Hybrid Classifiers for Object Classification with a Rich Background

Margarita Osadchy, Daniel Keren, and Bella Fadida-Specktor

Computer Science, University of Haifa,
Carmel, Haifa 31905,Israel
{rita,dkeren@cs.haifa.ac.il,belkasp@gmail.com}

Abstract. The majority of current methods in object classification use the one-against-rest training scheme. We argue that when applied to a large number of classes, this strategy is problematic: as the number of classes increases, the negative class becomes a very large and complicated collection of images. The resulting classification problem then becomes extremely unbalanced, and kernel SVM classifiers trained on such sets require long training time and are slow in prediction. To address these problems, we propose to consider the negative class as a *background* and characterize it by a prior *distribution*. Further, we propose to construct "hybrid" classifiers, which are trained to separate this distribution from the samples of the positive class. A typical classifier first projects (by a function which may be non-linear) the inputs to a one-dimensional space, and then thresholds this projection. Theoretical results and empirical evaluation suggest that, after projection, the background has a relatively simple distribution, which is much easier to parameterize and work with. Our results show that hybrid classifiers offer an advantage over SVM classifiers, both in performance and complexity, especially when the negative (background) class is large.

1 Introduction

One of the central problems in computer vision is recognizing objects in realistic scenes. We deal with the classification problem, defined as predicting whether at least one object of a given class is present in an image. The basic recipe for this kind of problems has been 1) constructing a Bag of Visual Words or spatial pyramids [1] of multiple features, 2) vector quantization, 3) training SVM classifiers with histogram intersection[1], Fisher [3, 4] or other kernels, and 4) integrating classifiers using voting or MKL [5]. Recent work focused on devising new and better features and kernels (e.g. [6]), various coding strategies (e.g. [7]), etc. Most of these methods adopt a one-against-rest strategy for training SVM classifiers, in which the positive class is composed of samples from a single class and the negative class comprises samples from all remaining classes. When the number of classes is relatively small, the one-against-rest training scheme was shown to be as good as multi-class classifiers [8]. However, in real problems, the negative class – the background – is much richer and includes all (up to tens of thousands) object categories (all except the positive class). When the number of classes is large, the one-against-all scheme faces two major problems: extremely unbalanced training sets, and high computational complexity [4].

Unbalanced Sets. It is a common observation that when trained on unbalanced sets, the class-boundary learned by SVMs can be severely skewed towards the smaller class and it becomes very sensitive to noise [9]. Several approaches have been proposed to solve this problem (a review of previous work is provided in [9, 10]), including setting different penalties for misclassifying the positive class relative to the negative one, various weighting techniques, undersampling the majority class or oversampling the minority class, adjusting the class boundary based on the spatial distribution of the support vectors, and various combinations of the above. All these methods, however, do not consider the complexity problem. Thus using weighted SVM or any other of these methods as a one-against-rest classifier for a large data set is impossible, especially when a kernel classifier is applied, since the number of support vectors increases linearly with the number of training examples [11]).

High Computational Complexity. Kernel SVM was shown to be the most successful among one-against-rest classifiers for object recognition tasks ([2, 1, 12]). However, it cannot be used in large-scale problems, because its training is slow and requires a large memory. Further, its prediction rule is too expensive when the number of support vectors is large. To address the complexity problem of kernel SVM, several methods have been proposed which can be divided into three groups: 1) post-processing methods that replace the set of support vector (SV) with an approximate sparser set (this requires to compute the original SV's first and thus it is not suitable for our problem), 2) methods that a-priori select a set of basis vectors from the training set in order to approximate the kernel matrix, 3) methods that choose vectors that are efficient for classification, and approximate the kernel matrix (see [13, 14] for a more detailed discussion). The empirical evaluation of these methods [14] shows that all of them trade accuracy for complexity. Although recent methods [14] come very close to the accuracy of the exact SVM, they have only been applied to balanced problems.

In order to design a tractable nonlinear classifier for the large-scale categorization problem, a special form of kernel, such as additive kernels [3] or an explicit mapping [4] have been used. However, no efficient solution exists for the general kernel formulation.

To summarize, there are solutions for unbalanced sets but these are computationally inefficient, and there are also solutions that approximate the kernel classifier efficiently, but these are not designed for unbalanced sets. Further, adding a new category requires retraining all the one-against-rest classifiers, making the approach even more problematic.

