Per-Sample Kernel Adaptation for Visual Recognition and Grouping

Borislav Antic and Björn Ommer
Heidelberg Collaboratory for Image Processing
IWR, Heidelberg University, Germany
borislav.antic@iwr.uni-heidelberg.de, ommer@uni-heidelberg.de

Abstract

Object, action, or scene representations that are corrupted by noise significantly impair the performance of visual recognition. Typically, partial occlusion, clutter, or excessive articulation affects only a subset of all feature dimensions and, most importantly, different dimensions are corrupted in different samples. Nevertheless, the common approach to this problem in feature selection and kernel methods is to down-weight or eliminate entire training samples or the same dimensions of all samples. Thus, valuable signal is lost, resulting in suboptimal classification.

Our goal is, therefore, to adjust the contribution of individual feature dimensions when comparing any two samples and computing their similarity. Consequently, per-sample selection of informative dimensions is directly integrated into kernel computation. The interrelated problems of learning the parameters of a kernel classifier and determining the informative components of each sample are then addressed in a joint objective function. The approach can be integrated into the learning stage of any kernel-based visual recognition problem and it does not affect the computational performance in the retrieval phase. Experiments on diverse challenges of action recognition in videos and indoor scene classification show the general applicability of the approach and its ability to improve learning of visual representations.

1. Introduction

Visual recognition is one of the central problems of computer vision. It involves detection of objects and their localization in images [9], classification of actions and complex activities [22, 1], scene recognition [25], and related problems. Over the last decade the field has seen a tremendous increase in performance. Among the many advancements, two broad directions, which have boosted performance significantly, are standing out. First, there has been a lot of work on improving feature descriptors, interest points, and the like. As a result, models of image and video content have become significantly richer over the years, while constantly increasing their dimensionality. Influential work such as SIFT [20], HoG [7], video descriptors like HoF [16] and dense trajectories [28], or most recently the filters created by deep convolutional neural networks [15] have laid the basis for many other approaches. A second broad theme have been part-based models that integrate the content of many local features and the spatial layout (e.g., geometry, shape) of the class of interest. Popular models range all the way from simple bag-of-features to object representations with rich spatial structure such as deformable part models (DPM) [9].

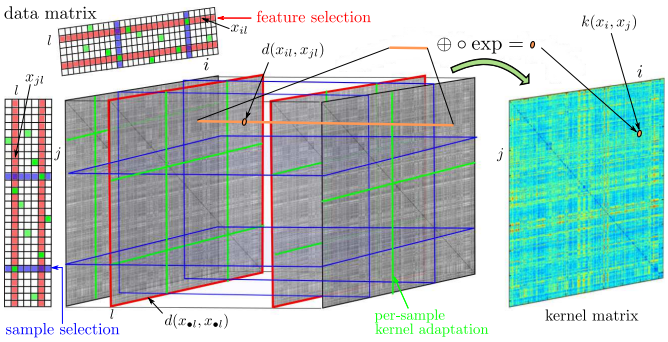

Figure 1: Computing a kernel for a given data matrix. Red are components affected by feature selection, blue are samples selected by SVM, and green are individual noisy components suppressed by our approach. Note that per-sample kernel adaptation only affects individual noisy comparisons, whereas red and blue manipulate entire feature or sample planes.

Although features and the object category models that integrate them have been significantly improved over the years, the currently popular theme of utilizing semi-local, high-dimensional parts creates issues of its own. Due to their spatial extent and high dimensionality these descriptors are likely to contain noisy feature dimensions caused by partial occlusion or clutter. Moreover, the dimensions that are affected typically vary between instances of a visual category, e.g., since different areas of an object may be corrupted. Nevertheless, typical solutions to this prob-

lem are feature selection [12, 3], where the same feature dimensions are eliminated from all samples, or the removal or down-weighting of entire training feature vectors as in SVM, see Fig. 1. So when facing *individual* noisy, unreliable feature dimensions, a lot of valuable information in the other meaningful dimensions is lost by affecting *entire* feature dimensions or samples.

Given a category representation, our goal is thus to eliminate the noisy components of each training sample, while retaining the complete signal. Simply suppressing noisy features per sample [30] would not solve the problem: Recognition requires comparisons between instances, for example to find the closest category to a sample. However, comparing two samples, which have different feature dimensions set to zero, yields flawed similarities. Thus, we need to down-weight the contribution of the individual feature dimensions while computing similarities. This integrates per-sample selection of reliable dimensions directly into a kernel computation. The challenge is then to learn the parameters that specify the kernel classifier, while determining the reliable components of each sample that in turn yield the kernel.

Existing per-sample feature selection (PSFS) methods [11, 31, 19, 4] work only with additive kernels (e.g. linear kernel). They represent a kernel as a weighted sum of other kernel functions, which measure similarity of individual feature vector dimensions. However, in many vision problems the non-additive kernel functions such as radial basis function (RBF) [33, 16], due to their non-linear coupling between individual feature dimensions, demonstrated better performance than the additive kernel functions that decouple individual feature dimensions. Despite their superior performance, non-additive kernel functions have not yet been covered by the existing methods for PSFS. Simply applying PSFS to an additive kernel and wrapping a non-linear function around the result would yield a completely new optimization problem that cannot be solved by existing PSFS techniques. Therefore, we generalize PSFS to the non-additive kernel functions, and propose an optimization method which directly finds and eliminates individual noisy feature distances between pairs of samples inside a non-additive kernel function.

