Polyhedral Conic Classifiers for Visual Object Detection and Classification

Hakan Cevikalp
Eskisehir Osmangazi University
Meselik Kampusu, 26480, Eskisehir, Turkey
hakan.cevikalp@gmail.com

Bill Triggs
Laboratoire Jean Kuntzmann
B.P. 53, 38041 Grenoble Cedex 9, France
bill.triggs@imag.fr

Abstract

We propose a family of quasi-linear discriminants that outperform current large-margin methods in sliding window visual object detection and open set recognition tasks. In these tasks the classification problems are both numerically imbalanced – positive (object class) training and test windows are much rarer than negative (non-class) ones – and geometrically asymmetric – the positive samples typically form compact, visually-coherent groups while negatives are much more diverse, including anything at all that is not a well-centred sample from the target class. It is difficult to cover such negative classes using training samples, and doubly so in 'open set' applications where run-time negatives may stem from classes that were not seen at all during training. So there is a need for discriminants whose decision regions focus on tightly circumscribing the positive class, while still taking account of negatives in zones where the two classes overlap. This paper introduces a family of quasi-linear "polyhedral conic" discriminants whose positive regions are distorted L1 balls. The methods have properties and run-time complexities comparable to linear Support Vector Machines (SVMs), and they can be trained from either binary or positive-only samples using constrained quadratic programs related to SVMs. Our experiments show that they significantly outperform both linear SVMs and existing one-class discriminants on a wide range of object detection, open set recognition and conventional closed-set classification tasks.

1. Introduction

Conventional machine learning classifiers such as large-margin discriminants [6,9,4] are intended for "closed set" scenarios [29] in which the class labels are mutually exclusive and exhaustive and every class seen at test time is known during training. These methods try to attribute each test sample to a class even when it has little resemblance to the training samples of any known one – a semantics that is fragile because it ignores the possibility that outliers (samples with no meaningful class) and novel classes (ones not foreseen during training) may occur at test time. In contrast, "open set" methods [29] try to handle these issues by rejecting test samples that do not appear to belong to any of the known training classes. To do this they need to estimate some kind of inlier or validation region for each target class in addition to the conventional inter-class decision boundaries.

Visual object detection should also benefit from discriminants that tightly constrain the positive class. In sliding-window detection, the discrimination problem is highly asymmetric because the positive samples (windows that correctly frame instances of the target class) form a variable-but-coherent appearance class, whereas the negatives (anything at all that is not a well framed object instance) are much more diverse. Moreover the data is highly imbalanced in that there are many more negative (non-object) training and test windows than positive (object) ones. For both reasons it is useful for the discriminant to focus on tightly bounding the positive class whereas conventional discriminants such as Support Vector Machines (SVMs) treat the two classes as though they were equal, interchangeable alternatives. Owing to the many ways in which a window can fail to be a positive most of the SVM support vectors turn out to be 'hard negatives' and with existing feature sets it is not unusual to find that these completely surround the positives in feature space (*c.f.* the scatter plots of projected class densities in [15,28]).

In both applications there is a need for reliable, scalable, asymmetric discriminants that focus on modeling the positive class as a compact, coherent set surrounded by a disparate sea of negatives. The pitfalls of not doing so are illustrated in Fig. 1. This is for a recognition problem in which unforeseen classes occur at run time, but object detectors face similar issues with unforeseen kinds of hard negatives. This paper introduces a new family of quasi-linear discriminants that achieve these goals by using polyhedral decision boundaries based on linear sections through L_1 cones. By supplying tighter bounds on the positive class, this geometry systematically outperforms half-space based

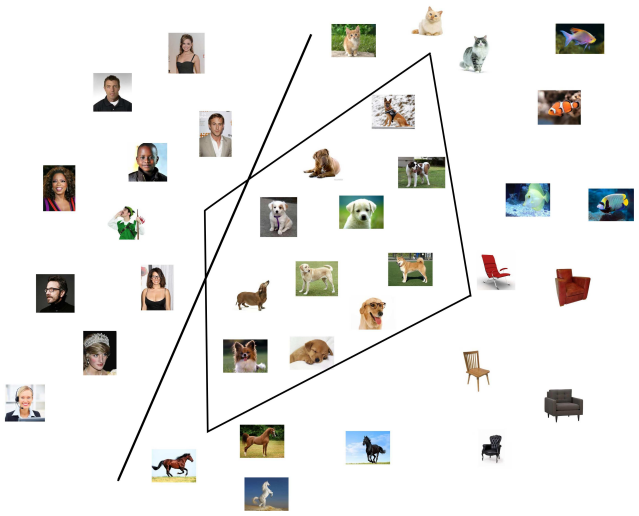

Figure 1. A decision hyperplane returned by an SVM successfully separates its training classes, dogs (positive) and people (negative). However it also assigns instances of novel classes such as cats, horses, fish and chairs to the dog class, sometimes with higher confidence scores than for dogs themselves. The problem is the over-large acceptance region – SVM only tries to separate dogs and people, not to bound the dog class. A tighter (*e.g.* polyhedral or ellipsoidal) decision boundary improves this localization, reducing mis-classifications caused by unforeseen classes and outliers.

decision rules such as linear SVMs in both open set recognition problems and detection problems with unforeseen hard negatives. In fact it often improves the performance even in conventional closed set problems. Training is formulated as an efficient convex program as for linear SVM, and run times are also similar to linear SVM.

