Generative and Latent Mean Map Kernels

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce two kernels that extend the mean map, which embeds distributions in Hilbert spaces. The generative mean map kernel (MMK) measures similarity between probabilistic models of structured data such as sequences. The latent mean map kernel extends the non-iid data formulation of the empirical mean map to handle latent variable models. We present classification results on synthetic and DNA data, comparing support vector machines (SVMs) using these two kernels to a Bayes classifier and SVMs using other generative kernels. The generative MMK outperformed all other methods, while the latent MMK was competitive for the synthetic data. We also demonstrate the generative MMK as a similarity measure between kernel density estimators for a manifold visualization of biodiversity data.

1 Introduction

Generative kernels offer an elegant way to apply kernel methods for classification, clustering, and manifold learning to structured data. By using kernels to address structured data's lack of the independently and identically distributed (iid) property, we can leverage a rich set of existing methods such as support vector machines (SVMs) and kernel principal components analysis (kPCA) [1] to learn from this data. For example, we consider sequence classification as a particular instance of learning with structured data. In sequence classification, the goal is to label a sequence $(X_1, X_2, \ldots, X_T)$ with one label Y, where the example sequences can be of varying lengths and the ordering of the observations within a sequence informs a non-trivial dependence structure.

SVMs using nonlinear kernels such as the polynomial and Gaussian kernels have performed strongly for a variety of non-sequential data classification tasks [2]. More recent work has applied kernels to measure similarity between sequences via similarity between generative models trained on those sequences [3] or by making use of metrics on statistical manifolds [4, 5]. Missing from this spectrum of kernels is a measure of similarity that combines the power of highly nonlinear kernels such as the Gaussian kernel with the structural information provided by a generative model.

Our contribution in this work is to show two data-driven kernels, the generative mean map kernel and the latent mean map kernel (MMK), that yield this combination. The generative MMK measures similarity between two structured data observations by providing a nonlinear similarity between generative models estimated from each observation. The latent MMK measures similarity between two structured data observations with respect to a generative model θ. This is accomplished by measuring the similarity of sufficient empirical and posterior distributions from two structured data observations. We note that the empirical MMK previously was used in various applications to measure and optimize dependence [6, 7, 8], but so far it has not been used for structured data classification.

We now review Hilbert space embeddings of distributions, which are key in deriving the generative and latent MMKs.

The mean map. The mean map was introduced with respect to Hilbert space embeddings of distributions [9]. For $\mathcal{X}$ a domain of observations with probability measure P_x and $X = \{x_1, \ldots, x_m\}$ a set of m samples drawn iid, consider the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with feature map $\phi : \mathcal{X} \mapsto \mathcal{H}$, for $\phi(x) = k(x, \cdot)$, $\langle f, \phi(x) \rangle = f(x)$ and $\langle \phi(x), \phi(y) \rangle = k(x, y)$. The mean map μ of the true and sample distributions respectively are defined as $\mu[P_x] := \mathsf{E}_x[k(x, \cdot)]$ and $\mu[X] := \frac{1}{m} \sum_{i=1}^m k(x_i, \cdot)$. The operator μ maps distributions to elements of the RKHS and has the useful property of being injective for RKHSs induced by universal kernels [10]. Two examples of universal kernels are the Gaussian radial basis function (RBF) kernel $k(x, y) = e^{-\lambda \|x-y\|^2}$ and the Laplace kernel $k(x, y) = e^{-\beta \sum_{i=1}^n |x_i - y_i|}$ for $\lambda, \beta \in \mathbb{R}^+$.

Previous work [6, 7] has exploited the linear convergence of $\mu[X]$ to $\mu[P_x]$ in order to compute kernels between two sets of samples; the methods introduced here are departures from these applications of the MMK. In Section 2 we show how to compute the generative MMK for several widely-used distributions, and we relate this kernel to other methods in Section 3. We then discuss the latent MMK, an extension of the non-iid empirical MMK for latent models. We conclude with promising results on sequence classification and learning a species manifold from biodiversity data.

2 Generative mean map kernel

Suppose we are given two objects x and y. In general, these objects could be documents, sequences, images, or unstructured points in $\mathbb{R}^n$. For all but the last case, where standard kernels on vectors can be used, special effort is necessary to form a suitable kernel between the objects that captures their underlying similarity well. As we are armed with a kernel on distributions, let us learn a generative model that describes each object, yielding distributions $\hat{P}_x$ and $\hat{P}_y$ respectively. We refer to the mean map kernel using these generative models as the generative mean map kernel, which is defined as

$$\langle \mu[\hat{P}_x], \mu[\hat{P}_y] \rangle = \mathsf{E}_{x \sim \hat{P}_x, y \sim \hat{P}_y}[k(x, y)] = \int \int \hat{P}_x(x) \hat{P}_y(y) k(x, y) dx\, dy.$$

Proposition 1. *The generative mean map kernel with positive definite kernel is also positive definite.*

Proof. The generative MMK is an inner product between mean elements in $\mathcal{H}$, where a mean element has the form $\mu[p] = \int p(x)\phi(x)dx$ for probability measure p. Since $k(\cdot, \cdot)$ is pd, $\forall x \in \mathcal{X}$, $\exists \phi(x) \in \mathcal{H}$. Since $\mathcal{H}$ is convex the mean element also is $\mathcal{H}$, and so the generative MMK is pd. □

Let us explore this kernel for several interesting generative models of increasing structure.

