Efficient Additive Kernels via Explicit Feature Maps

Andrea Vedaldi Andrew Zisserman
Dept. of Engineering Science, University of Oxford, UK
{vedaldi,az}@robots.ox.ac.uk

Abstract

Maji and Berg [13] have recently introduced an explicit feature map approximating the intersection kernel. This enables efficient learning methods for linear kernels to be applied to the non-linear intersection kernel, expanding the applicability of this model to much larger problems.

In this paper we generalize this idea, and analyse a large family of additive kernels, called homogeneous, in a unified framework. The family includes the intersection, Hellinger's, and χ^2 kernels commonly employed in computer vision. Using the framework we are able to: (i) provide explicit feature maps for all homogeneous additive kernels along with closed form expression for all common kernels; (ii) derive corresponding approximate finite-dimensional feature maps based on the Fourier sampling theorem; and (iii) quantify the extent of the approximation.

We demonstrate that the approximations have indistinguishable performance from the full kernel on a number of standard datasets, yet greatly reduce the train/test times of SVM implementations. We show that the χ^2 kernel, which has been found to yield the best performance in most applications, also has the most compact feature representation. Given these train/test advantages we are able to obtain a significant performance improvement over current state of the art results based on the intersection kernel.

1. Introduction

Recent advances have made it possible to learn linear support vector machines (SVMs) in time linear with the number of training examples [10], extending the applicability of these models to large scale datasets, on-line learning, and structural problems. Since non-linear SVMs can be seen as linear SVMs operating in an appropriate feature space, there is at least the theoretical possibility of extending such efficient learning methods to a much more general class of models. The success of this idea requires that (i) the feature map can be efficiently computed, and (ii) the corresponding feature space is sufficiently low dimensional.

Many computer vision representations, such as bag of visual words [3, 19] and spatial pyramids [8, 12], can be regarded as probability distributions. To use them in the context of SVMs, one needs to define an appropriate similarity function $K(\mathbf{x}, \mathbf{y})$ between finite probability distributions $\mathbf{x}, \mathbf{y}$ (i.e. normalized histograms). The only requirement is that K must be a positive definite (PD) function. The latter property also guarantees the existence of a feature map, but this is usually difficult to compute and high dimensional.

The subfamily of additive PD kernels, which includes the intersection, χ^2, and Hellinger's kernels, has consistently been found to give good performances in applications. Recently Maji *et al.* [15] showed that such kernels yield SVM classifiers with a greatly reduced test cost, leading to a 10^3-fold speed-ups in certain applications. In [13], Maji and Berg proposed an approximate, closed form finite feature map for the intersection kernel, one member of the additive subfamily. As suggested above, such a feature map was shown to speed-up training substantially.

In this work we seek finite dimensional feature maps for additive kernels. 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.

To this end, inspired by [9], we present a novel unified analysis of a large family of additive kernels, known as homogeneous kernels (Sect. 2). This class includes the intersection as well as the χ^2 and Hellinger's (Battacharyya's) kernels, and many others. We show that any such kernel can be described by a function of a scalar variable, which we call the *kernel signature*. The signature is a powerful tool that enables: (i) the derivation of closed form feature maps based on 1D Fourier analysis (the 1D following from the scalar variable); (ii) the computation of finite, low dimensional, tight approximations of these feature maps for all common kernels (Sect. 3); and (iii) the analysis of the error the approximation (Sect. 4). We then generalise the kernels to their γ-homogeneous variants and study the related problem of histogram normalisation (Sect. 5). We conclude the theoretical analysis by contrasting our method to the one of Maji and Berg (MB) [13] (Sect. 6).

Empirically (Sect. 7) we compare our approximations to

the exact kernels, and obtain virtually the same performance despite using extremely compact feature maps and requiring a fraction of the training time. In this regime, we do either as well or better than the MB approximations. While the MB approximation can be used efficiently with a larger number of dimensions (at the cost of modifying the learning algorithms), this does not seem necessary as performance is already saturated. As an additional example of the flexibility of our compact feature maps, we show how they can be used in learning a χ^2 sliding window HOG detector in the context of the structural SVMs, and how they can be used to speedup testing as well.

We test our method on the DaimlerChrysler pedestrian dataset [16], the Caltech-101 dataset [6], and the INRIA pedestrian dataset [4]. In all cases we obtain results better than the ones reported by Maji *et al.* [13, 15]. In particular, our baseline INRIA detector has performance comparable to the state of the art on this dataset (and could be further enhanced by using an the improved HOG features of [17]), and our kernel map yields a larger performance improvement than that obtained by the intersection kernel map of [15].

Efficient code to compute our feature maps is available as part of the open source VLFeat library [20]. This code can be used to kernelise most linear models with minimal or no changes to their implementation.