Our approach is generally applicable as it improves the learning of kernels and the resulting classifiers when facing noisy samples and it excels other per-sample learning approaches that are applicable only to additive kernel functions. However, the retrieval phase of our method remains unchanged, so that computational performance is not affected and it can be readily integrated into existing systems. The proposed per-sample kernel adaptation efficiently suppresses noisy feature components and, thus, yields more accurate similarities between samples. Besides recognition we utilize the improved kernel also for action reconstruction in video by grouping related fragments from a large corpus (Fig. 2).

Figure 2: An example frame of an sport video (left image) and its reconstructions based on closest video fragments found by similarities improved due to our kernel adaptation (right image) and without (middle image).

2. Robust Visual Recognition by Estimating Per-Sample Feature Reliability

In visual recognition problems, such as action recognition or scene classification, the goal is to find a classifier $f : \mathbb{R}^D \rightarrow \mathcal{Y}$ that maps a feature vector $\boldsymbol{x} \in \mathbb{R}^D$, which represents an instance of a visual category (for example by a bag-of-features [5], or a pyramid match kernel descriptor [17]), to a visual class label $y = f(\boldsymbol{x}) \in \mathcal{Y}$. To simplify notation we adopt the commonly used two-class scenario $\mathcal{Y} = \{+1, -1\}$, which is often extended to multiple classes by a one vs. all approach. To learn the classifier f, a set of training examples $\mathcal{X} = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, N\}$ is utilized. Kernel methods, in particular Support Vector Machines, are used in many object, action, and scene classification problems. Here a kernel k implicitly induces a transformation ϕ, $k(\cdot, \cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$, so that a hyperplane in the mapped feature space maximizes the margin between the classes, $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$. According to the representer theorem [26], the classifier f can be expressed as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b, \tag{1}$$

where $(\alpha_1, \ldots, \alpha_N)$ are the dual variables (Lagrange multipliers) associated with training samples.

2.1. Reliability of Individual Features in Kernel Computation

The kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a measure of similarity between feature vectors. Thus, the individual components of both feature vectors are compared by computing distances $d(x_{il}, x_{jl})$ before summarizing over all components, e.g., for Radial Basis Function (RBF) kernel the similarity measure is given as $k_\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \sum_{l=1}^{D} (x_{il} - x_{jl})^2)$. However, the individual components x_{il} can be noisy due to occlusion or clutter, and for each sample $\boldsymbol{x}_i$ different feature dimensions x_{il} may be affected, since oftentimes partial occlusion and noise are corrupting only parts of the feature vector. Consequently, the noise in individual features degrades the kernel values, yielding corrupted similarities.

There are two converse approaches to deal with noisy features in kernel methods. *Feature selection*, which assumes all samples to have the same corrupted feature dimensions l, eliminates identical dimensions from all samples. Contrary to this, SVM multiplies each sample by a non-negative weight α_i, which can limit the influence of entire samples. However, since all dimensions of a sample are multiplied by the same weight, both signal and noisy feature components are either down-weighted or amplified. Consequently, when facing individual noisy, unreliable feature dimensions, a lot of valuable information in the other meaningful dimensions is lost by affecting entire feature dimensions or samples of the data matrix $(x_{il})_{il}$. In both cases we will lose valuable information, when noisy dimensions vary from sample to sample.

To establish a middle ground between both extremes of affecting entire rows or columns of the data matrix, we need for each sample i to estimate the reliability or importance $z_{il} \in [0,1]$ of each of its feature components x_{il}. Feature selection is a special case of this per-sample feature importance, where the z_{il} has the same value for all samples i. However, in the general case of sparse z_{il} simply multiplying them with the x_{il} impairs distance computation and thus corrupts the kernel. For instance replacing $d(x_{il}, x_{jl})$ with $d(0 \cdot x_{il}, 1 \cdot x_{jl})$ will not help to reduce noise, but rather creates a bias. To overcome this problem, the difference of feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in dimension l should affect the kernel value only, if both feature components x_{il} and x_{jl} are important, i.e. z_{il} and z_{jl} have high value. If at least one of them is noisy (unimportant), z_{il} or z_{jl} has low value and the difference of feature components $d(x_{il}, x_{jl})$ should not affect the kernel entry $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

A very broad class of kernel functions, known as Generalized Radial Basis Functions (GRBF), used in many visual recognition problems, such as image classification [33] or action recognition [16], is defined for distance function $d(\cdot,\cdot)$ as

$$k_\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j) := \exp\big(-\gamma \sum_{l=1}^{D} d(x_{il}, x_{jl})\big). \qquad (2)$$

Following up on above ideas, we should concentrate only on those terms in the sum, where both components x_{il} and x_{jl} are important. Thus, the terms are weighted with $z_{il} \cdot z_{jl}$. The entries of the kernel function are therefore weighted by the Hadamard (component-wise) product of feature reliability vectors, $\boldsymbol{z}_i \circ \boldsymbol{z}_j = (z_{i1}z_{j1}, \ldots, z_{iD}z_{jD})^\top$,

$$k_{\gamma(\boldsymbol{z}_i \circ \boldsymbol{z}_j)}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \sum_{l=1}^{D} z_{il}z_{jl}d(x_{il}, x_{jl})). \qquad (3)$$

