Learning parameterized histogram kernels on the simplex manifold for image and action classification

Vitaly Ablavsky
CVLab, EPFL, Switzerland
vitaly.ablavsky@epfl.ch

Stan Sclaroff
Computer Science Dept., Boston University, USA
sclaroff@cs.bu.edu

Abstract

State-of-the-art image and action classification systems often employ vocabulary-based representations. The classification accuracy achieved with such vocabulary-based representations depends significantly on the chosen histogram-distance. In particular, when the decision function is a support-vector-machine (SVM), the classification accuracy depends on the chosen histogram kernel. In this paper we focus on smoothly-parameterized kernels in the space of histograms, such as, but not limited to, kernels that are derived from smoothly-parameterized histogram-distance functions. We learn parameters of histogram kernels so that the SVM accuracy is improved. This is accomplished by simultaneously maximizing the SVM's geometric margin and minimizing an estimate of its generalization error. We validate our approach on a previously-published two-class synthetic dataset and three real-world multi-class datasets: Oxford5K, KTH, and UCF. On these datasets our approach yields results that compare favorably to or exceed the state of the art.

1. Introduction

Vocabulary-based representations have shown to be effective for many computer vision problems such as action recognition, image classification and retrieval, etc. Vocabularies are typically computed by finding interest points in an image or a video volume, representing each interest point by a high-dimensional vector, called a *descriptor*, and clustering these descriptors. The resulting cluster-labels, sometimes called visual words, form the vocabulary. By mapping each interest point to its nearest cluster(s) [31] a training or test image/video can then be represented as a histogram, i.e., frequency counts, over the vocabulary of visual words. Many effective vocabulary-based representations have been proposed, ranging from a *bag-of-words* [9], i.e., frequency counts with no spatial information, to concatenation of histograms over spatial partitions in a pyramid [19], to histograms with respect to a hierarchy of vocabularies [17].

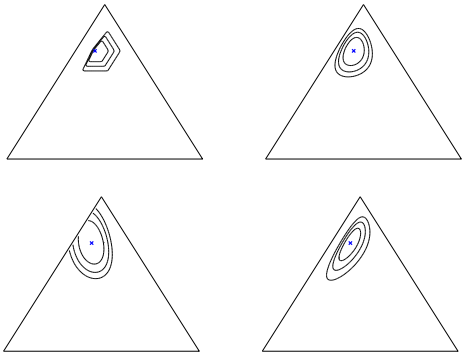


Figure 1. Iso-contours for several parameterized histogram-similarity/distance functions are shown on a 2-simplex; they are from left to right, top-to-bottom: weighted-intersection similarity with weights $(0.1, 0.8, 0.1)$, χ^2 distance, quadratic-chi distance [24] $\mathbf{A} = \mathbf{U}^T \mathrm{diag}(0.1, 0.8, 0.1)\mathbf{U}$ with $\mathbf{U}$ a random orthogonal matrix, and Fisher-information pullback metric [20] $\boldsymbol{\lambda} = (0.1, 0.8, 0.1)$. We learn histogram kernels derived from such functions by minimizing a bound on the SVM generalization error.

When using vocabulary-based representations, the choice of histogram-similarity/distance function can significantly impact the classification accuracy of a complete system. Indeed, the space of normalized histograms over a vocabulary is a *simplex* manifold, as noted in e.g., [20, 6]. Histogram-distance functions that exploit properties of this space have demonstrated their effectiveness for recognition tasks. For example, the χ^2 histogram-distance function attenuates the contribution of frequent bins, the earth-mover's distance (EMD) [28] and quadratic-chi distance [24] model cross-bin interactions, and the distances of [20, 6] are derived via *pullback* of a metric; Fig. 1 shows some examples.

Some classification algorithms, e.g., support vector machines (SVM's), require a histogram kernel rather than a distance function. As is the case with distances, kernels that are effective for Euclidean spaces are not necessarily effective in the histogram-space [4]. Instead, state-of-the-art classification results are obtained with SVM's in conjunction with e.g., histogram-intersection kernel [2, 19] or a χ^2 kernel [4, 17]. In practice, given a chosen histogram-

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

distance function one can construct a similarity function as suggested in [4], although it may not satisfy all conditions to be a kernel; the requirements for a distance function to induce a valid kernel for use in SVM's were derived in [13].

In this paper we focus on smoothly-parameterized kernels in the space of histograms. Such kernels may happen to be derived from smoothly-parameterized histogram-distance functions, but this connection is not required. We pose the problem of learning parameters of these histogram kernels in a way that improves SVM classification accuracy. Our formulation determines the kernel parameter settings that both maximize the geometric margin of the SVM and minimize the estimate of its generalization error. We validate our approach on a standard two-class synthetic dataset and three standard real-world multi-class datasets: Oxford5K, KTH, and UCF. On these datasets our approach yields results that compare favorably to or exceed the state of the art.