Gaussian distribution.

Proposition 2. *Let p and p' be multivariate Gaussian probability measures $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$ respectively. For the Gaussian RBF kernel $k(x, x') = e^{-\frac{1}{2}\lambda \|x-x'\|^2}$, the MMK for p and p' is*

$$k_{\mathrm{mm}}(p, p') = \int \int \frac{e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}}{(2\pi)^{d/2}|\Sigma|^{1/2}} \frac{e^{-\frac{1}{2}(x'-\mu')^T \Sigma'^{-1}(x'-\mu')}}{(2\pi)^{d/2}|\Sigma'|^{1/2}} k(x, x') dx\, dx'$$

which can be computed in closed form as[1]

$$\frac{e^{-\frac{1}{2}(\beta^T \alpha^{-1} \beta - \delta)}}{|I + \lambda(\Sigma + \Sigma')|^{1/2}}, \tag{1}$$

where $\alpha = \Sigma^{-1}(\Sigma^{-1} + \lambda I)^{-1}\Sigma^{-1} + \Sigma'^{-1} + \Sigma^{-1}$, $\beta = \lambda\Sigma^{-1}(\Sigma^{-1} + \lambda I)^{-1}\mu + \Sigma'^{-1}\mu'$, *and* $\delta = -\lambda^2 \mu^T (\Sigma^{-1} + \lambda I)^{-1}\mu + \mu'^T \Sigma'^{-1}\mu' + \lambda \mu^T \mu$.

The proof follows from straightforward linear algebra.

Discrete distribution. The generative MMK between two discrete distributions arises for the generative MMK between hidden Markov models. For p and p' discrete probability distributions with mean parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ and $\alpha' = (\alpha'_1, \ldots, \alpha'_k)$ respectively, the generative MMK is

$$k_{\mathrm{mm}}(p, p') = \sum_{i=1}^k \sum_{j=1}^k \alpha_i \alpha'_j e^{-\lambda(1-\delta_{ij})}. \tag{2}$$

[1] The exponential term should admit a more elegant form. We have confirmed that it is symmetric in the p and p' terms. See the simplified isotropic case in (5).

Hidden Markov models. For a hidden Markov model (HMM) with probability measure p, let $\mathbf{q} = (q_0, \ldots, q_T)$ be the latent random variables and $\mathbf{x} = (x_0, \ldots, x_T)$ be the observable random variables. We similarly define p', $\mathbf{q}'$, and $\mathbf{x}'$ for a second HMM. Now suppose that we have learned HMMs with probability measures p and p' and wish to compute the generative MMK $k_{\mathrm{mm}}(p, p')$ for the observable variables of length T sequences drawn from these HMMs. Note that the parameter T serves as a witness length which allows control over the length of the sequences to be embedded in the RKHS. A small T can be used to yield a similarity measure more sensitive to each model's initial conditions, while increasing T shifts toward a similarity measure on their stationary distributions. The following proposition establishes the complexity of an efficient algorithm to compute the kernel.

Proposition 3. *The generative MMK between an HMM of n states with probability measure p and an HMM of n' states with probability measure p' can be computed in time:*

1. *$O(n^3T + n^2k^2)$ for a discrete observation HMM on k symbols with $n' = n$.*
2. *$O(n^3T + n^2m^2\rho_d)$ for a continuous observation HMM with mixture of Gaussians state distributions, $n' = n$, and $m' = m$, for d the observation dimensionality, m and m' the number of Gaussians in the mixtures, and ρ_d the cost of inverting a covariance matrix[2].*

Proof. The quantity of interest is $k_{\mathrm{mm}}(p, p') = \sum_{\mathbf{x},\mathbf{x}',\mathbf{q},\mathbf{q}'} p(\mathbf{q}, \mathbf{x})p'(\mathbf{q}', \mathbf{x}')k(\mathbf{x}, \mathbf{x}')$. In order to treat the discrete and continuous observation cases simultaneously, for the discrete case we use the 1-of-k encoding such that, if the t^{th} observation takes on value i out of k possible values, then $x_t \in \{0, 1\}^k$ and $[x_t]_j = \delta_{ij}, \forall j \in [k]$. In the derivation for the discrete case below[3], we use the Gaussian RBF kernel's isotropicity such that it factorizes as $k(\mathbf{x}, \mathbf{x}') = \prod_{t=1}^{T} k(x_t, x'_t)$. Also useful is the linear chain structure of the HMM graphical model to factorize the expectation as shown:

