Efficient Additive Kernels via Explicit Feature Maps

Andrea Vedaldi, *Member, IEEE,* Andrew Zisserman, *Member, IEEE*

Abstract—Large scale non-linear support vector machines (SVMs) can be approximated by linear ones using a suitable feature map. The linear SVMs are in general much faster to learn and evaluate (test) than the original non-linear SVMs. This work introduces explicit feature maps for the additive class of kernels, such as the intersection, Hellinger's, and χ^2 kernels, commonly used in computer vision, and enables their use in large scale problems. In particular, we: (i) provide explicit feature maps for all additive homogeneous kernels along with closed form expression for all common kernels; (ii) derive corresponding approximate finite-dimensional feature maps based on a spectral analysis; and (iii) quantify the error of the approximation, showing that the error is independent of the data dimension and decays exponentially fast with the approximation order for selected kernels such as χ^2. We demonstrate that the approximations have indistinguishable performance from the full kernels yet greatly reduce the train/test times of SVMs. We also compare with two other approximation methods: the Nystrom's approximation of Perronnin *et al.* [1] which is data dependent; and the explicit map of Maji and Berg [2] for the intersection kernel which, as in the case of our approximations, is data independent. The approximations are evaluated on on a number of standard datasets including Caltech-101 [3], Daimler-Chrysler pedestrians [4], and INRIA pedestrians [5].

Index Terms—Kernel methods, feature map, large scale learning, object recognition, object detection.

1 INTRODUCTION

Recent advances have made it possible to learn linear support vector machines (SVMs) in time linear with the number of training examples [6], extending the applicability of these learning methods to large scale datasets, on-line learning, and structural problems. Since a non-linear SVM can be seen as a linear SVM operating in an appropriate feature space, there is at least the theoretical possibility of extending these fast algorithms to a much more general class of models. The success of this idea requires that (i) the feature map used to project the data into the feature space can be computed efficiently and (ii) that the features are sufficiently compact (e.g., low dimensional or sparse).

Finding a feature map with such characteristics is in general very hard, but there are several cases of interest where this is possible. For instance, Maji and Berg [2] recently proposed a sparse feature map for the intersection kernel obtaining up to a 10^3 speedup in the learning of corresponding SVMs. In this paper, we introduce the *homogeneous kernel maps* to approximate all the additive homogeneous kernels, which, in addition to the intersection kernel, include the Hellinger's, χ^2, and Jensen-Shannon kernels. Furthermore, we combine these maps with Rahimi and Recht's randomized Fourier features [7] to approximate the Gaussian Radial Basis Function (RBF) variants of the additive kernels as well. Overall, these kernels include many of the most popular and best performing kernels used in computer vision applications [8]. In fact, they are particularly suitable for data in the form of finite probability distributions, or normalized histograms, and many computer vision descriptors, such as bag of visual words [9], [10] and spatial pyramids [11], [12], can be regarded as such.

Our aim is to obtain compact and simple representations that are efficient in both training and testing, have excellent performance, and have a satisfactory theoretical support.

Related work. Large scale SVM solvers are based on bundle methods [6], efficient Newton optimizers [13], or stochastic gradient descent [14], [15], [16], [17]. These methods require or benefit significantly from the use of linear kernels. Consider in fact the simple operation of evaluating an SVM. A linear SVM is given by the inner product $F(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ between a data vector $\mathbf{x} \in \mathbb{R}^D$ and a weight vector $\mathbf{w} \in \mathbb{R}^D$. A non linear SVM, on the other hand, is given by the expansion $F(\mathbf{x}) = \sum_{i=1}^N \beta_i K(\mathbf{x}, \mathbf{x}_i)$ where K is a non-linear kernel and $\mathbf{x}_1, \dots, \mathbf{x}_N$ are N "representative" data vectors found during training (support vectors). Since in most cases evaluating the inner product $\langle \mathbf{w}, \mathbf{x} \rangle$ is about as costly as evaluating the kernel $K(\mathbf{x}, \mathbf{x}_i)$, this makes the evaluation of the non-linear SVM N times slower than the linear one. The training cost is similarly affected.

A common method of accelerating the learning of non-linear SVMs is the computation of an explicit feature map. Formally, for any positive definite (PD) kernel $K(\mathbf{x}, \mathbf{y})$ there exists a function $\Psi(\mathbf{x})$ mapping the data $\mathbf{x}$ to a Hilbert space $\mathcal{H}$ such that $K(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathcal{H}}$ [18]. This Hilbert space is often known as a *feature space* and the function $\Psi(\mathbf{x})$ as a *feature map*. While conceptually useful, the feature map $\Psi(\mathbf{x})$ can

• *A. Vedaldi and A. Zisserman are with the Department of Engineering Science, Oxford University, Oxford, UK, OX1 3PJ.*
E-mail: {vedaldi,az}@robots.ox.ac.uk

rarely be used in computations as the feature space $\mathcal{H}$ is usually infinite dimensional. Still, it is often possible to construct a n-dimensional feature map $\hat{\Psi}(\mathbf{x}) \in \mathbb{R}^n$ that *approximates* the kernel sufficiently well. Nystrom's approximation [19] exploits the fact that the kernel is linear in feature space to project the data to a suitably selected subspace $\mathcal{H}_n \subset \mathcal{H}$, spanned by n vectors $\Psi(\mathbf{z}_1), \dots, \Psi(\mathbf{z}_n)$. Nystrom's feature map $\hat{\Psi}(\mathbf{x})$ is then expressed as $\hat{\Psi}(\mathbf{x}) = \Pi k(\mathbf{x}; \mathbf{z}_1, \dots, \mathbf{z}_n)$ where Π is a $n \times n$ matrix and $k_i(\mathbf{x}; \mathbf{z}_1, \dots, \mathbf{z}_n) = K(\mathbf{x}, \mathbf{z}_i)$ is the projection of $\mathbf{x}$ onto the basis element $\mathbf{z}_i$. The points $\mathbf{z}_1, \dots, \mathbf{z}_n$ are determined to span as much as possible of the data variability. Some methods select $\mathbf{z}_1, \dots, \mathbf{z}_n$ from the N training points $\mathbf{x}_1, \dots, \mathbf{x}_N$. The selection can be either random [20], or use a greedy optimization criterion [21], or use the incomplete Cholesky decomposition [22], [23], or even include side information to target discrimination [24]. Other methods [25], [26] synthesize $\mathbf{z}_1, \dots, \mathbf{z}_n$ instead of selecting them from the training data. In the context of additive kernels, Perronnin *et al.* [1] apply Nystrom's approximation to *each dimension of the data* $\mathbf{x}$ *independently*, greatly increasing the efficiency of the method.

An approximated feature map $\hat{\Psi}(\mathbf{x})$ can also be defined directly, independent of the training data. For instance, Maji and Berg [2] noted first that the intersection kernel can be approximated efficiently by a sparse closed-form feature map of this type. [7], [27] use random sampling in the Fourier domain to compute explicit maps for the translation invariant kernels. [28] specializes the Fourier domain sampling technique to certain multiplicative group-invariant kernels.

The spectral analysis of the homogeneous kernels is based on the work of Hein and Bousquet [29], and is related to the spectral construction of Rahimi and Recht [7].

Contributions and overview. This work proposes a unified analysis of a large family of additive kernels, known as γ-homogeneous kernels. Such kernels are seen as a logarithmic variants of the well known stationary (translation-invariant) kernels and can be characterized by a single scalar function, called a signature (Sect. 2). Homogeneous kernels include the intersection as well as the χ^2 and Hellinger's (Battacharyya's) kernels, and many others (Sect. 2.1). The signature is a powerful representation that enables: (i) the derivation of closed form feature maps based on 1D Fourier analysis (Sect. 3); (ii) the computation of finite, low dimensional, tight approximations (homogeneous kernel maps) of these feature maps for all common kernels (Sect. 4.2); and (iii) an analysis of the error of the approximation (Sect. 5), including asymptotic rates and determination of the optimal approximation parameters (Sect. 5.1). A surprising result is that there exist a uniform bound on the approximation error which is *independent of the data dimensionality and distribution*.

The method is contrasted to the one of Perronnin *et al.* [1], proving its optimality in the Nystrom's sense (Sect. 4.1), and to the one of Maji and Berg (MB) [2] (Sect. 6). The theoretical analysis is concluded by combining Rahimi and Recht [7] random Fourier features with the homogeneous kernel map to approximate the Gaussian RBF variants of the additive kernels (Sect. 7, [30]).

Sect. 8 compares empirically our approximations to the exact kernels, and shows that we obtain virtually the same performance despite using extremely compact feature maps and requiring a fraction of the training time. This representation is therefore complementary to the MB approximation of the intersection kernel, which relies instead on a high-dimensional but sparse expansion. In machine learning applications, the speed of the MB representation is similar to the homogeneous kernel map, but requires a modification of the solver to implement a non-standard regularizer. Our feature maps are also shown to work as well as Perronnin *et al.*'s approximations while not requiring any training.

Our method is tested on the DaimlerChrysler pedestrian dataset [4] (Sect. 8.2), the Caltech-101 dataset [3] (Sect. 8.1), and the INRIA pedestrians dataset [5] (Sect. 8.3), demonstrating significant speedups over the standard non-linear solvers without loss of accuracy.

Efficient code to compute the homogeneous kernel maps is available as part of the open source VLFeat library [31]. This code can be used to kernelise most linear algorithms with minimal or no changes to their implementation.

2 Homogeneous and stationary kernels

The main focus of this paper are additive kernels such as the Hellinger's, χ^2, intersection, and Jensen-Shannon ones. All these kernels are widley used in machine learning and computer vision applications. Beyond additivity, they share a second useful property: homogeneity. This section reviews homogeneity and the connected notion of stationarity and links them through the concept of kernel signature, a scalar function that fully characterizes the kernel. These properties will be used in Sect. 3 to derive feature maps and their approximations.

Homogeneous kernels. A kernel $k_{\mathrm{h}} : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}$ is γ-*homogeneous* if

$$\forall c \geq 0 : \quad k_{\mathrm{h}}(cx, cy) = c^\gamma k_{\mathrm{h}}(x, y). \tag{1}$$

When $\gamma = 1$ the kernel is simply said to be *homogeneous*. By choosing $c = 1/\sqrt{xy}$ a γ-homogeneous kernel can be written as

$$\begin{aligned} k_{\mathrm{h}}(x, y) = c^{-\gamma} k_{\mathrm{h}}(cx, cy) &= (xy)^{\frac{\gamma}{2}}\, k_{\mathrm{h}}\left(\sqrt{\frac{y}{x}}, \sqrt{\frac{x}{y}}\right) \\ &= (xy)^{\frac{\gamma}{2}}\, \mathcal{K}(\log y - \log x), \end{aligned} \tag{2}$$

where we call the scalar function

$$\mathcal{K}(\lambda) = k_{\mathrm{h}}\left(e^{\frac{\lambda}{2}}, e^{-\frac{\lambda}{2}}\right), \quad \lambda \in \mathbb{R} \tag{3}$$

kernel	$k(x,y)$	signature $\mathcal{K}(\lambda)$	$\kappa(\omega)$	feature $\Psi_\omega(x)$
Hellinger's	$\sqrt{xy}$	1	$\delta(\omega)$	$\sqrt{x}$
χ^2	$2\frac{xy}{x+y}$	$\operatorname{sech}(\lambda/2)$	$\operatorname{sech}(\pi\omega)$	$e^{i\omega \log x}\sqrt{x \operatorname{sech}(\pi\omega)}$
intersection	$\min\{x,y\}$	$e^{-\lvert\lambda\rvert/2}$	$\frac{2}{\pi}\frac{1}{1+4\omega^2}$	$e^{i\omega \log x}\sqrt{\frac{2x}{\pi}\frac{1}{1+4\omega^2}}$
JS	$\frac{x}{2}\log_2\frac{x+y}{x} + \frac{y}{2}\log_2\frac{x+y}{y}$	$\frac{e^{\frac{\lambda}{2}}}{2}\log_2\left(1+e^{-\lambda}\right) + \frac{e^{\frac{-\lambda}{2}}}{2}\log_2\left(1+e^{\lambda}\right)$	$\frac{2}{\log 4}\frac{\operatorname{sech}(\pi\omega)}{1+4\omega^2}$	$e^{i\omega \log x}\sqrt{\frac{2x}{\log 4}\frac{\operatorname{sech}(\pi\omega)}{1+4\omega^2}}$