In visual classification problem, the negative class approaches the complement of the positive class and thus it can be viewed as a general "background class". In this work we propose classifiers that are specifically designed to separate a class from a rich background. By "background" we mean *all images* except the category to be recognized. Learning this background from samples is highly problematic. We suggest *replacing the background samples by a distribution.* The idea is straightforward – instead of minimizing the *number* of background samples in the classifier's acceptance region, we minimize the overall *probability volume* of the background prior in the acceptance region. This formulation eliminates the problem of unbalanced training set (since there are no negative samples) and the high complexity due to the large number of "negative" support vectors. From now on we shall refer to this type of classifiers as *hybrid.* The notion of hybrid classifier we use here characterizes the mixed input to the training phase: samples from the positive class vs. probability distribution on the background.

The idea of a hybrid classifier was first introduced in [15], but the solution proposed there was restricted to grey-level images and applied a very simple prior, which is not robust to illumination variation and image deformations. Further, only a linear classifier was presented. Here we extend the basic paradigm to realistic scenes and propose the following contributions:
1. Although modeling the background accurately is difficult, we observe that in classification tasks typically one seeks to separate the values of the two categories (or in this case, a single category and the background) after they were projected (either linearly, as in linear SVM, or by a more complicated function, e.g. a kernel) into the real line. We show that the projection of a complicated background can be well-approximated by a simple distribution (e.g., Gaussian). Thus, we suggest that as the number of image categories in the background class increases, the method described here will become even more suitable.
2. We built priors for robust features, such as Bag of Words constructed from densely sampled SIFT features [16, 17]. The prior assumed in [15] was based on the observation that typical images are "smooth", that is, most of their energy is concentrated in the low frequencies. Although BoW features obviously do not posses this property, we show that they can be successfully used with the hybrid classifiers, suggesting that the basic paradigm is very general and can be applied to other features and domains.
3. We developed a kernel hybrid classifier that can be used with a kernel of general form, and is much more efficient than kernel SVM in both training and classification, while it enjoys an even better classification accuracy.

1.1 Modeling the Background Distribution

Compared to a single object class, the background distribution is so wide that it can be assumed to be approximately equal to the distribution of all natural images, hence we can use this distribution to model the background class (this model will therefore be applicable to *all* single classes one wishes to detect, thus drastically reducing training complexity). Modeling the distribution of natural images is, however, a challenging task. A number of energy-based models have been proposed to learn this distribution from examples (e.g. [18–21]). These models attempt to find a set of linear filters in order to decompose the image into channels, as well as the corresponding energy functions. Training most of these models is very long, which is not a burden if it's computed once and then used

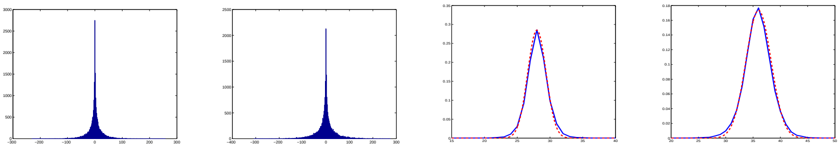

Fig. 1. Examples of 1D random projections of the background class. The two histograms on the left correspond to grey-level with 8×8 filter size (similarly to previous work on image statistics). The projections are clearly non-Gaussian. The other two histograms correspond to BoW of SIFT features (shown in blue solid line). The projections are very close to Gaussians (dashed red line).

for an application that employs a fixed prior. We are interested in determining a suitable prior on natural images and applying it to classification. In light of this we need to evaluate, during training, the probability of background images to be accepted by the classifier; this probability reflects the percentage of false positives, which the classifier seeks to minimize. Such an evaluation is performed for each choice of parameters for the candidate classifier.

Since the final step in classification consists of thresholding a scalar-valued function, we are interested in modeling the projections or outputs of scalar-valued functions applied on the space of natural images. Modeling projections of natural images has also been studied in low-level vision. The most striking difference between the functions applied to features commonly used in object recognition and the linear filters applied to grey-levels in low-level vision [18] is the form of the distribution they produce. Applying linear filters, such as derivative-like filters, wavelets etc. to natural images, represented by grey-levels, produces outputs whose distribution is highly non-Gaussian – it is peaked at zero and has heavy tails [21] (Figure 1). We are interested in non-linear functions of grey-levels, such as Bag of Words [17], constructed from SIFT features [16]. Our experiments suggest that projections of these representations are Gaussian-like (Figure 1). As elaborated in Section 2, this allows to efficiently approximate the distribution of the projections of the background class.