2. Related work

In this section we review relevant work on distance- and kernel-learning. The reviewed approaches are contrasted on the amount of supervision required and the optimization objectives.

Unsupervised distance learning. One example of such approaches is [20], which learned a parameterized distance function from unlabelled training data sampled from the simplex manifold. The parameters were adapted to maximize the metric's inverse volume element. In [33] it was noted that independent weighting of histogram bins in prior work, e.g., [20], might be too limiting for real-world problems. A mixture of Dirichlet distributions was fitted to the normalized histograms; the learned mixture coefficients were constrained to yield a valid bin-to-bin affinity matrix, which, in turn, defined an earth-mover's distance (EMD).

The above approaches are attractive when the training labels are incomplete or unavailable. However, when each training sample has its class label, it is desirable to incorporate this information into the metric-learning algorithm so that the metric is trained discriminatively.

(Semi)-Supervised distance learning. These approaches attempt to adjust the distance between each training sample and its nearest neighbors while taking side-information, e.g., similar/dissimilar labels, into account.

For example, in [7] a space of Mahalanobis distance functions was considered. The distance-learning algorithm searched for a distance function under which the distance between any pair of similar training samples was smaller than a threshold and a distance between any pair of dissimilar training samples was larger than a threshold.

A metric whose matrix representation is a product of a non-negative diagonal matrix and an unconstrained square matrix was studied in [30]. In this SVM-inspired formulation, the matrix-Frobenius norm was taken as the objective function to be minimized; same/different side information yielded distance inequality constraints.

The problem of learning an optimal Mahalanobis distance was revisited in [34]. The connection to an SVM formulation was established via the large-margin nearest-neighbor (LMNN) formulation, and constraints similar to those in [7] were imposed. The resulting optimization problem was treated as an instance of semi-definite programming.

In [3] the LMNN formulation was studied for the weighted histogram-intersection similarity function. The optimization minimized the squared l_2 norm of the weight vector subject to the same constraints as in [30, 7, 34].

The approaches mentioned thus far are designed to improve accuracy of nearest-neighbor classifiers. Yet, for image- and action- classification problems state-of-the-art accuracy tends to be attained by SVM's in conjunction with histogram kernels [31, 17]. Furthermore, these approaches tend to specialize to one family of metrics.

Multiple-kernel learning. Formulations that learn combinations of fixed kernels from a finite family, or combinations of kernels from a parameterized family have shown promise on some image- and action-recognition problems. For example, the vocabulary-learning approach to action recognition proposed in [17] employed the MKL approach of [1]. Learning histogram weights for image classification via *infinite*-kernel-learning was proposed in [10]. Learning multiplicative weights of kernels for image classification was proposed in [32]. We compare these MKL approaches with respect to three criteria: the applicable family of *base* kernels, the optimization objective, and the convexity of the resulting optimization problem.

The Support-Kernel-Machine (SKM) approach of [1] is applicable to a convex combination of a finite set of base kernels; the objective function is formulated as a maximization of the geometric margin of a hyperplane-based classifier; and the resulting optimization is provably convex. The infinite-kernel-learning approach of [10] was applied to combinations of diagonally-scaled Gaussian RBF kernels and to non-sparse linear combinations of χ^2 kernels; the objective is formulated as the maximization of the geometric margin of an SVM classifier, and the resulting optimization function is according to [10], "possibly complicated" in some parameters, necessitating a sampling-based optimization. The generalized-MKL approach of [32] was applied to the products of Gaussian RBF kernels; the objective is formulated as the maximization of the SVM geometric margin; and as noted in [32] "the price that one has to pay for such generality, is that the new MKL formulation is no longer convex."

Some histogram distance/kernel functions that are useful in practice, e.g., [33, 20, 24], do not readily fit the

Table 1. Learning parameterized metrics or kernels on $\mathbb{P}^{M-1}$ may improve the classification accuracy of histogram-based classifiers. Prior approaches might not have realized the benefit of such parameterizations, tend to specialize to one metric/kernel, and might not be applicable to a classifier that yields state-of-the-art classification accuracy. Our proposed approach is applicable to SVM classifiers and only requires that a histogram kernel on $\mathbb{P}^{M-1}$ varies smoothly with its parameters.

