Learning from Distributions via Support Measure Machines

Krikamol Muandet
MPI for Intelligent Systems, Tübingen
krikamol@tuebingen.mpg.de

Kenji Fukumizu
The Institute of Statistical Mathematics, Tokyo
fukumizu@ism.ac.jp

Francesco Dinuzzo
MPI for Intelligent Systems, Tübingen
fdinuzzo@tuebingen.mpg.de

Bernhard Schölkopf
MPI for Intelligent Systems, Tübingen
bs@tuebingen.mpg.de

Abstract

This paper presents a kernel-based discriminative learning framework on probability measures. Rather than relying on large collections of vectorial training examples, our framework learns using a collection of probability distributions that have been constructed to meaningfully represent training data. By representing these probability distributions as mean embeddings in the reproducing kernel Hilbert space (RKHS), we are able to apply many standard kernel-based learning techniques in straightforward fashion. To accomplish this, we construct a generalization of the support vector machine (SVM) called a support measure machine (SMM). Our analyses of SMMs provides several insights into their relationship to traditional SVMs. Based on such insights, we propose a flexible SVM (Flex-SVM) that places different kernel functions on each training example. Experimental results on both synthetic and real-world data demonstrate the effectiveness of our proposed framework.

1 Introduction

Discriminative learning algorithms are typically trained from large collections of vectorial training examples. In many classical learning problems, however, it is arguably more appropriate to represent training data not as individual data points, but as probability distributions. There are, in fact, multiple reasons why probability distributions may be preferable.

Firstly, uncertain or missing data naturally arises in many applications. For example, gene expression data obtained from the microarray experiments are known to be very noisy due to various sources of variabilities [1]. In order to reduce uncertainty, and to allow for estimates of confidence levels, experiments are often replicated. Unfortunately, the feasibility of replicating the microarray experiments is often inhibited by cost constraints, as well as the amount of available mRNA. To cope with experimental uncertainty given a limited amount of data, it is natural to represent each array as a probability distribution that has been designed to approximate the variability of gene expressions across slides.

Probability distributions may be equally appropriate given an abundance of training data. In data-rich disciplines such as neuroinformatics, climate informatics, and astronomy, a high throughput experiment can easily generate a huge amount of data, leading to significant computational challenges in both time and space. Instead of scaling up one's learning algorithms, one can scale down one's dataset by constructing a smaller collection of distributions which represents groups of similar samples. Besides computational efficiency, aggregate statistics can potentially incorporate higher-level information that represents the collective behavior of multiple data points.

Previous attempts have been made to learn from distributions by creating positive definite (p.d.) kernels on probability measures. In [2], the probability product kernel (PPK) was proposed as a generalized inner product between two input objects. This kernel is in fact closely related to well-known kernels such as the Bhattacharyya kernel [3] and the exponential symmetrized Kullback-Leibler (KL) divergence [4]. In [5], an extension of a two-parameter family of Hilbertian metrics of Topsøe was used to define Hilbertian kernels on probability measures. In [6], the semi-group kernels were designed for objects with additive semi-group structure such as positive measures. Recently, [7] introduced nonextensive information theoretic kernels on probability measures based on new Jensen-Shannon-type divergences. Although these kernels have proven successful in many applications, they are designed specifically for certain properties of distributions and application domains. Moreover, there has been no attempt in making a connection to the kernels on corresponding input spaces.

The contributions of this paper can be summarized as follows. First, we prove the representer theorem for a regularization framework over the space of probability distributions, which is a generalization of regularization over the input space on which the distributions are defined (Section 2). Second, a family of positive definite kernels on distributions is introduced (Section 3). Based on such kernels, a learning algorithm on probability measures called *support measure machine* (SMM) is proposed. An SVM on the input space is provably a special case of the SMM. Third, the paper presents the relations between sample-based and distribution-based methods (Section 4). If the distributions depend only on the locations in the input space, the SMM particularly reduces to a more flexible SVM that places different kernels on each data point.

2 Regularization on probability distributions

Given a non-empty set $\mathcal{X}$, let $\mathscr{P}$ denote the set of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \mathcal{A})$, where $\mathcal{A}$ is a σ-algebra of subsets of $\mathcal{X}$. The goal of this work is to learn a function $h : \mathscr{P} \to \mathcal{Y}$ given a set of example pairs $\{(\mathbb{P}_i, y_i)\}_{i=1}^m$, where $\mathbb{P}_i \in \mathscr{P}$ and $y_i \in \mathcal{Y}$. In other words, we consider a supervised setting in which input training examples are probability distributions. In this paper, we focus on the binary classification problem, i.e., $\mathcal{Y} = \{+1, -1\}$.