2 Hybrid Classifier

We propose to incorporate the background prior in a hybrid classifier $f(x)$, which is trained to attain positive scores on the samples of the target class and for which $\int\limits_{H} \Pr(z)dz$ is very small; z belongs to the background distribution and H is the acceptance region of $f(x)$ (i.e. all x for which $f(x) \geq 0$). Thus the standard constraints of excluding background *samples* are replaced by a *single* constraint of excluding a large volume of background *probability*.

2.1 Linear Classifier

We search for a separating hyperplane $(\mathbf{w}, b)$ which yields a maximum margin between itself and the positive samples, under the constraint that the integral

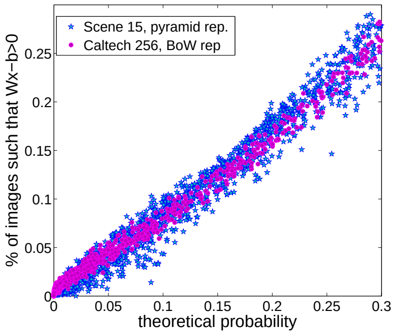

Fig. 2. Relation between the percentage of natural images in the positive half-space and the Gaussian approximation in Eq.1 tested on Caltech-256 and Scenes-15 data sets. The plot is zoomed on the [0, 0.3] interval of the probability which is more relevant to the proposed formulation.

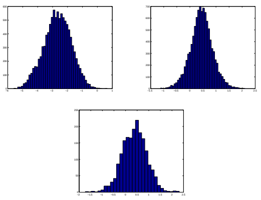

Fig. 3. Histograms of values of histogram intersection (top left), χ^2 (top right), and SPM (bottom) kernels with randomly selected parameters, applied to many background samples represented by a BoW of SIFT features. The top row corresponds to the Caltech-256, the bottom to Scenes-15.

of the probability density of the background (natural images) in its acceptance region $\mathcal{H} = \{\mathbf{x}|\mathbf{w}\cdot\mathbf{x} \geq b\}$ is small. We bound the probability of natural images to fall in the acceptance region: $\Pr(\mathbf{w}\cdot\mathbf{x} \geq b) < \delta$, where the constant δ is close to zero. Diaconis and Freedman [22] showed that under certain independence conditions, low-dimensional projections of high-dimensional data are typically close to Gaussian. We empirically demonstrate that this proposition holds for one-dimensional random projections applied to two large sets of images: Caltech-256 [23] and Scenes-15 [1]. We used all 30,607 images of 256 categories from Caltech-256 and 3,000 images of 15 scenes from Scene-15. Images in Caltech-256 are quite diverse and objects appear in various scales and orientations; images of scenes contain many objects. Thus these sets can serve as an approximation to the set of natural images. We used the BoW representation provided in [5][1] for Caltech-256 and 3-level pyramids of BoW [1] for Scenes-15. We tested hundreds of random projections for both sets, and all of them are well-approximated by one-dimensional Gaussians (Figure 1, the two histograms on the right). The first two images in Table 1 show that the distribution of the projections which correspond to the learned classifiers is quite similar to the distribution of random projections, which supports the Gaussian assumption. Our experiments show that the Gaussian approximation of the projections bounds the background probability in the acceptance region of the learned classifiers in all our tests (see Section 3).

In order to obtain a closed-form, general expression for the distribution of the projections, we first estimate the mean and covariance matrix of the high-dimensional distribution, denoted $\bar{\mathbf{x}}$ and Σ_x respectively. Then, the projection is

[1] http://www.vision.ee.ethz.ch/ pgehler/projects/iccv09/index.html

a random variable with mean $\mathbf{w}^T\bar{\mathbf{x}}$ and variance $\mathbf{w}^T\Sigma_x\mathbf{w}$. Following the previous discussion, we approximate this variable by a Gaussian, thus the probability of a background image to be accepted by the classifier is

$$\Pr(\mathbf{w}\cdot\mathbf{x}\geq b)=\frac{1}{2}\left[1-\mathsf{erf}\left(\frac{1}{\sqrt{2}}\frac{b-\mathbf{w}^T\bar{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}}\right)\right] \tag{1}$$