2. Signature of an homogeneous kernel

For finite dimensional distributions (histograms) $\mathbf{x}, \mathbf{y}$, an *additive kernel* is given by

$$K(\mathbf{x}, \mathbf{y}) = \sum_{b=1}^{B} k(\mathbf{x}_b, \mathbf{y}_b). \tag{1}$$

where b is the bin index. Here $k : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is itself a PD kernel on the non-negative reals. We focus our attention on the cases in which $k(x, y)$ is *homogeneous*, i.e.

$$\forall c \geq 0 : k(cx, cy) = ck(x, y). \tag{2}$$

Examples: computer vision kernels. All common additive kernels used in computer vision, such as the intersection, χ^2, and Hellinger's kernels, are homogeneous kernels. Their expression is given in Fig. 1. These and other known kernels, such as the symmetrized Kullback-Leibler kernel, are member of a parametric family of homogeneous kernels introduced in [9].

Signature. By setting $c = \sqrt{xy}$ in (2), we can decompose any homogeneous kernel as

$$k(x, y) = \sqrt{xy}\, k\left(\sqrt{\frac{x}{y}}, \sqrt{\frac{y}{x}}\right) = \sqrt{xy}\, \mathcal{K}\left(\log \frac{y}{x}\right) \tag{3}$$

where we call

$$\mathcal{K}(\omega) = k\left(e^{-\omega/2}, e^{\omega/2}\right), \quad \omega = \log \frac{y}{x}$$

the *kernel signature*. Notice that the signature $\mathcal{K}(\omega)$ fully characterizes an homogeneous kernel and depends only on the scalar variable ω, equal to the logarithmic ratio $\log(y/x)$ of the kernel arguments x, y. As we will show next, the signature can be used to analyse the kernel properties, including computing explicit feature map representations and evaluating the error of finite approximations.

A note on the χ^2 kernel. Some authors define the additive 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]. Here we use instead the definition of Fig. 1, which makes the additive χ^2 kernel PD. If the histograms $\mathbf{x}, \mathbf{y}$ are l^1-normalised, the two definitions differ by a constant offset.

A note on the Hellinger's kernel. The Hellinger's kernel $k(x, y) = \sqrt{xy}$ is the simplest homogeneous kernel, as its signature is the constant $\mathcal{K}(\omega) = 1$. Any other kernel can be obtained from it by multiplying by an appropriate signature. This kernel is also known as Bhattacharyya's coefficient and takes his name from the fact that the corresponding metric is the Hellinger's distance.

3. From signatures to feature maps

A *feature map* $\Psi(x)$ for a kernel $k(x, y)$ is a function mapping x into a vector space with an inner product $\langle \cdot, \cdot \rangle$ such that

$$\forall x, y : k(x, y) = \langle \Psi(x), \Psi(y) \rangle.$$

In principle, a feature map can be used to convert the data into a format suitable for linear SVM solvers, which are much more efficient than generic kernelised solvers. Unfortunately, while all PD kernels have an associated feature map (the reproducing kernel Hilbert space), this is usually difficult to compute and high dimensional.

We introduce here a simple technique to analytically construct a feature map for the homogeneous kernels. This construction results in closed form feature maps for all commonly used kernels. In this sense, it is much more general than the method proposed in [13], which is restricted to the intersection kernel.

The derivation starts from Corollary 3.1 of [9], which states that, for any homogeneous kernel $k(x, y)$, there exists a symmetric non-negative measure $\kappa(\lambda)d\lambda$ on $\mathbb{R}$ (we assume for simplicity that the measure has a density function) such that

$$k(x, y) = \int_{-\infty}^{+\infty} x^{\gamma + i\lambda} y^{\gamma - i\lambda} \kappa(\lambda)\, d\lambda, \quad \gamma = \frac{1}{2}.$$

kernel	$k(x,y)$	$\mathcal{K}(\omega)$	$\kappa(\lambda)$	feature $[\Psi(x)]_\lambda$	γ-homogeneous variant
Hellinger's	$\sqrt{xy}$	1	$\delta(\lambda)$	$\sqrt{x}$	$(xy)^{\frac{\gamma}{2}}$
χ^2	$2\frac{xy}{x+y}$	$\operatorname{sech}(\omega/2)$	$\operatorname{sech}(\pi\lambda)$	$e^{i\lambda\log x}\sqrt{x\operatorname{sech}(\pi\lambda)}$	$2\frac{(xy)^{\frac{\gamma+1}{2}}}{x+y}$
intersection	$\min\{x,y\}$	$e^{-\lvert\omega\rvert/2}$	$\frac{2}{\pi}\frac{1}{1+4\lambda^2}$	$e^{i\lambda\log x}\sqrt{\frac{2x}{\pi}\frac{1}{1+4\lambda^2}}$	$\min\left\{x^{\frac{\gamma+1}{2}}y^{\frac{\gamma-1}{2}}, x^{\frac{\gamma-1}{2}}y^{\frac{\gamma+1}{2}}\right\}$