$$
\begin{aligned}
k_{\mathrm{mm}}(p, p') = & \sum_{q_T, x_T} p(x_T \mid q_T) \sum_{q'_T, x'_T} p'(x'_T \mid q'_T) k(x_T, x'_T) \\
& \prod_{t=0}^{T-1} \sum_{q_t, q'_t} p(q_{t+1} \mid q_t) p'(q'_{t+1} \mid q'_t) \sum_{x_t, x'_t} p(x_t \mid q_t) p'(x'_t \mid q'_t) k(x_t, x'_t) p(q_0) p'(q'_0) \\
= & \sum_{q_T, q'_T} \psi(q_T, q'_T) \prod_{t=0}^{T-1} \sum_{q_t} p(q_{t+1} \mid q_t) \sum_{q'_t} p'(q'_{t+1} \mid q'_t) \psi(q_t, q'_t) p(q_0) p'(q'_0). \qquad (3)
\end{aligned}
$$

Note that $\psi(q_t, q'_t) = \sum_{x_t, x'_t} p(x_t \mid q_t) p'(x'_t \mid q'_t) k(x_t, x'_t)$ is itself a generative MMK on the state distributions for q_t and q'_t. These kernels need be computed only once for each pair of states (q_t, q'_t), yielding cost n^2 times the complexity to evaluate this kernel once, which is $O(k^2)$ for the discrete distribution and $O(m^2\rho_d)$ for a mixture of Gaussians distribution. From the factorized structure of the rest of the computation in (3) we see that it is $O(Tn^3)$ (see Algorithm 1 in Figure 1(a)), as all $O(T)$ latent variable marginalizations are done over functions of at most 3 variables. □

In general, the sequences for which one would like a similarity measure are of different lengths, precluding computation of a kernel without resorting to truncation or other compromises. Even for sequences of the same length, *a priori* there is no reason why the sample indices of the sequences should be considered aligned. Hence, application of standard kernels invariably relies upon distance computations made between mismatched random variables. The generative MMK addresses this alignment issue head-on by first learning a generative model for each sequence and then performing kernel computations on the expected sequences that result from each generative model. While sequences drawn from similar distributions can appear to be very different due to the stochastic nature of their generation, by using a measure between the distributions of sequences themselves, we bypass this problem and achieve a more robust similarity measure.

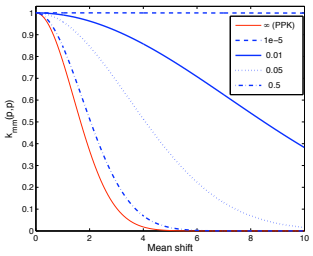
Generative mean map of kernel density estimators. The generative MMK can be used as a kernel on density estimators. The idea of kernels between density models of sets has been explored

[2]This cost depends on what choices are made to restrict the covariance structure. For the popular case of diagonal covariance matrices, we have $\rho_d = d$.

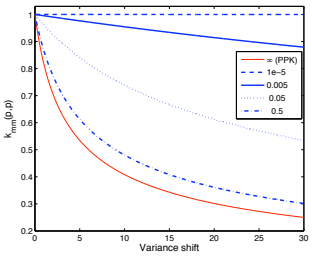
[3]For the continuous observation case, replace summations with integrals.

for $i = 1$ to n **do**
  for $j = 1$ to n' **do**
    $\boldsymbol{\psi}_{ji} = k_{\mathrm{mm}}(p(x|q=i), p(x'|q'=j))$
$\boldsymbol{\phi} = \boldsymbol{\pi}\boldsymbol{\pi}'^T$
$\boldsymbol{\phi} = \boldsymbol{\phi} \bullet \boldsymbol{\psi}$
for $t = 1$ to T **do**
  $\boldsymbol{\phi} = (\mathbf{A}'^T \boldsymbol{\phi} \mathbf{A}) \bullet \boldsymbol{\psi}$
return $\sum_{i=1}^{n} \sum_{j=1}^{n'} \boldsymbol{\phi}_{ji}$

(a) Algorithm 1



(b) MMK versus Mean Shift



(c) MMK versus Variance Shift

Figure 1: At left, we show how to compute the generative MMK for HMMs. $\bullet$ is the Hadamard product, $[\mathbf{A}]_{ij} = P(q_{t+1} = j \mid q_t = i)$, and $[\boldsymbol{\pi}]_i = P(q_0 = i)$. At center and right, for Gaussian p and p', are the sensitivities of the generative MMK $k_{\mathrm{mm}}(p, p')$ and the PPK $k_{\mathrm{pp}}(p, p')$ to mean shifts Δ_μ and variance shifts Δ_σ for different settings of λ. $p = \mathcal{N}(0, 1)$ in both plots, while $p' = \mathcal{N}(\Delta_\mu, 1)$ at center and $p = \mathcal{N}(0, 1 + \Delta_\sigma)$ at right. By increasing λ, the kernel becomes more sensitive to deviations between the two distributions, with $k_{\mathrm{mm}}(p, p') = k_{\mathrm{pp}}(p, p')$ as $\lambda \to \infty$.

previously by Jebara and Kondor [11] with the Bhattacharyya kernel . Whereas they implicitly map the data to an RKHS using the Gaussian kernel and then learn single Gaussian models in the feature space, here we use kernel density estimators in the original space. An advantage of using kernel density estimation (KDE) is that it is known to be consistent [12].

