Bhattacharyya and Expected Likelihood Kernels

Tony Jebara and Risi Kondor

Columbia University, New York, NY 10027, USA *

Abstract. We introduce a new class of kernels between distributions. These induce a kernel on the input space between data points by associating to each datum a generative model fit to the data point individually. The kernel is then computed by integrating the product of the two generative models corresponding to two data points. This kernel permits discriminative estimation via, for instance, support vector machines, while exploiting the properties, assumptions, and invariances inherent in the choice of generative model. It satisfies Mercer's condition and can be computed in closed form for a large class of models, including exponential family models, mixtures, hidden Markov models and Bayesian networks. For other models the kernel can be approximated by sampling methods. Experiments are shown for multinomial models in text classification and for hidden Markov models for protein sequence classification.

1 Introduction

A variety of efforts in machine learning have explored the fusion of discriminative and generative estimation to exploit their complementary advantages. Some approaches use discriminative learning algorithms and paradigms for generative models. For instance, a generative model may be estimated conditionally [5, 3, 21] or discriminatively [11, 23] to improve its performance for classification. Other approaches explore the use of generative models within standard discriminative classifiers such as support vector machines (SVMs). These generative models help induce appropriate feature space mappings or kernels. For example, the Fisher kernel method forms a generative model of the aggregated data set to compute a kernel on the resulting statistical manifold [10]. Alternatively, information diffusion gives kernels by solving heat equations on a statistical manifold over a given generative model's parameter space [16]. Nevertheless, kernels are frequently engineered independently of generative modeling to obtain desired properties [20, 9]. For instance, string kernels and sequential data kernels [18, 17, 6, 7, 26, 25] do not specifically address the generative hidden Markov model (HMM) literature. However, there may potentially be much to gain by building upon generative modeling, HMM-variants and statistical tools to facilitate the kernel design process.

In this paper, we propose another point of contact between generative models and kernels. We describe a general class of kernels that are computed by estimating a generative probability model for each given datum (or multiple data

* For email contact: jebara@cs.columbia.edu or risi@cs.columbia.edu.

points) in the input space via maximum likelihood or another criterion. The kernel's output value for a pair of data points is then obtained by integrating the product of their corresponding probability models taken to a power. This measure of affinity is a generalized form of the Bhattacharyya similarity measure. The kernel readily accommodates many popular distributions allowing us to consider a variety of input spaces (sequences, discrete structures, etc.) while inheriting properties and invariances of the probabilistic modeling. For instance, the kernel applies to exponential family distributions, mixtures and HMMs.

Previous efforts involved generative modeling with statistical manifolds using the Kullback-Leibler (KL) divergence to set up affinity measures between probabilistic models [10, 16]. The KL-divergence is asymmetric and typically is approximated by a local metric (i.e. in the neighborhood of a single maximum likelihood estimate for the whole data set) to generate, for instance, the Fisher kernel [10]. One disadvantage of such a local approximation is that exponential family distributions only generate linear Fisher kernels (see Section 3). Recent work in information diffusion kernels [16] proposes an alternative way of dealing with the statistical manifold by partial differential methods and heat equations as opposed to a local maximum likelihood estimate for the whole dataset. The authors explicate the cases of the multinomial on the sphere and the Gaussian variance on a hyperbolic space which are both solvable and yield interesting nonlinear kernels. However, the latter work has yet to be extended to the wide class of exponential family or mixture model distributions due to the difficulty in finding closed form solutions to the heat equation for arbitrary geometries. In contrast, the measure we choose gives a symmetric kernel from the outset which handles a wide variety of generative models in closed form and can even be computed via sampling methods for arbitrary distributions.

This paper is organized as follows. We first present the general form of our kernel as a product of two distributions each induced from data and note certain properties. We then show how the kernel can be computed in closed form for any distribution in the exponential family, thereby covering a wide range of classical generative models. We derive the particular formulas for the Gaussian, the Bernoulli and the multinomial distribution. We then discuss how to extend the kernel to any mixture model as well as structured mixture models such as HMMs. For generative models that are not straightforward, we show how we can readily use sampling methods to compute the kernel. We then present other implications of the kernel in terms of the regularization and the reproducing kernel Hilbert space. Preliminary experiments are shown for the SCOP protein sequence dataset and the WebKB text dataset. We conclude with discussions.

2 A Kernel on Distributions

Given a positive (semi-) definite kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ on the input space $\mathcal{X}$, and examples $\chi_1, \chi_2, \ldots, \chi_m \in \mathcal{X}$ with labels $y_1, y_2, \ldots y_m \in \mathcal{Y}$, kernel based learning algorithms return hypotheses of the form $h(\chi) = \sum_i \alpha_i K(\chi_i, \chi) + b$. Instead of defining a kernel directly between examples $\chi, \chi' \in \mathcal{X}$, in this paper we define a class of kernels $K_\rho : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$ on the space of normalized probability distributions over some probability space Ω. Specifically, we define the general **Probability Product Kernel** between distributions p and p' as

$$K_\rho(p, p') = \int_\Omega p(x)^\rho \, p'(x)^\rho \, dx. \tag{1}$$

Examples can be of the form of a single data point $\chi = \{x \in \Omega\}$ or a set of data points $\chi = \{x_1, x_2, \ldots, x_n \;:\; x_i \in \Omega\}$. We assume that for each χ there is an underlying distribution generating data points, and that χ is a set of independent, identically distributed set of samples from that distribution. We then induce a kernel between χ and χ' by forming estimates p and p' of their underlying distributions and computing the probability product kernel between these estimates:

$$\overline{K}_\rho(\chi, \chi') = K_\rho(p, p') = \int_\Omega p(x)^\rho \, p'(x)^\rho \, dx.$$