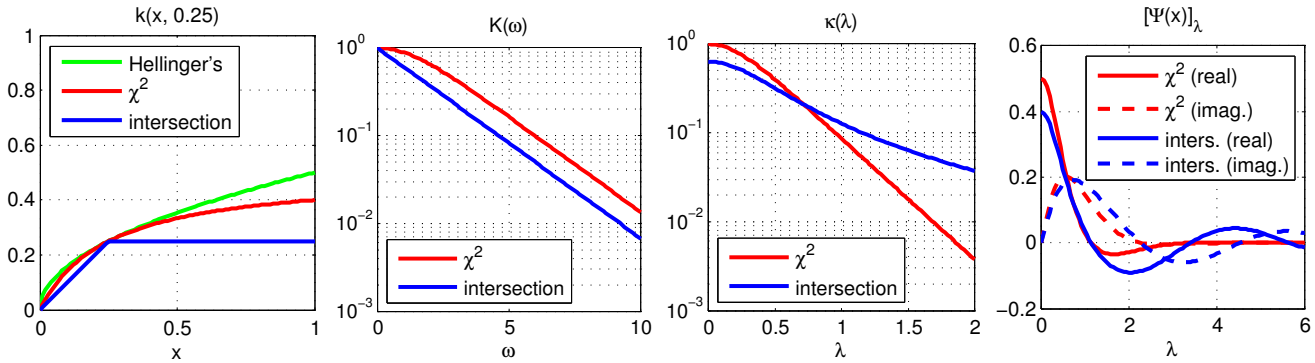

Figure 1: **Common kernels, their signatures, and their closed-form feature maps.** *Top*: closed form expressions for the χ^2, intersection, and Hellinger's (Battacharyya's) kernel, their signatures $\mathcal{K}(\omega)$ (Sect. 2), the inverse Fourier transform of the signatures $\kappa(\lambda)$, the feature maps $[\Psi(x)]_\lambda$ (Sect. 3), and the γ-homogeneous variant of the kernels (Sect. 5). *Bottom*: Plots of the various functions. On the left the χ^2 kernel is a smoother function than the intersection kernel. This is reflected in a faster fall off of the inverse Fourier transform of the signature $k(\lambda)$ (while the fall-off of the signature $K(\omega)$ is fast for both kernels), and ultimately in a better low-dimensional approximation (Fig. 2 and Sect. 4).

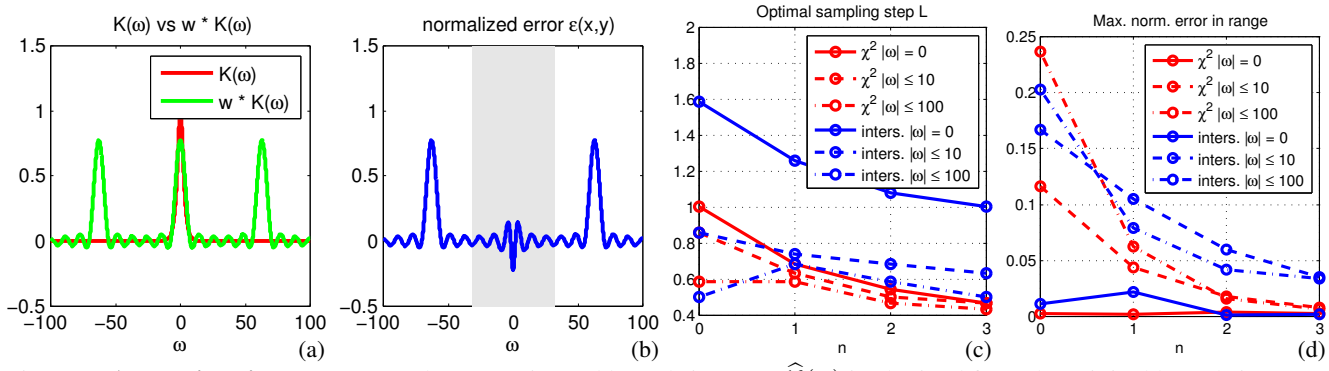

Figure 2: **Approximation error.** *(a)*: The approximated kernel signature $\widehat{\mathcal{K}}(\omega)$ is obtained from the original kernel signature $\mathcal{K}(\omega)$ by convolution with a periodic sinc (Sect. 4). *(b)*: The normalized error $\epsilon(x,y)$ is obtained, as a function of the logarithmic ratio $\omega = \log(y/x)$, by subtracting the exact signature $\mathcal{K}(\omega)$ from the approximated signature $\widehat{\mathcal{K}}(\omega)$. The approximation is good in the region $|\omega| < \pi/L$ (shaded area). *(c)*: optimal sampling period L obtained by minimizing the approximation error for a given number of samples n and a given validity range $|\omega| \leq M$. Results are shown for the intersection and χ^2 kernels, $n = 0, \ldots, 3$ and $M = 0, 10, 100$. *(d)*: Corresponding approximation errors (as suggested in Fig. 1 the χ^2 kernel is easier to approximate than the intersection kernel).

This can be rewritten as

$$k(x,y) = \sqrt{xy}\int_{-\infty}^{+\infty} e^{-i\lambda\log\frac{y}{x}}\kappa(\lambda)\,d\lambda. \tag{4}$$

Comparing this equation to (3) shows that the kernel signature $\mathcal{K}(\omega)$ must be equal to the Fourier transform of the measure $\kappa(\lambda)$, i.e.

$$\mathcal{K}(\omega) = \int_{-\infty}^{+\infty} e^{-i\lambda\omega}\kappa(\lambda)\,d\lambda. \tag{5}$$

Analytic form of feature maps. We can rewrite (4) as

$$k(x,y) = \int_{-\infty}^{+\infty} [\Psi(x)]_\lambda^*[\Psi(y)]_\lambda\,d\lambda$$

where we defined the complex function of the real variable λ

$$\boxed{[\Psi(x)]_\lambda = e^{-i\lambda\log x}\sqrt{x\kappa(\lambda)}}. \tag{6}$$