2.2. Learning Per-Sample Feature Reliability

Subsequently we will discuss how to learn the per-sample feature importance $\boldsymbol{z}_i \in [0,1]^D$ by embedding it into regular SVM training. In the primal we therefore need to optimize the regularized risk functional for SVM parameters $\boldsymbol{w}$ and b and the $\boldsymbol{z}_i$. The kernel function implicitly defines a mapping $\phi_{\boldsymbol{z}}(\boldsymbol{x})$ by $k_{\gamma(\boldsymbol{z}_i \circ \boldsymbol{z}_j)}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi_{\boldsymbol{z}_i}(\boldsymbol{x}_i), \phi_{\boldsymbol{z}_j}(\boldsymbol{x}_j) \rangle$. Thus, the decision function is $f_{\boldsymbol{z}_i}(\boldsymbol{x}_i) = \boldsymbol{w}^\top \phi_{\boldsymbol{z}_i}(\boldsymbol{x}_i) + b$. Moreover, we adopt a soft-margin formulation by introducing slack variables $\xi_i \geq 0$. Since the noisy features are sparse, we use ℓ_1-regularization of the feature vectors $\mathbf{1} - \boldsymbol{z}_i$. The learning problem is then

$$\min_{\boldsymbol{z}_i \in [0,1]^D} \min_{\boldsymbol{w}, b, \xi_i} \lambda \sum_{i=1}^{N} \|\mathbf{1} - \boldsymbol{z}_i\|_1 + \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \xi_i \quad (4)$$
$$\text{subject to } \; y_i(\boldsymbol{w}^\top \phi_{\boldsymbol{z}_i}(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \;\wedge\; \xi_i \geq 0.$$

The sum of slack variables ξ_i can be replaced by the sum of hinge loss functions $\ell(\boldsymbol{x}_i) = \max\big(0, 1 - y_i \cdot f_{\boldsymbol{z}_i}(\boldsymbol{x}_i)\big)$ that act as an upper bound on the training error. The learning problem can thus be formulated as

$$\min_{\boldsymbol{z}_i \in [0,1]^D} \min_{\boldsymbol{w}, b} \lambda \sum_{i=1}^{N} \|\mathbf{1} - \boldsymbol{z}_i\|_1 + \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \max\Big(0, 1 - y_i f_{\boldsymbol{z}_i}(\boldsymbol{x}_i)\Big). \qquad (5)$$

To avoid explicit computation of the mapping $\phi_\bullet(\bullet)$, we solve the inner optimization problem in its dual form. This

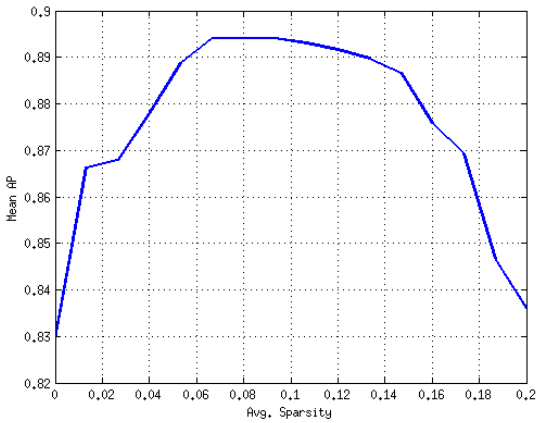

Figure 3: Mean average precision depending on the average per-sample feature sparsity of vector $\mathbf{1} - \boldsymbol{z}_i$ that is controlled by parameter λ in Eq. (6).

Method	Mean AP	Method	Mean AP
ITF [29]	83.3	S-T Graphs [2]	77.3
ITF (*fast*)	82.9	MRP [13]	80.6
Feature Selection [12]	82.9	CTT [10]	82.7
Per sample Feature Suppression	79.4	MKL (additive PSFS) [11]	85.5
PSKA (Eq. 6)	89.4	[29] w/ postproc.	91.1
PSKA (Eq. 6) w/ postproc. of [29]	**91.6**		

Table 1: Mean average precision of various methods for action recognition on the Olympic Sports dataset. Left side: Comparing on the same baseline feature representation (ITF [29]) feature selection, per-sample feature suppression, and our per-sample feature adaptation method (PSKA). Our PSKA with post-processing from [29] outperforms the state-of-the-art methods listed in the right part of the table.

leads to the following minimax problem

$$\min_{\boldsymbol{z}_i \in [0,1]^D} \max_{\alpha_i} \lambda \sum_{i=1}^{N} \|\mathbf{1} - \boldsymbol{z}_i\|_1 + \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j k_{\gamma(\boldsymbol{z}_i \circ \boldsymbol{z}_j)}(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad (6)$$
$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0 \ \wedge \ 0 \leq \alpha_i \leq C.$$

We have seen that $\mathbf{1} - \boldsymbol{z}_i$, which indicates unreliable features, should be sparse. This sparsity is controlled by λ in the learning equation (6), and it is determined via cross-validation. In our experiments on the OlympicSports action recognition dataset (Sec. 4.1), we change λ to induce different degrees of sparsity on the per-sample unreliability $\mathbf{1} - \boldsymbol{z}_i$ and plot the obtained mean AP as a function of the average per-sample sparsity (Fig. 3). As expected, zero sparsity yields the baseline performance [29]. As the sparsity level increases, more and more noisy feature components are suppressed and performance rapidly increases. Fig. 3 shows that the mean AP is constant in a 5% wide interval around the optimum and varies by less than a percent in a 10% wide range, which implies that our method is not sensitive to the value of the sparsity parameter λ. Since performance peaks within a sufficiently wide range, cross-validation can easily find the optimal parameter λ, as we confirmed in our experiments.