Classification approach	Classifier	Metric or kernel on $\mathbb{P}^{M-1}$	
		Type(s)	Parametric?
Lafferty et al. 2005 [18]	SVM	metric defined via a pullback of $\mathbb{P}^{M-1} \xrightarrow[g]{} \mathbb{S}_+^{M-1}$	no
Lebanon 2006 [20]	nearest-neighbor	pullback of $\mathbb{P}^{M-1} \xrightarrow[F_{\boldsymbol{\lambda}}]{} \mathbb{P}^{M-1} \xrightarrow[g]{} \mathbb{S}_+^{M-1}$	yes
Chaudhry et al. 2009 [6]	nearest-neighbor	pullback of $\mathbb{P}^{M-1} \xrightarrow[g]{} \mathbb{S}_+^{M-1}$, or histogram intersect., or χ^2	no
Cai et al. 2010 [3]	nearest-neighbor	weighted histogram intersection	yes
Proposed approach	SVM	any smoothly-parameterized kernel on $\mathbb{P}^{M-1}$	yes

provably-convex MKL formulations, e.g., [1]. Although a non-convex MKL formulation, e.g., [32, 10] might be considered, the benefits for histogram kernels remain unclear.

Our proposed approach is surprisingly simple yet applicable to histogram kernels. Its optimization objective is guided by any estimate of the SVM generalization error, and it may be used to simultaneously learn the SVM regularization parameter. The advantages of our approach are summarized in Table 1, and the relevant metrics on the simplex manifold are defined in the next section.

3. Approach

We choose a representation for still images and video clips that is based on a vocabulary of visual words. We assume that such a vocabulary has already been learned, and that each still image or a video clip is represented via a histogram with M bins. These histograms may arise from a bag-of-words representation, in which case M would equal the number of words in the vocabulary, a spatial pyramid [19], a hierarchy of vocabularies [17], etc.

Metrics in the space of normalized histograms. Since a number of useful histogram-distance functions, such as χ^2, and also [12, 28, 20], are defined on normalized histograms we discuss metrics in this space.

Let $\mathbf{x} \in \mathbb{R}_+^M$ be a normalized histogram. The set of all such histograms is a simplex $\mathbb{P}^{M-1}$,

$$\mathbb{P}^{M-1} = \left\{ x_1, \ldots, x_M \,\middle|\, \sum\nolimits_{m=1}^{M} x_m = 1,\ x_m \geq 0 \right\}, \tag{1}$$

which is a smooth manifold with corners [21]. A metric on $\mathbb{P}^{M-1}$ may be inherited from $\mathbb{R}^M$; defined explicitly via the metric tensor; defined implicitly via a distance function; or defined via a *pullback* of the metric on another manifold related to $\mathbb{P}^{M-1}$ via a *diffeomorphism*.

For example, [18, 6] defined the distances on $\mathbb{P}^{M-1}$ as a pullback of a mapping g, where $g : \mathbf{x} \mapsto (\sqrt{x_1}, \ldots, \sqrt{x_M})$. A composition of pullbacks was defined in [20] with respect to $g \circ F_{\boldsymbol{\lambda}}$, where $F_{\boldsymbol{\lambda}}$ acts on $\mathbf{x}$ as $x_m \mapsto x_m \lambda_m / \langle \mathbf{x}, \boldsymbol{\lambda} \rangle$, for $m = 1, \ldots, M$, and where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^M$.

In some cases the metric on $\mathbb{P}^{M-1}$ might be parameterized by $\boldsymbol{\lambda}$ in such a way that the resulting distance or kernel functions vary smoothly with $\boldsymbol{\lambda}$. The benefit of a parameterized metric on $\mathbb{P}^{M-1}$ might then be realized to improve the accuracy of histogram-based classifiers. However, as Table 1 indicates, prior approaches might not have always taken advantage of parameterization, and approaches that did learn the parameters tended to specialize to one metric or kernel. Our proposed approach is applicable to a larger family of kernels on $\mathbb{P}^{M-1}$ and is designed to work with SVM classifiers; SVM classifiers tend to yield state-of-the-art accuracy on challenging action-recognition tasks.

Problem Definition. Let $\mathbf{x} \in \mathbb{R}_+^M$ be a histogram[1]. In order to make predictions of class labels given $\mathbf{x}$ using SVM classifiers we define a kernel function on all pairs $(\mathbf{x}_1, \mathbf{x}_2)$.

We are particularly interested in kernels that are differentiable with respect to parameters $\boldsymbol{\lambda}$. This is the case when a kernel is derived from a histogram-similarity/distance function that itself is smoothly parameterized by $\boldsymbol{\lambda}$. For example, the *information-diffusion* kernel for the simplex [18] is based on the parameterized distance in Eq. 6 and thus depends smoothly on $\boldsymbol{\lambda}$.

For a training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, comprising histograms $\mathbf{x}_n$ and class labels y_n the SVM algorithm will learn a decision function $y = f(\mathbf{x}; \boldsymbol{\lambda})$ for a given $\boldsymbol{\lambda}$. Different choices of $\boldsymbol{\lambda}$ will, in general, yield a different decision function on the space of histograms. Our goal is to find $\boldsymbol{\lambda}^*$ such that $f(\mathbf{x}; \boldsymbol{\lambda}^*)$ has the lowest generalization error.

