The One-Shot Similarity Kernel

Lior Wolf[1] Tal Hassner[2] Yaniv Taigman[1,3]

[1] The Blavatnik School of Computer Science, Tel-Aviv University, Israel

[2] Computer Science Division, The Open University of Israel

[3] face.com

Abstract

The One-Shot similarity measure has recently been introduced in the context of face recognition where it was used to produce state-of-the-art results. Given two vectors, their One-Shot similarity score reflects the likelihood of each vector belonging in the same class as the other vector and not in a class defined by a fixed set of "negative" examples. The potential of this approach has thus far been largely unexplored. In this paper we analyze the One-Shot score and show that: (1) when using a version of LDA as the underlying classifier, this score is a Conditionally Positive Definite kernel and may be used within kernel-methods (e.g., SVM), (2) it can be efficiently computed, and (3) that it is effective as an underlying mechanism for image representation. We further demonstrate the effectiveness of the One-Shot similarity score in a number of applications including multi-class identification and descriptor generation.

1. Introduction

The ability to compare two signals and decide if they represent the same information is key to many Computer Vision and Pattern Recognition tasks. For example, a typical Computer Vision problem is "Are two images portraying the same object?" where different viewing conditions, poses and other sources of variability are considered. There are, of course, many methods for estimating such similarities. Some use metrics (either "learned" from examples or otherwise) to measure the distance between two signal vectors. Others, such as the Discriminative Learning approaches, use labeled training sets to build models describing different signal classes.

Recently, the One-Shot Similarity (OSS) measure was presented as an alternative approach [38]. OSS compares two vectors by considering a single, *unlabeled*, *negative* example set, and using it to learn what signals are considered "different". Given two vectors, their OSS score is computed by first learning a model for each vector, discriminating it from this set of negative examples. These models are then used to determine if each vector shares the same label as its counterpart or belongs with the set of negative examples. The average of these two prediction scores is the OSS score for the two vectors.

This approach has several advantages. On one hand, unlike straightforward measurement of distances between vectors, the OSS uses Discriminative Learning to explicitly build models which underscore the differences between them. On the other hand, unlike standard Discriminative Learning approaches, labeled training data is not required (but may well be useful if available). Moreover, the discriminative models are produced *per* the vectors being compared and so are often better suited to comparing them.

The OSS has already shown promising results in the image "pair-matching" problem [38]. Here we further analyze this measure and provide the following contributions:

1. We show that the OSS measure, when built on top of a variant of the Linear Discriminant Analysis classifier, is a conditionally positive definite (CPD) kernel. It can therefore be used directly with translation invariant kernel methods such as SVM and kernel PCA, or give rise to a positive definite kernel (PD) that can be used with any kernel-method.

2. We demonstrate how, employing pre-processing, the OSS scores can be computed efficiently.

3. We show how the OSS score may be used as a mechanism for image representation, by applying it within a random sub-window scheme.

4. Finally, we provide empirical results, both quantitative and qualitative, demonstrating the performance of OSS scores for image pair-matching as well as multi-class identification problems.

In the next section we review related methods. In Section 2 we formally define the OSS measure and demonstrate how it may be computed efficiently. We analyze the OSS similarity in Section 3. Experiments are reported in Section 4. We conclude in Section 5.

1.1. Related work

The literature on similarity functions, their design and applications, is extensive. Some of the similarity measures proposed in the past have been hand crafted (e.g., [2, 40]). Alternatively, a growing number of authors have proposed tailoring the similarity measures to available training data by applying learning techniques (e.g., [3, 9, 16, 34, 37, 39]). In all these methods testing is performed using models (or similarity measures) learned beforehand, whereas the OSS

score proposed here learns discriminative models exclusive to the vectors being compared.

We note that our algorithm in effect combines multiple classifiers. It does so, however, in a manner which is somewhat different than algorithms such as "Bagging" (e.g., [6]): in our case, classifiers are trained not with a subset of the original training set, but with a combination of one training example and a separate auxiliary set.

In this paper we show that for the existing implementation of the OSS in the literature [38], which uses LDA, the OSS measure is not a positive definite (PD) kernel, however, it empirically behaves as a conditionally positive definite kernel (CPD) [32], and a simple variant of it is a CPD. CPD kernels are important as they may be used directly, or once converted to PD kernels, in a wide family of classification and clustering tools [33]. In particular, popular classification methods such as SVM may use conditionally positive definite kernels as substitutes for vector inner-products as measures of similarity.