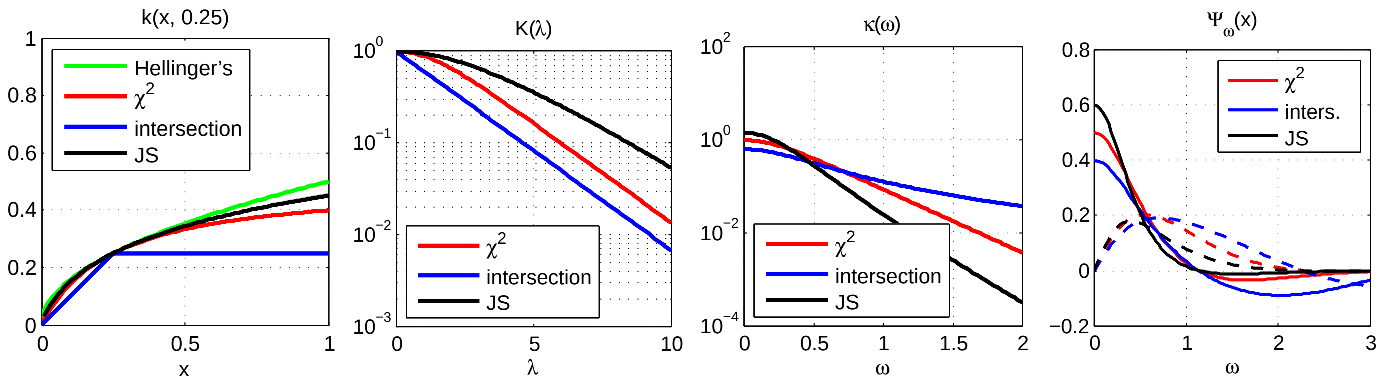


Fig. 1. **Common kernels, their signatures, and their closed-form feature maps.** *Top*: closed form expressions for the χ^2, intersection, Hellinger's (Battacharyya's), and Jensen-Shannon's (JS) homogeneous kernels, their signatures $\mathcal{K}(\lambda)$ (Sect. 2), the spectra $\kappa(\omega)$, and the feature maps $\Psi_\omega(x)$ (Sect. 3). *Bottom*: Plots of the various functions for the homogeneous kernels. The χ^2 and JS kernels are smoother than the intersection kernel (left panel). The smoothness corresponds to a faster fall-off of the the spectrum $\kappa(\omega)$, and ultimately to a better approximation by the homogeneous kernel map (Fig. 5 and Sect. 4.2).

the kernel *signature*.

Stationary kernels. A kernel $k_s : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is called *stationary* (or translation invariant, shift invariant) if

$$\forall c \in \mathbb{R}: \quad k_s(c+x, c+y) = k_s(x,y). \tag{4}$$

By choosing $c = -(x+y)/2$ a stationary kernel can be rewritten as

$$\begin{aligned} k_s(x,y) = k_s(c+x,c+y) &= k_s\left(\frac{y-x}{2}, \frac{x-y}{2}\right) \\ &= \mathcal{K}(y-x), \end{aligned} \tag{5}$$

where we also call the scalar function

$$\mathcal{K}(\lambda) = k_s\left(\frac{\lambda}{2}, -\frac{\lambda}{2}\right) \qquad \lambda \in \mathbb{R}. \tag{6}$$

the kernel signature.

Any scalar function that is PD is the signature of a kernel, and vice versa. This is captured by the following Lemma, proved in Sect. 10.2.

Lemma 1. *A γ-homogeneous kernel is PD if, and only if, its signature $\mathcal{K}(\lambda)$ is a PD function. The same is true for stationary a stationary kernel.*

Additive and multiplicative combinations. In applications one is interested in defining kernels $K(\mathbf{x},\mathbf{y})$ for multi-dimensional data $\mathbf{x},\mathbf{y} \in \mathcal{X}^D$. In this paper $\mathcal{X}$ may denote either the real line $\mathbb{R}$, for which $\mathcal{X}^D$ is the set of all D-dimensional real *vectors*, or the non-negative semiaxis $\mathbb{R}_0^+$, for which $\mathcal{X}^D$ is the set of all D-dimensional *histograms*.

So far homogeneous and stationary kernels $k(x,y)$ have been introduced for the scalar ($D=1$) case. Such kernels can be extended to the multi-dimensional case by either *additive or multiplicative combination*, which are given respectively by:

$$K(\mathbf{x},\mathbf{y}) = \sum_{l=1}^{D} k(\mathbf{x}_l,\mathbf{y}_l), \text{ or } K(\mathbf{x},\mathbf{y}) = \prod_{l=1}^{D} k(\mathbf{x}_l,\mathbf{y}_l). \tag{7}$$

Homogeneous kernels with negative components. Homogeneous kernels, which have been defined on the non-negative semiaxis $\mathbb{R}_0^+$, can be extended to the whole real line $\mathbb{R}$ by the following lemma:

Lemma 2. *Let $k(x,y)$ be a PD kernel on $\mathbb{R}_0^+$. Then the extensions $\operatorname{sign}(xy)k(|x|,|y|)$ and $\frac{1}{2}(\operatorname{sign}(xy)+1)k(|x|,|y|)$ are both PD kernels on $\mathbb{R}$.*

Metrics. It is often useful to consider the metric $D(\mathbf{x},\mathbf{y})$ that corresponds to a certain PD kernel $K(\mathbf{x},\mathbf{y})$. This is given by [32]

$$D^2(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{x}) + K(\mathbf{y},\mathbf{y}) - 2K(\mathbf{x},\mathbf{y}). \tag{8}$$

Similarly $d^2(x,y)$ will denote the squared metric obtained from a scalar kernel $k(x,y)$.

2.1 Examples

2.1.1 Homogeneous kernels

All common additive kernels used in computer vision, such as the intersection, χ^2, Hellinger's, and Jensen-

Shannon kernels, are additive combinations of homogeneous kernels. These kernels are reviewed next and summarized in Fig. 1. A wide family of homogeneous kernel is given in [29].

Hellinger's kernel. The Hellinger's kernel is given by $k(x,y) = \sqrt{xy}$ and is named after the corresponding additive squared metric $D^2(\mathbf{x},\mathbf{y}) = \|\sqrt{\mathbf{x}} - \sqrt{\mathbf{y}}\|_2^2$ which is the squared Hellinger's distance between histograms $\mathbf{x}$ and $\mathbf{y}$. This kernel is also known as Bhattacharyya's coefficient. Its signature is the constant function $\mathcal{K}(\lambda) = 1$.

Intersection kernel. The intersection kernel is given by $k(x,y) = \min\{x,y\}$ [33]. The corresponding additive squared metric $D^2(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_1$ is the l^1 distance between the histograms $\mathbf{x}$ and $\mathbf{y}$. Note also that, by extending the intersection kernel to the negative reals by the second extension in Lemma 2, the corresponding squared metric $D^2(\mathbf{x},\mathbf{y})$ becomes the l^1 distance between *vectors* $\mathbf{x}$ and $\mathbf{y}$.

χ^2 **kernel.** The χ^2 kernel [34], [35] is given by $k(x,y) = 2(xy)/(x+y)$ and is named after the corresponding additive squared metric $D^2(\mathbf{x},\mathbf{y}) = \chi^2(\mathbf{x},\mathbf{y})$ which is the χ^2 distance. Notice that some authors define the χ^2 kernel as the negative of the χ^2 distance, i.e. $K(\mathbf{x},\mathbf{y}) = -\chi^2(\mathbf{x},\mathbf{y})$. Such a kernel is only *conditionally* PD [18], while the definition used here makes it PD. If the histograms $\mathbf{x},\mathbf{y}$ are l^1 normalised, the two definitions differ by a constant offset.

Jensen-Shannon (JS) kernel. The JS kernel is given by $k(x,y) = (x/2)\log_2(x+y)/x + (y/2)\log_2(x+y)/y$. It is named after the corresponding squared metric $D^2(\mathbf{x},\mathbf{y})$ which is the Jensen-Shannon divergence (or symmetrized Kullback-Leibler divergence) between the histograms $\mathbf{x}$ and $\mathbf{y}$. Recall that, if $\mathbf{x},\mathbf{y}$ are finite probability distributions (l^1-normalized histograms), the Kullback-Leibler divergence is given by $KL(\mathbf{x}|\mathbf{y}) = \sum_{l=1}^d \mathbf{x}_l \log_2(\mathbf{x}_l/\mathbf{y}_l)$ and the Jensen-Shannon divergence is given by $D^2(\mathbf{x},\mathbf{y}) = KL(\mathbf{x}|(\mathbf{x}+\mathbf{y})/2)) + KL(\mathbf{y}|(\mathbf{x}+\mathbf{y})/2))$. The use of the base two logarithm in the definition normalizes the kernel: If $\|\mathbf{x}\|_1 = 1$, then $K(\mathbf{x},\mathbf{x}) = 1$.

γ**-homogeneous variants.** The kernels introduced so far are 1-homogeneous. An important example of 2-homogeneous kernel is the linear kernel $k(x,y) = xy$. It is possible to obtain a γ-homogeneous variant of any 1-homogeneous kernel simply by plugging the corresponding signature into (2). Some practical advantages of using $\gamma \neq 1,2$ are discussed in Sect. 8.

2.1.2 Stationary kernels

A well known example of stationary kernel is the *Gaussian kernel* $k_{\mathrm{s}}(x,y) = \exp(-(y-x)^2/(2\sigma^2))$. In light of Lemma 1, one can compute the signature of the Gaussian kernel from (6) and substitute it into (3) to get a corresponding homogeneous kernel $k_{\mathrm{h}}(x,y) = \sqrt{xy}\exp(-(\log y/x)^2/(2\sigma^2))$. Similarly, transforming the homogeneous χ^2 kernel into a stationary kernel yields $k_{\mathrm{s}}(x,y) = \operatorname{sech}((y-x)/2)$.

2.2 Normalization

Empirically, it has been observed that properly normalising a kernel $K(\mathbf{x},\mathbf{y})$ may boost the recognition performance (e.g. [36], additional empirical evidence is reported in Sect. 8). A way to do so is to scale the data $\mathbf{x} \in \mathcal{X}^D$ so that $K(\mathbf{x},\mathbf{x}) = $ const. for all $\mathbf{x}$. In this case one can show that, due to positive definiteness, $K(\mathbf{x},\mathbf{x}) \geq |K(\mathbf{x},\mathbf{y})|$. This encodes a simple consistency criterion: by interpreting $K(\mathbf{x},\mathbf{y})$ as a similarity score, $\mathbf{x}$ should be the point most similar to itself [36].