Related Work: Several recent works have introduced discriminants or detectors that abandon the symmetrical binary classification framework and adopt loss functions designed to provide tighter modeling of the positive class. These are often called "one class" approaches because most of them can learn a class from positive samples alone, although negatives (if available) can usually be incorporated to help refine the decision boundary[1]. For example, Support Vector Data Description (SVDD) [31] aims to find a closed compact hypersphere that includes the majority of the positive class samples, whereas the Generalized Eigenvalue Proximal Support Vector Machine (GEPSVM) [23] finds a hyperplane that best fits the positive class while avoiding the negatives as far as possible. Other forms of best-fitting-hyperplane classifier are proposed in [18,5,2]. Cevikalp and Triggs [3] use a cascade of convex model based classifiers to progressively cut out a compact, coherent positive region from a broad sea of negative examples for face and person detection. Other approaches such as Additive Kernels [32] and Random features [26] try to approximate kernel classifiers in a fixed-complexity setting by explicitly mapping samples to higher-dimensional spaces that provide nonlinear class separation circumscribing the positive class region.

Another strategy is exemplified by the colorectal cancer detector of Dundar *et al*. [8], which learns polyhedral acceptance regions by jointly optimizing a set of hyperplane classifiers, each designed to classify positives against a subgroup of the negative samples. However the required partitioning of the negative set is both expensive for large-scale problems and problematic if the negatives do not naturally separate into well defined clusters, particularly as the overall performance turns out to be sensitive to both the number and the detailed form of the partitions. There are several other methods for constructing polyhedra that approximately bound a positive class [13,14,1,21,24], but again these either scale poorly with training set size, suffer from local optima or over-fitting, or need ancillary clustering or labeling which makes them unsuitable for large-scale applications. In contrast, our methods have a convex formulation that ensures globally optimal solutions, they scale efficiently to large problems, they do not require negative samples to be clustered, and they resist over-fitting by using a robust margin-based cost function.

2. Polyhedral Conic Classifiers

Our classifiers use the polyhedral conic functions of [13] – essentially projections of hyperplane sections through L_1 cones – to define their acceptance regions for positives. This choice provides a convenient family of compact and convex (for suitable weights) region shapes for discriminating relatively well localized positive classes from broader negative ones. It naturally allows robust margin-based learning, and the number of free parameters remains modest, thus controlling both over-fitting and run times.

The *Polyhedral Conic Functions* and *Extended Polyhedral Conic Functions* respectively have the forms

$$f_{\mathbf{w},\gamma,\mathbf{c},b}(\mathbf{x}) = \mathbf{w}^\top(\mathbf{x}-\mathbf{c}) + \gamma\,\|\mathbf{x}-\mathbf{c}\|_1 - b \qquad \text{(PCF)} \quad (1)$$

$$f_{\mathbf{w},\boldsymbol{\gamma},\mathbf{c},b}(\mathbf{x}) = \mathbf{w}^\top(\mathbf{x}-\mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x}-\mathbf{c}| - b \qquad \text{(EPCF)} \quad (2)$$

Here $\mathbf{x} \in \mathbb{R}^d$ is a test point, $\mathbf{c} \in \mathbb{R}^d$ is the cone vertex, $\mathbf{w} \in \mathbb{R}^d$ is a weight vector and b is an offset. For PCF, $\|\mathbf{u}\|_1 = \sum_{i=1}^d |u_i|$ denotes the vector L_1 norm and γ is a corresponding weight, while for EPCF, $|\mathbf{u}| = (|u_1|,...,|u_d|)^\top$ denotes the component-wise modulus and $\boldsymbol{\gamma} \in \mathbb{R}^d$ is a corresponding weight vector.

Our polyhedral conic classifiers use functions of these forms, with decision regions $f(\mathbf{x}) < 0$ for positives and

[1]The name "one class" emphasizes the methods' origins in density modeling, but it is a misnomer in that negative samples usually can be, often are, and in some formulations must be included during training.

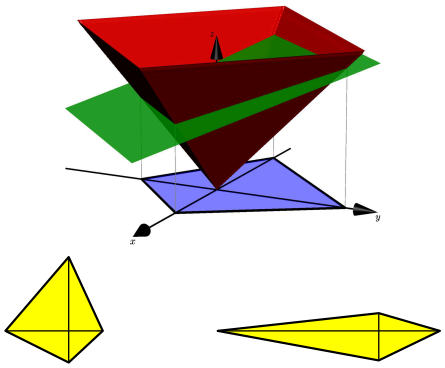

Figure 2. *Top:* For polyhedral conic classifiers, the positive acceptance regions are "kite-like" axis-aligned octahedroids containing the points for which a linear form lies above (within) an L_1 cone. *Bottom:* Typical acceptance regions for 2D classifiers based on *(left)* PCF and *(right)* EPCF decision functions.

$f(\mathbf{x}) > 0$ for negatives. Similarly, our margin based training methods enforce $f(\mathbf{x}) \leq -1$ for positives and $f(\mathbf{x}) \geq +1$ for negatives. In both cases the positive region is essentially a hyperplane-section through an L_1 cone centred at $\mathbf{c}$, specifically the region $\mathbf{x} \in \mathbb{R}^d$ in which the hyperplane $z = \mathbf{w}^\top(\mathbf{x} - \mathbf{c}) - b$ lies above the L_1 cone $z = \gamma\|\mathbf{x} - \mathbf{c}\|_1$ (PCF) or the diagonally-scaled L_1 cone $z = \boldsymbol{\gamma}^\top|\mathbf{x} - \mathbf{c}| = \|\mathrm{diag}(\boldsymbol{\gamma})\,(\mathbf{x} - \mathbf{c})\|_1$ (EPCF). See Fig. 2.

Note that for PCF with $b > 0$, $\gamma > 0$, $\|\mathbf{w}\|_\infty < \gamma$ (where $\|\mathbf{u}\|_\infty = \max_{i=1}^d |u_i|$ is the ∞ norm) and any τ, the region $f(\mathbf{x}) < \tau$ is convex and compact in $\mathbb{R}^d$ and it contains the vertex $\mathbf{c}$. Analogously, for EPCF with $b > 0$, $\boldsymbol{\gamma} > 0$, $|w_i| < \gamma_i, i = 1, ..., d$, and any τ, the region $f(\mathbf{x}) < \tau$ is again convex and compact and it again contains $\mathbf{c}$. It would be straightforward to enforce these inequalities during learning but at present we simply leave the decision regions free to adapt to the training data: compact positive classes naturally tend to produce compact acceptance regions in any case.