We present results for image classification using randomized sub-windows with OSS as the similarity score between image representations. This approach is motivated by existing methods for estimating the visual similarity of images. The method of [29], for example, uses Randomized Decision Trees [14] and Support Vector Machines (SVM) [8] to measure the similarity of two images. In their framework, image patches are selected at random from one image and then the most similar nearby patches are selected from the other image. An SVM classifier is then used to determine if the two images match by aggregating over all selected patches the output of pre-trained random decision trees. A similar method was used also for image classification in other domains [26, 27].

2. Definition of the OS similarity

The One-Shot Similarity measure draws its motivation from the growing number of so called "One-Shot Learning" techniques; that is, methods which learn from one or few training examples (see for example [1, 11, 12, 17]).

To compute the OSS score for two vectors x_i and x_j we use a set $\mathbf{A}$ of "negative" training examples. These are vectors which have different labels (identities) from those we wish to compare. The *symmetric* One-Shot score is then computed as follows (see also Fig. 1). We first learn a model using $\mathbf{A}$ as a set of negative examples and x_i as a single positive example (hence the term "One-Shot"). We then use this model to classify x_j, obtaining a classification confidence score, `Score1`. The particular value of this score depends on the classifier used. Using linear SVM as a classifier, for example, this value can be the signed distance from the separating hyperplane. Intuitively, this value gives us a measure of how likely x_j is to belong to the same class as x_i, given the one-shot training. This exercise is then repeated, switching the roles of x_i and x_j, providing us with a second prediction score, `Score2`. Finally, the One-Shot similarity score of x_i and x_j given $\mathbf{A}$ is the average of these two scores. A *one-sided* score can be computed by learning a classifier for only one of the two vectors.

```
One-Shot-Similarity(x_i, x_j,  A) =
     Model1 = train(x_i, A)
     Score1 = classify(x_j, Model1)

     Model2 = train(x_j, A)
     Score2 = classify(x_i, Model2)

     return ½(Score1+Score2)
```

Figure 1. Computing the symmetric One-Shot Similarity score for two vectors, x_i and x_j, given a set $\mathbf{A}$ of negative examples.

The computational cost of computing the OSS score depends on the particular classifier used. In particular, as we show below, the special structure of the learning problem can be exploited to reduce the computational complexity.

2.1. OS similarity with LDA

The Fisher Discriminant Analysis (FDA or LDA) [13, 15], is a well known learning algorithm that has been regularized, e.g. [24], to deal with small sample size, and kernelized [28] to deal with non-linear decision rules.

We employ LDA within the OSS scheme since it can be efficiently computed by exploiting the fact that the set $\mathbf{A}$ of negative examples is used repeatedly, and that the positive class, which contains just one element, does not contribute to the within class covariance matrix. We can consequently show that LDA based OSS can be directly computed in $O(d^2)$ operations, where d is the dimension of the vector space in which examples reside. Moreover, $O(d^2)$ operations are only required on the first time a vector is compared to another. Repeated comparisons are $O(d)$, i.e., at the order of magnitude of simple correlations.

We focus on the binary LDA case, which is the one of importance here. Let $p_i \in \mathbb{R}^d, i = 1, 2, ..., m_1$ be a set of positive training examples, and let $n_i \in \mathbb{R}^d, i = 1, 2, ..., m_2$ be a set of negative training examples. Let μ be the average of all points and μ_p (resp. μ_n) be the average of the positive (negative) training set. Two matrices are then considered [10], S_B which measures the covariance of the class centers, and S_W which is the sum of the covariance matrices of each class.

$$\begin{aligned} S_B = & \quad (\mu_p - \mu_n)(\mu_p - \mu_n)^\top \\ S_W = & \quad \frac{1}{m_1+m_2}\sum_{i=1}^{m_1}(p_i - \mu_p)(p_i - \mu_p)^\top + \\ & \quad \frac{1}{m_1+m_2}\sum_{i=1}^{m_2}(n_i - \mu_n)(n_i - \mu_n)^\top \end{aligned} \quad (1)$$

The LDA algorithm computes a projection v which maximizes the Raleigh quotient:

$$v = \arg\max_v \frac{v^\top S_B v}{v^\top S_W v} \tag{2}$$

In the two class case, v is easily determined since S_B is a rank one matrix whose columns and rows are spanned by $(\mu_p - \mu_n)$, we therefore get:

$$v = \frac{S_W^+(\mu_p - \mu_n)}{\|S_W^+(\mu_p - \mu_n)\|} \tag{3}$$

Note that we use the pseudo-inverse S_W^+ instead of the inverse S_W^{-1} in order to deal with cases where the within-class covariance matrix is not full rank. This is equivalent to requiring in Eq. 2 that v be spanned by the training vectors.