For stationary kernels, $k_{\mathrm{s}}(x,x) = \mathcal{K}(x-x) = \mathcal{K}(0)$ shows that they are automatically normalized. For γ-homogeneous kernels, (2) yields $k_{\mathrm{h}}(x,x) = (xx)^{\frac{\gamma}{2}}\mathcal{K}(\log(x/x)) = x^\gamma\mathcal{K}(0)$, so that for the corresponding additive kernel (7) one has $K(\mathbf{x},\mathbf{x}) = \sum_{l=1}^D k_{\mathrm{h}}(\mathbf{x}_l,\mathbf{x}_l) = \|\mathbf{x}\|_\gamma^\gamma\mathcal{K}(0)$ where $\|\mathbf{x}\|_\gamma$ denotes the l^γ norm of the histogram $\mathbf{x}$. Hence the normalisation condition $K(\mathbf{x},\mathbf{x}) = $ const. can be enforced by scaling the histograms $\mathbf{x}$ to be l^γ normalised. For instance, for the χ^2 and intersection kernels, which are homogeneous, the histograms should be l^1 normalised, whereas for the linear kernel, which is 2-homogeneous, the histograms should be l^2 normalised.

3 Analytic forms for feature maps

A *feature map* $\Psi(\mathbf{x})$ for a kernel $K(\mathbf{x},\mathbf{y})$ is a function mapping $\mathbf{x}$ into a Hilbert space with inner product $\langle\cdot,\cdot\rangle$ such that

$$\forall \mathbf{x},\mathbf{y} \in \mathcal{X}^D : K(\mathbf{x},\mathbf{y}) = \langle\Psi(\mathbf{x}),\Psi(\mathbf{y})\rangle.$$

It is well known that Bochner's theorem [7] can be used to compute feature maps for stationary kernels. We use Lemma 1 to extend this construction to the γ-homogeneous case and obtain closed form feature maps for all commonly used homogeneous kernels.

Bochner's theorem. For any PD function $\mathcal{K}(\boldsymbol{\lambda})$, $\boldsymbol{\lambda} \in \mathbb{R}^D$ there exists a non-negative symmetric measure $d\mu(\boldsymbol{\omega})$ such that $\mathcal{K}$ is its Fourier transform, i.e.:

$$\mathcal{K}(\boldsymbol{\lambda}) = \int_{\mathbb{R}^D} e^{-i\langle\boldsymbol{\omega},\boldsymbol{\lambda}\rangle}\, d\mu(\boldsymbol{\omega}). \tag{9}$$

For simplicity it will be assumed that the measure $d\mu(\mathbf{x})$ can be represented by a density $\kappa(\boldsymbol{\omega})\,d\boldsymbol{\omega} = d\mu(\boldsymbol{\omega})$ (this covers most cases of interest here). The function $\kappa(\boldsymbol{\omega})$ is called the *spectrum* and can be computed as the inverse Fourier transform of the signature $\mathcal{K}(\boldsymbol{\lambda})$:

$$\kappa(\boldsymbol{\omega}) = \frac{1}{(2\pi)^D}\int_{\mathbb{R}^D} e^{i\langle\boldsymbol{\omega},\boldsymbol{\lambda}\rangle}\mathcal{K}(\boldsymbol{\lambda})\,d\boldsymbol{\lambda}. \tag{10}$$

Stationary kernels. For a stationary kernel $k_{\mathrm{s}}(x,y)$ defined on $\mathbb{R}$, starting from (5) and by using Bochner's theorem (9), one obtains:

$$\begin{aligned} k_{\mathrm{s}}(x,y) &= \int_{-\infty}^{+\infty} e^{-i\omega\lambda}\kappa(\omega)\,d\omega, \quad \lambda = y-x, \\ &= \int_{-\infty}^{+\infty}\left(e^{-i\omega x}\sqrt{\kappa(\omega)}\right)^*\left(e^{-i\omega y}\sqrt{\kappa(\omega)}\right)d\omega. \end{aligned}$$

Define the function of the scalar variable $\omega \in \mathbb{R}$

$$\boxed{\Psi_\omega(x) = e^{-i\omega x}\sqrt{\kappa(\omega)}}. \tag{11}$$

Here ω can be thought as of the index of an infinite-dimensional vector (in Sect. 4 it will be discretized to obtain a finite dimensional representation). The function $\Psi(x)$ is a feature map because

$$k_s(x,y) = \int_{-\infty}^{+\infty} \Psi_\omega(x)^* \Psi_\omega(y)\, d\omega = \langle \Psi(x), \Psi(y)\rangle.$$

γ-homogeneous kernels. For a γ-homogenoeus kernel $k(x,y)$ the derivation is similar. Starting from (2) and by using Lemma 1 and Bochner's theorem (9):

$$k(x,y) = (xy)^{\frac{\gamma}{2}} \int_{-\infty}^{+\infty} e^{-i\omega\lambda}\kappa(\omega)\, d\omega, \quad \lambda = \log\frac{y}{x},$$
$$= \int_{-\infty}^{+\infty} \left(e^{-i\omega\log x}\sqrt{x^\gamma \kappa(\omega)}\right)^* \left(e^{-i\omega\log y}\sqrt{y^\gamma\kappa(\omega)}\right) d\omega.$$

The same result can be obtained as a special case of Corollary 3.1 of [29]. This yields the feature map

$$\boxed{\Psi_\omega(x) = e^{-i\omega\log x}\sqrt{x^\gamma\kappa(\omega)}}. \tag{12}$$

Closed form feature maps. For the common computer vision kernels, the density $\kappa(\omega)$, and hence the feature map $\Psi_\omega(x)$, can be computed in closed form. Fig. 1 lists the expressions.

Multi-dimensional case. Given a multi-dimensional kernel $K(\mathbf{x},\mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(y)\rangle$, $\mathbf{x} \in \mathcal{X}^D$ which is a additive/multiplicative combination of scalar kernels $k(x,y) = \langle\Psi(x),\Psi(y)\rangle$, the multi-dimensional feature map $\Psi(\mathbf{x})$ can be obtained from the scalar ones as

$$\Psi(\mathbf{x}) = \bigoplus_{l=1}^{D} \Psi(\mathbf{x}_l), \quad \Psi(\mathbf{x}) = \bigotimes_{l=1}^{D} \Psi(\mathbf{x}_l) \tag{13}$$

respectively for additive and multiplicative cases. Here $\oplus$ is the direct sum (stacking) of the vectors $\Psi(\mathbf{x}_l)$ and $\otimes$ their Kronecker product. If the dimensionality of $\Psi(x)$ is n, then the additive feature map has dimensionality nD and the multiplicative one n^D.

4 APPROXIMATED FEATURE MAPS

The features introduced in Sect. 3 cannot be used directly for computations as they are continuous functions; in applications low dimensional or sparse approximations are needed. Sect. 4.1 reviews Nyström's approximation, the most popular way of deriving low dimensional feature maps.

The main disadvantages of Nyström's approximation is that it is data-dependent and requires training. Sect. 4.2 illustrates an alternative construction based on the exact feature maps of Sect. 3. This results in data-independent approximations, that are therefore called *universal*. These are simple to compute, very compact, and usually available in closed form. These feature maps are also shown to be optimal in Nyström's sense.

4.1 Nyström's approximation

Givan a PD kernel $K(\mathbf{x},\mathbf{y})$ and a data density $p(\mathbf{x})$, the Nyström approximation of order n is the feature map $\bar\Psi : \mathcal{X}^D \to \mathbb{R}^n$ that best approximates the kernel at points $\mathbf{x}$ and $\mathbf{y}$ sampled from $p(\mathbf{x})$ [19]. Specifically, $\bar\Psi$ is the minimizer of the functional

$$E(\bar\Psi) = \int_{\mathcal{X}^D\times\mathcal{X}^D} \left(K(\mathbf{x},\mathbf{y}) - \langle\bar\Psi(\mathbf{x}),\bar\Psi(\mathbf{y})\rangle\right)^2 p(\mathbf{x})\,p(\mathbf{y})\,d\mathbf{x}\,d\mathbf{y}. \tag{14}$$

For all approximation orders n the components $\bar\Psi_i(\mathbf{x})$, $i = 0,1,\dots$ are eigenfunctions of the kernel. Specifically $\bar\Psi_i(\mathbf{x}) = \bar\kappa_i\bar\Phi_i(\mathbf{x})$ where

$$\int_{\mathcal{X}^D} K(\mathbf{x},\mathbf{y})\bar\Phi_i(\mathbf{y})\,p(\mathbf{y})\,d\mathbf{y} = \bar\kappa_i^2\bar\Phi_i(\mathbf{x}), \tag{15}$$

$\bar\kappa_0^2 \geq \bar\kappa_1^2 \geq \dots$ are the (real and non-negative) eigenvalues in decreasing order, and the eigenfunctions $\bar\Phi_i(\mathbf{x})$ have unitary norm $\int\bar\Phi_i(\mathbf{x})^2 p(\mathbf{x})d\mathbf{x} = 1$. Usually the data density $p(\mathbf{x})$ is approximated by a finite sample set $\mathbf{x}_1,\dots,\mathbf{x}_N \sim p(\mathbf{x})$ and one recovers from (15) the kernel PCA problem [18], [19]:

$$\frac{1}{N}\sum_{j=1}^{N} K(\mathbf{x},\mathbf{x}_j)\bar\Phi_i(\mathbf{x}_j) = \bar\kappa_i^2\bar\Phi_i(\mathbf{x}). \tag{16}$$

Operationally, (16) is sampled at points $\mathbf{x} = \mathbf{x}_1,\dots,\mathbf{x}_N$ forming a $N\times N$ eigensystem. The solution yields N unitary[1] eigenvectors $[\bar\Phi_i(\mathbf{x}_j)]_{j=1,\dots,N}$ and corresponding eigenvalues $\bar\kappa_i$. Given these, (16) can be used to compute $\bar\Psi_i(\mathbf{x})$ for any $\mathbf{x}$.

4.2 Homogeneous kernel map

This section derives the homogeneous kernel map, a universal (data-independent) approximated feature map. The next Sect. 5 derives uniform bounds for the approximation error and shows that the homogeneous kernel map converges exponentially fast to smooth kernels such as χ^2 and JS. Finally, Sect. 5.1 shows how to select automatically the parameters of the approximation for any approximation order.

Most finite dimensional approximations of a kernel start by limiting the span of the data. For instance, Mercer's theorem [37] assumes that the data spans a compact domain, while Nyström's approximation (15) assumes that the data is distributed according to a density $p(\mathbf{x})$. When the kernel domain is thus restricted, its spectrum becomes discrete and can be used to derive a finite approximation.

The same effect can be achieved by making the kernel periodic rather than by limiting its domain. Specifically, given the signature $\mathcal{K}$ of a stationary or homogeneous kernel, consider its periodicization $\hat{\mathcal{K}}$ obtained by making copies of it with period Λ:

$$\hat{\mathcal{K}}(\lambda) = \operatorname*{per}_{\Lambda} W(\lambda)\mathcal{K}(\lambda) = \sum_{k=\infty}^{+\infty} W(\lambda + k\Lambda)\mathcal{K}(\lambda + k\Lambda)$$

1. I.e. $\sum_j \Phi_i(\mathbf{x}_j)^2 = 1$. Since one wants $\int \Phi_i(\mathbf{x})^2\,p(\mathbf{x})\,d\mathbf{x} \approx \sum_j \Phi_i(\mathbf{x}_j)^2/N = 1$ the eigenvectors need to be rescaled by $\sqrt{N}$.

Here $W(\lambda)$ is an optional PD windowing function that will be used to fine-tune the approximation.