Let $\hat{f}_z(z)$ be a kernel density estimator $\hat{f}_z(x) = \frac{1}{m_z} \sum_{i=1}^{m_z} k_{h_x}(z_i, x)$, where h_x is the bandwidth. The generative MMK between two Gaussian RBF kernel density estimators $\hat{f}_x$ on observations $X = (x_1, \ldots, x_{m_x})$ and $\hat{f}_y$ on observations $Y = (y_1, \ldots, y_{m_y})$ is then

$$\langle \mu[\hat{f}], \mu[\hat{f}'] \rangle = \mathsf{E}_{\substack{x \sim \hat{f} \\ x' \sim \hat{f}'}} [k(x, x')] = \frac{1}{m_x} \frac{1}{m_y} \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} \int k_{h_x}(x_i, x) \int k_{h_y}(y_j, x') k(x, x') dx \, dx'. \quad (4)$$

For k the Gaussian RBF kernel, this expression only requires evaluations of the generative MMK on isotropic Gaussians. The form for general Gaussians is in (1). For two N-dimensional isotropic Gaussians $\mathcal{N}(\mu, hI)$ and $\mathcal{N}(\mu', h'I)$, the generative MMK admits the more pleasant form

$$\frac{1}{(1 + \lambda(h + h'))^{N/2}} \exp\left(-\frac{1}{2} \frac{\lambda \|\mu - \mu'\|^2}{1 + \lambda(h + h')} \right) = \frac{1}{h_0^{N/2}} \exp\left(-\frac{1}{2} \frac{\lambda \|\mu - \mu'\|^2}{h_0} \right), \quad (5)$$

where $h_0 := 1 + \lambda(h + h')$. Substituting (5) for the integrals in (4) yields KDE MMK closed form

$$\frac{1}{m_x m_y h_0^{N/2}} \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} \exp\left(-\frac{1}{2} \frac{\lambda \|x_i - y_i\|^2}{h_0} \right). \quad (6)$$

Controlling for uncertainty. We offer two interpretations of λ in the above kernels. First, λ controls the complexity of the mean elements in the RKHS, with maximal complexity as $\lambda \to \infty$ and minimal complexity for $\lambda = 0$. Second, λ corresponds to our degree of uncertainty with respect to the learned models. As the learned models diverge from the true models, we may decrease λ to reduce the sensitivity of the kernel to small deviations from the true model. In Figure 1, we plot how the generative MMK changes with respect to errors in mean and variance estimation for a true Gaussian model. For adequate comparison, the kernel is normalized for each λ such that $k_{\mathrm{mm}}(p, p) = 1$.

3 Related work

Probability product kernel. The probability product kernel (PPK) [3] is a special case of the generative MMK. For probability measures p and p' and $x \in \mathcal{X}$, the probability product kernel is $k_{\mathrm{pp}}(p, p') = \int_{\mathcal{X}} p(x)^\rho p'(x)^\rho dx$. For the case where $\rho = 1$, we have the following result:

Proposition 4. *The probability product kernel with $\rho = 1$ is a special case of the generative mean map kernel with convergence exponential in λ as $\lambda \to \infty$.*

Proof. We use the convergence of the scaled Gaussian kernel $\sqrt{\lambda}k_G(x,x) = \sqrt{\lambda}e^{-\lambda\|x-x'\|^2}$ to the identity kernel $k_\delta(x,x') = \delta(x-x')$ as $\lambda \to \infty$. The PPK $\lim_{\lambda\to\infty} k_{\mathrm{mm}}(p,p')$ expands to $\lim_{\lambda\to\infty} \int_{\mathcal{X}} \int_{\mathcal{X}'} p(x)p'(x')k_G(x,x')dx\,dx' = \lim_{\lambda\to\infty} \int_{\mathcal{X}} \int_{\mathcal{X}'} p(x)p'(x')\sqrt{\lambda}e^{-\lambda\|x-x'\|^2}dx\,dx'$.
Now, using $(x,x') \mapsto (a,b) := (x, x-x')$ and substituting $\sigma = 1/\sqrt{\lambda}$,
$\lim_{\sigma\to 0} \int_{\mathcal{A}} \int_{\mathcal{B}} p(a)p'(a-b)\frac{1}{\sigma}e^{-\|b\|^2/\sigma^2}da\,db = \int_{\mathcal{A}} p(a)p'(a)da = k_{\mathrm{pp}}(p,p')$. □

It further is possible to express a generative MMK analog to the PPK for the case of $\rho \neq 1$; however, the expectation operator is fundamental to the MMK's derivation and this operator requires $\rho = 1$. Provided that the generative MMK can be computed for the distribution of the observed random variables, graphical models for which the PPK is computable are also computable for the generative MMK; this can be seen by observing that the Gaussian kernel couples each pair of observed variables x_i and x_i' into a 2-clique. In the clique graph used by the junction tree algorithm, this 2-clique will appear wherever x_i would have appeared in the clique graph used by the PPK.