Once the projection direction has been computed, the classification of a new sample $x \in \mathbb{R}^d$ is given by the sign of $v^\top x - v_0$, where v_0 is the bias term (see below).

In the One-Shot case, the positive set is composed of one vector $p_1 = x_i$ (x_j), and the negative set is A. The positive set does not contribute to the within class covariance, and S_W (and hence S_W^+) depends only on A, is constant, and need only be computed once. Similarly, the average of the negative set μ_n, which we refer to below as μ_A to emphasize its dependent on A, need only be computed once.

Previously [38], OSS was computed using the midpoint between the projected means of the classes as the bias value. i.e., in the first stage of the OSS computation (Fig. 1), where x_i is used as the positive set, and A as the negative set of LDA, we get:

$$v_0 = v^\top \frac{x_i + \mu_A}{2}. \tag{4}$$

This specific choice is somewhat arbitrary. While it can be justified by employing assumptions on the class prior distributions and the variance of the positive class, these assumptions are unlikely to hold. However, in order to eliminate the need for further parameters, we adopt these bias terms for our experiments as well.

To summarize, when using LDA as the underlying classifier, the One-Shot Similarity between samples x_i and x_j given the auxiliary set A becomes:

$$\frac{(x_i - \mu_A)^\top S_W^+(x_j - \frac{x_i+\mu_A}{2})}{\|S_W^+(x_i - \mu_A)\|} + \frac{(x_j - \mu_A)^\top S_W^+(x_i - \frac{x_j+\mu_A}{2})}{\|S_W^+(x_j - \mu_A)\|} \tag{5}$$

The overall complexity for the one shot similarity per pair is thus $O(d^2)$ once the (pseudo) inverse S_W has been computed. Note that if similarities are computed for the same point repeatedly, one can factor the positive definite $S_W^+ = HH^\top$ and pre-multiply this point by the factor H.

2.2. OS similarity with free-scale LDA

The LDA formalization is based on a projection direction given v in Eq. 3. The free-scale LDA is a simplified version in which the projection is replaced with a dot product with the unnormalized vector $v = S_W^+(\mu_p - \mu_n)$. The bias term v_0 is computed similarly to LDA (Eq. 4 above).

For binary classification problems, LDA and free-scale LDA produce similar results (the sign does not change). However, in the computation of OSS the pre-threshold projection value plays a role, and the similarities based on the two classifiers differ. Specifically, similarities will be larger in magnitude (positive or negative) if $x_i - \mu_A$ has a large magnitude, i.e., in cases where x_i is distant from μ_A. This agrees with the intuition that similarities are more pronounce where the one-sample positive class (x_i) is well-separated from the negative class (the columns of A).

The OSS based on free-scale LDA is expressed as:

$$\begin{aligned}(x_i - \mu_A)^\top S_W^+(x_j - \frac{x_i + \mu_A}{2}) + \\ (x_j - \mu_A)^\top S_W^+(x_i - \frac{x_j + \mu_A}{2})\end{aligned} \tag{6}$$

2.3. OS similarity with SVM

The computation of OSS based on SVM also benefits from the special structure of the underlying classifications. Consider the hard-margin SVM case. In this case the single positive example becomes a support vector. The maximum margin will be along the line connecting this point and the closest point in set A, which serves as the negative set. Therefore, the two SVM computations per similarity computation are trivial once the points closest to x_i and x_j in A are identified. Such simple geometric arguments, which are used in some modern SVM solvers, e.g., [4], fail to work in the soft margin case, and efficient computation for this case is left for future research.

3. Analysis of the OS similarity

OSS is defined, given a binary classification algorithm and a set $\mathbf{A}$, between every two vectors of appropriate dimensionality. From its construction it is symmetric, but is it positive definite (PD)? We next show that:

1. The One-Shot similarity is not generally a PD or a Conditional positive definite (CPD) Kernel.
2. For free-scale LDA, OSS is indeed a CPD kernel.
3. Thus, for free-scale LDA the exponent of the OSS is a PD Kernel.

We begin with the basic definitions.

