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\tilde{\boldsymbol{G}}\boldsymbol{\alpha}\right)\Phi^\intercal,\end{aligned} \tag{9}$$

where $\mathcal{C}_{XY}^\pi$ is the *prior modified cross-covariance operator*.

Kernel Bayes' Rule Given a prior distribution $\pi(Y)$ and a likelihood function $P(X|Y)$, Bayes' rule computes the posterior distribution $P(Y|x)$ of Y given an instance x of X. Fukumizu *et al.* (2013) derived the kernel Bayes' rule (KBR) with the prior modified covariance operator $\mathcal{C}_{XX}^\pi$, obtained from the kernel sum rule, and the prior modified cross-covariance operator $\mathcal{C}_{YX}^\pi$, obtained from the kernel chain rule, similar to the conditional operator as

$$\mu_{Y|x}^\pi = \mathcal{C}_{Y|X}^\pi\phi(x) = \mathcal{C}_{YX}^\pi\left(\mathcal{C}_{XX}^\pi\right)^{-1}\phi(x). \tag{10}$$

By applying the finite sample estimates of the kernel sum rule and the kernel chain rule, they arrive at

$$\begin{aligned}\boldsymbol{D} &= \mathrm{diag}\left(\left(\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\tilde{\boldsymbol{G}}\boldsymbol{\alpha}\right) \\ \mu_{Y|x}^\pi &= \mathcal{C}_{YX}^\pi\left(\mathcal{C}_{XX}^\pi\right)^{-1}\phi(x) \\ &= \mathcal{C}_{YX}^\pi\left(\left(\mathcal{C}_{XX}^\pi\right)^2 + \kappa\boldsymbol{I}\right)^{-1}\mathcal{C}_{XX}^\pi\phi(x) \qquad (11) \\ &= \tilde{\Phi}\boldsymbol{D}\boldsymbol{K}\left(\left(\boldsymbol{D}\boldsymbol{K}\right)^2 + \kappa\boldsymbol{I}\right)^{-1}\boldsymbol{D}\boldsymbol{K}_{:x}, \qquad (12)\end{aligned}$$

with the kernel vector $\left(\boldsymbol{K}_{:\bar{x}}\right)_i = k(x_i,\bar{x})$ of the observation $\bar{x}$. Since the weights α_i can be negative, Fukumizu *et al.* (2013) make use of the Tikhonov regularization for the inversion of $\mathcal{C}_{XX}^\pi$ in Equation 11.

3. Efficient Nonparametric Inference in a Subspace

A pervasive problem of kernel methods is the trade-off between accuracy and computational efficiency. On the one hand, large sample sets are a severe computational problem, especially due to the inversion of the Gram matrix which is in $O(m^3)$. On the other hand, we demand large sample sets due to two reasons. First, we want representative sample sets that cover a large range of the problem domain and still provide a reasonable accuracy. Second, to get good estimations of the conditional operators for highly stochastic systems, a large number of transitions and thus samples is required. Since the second motivation is more important for highly stochastic systems, we want to use only a representative subset for the mean embedding, while still using the entire sample set for estimating the conditional operators.

3.1. The Subspace Conditional Operator

In this paper, we introduce the *subspace conditional operator* which maintains computational efficiency by reducing the operator size with an appropriate sparsification technique, while still using the whole set of training samples for learning the conditional operator. Based on the sample sets $\Phi = \{\phi(x_1),\ldots,\phi(x_m)\}$ and $\Upsilon = \{\varphi(y_1),\ldots,\varphi(y_m)\}$ introduced in Section 2, we define the respective subsets $\Gamma \subset \Upsilon$ and $\Psi \subset \Phi$, with $|\Gamma| = |\Psi| = n \ll m$, and assume these subsets to be sufficient for representing the mean embeddings. While the subspace conditional operator $\mathcal{C}_{X|Y}^S$ applied to an embedding $\phi(x) \in \mathcal{H}_X$ gives the mean embedding $\mu_{Y|x} \in \mathcal{H}_Y$, we can derive it by first defining an auxiliary conditional operator $\tilde{\mathcal{C}}_{Y|X}^S$ as