Geometrically, under the above constraints the resulting regions are bounded octahedroids with $2d$ vertices, one along each positive and negative coordinate half-axis starting from $\mathbf{c}$. The lines joining opposite vertices thus intersect at $\mathbf{c}$, giving the region a deformed but still axis-aligned octahedral "kite" shape with overall size governed by b. In EPCF the region widths can be scaled independently along each axis, while in PCF they are coupled together but a more limited form of anisotropy is still possible.

To define margin-based classifiers over input feature vectors $\mathbf{x}$ from this, for PCF we augment the feature vector to $\tilde{\mathbf{x}} \equiv \begin{pmatrix} \mathbf{x}-\mathbf{c} \\ \|\mathbf{x}-\mathbf{c}\|_1 \end{pmatrix} \in \mathbb{R}^{d+1}$ and the weight vector to $\tilde{\mathbf{w}} \equiv \begin{pmatrix} -\mathbf{w} \\ -\gamma \end{pmatrix} \in \mathbb{R}^{d+1}$, and let $\tilde{b} = b$. Then the PCF decision function takes the familiar linear SVM form $\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}} + \tilde{b} > 0$ for positives and < 0 for negatives. Similarly, for EPCF we augment the feature vector to $\tilde{\mathbf{x}} \equiv \begin{pmatrix} \mathbf{x}-\mathbf{c} \\ |\mathbf{x}-\mathbf{c}| \end{pmatrix} \in \mathbb{R}^{2d}$ and the weight vector to $\tilde{\mathbf{w}} \equiv \begin{pmatrix} -\mathbf{w} \\ -\boldsymbol{\gamma} \end{pmatrix} \in \mathbb{R}^{2d}$ and again let $\tilde{b} = b$, again giving the SVM form $\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}} + \tilde{b} > 0$ for positives, but now in $2d$ dimensions. The above ∓ 1 margins for PCF and EPCF translate to the familiar ± 1 SVM margins, allowing us to use standard SVM software for maximum margin training[2]. It thus suffices to run the familiar SVM quadratic program on the augmented feature vectors:

$$\begin{aligned} \arg\min_{\tilde{\mathbf{w}},\tilde{b}} \;\; & \tfrac{1}{2}\tilde{\mathbf{w}}^\top\tilde{\mathbf{w}} + C_+ \textstyle\sum_i \xi_i + C_- \sum_j \xi_j \\ \text{s.t.} \;\; & \tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_i + \tilde{b} + \xi_i \geq +1, \;\; i \in I_+, \\ & \tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_j + \tilde{b} - \xi_j \leq -1, \;\; j \in I_-, \\ & \xi_i,\, \xi_j \geq 0, \end{aligned} \tag{3}$$

where the $I_\pm$ are indexing sets for the positive and negative training samples, the ξ's are slack variables for the samples' margin constraint violations, and the $C_\pm$ are corresponding penalty weights.

Inserting the PCF and EPCF feature vectors into the above training procedure respectively gives our *Polyhedral Conic Classifier* (PCC) and *Extended Polyhedral Conic Classifier* (EPCC) methods. Note that despite their ostensibly linear symmetric form, these classifiers are intrinsically asymmetric: they force the positives to lie inside, and the negatives to lie outside, polyhedral conic regions that are typically compact and centred on the positives. Our formulation is robust to overfitting and it scales well because standard SVM technology such as cutting plane methods [12] and fast primal space solvers (*e.g.* [30]) can be used.

The above procedure does not attempt to optimize the position $\mathbf{c}$ of the cone vertex as that would lead to a non-convex problem. It would be possible to optimize for $\mathbf{c}$ at least locally, but here we simply set it to a pre-specified position in the positive training set. The mean, medoid, or coordinate-wise median of the training positives can all be used for this with good results. In our experiments we used the mean. Note that the classifier assigns its highest positive confidence scores to the samples near the cone vertex.

2.1. One-Class EPCC (OC-EPCC)

EPCC usually outperforms both linear SVM and PCC owing to its flexibility, but its positive acceptance regions are bounded and convex only when $|w_i| < \gamma_i$ for all i – *i.e.* when the hyperplane section has a shallower slope than every facet of the L_1 cone. This sometimes fails to hold for feature space dimensions along which the negatives do not surround the positives on all sides. Even though such EPCC acceptance regions are typically still much smaller than the corresponding linear SVM ones, to ensure tighter bounding we would like to enforce $|w_i| < \gamma_i,\; i = 1, \ldots, d$. Moreover, in EPCC the ∓ 1 margin is the only thing that fixes the

[2]This only holds if we agree to ignore the optional compact-convex-region constraints $\|\mathbf{w}\|_\infty < \gamma$ (PCC) or $|w_i| < \gamma_i, i = 1, ..., d$ (EPCC).

Algorithm 1 Stochastic Gradient Based Solver for One-Class EPCC

Initialize

$\mathbf{w}_1$, $\boldsymbol{\gamma}_1$, $T > 0$, $\alpha_0 > 0$, $\epsilon_{\mathbf{w}} > 0$, $\epsilon_{\boldsymbol{\gamma}} > 0$, n_+ is the number of positive examples, n_- is the number of negative examples, $n = n_+ + n_-$

Description:

for $t \in 1, ..., T$ **do**

$\alpha_t \leftarrow \alpha_0/t$;