For any $p_1, p_2, \ldots, p_n \in \mathcal{P}$ and $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}$,

$$\sum_i \sum_j \alpha_i \alpha_j K_\rho(p_i, p_i) = \int_\Omega \left(\textstyle\sum_i \alpha_i p_i(x)^\rho \right)^2 dx \geq 0 \,, \tag{2}$$

hence K is trivially positive definite on $\mathcal{P}$. This implies that for any deterministic estimation procedure $\Phi : \chi \mapsto p$, $\overline{K}$ is positive definite on $\mathcal{X}$ and hence a suitable kernel for use in learning algorithms in its own right. Additionally, $\overline{K}$ is invariant (symmetric) with respect to permutations of the individual data points comprising χ and χ'. In the following, we shall omit the bar sign over the induced kernel and may omit the subscript ρ when that does not risk causing confusion.

The space of distributions $\mathcal{P}$ can trivially be embedded in the Hilbert space of functions $L_1(\Omega)$, and the estimation mapping $\Phi : \mathcal{X} \mapsto \mathcal{P}$ can be regarded as the feature map. By appropriate choice of Φ and ρ, a powerful family of kernels can be constructed, combining the advantages of parametric and non-parametric statistical methods. Essentially, the Probability Product Kernel acts as a measure of the degree of similarity or affinity [1] between the two distributions.

For $\rho = 1/2$,

$$K(\chi, \chi') = \int \sqrt{p(x)} \sqrt{p'(x)} \, dx \tag{3}$$

which we shall call the **Bhattacharyya Kernel**, because in the statistics literature it is known as Bhattacharyya's measure of affinity between distributions [4, 1], related to the better known Hellinger's distance

$$H(p, p') = \frac{1}{2} \int \left(\sqrt{p(x)} - \sqrt{p'(x)} \right)^2 dx$$

[1] The proposed kernels (probability product, Bhattacharyya and expected likelihood) are not the only possible measures of similarity between distributions. A more customary measure is the Kullback-Leibler divergence $D(p_1 \| p_2) = \int_\Omega p_1(x) \log p_1(x) \, dx - \int_\Omega p_1(x) \log p_2(x) \, dx$ yet it not positive definite, not symmetric, and often not as straightforward to use as a kernel.

by $H = \sqrt{2-2K}$. Note that the Hellinger distance can be seen as a principled symmetric approximation of the Kullback Leibler (KL) divergence and in fact is a bound on KL as shown in [24] where relationships between many information theoretic divergences are characterized. Unlike some divergences, Hellinger naturally implies a symmetric (Bhattacharyya) affinity. Kernels of this form were introduced in [15] and have the special property $K(\chi, \chi') = 1$. For $\rho = 1$, we note another interesting configuration where the kernel behaves as the expectation of one distribution under the other:

$$K(\chi, \chi') = \int p(x)\, p'(x)\, dx = \mathrm{E}_p[p'(x)] = \mathrm{E}_{p'}[p(x)], \tag{4}$$

which we shall refer to as the **Expected Likelihood Kernel**. This kernel is particularly easy to evaluate by sampling methods, as we discuss in Section 6.

2.1 Frequentist and Bayesian methods of estimation

Various strategies may be used to estimate $p(x)$ from the sample χ. Given a parametric family $\{p(x|\theta)\}_\theta$, the simplest approach is to choose $p(x) = p(x|\hat{\theta})$ corresponding to the maximum likelihood estimator $\hat{\theta} = \arg\max \log p(\chi|\theta)$, but other point estimators can plugged into $\hat{\theta}$. The Bayesian approach postulates a prior $p(\theta)$ on the parameters and invokes Bayes' rule

$$p(\theta|\chi) = \frac{p(\chi|\theta)\, p(\theta)}{\int p(\chi|\theta)\, p(\theta)\, d\theta}\,.$$

One could use the Maximum a Posteriori estimate $p(x|\hat{\theta}_{\mathrm{MAP}})$, where $\hat{\theta}_{\mathrm{MAP}} = \arg\max p(\theta|\chi)$, or the true posterior

$$p(x|\chi) = \int p(x|\theta)\, p(\theta|\chi)\, d\theta. \tag{5}$$

In practice, the samples χ_i are often very small, or consist of just a single datum, and in this case the Bayesian approach may provide regularization to avoid over-fitting. Both MAP and maximum likelihood estimators can be seen as approximations to the full posterior. Another type of regularization to consider is a form of shrinkage which draws estimates from all training points closer together:

$$\theta = \arg\max \left[\log p(\chi|\theta) + \lambda \sum_i \log(\chi_i|\theta) \right].$$

In the following we shall investigate particular estimation methods for which the kernel can be computed in closed form.