For the approximation to be valid, the real (empirical) probability of a background image to lie in a certain half-space should be close to the one derived from the prior (or the "theoretical probability"). The empirical probability can be estimated by testing many randomly chosen background images, while the theoretical probability can be computed (as in Eq. 1). We tested the similarity between the two probabilities on Caltech-256 and Scenes-15 using the image representations, described above; results are presented in Figure 2. We used disjoint sets: one to estimate the mean and the covariance of $\mathbf{x}$ and the other to estimate the probability that a natural image falls in the "positive" half-space. We randomly chose $\mathbf{w}$, constraining its norm to be 1, and a value for b in the range [-0.5, 0,5]. For each choice of $(\mathbf{w},b)$ we computed the expression in Eq. 1 and used it as the x-coordinate of a point in the scatter plot in Figure 2. The y-coordinate represents the empirical probability, and it is computed as the actual percentage of the images that fall in the positive half-space. The scatter plot supports the validity of the proposed approximation. A similar relation has been shown in [15] for the class of natural images represented in the frequency domain. This suggests that such relations hold for different features and different data sets.

Based on the above observations, the constraint on the probability of background misclassification is given by:

$$\Pr(\mathbf{w}\cdot\mathbf{x}\geq b)=\frac{1}{2}\left[1-\mathsf{erf}\left(\frac{1}{\sqrt{2}}\frac{b-\mathbf{w}^T\bar{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}}\right)\right]\leq\delta \tag{2}$$

Since we seek to minimize $\Pr(\mathbf{w}\cdot\mathbf{x}\geq b)$, we assume that $\delta<1/2$, and thus $\gamma\triangleq\sqrt{2}\mathsf{erf}^{-1}(1-2\delta)>0$. By formulating the constraint in Eq. 2 in terms of γ, and rearranging, we obtain a convex constraint:

$$\gamma\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}+\mathbf{w}^T\bar{\mathbf{x}}-b\leq 0 \tag{3}$$

A more general argument, which does not require the Gaussian approximation assumption, can be applied to justify the constraint in Eq. 3. To show this we apply a result from [24], which states that for a half space $S=\{\mathbf{w}\cdot\mathbf{y}\geq b\}$, and all distributions y with expectation $\bar{\mathbf{y}}$ and covariance matrix Σ_y:

$$\sup_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_y)}\Pr(\mathbf{w}\cdot\mathbf{y}\geq b)=\frac{1}{1+d^2},d^2=\frac{b-\mathbf{w}^t\bar{\mathbf{y}}}{\mathbf{w}^t\Sigma_y\mathbf{w}} \tag{4}$$

Now, instead of constraining the probability, we constrain its supremum over all distributions for $\mathbf{x}$ having mean $\bar{\mathbf{x}}$ and covariance Σ_x. Using Eq. 4 we obtain:

$$\sqrt{\frac{1-\delta}{\delta}}\sqrt{\mathbf{w}^T\Sigma_x\mathbf{w}}+\mathbf{w}^T\bar{\mathbf{x}}-b\leq 0 \tag{5}$$

which turns out to be the same as Eq. 3 with $\gamma = \sqrt{\frac{1-\delta}{\delta}}$.

Using the above considerations, we define the linear hybrid classifier as the solution to the following optimization problem: Given a set $\{x_j\}_{j=1}^{n}$ of positive examples: minimize $\|\mathbf{w}\|^2$, subject to $\mathbf{w}\cdot\mathbf{x}_j - b \geq 1$ $(j = 1,..,n)$ and the probability constraint in Eq. 3. This formulation resembles the usual SVM algorithm but with the many constraints on the negative examples replaced by *one* constraint on the probability. Note that the background slackness is controlled by the parameter δ. Adding slacks to positive samples could be done similar to SVM, but for a small number of positive examples it has no effect.

2.2 Kernel Classifier

We use a standard kernel decision function:

$$f(\mathbf{x}) = \mathbf{sign}(\sum_{i=1}^{l} \alpha_i K(\mathbf{s}_i, \mathbf{x}) - b)$$

where α_i, $\mathbf{s}_i$, and b are the model parameters. The $\mathbf{s}_i$'s are chosen from a set of unlabeled training examples, as described later.

To compute the probability

$$\Pr(\sum_{i=1}^{l} \alpha_i K(\mathbf{s}_i, \mathbf{x}) \geq b)$$

on the background class, we define a random variable in the kernel space $\mathbf{z} = [z_1,..,z_l]^t$, where $z_i \triangleq K(\mathbf{s}_i, \mathbf{x})$ $(i = 1,..,l)$, ($\mathbf{x}$ is a random variable in the input space, representing the background). Then, we write the probability constraint as

$$\Pr(\sum_{i=1}^{l} \alpha_i z_i \geq b) \leq \delta \tag{6}$$

This constraint has the same form as in our linear classifier. Similarly, we can apply the Gaussian approximation and obtain the same expression as in Eq. 3, with the only difference that $\mathbf{x}$ is replaced by $\mathbf{z}$, which is obtained by applying a non-linear function $K(\mathbf{s}_i, \mathbf{x})$. Thus the constraint is

$$\gamma\sqrt{\alpha^t \Sigma_z \alpha} + \alpha^t \mu_z - b \leq 0 \tag{7}$$

where μ_z is the mean and Σ_z the covariance matrix of $\mathbf{z}$.