Other related kernels. On the surface the generative MMK appears similar to the empirical MMK. Unfortunately, the empirical version relies upon observing the labels in order to include them in a mean map of the full joint distribution over the observed variables and the labels (the latent variables). Without learning a generative model with latent state variables, the empirical mean map kernel is limited to graphical model dependencies between only the observed variables. We expand on this point in the next section by extending the empirical mean map to handle latent variables.

To our knowledge, the Bhattacharyya kernel is the earliest form of a similarity measure between probability distributions [13], but the Fisher kernel [4] is the earliest one used as a machine learning method. The kernel is based on the score $\nabla_\theta \log p(X \mid \hat{\theta})$ for θ the maximum likelihood estimate of a model. The Fisher kernel is sensitive to the parameterization of the statistical family used, whereas the heat kernel [5] is not sensitive to the parameterization. Although it has not yet been shown, the heat kernel may be computable for HMMs with multinomial observation distributions.

The generative MMK is similar to the marginalized kernel [14], although their work focuses on count kernels rather than higher order nonlinear kernels such as the Gaussian. The derivation of the MMK also is different, coming from the direction of Hilbert space embeddings of distributions to arrive at a pd kernel with an implicit injective mapping; in contrast, the marginalized kernel is shown only to be positive semi-definite. The injective property enables the MMK to be used for manifold learning, while the parameter λ affords the generative MMK an additional level of control.

4 Latent mean map kernel

Empirical mean map limitations. The empirical mean map injects empirical probability distributions on sets as points in an RKHS. Originally this method was applied to iid observations, but it has since been extended to non-iid observations as may occur in sequences and time series [7]. The extension involves fixing a depedency model for the observations, such that we consider the empirical distributions induced by the maximal cliques of this model. Assuming the model is correct, the empirical mean map of non-iid data can be decomposed into the sum of the empirical mean maps for each maximal clique's distribution. As in [7], for graphical model G with variable set Z and maximal cliques set $\mathcal{C}$, let v_c be universal kernels on the variable subset of Z induced by clique $c \in \mathcal{C}$. From [15], $v(z,z') = \sum_{c\in\mathcal{C}} v_c(z_c, z_c')$ describes all probability distributions with the specified conditional independence relations using an exponential family model with kernel v.

Markov models straightforwardly capture dependence between observed random variables; however, assuming Markov dependence in the observable space may be overly restrictive. This point is highlighted by the strong performance of latent space models such as HMMs for many prediction problems. The empirical mean map uses only observed data and similarity measures between this data, and so it cannot assume latent variable dependency models. Whereas [7] uses the empirical mean map with latent space models, the latent variables are used for optimization of a kernel-based objective. Hence, the latent variables are used to optimize dependency rather than to model it.

In relaxing the dependency models to include latent variables, we arrive at a somewhat different kernel that admits richer dependency models such as dynamic Bayesian networks and hidden Markov random fields. As with the non-iid mean map, we need only apply the mean map to the distribution

of each maximal clique. The maximal cliques now fall into two sets: fully observable cliques and cliques containing at least one latent random variable. The distributions for the former cliques can be computed empirically similar to [7]; however, applying the empirical mean map to the distributions of the latter cliques is not possible on account of the latent variables.

Latent mean map. The latent mean map augments the empirical mean map by using the posterior distribution of the latent variables, conditional on the observed variables. Let (u, v) be the concatenation of the components of vectors u and v to form a higher dimensional vector. For observed variables X, latent variables Y, and clique-restricted subsets X_c and Y_c , the latent mean map is

$$\mu_c[(X_c, Y_c)] = \mathsf{E}_{(X_c,Y_c)|x_{1:m}}\left[\phi_c((X_c, Y_c))\right] = \frac{1}{m_c}\sum_{i=1}^{m_c} \mathsf{E}_{Y_c^{(i)}|x_{1:m}}\left[\phi_c((x_c^{(i)}, Y_c^{(i)}))\right] \tag{7}$$

for $Y_c^{(i)} \sim P(Y_c^{(i)} \mid x_{1:m})$, the posterior distribution of the random variable $Y_c^{(i)}$ conditioned on *all* observations $x_{1:m}$. This expression captures our best estimate of the clique distribution.