The Fourier basis of the Λ-perdiodic functions are the harmonics $\exp(-ijL\lambda)$, $j = 0, 1, \dots,$ where $L = 2\pi/\Lambda$. This yields a discrete version of Bochner's result (9):

$$\hat{\mathcal{K}}(\lambda) = \sum_{j=-\infty}^{+\infty} \hat{\kappa}_j e^{-ijL\lambda} \tag{17}$$

where the discrete spectrum $\hat{\kappa}_j$, $j \in \mathbb{Z}$ can be computed from the continuous one $\kappa(\omega)$, $\omega \in \mathbb{R}$ as $\hat{\kappa}_j = L(w * \kappa)(jL)$, where $*$ denotes convolution and w is the inverse Fourier transform of the window W. In particular, if $W(\lambda) = 1$, then the discrete spectrum $\hat{\kappa}_j = L\kappa(jL)$ is obtained by *sampling* and rescaling the continuous one.

As (9) was used to derive feature maps for the homogeneous/stationary kernels, so (17) can be used to derive a discrete feature map for their periodic versions. We give a form that is convenient in applications: for stationary kernels

$$\boxed{\hat{\Psi}_j(x) = \begin{cases} \sqrt{\hat{\kappa}_0}, & j = 0, \\ \sqrt{2\hat{\kappa}_{\frac{j+1}{2}}} \cos\left(\frac{j+1}{2} Lx\right) & j > 0 \text{ odd}, \\ \sqrt{2\hat{\kappa}_{\frac{j}{2}}} \sin\left(\frac{j}{2} Lx\right) & j > 0 \text{ even}, \end{cases}} \tag{18}$$

and for γ-homogeneous kernels

$$\boxed{\hat{\Psi}_j(x) = \begin{cases} \sqrt{x^\gamma \hat{\kappa}_0}, & j = 0, \\ \sqrt{2x^\gamma \hat{\kappa}_{\frac{j+1}{2}}} \cos\left(\frac{j+1}{2} L \log x\right) & j > 0 \text{ odd}, \\ \sqrt{2x^\gamma \hat{\kappa}_{\frac{j}{2}}} \sin\left(\frac{j}{2} L \log x\right) & j > 0 \text{ even}, \end{cases}} \tag{19}$$

To verify that definition (11) is well posed observe that

$$\begin{aligned} \langle \hat{\Psi}(x), \hat{\Psi}(y) \rangle &= \hat{\kappa}_0 + 2 \sum_{j=1}^{+\infty} \hat{\kappa}_j \cos\left(jL(y - x)\right) \\ &= \sum_{j=-\infty}^{\infty} \hat{\kappa}_j e^{-ijL(y-x)} = \hat{\mathcal{K}}(y - x) = \hat{k}_{\mathrm{s}}(x, y) \end{aligned} \tag{20}$$

where $\hat{k}_{\mathrm{s}}(x, y)$ is the stationary kernel obtained from the periodicized signature $\hat{\mathcal{K}}(\lambda)$. Similarly, (12) yields the γ-homogeneous kernel $\hat{k}_{\mathrm{h}}(x, y) = (xy)^{\gamma/2} \hat{\mathcal{K}}(\log y - \log x)$.

While discrete, $\hat{\Psi}$ is still an infinite dimensional vector. However, since the spectrum $\hat{\kappa}_j$ decays fast in most cases, a sufficiently good approximation can be obtained by truncating $\hat{\Psi}$ to the first n components. If n is odd, then the truncation can be expressed by multiplying the discrete spectrum $\hat{\kappa}_j$ by a window t_j, $j \in \mathbb{Z}$ equal to one for $|i| \leq (n - 1)/2$ and to zero otherwise. Overall, the signature of the approximated $\hat{\mathcal{K}}$ kernel is obtained from the original signature $\mathcal{K}$ by windowing by W, periodicization, and smoothing by a kernel that depends on the truncation of the spectrum. In symbols:

$$\hat{\mathcal{K}}(\lambda) = \frac{1}{\Lambda} \left(T * \operatorname*{per}_{\Lambda} W\mathcal{K} \right)(\lambda), \quad \hat{\kappa}_j = t_j L(w * \kappa)(jL). \tag{21}$$

where $T(\lambda) = \sin(nL\lambda/2)/\sin(L\lambda/2)$ is the periodic sinc, obtained as the inverse Fourier transform of t_i.

Nyström's approximation viewpoint. Although Nström's approximation is data dependent, the universal feature maps (18) and (19) can be derived as Nyström's approximations provided that the data is sampled uniformly in one period and the periodicized kernel is considered. In particular, for the stationary case let $p(x)$ in (14) be uniform in *one period* (and zero elsewhere) and plug-in a stationary and *periodic* kernel $k_{\mathrm{s}}(x, y)$. One obtains the Nyström's objective functional

$$E(\bar{\Psi}) = \frac{1}{\Lambda^2} \int_{[-\Lambda/2, \Lambda/2]^2} \epsilon_{\mathrm{s}}(x, y)^2 dx\, dy. \tag{22}$$

where

$$\epsilon_{\mathrm{s}}(x, y) = k_{\mathrm{s}}(x, y) - \langle \bar{\Psi}(x), \bar{\Psi}(y) \rangle \tag{23}$$

is the point-wise approximation error. This specializes the eigensystem (15) to

$$\frac{1}{\Lambda} \int_{-\Lambda/2}^{\Lambda/2} \mathcal{K}(y - x) \bar{\Phi}_i(y)\, dy = \bar{\kappa}_i^2 \bar{\Phi}_i(x).$$

where $\mathcal{K}$ is the signature of k_{s}. Since the function $\mathcal{K}$ has period Λ, the eigenfunctions are harmonics of the type $\bar{\Phi}(x_i) \propto \cos(iL\omega + \psi)$. It is easy to see that one recovers the same feature map (18)[2]. Conversely, the γ-homogeneous feature map (19) can be recovered as the minimizer of

$$E(\bar{\Psi}) = \frac{1}{\Lambda^2} \int_{[-\Lambda/2, \Lambda/2]^2} \epsilon_{\mathrm{h}}(e^x, e^y)^2 dx\, dy.$$

where the *normalized approximation error* $\epsilon(x, y)$ is defined as

$$\epsilon_{\mathrm{h}}(x, y) = \frac{k_{\mathrm{h}}(x, y) - \langle \bar{\Psi}(x), \bar{\Psi}(y) \rangle}{(xy)^{\frac{\gamma}{2}}}. \tag{24}$$

5 Approximation error analysis

This section analyzes the quality of the approximated γ-homogeneous feature map (19). For convenience, details and proofs are moved to Sect. 10.1 and 10.2. A qualitative comparison between the different methods is included in Fig. 2. The following uniform error bound holds:

Lemma 3. *Consider an additive γ-homogeneous kernel $K(\mathbf{x}, \mathbf{y})$ and assume that the D-dimensional histograms $\mathbf{x}, \mathbf{y} \in \mathcal{X}^D$ are γ-normalized. Let $\hat{\Psi}(\mathbf{x})$ be the approximated feature map (19) obtained by choosing the uniform window $W(\lambda) = 1$ and a period Λ in (21). Then uniformly for all histograms $\mathbf{x}$,$\mathbf{y}$*

$$\boxed{|K(\mathbf{x}, \mathbf{y}) - \langle \hat{\Psi}(\mathbf{x}), \hat{\Psi}(\mathbf{y}) \rangle| \leq 2 \max\left\{\epsilon_0, \epsilon_1 e^{\frac{-\Lambda\gamma}{4}}\right\}} \tag{25}$$

where $\epsilon_0 = \sup_{|\lambda| \leq \Lambda/2} |\epsilon(\lambda)|$, $\epsilon_1 = \sup_\lambda |\epsilon(\lambda)|$, and $\epsilon(\lambda) = \mathcal{K}(\lambda) - \hat{\mathcal{K}}(\lambda)$ is the difference between the approximated and exact kernel signatures.

2. Up to a reordering of the components if the spectrum does not decay monotonically.

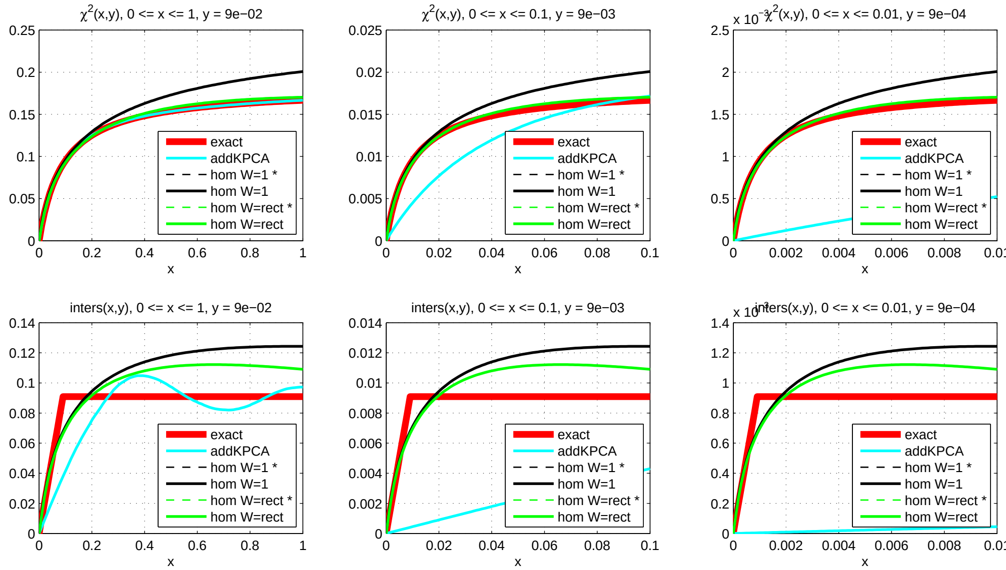


Fig. 2. **Universal approximations.** A useful property of the approximations (18) and (19) is universality. The figure compares approximating the χ^2 and intersection kernels $k_{\mathsf{h}}(x,y)$ using features with $n = 3$ components. From left to right the dynamic range B of the data is set to $B = 1, 0.1, 0.01$. The addKPCA approximation [1] is learned by sampling 100 points uniformly at random from the interval $[0,1]$, which tunes it to the case $B = 1$. The homogeneous maps are scale invariant, so no tuning (nor training) is required. Each panel plots $k_h(x,y)$ and the approximations for $x \in [0, B]$ and $y = B/10$. The figure also compares using a flat or rectangular window in the computation of (19). As a verification, the homogeneous maps (19) are also derived as Nystrom's approximations as explained in Sect. 3. Two things are noteworthy: (i) addKPCA works only for the dynamic range for which it is tuned, while the homogeneous maps yield exactly the same approximation for all the dynamic ranges; (ii) the rectangular window works better than the flat one in the computation of (19).

Note that the bound is *uniform* (i.e. valid simultaneously for all normalized histograms) and *dimension independent* (as ϵ_0 and ϵ_1 do not depend on D).

The quality of the approximation is then controlled by ϵ_0 and $\epsilon_1 e^{-\Lambda\gamma/4}$. It is shown in Sect. 10.1 that ϵ_0 decays either exponentially or polynomially fast, depending on the smoothness of the kernel being approximated, and ϵ_1 stays bounded. By setting Λ as a function of n appropriately, it is also shown that the overall approximation error (25) decays as $O(\exp(-\sqrt{\min\left\{\frac{\alpha}{2},\frac{\gamma}{4}\right\}\pi(n+1)\beta}))$ for the smoother χ^2 and JS kernels, and as $O((n/\log n)^{1-\beta})$ for the intersection kernel ($\alpha > 0$, $\beta > 1$ are constants). Since the former is an exponential rate and the second is only polynomial, the intersection kernel is harder to approximate.

Rectangular window. Consider now using a rectangular $W(\lambda) = \mathrm{rect}(\lambda/\Lambda)$. The periodicization does not introduce any error: $\mathrm{per}_\Lambda W\mathcal{K}(\lambda) = \mathcal{K}(\lambda)$, for all $|\lambda| \leq \Lambda/2$. Unfortunately, this window is not a PD, which may cause some of the samples of the spectrum (21) to be negative. This problem can be solved by truncating the values to the non-negative reals, which amounts to project the kernel back to the space of PD functions. In applications the rectangular and uniform window perform as well, but the rectangular window tends to result in a tighter reconstruction of the exact kernel (Fig. 2).