2.3. Optimization

To solve (6) during learning we follow a coordinate descent approach. The dual of the SVM optimization problem is solved to find the α_i. Then we find the $\boldsymbol{z}_i$ for each training sample by solving an optimization problem that is expressed as a difference of two convex functions. We solve this problem by employing the Concave-Convex Procedure (CCCP) [32].

To begin, the reliability of all feature dimensions is set to one, $z_{il} = 1$, which corresponds to the standard SVM problem. Given the current estimate of the $\boldsymbol{z}_i$, variables α_i are found as a solution of the convex SVM optimization problem in the dual form (solved using LIBSVM),

$$\max_{\alpha_i} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j k_{\gamma(\boldsymbol{z}_i \circ \boldsymbol{z}_j)}(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad (7)$$
$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0 \ \wedge \ 0 \leq \alpha_i \leq C.$$

To assure convexity of SVM dual optimization problem, and hence the positive semi-definiteness of the kernel matrix, we assumed that it might be necessary to multiply each entry $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ of the kernel matrix by $\exp(-\beta(1 - \delta_{ij}))$, where β is a small non-negative value and δ denotes the Kronecker delta. However, in our experiments the kernel matrix was always positive semi-definite, so β was always set to zero.

Then a $\boldsymbol{z}_i$ is updated given all other $\boldsymbol{z}_j, j \neq i$ and all α_i,

$$\min_{\boldsymbol{z}_i \in [0,1]^D} \lambda \|\mathbf{1} - \boldsymbol{z}_i\|_1 - \frac{1}{2} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k_{\gamma(\boldsymbol{z}_i \circ \boldsymbol{z}_j)}(\boldsymbol{x}_i, \boldsymbol{x}_j). \quad (8)$$

The objective function in Eq. 8 is non-convex, but it can be expressed as the difference $g(\boldsymbol{z}_i) - h(\boldsymbol{z}_i)$ of two convex

Figure 4: Further example frames of sport videos (left column) and their reconstructions based on closest video fragments found by similarities improved due to our per-sample kernel adaptation (right column) and without (middle column). For the whole action video dataset [22], kernel adaption improves similarities leading to on average 0.51 ± 0.05 lower reconstruction error (rms) than without.

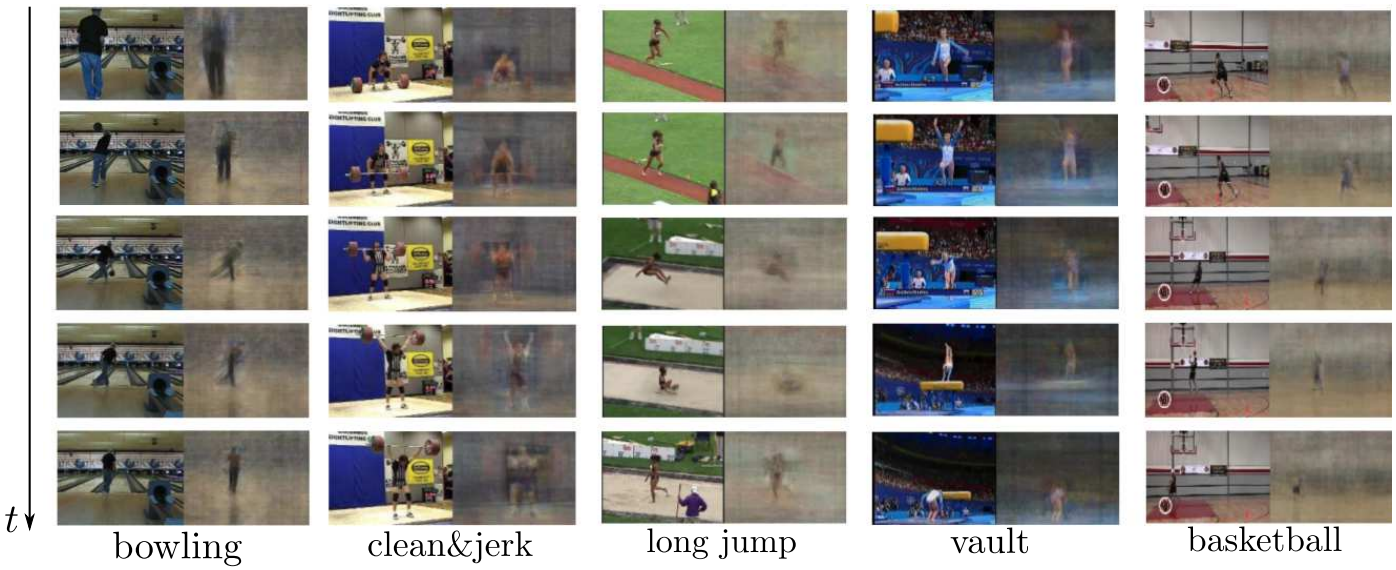

Figure 5: Reconstructing sport action sequences and their characteristic poses using per-sample kernel adaptation (PSKA), cf. Sect. 4.2. Please view in color.