Family	$\mathcal{A}(X)$	$\mathcal{K}(\theta)$	Parameter
Gaussian (mean)	$-\frac{1}{2}X^T X - \frac{D}{2}\log(2\pi)$	$\frac{1}{2}\theta^T\theta$	$\theta \in \mathbb{R}^D$
Gaussian (variance)	$-\frac{1}{2}\log(2\pi)$	$-\frac{1}{2}\log(\theta)$	$\theta \in \mathbb{R}_+$
Multinomial	$\log(\Gamma(\eta+1)) - \log(\nu)$	$\eta \log(1+\sum_{d=1}^{D}\exp(\theta_d))$	$\theta \in \mathbb{R}^D$
Exponential	0	$-\log(-\theta)$	$\theta \in \mathbb{R}_-$
Gamma	$-\exp(X) - X$	$\log \Gamma(\theta)$	$\theta \in \mathbb{R}_+$
Poisson	$\log(X!)$	$\exp(\theta)$	$\theta \in \mathbb{R}$

Table 1. Definition of $\mathcal{A}$ and $\mathcal{K}$ in natural form for some exponential families.

3 Exponential families

A family of distributions is said to form an exponential family [2] if it can be written in the form

$$p(x|\theta) = \exp(\mathcal{A}(x) + \theta^T \mathcal{T}(x) - \mathcal{K}(\theta))$$

where the measure is denoted $\mathcal{A}$, the cumulant generating function is denoted $\mathcal{K}$, the so-called sufficient statistics are computed via $\mathcal{T}$ and the θ is the natural parameter of the distribution. Often, $\mathcal{T}(x)$ is just x.

Many familiar distributions, such as the Normal, Bernoulli, Multinomial, Poisson and Gamma distributions can be written in this form (Table 1). Note that $\mathcal{A}$ and $\mathcal{K}$ are related through the Laplace transform

$$\mathcal{K}(\theta) = \log \int \exp(\mathcal{A}(x) + \theta^T \mathcal{T}(x))\, dx$$

since $p(x|\theta)$ is normalized. Furthermore, it is straightforward to show that $\mathcal{K}$ is convex. The maximum likelihood estimate for θ under this distribution is given by equating the gradient (which we will denote as $\mathcal{G}(\theta)$) of the cumulant generating function to the (empirical) expected value of the sufficient statistic:

$$\mathcal{G}(\theta) \;=\; \frac{\partial \mathcal{K}(\theta)}{\partial \theta} \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathcal{T}(x_i)$$

For exponential families, the Bhattacharyya kernel ($\rho = 1/2$) is:

$$\begin{aligned} K(\chi, \chi') = K(p, p') &= \int p(x|\theta)^{1/2} p(x|\theta')^{1/2} dx \\ &= \exp\left(\mathcal{K}\left(\tfrac{1}{2}\theta + \tfrac{1}{2}\theta'\right) - \tfrac{1}{2}\mathcal{K}(\theta) - \tfrac{1}{2}\mathcal{K}(\theta')\right) . \end{aligned}$$

We can expand the above in terms of the actual data χ and χ' by using (for instance) their corresponding maximum likelihood settings for θ and θ'.

It is interesting to note that the above kernel is in general nonlinear for e-family models (and possibly infinite dimensional in feature space) and we expect the choice of generative distribution to greatly influence the resulting kernel formula we obtain (the Fisher kernel for e-family models which is typically linear[2] in $\mathcal{T}(x)$ unlike our kernel). For particular families, more explicit formulae also exist for general ρ. In the following, we examine some of these cases.

3.1 Gaussian models

The D dimensional Gaussian distribution $p(x) \sim \mathcal{N}(\mu, \Sigma)$ is of the form

$$p(x) = (2\pi)^{-D/2} \,|\, \Sigma \,|^{-1/2} \, \exp\left(-\tfrac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

where Σ is a positive definite matrix and $|\,\Sigma\,|$ denotes its determinant. For a pair of Gaussians $p \sim \mathcal{N}(\mu, \Sigma)$ and $p' \sim \mathcal{N}(\mu', \Sigma')$, completing the square in the exponent gives the general probability product kernel:

$$\begin{aligned} K_\rho(\chi, \chi') = K_\rho(p, p') &= \int_{\mathbb{R}^D} p(x)^\rho p'(x)^\rho \, dx \\ &= (2\pi)^{(1-2\rho)D/2} \, \left|\, \Sigma^\dagger \,\right|^{1/2} |\, \Sigma \,|^{-\rho/2} \, |\, \Sigma' \,|^{-\rho/2} \\ &\quad \exp\left(-\frac{\rho}{2} \mu^\top \Sigma^{-1} \mu - \frac{\rho}{2} {\mu'}^\top {\Sigma'}^{-1} \mu' + \frac{1}{2} {\mu^\dagger}^\top \Sigma^\dagger \mu^\dagger \right) \end{aligned}$$

where $\Sigma^\dagger = \left(\rho \Sigma^{-1} + \rho {\Sigma'}^{-1}\right)^{-1}$ and $\boldsymbol{\mu}^\dagger = \rho \Sigma^{-1} \boldsymbol{\mu} + \rho {\Sigma'}^{-1} \boldsymbol{\mu}'$. If the covariance is isotropic and fixed: $\Sigma = \sigma^2 I$, this simplifies to:

$$K_\rho(p, p') = 2^{D/2} \, (2\pi\sigma^2)^{(1-2\rho)D/2} \, \exp\left((\rho-1) \frac{\mu^T \mu + {\mu'}^T \mu'}{2\sigma^2} - \frac{(\mu'-\mu)^T (\mu'-\mu)}{4\sigma^2}\right),$$

which, for $\rho = 1$ (the expected likelihood kernel) simply gives the following Gaussian (whose variance is effectively double the original $\Sigma = \sigma^2 I$):

$$K(p, p') = \tfrac{1}{(4\pi\sigma^2)^{D/2}} \, e^{-\|\,\mu' - \mu\,\|^2/(4\sigma^2)}$$

Writing the above explicitly in terms of the maximum likelihood setting $\mu = \overline{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\mu' = \overline{x}' = \frac{1}{n'} \sum_{i=1}^{n'} x'_i$, yields the traditional radial basis function (RBF) kernel:

$$K(\chi, \chi') = \tfrac{1}{(4\pi\sigma^2)^{D/2}} \, \exp\left(-\|\overline{x} - \overline{x}'\|^2/(4\sigma^2)\right)$$

Similarly for $\rho = 1/2$ (the Bhattacharyya kernel), we also obtain the RBF kernel.