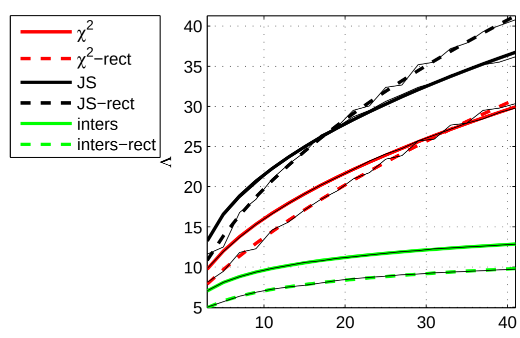


Fig. 3. **Optimal approximation parameters.** Optimal parameter $\Lambda^*(n)$ for the homogeneous kernel maps (19) approximating χ^2, JS, and intersection kernels. A slightly different tuning is needed depending on the type of window used in the computation of (19). The thin black lines are the empirical values of Λ determined by grid search on a validation set and are very well predicted the theory (Sect. 5.1).

5.1 Low dimensional regime

As detailed in Sect. 10.1, the error (25) is minimized if the period Λ is selected as $\Lambda^*(n) = a\sqrt{n+b} + c$ for the χ^2 and JS kernels and as $\Lambda^*(n) = a\log(n+b) + c$ for the intersection kernel, where $a, b, c \in \mathbb{R}$ are parameters and n is the order of the approximation. To obtain a good choice of Λ for the low dimensional regime, we determined the parameters empirically by minimizing the average approximation error of each kernel as a function of Λ on validation data (including bag of visual words and HOG histograms). Fig. 3 shows the fits for the different cases. These fits are quite tight, validating the theoretical analysis. These curves allow selecting automatically the sampling period Λ, which is the only free parameter in the approximation, and are used for all the experiments.

6 Comparison with other feature maps

MB encoding. Maji and Berg [2] propose for the intersection kernel $k(x,y) = \min\{x,y\}$ the infinite dimensional feature map $\Psi(x)$ given by $\Psi_\omega(x) = H(x-\omega), \omega \geq 0$, where $H(\omega)$ denotes the Heaviside function. In fact $\min\{x,y\} = \int_0^{+\infty} H(x-\omega)H(y-\omega)\,d\omega$. They also propose a n-dimensional approximation $\tilde{\Psi}(x)_j$ of the type $(1,\dots,1,a,0,\dots,0)/\sqrt{n}$ (where $0 \leq a < 1$), approximating the Heaviside function by taking its average in n equally spaced intervals. This feature map corresponds to the approximated intersection kernel

$$\tilde{k}(x,y) = \begin{cases} \min\{x,y\}, & \lfloor x \rfloor_n \neq \lfloor y \rfloor_n, \\ \lfloor x \rfloor_n + (x - \lfloor x \rfloor_n)(y - \lfloor y \rfloor_n)/n & \lfloor x \rfloor_n = \lfloor y \rfloor_n. \end{cases}$$

where $\lfloor x \rfloor_n = \text{floor}(nx)/n$. We refer to this approximation as MB, from the initials of the authors.

Differently from the homogeneous kernel map and the additive kernel PCA map [1], the MB approximation is designed to work in a relatively high dimensional regime ($n \gg 3$). In practice, the representation can be used as efficiently as a low-dimensional one by encoding the features sparsely. Let $D \in \mathbb{R}^{n \times n}$ be the difference operator

$$D_{ij} = \begin{cases} 1, & i = j, \\ -1, & j = i+1, \\ 0, & \text{otherwise.} \end{cases}$$

Then $D\tilde{\Psi}(x) = (0,\dots,1-a,a,0,\dots,0)$ has only two non-zero elements regardless of the dimensionality n of the expansion. Most learning algorithms (e.g. PEGASOS and cutting plane for the training of SVMs) take advantage of sparse representations, so that using either high-dimensional but sparse or low-dimensional features is similar in term of efficiency. The disadvantage is that the effect of the linear mapping D_{ij} must be accounted for in the algorithm. For instance, in learning the parameter vector $\mathbf{w}$ of an SVM, the regularizer $\mathbf{w}^\top\mathbf{w}$ must be replaced with $\mathbf{w}^\top H \mathbf{w}$, where $H = DD^\top$ [2].

Random Fourier features. Rahimi and Recht [7] propose an encoding of stationary kernels based on randomly sampling the spectrum. Let $K(\mathbf{x},\mathbf{y}) = \mathcal{K}(\mathbf{y}-\mathbf{x})$ be a D-dimensional stationary kernel and assume, without loss of generality, that $K(\mathbf{0}) = 1$. From Bochner's theorem (9) one has

$$K(\mathbf{x},\mathbf{y}) = \int_{\mathbb{R}^D} e^{-i\langle \boldsymbol{\omega}, \mathbf{y}-\mathbf{x}\rangle} \kappa(\boldsymbol{\omega}) d\boldsymbol{\omega} = E\left[e^{-i\langle \boldsymbol{\omega}, \mathbf{y}-\mathbf{x}\rangle}\right] \quad (26)$$

where the expected value is taken w.r.t. the probability density $\kappa(\boldsymbol{\omega})$ (notice in fact that $\kappa \geq 0$ and that $\int_{\mathbb{R}^D} \kappa(\boldsymbol{\omega})\,d\boldsymbol{\omega} = \mathcal{K}(0) = 1$ by assumption). The expected value can be approximated as $K(\mathbf{x},\mathbf{y}) \approx \frac{1}{n}\sum_{i=1}^n e^{-i\langle \boldsymbol{\omega}_i, \mathbf{y}-\mathbf{x}\rangle} = \langle \hat{\Psi}(\mathbf{x}), \hat{\Psi}(\mathbf{y})\rangle$ where $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n$ are sampled from the density $\kappa(\boldsymbol{\omega})$ and where

$$\hat{\Psi}(\mathbf{x}) = \frac{1}{\sqrt{n}} \left[e^{-i\langle \boldsymbol{\omega}_1, \mathbf{x}\rangle} \dots e^{-i\langle \boldsymbol{\omega}_n, \mathbf{x}\rangle}\right]^\top. \quad (27)$$

is the *random Fourier feature map*.

Li *et al.* [28] extend the random Fourier features to any group-invariant kernel. These kernels include the additive/multiplicative combinations of the skewed-χ^2 kernel, a generalization of the 0-homogeneous χ^2 kernel. An interesting aspect of this method is that all the dimensions of a D-dimensional kernel are approximated simultaneously (by sampling from a D-dimensional spectrum). Unfortunately, γ-homogeneous kernels for $\gamma > 0$ are *not* group-invariant; the random features can still be used by removing first the homogeneous factor $(xy)^{\gamma/2}$ as we do, but this is possible only for a component-wise approximation.

Random sampling does not appear to be competitive with the homogeneous kernel map for the component-wise approximation of additive kernels. The reason is that reducing the variance of the random estimate requires drawing a relatively large number of samples. For instance, we verified numerically that, on average, about 120 samples are needed to approximate the χ^2 kernel as accurately as the homogeneous kernel map using only three samples.

Nevertheless, random sampling has the crucial advantage of scaling to the approximation of D-dimensional kernels that do not decompose component-wise, such as the multiplicative combinations of the group-invariant kernels [28], or the Gaussian RBF kernels [7]. Sect. 7 uses the random Fourier features to approximate the generalized Gaussian RBF kernels.

Additive KPCA. While Nyström's method can be used to approximate directly a kernel $K(\mathbf{x},\mathbf{y})$ for multi-dimensional data (Sect. 4.1), Perronnin *et al.* [1] noticed that, for additive kernels $K(\mathbf{x},\mathbf{y}) = \sum_{l=1}^D k(\mathbf{x}_l, \mathbf{y}_l)$, it is preferable to apply the method to *each of the D dimensions* independently. This has two advantages: (i) the resulting approximation problems have very small dimensionality and (ii) each of the D feature maps $\bar{\Psi}^{(l)}(\mathbf{x}_l)$ obtained in this way is a function of a *scalar* argument $\mathbf{x}_l$ and can be easily tabulated. Since addKPCA optimizes the representation for a particular data distribution, it can result in a smaller reconstruction error of the kernel than the homogeneous kernel map (although, as shown in Sect. 8, this does not appear to result in better classification). The main disadvantage is that the representation must be learned from data, which adds to the computational cost and is problematic in applications such as on-line learning.

7 Approximation of RBF kernels

Any algorithm that accesses the data only through inner products (*e.g.*, a linear SVM) can be extended to operate in non-linear spaces by replacing the inner product with an appropriate kernel. Authors refer to this idea as the *kernel trick* [18]. Since for each kernel one has an associated metric (8), the kernel trick can be applied to any algorithm that is based on distances [32] as well (*e.g.*, nearest-neighbors). Similarly, the approximated feature maps that have been introduced in order to speed-up kernel-based algorithms can be used for distance-based algorithms too. Specifically, plugging the approximation $\mathcal{K}(\mathbf{x},\mathbf{y}) \approx \langle\hat{\Psi}(\mathbf{x}),\hat{\Psi}(\mathbf{y})\rangle$ back into (8) shows that the distance $D(x,y)$ is approximately equal to the Euclidean distance $\|\hat{\Psi}(\mathbf{x})-\hat{\Psi}(\mathbf{y})\|_2$ between the n-dimensional feature vectors $\hat{\Psi}(\mathbf{x}),\hat{\Psi}(\mathbf{y}) \in \mathbb{R}^n$.

Feature maps for RBF kernels. A (Gaussian) *radial basis function (RBF)* kernel has the form

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{y}\|_2^2\right). \tag{28}$$

A *generalized RBF kernel (GRBF)* is a RBF kernel where the Euclidean distance $\|\cdot\|$ in (28) is replaced by another metric $D(x,y)$:

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2}D^2(\mathbf{x},\mathbf{y})\right). \tag{29}$$

GRBF kernels based on additive distances such as χ^2 are some of the best kernels for certain machine learning and computer vision applications. Features for such kernels can be computed by combining the random Fourier features and the features introduced here for the the homogeneous additive kernels. Specifically, let $\hat{\Psi}(\mathbf{x}) \in \mathbb{R}^n$ be an approximated feature map for the metric $D^2(\mathbf{x},\mathbf{y})$ and let $\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_n \in \mathbb{R}^n$ be sampled from $\kappa(\boldsymbol{\omega})$ (which is a Gaussian density). Then

$$\hat{\Psi}_{\mathrm{GRBF}}(\mathbf{x}) = \frac{1}{\sqrt{n}}\left[e^{-i\langle\boldsymbol{\omega}_1,\hat{\Psi}(\mathbf{x})\rangle}\dots e^{-i\langle\boldsymbol{\omega}_n,\hat{\Psi}(\mathbf{x})\rangle}\right]^\top \tag{30}$$

is an approximated feature map for the GRBF kernel (29).

Computing $\hat{\Psi}_{\mathrm{GRBF}}(\mathbf{x})$ requires evaluating n random projections $\langle\boldsymbol{\omega}_i,\hat{\Psi}(\mathbf{x})\rangle$, which is relatively expensive. [30] proposes to reduce the number of projections in training, for instance by using a sparsity-inducing regularizer when learning an SVM.