From (7), the latent mean map kernel $\sum_{c\in\mathcal{C}}\langle\mu_c[(X_c, Y_c)], \mu_c[(X'_c, Y'_c)]\rangle$ expands to

$$\sum_{c\in\mathcal{C}} \frac{1}{m_c m'_c}\sum_{i=1}^{m_c}\sum_{j=1}^{m'_c} \mathsf{E}_{\substack{Y_c^{(i)}|x_{1:m} \\ Y_c'^{(j)}|x'_{1:m'}}}\left[v_c((x_c^{(i)}, Y_c^{(i)}), (x_c'^{(j)}, Y_c'^{(j)}))\right].$$

Our end goal is to compute the kernel on many object pairs for SVM classification or kPCA, but even moderate m_c and m'_c render the above expectation intractable. An often exploited trick of kernel methods is the ability to compute inner products in a potentially infinite dimensional space without the need for explicit representations in that space. Here, however, an approximate explicit representation yields computational tractability by allowing us to work with the efficient form in (7). For example, the Gaussian RBF kernel on univariate continuous data admits a truncated Taylor expansion of the exponential [16, Theorem 4.6], empirically yielding low error for low order truncations [17]. For multivariate data, two recent explicit representations approximate the RKHS using random features, with error decreasing exponentially in the number of features chosen [18].

It may be useful to use nonlinear representations even in the space of distributional Hilbert space embeddings. Given a latent mean map kernel matrix K, this can be accomplished by using the alternate kernel matrix $\tilde{K}$ such that $\tilde{K}_{ij} = e^{-\nu(K_{ii}-2K_{ij}+K_{jj})}$ for some parameter ν. In our latent MMK experiments, we push λ toward infinity and consider different values of ν rather than λ.

Latent mean map of HMMs. Learning using the latent MMK requires a dependency model to induce latent variables and a set of conditional distributions sufficient for the model. This model identifies a set of maximal cliques and allows us to compute the posteriors. Suppose we have an HMM θ as described earlier. Assuming stationarity, the model's maximal cliques (x_t, q_t) and (q_t, q_{t+1}) yield T instances of the former and $T-1$ instances of the latter clique:

$$\mu_{xq}[(x_t, q_t)] = \frac{1}{T}\sum_{i=1}^{T} \mathsf{E}_{Q_t|x}\left[\phi_{xq}((x_t, Q_t))\right] = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} \mathsf{P}(Q_t = i \mid x)\phi_{xq}((x_t, Q_t))$$

$$\begin{aligned}\mu_{qq}[(q_t, q_{t+1})] &= \frac{1}{T-1}\sum_{i=1}^{T-1} \mathsf{E}_{Q_t,Q_{t+1}|x,\theta}\left[\phi_{qq}((Q_t, Q_{t+1}))\right] \\ &= \frac{1}{T-1}\sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathsf{P}(Q_t = i, Q_{t+1} = j \mid x, \theta)\phi_{qq}((Q_t, Q_{t+1})).\end{aligned}$$

The forward-backward algorithm can compute the conditional probabilities [19]. We adopt Rabiner's notation [19] for the conditional probabilities so that $\gamma_t(i) = \mathsf{P}(Q_t = i \mid x, \theta)$ and $\xi_t(i, j) = \mathsf{P}(Q_t = i, Q_{t+1} = j \mid x, \theta)$. We use the joint kernel on cliques $v_c((x_c, y_c), (x'_c, y'_c)) = k_c(x_c, x'_c)l_c(y_c, y'_c)$ such that $v_{xq}((x_t, q_t), (x'_t, q'_t)) = k_x(x_t, x'_t)l_q(q_t, q'_t)$ with $v_{qq}((q_t, q_{t+1}), (q'_t, q'_{t+1}))$ defined similarly. As before, we use the 1-of-k encoding to treat the discrete and continuous cases identically. The kernel k will be the Gaussian RBF kernel. For an N-state latent space, K possible symbols, and $\gamma_{(c)}(j) := \sum_{t:x_t=c}^{T}\gamma_t(j)$, the kernel v_{xq} is

$$\frac{1}{TT'}\sum_{\substack{a\in[K]\\ i\in[N]}} \gamma_{(a)}(i)\left(\gamma'_{(a)}(i) + e^{-\lambda}\sum_{j\in[N]\setminus i}\gamma'_{(a)}(j) + e^{-\lambda}\sum_{b\in[K]\setminus a}\left(\gamma'_{(b)}(i) + e^{-\lambda}\sum_{j\in[N]\setminus i}\gamma'_{(b)}(j)\right)\right),$$

which has $O(N^2K^2T)$ complexity. The continuous observation case of v_{xq} is

$$\frac{1}{TT'}\sum_{i=1}^{N}\left(\left\langle\sum_{s=1}^{T}\gamma_s(i)\phi(x_s),\sum_{t=1}^{T'}\gamma'_t(i)\phi(x'_t)\right\rangle+e^{-\lambda}\sum_{j=1}^{N}\left\langle\sum_{s=1}^{T}\gamma_s(i)\phi(x_s),\sum_{t=1}^{T'}\gamma'_t(j)\phi(x'_t)\right\rangle\right) \quad (8)$$

$$=\frac{1}{TT'}\left(\left(\sum_{s,t}k(x_s,x'_t)\sum_{i=1}^{N}\gamma_s(i)\gamma'_t(i)\right)+e^{-\lambda}\left(\sum_{s,t}k(x_s,x'_t)\sum_{i,j}\gamma_s(i)\gamma'_t(j)\right)\right). \quad (9)$$