In order to learn from distributions, we employ a compact representation that not only preserves necessary information of individual distributions, but also permits efficient computations. That is, we adopt a Hilbert space embedding to represent the distribution as a mean function in an RKHS [8, 9]. Formally, let $\mathcal{H}$ denote an RKHS of functions $f : \mathcal{X} \to \mathbb{R}$, endowed with a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The mean map from $\mathscr{P}$ into $\mathcal{H}$ is defined as

$$\mu : \mathscr{P} \to \mathcal{H}, \ \ \mathbb{P} \longmapsto \int_{\mathcal{X}} k(x, \cdot) \, \mathrm{d}\mathbb{P}(x) \ . \tag{1}$$

We assume that $k(x, \cdot)$ is bounded for any $x \in \mathcal{X}$. It can be shown that, if k is characteristic, the map (1) is injective, i.e., all the information about the distribution is preserved [10]. For any $\mathbb{P}$, letting $\mu_{\mathbb{P}} = \mu(\mathbb{P})$, we have the reproducing property

$$\mathbb{E}_{\mathbb{P}}[f] = \langle \mu_{\mathbb{P}}, f \rangle_{\mathcal{H}}, \ \ \forall f \in \mathcal{H} \ . \tag{2}$$

That is, we can see the mean embedding $\mu_{\mathbb{P}}$ as a feature map associated with the kernel $K : \mathscr{P} \times \mathscr{P} \to \mathbb{R}$, defined as $K(\mathbb{P}, \mathbb{Q}) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$. Since $\sup_x \|k(x, \cdot)\|_{\mathcal{H}} < \infty$, it also follows that $K(\mathbb{P}, \mathbb{Q}) = \iint \langle k(x, \cdot), k(z, \cdot) \rangle_{\mathcal{H}} \, \mathrm{d}\mathbb{P}(x) \, \mathrm{d}\mathbb{Q}(z) = \iint k(x, z) \, \mathrm{d}\mathbb{P}(x) \, \mathrm{d}\mathbb{Q}(z)$, where the second equality follows from the reproducing property of $\mathcal{H}$. It is immediate that K is a p.d. kernel on $\mathscr{P}$.

The following theorem shows that optimal solutions of a suitable class of regularization problems involving distributions can be expressed as a finite linear combination of mean embeddings.

Theorem 1. *Given training examples $(\mathbb{P}_i, y_i) \in \mathscr{P} \times \mathbb{R}$, $i = 1, \ldots, m$, a strictly monotonically increasing function $\Omega : [0, +\infty) \to \mathbb{R}$, and a loss function $\ell : (\mathscr{P} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{+\infty\}$, any $f \in \mathcal{H}$ minimizing the regularized risk functional*

$$\ell \left(\mathbb{P}_1, y_1, \mathbb{E}_{\mathbb{P}_1}[f], \ldots, \mathbb{P}_m, y_m, \mathbb{E}_{\mathbb{P}_m}[f] \right) + \Omega \left(\|f\|_{\mathcal{H}} \right) \tag{3}$$

admits a representation of the form $f = \sum_{i=1}^m \alpha_i \mu_{\mathbb{P}_i}$ for some $\alpha_i \in \mathbb{R}$, $i = 1, \ldots, m$.

Theorem 1 clearly indicates how each distribution contributes to the minimizer of (3). Roughly speaking, the coefficients α_i controls the contribution of the distributions through the mean embeddings $\mu_{\mathbb{P}_i}$. Furthermore, if we restrict $\mathscr{P}$ to a class of Dirac measures δ_x on $\mathcal{X}$ and consider the training set $\{(\delta_{x_i}, y_i)\}_{i=1}^m$, the functional (3) reduces to the usual regularization functional [11] and the solution reduces to $f = \sum_{i=1}^m \alpha_i k(x_i, \cdot)$. Therefore, the standard representer theorem is recovered as a particular case (see also [12] for more general results on representer theorem).

Note that, on the one hand, the minimization problem (3) is different from minimizing the functional $\mathbb{E}_{\mathbb{P}_1} \dots \mathbb{E}_{\mathbb{P}_m} \ell(x_1, y_1, f(x_1), \dots, x_m, y_m, f(x_m)) + \Omega(\|f\|_{\mathcal{H}})$ for the special case of the additive loss ℓ. Therefore, the solution of our regularization problem is different from what one would get in the limit by training on an infinitely many points sampled from $\mathbb{P}_1, \dots, \mathbb{P}_m$. On the other hand, it is also different from minimizing the functional $\ell(M_1, y_1, f(M_1), \dots, M_m, y_m, f(M_m)) + \Omega(\|f\|_{\mathcal{H}})$ where $M_i = \mathbb{E}_{x\sim\mathbb{P}_i}[x]$. In a sense, our framework is something in between.

3 Kernels on probability distributions

As the map (1) is linear in $\mathscr{P}$, optimizing the functional (3) amounts to finding a function in $\mathcal{H}$ that approximate well functions from $\mathscr{P}$ to $\mathbb{R}$ in the function class $\mathcal{F} \triangleq \{\mathbb{P} \to \int_{\mathcal{X}} g \, \mathrm{d}\mathbb{P} \,|\, \mathbb{P} \in \mathscr{P}, \, g \in C(\mathcal{X})\}$ where $C(\mathcal{X})$ is a class of bounded continuous functions on $\mathcal{X}$. Since $\delta_x \in \mathscr{P}$ for any $x \in \mathcal{X}$, it follows that $C(\mathcal{X}) \subset \mathcal{F} \subset C(\mathscr{P})$ where $C(\mathscr{P})$ is a class of bounded continuous functions on $\mathscr{P}$ endowed with the topology of weak convergence and the associated Borel σ-algebra. The following lemma states the relation between the RKHS $\mathcal{H}$ induced by the kernel k and the function class $\mathcal{F}$.