functions,

$$g(\boldsymbol{z}_i) := \frac{1}{2} \sum_{j:y_j \neq y_i} \alpha_i \alpha_j \exp\Big(-\gamma \sum_{l=1}^{D} z_{il} z_{jl} d(x_{il}, x_{jl})\Big) + \lambda \|1 - \boldsymbol{z}_i\|_1,$$

$$h(\boldsymbol{z}_i) := \frac{1}{2} \sum_{j:y_j = y_i} \alpha_i \alpha_j \exp\Big(-\gamma \sum_{l=1}^{D} z_{il} z_{jl} d(x_{il}, x_{jl})\Big).$$

We use the concave-convex procedure (CCCP) [32] to minimize the difference of convex functions $g(\cdot)$ and $h(\cdot)$. By linearizing the concave part, CCCP iteratively solves a sequence of convex optimization problems,

$$\boldsymbol{z}_i^{t+1} = \operatorname*{argmin}_{\boldsymbol{z}_i} g(\boldsymbol{z}_i) - \nabla h(\boldsymbol{z}_i^t)^\top \boldsymbol{z}_i, \quad \text{subject to } \mathbf{0} \leq \boldsymbol{z}_i \leq \mathbf{1}. \tag{9}$$

The constrained convex optimization problem in Eq. 9 is solved using the standard projected gradient descent method. This optimization converges quickly with number of iterations that scales as $O(1/\epsilon)$. Applying accelerated gradient descent [21] further reduces this to $O(1/\sqrt{\epsilon})$. The whole CCCP procedure is fast and typically converges in order of ten iterations.

During training we embed our method within the standard kernel learning whose worst-case complexity $O(N^3)$ dominates the complexity of the kernel matrix computation and the CCCP-based optimization of our method that are both $O(N^2 D)$. Consequently, our method scales just as the standard kernel learning, whose performance it is improving. Moreover, using recent highly efficient techniques for large-scale kernel learning by Dai et al [6] further reduces the overall complexity of our method.

2.4. Recognition Procedure

In the recognition phase a query sample $\boldsymbol{x}_q$ is to be classified by labeling with the according visual category. We have focused on the learning stage to enhance the decision function which then improves recognition. By embedding the per-sample feature selection within the standard SVM approach, the procedure for retrieval remains basically unchanged and the computational burden during recognition is not affected. Therefore, we set $\boldsymbol{z}_q = \mathbf{1}$ and compute the similarities between query sample and all training samples, $k_{\gamma(\boldsymbol{z}_i \circ \mathbf{1})}(\boldsymbol{x}_i, \boldsymbol{x}_q)$ using Eq. (3). Given this kernel, Eq. (1) yields the final classification score. As our method affects only the training phase, the computational complexity of recognition stage is the same as in the underlying standard SVM.

3. Beyond Recognition: Action Reconstruction and Category Summarization

Per-sample kernel adaptation suppresses noisy feature components to improve the kernel function and yield a more robust measure of similarity between samples. Reliable similarities are in turn crucial for finding related samples in a dataset and grouping them. Thus our approach can help to reconstruct a query from a large number of related samples or summarize a category by identifying commonly occurring patterns.

Action Reconstruction. We explore this in the context of video analysis where for a short fragment of a novel

Method	Acc. (%)	Mean AP
RM^2C (baseline) [8]	51.3	46.7
PSKA w/ postproc. of [14]	**63.6**	**64.4**
Prototypes [25]	-	25.1
Object Bank [18]	37.6	-
RBoW [24]	37.9	-
DPM+GIST-color+SP [23]	43.1	-
Patches+GIST+SP+DPM [23]	49.4	-
Mid-Level Patches [27]	38.1	-
BoP w/ postproc. [14]	63.1	63.2

Table 2: Classification accuracy and mean average precision on the MITIndoor dataset. The first part of the table shows the results for the baseline RM^2C method [8] and our approach that uses the same features representation (mid-level part scores). The per-sample kernel adaptation outperforms the baseline RM^2C by 17.7% mean AP. Our PSKA with post-processing from [14] outperforms the state-of-the-art methods listed in the second part of the table.

video we find a large number of related subsequences from training videos. From these videos we densely extract short fragments (10 frames long) and represent these chunks using the standard feature representation of [29] discussed in Sect. 4.1. Per-sample kernel adaptation then yields reliable similarities and, thus, video fragments from the database are ranked according to their relatedness to a query fragment. By averaging over the best matching samples (we cut-off after 50 and weight them according to their similarity) we obtain a reconstruction of the original query and repeat this for all subsequences of the query video. Results are presented in Fig. 4 utilizing our approach of per-sample kernel adaptation (right column) and without (middle). Since the similarities are measured based on the detected persons, their characteristics are nicely preserved, whereas the large deviations in the background blur out the clutter.

Fig. 5 shows the reconstruction frame sequences of five different sports from the OlympicSports dataset [22]. The approach to reconstruction based on PSKA is the same as in Fig. 4. We see that our method nicely captures the temporal evolution of the action and reveals the structure of a sport as a sequence of characteristically articulated human poses.