8 Experiments

The following experiments compare the exact χ^2, intersection, Hellinger's, and linear kernels to the homogeneous kernel map (19), addKPCA [1] and MB [2] approximations in term of accuracy and speed. Methods are evaluated on three datasets: Caltech-101 and the DaimlerChrylser and INRIA pedestrians. In the Caltech-101 experiments we use a single but strong image feature (multi-scale dense SIFT) so that results are comparable to the state of the art on this dataset for methods that use only one visual descriptor. For the DaimlerChrylser pedestrians we used an improved version of the HOG-like features of [2]. In the INRIA pedestrians dataset, we use the standard HOG feature and compare directly to the state of the art results on this dataset, including those that have enhanced the descriptor and those that use non-linear kernels. In this case we investigate also stronger (structured output) training methods using our feature map approximations, since we can train kernelised models with the efficiency of a linear model, and without changes to the learning algorithm.

8.1 Caltech-101

This experiment tests the approximated feature maps on the problem of classifying the 102 (101 plus background) classes of the Caltech-101 benchmark dataset [3]. Images are rescaled to have a largest side of 480 pixels; dense SIFT features are extracted every four pixels at four scales (we use the `vl_phow` function of [31]); these are quantized in a 600 visual words dictionary learned using k-means. Each image is described by a 4200-dimensional histogram of visual words with 1×1, 2×2, and 4×4 spatial subdivisions [12].

For each of the 102 classes, 15 images are used for training and 15 for testing. Results are evaluated in term of the average classification accuracy across classes. The training time lumps together learning (if applicable) and evaluating the encoding and running the linear (LIBLINEAR) or non-linear (LIBSVM extended to support the homogeneous kernels) solvers [38]. All the experiments are repeated three times for different random selections of the 15+15 images and the results are reported as the mean accuracy and its standard error for the three runs.

The parameter Λ used by the homogeneous kernel map is selected automatically as a function of the approximation order n as described in Sect. 5.1. The only other free parameter is the constant C of the SVM, which controls the regularity of the learned classifier. This is tuned for each combination of kernel, feature map, and approximation order by using five-fold cross validation (*i.e.*, of the 15 training images per class, 12 are used for training and 3 for validating, and the optimal constant C is selected) by sampling logarithmically the interval $[10^{-1},10^2]$ (this range was determined in preliminary experiments). In practice, for each family of kernels (linear, homogeneous, and Gaussian RBF) the accuracy is very stable for a wide range of choices of C; among these optimal values, the smallest is preferred as this choice makes the SVM solvers converge faster.

Approximated low-dimensional feature maps. Tab. 1 compares the exact kernels, the homogeneous kernel maps, and the addKPCA maps of [1]. The accuracy of the linear kernel is quite poor (its accuracy is 10–15%

TABLE 1
Caltech-101: Approximated low-dimensional feature maps. Average per-class classification accuracy (%) and training time (in seconds) are reported for 15 training images per class according to the standard Caltech-101 evaluation protocol. The figure compares: the exact χ^2, intersection, and JS kernels, their homogeneous approximated feature maps (19), their $1/2$-homogeneous variants, and the addKPCA feature maps of [1]. The accuracy of the linear kernel is 41.6 ± 1.5 and its training time 108 ± 8; the accuracy of the Hellinger's kernel is 63.8 ± 0.8 and time 107.5 ± 8.

			χ^2		JS		inters.	
feat.	dm.	solver	acc.	time	acc.	time	acc.	time
exact	–	libsvm	64.1 ± 0.6	1769 ± 16	64.2 ± 0.5	6455 ± 39	62.3 ± 0.3	1729 ± 19
hom	3	liblin.	64.4 ± 0.6	312 ± 14	64.2 ± 0.4	483 ± 52	64.6 ± 0.5	367 ± 18
hom	5	liblin.	64.2 ± 0.5	746 ± 58	64.1 ± 0.4	804 ± 82	63.7 ± 0.6	653 ± 54
$\frac{1}{2}$-hom	3	liblin.	67.2 ± 0.7	380 ± 10	67.3 ± 0.6	456 ± 54	67.0 ± 0.6	432 ± 55
pca	3	liblin.	64.3 ± 0.5	682 ± 67	64.5 ± 0.6	1351 ± 93	62.7 ± 0.4	741 ± 41
pca	5	liblin.	64.1 ± 0.5	1287 ± 156	64.1 ± 0.5	1576 ± 158	62.6 ± 0.4	1374 ± 257

TABLE 2
Caltech-101: MB embedding. MB denotes the dense encoding of [2], sp-MB the sparse version, and hom. the homogeneous kernel map for the intersection kernel. In addition to LIBLINEAR and LIBSVM, a cutting plane solver is compared as well.

feat.	dm.	solver	regul.	acc.	time
exact	–	libsvm	diag	62.3 ± 0.3	1729 ± 19
MB	3	liblin.	diag.	62.0 ± 0.7	220 ± 14
MB	5	liblin.	diag.	62.2 ± 0.4	253 ± 17
hom.	3	liblin.	diag.	64.6 ± 0.5	367 ± 18
sp-MB	16	cut	H	62.1 ± 0.6	1515 ± 243
hom.	3	cut	diag.	64.0 ± 0.5	1929 ± 248

worse than the other kernels). The χ^2 and JS kernels perform very well, closely followed by the Hellinger's and the intersection kernels which lose 1-2% classfication accuracy. Our and the addKPCA approximation match/outperform the exact kernels even with as little as 3D. The $\gamma = 1/2$ variants (Sec. 2.1) perform better still, probably because they tend to reduce the effect of large peaks in the feature histograms. As expected, training with LIBLINEAR using the approximations is much faster than using exact kernels in LIBSVM (moreover this speedup grows linearly with the size of the dataset). The speed advantage of the homogeneous kernel map over the addKPCA depends mostly on the cost of learning the embedding, and this difference would be smaller for larger datasets.

MB embedding. Tab. 2 compares using the dense and sparse MB embeddings and the homogeneous kernel map. LIBLINEAR does not implement the correct regularization for the sparse MB embedding [2]; thus we evaluate also a cutting plane solver (one slack formulation [6]) that, while implemented in MATLAB and thus somewhat slower in practice, does support the appropriate regularizer H (Sect. 6). Both the dense low-dimensional and the sparse high-dimensional MB embeddings saturate the performance of the exact intersection kernel. Due to sparsity, the speed of the high dimensional variant is similar to the homogeneous kernel map of dimension 3.

8.2 DaimlerChrysler pedestrians

The DaimlerChrysler pedestrian dataset and evaluation protocol are described in detail in [4]. Here we just remind the reader that the task is to discriminate 18×36 gray-scale image patches portraying a pedestrian (positive samples) or clutter (negative samples). Each classifier is trained using 9,600 positive patches and 10,000 negative ones and tested on 4,800 positive and 5,000 negative patches. Results are reported as the average and standard error of the Equal Error Rate (EER) of the Receiver Operating Characteristic (ROC) over six splits of the data [4]. All parameters are tuned by three-fold cross-validation on the three training sets as in [4].

Two HOG-like [5] visual descriptors for the patches were tested: the multi-scale HOG variant of [2], [39] (downloaded from the authors' website), and our own implementation of a further simplified HOG variant which does not use the scale pyramid. Since the latter performed better in all cases, results are reported for it, but they do not change qualitatively for the other descriptor. A summary of the conclusions follows.

Approximated low-dimensional feature maps. In Tab. 3 the χ^2, JS, and intersection kernels achieve a 10% relative reduction over the EER of the Hellinger's and linear kernels. The homogeneous feature maps (19) and addKPCA [1] with $n \geq 3$ dimensions perform as well as the exact kernels. Speed-wise, the approximated feature maps are slower than the linear and Hellinger's kernels, but roughly one-two orders of magnitude faster than the exact variants. Moreover the speedup over the exact variants is proportional to the number of training data points. Finally, as suggested in Sect. 2.2, using an incorrect normalization for the data (l^2 for the homogeneous kernels) has a negative effect on the performance.

Generalized Gaussian RBF variants. In Tab. 4 the exact Generalized Gaussian RBF kernels are found to perform significantly better (20–30% EER relative reduction) than the homogeneous and linear kernels. Each RBF kernel

TABLE 3

DC pedestrians: Approximated low-dimensional feature maps. Average ROC EER (%) and training times (in seconds) on the DC dataset. The table compares the same kernels and approximations of Tab. 1. The EER of the linear kernel is $10.4_{\pm 0.5}$ (training time $1_{\pm 0.1}$) and the Hellinger's kernel $10.4_{\pm 0.5}$ (training time $2_{\pm 0.1}$.) The table reports also the case in which an incorrect data normalization is used (l^2 norm for the homogeneous kernels).

				χ^2		JS		inters.	
feat.	dm.	solver	norm	EER	time	EER	time	EER	time
exact	–	libsvm	l^1	$8.9_{\pm 0.5}$	$550_{\pm 36}$	$9.0_{\pm 0.5}$	$2322_{\pm 130}$	$9.2_{\pm 0.5}$	$797_{\pm 52}$
hom	3	liblin.	l^1	$9.1_{\pm 0.5}$	$12_{\pm 0}$	$9.2_{\pm 0.5}$	$10_{\pm 0}$	$9.0_{\pm 0.5}$	$9_{\pm 0}$
hom	5	liblin.	l^1	$9.0_{\pm 0.5}$	$15_{\pm 0}$	$9.2_{\pm 0.5}$	$15_{\pm 0}$	$9.0_{\pm 0.5}$	$13_{\pm 0}$
pca	3	liblin.	l^1	$9.0_{\pm 0.4}$	$26_{\pm 1}$	$9.1_{\pm 0.5}$	$59_{\pm 1}$	$9.2_{\pm 0.5}$	$22_{\pm 1}$
pca	5	liblin.	l^1	$9.0_{\pm 0.5}$	$41_{\pm 2}$	$9.1_{\pm 0.5}$	$74_{\pm 1}$	$9.0_{\pm 0.5}$	$40_{\pm 1}$
hom	3	liblin.	l^2	$9.5_{\pm 0.5}$	$10_{\pm 0}$	$9.4_{\pm 0.5}$	$14_{\pm 1}$	$9.4_{\pm 0.6}$	$12_{\pm 0}$
hom	5	liblin.	l^2	$9.4_{\pm 0.5}$	$16_{\pm 0}$	$9.4_{\pm 0.5}$	$23_{\pm 2}$	$9.6_{\pm 0.5}$	$19_{\pm 1}$

TABLE 4

DC pedestrians: Generalized Gaussian RBFs. The table compares the exact RBF kernels to the random Fourier features (combined with the homogeneous kernel map approximations for the χ^2, JS, and intersection variants).