Lemma 2. *Assuming that $\mathcal{X}$ is compact, the RKHS $\mathcal{H}$ induced by a kernel k is dense in $\mathcal{F}$ if k is universal, i.e., for every function $F \in \mathcal{F}$ and every $\varepsilon > 0$ there exists a function $g \in \mathcal{H}$ with $\sup_{\mathbb{P}\in\mathscr{P}} |F(\mathbb{P}) - \int g \, \mathrm{d}\mathbb{P}| \leq \varepsilon$.*

Proof. Assume that k is universal. Then, for every function $f \in C(\mathcal{X})$ and every $\varepsilon > 0$ there exists a function $g \in \mathcal{H}$ induced by k with $\sup_{x\in\mathcal{X}} |f(x) - g(x)| \leq \varepsilon$ [13]. Hence, by linearity of $\mathcal{F}$, for every $F \in \mathcal{F}$ and every $\varepsilon > 0$ there exists a function $h \in \mathcal{H}$ such that $\sup_{\mathbb{P}\in\mathscr{P}} |F(\mathbb{P}) - \int h \, \mathrm{d}\mathbb{P}| \leq \varepsilon$. ■

Nonlinear kernels on $\mathscr{P}$ can be defined in an analogous way to nonlinear kernels on $\mathcal{X}$, by treating mean embeddings $\mu_{\mathbb{P}}$ of $\mathbb{P} \in \mathscr{P}$ as its feature representation. First, assume that the map (1) is injective and let $\langle \cdot, \cdot \rangle_{\mathscr{P}}$ be an inner product on $\mathscr{P}$. By linearity, we have $\langle \mathbb{P}, \mathbb{Q} \rangle_{\mathscr{P}} = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$ (cf. [8] for more details). Then, the nonlinear kernels on $\mathscr{P}$ can be defined as $K(\mathbb{P}, \mathbb{Q}) = \kappa(\mu_{\mathbb{P}}, \mu_{\mathbb{Q}}) = \langle \psi(\mu_{\mathbb{P}}), \psi(\mu_{\mathbb{Q}}) \rangle_{\mathcal{H}_\kappa}$ where κ is a p.d. kernel. As a result, many standard nonlinear kernels on $\mathcal{X}$ can be used to define nonlinear kernels on $\mathscr{P}$ as long as the kernel evaluation depends entirely on the inner product $\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$, e.g., $K(\mathbb{P}, \mathbb{Q}) = (\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + c)^d$. Although requiring more computational effort, their practical use is simple and flexible. Specifically, the notion of p.d. kernels on distributions proposed in this work is so generic that standard kernel functions can be reused to derive kernels on distributions that are different from many other kernel functions proposed specifically for certain distributions.

It has been recently proved that the Gaussian RBF kernel given by $K(\mathbb{P}, \mathbb{Q}) = \exp(-\frac{\gamma}{2}\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}^2), \ \forall \mathbb{P}, \mathbb{Q} \in \mathscr{P}$ is universal w.r.t $C(\mathscr{P})$ given that $\mathcal{X}$ is compact and the map μ is injective [14]. Despite its success in real-world applications, the theory of kernel-based classifiers beyond the input space $\mathcal{X} \subset \mathbb{R}^d$, as also mentioned by [14], is still incomplete. It is therefore of theoretical interest to consider more general classes of universal kernels on probability distributions.

3.1 Support measure machines

This subsection extends SVMs to deal with probability distributions, leading to *support measure machines* (SMMs). In its general form, an SMM amounts to solving an SVM problem with the expected kernel $K(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{x\sim\mathbb{P}, z\sim\mathbb{Q}}[k(x, z)]$. This kernel can be computed in closed-form for certain classes of distributions and kernels k. Examples are given in Table 1.

Alternatively, one can approximate the kernel $K(\mathbb{P}, \mathbb{Q})$ by the empirical estimate:

$$K_{\text{emp}}(\widehat{\mathbb{P}}_n, \widehat{\mathbb{Q}}_m) = \frac{1}{n \cdot m} \sum_{i=1}^n \sum_{j=1}^m k(x_i, z_j) \tag{4}$$

where $\widehat{\mathbb{P}}_n$ and $\widehat{\mathbb{Q}}_m$ are empirical distributions of $\mathbb{P}$ and $\mathbb{Q}$ given random samples $\{x_i\}_{i=1}^n$ and $\{z_j\}_{j=1}^m$, respectively. A finite sample of size m from a distribution $\mathbb{P}$ suffices (with high probability)

Table 1: the analytic forms of expected kernels for different choices of kernels and distributions.