Next, we check the validity of the proposed approximation for several kernels, used in object recognition, namely, the histogram intersection, χ^2, and Spatial Pyramid Match (SPM) [1] kernels. Figure 3 shows examples of outputs of these kernels. To create random projections in kernel space we randomly chose 1000 samples as $\mathbf{s}_i, i = 1...1000$, and 1000 scalars as $\alpha_i, i = 1...1000$, and evaluated, using a large collection of images $\mathbf{x}$, the value of $\sum \alpha_i K(\mathbf{s}_i, \mathbf{x})$ (where $K()$ are the above-mentioned kernels). Table 1 (images 3-6) depicts examples of projections on learned classifiers. These distributions do not differ much from the random projections, which supports the Gaussian assumption.

The result from [24] can be applied to the kernel classifier too (here we consider the supremum over all distributions for $\mathbf{z}$ having mean μ_z and covariance Σ_z) and leads to the following constraint:

$$\sqrt{\frac{1-\delta}{\delta}}\sqrt{\alpha^t \Sigma_z \alpha} + \alpha^t \mu_z - b \leq 0 \tag{8}$$

which is essentially the same as Eq. 7.

We formulate the following convex optimization problem to learn the hybrid kernel classifiers. Given a set $\{x_j\}_{j=1}^n$ of positive examples:

$$\min_{\alpha} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j K(\mathbf{s}_i, \mathbf{s}_j) \tag{9}$$

subject to

$$\sum_{i=1}^{l} \alpha_i K(\mathbf{s}_i, \mathbf{x}_j) - b \geq 1 \quad \forall j = 1, .., n$$

$$\gamma\sqrt{\alpha^t \Sigma_z \alpha} + \alpha^t \mu_z - b \leq 0$$

Here we use a standard kernel regularizer as an objective function (Eq. 9).

We now return to the question of choosing the s_i's which define the classifier $f(x) = \sum_i \alpha_i K(s_i, x)$. The basic idea is to find a family $\{s_i\}$ such that the span of $K(s_i, x)$ approximates all the functions $K(s, x)$, where s ranges over the sample space. A similar approximation problem has been addressed in [25]: find a subset of indexes $I = \{i_1, .., i_m\} \subset [t]$ (where t is the size of the full kernel matrix K) such that $\tilde{K}_i = \sum_{j=1}^{m} K_{i_j} T_{ji}$, where K_i are the columns of the kernel matrix, T is an $m \times t$ matrix containing the expansion coefficients for an approximation of the columns of K and $T_{ii_j} = \delta_{ij}$. I and T are chosen to minimize the Frobenius norm $\|\tilde{K} - K\|_{Frob}$. A greedy, probabilistic algorithm from [25] chooses I in $O(mt)$ time complexity per index. Here we define K to be the kernel matrix of the unlabeled training samples and $s_i = \tilde{K}_i$; then we apply the algorithm from [25] for finding $\tilde{K}_i$.

In our formulation, $\mathbf{s}_i$'s represent the background, and are chosen independently and prior to the training of the classifiers. Thus we will refer to this set as the "common" $\mathbf{s}_i$'s. To learn the classifier for a specific class, we add its positive examples to the common $\mathbf{s}_i$'s and run the optimization in Eq. 9. This is much faster than training kernel SVM, as far less parameters need to be optimized over. Our experiments also suggest that the number of common $\mathbf{s}_i$'s required to represent a rich background is small and it doesn't increases as the number of background categories increases.

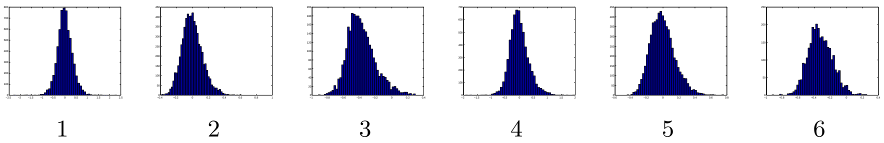

Table 1. Examples of 1D projections of test images on separating hyperplane corresponding to different hybrid classifiers: the first four distributions correspond to classifiers trained on different categories from Caltech-256, the first two –linear classifier, the third and forth – SPM kernel classifiers; the last two correspond to SPM kernel trained on two different categories from Scene-15.