Category Summarization. Let us now summarize an action category by identifying commonly occurring fragments. Therefore, we measure similarities between all short subsequences extracted from the videos of an action class based on our approach (typically around 3000 fragments for the classes of [22]). All fragments in a video get the action class label of the whole video, so Eq. 6 becomes applicable. Performing clustering using the normalized cut algorithm yields clusters of related fragments and we choose the samples closest to the cluster center as representative for visualization. Bearing paper length in mind we compute and portray a five cluster solution for the basketball class in Fig. 6, with individual frames of each cluster being shown as filmstrip. The clustering summarizes characteristic aspects of this category that are occurring frequently such as running, walking, blocking, passing or jumping.

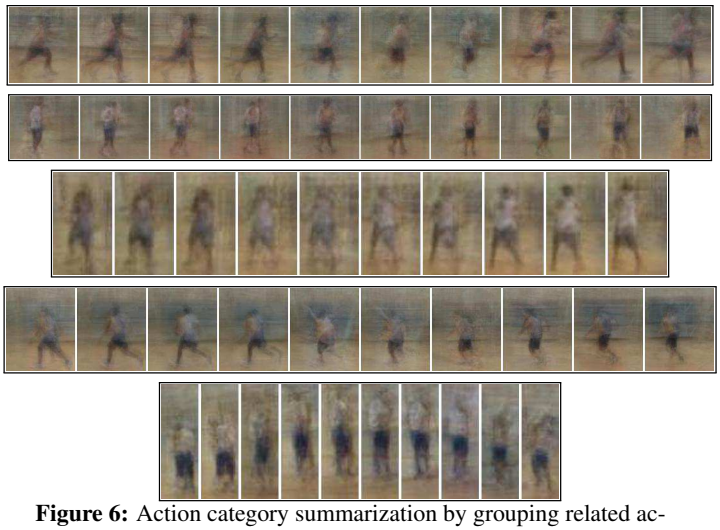

Figure 6: Action category summarization by grouping related action components using the proposed kernel adaptation based distance measure. The basketball class splits up into groups that represent *running* (first filmstrip), *walking* (second row), *blocking* (third row), *passing* (fourth row), and *jumping* (fifth row).

4. Experimental Evaluation

Subsequently, we evaluate the proposed per-sample kernel adaptation on two diverse visual recognition problems, for which there is a consensus in literature about standard features and preprocessing techniques. This allows for a fair comparison to state-of-the-art, so difference in performance can be attributed to the method and not to different features or preprocessing. First we investigate the potential of our approach for action recognition in videos and their reconstruction, where we use the popular Olympic Sports benchmark set [22]. Afterwards we study recognition of indoor scenes on the standard MITIndoor benchmark dataset [25]. In both cases, we build on top of the most recent feature representations and classifiers proposed for these datasets to investigate the ability of our method to yield a further improvement. Our per-sample kernel adaptation is then simply integrated into these state-of-the-art approaches to eliminate noisy feature contributions when computing the kernel matrix. Essentially, we replace existing kernel-based learning by Eq. (6) and the corresponding optimization from Sect. 2.3. The recognition procedure remains unaltered.

4.1. Action Recognition in Videos

To evaluate the potential of our approach for action recognition we utilize the Olympic Sports dataset [22] that features athletes performing 16 different sport actions, such

	basket ball	bowl ing	clean &jerk	discus throw	dive 10m	dive 3m	hamm. throw	high jump	jave lin	long jump	pole vault	shot put	snatch	tennis	tripple jump	vault	mean
ITF (*fast*)	97.5	77.9	78.0	85.1	**100**	**100**	95.2	61.4	**100**	88.3	89.5	71.0	72.1	94.5	35.0	81.1	82.9
Feature Selection [12]	96.7	78.6	78.8	85.7	**100**	**100**	95.2	58.3	**100**	84.8	85.0	72.3	75.0	97.7	33.6	82.5	82.9
Per Sample Feat. Suppr.	97.5	80.5	78.6	85.1	**100**	**100**	93.9	53.7	**100**	81.8	58.2	54.3	67.4	91.1	47.4	80.6	79.4
PSKA	98.3	82.3	87.2	89.9	**100**	**100**	**100**	63.0	**100**	93.5	**98.0**	82.7	81.8	**100**	**67.5**	**86.3**	89.4
PSKA w/ postproc. of [29]	**100**	**86.4**	**91.7**	**95.7**	**100**	**100**	98.0	**86.5**	**100**	**100**	88.6	**83.5**	**91.4**	**100**	60.6	82.8	**91.6**

Table 3: Per-category average precision for the Olympic Sports dataset. All methods use the same feature representation as in the ITF baseline [29]. Abbreviations are the same as in Table 1. The proposed per-sample kernel adaptation (last two rows) significantly improves performance.

as high jump, vault, bowling etc. The dataset consists of 783 videos obtained from YouTube. We follow the standard experimental setup of [22], i.e., use 649 videos for training and test on the remaining 134 video sequences. For each category we train a one vs. all classifier, and report its average precision. Finally, mean average precision is computed over all categories.