Distributions	**Embedding kernel** $k(x,y)$	$K(\mathbb{P}_i,\mathbb{P}_j)=\langle\mu_{\mathbb{P}_i},\mu_{\mathbb{P}_j}\rangle_{\mathcal{H}}$
Arbitrary $\mathbb{P}(m;\Sigma)$	Linear $\langle x,y\rangle$	$m_i^\mathsf{T} m_j+\delta_{ij}\mathrm{tr}\,\Sigma_i$
Gaussian $\mathcal{N}(m;\Sigma)$	Gaussian RBF $\exp(-\frac{\gamma}{2}\\|x-y\\|^2)$	$\exp(-\frac{1}{2}(m_i-m_j)^\mathsf{T}(\Sigma_i+\Sigma_j+\gamma^{-1}\mathbf{I})^{-1}(m_i-m_j))$ $/\lvert\gamma\Sigma_i+\gamma\Sigma_j+\mathbf{I}\rvert^{\frac{1}{2}}$
Gaussian $\mathcal{N}(m;\Sigma)$	Polynomial degree 2 $(\langle x,y\rangle+1)^2$	$(\langle m_i,m_j\rangle+1)^2+\mathrm{tr}\,\Sigma_i\Sigma_j+m_i^\mathsf{T}\Sigma_j m_i+m_j^\mathsf{T}\Sigma_i m_j$
Gaussian $\mathcal{N}(m;\Sigma)$	Polynomial degree 3 $(\langle x,y\rangle+1)^3$	$(\langle m_i,m_j\rangle+1)^3+6m_i^\mathsf{T}\Sigma_i\Sigma_j m_j$ $+3(\langle m_i,m_j\rangle+1)(\mathrm{tr}\,\Sigma_i\Sigma_j+m_i^\mathsf{T}\Sigma_j m_i+m_j^\mathsf{T}\Sigma_i m_j)$

to compute an approximation within an error of $O(m^{-\frac{1}{2}})$. Instead, if the sample set is sufficiently large, one may choose to approximate the true distribution by simpler probabilistic models, e.g., a mixture of Gaussians model, and choose a kernel k whose expected value admits an analytic form. Storing only the parameters of probabilistic models may save some space compared to storing all data points.

Note that the standard SVM feature map $\phi(x)$ is usually nonlinear in x, whereas $\mu_{\mathbb{P}}$ is *linear* in $\mathbb{P}$. Thus, for an SMM, the first level kernel k is used to obtain a vectorial representation of the measures, and the second level kernel K allows for a nonlinear algorithm on distributions. For clarity, we will refer to k and K as the **embedding kernel** and the **level-2 kernel**, respectively

4 Theoretical analyses

This section presents key theoretical aspects of the proposed framework, which reveal important connection between kernel-based learning algorithms on the space of distributions and on the input space on which they are defined.

4.1 Risk deviation bound

Given a training sample $\{(\mathbb{P}_i,y_i)\}_{i=1}^m$ drawn i.i.d. from some unknown probability distribution $\mathcal{P}$ on $\mathscr{P}\times\mathcal{Y}$, a loss function $\ell:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$, and a function class Λ, the goal of statistical learning is to find the function $f\in\Lambda$ that minimizes the expected risk functional $\mathcal{R}(f)=\int_{\mathscr{P}}\int_{\mathcal{X}}\ell(y,f(x))\,\mathrm{d}\mathbb{P}(x)\,\mathrm{d}\mathcal{P}(\mathbb{P},y)$. Since $\mathcal{P}$ is unknown, the empirical risk $\mathcal{R}_{\text{emp}}(f)=\frac{1}{m}\sum_{i=1}^m\int_{\mathcal{X}}\ell(y_i,f(x))\,\mathrm{d}\mathbb{P}_i(x)$ based on the training sample is considered instead. Furthermore, the risk functional can be simplified further by considering $\frac{1}{m\cdot n}\sum_{i=1}^m\sum_{x_{ij}\sim\mathbb{P}_i}\ell(y_i,f(x_{ij}))$ based on n samples x_{ij} drawn from each $\mathbb{P}_i$.

Our framework, on the other hand, alleviates the problem by minimizing the risk functional $\mathcal{R}^\mu(f)=\int_{\mathscr{P}}\ell(y,\mathbb{E}_{\mathbb{P}}[f(x)])\,\mathrm{d}\mathcal{P}(\mathbb{P},y)$ for $f\in\mathcal{H}$ with corresponding empirical risk functional $\mathcal{R}^\mu_{\text{emp}}(f)=\frac{1}{m}\sum_{i=1}^m\ell(y_i,\mathbb{E}_{\mathbb{P}_i}[f(x)])$ (cf. the discussion at the end of Section 2). It is often easier to optimize $\mathcal{R}^\mu_{\text{emp}}(f)$ as the expectation can be computed exactly for certain choices of $\mathbb{P}_i$ and $\mathcal{H}$. Moreover, for universal $\mathcal{H}$, this simplification preserves all information of the distributions. Nevertheless, there is still a loss of information due to the loss function ℓ.

Due to the i.i.d. assumption, the analysis of the difference between $\mathcal{R}$ and $\mathcal{R}^\mu$ can be simplified w.l.o.g. to the analysis of the difference between $\mathbb{E}_{\mathbb{P}}[\ell(y,f(x))]$ and $\ell(y,\mathbb{E}_{\mathbb{P}}[f(x)])$ for a particular distribution $\mathbb{P}\in\mathscr{P}$. The theorem below provides a bound on the difference between $\mathbb{E}_{\mathbb{P}}[\ell(y,f(x))]$ and $\ell(y,\mathbb{E}_{\mathbb{P}}[f(x)])$.