2.3 Complexity

The training of hybrid classifiers consists of two steps. The first is performed only once and it includes the selection of $\mathbf{s}_i$'s (only for the kernel classifier) and the estimation of the the background covariance matrix. The second step is the actual training of the classifier, which is done per object class and thus repeated the number of times equal to the number of classes one wishes to recognize. Recall that n is the number of positive examples, m the number of common $\mathbf{s}_i$'s, p the number of unlabeled samples for selecting $\mathbf{s}_i$'s, and C is the number of categories comprising the background class in the one-against-rest training phase.

Linear

Estimation of the background covariance matrix: Even though an accurate estimation of the covariance matrix of a high-dimensional random variable requires many samples, here we are only interested in its 1D projections, thus an approximation of the covariance matrix suffices. We observed that the number of background samples required to derive this approximation is relatively small: in the Caltech 256 experiments, increasing the number of samples beyond five per category had a negligible effect on the projection's parameters as well as on the performance. Note that the background covariance matrix has to be estimated *only once*, and then it is applied for training classifiers for *all* classes.

Training a classifier per category: Our optimization has only $n+1$ constraints (n positive examples and one probability constraint), while the number of constraints in one-against-all SVM training is nC. Another important advantage is that we do not need to keep a huge number of negative examples in memory, which allows using off-the-shelf solvers for convex optimization even for a large scale classification problems. The classification process is the same as for linear SVM.

Kernel

Choosing $\mathbf{s}_i$*'s:* To find the common $\mathbf{s}_i$'s we use the algorithm from [25], which runs in $O(mp)$ per vector, thus the entire process runs in $O(m^2p)$. The selection is performed only once, and even for a very rich background, the size of the basis m is small (about 200). (Section 3.1).

Estimation of the background covariance matrix: The size of the covariance matrix is $(m+n)^2$, of which the block of size $m \times m$ is identical for all classes (since common $\mathbf{s}_i$'s do not depend on the class one wishes to recognize), and the block including the class-related $\mathbf{s}_i$ of size $n \times (m+n)$, that must be estimated for each class. Typically both n and m are quite small (see Section 3.1), thus estimating the covariance matrix is not a burden.

Training a category classifier: Our optimization is limited to $m+n$ parameters, compared to nC in kernel SVM trained in one-against-rest manner. Similarly to the linear case, our formulation has $n+1$ constraints. The space complexity is $O(m+n)^2$, compared to $O(C^2n^2)$ in SVM.

Classification using kernel classifier: In kernel SVM, the number of kernel evaluations required to classify an input image is equal to the number of support vectors, which is linear in the size of the training set [11]. The number of kernel evaluations when applying kernel hybrid classifier is $(m+n)$, which is typically small and independent of the number of categories one wished to recognize (Section 3 provides an empirical study, showing that beyond a small number of categories the number of $\mathbf{s}_i$'s doesn't increase).

Kernel approximation methods (e.g. [14]) also solve the complexity problem of kernel SVM. However, they trade accuracy for complexity and address only balanced problems, while the main computational burden in one-against-all training is due to the negative class (the "rest" class). The hybrid classifiers proposed here significantly reduce the amount of computations since they replace the constraints on the negative examples with a single probability constraint and do not use negative examples in training.

3 Experiments

We tested the proposed hybrid classifiers in a classification problem, formulated as: given a class, predict the presence/absence of an example of that class in the test image. We do not restrict the test image to a limited (small) number of categories. Our goal is to recognize a given class against a very rich background class. The Caltech 256 [23] data set contains images from 256 diverse classes and thus approximates the set of *all* natural images quite well. The Scene-15 data set [1] contains far fewer classes, but images of scenes are richer than images containing objects (as in the Caltech dataset), thus it also provides an approximation of a rich background. Let us note here that even though these sets are usually used

	SVM	weighted SVM	hybrid
linear	71%	73.9%	73.8%
kernel	83.4%	83.6%	84.0%

Table 2. Average EER. Each number in the table corresponds to the average EER of 256 binary classifiers (of the corresponding type) produced on a test set constructed from 256 categories of Caltech 256.