[2] Recall the Fisher kernel computed at the dataset's maximum likelihood estimate θ^* has the following form: $K(\chi, \chi') = U_\chi I_{\theta^*}^{-1} U_{\chi'}$ where $U_\chi = \nabla_\theta \log P(\chi|\theta)|_{\theta^*}$ is the general formula. For the exponential family, this reduces to $U_\chi = \mathcal{T}(\chi) - \mathcal{G}(\theta^*)$ which is linear in $\mathcal{T}(\chi)$.

3.2 Bernoulli and Naive Bayes Models

The Bernoulli distribution $p(x) = \gamma^x(1-\gamma)^{1-x}$ with parameter $\gamma \in [0,1]$, and its D dimensional variant, sometimes referred to as Naive Bayes,

$$p(x) = \prod_{d=1}^{D} \gamma_d^{x_d}(1-\gamma_d)^{1-x_d}$$

with $\gamma \in [0,1]^D$, are used to model binary $x \in \{0,1\}$ or multidimensional binary $x \in \{0,1\}^D$ observations, respectively. The Bhattacharyya kernel between a pair of such distributions

$$K_\rho(\chi, \chi') = K_\rho(p, p') = \sum_{x \in \{0,1\}^D} \prod_{d=1}^{D} (\gamma_d \gamma_d')^{\rho x_d} ((1-\gamma_d)(1-\gamma_d'))^{\rho(1-x_d)}$$

factorizes trivially (for any setting of ρ) as:

$$K_\rho(p, p') = \prod_{d=1}^{D} \left[(\gamma_d \gamma_d')^\rho + (1-\gamma_d)^\rho (1-\gamma_d')^\rho \right].$$

3.3 Multinomial Models

For discrete count data, when $x = (x_1, x_2, \ldots, x_D)$ is a vector of non-negative integer counts summing to X, we can use the multinomial model

$$p(x) = \frac{X!}{x_1!\, x_2! \ldots x_D!}\, \alpha_1^{x_1} \alpha_2^{x_2} \ldots \alpha_D^{x_D}$$

with parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_D)$ subject to $\sum_{d=1}^{D} \alpha_d = 1$. The maximum likelihood estimate given observations $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ is

$$\hat{\alpha}_d = \frac{\sum_{i=1}^{n} x_d^{(i)}}{\sum_{i=1}^{n} \sum_{d=1}^{D} x_d^{(i)}}.$$

For the case $\rho = 1/2$, fixing X, the Bhattacharyya kernel $K(\chi, \chi') = K(p, p')$ for counts can be computed explicitly using the multinomial theorem, giving:

$$K(p, p') = \sum_{\substack{x=(x_1, x_2, \ldots, x_D) \\ \sum_i x_i = X}} \frac{X!}{x_1!\, x_2! \ldots x_D!} \prod_{d=1}^{D} (\alpha_d \alpha_d')^{x_d/2} = \left[\sum_{d=1}^{D} (\alpha_d \alpha_d')^{1/2} \right]^X \tag{6}$$

which is equivalent to the homogeneous polynomial kernel of order $X/2$ between $(\alpha_1, \alpha_2, \ldots, \alpha_D)$ and $(\alpha_1', \alpha_2', \ldots, \alpha_D')$. If we do not wish to hold X constant, we may sum over all its possible values

$$K(p, p') = \sum_{X=0}^{\infty} \left[\sum_{d=1}^{D} (\alpha_d \alpha_d')^{1/2} \right]^X = \left(1 - \sum_{d=1}^{D} (\alpha_d \alpha_d')^{1/2} \right)^{-1}$$

or weight each power differently (i.e. a power series expansion). For the general case of $\rho \neq 1/2$, a general formula for discrete multinomial events is available (if $\rho \neq 1/2$ there is no general closed form formula for counts except for $X=1$ where, as opposed to counts, we really have single mutually exclusive events). This discrete events scenario if arguably more relevant and yields the form:

$$K(p,p') = \sum_{d=1}^{D} (\alpha_d \alpha'_d)^\rho.$$

4 Mixture Models

For extensions to a mixture model setting, it is clear that the Bhattacharyya kernel with $\rho = 1/2$ becomes less attractive than the expected likelihood kernel since the square root of a mixture probability is unwieldly[3]. However, with $\rho = 1$, we can easily evaluate any mixture model via the subkernel evaluations over the cross-product of all the hidden states as follows. Consider the case of mixture models $p = \sum_m p(m)p(x|m)$ and $p' = \sum_n p'(n)p'(x|n)$ (with slight abuse of notation). Here, the first mixture is over M configurations while the second is over N configurations. The expected likelihood kernel trivially reduces to a sum of $M \times N$ elementary expected likelihood subkernels $K_{i,j}(\chi, \chi')$ for each setting of the hidden variables:

$$K(\chi,\chi') = \sum_m \sum_n p(m)p'(n) \int p(x|m)p'(x|n)\, dx = \sum_{m,n} p(m)p'(n) K_{m,n}(\chi,\chi').$$