$\mathbf{w}_{t-1} = \mathbf{w}_t$; $\boldsymbol{\gamma}_{t-1} = \boldsymbol{\gamma}_t$;

for $i \in$ randperm(n) **do**

– Compute sub-gradients

$$\mathbf{g}_{\mathbf{w}}^t = \begin{cases} \frac{\lambda \mathbf{w}}{n} + \frac{\mathbf{x}_i}{n_+}, & \text{if } y_i = 1 \ \& \ y_i(\mathbf{w}_t^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}_t^\top|\mathbf{x}_i - \mathbf{c}| - 1) \geq 0 \\ \frac{\lambda \mathbf{w}}{n} - \frac{\mathbf{x}_i}{n_-}, & \text{if } y_i = -1 \ \& \ y_i(\mathbf{w}_t^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}_t^\top|\mathbf{x}_i - \mathbf{c}| - 1 - \rho_t) \geq 0 \\ \frac{\lambda \mathbf{w}}{n}, & \text{otherwise.} \end{cases}$$

$$\mathbf{g}_{\boldsymbol{\gamma}}^t = \begin{cases} \frac{\mathbf{x}_i}{n_+} - \frac{\mathbf{s}}{n}, & \text{if } y_i = 1 \ \& \ y_i(\mathbf{w}_t^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}_t^\top|\mathbf{x}_i - \mathbf{c}| - 1) \geq 0 \\ -\frac{\mathbf{x}_i}{n_-} - \frac{\mathbf{s}}{n}, & \text{if } y_i = -1 \ \& \ y_i(\mathbf{w}_t^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}_t^\top|\mathbf{x}_i - \mathbf{c}| - 1 - \rho_t) \geq 0 \\ -\frac{\mathbf{s}}{n}, & \text{otherwise.} \end{cases}$$

– Update polyhedral cone parameters

$\mathbf{w}_t \leftarrow \mathbf{w}_t - \alpha_t \mathbf{g}_{\mathbf{w}}^t$

$\boldsymbol{\gamma}_t \leftarrow \boldsymbol{\gamma}_t - \alpha_t \mathbf{g}_{\boldsymbol{\gamma}}^t$

end for

if $\|\mathbf{w}_t - \mathbf{w}_{t-1}\| < \epsilon_{\mathbf{w}}$ & $\left\|\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1}\right\| < \epsilon_{\boldsymbol{\gamma}}$, **break**

end for

overall weight scale and hence prevents a degenerate solution, and negative data is essential for this. To ensure that EPCC works well for open set problems and ones with only positive samples, we need to force its acceptance regions to stay bounded and compact. The acceptance region has width $O(b/\gamma_i)$ along axis i, so we need to ensure that the γ_i can not shrink to zero. The easiest way to achieve this is to replace the ± 1 margin scaling with a $b = 1$ offset scaling and include negative cost penalties on the γ_i and on the geometric width of the new positive-negative margin $[0, 1]$ so that these quantities will tend to increase and hence keep the acceptance region widths small and the sets well separated. This leads to the following "One-Class EPCC" formulation:

$$\begin{aligned} \underset{\mathbf{w},\boldsymbol{\gamma}}{\arg\min} \ & \tfrac{\lambda}{2}\mathbf{w}^\top\mathbf{w} + \tfrac{1}{n_+}\textstyle\sum_i \xi_i + \tfrac{1}{n_-}\sum_j \xi_j - \mathbf{s}^\top\boldsymbol{\gamma} \\ \text{s.t.} \ & \mathbf{w}^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x}_i - \mathbf{c}| - 1 \leq \xi_i, \ \ i \in I_+, \\ & \mathbf{w}^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x}_i - \mathbf{c}| - 1 \geq 1 - \xi_j, \ \ j \in I_-, \\ & \xi_i, \, \xi_j \geq 0. \qquad \text{(OC-EPCC)} \end{aligned} \tag{4}$$

Here λ is a regularization weight for $\mathbf{w}$ and $\mathbf{s} > 0$ is a user-supplied vector of cost penalties for increasing $\boldsymbol{\gamma}$. At present we use the simple stochastic gradient (SG) method given in Algorithm 1 to solve this optimization problem.

3. Experiments

We tested the proposed polyhedral conic classifiers[3] on both synthetic and real datasets for object detection, open set recognition and classical closed-set multiclass discrimination. For comparison we report results for several other linear and quasi-linear methods including SVM, the 1-Sided Best Fitting Hyperplane Classifier (1S-BFHC) of [5], GEPSVM [23], one-class SVM (SVDD) [31], and Additive Kernels method of [32]. In addition, we tested Kernel SVM (KSVM) using a 2nd order polynomial kernel function. For open set recognition problems, we also compared the proposed methods to 1-vs-Set Machine method of [29]. We could not test against the polyhedral classifier of [8] as this software is not available.

We emphasize out that our polyhedral classifiers are best viewed as drop-in replacements for *linear* SVM, which they systematically outperform in the tests below regardless of the application and the features used, with only modest increases in memory usage and run time. Kernel SVMs and similar instance-based methods will typically have even better absolute accuracy but they are usually much too slow for practical use in applications of these kinds, except perhaps as the final stages of classifier cascades with faster early stages such as our methods. This applies to training too: in the face detection study below the final training set size is about 250k and kernel SVM algorithms like Sequential

[3] Our code is available at http://mlcv.ogu.edu.tr/softwarepcc.html.

Method	AP Score (%)
Bayes Optimal	**90.89**
EPCC	86.62
OC-EPCC	84.87
PCC	79.90
Additive Kernels	76.80
SVDD	71.14
GEPSVM	44.25
SVM	22.85

Table 1. Average Precision (%) on the 2D synthetic dataset.

Minimal Optimization [25] struggle to handle datasets of this scale. For this reason it was not practical to include results for kernelized methods in the object detection tests. But, we tested Additive Kernels method of [32] that approximates the kernelized methods. To assess performance we report classification rates or PASCAL VOC style Average Precision (AP) scores [10]. For multi-class problems we used the one-against-rest (OAR) formulation as this worked best for all methods.