This is a key result as it gives an explicit form for the feature map as an infinite dimensional vector. Here $\lambda \in \mathbb{R}$ can be

though of as the index of the feature vector $\Psi(x)$ (and in fact in it will be converted into a discrete index in Sect. 4). The function $\kappa(\lambda)$ can be computed analytically as the inverse Fourier transform of the signature $\mathcal{K}(\omega)$. In fact, from (5) we have

$$\kappa(\lambda) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{i\lambda\omega} \mathcal{K}(\omega)\, d\omega. \tag{7}$$

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

4. Approximated finite feature maps

The infinite dimensional feature map (6) can be approximated by a finite number of samples, and this then defines a finite dimensional vector that can be used with a linear kernel. In this section we first use standard sampling theory to compute the error in the kernel resulting from this finite approximation, and then discuss choices of the sampling parameters (the number of samples n and the period L) which give an optimal approximation. This choice is important as we are interested in feature maps that are both easy to compute *and* low dimensional.

The finite dimensional feature map $\widehat{\Psi}(x)$ which approximates $\Psi(x)$ can be obtained by sampling (6) at points $\lambda = -nL, (-n+1)L, \ldots, +nL$. By exploiting the symmetry of $\Psi(x)$ and by assigning the real parts to the odd components of $\widehat{\Psi}(x)$ and the imaginary parts to the even components, the vector $\widehat{\Psi}(x) \in \mathbb{R}^{2n+1}$ can be defined as

$$\frac{[\widehat{\Psi}(x)]_j}{\sqrt{xL}} = \begin{cases} \sqrt{\kappa(0)}, & j = 0, \\ \sqrt{2\kappa(\frac{j+1}{2}L)} \cos\left(\frac{j+1}{2} L \log x\right) & j > 0 \text{ odd}, \\ \sqrt{2\kappa(\frac{j}{2}L)} \sin\left(\frac{j}{2} L \log x\right) & j > 0 \text{ even}, \end{cases} \tag{8}$$

where $j = 0, 1, \ldots, 2n$. For the common kernels, (8) yields closed form feature maps which are very simple and efficient to compute (see Fig. 1).

Definition (8) can be justified by analysing the corresponding kernel $\widehat{k}(x,y)$, and how this approximates the exact kernel $k(x,y)$. By using the symmetry of $\kappa(\lambda)$, we obtain, after a short computation,

$$\begin{aligned} \widehat{k}(x,y) &= \langle \widehat{\Psi}(x), \widehat{\Psi}(y) \rangle \\ &= \sqrt{xy} \sum_{j=-n}^{n} L\, \kappa(jL) \cos\left(jL \log \frac{x}{y}\right) \\ &= \sqrt{xy}\, \widehat{\mathcal{K}}\left(\log \frac{y}{x}\right) \end{aligned} \tag{9}$$

where the approximated signature $\widehat{\mathcal{K}}(\omega)$ is given by

$$\widehat{\mathcal{K}}\left(\log \frac{y}{x}\right) = \sum_{j=-n}^{n} L\, \kappa(jL) e^{-ijL \log \frac{y}{x}} \tag{10}$$

The approximated signature $\widehat{\mathcal{K}}(\omega)$ can be computed from the exact signature $\mathcal{K}(\omega)$ in a simple way (Fig. 2). In fact, as (5) states that the signature $\mathcal{K}(\omega)$ is the Fourier transform of $\kappa(\lambda)$, so (10) states that the approximated signature $\widehat{\mathcal{K}}(\omega)$ is the Fourier transform of the sampled and truncated signal $\kappa(nL)$, $j = -n-1, \ldots, n+1$. In the Fourier domain, subsampling and truncation correspond to convolution by the periodic sinc function

$$w(\omega) = \frac{L}{2\pi} \frac{\sin((2n+1)L\omega/2)}{\sin(L\omega/2)},$$

i.e. $\widehat{\mathcal{K}} = w * \mathcal{K}$. Substituting this relation into (3) yields an analytical expression for the approximation error:

$$\epsilon(x,y) = \frac{\widehat{k}(x,y)}{\sqrt{xy}} - \frac{k(x,y)}{\sqrt{xy}} = ((w-\delta) * \mathcal{K}) \left(\log \frac{y}{x}\right). \tag{11}$$

Notice that (i) the expression of the error is normalized by $\sqrt{xy}$ and that (ii) the function $\epsilon(x,y)$ depends on the ratio y/x only.

Conditions for a good approximation. In order to understand the structure of the error (11), notice that the sinc function $w(\omega)$ has two effects (Fig. 2.a): it smooths $\mathcal{K}(\omega)$ and makes it periodic. The period is $2\pi/L$ and the smoothing corresponds to a truncation in the λ domain by a window of length $(2n+1)L$. Hence $\widehat{\mathcal{K}}(\omega) \approx \mathcal{K}(\omega)$ for $|\omega| \leq \pi/L$ (shaded area in Fig. 2.b) provided that (i) $\mathcal{K}(\omega)$ falls off quickly outside the range $|\omega| > \pi/L$, and (ii) $\kappa(\lambda)$ falls off quickly outside the range $|\lambda| > (n+1/2)L$. Correspondingly, the normalized error $\epsilon(x,y)$ is small for $|\log(y/x)| = |\omega| \leq \pi/L$.