A generalization of the above is possible for $\rho = 2, 3, \ldots$ provided that the higher order kernel $K(p_1, p_2, \ldots, p_{2\rho}) = \int p_1(x)p_2(x)\ldots p_{2\rho}(x)\, dx$ is easy to compute. The above mixture models can be readily applied to our previous solutions for the Gaussian, the multinomial (if we have a single event, i.e. $X=1$ as opposed to counts) and the Bernoulli since these were computed explicitly for $\rho=1$. No such solution is readily available for $\rho = 1/2$, and general mixture models of other exponential family forms need to be derived specifically for $\rho = 1$. One heuristic is to simply impute the $\rho = 1/2$ value for the subkernel exponential family evaluations while maintaining $\rho = 1$ for handling the mixture model.

5 Hidden Markov Models and Bayesian Networks

Perhaps more interestingly, the above mixture modeling and latent variable framework extends naturally to HMMs and general latent Bayesian networks without considering the brute force cross product of their hidden variables. This is done by taking advantage of conditional independencies in the graphical models. We thus consider new forms of sequence-based or network-based kernels.

[3] Approximations may be possible for the setting $\rho = 1/2$ via Jensen's inequality.

Recall, for instance, the general form of an HMM for a sequence of observations $X = (x_1, \ldots, x_T)$ (as discrete or continuous vectors):

$$p(X) = \sum_{S=s_1,\ldots,s_T} p(s_1)\, p(x_1|s_1) \prod_{t=2}^{T} p(s_t|s_{t-1})\, p(x_t|s_t)\ .$$

The expected likelihood kernel is merely the co-emission probability of two different HMMs [19, 14]. We compute a kernel between two sequences χ and χ' by fitting an HMM to each and summing (or integrating) the product over all possible input sequences X. Given an HMM $p(X|\theta)$ with discrete states s_t of cardinality M and an HMM $p(X|\theta')$ with discrete states u_t of cardinality N, a brute force evaluation would explore $M^T \times N^T$ configurations of their joint hidden variables and compute a subkernel for each. This is because both HMMs need to be marginalized over their hidden variables $S = (s_1, \ldots, s_T)$ and $U = (u_1, \ldots, u_T)$. However, due to the Markov structure, we need not consider all possible configurations of each HMM as shown below [4]:

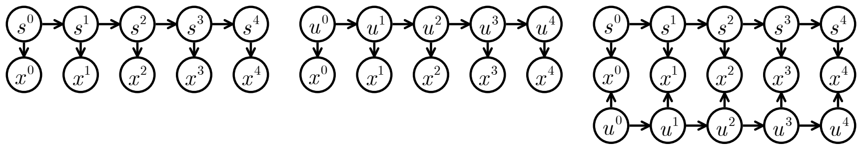
$$\begin{aligned}
K(\chi, \chi') &= \sum_{X=(x_1,\ldots,x_T)} p(X)\, p'(X) \\
&= \sum_S \sum_U \prod_{t=1}^{T} p(s_t|s_{t-1}) p'(u_t|u_{t-1}) \sum_{x_t} p(x_t|s_t) p'(x_t|u_t) \\
&= \sum_S \sum_U \prod_{t=1}^{T} p(s_t|s_{t-1}) p'(u_t|u_{t-1}) \psi(s_t, u_t)\ .
\end{aligned}$$

The above indicates only subkernels $K_{s_t,u_t} = \sum_{x_t} p(x_t|s_t)p'(x_t|u_t)$ need to be computed for each of the T x_t variables independently under each setting of their parent variables s_t and u_t. Thus, we evaluate $T \times M \times N$ subkernels. These effectively form positive clique functions $\psi(s_t, u_t) = K_{s_t,u_t}$ over the common parents of each x_t variable in the network. It is then straightforward to sum over hidden states of the resulting graphical model via a junction tree algorithm (see Figure 1). The two graphs are coupled via common children in X and cliques over their joint parents emerge as we propagate messages [13].

The above efficient approach extends to general Bayesian networks. These are directed acyclic graphs whose probability distribution factorizes as $P(X) = \prod_i P(x_i|\pi_i)$ where π_i is the set of random variables that are parents of the variable x_i in the graph. Some of the variables may be latent while others are in the input or sample space. Ultimately, we only need to compute subkernels over the configurations of the common parents for each subvariable of X in our network. These form positive clique functions that couple the common parents:

$$\psi(\pi_i, \pi_i') = K_{\pi_i,\pi_i'} = \int p(x_i|\pi_i) p'(x_i|\pi_i') dx.$$

[4] For brevity in the product over t we assume $p(s_1|s_0) = p(s_1)$ and $p(u_1|u_0) = p(u_1)$.



(a) HMM for $p(X)$. (b) HMM for $p'(X)$. (c) Resulting graph.

Fig. 1. The resulting graphical model from two hidden Markov models as the kernel couples common parents for each node creating undirected edges between them.

The two Bayesian networks need not be the same as long as, when marginalized over their hidden variables, they are distributions over the sample space X. Computations grow tractably with the enlarged clique sizes of the joint parents. Furthermore, if the original networks do not have loops, the resulting fused network from the expected likelihood kernel will not give rise to loops itself.