Whereas (9) is $O(N^2T^2)$, this can be reduced by explicitly representing RKHS elements in (8). Defining $\xi(i,j):=\frac{1}{T-1}\sum_{t\in T}\xi_t(i,j)$, the kernel on the random variable clique (q_t,q_{t+1}) v_{qq} is

$$\sum_{\substack{i\in[N]\\j\in[N]}}\xi(i,j)\left(\xi'(i,j)+e^{-\lambda}\sum_{j'\in[N]\setminus j}\xi'(i,j')+e^{-\lambda}\sum_{i'\in[N]\setminus i}\left(\xi'(i',j)+e^{-\lambda}\sum_{j'\in[N]\setminus j}\xi'(i',j')\right)\right).$$

Interestingly, as λ approaches infinity, the latent MMK on HMMs takes a form similar to the Fisher kernel on HMMs [4] when the Fisher kernel uses the identity matrix instead of the Fisher information matrix. For space we do not reproduce the Fisher kernel formulation here. From our results it will appear that the differences in the computation of two kernels significantly affect their performance.

5 Results

Sequence data. We evaluated the generative and latent MMKs through classification of synthetic data and real DNA sequence data. We explored the performance of SVMs using the generative MMK and the PPK (k_{mm} for $\lambda\to\infty$) on discrete observation HMMs, as well as the latent MMK using a model learned from one class. All HMMs were initialized using uniform distribution initialization for initial state and state transition probabilities and random initialization of emission probabilities, followed by a segmental clustering update via the Viterbi algorithm [19]. We then ran the Baum-Welch algorithm until the first of log-likelihood convergence within 10^{-6} or 1000 iterations.

For the MMKs we used the expectation of the Gaussian kernel, where each symbol is mapped to a vector using the 1-in-k encoding, and we explored logarithmically spaced settings of the parameter λ. For the generative MMK we experimented with linearly spaced settings of the witness length T. For all kernels, the regularization parameter C was tested at logarithmically spaced values. We compare these results with a maximum-likelihood Bayes classifier which maintains an HMM model for each class, the Fisher kernel using a model for one class, and the empirical MMK using Markov dependency models of orders[4] 1, 2, and 3. We report all results for stratified 10-fold cross-validation, which empirically is well behaved [20].

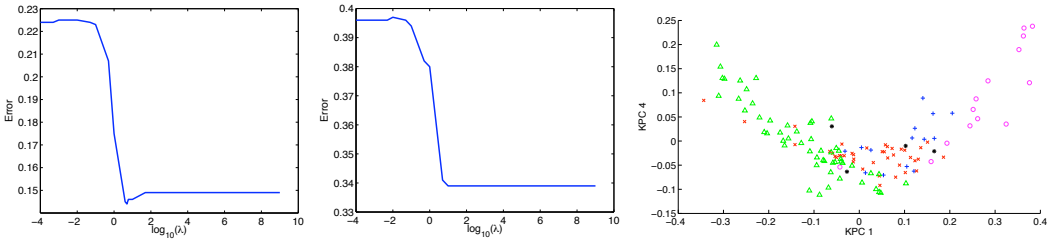
The synthetic dataset consists of 2 classes. For each class, we manually constructed a 3-state 2-symbol HMM to serve as a sequence generator. 500 binary sequences of length 100 were generated for each class. All HMMs learned had 3 states. Table 1 shows the Bayes classifier and the generative MMK perform almost equally, whereas the Fisher kernel and empirical/latent MMKs perform slightly worse. The PPK exhibits much higher loss than the other methods, possibly owing to its higher sensitivity to errors in model estimation.

We also ran two-class sequence classification experiments on a random subset of 500 exons and 500 introns from the HS^{3}D dataset [21]. The sequence lengths vary from order of 10^2 to 10^4 symbols in length, making for a challenging classification problem. For the generative MMK and PPK, we adopted a heuristic formula [3] to choose the number of states n in the HMM learned for a particular sequence: $n=\lfloor\frac{1}{2}\sqrt{k^2+4(T\gamma+k+1)}-\frac{1}{2}k\rfloor+1$, for k symbols and a constant γ set to 0.1. For the Bayes classifier, we used a 4-state model for the exons and an 8-state model for the introns because this configuration produced the lowest loss. For the latent MMK and the Fisher kernel, we used a 4-state model trained on the exons. For the empirical MMK, we report results for a Markov dependency model of order 3, which produced the best results among orders $\{1,2,3,4\}$.

[4] An order-k Markov model induces maximal cliques of size $k+1$, with the kernel described in [7]. An empirical MMK with this model is effectively a string kernel whose feature space representation consists of counts of each string of length $k+1$.