Basic Formulation. We assume that our kernel is differentiable with respect to parameters $\boldsymbol{\lambda}$ and write it as $k(\mathbf{x}_1, \mathbf{x}_2; \boldsymbol{\lambda})$. Given a histogram $\mathbf{x}$ we will decide its label y, say $y \in \{-1, 1\}$, using an SVM. Thus, we consider decisions functions of the form $y = \operatorname{sign} f(\mathbf{x}; \boldsymbol{\lambda})$, where

$$f(\mathbf{x}; \boldsymbol{\lambda}) = \sum_{n=1}^{N} \alpha_n k(\mathbf{x}, \mathbf{x}_n; \boldsymbol{\lambda}) + b. \tag{2}$$

[1] Depending on the kernel, $\mathbf{x}$ may be normalized, i.e. sum to one, and/or may have strictly-positive bins.

If misclassifications of the training examples are penalized via *hinge-loss*, then $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ is obtained via

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & W(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n \\ & -\tfrac{1}{2} \sum_{n,n'=1}^{N} y_n y_{n'} \alpha_n \alpha_{n'} k(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\lambda}) \quad (3) \\ \text{subject to} \quad & \sum_{n=1}^{N} y_n \alpha_n = 0\,, \quad 0 \leq \alpha_n \leq C\,, \end{aligned}$$

where $C > 0$ is the regularization parameter. Since the objective function W in Eq. 3 depends on $\boldsymbol{\lambda}$ we rewrite it as $W(\boldsymbol{\alpha}; \boldsymbol{\lambda})$, and since the solution $\boldsymbol{\alpha}^*$ depends on $\boldsymbol{\lambda}$ we rewrite it as $\boldsymbol{\alpha}^*(\boldsymbol{\lambda})$.

Given the training set $\{\mathbf{x}_n, y_n\}_{n=1}^N$ we want to learn $\boldsymbol{\lambda}^*$ such that $f(\mathbf{x}; \boldsymbol{\lambda}^*)$ yields the lowest test or generalization error. One way to estimate the generalization error of f is by computing a cross-validation error on the training set. For example, the leave-one-out (LOO) error is often used as an unbiased estimator of the generalization error.

Let $T(\boldsymbol{\alpha}(\boldsymbol{\lambda}))$ be an estimate of f's generalization error. The function T depends on the training set, which is fixed, and on the weights $\boldsymbol{\alpha}$, which are themselves a function of $\boldsymbol{\lambda}$. When T is evaluated at $\boldsymbol{\alpha}^*$, the solution of Eq. 3, the generalization error is decreased by changing $\boldsymbol{\lambda}$ along the negative gradient of T. Therefore, we learn $\boldsymbol{\lambda}^*$ using [5]:

Algorithm 1: Learning $\boldsymbol{\lambda}$

Starting with an initial guess for $\boldsymbol{\lambda}$ iterate:

1. given $\boldsymbol{\lambda}$ solve the SVM problem $\boldsymbol{\alpha}^* = \max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}; \boldsymbol{\lambda})$
2. given $\boldsymbol{\alpha}^*(\boldsymbol{\lambda})$ take a step along $-\nabla_{\boldsymbol{\lambda}} T(\boldsymbol{\alpha}(\boldsymbol{\lambda}))|_{\boldsymbol{\alpha}^*}$

$\boldsymbol{\lambda}^* \leftarrow \boldsymbol{\lambda}$ at convergence

An advantage of Alg. 1 is that the optimal SVM regularization parameter C may be learned simultaneously with $\boldsymbol{\lambda}$. By contrast, the LMNN approaches, e.g., [3] or MKL approaches like [1, 10], require that C be specified.

In Alg. 1, T may be extended with a $\boldsymbol{\lambda}$-regularization term; alternatively early-stopping may be used to avoid over-fitting. The analysis of relative benefits of regularization with respect to the amount of available training data is left for future work.

One can (inefficiently) obtain an SVM solution via repeated kernel-matrix inversion; thus, the computational complexity of step 1 in the above algorithm is $O(N^3)$. The computational complexity of the second step is linear in M, the number of histogram bins. It turns out that computing derivatives with respect to α the naive way requires kernel-matrix inversion. Therefore, the overall complexity of step 2 is $O(N^3 M)$, although in practice SVM solvers are more efficient and the computational complexity of estimating the gradient can be reduced [16].