	SVM (weighted)	hybrid
number of kernel evaluations	600-1000	230
number of parameters in optimization	7680	230
number of constraints in optimization	7680	31
memory usage	450M	4.5M

Table 3. Resource comparison of the kernel SVM (and weighted SVM) and kernel hybrid classifiers for Caltech 256

for categorization, we do not address the multi-class problem. We are interested in binary classification, and so we test hybrid classifiers on all classes from these data sets and report the binary classification results.

Accuracy: For the Caltech 256 data set, we used the image representation provided in [5] for a codebook with 1000 words. We compared the performance of linear and kernel hybrid classifiers to linear and kernel SVMs and their weighted versions trained in one-against-rest manner. We used SPM kernel [1] in the kernel classifiers.

We used 30 images per class as a positive sample. In SVM the negative class consequently contained the rest of the classes, resulting in 7650 samples. For hybrid classifiers we used 1280 samples from the same domain to estimate the mean and covariance matrix of the background. For each classifier we computed the EER of the binary classification in which the positive class contained 25 test samples of the corresponding category and the negative class comprised 25 test images per category for all the other categories (in total 6350 negative examples). We performed training and testing 10 times with random splits into training and test sets and averaged the results. To train the hybrid classifiers we used the CVX optimization package [2]; the SVM was trained using C-SVC option in LIBSVM [3]. All the parameters have been chosen using cross validation. The results are shown in Table 2. Hybrid classifiers outperformed SVM in both the linear and kernel cases, and have accuracy similar to that of weighted SVM, but the classification and training of hybrid classifiers enjoys much lower time and space complexities.

The Scene-15 data set contains only 15 categories. We followed the same test protocol as in Caltech 256 experiment, with 30 training and 30 test samples per category, and used the SPM kernel. The average EER rate of the kernel hybrid classifier was 89.36% and for kernel SVM it was 89.16%.

Probabilistic Model Validation: We tested the validity of the probability constraint for the linear classifier (Eq. 3) and the kernel classifier (Eq. 7) by projecting the test set on the separating boundaries corresponding to learned classifiers (we used random splits of data to create different classifiers), and

[2] http://cvxr.com/cvx/download/

[3] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

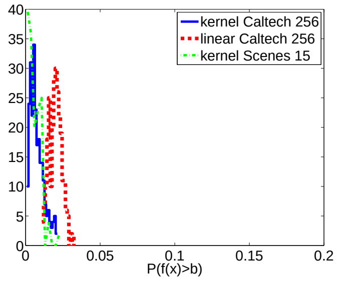

Fig. 4. Histograms of the background empirical probability values in the acceptance region of the hybrid classifiers.

measuring the empirical probability of the background class in their acceptance regions. The histograms of the resulting probabilities are shown in Figure 4. The value of the probability bound in the training was 0.006 for linear classifiers on Caltech-256, 0.004 for kernel classifiers on Caltech-256 and 0.003 for kernel classifiers on Scenes-15. These plots show that the proposed model indeed bounds the probability volume of the background in the acceptance region of the classifiers.

Complexity: In Table 3 the computational and space requirements of hybrid and SVM kernel classifiers are compared on 256 classes, showing a clear advantage of the proposed method.

The number of support vectors and of common $\mathbf{s}_i$'s on the much smaller Scene-15 set was very similar, about 200. The training time of the hybrid classifiers was still faster (we omit the details due to lack of space).

3.1 Scalability of the kernel hybrid classifier

To check the scalability of the classifier vs. the diversity of the background class, we investigate how the number of $\mathbf{s}_i$'s grows as a function of the number of categories which appear in the background class. To this end, we used background classes with increasing numbers of categories from the Caltech 256 data set. For each size of the background class (the x-axis) we found the number of vectors required to reach a fixed reconstruction error (the y-axis); here the reconstruction error was set to 0.005 (i.e., on the average the error in approximating a vector was 0.005 of its norm), and other error thresholds yielded similar behavior. The plot in Figure 5 shows the resulting dependency. The number of vectors is large for a small number of categories and then decreases and hardly changes as the number of categories increases. This behavior – which suggests that the complexity of training and classification does not increase beyond a certain number of categories – can be explained by the fact that we restrict the basis to be a subset of the vectors which need to be approximated. When the background set contains a small number of categories, its diversity is restricted, thus we have to use many vectors to well-approximate the sample set. When the number of categories is large, we can choose fewer – but much better – vectors to approximate