For action recognition, dense trajectory features [28] and their extension, the improved trajectory features (ITF) [29], are very popular descriptors that have been widely employed. Therefore, we also use ITF features, which capture local appearance and motion information of a video by combining HOG, HOF, motion trajectories, and motion boundary histograms (MBH). We follow the standard setup and create a bag-of-features from the quantized ITF features and train a non-linear SVM classifier using a Gaussian kernel (γ being set to the inverse of the average pair-wise dissimilarity of all training samples). To speed up the computation of the kernel matrix during training, we replace the χ^2 distance in the Gaussian kernel with the ℓ_1 distance. This reduces training time by a factor of two (30 min on an *i7* desktop) and decreases mean AP only marginally by 0.4% (our fast ITF feature implementation achieves 82.9% compared to 83.3% of [29], c.f. Table 1). Since the underlying features and setup is the same as in [29], the proposed method of per-sample kernel adaptation (PSKA) is the only difference. It yields 89.4% mean AP (Tab. 1), which is a 6.5% improvement over the fast implementation of the baseline ITF method [29].

Furthermore, we compare with standard feature selection [12] that eliminates the same feature dimensions in all samples. The number of eliminated components by feature selection is set to be the same as the number of feature components that are assigned low reliability ($z_{il} < 1/2$) by our PSKA method (we also experimented with different numbers, which however did not increase results). The feature selection (cf. Table 1) does not improve over the baseline method (fast ITF), since it also removes meaningful components. This emphasizes the benefit of the proposed approach, which achieves 6.5% higher mean AP, as it only eliminates the noise.

We also examine the effect of directly suppressing individual feature components, i.e., by multiplying $z_{il} \cdot x_{il}$, and compare it with our approach, which eliminates noisy pairwise distances in the kernel space. For a fair comparison of these complementary approaches, the z_{il} are the same in both cases. The result in Table 1 shows that per-sample feature suppression significantly reduces performance by 10% compared to our per-sample kernel adaptation (PSKA). This is understandable since comparisons between $z_{il} \cdot x_{il}$ are flawed when z_{il} differ between samples, cf. Sect. 2.1.

Table 1 also shows that our method yields competitive performance with respect to state-of-the-art methods on the Olympic Sports dataset [29, 2, 13, 10, 11]. In particular, the comparison shows that using the same features, our approach achieves an improvement of 4% with respect to MKL [11] that is a current, competitive method for additive-kernel per-sample feature selection (PSFS). Moreover, using the post-processing of [29] we gain an additional 2.2%, yielding a competitive performance of 91.6%.

In Table 3 we compare for each category of the Olympic Sports dataset the baseline ITF method [29], feature selection [12], per-sample feature suppression, and our PSKA approach. Compared to the fast implementation of ITF [29] we see a significant improvement of about 10% on four categories and more than 30% on triple jump.

4.2. Action Reconstruction Using Improved Kernel Similarity

Fig. 2 and Fig. 4 show that the proposed kernel adaptation improves similarities between fragments of action sequences. Based on these robust relations between subsequences, related frames can be aggregated, thus enabling a reconstruction of meaningful parts of videos as discussed in Sect. 3. We evaluate reconstruction quality using the whole action video dataset (cf. Fig. 4). On average the proposed PSKA leads to 0.51 ± 0.05 lower reconstruction error (root-mean-square deviation) than the baseline method that uses standard kernel similarity. Furthermore, Fig. 5 shows that through aggregating related frames PSKA nicely retains the visual information relevant for the action while suppressing irrelevant background clutter. In that process meaningful aspects such as characteristic poses and their change over time are retained, while irrelevant and highly variable prop-

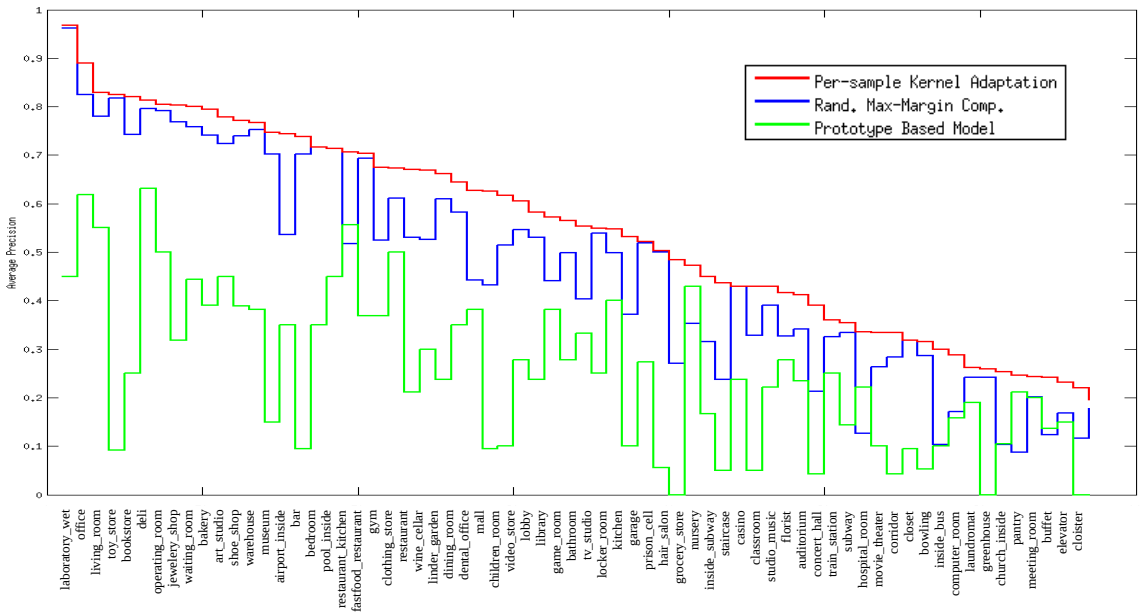

Figure 7: Average precision for all categories of the MIT indoor dataset. Categories are ordered according to our method's AP (red curve). There is a consistent improvement compared to the baseline of this benchmark [25] (green) and over RM^{2}C (blue) [8], which is the basis for our scene representation.

erties such as the color of athletes' shirts are disregarded. Similarly, less characteristic poses are extenuated such as in the long jump category the highly volatile twisting in the sand upon landing.