$$\mu_{Y|x} = \tilde{\mathcal{C}}_{Y|X}^S\Psi^\intercal\phi(x), \tag{13}$$

that maps from the subspace projected embedding $\Psi^\intercal\phi(x)$ to the mean embedding $\mu_{Y|x}$. We can then obtain this auxiliary conditional operator by minimizing the squared error

$$\begin{aligned}0 &= \frac{\partial}{\partial\tilde{\mathcal{C}}_{X|Y}}\left\|\Upsilon - \tilde{\mathcal{C}}_{X|Y}\Psi^\intercal\Phi\right\|_2 \\ 0 &= -2\left(\Upsilon - \tilde{\mathcal{C}}_{X|Y}\Psi^\intercal\Phi\right)\Phi^\intercal\Psi \\ \tilde{\mathcal{C}}_{X|Y}^S\Psi^\intercal\Phi\Phi^\intercal\Psi &= \Upsilon\Phi^\intercal\Psi \\ \tilde{\mathcal{C}}_{X|Y}^S &= \Upsilon\Phi^\intercal\Psi\left(\Psi^\intercal\Phi\Phi^\intercal\Psi + \lambda\boldsymbol{I}\right)^{-1}\end{aligned}$$

and obtain the subspace conditional operator as

$$\mathcal{C}_{X|Y}^S = \tilde{\mathcal{C}}_{X|Y}^S\Psi^\intercal \tag{14}$$

$$= \Upsilon\Phi^\intercal\Psi\left(\Psi^\intercal\Phi\Phi^\intercal\Psi + \lambda\boldsymbol{I}\right)^{-1}\Psi^\intercal \tag{15}$$

$$= \Upsilon\bar{\boldsymbol{G}}\left(\bar{\boldsymbol{G}}^\intercal\bar{\boldsymbol{G}} + \lambda\boldsymbol{I}\right)^{-1}\Psi^\intercal, \tag{16}$$

where $\bar{\boldsymbol{G}}_{i,j} = g(\varphi(y_i), \varphi(\bar{y}_j)) \in \mathbb{R}^{m \times n}$ is the kernel matrix of the sample feature set Φ and its subset Ψ. Since we assume that $n \ll m$, the inverse in the subspace conditional operator is in $\mathbb{R}^{n \times n}$ and, thus, of a much smaller size than the inverse in the standard conditional operator shown in Equation 5. Additionally, we can exploit the feature matrix $\Psi^\intercal$ on the right hand side of the subspace conditional operator, and represent the state estimate always in the subspace. Hence, we are able to completely avoid representations and computations in the high-dimensional space spanned by the full sample set.

In the following sections we will, analogously to (Song *et al.*, 2013; Fukumizu *et al.*, 2013), use the subspace conditional operator to derive the subspace versions of the kernel sum rule, the kernel chain rule and the kernel Bayes' rule.

3.2. The Subspace Kernel Sum Rule

For the kernel sum rule, (Song *et al.*, 2013) applied the conditional operator to the mean map of a distribution $\pi(Y)$. Analogously, the subspace kernel sum rule for a marginal mean map becomes

$$\mu_X^\pi = \mathcal{C}_{X|Y}^S \mu_Y^\pi = \Upsilon \bar{\boldsymbol{G}} \left(\bar{\boldsymbol{G}}^\intercal \bar{\boldsymbol{G}} + \lambda \boldsymbol{I}\right)^{-1} \tilde{\bar{\boldsymbol{G}}}^\intercal \boldsymbol{\alpha}, \quad (17)$$

where $\mu_Y^\pi = \tilde{\Phi}\boldsymbol{\alpha}$ is the embedding of the prior distribution $\pi(Y)$ that is in general represented with a different sample set $\tilde{\Phi}$ which results in the kernel matrix $\tilde{\bar{\boldsymbol{G}}} = \tilde{\Phi}^\intercal \Psi$. In contrast to Song *et al.* (2013), who construct the kernel sum rule for tensor product features by applying the conditional operator to the embedding μ_Y^π and then construct a covariance operator with the conditioned weights $\boldsymbol{\alpha}'$, we first construct the covariance operator and then apply the conditional operator to both sides as