Optimization objective. The choice of objective function $T(\boldsymbol{\alpha}(\boldsymbol{\lambda}))$ has a significant impact on the $\boldsymbol{\lambda}$-learning algorithm. First, decreasing T should lower f's generalization error. Second, evaluation of the gradient vector of T must be computationally tractable. Third, T must be differentiable with respect to $\boldsymbol{\lambda}$. In this paper we consider two choices of T, the *span estimate* and the maximum-likelihood.

The *span bound* of [5] is an upper bound on the leave-one-out error. Given an SVM solution in the form of Eq. 2, training examples for which $\alpha \neq 0$, i.e., *support vectors*, are identified. The minimum distance from a support vector to a polytope, defined by a constrained linear combination of the remaining support vectors, is called the *span* of this support vector. Although support-vector spans can be used to define an upper bound on the generalization error, the bound turns out to be loose [5].

Under some assumptions, a *span estimate* can be defined. It was demonstrated in [5] that the span estimate is an accurate estimate of the leave-one-out error. For $O(N)$ support vectors, computing a single support-vector span has $O(N^3)$ computational complexity; therefore, a brute-force computation of the span estimate is $O(N^4)$, no better than the explicit LOO procedure. Fortunately, the assumption used to define the span estimate yields an $O(N^3)$ algorithm. One difficulty with using the span estimate in gradient-based optimization is that T becomes discontinuous with respect to $\boldsymbol{\lambda}$ when the set of support vector changes. A smoothed approximation is proposed in [5], but requires an additional tuning parameter to be set.

The SVM-max-likelihood of [11] is neither a bound on nor an estimate of the leave-one-out error, but a heuristically-derived objective function. It is defined as $\mathcal{L}(\boldsymbol{\lambda}; f(\cdot; \boldsymbol{\lambda}), \{(\mathbf{x}_{n'}, y_{n'})\})$, where $f(\cdot; \boldsymbol{\lambda})$ is from step 1 of Algorithm 1, $\{(\mathbf{x}_{n'}, y_{n'})\}$ is a (randomly-chosen) validation subset of the training set, and $\mathcal{L}$ is the negative of Platt's log-likelihood [26]. Experimental evaluation on non-computer-vision problems, [11] showed that minimizing T based on the sum of $\mathcal{L}$'s yielded better generalization than minimizing T based on the span-estimate; this trend was confirmed on the majority of data sets considered.

4. Implementation details

We now present details of our experimental setup.

Kernels. We evaluate two kernels in our system: the weighted-intersection kernel

$$k(\mathbf{x}_1, \mathbf{x}_2; \boldsymbol{\lambda}) = \sum_{m=1}^{M} \lambda_m \min(x_{1,m}, x_{2,m}),\ \boldsymbol{\lambda} \in \mathbb{R}_+^M \quad (4)$$

since it applies to normalized and unnormalized histograms,

and the *information-diffusion* kernel on the simplex

$$k(\mathbf{x}_1, \mathbf{x}_2; \gamma, \boldsymbol{\lambda}) = (2\sqrt{\pi}\gamma)^{-M} \exp\left\{-\frac{1}{\gamma^2} d^2_{\boldsymbol{\lambda}}(\mathbf{x}_1, \mathbf{x}_2)\right\} \tag{5}$$

with $d_{\boldsymbol{\lambda}}$ defined as

$$d_{\boldsymbol{\lambda}}(\mathbf{x}_1, \mathbf{x}_2) = \operatorname{acos} \sum_{m=1}^{M} \lambda_m \frac{\sqrt{x_{1,m}\, x_{2,m}}}{\sqrt{\langle \mathbf{x}_1, \lambda \rangle \langle \mathbf{x}_2, \lambda \rangle}}. \tag{6}$$

The information-diffusion kernel has been shown to yield high accuracy on vocabulary-based text-classification tasks [18, 20]. For a given $\mathbf{x}_1, \mathbf{x}_2$ each kernel varies smoothly with $\boldsymbol{\lambda}$.

Software used and parameter settings. We use `Shark` [14] to learn hinge-loss SVM's and the publicly-available framework [11] for experimenting with generalization bounds and estimates. As is standard practice [5, 16, 11] we initialize the SVM regularization parameter to $C = 1$ and learn it jointly with kernel parameters. As is standard practice [19, 17] for multi-class problems, we learn one-versus-all SVM classifiers.

For consistency with [11] we use the same gradient-based optimization algorithm, iRprop+ [15] with the same parameter settings. This first-order optimization algorithm is designed to increase the step size in the plateau regions of the objective functions, which in some cases requires fewer steps to reach the local optimum. However, in our experiments we have found this design choice made it harder to monitor convergence. We therefore limit the number of iterations to four, which is less than the 10-20 iterations used in [16] that relied on BFGS. The issue of convergence and the number of iterations is also related to regularization, as noted in [32], and we leave this for future work.