For instance, as it can be seen in Fig. 1 and Fig. 2.d, the χ^2 kernel is easier to approximate than the intersection kernel, as it has fast fall off (exponential) in both domains λ and ω, while the intersection kernel fall off in the λ domain is slow (polynomial). An intuitive reason for this behaviour is that the χ^2 kernel is smoother than the intersection one.

Intrinsic limitations of a finite representation. Even if conditions (i) and (ii) are satisfied, the normalized error $\epsilon(x,y)$ may still be large (although bounded) for $|\omega| > \pi/L$ (Fig. 2.b). This corresponds to the case in which either x is much larger than y, or vice-versa. Fortunately, in these cases the *actual* error $\sqrt{xy}\,\epsilon(x,y)$ is still small (for instance the actual error for $x = 0$ is zero). In applications, it suffices to have a good approximation in a limited range of ratios $|\log(y/x)| = |\omega| \leq M$, outside which either y or x can effectively be considered null. This yields the condition

$\pi/L \geq M$, which is usually satisfied by a relatively coarse sampling step L due to the logarithmic dependency of M on the ratio y/x (Fig. 2).

Optimal approximation parameters. Given a kernel $k(x, y)$ to be approximated and the approximation parameters n and L, (11) can be used to quickly predict the maximum (or mean) error $\epsilon(x, y)$ for $\omega = \log(y/x)$ in a given range $[-M, +M]$. In Fig. 2 this is used to determine optimal approximation parameters for the χ^2 and intersection kernels for different choices of M.

5. γ-homogeneity and normalisation

In this section we first illustrate a generalization of the family of additive kernels discussed so far, and then consider the important issue of normalization.

We generalise (3) to the case $k(cx, cy) = c^\gamma k(x, y)$, where γ is a real parameter, and term this a *γ-homogeneous kernel* (setting $\gamma = 1$ yields an homogeneous kernel). For instance, the linear kernel $k(x, y) = xy$ is a 2-homogeneous kernel. Then (3) generalises to

$$k(x, y) = (xy)^{\frac{\gamma}{2}} k\left(\sqrt{\frac{y}{x}}, \sqrt{\frac{x}{y}}\right) = (xy)^{\frac{\gamma}{2}} \mathcal{K}\left(\log \frac{x}{y}\right). \tag{12}$$

It is possible to obtain a γ-homogeneous variant of any homogeneous kernel simply by plugging the corresponding signature into (12) (see examples in Fig. 1). Moreover, the approximations and error analysis discussed in Sect. 4 apply unchanged to the γ-homogeneous case as they work at the level of the signatures. Some practical advantages of using $\gamma \neq 1, 2$ are discussed in Sect. 7.

Normalization. Empirically, it has been observed that properly normalising a kernel $K(\mathbf{x}, \mathbf{y})$ may boost the recognition performance. A way to do so is to scale the histograms $\mathbf{x}$ so that $K(\mathbf{x}, \mathbf{x}) = 1$ for any $\mathbf{x}$. Then, as long as K is PD, one must also have $K(\mathbf{x}, \mathbf{x}) \geq |K(\mathbf{x}, \mathbf{y})|$, which encodes a simple consistency criterion: $\mathbf{x}$ should be the histogram most similar to itself [21].

For a γ-homogeneous kernel $k(x, y)$, (12) yields $k(x, x) = (xx)^{\frac{\gamma}{2}} \mathcal{K}(\log(x/x)) = x^\gamma \mathcal{K}(0)$, so that for the corresponding additive kernel (1) one has $K(\mathbf{x}, \mathbf{x}) = \sum_{b=1}^{B} k(\mathbf{x}_b, \mathbf{x}_b) = \|\mathbf{x}\|_\gamma^\gamma \mathcal{K}(0)$ where $\|\mathbf{x}\|_\gamma$ denotes the l^γ norm of the vector $\mathbf{x}$. Hence the normalisation condition $K(\mathbf{x}, \mathbf{x}) = 1$ 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.

6. Comparison with Maji and Berg method

Maji and Berg propose for the intersection kernel $k(x, y) = \min\{x, y\}$ the infinite dimensional feature map $\Psi(x)$ given by

$$[\Psi(x)]_\lambda = H(x - \lambda), \quad \lambda \geq 0$$

where $H(\lambda)$ denotes the Heaviside (step) function. In fact, as can be easily verified,

$$\min\{x, y\} = \int_0^{+\infty} H(x - \lambda) H(y - \lambda)\, d\lambda.$$

They also propose a n-dimensional approximation $\tilde{\Psi}(x)_j$ of the type $(1, \ldots, 1, a, 0, \ldots, 0)/\sqrt{n}$ (where $0 \leq a < 1$), approximating the step function by taking its average in n equally spaced intervals. This feature map results in 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 $\lfloor x \rfloor_n = \mathrm{floor}(nx)/n$. We refer to this approximation as MB, from the initials of the authors.