3.1. Illustration on Synthetic Data

Fig. 3 illustrates the proposed conic classifiers on a synthetic 2D dataset consisting of random points with the positive class being Gaussian with mean $\left(\begin{smallmatrix}3\\3\end{smallmatrix}\right)$ and axis-aligned standard deviation $\left(\begin{smallmatrix}0.1\\0.9\end{smallmatrix}\right)$, while the negative class is a mixture of Gaussians with the same standard deviation and several means surrounding the positive one. Quantitatively, Table 1 gives empirical Average Precisions for a 250 positive / 750 negative test set sampled from these distributions. The best accuracy is obtained by the statistically-optimal Bayes classifier followed by EPCC. One-class EPCC (OC-EPCC) also does very well even though the version tested here was trained using positive samples alone. Linear SVM fares poorly because the problem is not linearly separable. An Additive Kernel method that explicitly maps the data to an 18-D feature space does better, but not as well as our methods which only use 3 or 4 dimensional embeddings.

3.2. Object Detection Experiments

3.2.1 Face Detection Experiments

To allow a direct comparison of methods we trained several sliding window face detectors that were identical except for the (quasi-)linear classifiers used, testing the proposed PCC and EPCC methods, the 1S-BFHC hyperplane-fitting classifier of [5], linear SVM, and Additive Kernels. For training we used 20 000 frontal upright faces from images collected on the web. For the negative set we randomly sampled 10 000 windows from face-free regions of the same images with complex backgrounds. The subimages were rescaled and cropped to size 35×28 then represented as 620-D LBP+HOG feature vectors. Note that as is often the case in face detection, there are many more positive training samples than feature dimensions.

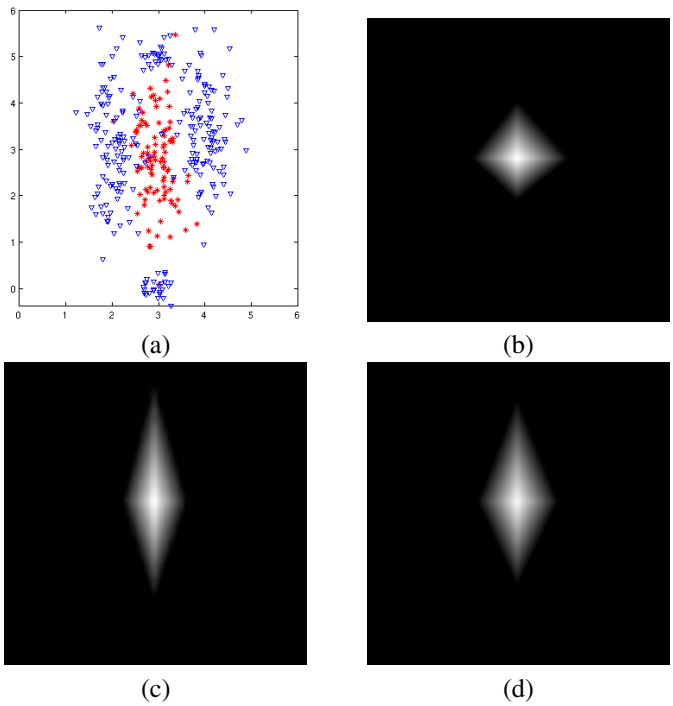

Figure 3. 2D synthetic data set (a) and the decision boundaries returned by (b) PCC, (c) EPCC, (d) OC-EPCC. Brighter pixels correspond to higher scores.

To allow for partial profile pose variation we used spectral clustering to partition the positive training set into three groups and trained a separate classifier of the given type on each group. Each initial detector was used to scan a set of thousands of images to collect additional hard negatives, and the classifiers were retrained to create the final detector. The final size of the training sets is around 250k. The standard sliding window approach of [11] was used for testing, stepping the detector window by 3 pixels horizontally, 4 vertically, and 1.15 in scale and using greedy non-maximum suppression.

We tested the resulting detectors on two datasets, the 2845 image Face Detection Data set and Benchmark (FDDB) [17], and ESOGU Faces[4], which includes 667 high-resolution color images with 2042 annotated frontal faces. Both include faces at a wide range of image positions and scales, complex backgrounds, occlusions and illumination variations.

Table 2 gives Average Precision scores for the above detectors and three publicly available ones: the boosted frontal face detector of Kalal *et al*. [20], the short cascade of Cevikalp & Triggs [3], and the OpenCV Viola-Jones detector

[4] http://mlcvdb.ogu.edu.tr/facedetection.html

Method	FDDB	ESOGU
EPCC	**71.9**	**89.1**
PCC	67.2	78.8
SVM	37.6	47.7
Additive Kernels	55.7	78.7
1S-BFHC	70.5	80.0
Cevikalp-Triggs [3]	**74.1**	87.4
Kalal *et al*. [20]	66.3	79.7
Viola-Jones [33]	67.6	76.2

Table 2. Average Precision (%) for various face detectors on the FDDB and ESOGU Faces datasets.

Method	AP Score (%)	Run Time (s)
EPCC	**85.6**	1.8
PCC	83.6	1.8
SVM	80.4	1.6
Additive Kernels	80.9	19.1
1S-BFHC	78.5	1.6
Felzenszwalb [11]	**86.9**	3.5
Hussain-Triggs [16]	84.1	–
Dalal-Triggs [7]	75.0	–

Table 3. Average Precision (%) on the INRIA Person dataset.

[33]. The scores of the latter detectors are not strictly comparable because they used different non-publicly-available training sets and multi-stage cascades with nonlinear final stages whereas our detectors used only a single linear stage. Nevertheless, the proposed EPCC method still achieved the best result on ESOGU and the second best on FDDB after the method of Cevikalp & Triggs, which also came second on ESOGU. Of the remaining single-stage methods, 1S-BFHC came third on both datasets, PCC followed, and SVM came a poor last, suggesting that simple half-space acceptance regions are inadequate here and that the positive classes need to be bounded more tightly for good results. (EPCC, PCC and 1S-BFHC all constrain them to finite regions). Using Additive Kernels to provide nonlinear decision boundaries is a significant improvement over linear SVM, but its accuracy remains lower than the proposed methods and 1S-BFHC, suggesting that it does not manage to constrain the positive region as well as they do.