			χ^2-RBF		JS-RBF		inters.-RBF		linear-RBF		Hell.-RBF	
feat.	dm.	solver	EER	time	EER	time	EER	time	EER	time	EER	time
exact	–	libsvm	$7.0_{\pm 0.4}$	$1580_{\pm 75}$	$7.0_{\pm 0.4}$	$7959_{\pm 373}$	$7.5_{\pm 0.4}$	$1367_{\pm 69}$	$7.2_{\pm 0.4}$	$1634_{\pm 253}$	$7.4_{\pm 0.3}$	$2239_{\pm 99}$
r.f.	2	liblin.	$13.0_{\pm 0.4}$	$14_{\pm 0}$	$12.9_{\pm 0.4}$	$14_{\pm 0}$	$12.4_{\pm 0.4}$	$13_{\pm 0}$	$12.9_{\pm 0.4}$	$6_{\pm 0}$	$12.6_{\pm 0.3}$	$7_{\pm 0}$
r.f.	16	liblin.	$8.6_{\pm 0.4}$	$81_{\pm 2}$	$8.6_{\pm 0.4}$	$92_{\pm 2}$	$8.4_{\pm 0.5}$	$77_{\pm 1}$	$9.2_{\pm 0.4}$	$58_{\pm 9}$	$8.8_{\pm 0.3}$	$72_{\pm 13}$
r.f.	32	liblin.	$8.0_{\pm 0.5}$	$154_{\pm 2}$	$8.2_{\pm 0.4}$	$177_{\pm 7}$	$8.0_{\pm 0.4}$	$222_{\pm 39}$	$8.6_{\pm 0.4}$	$113_{\pm 4}$	$8.4_{\pm 0.4}$	$112_{\pm 4}$

is also approximated by using the random Fourier feature map (Sect. 7), combined with the homogeneous kernel map (of dimension 3) for the χ^2-RBF, JS-RBF, and intersection-RBF kernels. For consistency with the other tables, the dimensionality of the feature is given relatively to the dimensionality of the original descriptor (for instance, since the HOG descriptors have dimension 576, dimension 32 corresponds to an overall dimension of $32 \times 576 = 18{,}432$, *i.e.* $9{,}216$ random projections). As the number of random projections increases, the accuracy approaches the one of the exact kernels, ultimately beating the additive kernels, while still yielding faster training than using the exact non-linear solver. In particular, combining the random Fourier features with the homogeneous kernel map performs slightly better than approximating the standard Gaussian kernel. However, the computation of a large number of random projections makes this encoding much slower than the homogeneous kernel map, the addKPCA map, and the MB encoding. Moreover, a large and dense expansion weights negatively in term of memory consumption as well.

8.3 INRIA pedestrians

The experiment (Fig. 4) compares our low-dimensional χ^2 approximation to a linear kernel in learning a pedestrian detector for the INRIA benchmark [5]. Both the standard HOG descriptor (insensitive to the gradient direction) and the version by [41] (combining direction sensitive and insensitive gradients) are tested. Training uses a variant of the structured output framework proposed by [42] and the cutting plane algorithm by [43]. Compared to conventional SVM based detectors, for which negative detection windows must be determined through retraining [5], the structural SVM has access to a virtually infinite set of negative data. While this is clearly an advantage, and while the cutting plane technique [43] is very efficient with linear kernels, its kernelised version is extremely slow. In particular, it was not feasible to train the structural SVM HOG detector with the exact χ^2 kernel in a reasonable time, but it was possible to do so by using our low dimensional χ^2 approximation in less than an hour. In this sense, our method is a key enabling factor in this experiment.

We compare our performance to state of the art methods on this dataset, including enhanced features and non-linear kernels. As shown in Fig. 4, the method performs very well. For instance, the miss rate at false positive per window rate (FPPW) 10^{-4} is 0.05 for the HOG descriptor from Felzenszwalb *et al.* [41] with the χ^2 3D approximated kernel, whereas Ott and Everingham [40] reports 0.05 integrating HOG with image segmentation and using a quadratic kernel, Wang *et al.* [44] reports 0.02 integrating HOG with occlusion estimation and a texture descriptor, and Maji *et al.* [39] reports 0.1 using HOG with the exact intersection kernel (please refer to the corrected results in [45]).

Notice also that adding the χ^2 kernel approximation yields a significant improvement over the simple linear detectors. The relative improvement is in fact larger than the one observed by [39] with the intersection kernel, and by [40] with the quadratic kernel, both exact.

Compared to Maji *et al.* [39], our technique also has an edge on the testing efficiency. [39] evaluates an ad-

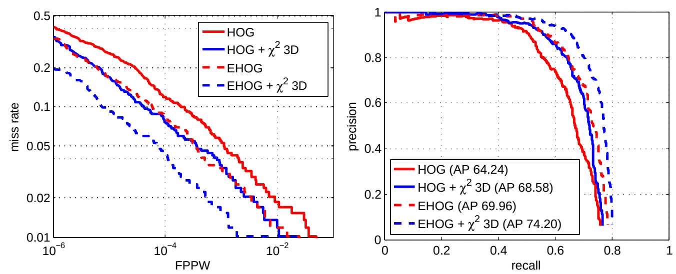


Fig. 4. **INRIA pedestrians.** Evaluation of HOG based detectors learned in a structural SVM framework. DET [5] and PASCAL-style precision-recall [40] curves are reported for both HOG and an extended version [41], dubbed EHOG, detectors. In both cases the linear detectors are shown to be improved significantly by the addition of our approximated 3D χ^2 feature maps, despite the small dimensionality.

ditive kernel HOG detector in time $T_{\text{look}}BL$, where B is the number of HOG components, L the number of window locations, and T_{look} the time required to access a look-up table (as the calculation has to be carried out independently for each component). Instead, our 3D χ^2 features can be precomputed once for all HOG cells in an image (by using look-up tables in time $T_{\text{look}}L$). Then the additive kernel HOG detector can be computed in time $T_{\text{dot}}BL$, where T_{dot} is the time required to multiply two feature vectors, i.e. to do three multiplications. Typically $T_{\text{dot}} \ll T_{\text{look}}$, especially because fast convolution code using vectorised instructions can be used.

9 Summary

Supported by a novel theoretical analysis, we derived the homogeneous kernel maps, which provide fast, closed form, and very low dimensional approximations of all common additive kernels, including the intersection and χ^2 kernels. An interesting result is that, even though each dimension is approximated independently, the overall approximation quality is independent of the data dimensionality.

The homogeneous kernel map can be used to train kernelised models with algorithms optimised for the linear case, including standard SVM solvers such as LIBSVM [38], stochastic gradient algorithms, on-line algorithms, and cutting-plane algorithms for structural models [6]. These algorithms apply unchanged; however, if data storage is a concern, the homogeneous kernel map can be computed on the fly inside the solver due to its speed.

Compared to the addKPCA features of Perronnin *et al.* [1], the homogeneous kernel map has the same classification accuracy (for a given dimension of the features) while not requiring any training. The main advantages of addKPCA are: a better reconstruction of the kernel for a given data distribution and the ability to generate features with an even number of dimensions ($n = 2, 4, ...$).

Compared to the MB sparse encoding [2], the homogeneous kernel map approximates all homogeneous kernels (not just the intersection one), has a similar speed in training, and works with off-the-shelf solvers because it does not require special regularizers. Our accuracy is in some cases a little better due to the ability of approximating the χ^2 that was found to be occasionally superior to the intersection kernel.

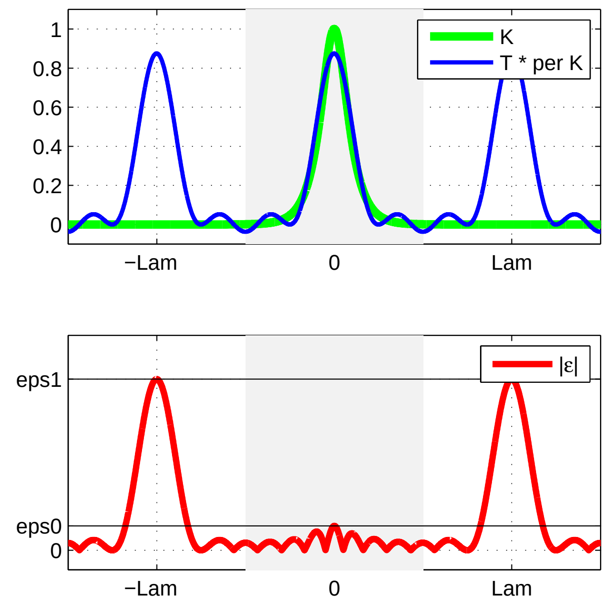


Fig. 5. **Error structure.** Top: the signature $\mathcal{K}$ of the a χ^2 kernel and its approximation $\hat{\mathcal{K}} = T * \mathrm{per}_\Lambda \mathcal{K} W$ for $W(\lambda) = 1$ (see (21)). Bottom: the error $\epsilon(\lambda) = \hat{\mathcal{K}}(\lambda) - \mathcal{K}(\lambda)$ and the constants ϵ_0 and ϵ_1 in Lemma 3. The approximation is good in the region $|\lambda| < \Lambda/2$ (shaded area).

The Hellinger's kernel, which has a trivial encoding, was found sometimes to be competitive with the other kernels as already observed in [1]. Finally, the γ-homogeneous variants of the homogeneous kernels was shown to perform better than the standard kernels on some tasks.

10 Proofs

10.1 Error bounds

This section shows how to derive the asymptotic error rates of Sect. 5. By measuring the approximation error of the feature maps from (18) or (19) based on the quantities $\epsilon_{\text{s}}(x, y)$ and $\epsilon_{\text{h}}(x, y)$, as defined in (23) and (24), it is possible to unify the analysis of the stationary and homogeneous cases. Specifically, let $\mathcal{K}$ denote the signature of the original kernel and $\hat{\mathcal{K}}$ of the approximated kernel obtained from (18) or (19). Define as in Lemma 3 the function $\epsilon(\lambda) = \mathcal{K}(\lambda) - \hat{\mathcal{K}}(\lambda)$. Then, based on (23) and (24), the approximation errors are given respectively by $\epsilon_{\text{s}}(x, y) = \epsilon(y - x)$, $\epsilon_{\text{h}}(x, y) = \epsilon(\log y - \log x)$.

To use Lemma 3 we need to bound the quantities $\epsilon_1 = \sup_\lambda |\epsilon(\lambda)|$ (maximum error) and $\epsilon_0 = \sup_{|\lambda|\leq\Lambda/2} |\epsilon(\lambda)|$ (maximum error in a period). Since the signature $\mathcal{K}(\lambda)$ of most kernels decays while the signature $\hat{\mathcal{K}}(\lambda)$ of the approximation is periodic, we can hope that ϵ_0 to be small, but ϵ_1 is at best bounded. In fact we have the following Lemma, proved in Sect. 10.2:

Lemma 4. *If there exist constants $A, \alpha > 0$ and $B, \beta > 0$ such that the kernel signature is bounded by $K(\lambda) \leq Ae^{-\alpha|\lambda|}$ and the spectrum by $\kappa(\omega) \leq Be^{-\beta|\omega|}$, then the constants ϵ_0 and ϵ_1 of Lemma 3 can be set to*

$$\begin{aligned} \epsilon_0(n, \Lambda) &= \frac{2Ae^{-\frac{\alpha\Lambda}{2}}}{1 - e^{-\alpha\Lambda}} + \frac{4\pi B}{\Lambda} \frac{e^{-\frac{\pi(n+1)\beta}{\Lambda}}}{1 - e^{-\frac{2\pi\beta}{\Lambda}}}, \\ \epsilon_1(\Lambda) &= A\frac{1 + e^{-\alpha\Lambda}}{1 - e^{-\alpha\Lambda}}. \end{aligned} \tag{31}$$

If there exist $A, \alpha > 0$ as above and B and $\beta > 1$ such that $\kappa(\omega) \leq B|\omega|^{-\beta}$, then $\epsilon_0(n, \Lambda)$ should be replaced with

$$\epsilon_0(n, \Lambda) = \frac{2Ae^{-\frac{\alpha\Lambda}{2}}}{1 - e^{-\alpha\Lambda}} + \frac{2B}{\beta - 1} \left(\frac{\pi(n-1)}{\Lambda}\right)^{1-\beta}. \tag{32}$$

Based on (31) and (32), in order to minimize the error (25) one has to simultaneously increase the feature dimensionality n and the period Λ. Substituting (31) into (25) yields the approximation error upper bound $O(e^{-\min\{\frac{\alpha}{2},\frac{\gamma}{4}\}\Lambda} + e^{-\frac{\pi(n+1)\beta}{\Lambda}})$; equating the rates of the two terms yields $\Lambda = \sqrt{2\pi(n+1)\beta/\min\left\{\frac{\alpha}{2}, \frac{\gamma}{4}\right\}}$ which results in an overall rate of the error bound of $O(e^{-\sqrt{\min\{\frac{\alpha}{2},\frac{\gamma}{4}\}\pi(n+1)\beta}})$. Similarly, in the case of (32), by setting $\Lambda = (\beta - 1)(\log n)/\min\left\{\frac{\alpha}{2}, \frac{\gamma}{4}\right\}$ one obtains the rate $\epsilon_0 = O((n/\log n)^{1-\beta})$.