Theorem 3. *Given an arbitrary probability distribution $\mathbb{P}$ with variance σ^2, a Lipschitz continuous function $f:\mathbb{R}\to\mathbb{R}$ with constant C_f, an arbitrary loss function $\ell:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ that is Lipschitz continuous in the second argument with constant C_ℓ, it follows that $|\mathbb{E}_{x\sim\mathbb{P}}[\ell(y,f(x))]-\ell(y,\mathbb{E}_{x\sim\mathbb{P}}[f(x)])|\leq 2C_\ell C_f\sigma$ for any $y\in\mathbb{R}$.*

Theorem 3 indicates that if the random variable x is concentrated around its mean and the function f and ℓ are well-behaved, i.e., Lipschitz continuous, then the loss deviation $|\mathbb{E}_{\mathbb{P}}[\ell(y,f(x))]-\ell(y,\mathbb{E}_{\mathbb{P}}[f(x)])|$ will be small. As a result, if this holds for any distribution $\mathbb{P}_i$ in the training set $\{(\mathbb{P}_i,y_i)\}_{i=1}^m$, the true risk deviation $|\mathcal{R}-\mathcal{R}^\mu|$ is also expected to be small.

4.2 Flexible support vector machines

It turns out that, for certain choices of distributions $\mathbb{P}$, the linear SMM trained using $\{(\mathbb{P}_i, y_i)\}_{i=1}^m$ is equivalent to an SVM trained using some samples $\{(x_i, y_i)\}_{i=1}^m$ with an appropriate choice of kernel function.

Lemma 4. *Let $k(x, z)$ be a bounded p.d. kernel on a measure space such that $\iint k(x, z)^2 \,\mathrm{d}x \,\mathrm{d}z < \infty$, and $g(x, \tilde{x})$ be a square integrable function such that $\int g(x, \tilde{x}) \,\mathrm{d}\tilde{x} < \infty$ for all x. Given a sample $\{(\mathbb{P}_i, y_i)\}_{i=1}^m$ where each $\mathbb{P}_i$ is assumed to have a density given by $g(x_i, x)$, the linear SMM is equivalent to the SVM on the training sample $\{(x_i, y_i)\}_{i=1}^m$ with kernel $K_g(x, z) = \iint k(\tilde{x}, \tilde{z}) g(x, \tilde{x}) g(z, \tilde{z}) \,\mathrm{d}\tilde{x} \,\mathrm{d}\tilde{z}$.*

Note that the important assumption for this equivalence is that the distributions $\mathbb{P}_i$ differ only in their location in the parameter space. This need not be the case in all possible applications of SMMs.

Furthermore, we have $K_g(x, z) = \left\langle \int k(\tilde{x}, \cdot) g(x, \tilde{x}) \,\mathrm{d}\tilde{x}, \int k(\tilde{z}, \cdot) g(z, \tilde{z}) \,\mathrm{d}\tilde{z} \right\rangle_{\mathcal{H}}$. Thus, it is clear that the feature map of x depends not only on the kernel k, but also on the density $g(x, \tilde{x})$. Consequently, by virtue of Lemma 4, the kernel K_g allows the SVM to place different kernels at each data point. We call this algorithm a *flexible SVM* (Flex-SVM).

Consider for example the linear SMM with Gaussian distributions $\mathcal{N}(x_1; \sigma_1^2 \cdot \mathbf{I}), \ldots, \mathcal{N}(x_m; \sigma_m^2 \cdot \mathbf{I})$ and Gaussian RBF kernel k_{σ^2} with bandwidth parameter σ. The convolution theorem of Gaussian distributions implies that this SMM is equivalent to a flexible SVM that places a data-dependent kernel $k_{\sigma^2 + 2\sigma_i^2}(x_i, \cdot)$ on training example x_i, i.e., a Gaussian RBF kernel with larger bandwidth.

5 Related works

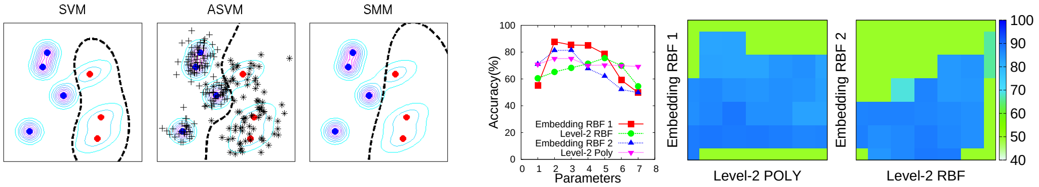
The kernel $K(\mathbb{P}, \mathbb{Q}) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$ is in fact a special case of the Hilbertian metric [5], with the associated kernel $K(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(x, \tilde{x})]$, and a generative mean map kernel (GMMK) proposed by [15]. In the GMMK, the kernel between two objects x and y is defined via $\hat{p}_x$ and $\hat{p}_y$, which are estimated probabilistic models of x and y, respectively. That is, a probabilistic model $\hat{p}_x$ is learned for each example and used as a surrogate to construct the kernel between those examples. The idea of surrogate kernels has also been adopted by the Probability Product Kernel (PPK) [2]. In this case, we have $K_\rho(p, p') = \int_{\mathcal{X}} p(x)^\rho p'(x)^\rho \,\mathrm{d}x$, which has been shown to be a special case of GMMK when $\rho = 1$ [15]. Consequently, GMMK, PPK with $\rho = 1$, and our linear kernels are equivalent when the embedding kernel is $k(x, x') = \delta(x - x')$.