3.2.2 Pedestrian Detection Experiments

We trained and tested an analogous series of detectors on the INRIA Person dataset [7], again testing linear EPCC, PCC, 1S-BFHC, SVM, and Additive Kernels with identical settings for each. We used the latent training methodology of Felzenszwalb *et al*. [11], training one symmetric pair of roots without parts. The roots were initialized by applying K-Means clustering to mirror-image pairs. We used HOG features as in [11]: 8×8 pixel cells with window steps of 8 pixels and pyramid scales spaced by a factor of 1.07. For comparison we cite the published results of Felzenszwalb *et al*. [11] (linear latent SVM over HOG, using one symmetric pair of roots, each with 8 parts – 18 filters in total, and bounding box prediction), Hussain & Triggs [16] (a two stage, linear then quadratic cascade based on single root latent SVM over HOG+LBP+LTP), and Dalal & Triggs [7] (simple linear SVM over HOG without latency, multiple roots or parts).

Table 3 shows the resulting accuracies and testing times per image. The EPCC detector achieves the best results among those trained. Owing to its lack of parts it does not quite match the score of the Felzenszwalb multi-root, multi-part detector, but it does outperform the Hussain & Triggs method despite the latter's better features and two stages. PCC also performs well here. Note that despite their gains in accuracy, the run times for EPCC and PCC are very similar to those for SVM (and half of those for [11]), so EPCC is a promising drop-in replacement for linear SVM here. In contrast to the face detection results, Additive Kernels provides little improvement in accuracy over linear SVM even though it is the slowest method tested.

3.3. Visual Object Classification Experiments

3.3.1 Experiments on PASCAL VOC 2007 Dataset

We ran tests on the PASCAL 2007 Visual Object Classification dataset using a popular Convolutional Neural Net feature set. We ran the pre-trained ILSVRC2012 Caffe implementation [19] of the Krizhevsky *et al*. [22] CNN on images resized to 256×256, producing 4096-dimensional feature vectors for each of the methods shown. For comparability with the literature we used stock ILSVRC features without fine-tuning them on the PASCAL dataset. The results are given in Table 4, as PASCAL VOC Average Precision scores. The proposed methods along with Additive Kernels and KSVM achieve the best accuracies for all classes. The best performer is OC-EPCC, trained with samples of both positive and negative classes. It significantly out-performs a linear SVM over the same features, gaining about 4% on average and more than 5% on the classes bottle, bus, chair, dining table, dog, potted-plant, sofa and tv monitor. Additive Kernels improves results over linear SVM, but it uses a three-times larger feature space. GEPSVM was the worst performer here.

3.3.2 Experiments on Multi-Class Classification Datasets

We tested our methods on three conventional closed-set multiclass discrimination problems: Caltech-256 visual object classification, the Letter Recognition (LR) and Multiple Features (MF) pixel value datasets from the UCI repos-

Methods	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	TV Monitor	Average
OC-EPCC	85.1	79.7	82.9	81.3	36.4	**69.5**	83.2	80.7	**57.7**	61.6	**70.0**	**79.9**	**83.2**	74.0	**90.4**	51.0	73.4	58.6	84.5	66.7	**72.5**
EPCC	**87.2**	**80.0**	**83.3**	80.9	35.9	66.5	**83.4**	80.9	56.5	59.4	68.7	78.5	82.6	73.8	90.1	49.7	71.3	57.1	86.5	66.6	72.0
PCC	86.3	79.0	83.0	80.5	35.3	65.8	**83.4**	80.2	56.1	60.3	68.0	77.2	81.8	73.3	89.8	47.9	70.8	55.6	85.9	66.4	71.3
SVM	87.0	75.7	81.7	80.4	31.2	63.6	80.4	79.1	47.1	58.1	64.2	74.0	81.0	73.0	87.4	41.3	68.5	50.6	86.3	61.4	68.6
KSVM	83.9	77.3	82.2	**81.8**	**38.7**	**69.5**	81.9	79.6	57.5	60.2	69.8	79.2	79.1	71.2	89.0	**52.6**	**73.8**	**59.3**	84.8	**69.7**	72.1
Additive Kernels	86.6	78.5	83.0	81.2	35.6	68.0	82.0	**81.5**	51.0	**63.1**	65.5	76.2	82.7	**74.9**	88.7	47.3	72.7	54.0	**86.7**	64.2	71.2
1S-BFHC	85.9	74.0	79.9	77.4	30.3	63.0	78.5	78.0	46.2	56.6	62.0	72.0	79.7	71.9	83.2	39.2	63.1	51.0	84.4	59.5	66.8
GEPSVM	36.2	21.9	45.1	26.4	10.3	27.0	34.1	21.9	29.0	39.9	32.0	22.2	32.0	19.6	53.9	15.4	27.2	14.3	39.0	25.8	28.7
SVDD	65.5	32.4	25.0	26.0	21.5	31.2	37.1	48.7	28.3	23.1	17.7	25.5	39.3	31.8	58.8	12.3	21.2	18.5	59.2	25.5	32.4

Table 4. Average Precision scores (%) on PASCAL VOC 2007 classification datasets.

itory. The LR dataset includes 26 classes and 20K samples whereas MF includes 10 classes and 2000 samples. For Caltech-256 we followed the standard protocol, picking 30 training and 30 test images from each class and also testing with reversed test and training roles. Fisher Vector (FV) features were used with the setup of [27]. Specifically, we extracted approximately 10K descriptors per image from 24×24 patches sampled on a regular grid every four pixels, at 5 image scales. The descriptor dimension was reduced to 80 using Principal Component Analysis (PCA). We used 6×10^6 descriptors to learn the PCA projections and the 256-component Gaussian mixture model (GMM) components, leading to a final FV image descriptor dimension of about 164k. For the LR and MF datasets we used 10-fold cross-validation to evaluate the performance.