10.2 Proofs of the lemmas

Proof of Lemma 1: To show that $\mathcal{K}(\lambda)$ is a PD function one must show that, for any $\alpha_1, \dots, \alpha_K \in \mathbb{C}$ and $\lambda_1, \dots, \lambda_K \in \mathbb{R}$, $\sum_{ij} \alpha_i^* \alpha_j \mathcal{K}(\lambda_j - \lambda_i) \geq 0$. For a stationary kernel, substituting $\mathcal{K}(\lambda_j - \lambda_i) = k_s(\lambda_i, \lambda_j)$ yields the corresponding notion of positive definiteness of the kernel k_s. The two notions are therefore equivalent.

In the γ-homogeneous case, consider a PD kernel $k_h(x, y) = (xy)^{\gamma/2}\mathcal{K}(\log y - \log x)$ and define $x_i = e^{\lambda_i} > 0$. Then if k is PD:

$$\begin{aligned} &\sum_{ij} \alpha_i^* \alpha_j \mathcal{K}(\lambda_j - \lambda_i) \\ &= \sum_{ij} \frac{\alpha_i^*}{x_i^{\frac{\gamma}{2}}} \frac{\alpha_j}{x_j^{\frac{\gamma}{2}}} x_i^{\frac{\gamma}{2}} x_j^{\frac{\gamma}{2}} \mathcal{K}\left(\log \frac{x_j}{x_i}\right) = \sum_{ij} \tilde{\alpha}_i^* \tilde{\alpha}_j k_h(x_i, x_j) \geq 0 \end{aligned}$$

so that $\mathcal{K}$ is PD. The other direction is also true once one notices that, for homogeneity, $k_h(x_i, x_j) = 0$ if $x_i = 0$ or $x_j = 0$. □

Proof of Lemma 2: $\operatorname{sign}(xy)k(|x|, |y|)$ is real and symmetric and PD because for any sequence of points $x_1, \dots, x_n \in \mathbb{R}$ and coefficients $\alpha_1, \dots, \alpha_n \in \mathbb{C}$ one has $\sum_{ij} \alpha_i^* \alpha_j \operatorname{sign}(x_i x_j)k(|x_i|, |x_j|) = \sum_{ij}(\alpha_i \operatorname{sign} x_i)^*(\alpha_j \operatorname{sign} x_j)k(|x_i|, |x_j|) \geq 0$. Finally, $\frac{1}{2}(\operatorname{sign}(xy) + 1)k(|x|, |y|)$ is also PD because it is the non-negative additive combination of PD kernels. □

Proof of Lemma 4: Assume $\mathcal{K}(\lambda) \leq Ae^{-\alpha|\lambda|}$. Then for $|\lambda| \leq \Lambda/2$

$$|(\operatorname*{per}_\Lambda \mathcal{K} - \mathcal{K})(\lambda)| \leq 2A \sum_{t=1}^{+\infty} e^{-\alpha(\lambda+t\Lambda)} \leq \frac{2Ae^{-\frac{\alpha\Lambda}{2}}}{1 - e^{-\alpha\Lambda}}.$$

and for $|\lambda| > \Lambda/2$

$$|(\operatorname*{per}_\Lambda \mathcal{K} - \mathcal{K})(\lambda)| \leq |\operatorname*{per}_\Lambda \mathcal{K}(\lambda)| \leq A\frac{1 + e^{-\alpha\Lambda}}{1 - e^{-\alpha\Lambda}}.$$

Let $L = 2\pi/\Lambda$. Truncating the spectrum $\kappa(jL)$ to the interval $j = -n, \dots, n$ yields the approximated signature $\hat{\mathcal{K}}(\lambda)$. If $\kappa(\omega) \leq Be^{-\beta|\omega|}$ then

$$\begin{aligned} &|(\hat{\mathcal{K}} - \operatorname*{per}_\Lambda \mathcal{K})(\lambda)| = 2\left|\sum_{j=\frac{n+1}{2}}^{+\infty} L\kappa(jL)e^{-ijL\lambda}\right| \\ &\leq 2L \sum_{j=\frac{n+1}{2}}^{+\infty} Be^{-\beta jL} \leq 2LB\frac{e^{-\beta\frac{n+1}{2}L}}{1 - e^{-\beta L}} = \frac{4\pi B}{\Lambda} \frac{e^{-\frac{\pi(n+1)\beta}{\Lambda}}}{1 - e^{-\frac{2\pi\beta}{\Lambda}}}. \end{aligned}$$

Summing the two estimates yields (31). If on the other hand $\kappa(\omega) \leq B|\omega|^{-\beta}$, one has

$$|(\hat{\mathcal{K}} - \operatorname*{per}_\Lambda \mathcal{K})(\lambda)| \leq 2L \sum_{j=\frac{n+1}{2}}^{+\infty} B(jL)^{-\beta} \leq \frac{2B}{\beta - 1} \frac{1}{\left(\frac{L(n-1)}{2}\right)^{\beta-1}}.$$

which yields the estimate (32). □

Proof of Lemma 3: Consider the normalized approximation error $\epsilon_h(x, y)$ (24) for a γ-homogeneous kernel $k_h(x, y)$. Based on the the definiton (24), $\epsilon_h(x, y) = \sup_\lambda \epsilon(\lambda)$, where ϵ is the difference between the original and approximated signature as given in Lemma 3. Recall that in the homogeneous case the arguments x and y are non negative. Hence in general $\sqrt{xy} \leq \max\{x, y\}$. If $|\log y/x| \leq \Lambda/2$, then $\epsilon_h(x, y) \leq \epsilon_0$ where ϵ_0 is defined in Lemma 3. On the other hand, if $|\log y/x| > \Lambda/2$, then $\epsilon_h(x, y) \leq \epsilon_1$ applies, but one has $\min\{x, y\} \leq \max\{x, y\}2^{-\Lambda/2}$, so that $\sqrt{xy} \leq \max\{x, y\}e^{-\Lambda/4}$. Multiplying the error $\epsilon_h(x, y)$ by the homogenoeus factor $(xy)^\gamma$ yields the approximation error: bound

$$\begin{aligned} |k_h(x, y) - \langle\hat{\Psi}(x), \hat{\Psi}(y)\rangle| &= (xy)^{\gamma/2}|\epsilon_h(x, y)| \\ &\leq \max\left\{\epsilon_0, \epsilon_1 e^{\frac{-\Lambda\gamma}{4}}\right\} \max\{x^\gamma, y^\gamma\}. \end{aligned}$$

The overall error of approximationg the additive combination $K(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^D k_h(\mathbf{x}_l, \mathbf{y}_l)$ is then bounded by

$$|K(\mathbf{x}, \mathbf{y}) - \langle\hat{\Psi}(\mathbf{x}), \hat{\Psi}(\mathbf{y})\rangle| \leq \max\left\{\epsilon_0, \epsilon_1 e^{\frac{-\Lambda\gamma}{4}}\right\} (\|\mathbf{x}\|_\gamma^\gamma + \|\mathbf{y}\|_\gamma^\gamma)$$

Assuming that the histograms are normalized one obtains (25). □

ACKNOWLEDGMENTS

We are grateful for financial support from the Royal Academy of Engineering, Microsoft, ERC grant VisRec no. 228180, and ONR MURI N00014-07-1-0182.

References

[1] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Proc. CVPR*, 2010.

[2] S. Maji and A. C. Berg, "Max-margin additive classifiers for detection," in *Proc. ICCV*, 2009.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. ICCV*, 2003.

[4] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *PAMI*, vol. 28, no. 11, 2006.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005.

[6] T. Joachims, "Training linear SVMs in linear time," in *Proc. KDD*, 2006.

[7] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. NIPS*, 2007.

[8] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texure and object categories: An in-depth study," INRIA, Tech. Rep., 2005.

[9] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop on Stat. Learn. in Comp. Vision*, 2004.

[10] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. ICCV*, 2003.

[11] K. Grauman and T. Darrel, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. ICCV*, 2005.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006.

[13] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale L2-loss linear support vector machines," *Journal of Machine Learning Research*, 2008.

[14] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-GrAdient SOlver for SVM," in *Proc. ICML*, 2007.

[15] L. Bottou and C.-J. Lin, "Support vector machine solvers," in *Large scale kernel machines*, L. Bottou, O. Chappelle, D. DeCoste, and J. Weston, Eds. MIT Press, 2007.

[16] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. NIPS*, 2008.

[17] J. Duchi and Y. Singer, "Efficient learning using forward-backward splitting," in *Proc. NIPS*, 2009.

[18] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[19] C. K. I. Williams and M. Seeger, "The effect of the input density distribution on kernel-based classifiers," in *Proc. ICML*, 2000.

[20] ——, "Using the Nyström method to speed up kernel machines," in *Proc. NIPS*, 2001.

[21] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. ICML*, 2000.

[22] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *Journal of Machine Learninig Research*, 2001.

[23] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learninig Research*, vol. 3, no. 1, 2002.

[24] ——, "Predictive low-rank decomposition for kernel methods," in *ICML*, 2005.

[25] E. Snelson and Z. Ghaharamani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. NIPS*, 2006.

[26] L. Bo and C. Sminchisescu, "Efficient match kernels between sets of features for visual recognition," in *Proc. NIPS*, 2009.

[27] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. NIPS*, 2009.

[28] F. Li, C. Ionescu, and C. Sminchisescu, "Random fourier approximations for skewed multiplicative histogram kernels," in *Proc. DAGM*, 2010.

[29] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability measures," in *Proc. AISTAT*, 2005.

[30] V. Sreekanth, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Generalized RBF feature maps for efficient detection," in *Proc. BMVC*, 2010.

[31] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[32] B. Schölkopf, "The kernel trick for distances," *Proc. NIPS*, 2001.

[33] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *Proc. ICIP*, 2003.

[34] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in *Proc. ICCV*, 1999.

[35] D. R. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *PAMI*, vol. 1, 2004.

[36] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proc. ICCV*, 2009.

[37] B. Schölkopf, "Kernel means," 2007, lecture Slides.

[38] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, 2008.

[39] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. CVPR*, 2008.

[40] P. Ott and M. Everingham, "Implicit color segmentation features for pedestrian and object detection," in *Proc. ICCV*, 2009.

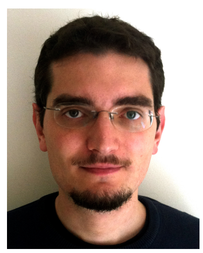
[41] P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, 2009.

[42] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *Proc. ECCV*, 2008.

[43] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, 2009.

[44] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. ICCV*, 2009.

[45] S. Maji, A. C. Berg, and J. Malik. http://www.cs.berkeley.edu/~smaji/projects/ped-detector/.



Andrea Vedaldi received the BSc degree (with honors) from the Information Engineering Department, University of Padua, Italy, in 2003 and the MSc and PhD degrees from the Computer Science Department, University of California at Los Angeles, in 2005 and 2008. Since 2008 he his research fellow at the University of Oxford, UK. His research interests include detection and recognition of visual object categories, visual representations, and large scale machine learning applied to computer vision. He is the recipient of the "Outstanding Doctor of Philosophy in Computer Science" and "Outstanding Master of Science in Computer Science" awards of University of California at Los Angeles. He is a member of the IEEE.

Andrew Zisserman is the Professor of Computer Vision Engineering at the Department of Engineering Science, University of Oxford.