6 Sampling Approximation

To accommodate the complete class of generative models (i.e. beyond mixture models and other latent models) when we can no longer find closed form formulas for the probability product kernel for any setting of ρ, the expected likelihood kernel can be approximated by sampling methods. This hinges on our ability to generate samples and evaluate their likelihood with a given generative model yet these operations are often assumed to be readily available.

For $\rho = 1$ (the expected likelihood kernel) the approximation (c.f. eq. 4)

$$K(\chi, \chi') = K(p, p') \approx \frac{\beta}{N} \sum_{\substack{x_i \sim p(x) \\ i=1,\ldots,N}} p'(x_i) \;+\; \frac{(1-\beta)}{N'} \sum_{\substack{x'_i \sim p'(x) \\ i=1,\ldots,N'}} p(x'_i)$$

(where $x_1, x_2, \ldots, x_N$ and $x'_1, x'_2, \ldots, x'_{N'}$ are iid samples from p and p' respectively and $\beta \in [0,1]$ is a parameter of our choosing) is guaranteed to converge to the true value of the kernel by the law of large numbers.

Often unusual distributions occur in the context of generative models. Hence, at least for the expected likelihood kernel, when the analytic approach to calculating $K(\chi, \chi')$ fails, we will often find that we can easily and efficiently generate samples and compute the kernel using this approximation. In the case of infinite samples, the above is an exact evaluation of the kernel. However, in practice we can use a finite number of samples yet still consistently obtain a rapidly converging, reliable numerical estimate for the kernel. Furthermore, in the cases where sampling from the distribution is difficult, we may use importance sampling and related methods to compute the kernel.

7 Reproducing Kernel Hilbert Spaces

The mapping $\Phi : \mathcal{X} \mapsto \mathcal{P}$ described in Section 2 is not the only Hilbert space representation of K satisfying $K(\chi, \chi') = \langle \Phi(\chi), \Phi(\chi') \rangle$. The so-called Reproducing Kernel Hilbert Space (RKHS) representation associates with each χ the function $\Phi_{\mathrm{RKHS}}(\chi) = f_\chi = K(\chi, \cdot)$. Defining the inner product as $\langle \Phi_{\mathrm{RKHS}}(\chi), \Phi_{\mathrm{RKHS}}(\chi') \rangle = K(\chi, \chi')$ lends the resulting Hilbert space $\mathcal{H}$ the special property that for any $f \in \mathcal{H}$, $\langle f, f_\chi \rangle = f(\chi)$, in particular, $\langle f_\chi, f_{\chi'} \rangle = f_\chi(\chi') = f_{\chi'}(\chi) = K(\chi, \chi')$.

Note that by construction of the kernel, $\mathcal{H}$ only contains functions symmetric in $\{x_1, x_2, \ldots, x_n\}$, i.e. invariant under permutations of the components of χ. The above inner product can be related to the standard product between functions by $\langle f, f' \rangle = \int_{\mathcal{X}} (Pf)(\chi)\, (Pf')(\chi)\, d\chi$ for some regularization operator $P : \mathcal{H} \mapsto \mathcal{H}$ [8].

Kernel based learning algorithms generally return hypotheses of the form $\hbar(\chi) = \langle h, \Phi(\chi) \rangle + b = h(\chi) + b$ where $h \in \mathcal{H}$ and $b \in \mathbb{R}$ together minimize the regularized risk $R_{\mathrm{reg}}(h, b) = \frac{1}{m} \sum_i^m L(y_i, \hbar(\chi_i)) + \frac{1}{2} \langle h, h \rangle$, where $(\chi_i)_{i=1}^m$ is the training data, $(y_i)_{i=1}^m$ are the training labels and L is a loss function. Hence, understanding P is the key to understanding the way our kernel implements capacity control, i.e. avoids over-fitting.

For our kernel defined by way of a kernel between distributions, if $\mathcal{P}$ is parameterized by $\theta \in \Theta$, we can introduce an analogous RKHS construction with respect to Θ by setting $f_\theta(\theta') = K(p_\theta, p_{\theta'})$ and $\langle f_\theta, f_{\theta'} \rangle = K(p_\theta, p_{\theta'})$ leading to $\langle f, f_\theta \rangle = f(\theta)$. A family of distributions indexed by $\theta \in \mathbb{R}^d$ is called a location family if $p_\theta(x) = p'_\theta(x - \theta' + \theta)$. An example of a location family is the family of unit variance Normal distributions on $\mathbb{R}^D$. When our parametric model for computing Bhattacharyya or expected likelihood kernels is chosen from a location family, the kernel will be translation invariant in the sense that

$$K(p_\theta, p_{\theta'}) = \int p_\theta(x)\, p_{\theta'}(x)\, dx = k(\theta' - \theta),$$

where, for simplicity, we have set $\rho = 1$, although the generalization to other values is obvious. We then have

$$k(\theta' - \theta) = \int p_0(x)\, p_0(x - \theta' + \theta)\, dx$$

and by the convolution theorem, the Fourier transform of k will be $\hat{k}(\omega) = [\hat{p}_0(\omega)]^2$. On the other hand, by the RKHS property, $\hat{f}_\theta(\omega) = e^{i\omega\theta} [\hat{p}_0(\omega)]^2$. Hence, we can recover our kernel in the form