4.3. Scene Classification

To show that our approach is not limited to video recognition but also applicable to static images, we evaluate our method on scene recognition. We use the challenging MITIndoor benchmark image dataset [25]. The dataset consists of 67 indoor scene categories, such as bookstore, lobby, classroom etc. The dataset contains in total 6700 images, such that each category contains 80 images for training, and 20 images for testing. As a baseline scene representation we employ the randomized parts (RM^{2}C) of [8], downloaded from the authors' website. We follow the standard protocol of [25] and train one vs. all classifiers for each scene category and evaluate them by computing classification accuracy and mean average precision. We use the same Gaussian kernel with ℓ_1 distance as in Sect. 4.1. Table 2 compares our per-sample kernel adaptation (PSKA) to RM^{2}C [8], which is the basis for our scene representation, as well as to other state-of-the-art methods for scene classification. By estimating the reliability of the randomized parts from [8] our method effectively suppresses noisy parts in kernel computation, which results in 17.7% improvement in mean AP over the baseline and 12.3% improvement in classification accuracy. We also compare to established methods for scene classification that use similar part-based representations. Our method's mean AP of 64.4% is an improvement of 1.2% over the state-of-the-art bag of parts (BoP) method [14], whose post-processing we also used and directly added to our model to be comparable, without adjusting any parameters.

Moreover, Fig. 7 shows per-class average precision for all 67 categories of the MITIndoor dataset. We see that our method consistently improves upon baseline RM^{2}C [8] and [25] on all categories of this dataset.

5. Conclusion

We have presented an approach to deal with one of the main problems of visual recognition—noisy representations that impair recognition performance. Per-sample removal of corrupted feature dimensions integrated into kernel computation retains reliable feature components while suppressing the flawed ones. Evaluation on the divergent tasks of action recognition, reconstruction, and scene classification has shown that this approach can be readily integrated into the learning stage of kernel-based recognition systems to improve their performance.[1]

[1]This research has been funded in part by the Ministry for Science, Baden-Württemberg and the Heidelberg Academy of Sciences, Heidelberg, Germany.

References

[1] B. Antic and B. Ommer. Learning latent constituents for recognition of group activities in video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. *Computer Vision, IEEE International Conference on*, 0:778–785, 2011.

[3] O. Chapelle and S. S. Keerthi. Multi-class feature selection with support vector machines. In *Proceedings of the American statistical association*, 2008.

[4] C. M. Christoudias, R. Urtasun, A. Kapoor, and T. Darrell. Co-training with noisy perceptual observations. In *CVPR*, pages 2844–2851. IEEE, 2009.

[5] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[6] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. F. Balcan, and L. Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems 27*, pages 3041–3049. Curran Associates, Inc., 2014.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, June 2005.

[8] A. Eigenstetter, M. Takami, and B. Ommer. Randomized Max-Margin Compositions for Visual Recognition. In *CVPR - International Conference on Computer Vision and Pattern Recognition*, Columbus, USA, 2014.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. In *BMVC 2012 - British Machine Vision Conference*, pages 30.1–30.13, Sept. 2012.

[11] M. Gönen and E. Alpaydın. Localized algorithms for multiple kernel learning. *Pattern Recognition*, 46(3):795–807, 2013.

[12] Y. Grandvalet and S. Canu. Adaptive scaling for feature selection in svms. In *NIPS*, pages 553–560, 2002.

[13] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[14] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Ziserman. Blocks that shout: Distinctive parts for scene classification. *CVPR*, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012.

[16] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*, jun 2008.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006 - IEEE Conference on Computer Vision & Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[18] L.-J. Li, H. Su, e. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. *NIPS*, 2010.

[19] X. Liu, L. Wang, J. Zhang, and J. Yin. Sample-adaptive multiple kernel learning. AAAI Conference on Artificial Intelligence, 2014.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[21] Y. Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004.

[22] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10, pages 392–405, Berlin, Heidelberg, 2010. Springer-Verlag.

[23] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. *ICCV*, 2011.

[24] S. N. Parizi, J. Oberlin, and P. F. Felzenswalb. Reconfigurable models for scene recognition. *CVPR*, 2012.

[25] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420, 2009.

[26] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[27] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. *ECCV*, 2012.

[28] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, May 2013.

[29] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *ICCV 2013 - IEEE International Conference on Computer Vision*, pages 3551–3558, Sydney, Australie, Dec. 2013. IEEE.

[30] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39, Sept 2009.

[31] J. Yang, Y. Li, Y. Tian, L.-Y. Duan, and W. Gao. Per-sample multiple kernel approach for visual concept learning. *J. Image Video Process.*, 2010:2:1–2:13, Jan. 2010.

[32] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Comput.*, 15(4):915–936, Apr. 2003.

[33] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, jun 2007.