$$\begin{aligned}\mathcal{C}_{XX}^{S,\pi} &= \mathcal{C}_{X|Y}^S \mathcal{C}_{YY}^\pi (\mathcal{C}_{X|Y}^S)^\intercal = \mathcal{C}_{X|Y}^S \Phi \operatorname{diag}(\boldsymbol{\alpha}) \Phi^\intercal (\mathcal{C}_{X|Y}^S)^\intercal \\ &= \Upsilon \bar{\boldsymbol{G}} \boldsymbol{L} \tilde{\bar{\boldsymbol{G}}}^\intercal \operatorname{diag}(\boldsymbol{\alpha}) \tilde{\bar{\boldsymbol{G}}} \boldsymbol{L} \bar{\boldsymbol{G}}^\intercal \Upsilon^\intercal, \end{aligned} \quad (18)$$

where $\mathcal{C}_{YY}^\pi$ is the embedding of the prior distribution $\pi(Y)$ as covariance operator and the Tikhonov regularized inverse $\boldsymbol{L} = (\bar{\boldsymbol{G}}^\intercal \bar{\boldsymbol{G}} + \lambda \boldsymbol{I})^{-1} \in \mathbb{R}^{n \times n}$.

3.3. The Subspace Kernel Chain Rule

The kernel chain rule computes the prior modified cross-covariance operator by applying the conditional operator to an embedding of $\pi(Y)$ in a tensor product RKHS. The subspace kernel chain rule is a straight forward modification of the kernel chain rule of (Song *et al.*, 2013), i.e.,

$$\begin{aligned}\mathcal{C}_{YX}^{S,\pi} &= \mathcal{C}_{Y|X}^S \mathcal{C}_{XX}^\pi \\ &= \Upsilon \bar{\boldsymbol{G}} \left(\bar{\boldsymbol{G}}^\intercal \bar{\boldsymbol{G}} + \lambda \boldsymbol{I}\right)^{-1} \tilde{\bar{\boldsymbol{G}}}^\intercal \operatorname{diag}(\boldsymbol{\alpha}) \tilde{\Phi}^\intercal. \end{aligned} \quad (19)$$

With the subspace kernel sum rule and the subspace kernel chain rule we can now construct the subspace kernel Bayes' rule.

3.4. The Subspace Kernel Bayes' Rule

Analogous to Fukumizu *et al.* (2013), we construct the subspace kernel Bayes' rule (subKBR) with the prior modified covariance operator from the subspace kernel sum rule and the prior modified from the subspace kernel chain rule as

$$\mu_{Y|x}^\pi = \mathcal{C}_{YX}^{S,\pi} \left(\left(\mathcal{C}_{XX}^{S,\pi}\right)^2 + \gamma \boldsymbol{I} \right)^{-1} \mathcal{C}_{XX}^{S,\pi} \phi(x). \quad (20)$$

By inserting the definitions from Equations 18 and 19 and applying the matrix identity $\boldsymbol{A}\left(\boldsymbol{B}\boldsymbol{A} + \lambda \boldsymbol{I}\right)^{-1} = \left(\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{A}$, we can define the following matrices

$$\boldsymbol{E} := \bar{\boldsymbol{G}}^\intercal \Upsilon^\intercal \Upsilon \bar{\boldsymbol{G}} = \bar{\boldsymbol{G}}^\intercal \boldsymbol{K} \bar{\boldsymbol{G}} \in \mathbb{R}^{n \times n}, \quad (21)$$

$$\boldsymbol{D} := \boldsymbol{L} \tilde{\bar{\boldsymbol{G}}}^\intercal \operatorname{diag}(\boldsymbol{\alpha}) \tilde{\bar{\boldsymbol{G}}} \boldsymbol{L} \in \mathbb{R}^{n \times n}, \quad (22)$$

and arrive at

$$\mu_{Y|x}^\pi = \Phi \operatorname{diag}(\alpha) \tilde{\bar{\boldsymbol{G}}} \boldsymbol{L} \boldsymbol{E} \boldsymbol{D} \left((\boldsymbol{E}\boldsymbol{D})^2 + \gamma \boldsymbol{I} \right)^{-1} \bar{\boldsymbol{G}}^\intercal \boldsymbol{K}_{:x}, \quad (23)$$