The use of expected kernels in dealing with the uncertainty in the input data has a connection to robust SVMs. For instance, a generalized form of the SVM in [16] incorporates the probabilistic uncertainty into the maximization of the margin. This results in a second-order cone programming (SOCP) that generalizes the standard SVM. In SOCP, one needs to specify the parameter τ_i that reflects the probability of correctly classifying the ith training example. The parameter τ_i is therefore closely related to the parameter σ_i, which specifies the variance of the distribution centered at the ith example. [17] showed the equivalence between SVMs using expected kernels and SOCP when $\tau_i = 0$. When $\tau_i > 0$, the mean and covariance of missing kernel entries have to be estimated explicitly, making the SOCP more involved for nonlinear kernels. Although achieving comparable performance to the standard SVM with expected kernels, the SOCP requires a more computationally extensive SOCP solver, as opposed to simple quadratic programming (QP).

6 Experimental results

In the experiments, we primarily consider three different learning algorithms: i) **SVM** is considered as a baseline algorithm. ii) **Augmented SVM (ASVM)** is an SVM trained on augmented samples drawn according to the distributions $\{\mathbb{P}_i\}_{i=1}^m$. The same number of examples are drawn from each distribution. iii) **SMM** is our distribution-based method that can be applied directly on the distributions[1].

[1] We used the LIBSVM implementation.



(a) decision boundaries.

(b) sensitivity of kernel parameters

Figure 1: (a) the decision boundaries of SVM, ASVM, and SMM. (b) the heatmap plots of average accuracies of SMM over 30 experiments using POLY-RBF (center) and RBF-RBF (right) kernel combinations with the plots of average accuracies at different parameter values (left).

Table 2: accuracies (%) of SMM on synthetic data with different combinations of embedding and level-2 kernels.

		Embedding kernels				
		LIN	POLY2	POLY3	RBF	URBF
Level-2 kernels	LIN	85.20±2.20	81.04±3.11	81.10±2.76	87.74±2.19	85.39±2.56
	POLY	83.95±2.11	81.34±1.21	82.66±1.75	88.06±1.73	86.84±1.51
	RBF	87.80±1.96	73.12±3.29	78.28±2.19	**89.65±1.37**	86.86±1.88

6.1 Synthetic data

Firstly, we conducted a basic experiment that illustrates a fundamental difference between SVM, ASVM, and SMM. A binary classification problem of 7 Gaussian distributions with different means and covariances was considered. We trained the SVM using only the means of the distributions, ASVM with 30 virtual examples generated from each distribution, and SMM using distributions as training examples. A Gaussian RBF kernel with $\gamma = 0.25$ was used for all algorithms.

Figure 1a shows the resulting decision boundaries. Having been trained only on means of the distributions, the SVM classifier tends to overemphasize the regions with high densities and underrepresent the lower density regions. In contrast, the ASVM is more expensive and sensitive to outliers, especially when learning on heavy-tailed distributions. The SMM treats each distribution as a training example and implicitly incorporates properties of the distributions, i.e., means and covariances, into the classifier. Note that the SVM can be trained to achieve a similar result to the SMM by choosing an appropriate value for γ (cf. Lemma 4). Nevertheless, this becomes more difficult if the training distributions are, for example, nonisotropic and have different covariance matrices.

Secondly, we evaluate the performance of the SMM for different combinations of embedding and level-2 kernels. Two classes of synthetic Gaussian distributions on $\mathbb{R}^{10}$ were generated. The mean parameters of the positive and negative distributions are normally distributed with means $m^+ = (1,\dots,1)$ and $m^- = (2,\dots,2)$ and identical covariance matrix $\Sigma = 0.5 \cdot \mathbf{I}_{10}$, respectively. The covariance matrix for each distribution is generated according to two Wishart distributions with covariance matrices given by $\Sigma^+ = 0.6 \cdot \mathbf{I}_{10}$ and $\Sigma^- = 1.2 \cdot \mathbf{I}_{10}$ with 10 degrees of freedom. The training set consists of 500 distributions from the positive class and 500 distributions from the negative class. The test set consists of 200 distributions with the same class proportion to the training set.

The kernels used in the experiment include linear kernel (LIN), polynomial kernel of degree 2 (POLY2), polynomial kernel of degree 3 (POLY3), unnormalized Gaussian RBF kernel (RBF), and normalized Gaussian RBF kernel (NRBF). To fix parameter values of both kernel functions and SMM, 10-fold cross-validation (10-CV) is performed on a parameter grid, $C \in \{2^{-3}, 2^{-2}, \dots, 2^7\}$ for SMM, bandwidth parameter $\gamma \in \{10^{-3}, 10^{-2}, \dots, 10^2\}$ for Gaussian RBF kernels, and degree parameter $d \in \{2,3,4,5,6\}$ for polynomial kernels. The average accuracy and ±1 standard deviation for all kernel combinations over 30 repetitions are reported in Table 2. Moreover, we also investigate the sensitivity of kernel parameters for two kernel combinations: RBF-RBF and POLY-RBF. In this case, we consider the bandwidth parameter $\gamma = \{10^{-3}, 10^{-2}, \dots, 10^3\}$ for Gaussian