Approximation error and data scaling. It is interesting to compare the MB approximated intersection kernel $\tilde{k}(x, y)$ to the approximation $\widehat{k}(x, y)$ of Sect. 4. As shown in Fig. 3, the corresponding normalised errors $\tilde{\epsilon}(x, y) = (\tilde{k}(x, y) - k(x, y))/\sqrt{xy}$ and $\widehat{\epsilon}(x, y) = (\widehat{k}(x, y) - k(x, y))/\sqrt{xy}$ are qualitatively quite different. As predicted in Sect. 4, our error $\widehat{\epsilon}(x, y)$ depends only on the ratio $\omega = \log(y/x)$ and is distributed rather uniformly. In contrast, the MB error $\tilde{\epsilon}(x, y)$ is either zero or relatively large, and is particularly poor when $x, y < 1/n$ (where the approximation reduces to a linear kernel).

In applications, when $\tilde{k}(\mathbf{x}_b, \mathbf{y}_b)$ is plugged into (1) to approximate the additive intersection kernel, and when $\mathbf{x}, \mathbf{y}$ are high dimensional histograms, the components $\mathbf{x}_b, \mathbf{y}_b$ tend to have varied and small dynamic ranges, so that $\mathbf{x}_b, \mathbf{y}_b < 1/n$ frequently unless n is very large. While not discussed explicitly in [13], we found that in the actual MB implementation this issue is addressed by considering the adapted kernel $\tilde{k}_{\text{scaled}}(\mathbf{x}_b, \mathbf{y}_b) = \tilde{k}(R_b \mathbf{x}_b, R_b \mathbf{y}_b)/\sqrt{R_b}$, where R_b is estimated from the training data to fit the dynamic range of each histogram dimension. In the experiments we also use this rescaling for fairness.

Normalisation. As discussed in Sect. 5, assuring the proper normalisation of the kernel can be important in applications. With the approximations proposed in Sect. 4 we have from (9) and the definition (1) $\widehat{k}(x, x) = x \sum_{j=-n-1}^{n+1} L\kappa(jL) \propto x$, so that $\widehat{K}(\mathbf{x}, \mathbf{y}) = \sum_{b=1}^{B} \widehat{k}(\mathbf{x}_b, \mathbf{y}_b) \propto \|x\|_1$. Hence, if the data is scaled so that the exact kernel is properly normalised, i.e. $K(\mathbf{x}, \mathbf{x}) = 1$, then our approximation $\widehat{K}(\mathbf{x}, \mathbf{x})$ is also properly normalised (up to a constant and irrelevant scaling factor). By contrast, the MB approximation $\tilde{k}(x, y)$ does not satisfy this property.

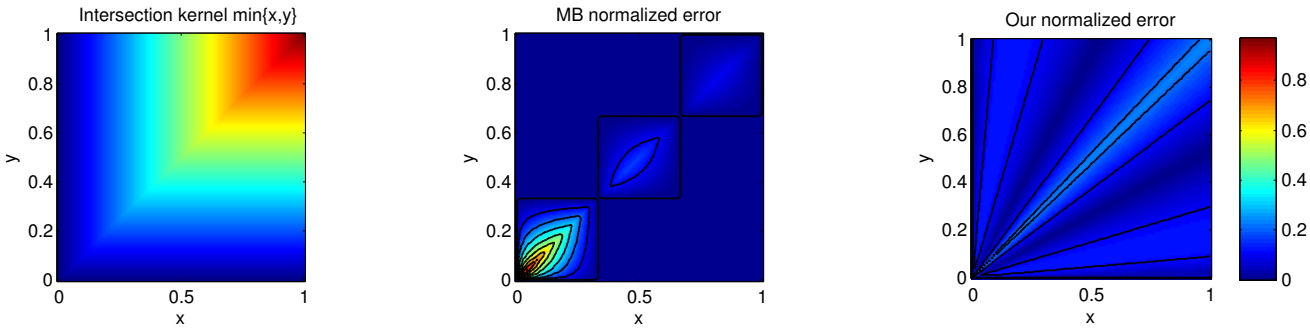

Figure 3: **Comparison of our and MB [13] approximations.** *Left*: The intersection kernel $k(x, y) = \min\{x, y\}$ for $x, y \in [0, 1]$. *Middle*: The normalised error $\tilde{\epsilon}(x, y)$ of the MB approximation [13] for a three dimensional approximated feature map $\tilde{\Psi}(x)$ (Sect. 6). The error is large if $x, y \leq 1/n$, where the approximation reduces to a linear kernel. *Right*: The normalised error $\widehat{\epsilon}(x, y)$ of our approximation $\widehat{\Psi}(x)$ (Sect. 4), also for the three dimensional case. The error depends only on the ratio y/x.

7. Experiments

The following experiments compare the exact χ^2, intersection, Hellinger's, and linear kernels to our approximations and the approximation of Maji and Berg [13] in term of accuracy and speed. Methods are evaluated on the same data sets used by [13]. In the DaimlerChrylser pedestrians comparison, we use exactly the same features as [13], so that the kernels and their approximations can be directly compared to the results in [13]. In the Caltech-101 experiments we use a stronger feature than that used in [13] so that results are more comparable to the state of the art, and again compare kernels and their approximations. In the INRIA pedestrian 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.