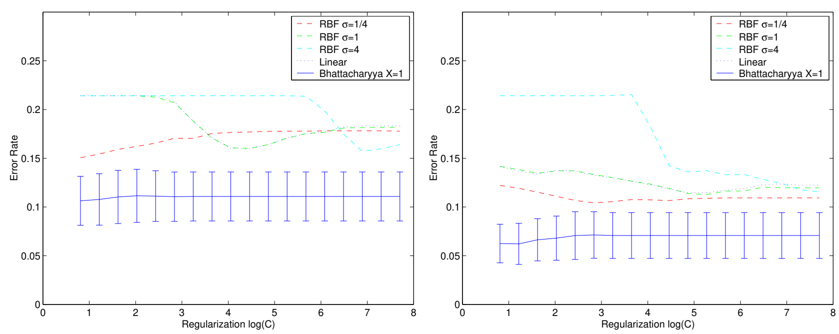
$$K(p_\theta, p_{\theta'}) = k(\theta' - \theta) = \int_\Theta (Pf_\theta)(\vartheta)\, (Pf_{\theta'})(\vartheta)\, d\vartheta = \int_{\hat{\Theta}} (\hat{P}\hat{f}_\theta)(\omega)\, \overline{(\hat{P}\hat{f}_{\theta'})(\omega)}\, d\omega$$

by setting $\hat{P} : \hat{f}(\omega) \mapsto \hat{f}(\omega) / |\, \hat{p}_0(\omega)\, |$.

The analogous result for "ordinary" stationary kernels has been well known for some time [22]. The significance of the above is that it explains the regularization properties implied by $K(\chi, \chi')$ in terms of the base distribution p_0 for our choice of models $\mathcal{P}$. For "smooth" distributions, $|\, \hat{p}_0(\omega)\, |$ drops off sharply with increasing $|\,\omega\,|$. For instance, for the unit Normal distribution, $\hat{p}_0(\omega) \sim e^{-\omega^2/2}$. The above expression for the regularization operator implies that our learning algorithm will correspondingly heavily penalize high frequency Fourier modes in $\hbar$, favoring hypotheses that appear "smooth" in the parameter space Θ.

8 Text Experiments

In one experiment we attempted to classify HTML documents for the freely available WebKB dataset using only the text component of each web page and discarding hyperlink information. Text was represented via a bag-of-words description which only tracks the frequency of appearance of words in each document without maintaining information on word orderings. The counts for each document are computed and normalized to sum to unity which corresponds to the maximum likelihood estimate of the document under a multinomial distribution over counts. This effectively gives the multinomial parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_D)$ subject to $\sum_{d=1}^{D} \alpha_d = 1$ for each document.



(a) Training Set Size = 77 (b) Training Set Size = 622

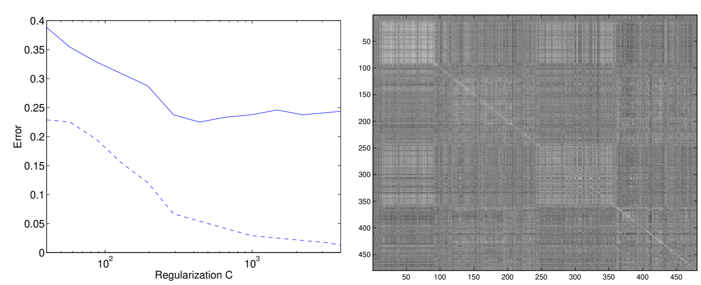
Fig. 2. SVM error rates (and standard deviation) for the Bhattacharyya kernel for multinomial models as well as error rates for traditional kernels on the WebKB dataset. Various levels of regularization are explored and various training set sizes are examined. Results are shown for 20-fold cross-validation.

SVMs were used to discriminate between two categories in the WebKB dataset: faculty web pages and student web pages. We compare the Bhattacharyya kernel for multinomials 6 (with the setting $X = 1$) against the (linear) dot product kernel and the Gaussian RBF kernel. The dataset contains a total of 1641 student web pages and 1124 faculty web pages. The data for each class is further split into 4 universities and 1 miscellaneous category and we performed the usual training and testing split as described by [16, 12] where testing is performed on a held out university. We averaged the results over 20-fold cross-validation and show the error rates for the various kernels in Figure 2.

The figure shows error rates for different sizes of the training set ranging over 77 and 622 training points. Each figure plots the average error rate of each kernel as a function of the SVM regularization parameter C. In addition, we show the standard deviation of the error rate for the Bhattacharyya kernel. Even though we only used a single arbitrary setting of $X = 1$ for the Bhattacharyya kernel, we note that it performs better than the linear kernel as well as the RBF at multiple settings of its σ parameter (where we attempted $\sigma = \{1/4, 1, 4\}$). Exploring other settings of X (or summing over all settings of X as previously discussed) as well as exploring various settings of λ to perform shrinkage-like regularization might further improve our results. Nevertheless, in this preliminary application the Bhattacharyya kernel is promising and the kernel provides a more appropriate affinity measure for count data which reduces error (although similar squashing functions on word frequencies have already been explored in text retrieval).

9 Sequence Experiments

In another preliminary experiment, we computed the expected likelihood kernel on HMMs for the SCOP protein sequence dataset [10, 17, 19]. These sequences are variable length discrete emissions from an alphabet of roughly 20 symbols. For simplicity, we only considered a single sub-task in the SCOP experiments, namely distinguishing proteins into negative and positive classes for SCOP subfamilies 2.1.1.4 and 2.1. We followed the same train and test split suggested for the SCOP 1.37 PDB-90 database experiments but reduced the size of the training set and testing set to keep computations simple. Therefore, we only used a total of 120 positive and 120 negative training examples and evaluated the resulting SVM on the appropriately held out 120 positive and 120 negative testing examples. The HMMs were trained on each sequence in the dataset. Thus, we have a total of 480 distinct HMM parameters with a fixed topology of 2 hidden states. Subsequently, we computed the Gram matrix over the whole dataset using the approach in Section 5.