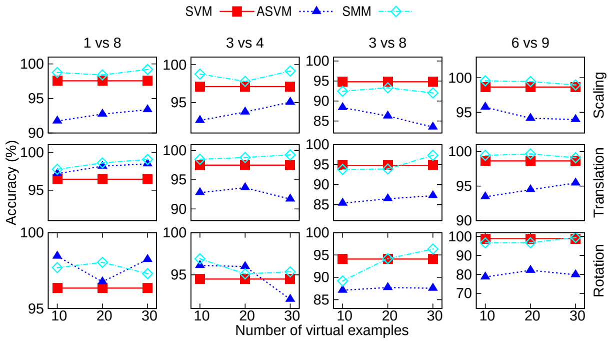


Figure 2: the performance of SVM, ASVM, and SMM algorithms on handwritten digits constructed using three basic transformations.

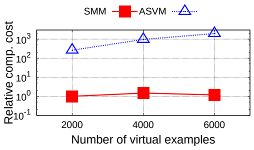


Figure 3: relative computational cost of ASVM and SMM (baseline: SMM with 2000 virtual examples).

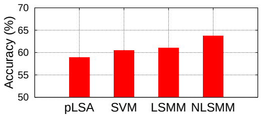


Figure 4: accuracies of four different techniques for natural scene categorization.

RBF kernels and degree parameter $d = \{2, 3, \ldots, 8\}$ for polynomial kernels. Figure 1b depicts the accuracy values and average accuracies for considered kernel functions.

The results in Table 2 indicate that both embedding and level-2 kernels are important for the performance of the classifier. The embedding kernels tend to have more impact on the predictive performance compared to the level-2 kernels. This conclusion also coincides with the results depicted in Figure 1b.

6.2 Handwritten digit recognition

In this section, the proposed framework is applied to distributions over equivalence classes of images that are invariant to basic transformations, namely, *scaling*, *translation*, and *rotation*. We consider the handwritten digits obtained from the USPS dataset. For each 16×16 image, the distribution over the equivalence class of the transformations is determined by a prior on parameters associated with such transformations. Scaling and translation are parametrized by the scale factors (s_x, s_y) and displacements (t_x, t_y) along the x and y axes, respectively. The rotation is parametrized by an angle θ. We adopt Gaussian distributions as prior distributions, including $\mathcal{N}([1, 1], 0.1 \cdot \mathbf{I}_2)$, $\mathcal{N}([0, 0], 5 \cdot \mathbf{I}_2)$, and $\mathcal{N}(0; \pi)$. For each image, the virtual examples are obtained by sampling parameter values from the distribution and applying the transformation accordingly.

Experiments are categorized into simple and difficult binary classification tasks. The former consists of classifying digit 1 against digit 8 and digit 3 against digit 4. The latter considers classifying digit 3 against digit 8 and digit 6 against digit 9. The initial dataset for each task is constructed by randomly selecting 100 examples from each class. Then, for each example in the initial dataset, we generate 10, 20, and 30 virtual examples using the aforementioned transformations to construct virtual data sets consisting of 2,000, 4,000, and 6,000 examples, respectively. One third of examples in the initial dataset are used as a test set. The original examples are excluded from the virtual datasets. The virtual examples are normalized such that their feature values are in $[0, 1]$. Then, to reduce computational cost, principle component analysis (PCA) is performed to reduce the dimensionality to 16. We compare the SVM on the initial dataset, the ASVM on the virtual datasets, and the SMM. For SVM and ASVM, the Gaussian RBF kernel is used. For SMM, we employ the empirical kernel (4) with Gaussian RBF kernel as a base kernel. The parameters of the algorithms are fixed by 10-CV over parameters $C \in \{2^{-3}, 2^{-2}, \ldots, 2^7\}$ and $\gamma \in \{0.01, 0.1, 1\}$.

The results depicted in Figure 2 clearly demonstrate the benefits of learning directly from the equivalence classes of digits under basic transformations[2]. In most cases, the SMM outperforms both the SVM and the ASVM as the number of virtual examples increases. Moreover, Figure 3 shows the benefit of the SMM over the ASVM in term of computational cost[3].

[2] While the reported results were obtained using virtual examples with Gaussian parameter distributions (Sec. 6.2), we got similar results using uniform distributions.

[3] The evaluation was made on a 64-bit desktop computer with Intel® Core™ 2 Duo CPU E8400 at 3.00Ghz×2 and 4GB of memory.

6.3 Natural scene categorization

This section illustrates benefits of the nonlinear kernels between distributions for learning natural scene categories in which the bag-of-word (BoW) representation is used to represent images in the dataset. Each image is represented as a collection of local patches, each being a codeword from a large vocabulary of codewords called codebook. Standard BoW representations encode each image as a histogram that enumerates the occurrence probability of local patches detected in the image w.r.t. those in the codebook. On the other hand, our setting represents each image as a distribution over these codewords. Thus, images of different scenes tends to generate distinct set of patches. Based on this representation, both the histogram and the local patches can be used in our framework.