To convert our constrained optimization problem into an unconstrained one, we parameterize $\boldsymbol{\lambda}$ for the iRprop+ algorithm via $\mathbf{t} \in \mathbb{R}^M$: $\lambda_m = t_m^2$ for Eq. 4 and $\lambda_m = \exp(t_m)$ for Eq. 5, $m = 1, \ldots, M$. As is standard practice , e.g., [10, 17], we do not learn γ, but instead set it to the average distance between the training samples.

Except for the different choices of the kernel function and the optimization objective we conduct all experiments in Sec. 5 with the same parameter settings.

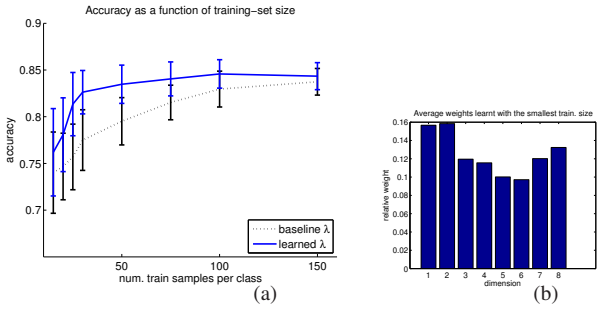
5. Experiments

For each dataset we first compute our *baseline* accuracy, obtained by setting $\lambda_m = 1$, $m \in 1, \ldots, M$. This baseline $\boldsymbol{\lambda}$ serves as initialization for $\boldsymbol{\lambda}$-learning using our two chosen objective functions: the span estimate and SVM-max-likelihood. In the tables of experimental results shown below, these objective functions are abbreviated as "span-est." and "max-lik." As is common practice, e.g., [17] for multi-class problems we report mean per-class accuracy.

The improvements reported with some published kernel-learning approaches tend to be incremental, e.g., [10, 17], and the statistical significance of such improvements might not always be clear. To guide the interpretation of our results we adopt the methodology of [8]. The recommendations in [8] include the use of the Wilcoxon signed-rank test with the p-value ≤ 0.05, to assess the statistical significance of a classifier's improvement over a baseline.

5.1. Experiments with a synthetic dataset

We validate our implementation on a binary classification task where the desired $\boldsymbol{\lambda}^*$ is known. To accomplish this we use the experimental protocol of [3] Sec. 5 by drawing eight-dimensional normalized histograms from the specified distribution; only the first two bins are informative. As can be seen from Fig. 2a, weights learned via the proposed approach consistently lead to higher accuracy; all improvements are statistically significant with $p \leq 0.004$. Furthermore, on average, the learned weights assign greater importance to the first two dimensions Fig. 2b.



(a) (b)

Figure 2. (a,b) Experimental results on the synthetic problem of [3]. Our approach consistently yields higher accuracy, and the learned $\boldsymbol{\lambda}$ correctly emphasizes the first two histogram bins.

5.2. Experiments with Oxford dataset

In this section we demonstrate the effectiveness of our approach on a standard synthetic dataset [3] and three standard real-world datasets [25, 29, 27].

This dataset, first presented in [25], comprises 5,062 images obtained using text-based web-search. The web-queries asked for 11 buildings at Oxford. Each retrieved image was manually labelled according to a subjective measure of how much of the queried building was visible. The set of labels was: *Good*, if the image contained a "nice, clear picture of the object/building", *OK*, if "more than 25% of the object was clearly visible," *Junk*, if "less than 25% of the object is visible, or there is a very high level of occlusion," and *Absent*, if "the object is not present."

In [25], images labelled *Junk* were treated "as though they are not present in the database." In training their retrieval system [25] used images labelled *Absent* as negative examples. The focus of this paper is on classification, rather than retrieval, therefore we use this dataset as follows.

Figure 3. Left: examples of *Good* and *OK* Oxford images used in our five-way classification task. Right: example actions from the KTH(a,b) and UCF(c,d) datasets — (a) boxing, (b) running, (c) kicking, and (d) lifting.

To formulate a multi-class classification problem we do not consider images labelled either *Junk* or *Absent*. We select classes with at least 50 examples so that train/validation/test splits can be chosen meaningfully. Examples of images from our multi-class classification problem are shown in Fig. 3 with *Good* images in the top row and *OK* images in the bottom row.

In [25] for each image affine-invariant Hessian regions are identified, and each such region is represented with a SIFT descriptor. From these descriptors a vocabulary with $M = 1,000,000$ words is learnt. The resulting encodings of each image are publicly-available [2].

It is known that training of SVM's with histograms over large vocabularies is computationally-expensive. Recently [23] proposed an efficient way to compute exactly and a practical way to approximate weighted-intersection kernels; this extension is left for future work. For the purpose of demonstration we employ a standard technique from the statistical-text-analysis community to reduce the vocabulary size. Specifically, we rank visual words according to their mutual information with class labels and retain the top 1,000; for this we employ a publicly-available tool `rainbow` [3].