The results are summarized in Table 5, in terms of simple classification accuracies. The proposed EPCC method achieved the best accuracy on MF whereas the Additive Kernels and KSVM methods gave the best accuracies on Caltech-256 and LR. However note that Additive Kernels used significantly longer feature vectors than EPCC: 3 times the original input space dimension for Caltech-256, and 5 times for LR and MF. In a similar manner the dimensionality of the new sample space is $\binom{d+2-1}{2}$ when a 2nd order polynomial kernel function is used. Although they were beaten by Additive Kernels and KSVM on these two datasets, the proposed methods did significantly outperform the remaining (quasi-)linear classifiers that were tested. The differences were especially large for LR, where EPCC had an error rate 16% lower than SVM, the best existing linear method tested. Also note that for Caltech-256, PCC significantly outperforms SVM even though it has just one additional feature (of 164k). This shows that it is the positive-class-bounding polyhedral cone geometry that is providing the improvement here, not the features used, and also that our training methods can gracefully handle very large feature vectors.

Method	Caltech-256	LR	MF
EPCC	40.1 ± 0.6	76.0 ± 1.2	$\mathbf{96.3} \pm 1.2$
PCC	40.4 ± 0.7	65.5 ± 0.9	94.5 ± 1.4
SVM	37.6 ± 0.7	59.8 ± 1.7	93.9 ± 1.1
KSVM	38.8 ± 0.7	$\mathbf{89.3} \pm 0.9$	95.9 ± 0.7
Additive Kernels	$\mathbf{42.6} \pm 0.7$	81.9 ± 1.5	95.4 ± 1.4
1S-BFHC	38.3 ± 1.0	25.3 ± 0.8	93.8 ± 1.5
GEPSVM	13.3 ± 0.6	30.5 ± 1.1	53.8 ± 4.0
SVDD	9.9 ± 0.2	37.5 ± 1.6	80.1 ± 3.5

Table 5. Classification rates (%) for the closed-set multi-class discrimination experiments.

Method/Class	Leopard	Face	Airplane	Chandelier
OC-EPCC	$\mathbf{76.6} \pm 2.9$	$\mathbf{70.0} \pm 2.8$	13.6 ± 1.5	6.0 ± 1.2
EPCC	69.8 ± 7.9	69.7 ± 2.8	15.5 ± 2.8	5.6 ± 0.6
PCC	65.3 ± 7.6	68.1 ± 3.3	$\mathbf{15.7} \pm 3.2$	5.4 ± 0.8
1-vs-Set Machine	76.5 ± 6.8	60.2 ± 3.8	12.0 ± 0.9	4.9 ± 1.1
1S-BFHC	62.6 ± 12.7	59.1 ± 3.1	12.4 ± 1.8	4.8 ± 0.7
SVM	63.2 ± 13.1	61.5 ± 4.3	12.0 ± 1.6	4.8 ± 0.5
KSVM	68.7 ± 5.8	63.0 ± 2.2	9.3 ± 1.1	$\mathbf{7.2} \pm 0.9$
GEPSVM	2.0 ± 1.0	8.4 ± 7.8	6.8 ± 1.1	2.6 ± 0.5
SVDD	3.6 ± 0.8	2.0 ± 0.4	3.6 ± 0.5	3.3 ± 0.7

Table 6. AP scores (%) for the open set visual object classification experiment.

3.4. Experiments on Open Set Recognition

3.4.1 Open Set Visual Object Classification

Here we use the 212 class open set recognition dataset and protocol from [29]. This setting reflects real-world classification tasks in which the test set may include samples from classes not present during training. Images from both Caltech-256[5] and ImageNet[6] were used to create the

[5]http://www.vision.caltech.edu/Image_Datasets/Caltech256
[6]http://www.image-net.org

data. The images are represented using HOG and LBP-like features. For each positive class, a training set is created from Caltech-256 images by randomly selecting 30 positive samples from the class and 30 negative samples from other classes (5 samples each from 6 randomly chosen other classes). For testing, 30 new images are selected from the positive class and 6330 negative ones are selected from the six classes used during training and from 206 random classes chosen from ImageNet (see [29] for details). This procedure is repeated 5 times, with the final accuracies being averages over the 5 trials. In our experiments we only used 4 positive classes: leopards, faces, airplanes and chandelier. (These are the only classes for which the best-performing methods achieve AP greater than 5% with the features provided). For all methods including OC-EPCC, the training used both positive and negative samples.

The results are given in Table 6. We report AP scores computed from Precision-Recall curves instead of the Classification Rates used in [29] because we believe that the latter may not reflect the attainable recognition performance in open set scenarios. Moreover, open set methods are expected to reject samples from unknown classes and the thresholds for this are most easily obtained from Precision-Recall curves. As can be seen, both PCC and EPCC outperform SVM and 1S-BFHC, and one-class EPCC (OC-EPCC) further improves the accuracy except for the airplane class. The difference is especially large for the leopard class: OC-EPCC has AP nearly 7% higher than the EPCC. It should be noted that the proposed methods significantly outperform even KSVM except for Chandelier class. 1-vs-Set Machine achieves a very high accuracy (similar to OC-EPCC) for Leopard class, but its accuracies are low compared to the best computed accuracies for the remaining classes. GEPSVM and SVDD are the worst performing methods.

3.4.2 Open Set USPS Digit Recognition

Next we give results for an open set recognition experiment based on the USPS Digits dataset. This contains 9298 16×16 gray-scale images of hand-written digits, with 7291 for training and validation and the remaining 2007 for testing. To make the problem harder we use the raw gray-scale pixel values as features without any pre-processing or feature extraction. For open set recognition we randomly choose three classes and train the methods on the training samples from these classes alone. In contrast, testing uses samples from all 10 classes. We compute AP scores from the Precision-Recall curves for the 3 classes, take the average of these, repeat the whole procedure over 10 trials, and report the final averaged average AP score. The results are given in Table 7. The OC-EPCC classifier again achieves the best results, followed by EPCC, KSVM and PCC. All of the proposed methods significantly outperform linear SVM, while SVDD is again the worst performer.