with $(\boldsymbol{K}_{:x})_i = k(x_i, x^*)$ the kernel vector of the new observation x^*. Since $\boldsymbol{E}$ and $\boldsymbol{D}$ are both in $\mathbb{R}^{n \times n}$, the matrix inversion is only in $O(n^3)$. The whole kernel Bayes' rule is in $O(mn^2)$ and, thus, scales linearly with the number of sample points (given a fixed reference set) instead of cubically as for the original kernel Bayes' rule.

4. Experimental Results

We compare the subspace kernel Bayes' rule (subKBR) to the standard KBR with two experiments on simulated data. In both experiments, we used the respective KBRs and conditional operators to perform kernel Bayes' filtering (KBF and subKBF) (or prediction) (Fukumizu *et al.*, 2013) on the observations. We map from the Hilbert space to the state space using a linear mapping similar to Zhu *et al.* (2014) We will use the term *training set* to talk about the set of samples that is used to learn the conditional operators and *subspace set* to talk about the set of samples that is used for the subspace projection.

Synthetic Data For the first experiment, we simulate a pendulum which we randomly initialize in the ranges $[0.1\pi, 0.4\pi]$ and $[-0.5\pi, 0.5\pi]$ for the angle θ and the angular velocity $\dot{\theta}$, respectively. The pendulum has a mass of 5kg and a friction coefficient of 1. The angular velocities are subject to Gaussian process noise $\xi \sim \mathcal{N}(0, 1)$ and the states are observed with Gaussian observation noise $\eta \sim \mathcal{N}(0, 0.1)$. Additionally, the observed angles are randomly perturbed by an offset of $\frac{\pi}{4}$. These random perturbations occur with a probability of 0.1 in every time step. Each episode consists of 30 time steps with $\Delta t = 0.1$.

We use a squared exponential kernel for the states as well as the observations, where we apply the median trick to select the bandwidths. The regularization parameters are set to $\lambda_T = \exp(-1)$ for the transition operator, $\lambda_O = \exp(-6)$ for the observation operator and $\gamma = \exp(-6)$ for the kernel Bayes' rule. We simulate 200 episodes to form a training set and choose the kernel samples randomly. We conduct the experiment for 100, 150, 300, 450 and 600 kernel samples in a randomly selected training set and fixed the size of the subspace set to 100 samples.

Figures 1, 2, and 3 show the results of this experiment in terms of performance, learning time, and run time, respectively. In Figure 1, we see that the subspace KBF has a slightly better performance when the training set equals the subspace set and maintains the performance of the standard KBF with an increasing number of training samples while the subspace set is fixed. Figures 2 and 3 show the improvement of efficiency for learning and filtering of the subKBR over the standard KBR.

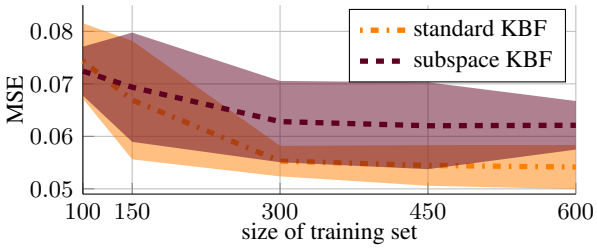

Figure 1. Comparison of the standard KBF to the subspace KBF. The subspace KBF is learned with a subspace set of 100 samples. Depicted are the median and the [0.25,0.75] quantiles over 20 evaluations.

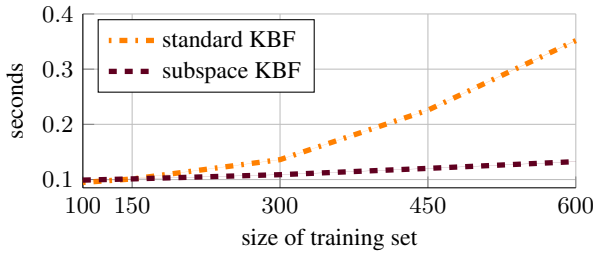

Figure 2. Evaluation of the training time of the KBF and the subKBF. Depicted is the median over 20 evaluations.