DaimlerChrysler pedestrians ([16], Fig. 4). The problem is to discriminate 18×36 patches portraying a pedestrian (positive samples) or clutter (negative samples). See [16] for details on the dataset and the evaluation protocol.

Patches are described by means of either one of two HOG-like [4] descriptors: MBHOG, the multi-scale HOG variant of [13, 15] (downloaded from the authors' website), and SHOG, our own implementation of a further simplified HOG variant which does not use the scale pyramid. Learning uses LIBLINEAR [5] as [13].

The SHOG features outperform the MBHOG features with all tested kernels. With the SHOG features, our approximations match the performance of the exact kernels with just three dimensions, performing better than the baseline linear and Hellinger's kernels, and marginally better than the MB approximation. With the MBHOG features, the same qualitative conclusions hold, but the differences are much more significant. Results are consistent with the ones reported in [13], except for the linear kernel, that worked much better for us. Finally, Fig. 4.d shows the significant impact of the choice of histogram normalisation on some kernels (especially the linear one).

Caltech-101 ([6], Fig. 5). The problem is to classify the 102 classes of the Caltech-101 benchmark dataset. Stronger image descriptors are used compared to those of Maji and Berg [13] to see if the homogeneous kernels can improve upon an already good representation (in the DaimlerChrysler experiment the advantage of the homogeneous kernels is limited with the stronger SHOG descriptors). Specifically, PHOW descriptors [2] are extracted to obtain 1200-dimensional visual word histograms with four spatial subdivisions [12]. Learning uses LIBLINEAR as before.

The linear kernel performs poorly. The Hellinger's kernel is better but outperformed by the χ^2 kernel. Our approximations do better that the MB approximation for the low dimensionality considered, and match the performance of the exact kernels. In a few cases, the approximations actually *outperform* the exact kernels, probably because of the additional smoothing caused by sampling the features (Sect. 4). The $\gamma = 1/2$ variants of the χ^2 and intersection kernel approximations perform better still, probably because they tend to reduce the effect of large peaks in the histograms. As expected, training speed is much better than for the exact kernels.

INRIA pedestrians ([4], Fig. 6). The experiment compares our low-dimensional χ^2 approximation to a linear kernel in learning a pedestrian detector for the INRIA benchmark [4]. Both the standard HOG descriptor (insensitive to the gradient direction) and the version by [7] (combining direction sensitive and insensitive gradients) are tested. Training

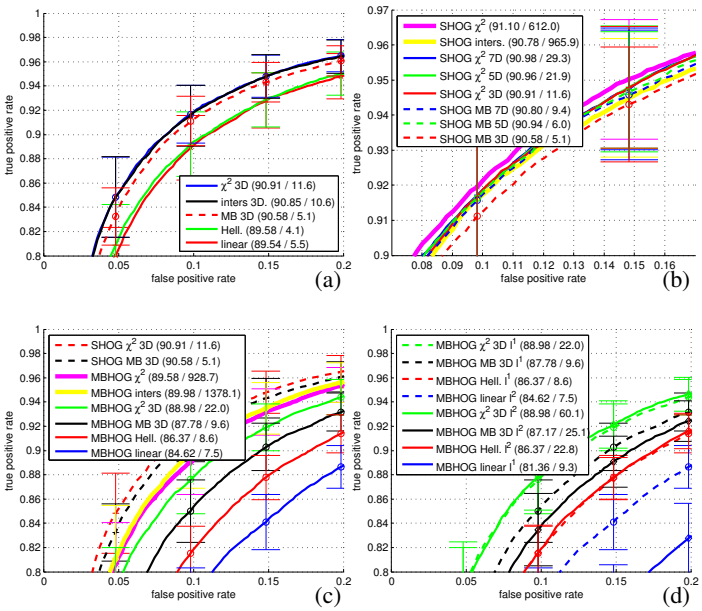

Figure 4: **ROC curves for the DaimlerChrysler dataset.** χ^2, *inters.*, *Hell.* and *linear* denote the exact χ^2, intersection, Hellinger's, and linear kernels; χ^2 nD, *inters.* nD, and *MB* nD denote our approximations and the MB approximation with n-dimensional feature maps. SHOG and MBHOG denote the two variants of HOG-like descriptors (see text). The legend reports the accuracy at equal error rate and the training time in seconds. *(a)*: With SHOG descriptors, all kernels do better than the baseline linear and Hellinger's kernels with just 3 components. Our approximations perform marginally better than the MB approximation. *(b)*: With SHOG features three components are sufficient to do as well as the exact kernels. *(c)*: The SHOG features outperform significantly the MBHOG features with all kernels. With the MBHOG features, however, the performance gap between the various kernels is more significant. *(d)*: Choosing the right histogram normalisation (either l^1 or l^2, depending on the kernel) affects significantly the MB and linear kernels, less the Hellinger's and approximated χ^2 ones.

linear kernel		Hellinger's kernel	
acc.	time	acc.	time
49.0±1.5	29.2±0.9	63.7±1.9	19.9±0.4

Figure 5: **Caltech-101 classification.** Average class accuracies are reported for 15 training images per class according to the standard protocol. *Top*: Exact linear and Hellinger's kernels; *Right*: Exact χ^2 and intersection kernels, our approximated feature maps, their 1/2-homogeneous variants, and the MB feature map. Our 3D feature maps already saturate at the performance of the exact kernels and are further improved by setting $\gamma = 1/2$.