Method	**AP Score (%)**
OC-EPCC	**82.2**
EPCC	80.6
PCC	76.2
1-vs-Set Machine	64.5
1S-BFHC	66.0
SVM	61.9
KSVM	76.4
GEPSVM	40.5
SVDD	11.3

Table 7. AP Scores (%) for the open set USPS experiment.

4. Summary and Conclusions

This study argues that in open set object recognition and sliding window object detection problems, it is advantageous to use asymmetric classifiers that focus on producing compact, well-constrained decision regions for the positive (target object) class. To this end we introduced PCC, EPCC and OC-EPCC, a family of robust scalable maximum margin learning methods whose positive acceptance regions are planar sections through L_1 cones. For appropriate parameter settings these methods give compact, convex acceptance regions that tightly constrain the extent of the positive class. A feature vector augmentation allows PCC and EPCC to be trained using standard linear SVM software, while OC-EPCC is currently trained using an analogous stochastic gradient descent method. We tested these methods with good results on a range of object detection, open set recognition and classical closed-set discrimination tasks. The detection and open set recognition results were particularly promising, giving significant improvements across the board against comparable (quasi-)linear classifiers including SVMs and several one-class approaches. Overall, we believe that our methods will prove to be useful drop-in replacements for linear discriminants such as SVMs in many current visual object detection and classification tasks.

As future work, we note that our formulation is not limited to polyhedral acceptance regions. Any other norm – or even an arbitrary convex function – could be used in place of the L_1 norm. For example, using the unsquared L_2 norm $\|\cdot\|$ to construct the augmented vector, $\tilde{\mathbf{x}} \equiv \begin{pmatrix} \mathbf{x}-\mathbf{c} \\ ||\mathbf{x}-\mathbf{c}|| \end{pmatrix} \in \mathbb{R}^{d+1}$, would give a PCC-style classifier that returned ellipsoidal decision regions and that was distinct from, and probably more robust than, existing "one-class $\|\cdot\|^2$ methods" such as SVDD.

Acknowledgments: This work was funded in part by the Scientific and Technological Research Council of Turkey (TUBİTAK) under Grant number EEEAG-116E080.

References

[1] A. Bagirov, J. Ugon, and D. Webb. An efficient algorithm for the incremental construction of piece-wise linear classifier. *Information Systems*, 36:782–790, 2011.

[2] H. Cevikalp. Best fitting hyperplanes for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14 DOI:10.1109/TPAMI.2016.2587647, 2017.

[3] H. Cevikalp and B. Triggs. Efficient object detection using cascades of nearest convex model classifiers. In *CVPR*, 2012.

[4] H. Cevikalp and B. Triggs. Hyperdisk based large margin classifier. *Pattern Recognition*, 46:1523–1531, 2013.

[5] H. Cevikalp, B. Triggs, and V. Franc. Face and landmark detection by using cascade of classifiers. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

[6] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

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

[8] M. M. Dundar, M. Wolf, S. Lakare, M. Salganicoff, and V. C. Raykar. Polyhedral classifier for target detection: A case study: Colorectal cancer. In *International Conference on Machine Learning*, 2008.

[9] S. Ertekin, L. Bottou, and C. L. Giles. Nonconvex online support vector machines. *IEEE Transactions on PAMI*, 33:368–381, 2011.

[10] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge. *Int. J. Computer Vision*, 88(2):303–338, 2010.

[11] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, 32(9), Sept. 2010.

[12] V. Franc and S. Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *The Journal of Machine Learning Research*, 10:2157–2192, 2009.

[13] R. N. Gasimov and G. Ozturk. Separation via polyhedral conic functions. *Optimization Methods and Software*, 21:527–540, 2006.

[14] M. K. H. Tenmoto and M. Shimbo. Piecewise linear classifiers with an appropriate number of hyperplanes. *Pattern Recognition*, 31:1627–1634, 1998.

[15] S. Hussain. *Machine learning methods for visual object detection*. PhD thesis, Laboratoire Jean Kuntzmann, 2011.

[16] S. Hussain and B. Triggs. Feature sets and dimensionality reduction for visual object detection. In *BMVC*, 2010.

[17] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[18] Jayadeva, R. Khemchandani, and S. Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelliegence*, 29:905–910, 2007.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[20] Z. Kalal, J. Matas, and K. Mikolajczyk. Weighted sampling for large-scale boosting. In *BMVC*, 2008.

[21] A. Kantchelian, M. C. Tschantz, L. Huang, P. L. Barlett, A. D. Joseph, and J. D. Tygar. Large-margin convex polytope machine. In *NIPS*, 2014.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] O. L. Mangasarian and E. W. Wild. Multisurface proximal support vcetor machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelliegence*, 28:69–74, 2006.

[24] N. Manwani and P. S. Sastry. Learning polyhedral classifiers using logistic function. In *Asian Conference on Machine Learning*, 2010.

[25] J. C. Platt. Fast training of support vector machines using sequential minimal optimization, 1998. Advances in Kernel Methods-Support Vector Learning, Cambridge, MA, MIT Press.

[26] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[27] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 34:1704–1716, 2013.

[28] A. Satpathy, X. Jiang, and H. L. Eng. Human detection by quadratic classification on subspace of extended histogram of gradients. *IEEE Transactions on Image Processing*, 23:287–297, 2014.

[29] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult. Towards open set recognition. *IEEE Transactions on PAMI*, 35:1757–1772, 2013.

[30] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-grdient solver for SVM. In *International Conference on Machine Learning*, 2007.

[31] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.

[32] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:480–492, 2012.

[33] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.