Video Frames In the second experiment, we filtered the frames of a video stream consisting of 30 frames. We use the same simulated pendulum as described in the previous paragraph. Here, we apply process noise $\xi \sim \mathcal{N}(0, 2)$ as well as the random pertubations to the pendulum angles, and render the pendulum movements into video frames with a width and height of 10 pixels. Finally, we project the frames into the space spanned by the first ten principal componentents of the training video data and use these projections as observations. Again, we use the squared exponential kernel, where the bandwiths are set to the median distance of the data points. The regularization parameters are set to $\lambda_T = \exp(-10)$ for the transition operator, $\lambda_O = \exp(-10)$ for the observation operator and $\gamma = \exp(0.8)$ for the kernel Bayes' rule. Similar to Song *et al.* (2010), we normalize $\boldsymbol{\alpha}$ to a maximum distance between the minimal and maximal value of 1 for numerical stability.

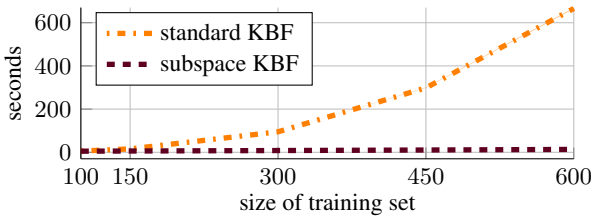

Figure 3. Evaluation of the run time of the KBF and the subKBF for filtering 30 episodes with each 30 steps. Depicted is the median over 20 evaluations.

We use a dataset of 100 episodes and conduct the experiment for a reference size of 100, 200 and 300 samples. The subspace conditional operator is always trained with training set of 1500 samples. Figure 4 shows the mean squared error of the angles extracted from the filtered video data to the groundtruth. We can see that the subspace kernel Bayes' filter already reaches better results with a small kernel size and maintains its performance with an increasing number of samples in the reference set.

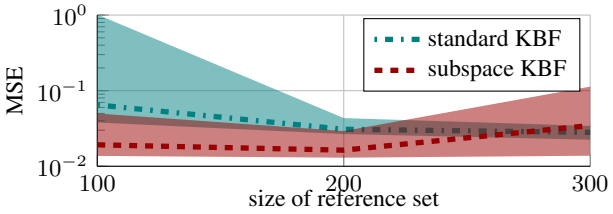

Figure 4. Comparison of the KBF and subKBF for filtering high-dimensional video frames. The subKBF is for all evaluations trained with 1500 sample points. Depicted are the median and the [0.25,0.75] quantiles over 20 evaluations.

5. Conclusions

In this paper, we presented a new formulation of the conditional embedding operator, called the *subspace conditional operator*. This formulation enables us to represent embeddings of probability distributions in a subspace of the kernel samples, while still using the whole sample set for learning the operators. We showed that the subspace conditional operator outperforms the standard conditional operator in terms of performance, learning time and run time.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreements #610967, #270327 and from the European Union's Horizon 2020 research and innovation programme under grant agreement #645582.

References

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, pages 337–404, 1950.

Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):408–423, Feb 2015.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *The Journal of Machine Learning Research*, 5:73–99, 2004.

Kenji Fukumizu, Le Song, and Arthur Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783, 2013.

Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression, 2003.

Alex J. Smola and Peter Bartlett. Sparse greedy gaussian process regression. In *Advances in Neural Information Processing Systems 13*, pages 619–625. MIT Press, 2001.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *In Algorithmic Learning Theory: 18th International Conference*, pages 13–31. Springer-Verlag, 2007.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.

Le Song, Byron Boots, Sajid M Siddiqi, Geoffrey J Gordon, and Alex J Smola. Hilbert space embeddings of hidden markov models. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 991–998, 2010.

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

Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

Pingping Zhu, Badong Chen, and Jose C Principe. Learning nonlinear generative models of time series with a kalman filter in rkhs. *Signal Processing, IEEE Transactions on*, 62(1):141–155, 2014.