Table. 2 summarizes our results on the Oxford dataset. In all experiments the overall accuracy has improved. Parameter-learning using the maximum-likelihood formulation in conjunction with the weighted-intersection kernel yields the largest improvement: 90.26% from the 89.23% baseline. Learning the information-diffusion kernel both improved accuracy with respect to its baseline and resulted in lower variance with respect to random test partitions. The improvements using the span-estimate objective are statistically-significant with $p = 0.01$ for the weighted-intersection kernel and with $p = 0.04$ for the information-diffusion kernel; for the max-lik objective the corresponding p-values are 0.12 and 0.23, which are less conclusive.

[2] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

[3] http://www.cs.umass.edu/~mccallum/bow

We also apply the publicly-available implementation of [31] to the Oxford dataset. To do this we treat the weighted-intersection kernel as a linear combination of base kernels, one for each histogram bin. Since [31] requires the SVM regularization parameter C to be specified manually, we set C to be the same as for our approach. Such a choice is made not to put GMKL of [31] at a disadvantage, but to ensure that both GMKL and our approach start with the same baseline mean per-class accuracy of 89.32%. Running GMKL with l_1 regularization ends up reducing the accuracy from 89.32% to $68.40\% \pm 5.95$. Running GMKL with l_2 regularization fares better overall than the l_1 regularization, but the accuracy does not improve: it changes from 89.32% to $89.22\% \pm 3.67$. Thus, for the Oxford dataset GMKL might not realize the benefit of learning a histogram kernel, while our approach is both conceptually simpler and tends to yield an improvement in the per-class accuracy.

Table 2. Results on the Oxford dataset. Improvements marked with $*$ are statistically significant with $p < 0.05$.

weighted intersection kernel		
baseline	89.23 ± 3.42	
	span-est.	max-lik.
learned $\boldsymbol{\lambda}$	90.26 ± 3.6 *	90.52 ± 3.19
information diffusion kernel		
baseline	89.23 ± 2.07	
	span-est.	max-lik.
learned $\boldsymbol{\lambda}$	90.12 ± 1.99 *	89.66 ± 2.22

5.3. Experiments with KTH dataset

This dataset was presented in [29]. It comprises six activities performed in four different conditions by twenty-five human subjects; thus there are 600 video sequences. Examples of "boxing" and "running" actions are shown in Fig. 3. Some of the challenges in this data set stem from variability in the subjects' clothing, camera jitter, and

zooming-in/out effects.

We follow the experimental protocol of [17], who use "standard partition following [29]". The partition information and level-0 histograms were kindly provided to us by the authors of [17], however their histograms based on space-time feature hierarchies were not made available. Each video was represented by a normalized histogram of optical flow (HoF) and a normalized histogram of oriented gradients (HoG). For the weighted-intersection kernel we concatenated HoF and HoG. Since the information-diffusion kernel requires normalized histograms we only used HoF.

Table 3 summarizes the results achieved by our method versus past methods for the KTH dataset. Results using the weighted intersection kernel and the information-diffusion kernel are reported. The weighted intersection kernel, when trained using the maximum-likelihood formulation of Algorithm 1 attains 95.83% accuracy; this improves from the baseline. Learning kernel parameters $\boldsymbol{\lambda}$ using the span-estimate also improves over the baseline, yielding 95.37% accuracy for weighted intersection kernel. (Stated differently, the error rate is reduced by 18%.) These results represent an improvement over the previous methods listed in the table. The span-estimate and maximum-likelihood formulations also yield improved accuracy over the baseline for the information-diffusion kernel; however, the information-diffusion kernel does not perform better (overall) than the weighted intersection kernel with this dataset.

Table 3. Results on the KTH dataset.

Approach	Year	Accuracy
Schüldt et al. [29]	2004	71.72%
Laptev et al. CVPR	2008	91.80%
Bregonzio et al. CVPR	2009	93.17%
Liu et al. CVPR	2009	93.80%
Gilbert et al. ICCV	2009	94.50%
Kovashka et al. [17]	2010	94.53%
weighted intersection kernel		
our baseline		94.90%
		span-est. max-lik.
our learned $\boldsymbol{\lambda}$		95.37% 95.83%
information diffusion kernel		
our baseline		92.59%
		span-est. max-lik.
our learned $\boldsymbol{\lambda}$		92.59% 93.51%

For completeness, we should also mention the results in [22] where 96.0% accuracy is reported using the protocol of [29]. However these results were obtained using a special protocol that included "spacetime alignment (location and frame cropping)" and then selecting 32 frames for each video sequence. Our method, using Algorithm 1 with the SVM maximum-likelihood objective function and the weighted intersection kernel attains comparable accuracy (95.83%), without the need for such specialized preprocessing.