		χ^2 kernel		inters. kernel	
mthd.	dm.	acc.	time	acc.	time
kernel	–	64.2±1.7	388.4±8.7	62.2±1.8	354.7±24.4
appr.	1	62.4±1.6	**20.7**±0.3	62.0±1.4	22.9±0.7
appr.	3	64.2±1.5	58.4±7.2	63.9±1.2	66.5±2.3
appr.	5	64.0±1.6	101.3±0.7	64.0±1.7	105.8±6.5
appr-γ	3	**65.8**±1.5	54.7±6.2	65.7±1.5	52.6±7.7
MB	1	–	–	55.9±0.9	26.9±0.8
MB	3	–	–	60.5±1.3	25.5±1.2
MB	5	–	–	61.3±1.1	22.1±3.3

uses a variant of the structured output framework proposed by [1] and the cutting plane algorithm by [11]. Compared to conventional SVM based detectors, for which negative detection windows must be determined through retraining [4], 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 [11] 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. 6, 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.* [7] with the χ^2 3D approximated kernel, whereas Ott and Everingham [17] reports 0.05 integrating HOG with image segmentation and using a quadratic kernel, Wang *et al.* [22] reports 0.02 integrating HOG with occlusion estimation and a texture descriptor, and Maji *et al.* [15] reports 0.1 using HOG with the exact intersection kernel (please refer to the corrected results in [14]).

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 [15] with the intersection kernel, and by [17] with the quadratic kernel, both exact.

Compared to Maji *et al.* [15], our technique also has an edge on the testing efficiency. [15] evaluates an additive 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 3D feature vectors, i.e. to do three multiplications. So typically $T_{\text{dot}} \ll T_{\text{look}}$, especially because fast convolution code using vectorised instructions can be used.

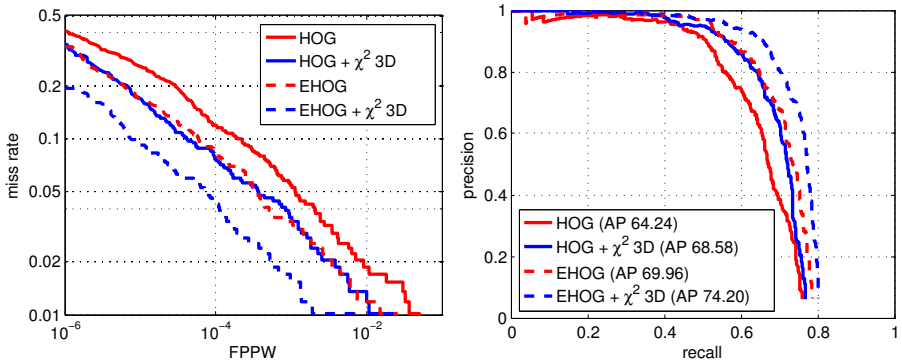

Figure 6: **INRIA dataset.** Evaluation of HOG based detectors learned in a structural SVM framework. DET [4] and PASCAL-style precision-recall [17] curves are reported for both HOG and an extended version [7], dubbed EHOG, detectors. In both cases the linear detectors are shown to be improved significantly by the addition of our approximated 3D χ^2 feature maps, despite the small dimensionality.

Summary

Supported by a novel theoretical analysis, we derived fast, closed form, and very low dimensional approximations of all common additive kernels, including the intersection and χ^2 kernels.

The approximations work as well as the exact kernels and better than Maji and Berg [13]'s approximation in the low dimensional regime. Empirically, the χ^2 kernel was shown to perform better than the intersection kernel, and to be easier to approximate. Note that the MB approximation applies only to the intersection kernel.

The approximations can be used to train kernelised models with algorithms optimised for the linear case, including standard SVM solvers such as LIBSVM [5], stochastic gradient algorithms, on-line algorithms, and cutting-plane algorithms for structural models [10]. Since our feature maps are so low dimensional, it is not necessary to use special encodings as in [13], which means that the algorithms apply unchanged. As linear algorithms scale linearly and the kernelised ones quadratically, the speedup grows linearly with the training set size. In particular, our technique was shown to be an enabling factor in structural training for a state-of-the-art pedestrian detector on the INRIA dataset.

Finally, we evaluated a γ-homogeneous variant of the homogeneous kernels that was shown to perform better than the standard kernels on some tasks.

Acknowledgements. 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] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Proc. ECCV*, 2008.

[2] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *Proc. CIVR*, 2007.

[3] 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.

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

[5] 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*, 9, 2008.

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

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

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

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

[10] T. Joachims. Training linear SVMs in linear time. In *Proc. KDD*, 2006.

[11] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1), 2009.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyound Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. CVPR*, 2006.

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

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

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

[16] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *PAMI*, 28(11), 2006.

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

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

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

[20] A. Vedaldi and B. Fulkerson. VLFeat library. `http://www.vlfeat.org/`, 2008.

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

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