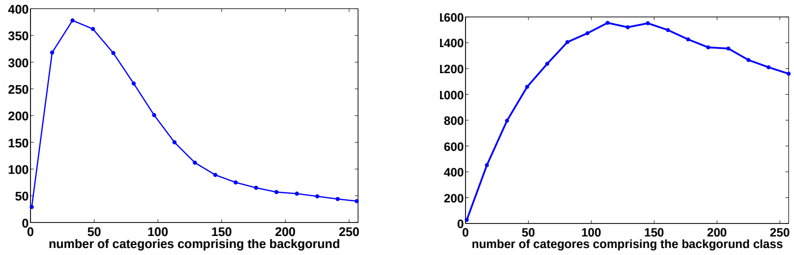

Fig. 5. Left: the relation between the number of vectors required for approximation (with a constant reconstruction error) of unlabeled samples (y-axis) vs. the number of categories these samples were taken from (x-axis). Right: the relation between the effective dimension of a set of unlabeled samples (y-axis) vs. the number of categories these samples were taken from (x-axis).

the set. At some point the sample is rich enough to allow finding vectors that approximate the entire background class, thus adding more categories does not necessitate increasing the basis. A somewhat similar behavior can be observed when looking at the effective dimension of PCA as a function of the number of categories (Figure 5).

4 Conclusions

We proposed to address problems arising when training SVM classifiers in one-against-rest manner, by replacing the negative samples with a distribution representing them. In real visual classification problems the negative class becomes so rich that it can be viewed as a "background" class and it approaches the distribution of all images. We introduced "hybrid" classifiers, which determine a separating hyperplane between positive samples and this probability distribution, and showed that modeling this distribution is simple, as we are only interested in its projections. Further, we estimated the distribution of the background only once, and then used the same model in training the classifiers for all visual classes. This significantly reduced training complexity, compared to SVM.

We tested the proposed approach in binary classification problems in which the negative class comprises many categories and is much larger than the positive class. In addition to performing well, hybrid classifiers proved to be faster to train and apply than SVM.

Future work will concentrate on alternative models for the background, generalizing the proposed formulation to the multi-class problem, and application to other domains, such as text and video classification.

References

1. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006) 2169 – 2178

2. Grauman, K., Darrell, T.: The Pyramid Match Kernel: Efficient Learning with Sets of Features. JMLR 8, 725–760 (2007)
3. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR. (2008) 1–8
4. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR. (2010) 2297–2304
5. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proc. of ICCV. (2009) 221–228
6. Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. In: CVPR. (2011) 1585–1592
7. Huang, Y., Huang, K., Yu, Y., Tan, T.: Salient coding for image classification. In: CVPR. (2011) 1753–1760
8. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. JMLR **5** (2004) 101–141
9. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: In Proc. of ECML. (2004) 39–50
10. Kotsiantis, S., Kanellopoulos, D.: Handling imbalanced datasets: A review. (2006)
11. Steinwart, I.: Sparseness of support vector machines. JMLR **4** (2003) 1071–1105
12. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV **73** (2007) 213–238
13. Keerthi, S.S., Chapelle, O., DeCoste, D.: Building support vector machines with reduced classifier complexity. JMLR **7** (2006) 1493–1515
14. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural svms. Machine Learning **77** (2009) 27–59
15. M.Osadchy, D.Keren: Incorporating the boltzmann prior in object detection using svm. In: CVPR. (2006) 2095–2101
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Computer Vision **60(2)** (2004) 91–110
17. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: ECCV. (2004)
18. Weiss, Y., Freeman, W.T.: What makes a good model of natural images. In: CVPR. (2007) 1–8
19. Zhu, S.C., Mumford, D.: Prior learning and gibbs reaction-diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **19** (1997) 1236–1250
20. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: CVPR. (2005) 860–867
21. Srivastava, A., Lee, A.B., Simoncelli, E.P., c. Zhu, S.: On advances in statistical modeling of natural images. JMIV **18** (2003) 17–33
22. Diaconis, P., Freedman, D.: Asymptotics of graphical projection pursuit. Ann. Statist. **12** (1984) 793–815
23. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. T. R. 7694, California Institute of Technology (2007)
24. Lanckriet, G., Ghaoui, L., Bhattacharyya, C., Jordan, M.: A robust minimax approach to classification. JMLR **3** (2002) 555–582
25. Smola, A.J., Schölkopf, B.: Sparse greedy matrix approximation for machine learning. In: ICML. (2000) 911–918
26. Tsai, D., Jing, Y., Liu, Y., A.Rowley, H., Ioffe, S., M.Rehg, J.: Large-scale image annotation using visual synset. In: ICCV. (2011)