(a) Error Rate (b) Gram Matrix

Fig. 3. SVM error rates for the expected likelihood kernel for HMMs. In (a) error under various levels of regularization is shown (dashed line is training error, solid is test error). In (b) the corresponding Gram matrix is shown.

Figure 3 shows the error rate under varying levels of regularization for the expected likelihood kernel. In addition, we show the Gram matrix which was verified to be positive definite by a singular value decomposition. One open issue is the potential of the individual HMMs to overfit under maximum likelihood due to the shortness of the sequences and the large alphabet size for protein sequences (this is less problematic with, e.g. gene sequences, which are longer yet have smaller 4-element alphabets). Further experiments and comparisons will be investigated in future work.

10 Discussion

We have introduced a new and simple kernel between probability distributions, the Probability Product Kernel, which eschews some of the complexities that kernels based on the Kullback-Leibler divergence often contend with. In special cases, our kernel reduces to Bhattacharyya's measure of similarity or the expected likelihood kernel. Furthermore, as a kernel between distributions the proposed computations are available in closed form for many common distributions and can be efficiently approximated in other cases.

To use the probability product kernel for learning from examples, we proposed the following general procedure. First, select a class of parametric generative models suitable for the data at hand. Then for each data point, estimate the parameters of the generative model using an appropriate frequentist or Bayesian procedure. Finally, for each pair of datapoints, define the kernel between them as the value of the probability product kernel between the corresponding distributions. The resulting kernel between datapoints can then be plugged into the kernel based learning algorithm of choice (SVM, Gaussian Process, Kernel ICA, etc.) for classification, regression or data analysis.

The proposed kernel marries discriminative learning frameworks with flexible generative modeling and can exploit advantages of both parametric an nonparametric approaches. We discussed the form our kernel takes for several members of the exponential family, mixture models, HMMs and Bayesian networks. For the special case of location families we also developed the regularization theory corresponding to our new kernel and discussed the link between the form of the distribution used as generative model and the regularization operator on parameter space. Experiments on text data and sequence data indicate that the approach is feasible and may be promising in practice.

Acknowledgments

Thanks to A. Jagota and R. Lyngsoe for profile HMM comparison code, C. Leslie and R. Kuang for SCOP data and the referees for important corrections.

References

1. F Aherne, N. Thacker, and P. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4):1–7, 1997.

2. O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory.* John Wiley & Sons, 1978.
3. Y. Bengio and P. Frasconi. Input-output HMM's for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, September 1996.
4. A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math Soc.*, 1943.
5. C. Bishop. *Neural Networks for Pattern Recognition.* Oxford Press, 1996.
6. M. Collins and N. Duffy. Convolution kernels for natural language. In *Neural Information Processing Systems 14*, 2002.
7. C. Cortes, P. Haffner, and M. Mohri. Rational kernels. In *Neural Information Processing Systems 15*, 2002.
8. F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural network architectures. *Neural Computation*, 7:219–269, 1995.
9. D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, 1999.
10. T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Neural Information Processing Systems 11*, 1998.
11. T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Neural Information Processing Systems 12*, 1999.
12. T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite kernels for hypertext categorisation. In *International Conference on Machine Learning*, 2001.
13. M. Jordan. *Learning in Graphical Models.* Kluwer Academic, 1997.
14. T. Kin, K. Tsuda, and K. Asai. Marginalized kernels for rna sequence data analysis. In *Proc. Genome Informatics*, 2002.
15. R. Kondor and T. Jebara. A kernel between sets of vectors. Machine Learning: Tenth International Conference, ICML 2003, February 2003.
16. J. Lafferty and G. Lebanon. Information diffusion kernels. In *Neural Information Processing Systems*, 2002.
17. C. Leslie, E. Eskin, J. Weston, and W.S. Noble. Mismatch string kernels for svm protein classification. In *Neural Information Processing Systems*, 2002.
18. H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, February 2002.
19. R.B. Lyngso, C.N.S. Pedersen, and H. Nielsen. Metrics and similarity measures for hidden markov models. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 1999.
20. C. Ong, A. Smola, and R. Williamson. Superkernels. In *Neural Information Processing Systems*, 2002.
21. C. Rathinavelu and L. Deng. Speech trajectory discrimination using the minimum classification error learning. In *IEEE Trans. on Speech and Audio Processing*, 1997.
22. A. J. Smola and B. Schölkopf. From regularization operators to support vector machines. In *Neural Information Processing Systems*, pages 343–349, 1998.
23. N. Tishby, W. Bialek, and F. Pereira. The information bottleneck method: Extracting relevant information from concurrent data. Technical report, NEC Research Institute, 1998.
24. F. Topsoe. Some inequalities for information divergence and related measures of discrimination. *J. of Inequalities in Pure and Applied Mathematics*, 2(1), 1999.
25. S.V.N. Vishawanathan and A.J. Smola. Fast kernels for string and tree matching. In *Neural Information Processing Systems 15*, 2002.
26. C. Watkins. *Advances in kernel methods*, chapter Dynamic Alignment Kernels. MIT Press, 2000.