5.4. Experiments with UCF dataset

This dataset was presented in [27]. It comprises ten actions with a total of 150 video sequences. Examples of "kicking" and "lifting" actions are shown in Fig. 3. The subjects' actions are unscripted, and the resulting videos exhibit variability in action-styles, background, imaging artifacts such as motion blur, etc.

The level-0 histograms for this data set were kindly provided by [17], however their histograms based on space-time feature hierarchies were not made available. We follow the leave-one-out experimental protocol of [17], which follows Wang et al. BMVC 2009.

Table. 4 summarizes our results with the weighted-intersection kernel. For both $\boldsymbol{\lambda}$-learning formulations, maximum-likelihood and span-estimate, an improvement with respect to the baseline of 81.75% is observed. Learning with the span-estimate formulation improves the accuracy to 86.27%. This makes our results comparable to the state of the art reported in [17]. The relative improvement with respect to our baseline is 5.53%; this compares favorably to the relative improvement of 2.08% in [17].

Table 4. Results on the UCF dataset.

Approach	Year	Accuracy
Wang et al. BMVC	2009	85.6%
Kovashka et al. CVPR	2010	
baseline		85.49%
learned		87.27%
our baseline		81.75%
		span-est. max-lik.
our learned $\boldsymbol{\lambda}$		86.27% 84.38%

Summary of experiments. In the experiments, our approach achieves classification accuracy that compares favorably to or exceeds the state of the art. All experiments are conducted with identical parameter settings. On the synthetic dataset our approach learns the correct weights; on the Oxford dataset our improvements are statistically significant; on the KTH and the UCF datasets our results compare favorably to [17] even though we only use the level-0 vocabularies, whereas [17] relies on a learned hierarchy of space-time features.

6. Conclusion

This work has focused on improving the accuracy of SVM classifiers used in vocabulary-based image- and action-recognition systems. Our learning formulation determines the kernel parameter settings that both maximize the

geometric margin of the SVM and minimize the estimate of its generalization error. In future work, we look forward to experiments with additional histogram kernels, e.g., [24]. We would like to develop efficient optimization strategies, in the spirit of e.g., [16] that are specific for histogram kernels derived from commonly-used histogram-distance functions.

7. Acknowledgments

This research was supported in part by US NSF grants IIS-0713168, IIS-0910908, and CNS-0855065. We thank Adriana Kovashka for sharing the video features from [17].

References

[1] F. Bach, G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, 2004.

[2] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *ICIP*, 2003.

[3] H. Cai, F. Yan, and K. Mikolajczyk. Learning weights for codebook in image classification and retrieval. In *CVPR*, 2010.

[4] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10(5), 1999.

[5] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46, 2002.

[6] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[8] J. Demšar. Statistical comparison of classifiers over multiple data sets. *JMLR*, 7, 2007.

[9] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[10] P. V. Gehler and S. Nowozin. Let the kernel figure it out: Principled learning of pre-processing for kernel classifiers. In *CVPR*, 2009.

[11] T. Glasmachers and C. Igel. Maximum likelihood model selection for 1-norm soft margin SVMs with multiple parameters. *PAMI*, 32(8), 2010.

[12] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *PAMI*, 17(7), 1995.

[13] M. Hein, O. Bousquet, and B. Schölkopf. Maximal margin classification for metric spaces. *Journal of Computer and System Sciences*, 71, 2005.

[14] C. Igel, V. Heidrich-Meisner, and T. Glasmachers. Shark. *Journal of machine learning research*, 9, 2008.

[15] C. Igel and M. Husken. Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing*, 50, 2003.

[16] S. Keerthi, V. Sindhwani, and O. Chapelle. An efficient method for gradient-based adaptation of hyperparameters in SVM models. In *NIPS*, 2006.

[17] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[18] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *JLMR*, 6, 2005.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[20] G. Lebanon. Metric learning for text documents. *PAMI*, 28(4), 2006.

[21] J. M. Lee. *Introduction to smooth manifolds*. Springer, 2002.

[22] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR*, 2010.

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

[24] O. Pele and M. Werman. The quadratic-chi histogram distance family. In *ECCV*, 2010.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[26] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances In Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[27] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH: A spatiotemporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[28] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a metric for image retrieval. *IJCV*, 40(2), 2000.

[29] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[30] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004.

[31] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *PAMI*, 32(7), 2010.

[32] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *ICML*, 2009.

[33] H.-Y. Wang, H. Zha, and H. Qin. Dirichlet aggregation: Unsupervised learning towards an optimal metric for proportional data. In *ICML*, 2007.

[34] K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *ICML*, 2008.