	GMMK	PPK ($\rho = 1$)	PPK ($\rho = 0.5$)	EMMK	LMMK	Fisher	Bayes
Synthetic	0.063	0.091	0.247	0.067[5]	0.068	0.066	**0.062**
HS^{3}D	**0.144**	0.149	0.165	0.215	0.279	0.170	0.192

Table 1: Optimal loss for the methods tested on synthetic and HS^{3}D datasets.



(a) Generative MMK - HS^{3}D (b) Latent MMK - HS^{3}D (c) Species Manifold from kPC's 1 and 4

Figure 2: Results on HS^{3}D data. (a) Error for generative MMK for varying λ and PPK (λ = 1e9). (b) Error for latent MMK for varying λ. (c) First versus fourth kernel principal components of the generative mean map kernel matrix for $\lambda = 1$. Plants are triangles, molluscs are circles, arachnids are asterisks, insects are x's, and fungi are crosses.

We restrict our results for the generative MMK to the setting of $T = 30$ which had the lowest mean loss over all values of λ. Our results in Table 1 show that the generative MMK performs slightly better than the PPK, both of which outperform the Fisher kernel. The generative MMK significantly outperforms the Bayes classifier, but the latent and empirical MMKs do relatively poorly.

Biodiversity data. We also applied the generative MMK to visualize ecological relationships between species sampled in Costa Rica by INBio[6]. For each species, observations consist of counts on a grid, where each grid location has the accompanying abiotic features altitude, median annual temperature, annual precipitation, isothermality, and temperature seasonality. Each species has a true density in this 5-dimensional space, which we estimate using Gaussian RBF kernel density estimators. For the sampling associated with each species, we optimized the bandwidth of a kernel density estimator by minimizing the mean integrated square error [22]. After obtaining the optimal bandwidth for each species sampling, we formed the kernel matrix between all pairs of the species kernel density estimators using the generative MMK with $\lambda = 1$. We then applied kPCA to identify whether particular components well-separate certain groups of species. The visualization of the first and fourth kernel principal components in 2(c) shows some separation between the different species groups. In particular, the insects and arachnids are clustered near the center, with plants towards the left and molluscs towards the right.

6 Conclusion

Our results show that the generative MMK successfully utilizes generative models learned on sequence data for achieving strong performance on two discriminative tasks, outperforming the empirical mean map and its latent model extension. Using sampling techniques [3], it is straightforward to extend the generative MMK to more complicated time series models such as switching linear dynamic systems and factorial HMMs. The experiments presented here all concern discrete sequences. For models with continuous random variables, such as HMMs with mixture of Gaussians observation distributions, estimation can be challenging; we anticipate that the generative and latent MMKs may have a larger impact for these models, because λ can serve to reduce the sensitivity of the kernel to small deviations between learned observation distributions; however, further testing in the continuous domain demands that we first overcome computational complexity challenges by using approximations of explicit representations of RKHS elements.

[5]EMMK results on synthetic data were fairly unstable with respect to λ.

[6]The National Biodiversity Institute of Costa Rica, `http://www.inbio.ac.cr/en`

References

[1] B. Schölkopf, A.J. Smola, and K.R. Muller. Kernel principal component analysis. *Lecture Notes in Computer Science*, 1327:583–588, 1997.

[2] B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT press Cambridge, Mass, 2002.

[3] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[4] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Neural Information Processing Systems*, pages 487–493, 1999.

[5] J. Lafferty and G. Lebanon. Diffusion Kernels on Statistical Manifolds. *Journal of Machine Learning Research*, 6:129–163, 2005.

[6] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. *Lecture Notes in Computer Science*, 3734:63, 2005.

[7] X. Zhang, L. Song, A. Gretton, and A. Smola. Kernel measures of independence for non-iid data. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *NIPS 22*. MIT Press, Cambridge, MA, 2009 (in press).

[8] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. *Journal of Machine Learning Research*, 1:1–10, 2008.

[9] A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. *Lecture Notes in Computer Science*, 4754:13, 2007.

[10] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2, 2002.

[11] R. Kondor and T. Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning*, 2003.

[12] A.B. Tsybakov. *Introduction to nonparametric estimation*. Springer, 2008.

[13] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35(99-109):4, 1943.

[14] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl 1):S268–S275, 2002.

[15] Y. Altun, A.J. Smola, and T. Hofmann. Exponential families for conditional random fields. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pages 2–9. AUAI Press Arlington, Virginia, United States, 2004.

[16] I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. *IEEE Transactions on Information Theory*, 52(10):4635, 2006.

[17] J. W. Xu, P. P. Pokharel, K. H. Jeong, and J. C. Principe. An explicit construction of a reproducing Gaussian kernel Hilbert space. In *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, volume 5, 2006.

[18] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 2007.

[19] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[20] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on AI*, volume 14, pages 1137–1145, 1995.

[21] P. Pollastro and S. Rampone. HS3D: Homo Sapiens Splice Site Dataset. *Nucleic Acids Research, Annual Database Issue*, 2002.

[22] L. Wasserman. All of statistics. *Statistics*, 2004.