Definition 1 (Positive definite kernel)**.** *Let X be a non-empty set. A symmetric function $k : X \times X \to \mathbb{R}$ for which for all $x_i \in X$, $m \in \mathbb{N}$, and $c \in \mathbb{R}^m$ satisfies $c^\top Kc \geq 0$, where $K \in \mathbb{R}^{m\times m}$ is the matrix $K_{ij} = k(x_i, x_j)$, is called a positive definite (PD) kernel.*

The definition of a conditionally positive definite kernel (CPD) [32] places an additional restriction on the vector c:

Definition 2 (Conditionally positive definite kernel)**.** *A symmetric function $k : X \times X \to \mathbb{R}$ for which for all $m \in \mathbb{N}$, $x_i \in X$, and vectors $c \in \mathbb{R}^m$ such that $\sum_{i=1}^m c_i = 0$, we have $c^\top Kc \geq 0$, where K is as in Definition 1 above, is called a conditionally positive definite (CPD) kernel.*

CPD kernels can be used directly in translation invariant kernel-methods [32], specifically within SVM [5]. Moreover, as Prop. 1 below shows, PD kernels can be readily constructed from CPD kernels.

Proposition 1 (Theorem 2.2 of [7])**.** *$k(a, b)$ is a conditionally positive definite kernel iff $k'(a, b) = \exp(tk(a, b))$ is a positive definite kernel for all $t > 0$.*

OSS similarity based on SVM or LDA is not a PD nor a CPD kernel. This is verified by numeric simulations that show that the point-wise exponent of the resulting similarity does not give rise to a PD kernel. However, OSS based on free-scale LDA is a CPD kernel as shown below.

Proposition 2. *The OSS based on free-scale LDA (Eq. 6) is a CPD kernel.*

Proof. This similarity measure can be broken down into terms and rearranged as a sum of two kernels $K1$ and $K2$, which are defined as

$$\begin{aligned} K1 &= (x_i - \mu_A)^\top S_W^+ (x_j - \mu_A) \\ K2 &= (x_i - \mu_A)^\top S_W^+ (x_j - \mu_A) - \\ &\quad \tfrac{1}{2}((x_i - \mu_A)^\top S_W^+ (x_i - \mu_A) + (x_j - \mu_A)^\top S_W^+ (x_j - \mu_A)) \end{aligned}$$

$K1$ is a PD kernel, and thus a CPD kernel. $K2$ has the form

$$\begin{aligned} &s(x_i - \mu_A, x_j - \mu_A) - \\ &\frac{1}{2}s(x_i - \mu_A, x_i - \mu_A) - \frac{1}{2}s(x_j - \mu_A, x_j - \mu_A) \\ &= s(x_i, x_j) - \frac{1}{2}s(x_i, x_i) - \frac{1}{2}s(x_j, x_j) \end{aligned} \tag{7}$$

for the positive definite kernel $s(a, b) = a^\top S_W^+ b$.

For all $m \in \mathbb{N}$, $x_i \in \mathbb{R}^d$, and vectors $c \in \mathbb{R}^m$ such that $\sum_{i=1}^m c_i = 0$, we get

$$\begin{aligned} &\sum_{i,j=1}^m c_i c_j (s(x_i, x_j) - \frac{1}{2}s(x_i, x_i) - \frac{1}{2}s(x_j, x_j)) = \\ &\sum_{i,j=1}^m c_i c_j s(x_i, x_j) - \frac{1}{2}\sum_{j=1}^m c_j \sum_{i=1}^m c_i s(x_i, x_i) - \\ &\frac{1}{2}\sum_{i=1}^m c_i \sum_{j=1}^m c_j s(x_j, x_j) = \sum_{i,j=1}^m c_i c_j s(x_i, x_j) \geq 0 \end{aligned}$$

Thus, $K2$ and the sum of $K1$ and $K2$ are CPD kernels. □

Note that the derivation in Prop. 2 provides more insights into the structure of the free-scale LDA based OSS. In the vector space that contains the original vectors translated such that $\mu_A = 0$, and in which the inner product $\langle x_i, x_j \rangle' = \langle S_W^+ x_i, x_j \rangle$ is employed, this similarity is given by the inner product (K_1) minus the corresponding squared distance (K_2).

4. Experiments

We next demonstrate not only the effectiveness of the OSS measure, but also its versatility. We apply OSS as a kernel basis used with SVM, directly as a similarity measure, and as the building block for an image representation.

4.1. Insect species identification

We test the performance of the OSS as an SVM kernel for multi-label image classification. Our goal here is to identify the species of an insect appearing in an image. We used the Moorea Biocode insect image collection [21] containing $6,162$ images and available from the CalPhotos project website [20] (See Fig. 2).