We use the dataset presented in [18]. According to their results, most errors occurs among the four indoor categories (830 images), namely, bedroom (174 images), living room (289 images), kitchen (151 images), and office (216 images). Therefore, we will focus on these four categories. For each category, we split the dataset randomly into two separate sets of images, 100 for training and the rest for testing.

A codebook is formed from the training images of all categories. Firstly, interesting keypoints in the image are randomly detected. Local patches are then generated accordingly. After patch detection, each patch is transformed into a 128-dim SIFT vector [19]. Given the collection of detected patches, K-means clustering is performed over all local patches. Codewords are then defined as the centers of the learned clusters. Then, each patch in an image is mapped to a codeword and the image can be represented by the histogram of the codewords. In addition, we also have an $M \times 128$ matrix of SIFT vectors where M is the number of codewords.

We compare the performance of a Probabilistic Latent Semantic Analysis (pLSA) with the standard BoW representation, SVM, linear SMM (LSMM), and nonlinear SMM (NLSMM). For SMM, we use the empirical embedding kernel with Gaussian RBF base kernel k: $K(\mathbf{h}_i, \mathbf{h}_j) = \sum_{r=1}^{M} \sum_{s=1}^{M} h_i(c_r)h_j(c_s)k(c_r, c_s)$ where $\mathbf{h}_i$ is the histogram of the ith image and c_r is the rth SIFT vector. A Gaussian RBF kernel is also used as the level-2 kernel for nonlinear SMM. For the SVM, we adopt a Gaussian RBF kernel with χ^2-distance between the histograms [20], i.e., $K(\mathbf{h}_i, \mathbf{h}_j) = \exp\left(-\gamma\chi^2(\mathbf{h}_i, \mathbf{h}_j)\right)$ where $\chi^2(\mathbf{h}_i, \mathbf{h}_j) = \sum_{r=1}^{M} \frac{(h_i(c_r)-h_j(c_r))^2}{h_i(c_r)+h_j(c_r)}$. The parameters of the algorithms are fixed by 10-CV over parameters $C \in \{2^{-3}, 2^{-2}, \ldots, 2^7\}$ and $\gamma \in \{0.01, 0.1, 1\}$. For NLSMM, we use the best γ of LSMM in the base kernel and perform 10-CV to choose γ parameter only for the level-2 kernel. To deal with multiple categories, we adopt the pairwise approach and voting scheme to categorize test images. The results in Figure 4 illustrate the benefit of the distribution-based framework. Understanding the context of a complex scene is challenging. Employing distribution-based methods provides an elegant way of utilizing higher-order statistics in natural images that could not be captured by traditional sample-based methods.

7 Conclusions

This paper proposes a method for kernel-based discriminative learning on probability distributions. The trick is to embed distributions into an RKHS, resulting in a simple and efficient learning algorithm on distributions. A family of linear and nonlinear kernels on distributions allows one to flexibly choose the kernel function that is suitable for the problems at hand. Our analyses provide insights into the relations between distribution-based methods and traditional sample-based methods, particularly the flexible SVM that allows the SVM to place different kernels on each training example. The experimental results illustrate the benefits of learning from a pool of distributions, compared to a pool of examples, both on synthetic and real-world data.

Acknowledgments

KM would like to thank Zoubin Gharamani, Arthur Gretton, Christian Walder, and Philipp Hennig for a fruitful discussion. We also thank all three insightful reviewers for their invaluable comments.

References

[1] Y. H. Yang and T. Speed. Design issues for cDNA microarray experiments. *Nat. Rev. Genet.*, 3(8):579–588, 2002.

[2] T. Jebara, R. Kondor, A. Howard, K. Bennett, and N. Cesa-bianchi. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math Soc.*, 1943.

[4] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2004.

[5] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability. In *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 136–143, 2005.

[6] M. Cuturi, K. Fukumizu, and J-P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005.

[7] André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Nonextensive information theoretic kernels on measures. *Journal of Machine Learning Research*, 10:935–975, 2009.

[8] A. Berlinet and Thomas C. Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers, 2004.

[9] A. Smola, A. Gretton, L. Song, and B. Schölkopf. A hilbert space embedding for distributions. In *Proceedings of the Eighteenth International Conf. on Algorithmic Learning Theory (ALT)*, pages 13–31. Springer-Verlag, 2007.

[10] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and Gert R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 99:1517–1561, 2010.

[11] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *COLT '01/EuroCOLT '01*, pages 416–426. Springer-Verlag, 2001.

[12] F. Dinuzzo and B. Schölkopf. The representer theorem for Hilbert spaces: a necessary and sufficient condition. In *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[13] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[14] A. Christmann and I. Steinwart. Universal kernels on non-standard input spaces. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 406–414. 2010.

[15] N. A. Mehta and A. G. Gray. Generative and latent mean map kernels. *CoRR*, abs/1005.0188, 2010.

[16] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

[17] H.S. Anderson and M.R. Gupta. Expected kernel for missing features in support vector machines. In *Statistical Signal Processing Workshop (SSP)*, pages 285–288, 2011.

[18] L. Fei-fei. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1150–1157, Washington, DC, USA, 1999.

[20] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 606–613, 2009.