In our tests we use standard Bags-of-Features (BoF) to represent the images [35]. We used the Hessian-Affine extractor and the SIFT [25] descriptor code made available by [22] to produce descriptors. Descriptors were then assigned to clusters to form the BoF representations using the 20k clusters learned from the Flickr60k image set [22].

We tested classification rates with 5, 10, and 50 insect classes selected at random from those having at least four images. Two image descriptors were selected from each class as probe and two as gallery images. We compare the performance of the following classifiers (Table 4.1).

Nearest Neighbor. For each probe, class identity is the label of the L2-nearest gallery BoF.

1-vs-all multi-class SVM. We train one SVM classifier per-class using only gallery images for training: Each classifier is trained using the gallery images of one class as positive examples and the remaining images as negative examples. A class label is selected based on the highest classification score obtained by any of these classifiers. Linear, Gaussian, and χ^2 SVM kernels are reported. The margin parameter

Figure 2. Examples of insect images from the Moorea Biocode collection [20, 21].

("C"), and the kernel parameter were searched over a wide range using cross validation on the training set.

1-vs-all linear SVM with additional negative examples. Following [38], we add to the training of each SVM classifier an additional negative examples set $\mathbf{A}$, which contains the $2,778$ images that have no species label and those 107 images belonging to classes with fewer than four images.

RCA followed by 1-vs-all linear SVM. RCA [34] is trained on $\mathbf{A}$ and applied to the data prior to classification. The reported results are the best obtained over a large range of dimensionality reduction parameter tried out ("r"). Here, and in the next item ("LDA then SVM") below, the grouping to classes was done based on the image label which contains either the biological order, family or species.

LDA followed by 1-vs-all linear SVM. The set A was used to compute the projection directions of multiclass LDA. Then, linear SVM was used as a classifier. Note that variants where LDA is followed by Guassian SVM, Nearest Neighbor, or by assigning to the nearest class center performed far worse in our experiments (same holds for RCA).

1-vs-all SVM with OSS kernel. We use LDA or free-scale LDA as the OSS classifier and the same set $\mathbf{A}$. We then employ either the resulting similarities as the kernel function, or the kernel function which is the exponent of $1/50$ times the OSS score. Hence, we have four kernel functions which are then used as the kernel of a 1-vs-all multi-class SVM.

Table 4.1 shows that SVM classifiers with OSS kernels outperformed other classifiers. This is especially true when using the exponential forms. These tests also imply that although OSS with the LDA classifier is not strictly conditionally positive definite, it can still be used as a kernel for SVM classification. Additional experiments (not shown) demonstrate that performance seems stable for a wide range of exponent values for the OSS, with no change in performance observed for values between 10^{-1} and 10^{-4}.

A note regarding statistical significance. Each experiment in this paper was repeated with the same training and testing split among all training examples. While the variance is sometimes high due to the nature of the datasets, all experiments showing improved results of OSS kernels compared to other method were tested using paired t-tests and shown to be significant at $p < 10^{-5}$.

4.2. Pair-matching using randomized subwindows

The image pair-matching ("same-not-same") problem is defined as follows. Given a training set of image pairs, labeled either "same" (both images portraying the same object) or "not-same" (two images of different objects), the goal is to classify novel image pairs as either similar or not. The Labeled Faces in the Wild (LFW [19]) image set was designed as a benchmark for this test. The images in this set contain faces of people detected in on-line news photos.

The LFW data set provides two pair matching benchmarks. We report results on the protocol called "Image Restricted Training", for which public results are available on the LFW web-site[1]. This benchmark consists of 6,000 pairs, half of which are labeled "same" and half not, partitioned into ten equally sized sets. Each experiment is repeated ten times, using one set for testing and nine others for training. The goal is to predict which of the testing pairs are matching, using only the training data.

We align image pairs using a commercial face alignment software. Each image pair is then represented by randomly selecting 1,000 image coordinates and sampling patches of normally distributed sizes, varying between 10×10 to 20×20 pixels. We then compute the similarities of corresponding patches comparing the performance of straightforward L2-norm and OSS. In the latter case, for each of the 10 repeats, we use one of the nine training splits to produce a set of negative example for each patch. Finally, the image pair is represented by a vector containing 1,000 OSS, one for every patch pair.

Our same-not-same classifier is a binary, linear SVM trained on the eight remaining training splits, each containing image pairs represented as described above. Note that the splits are designed to be mutually exclusive and so subjects used for the negative set $\mathbf{A}$ cannot appear in the sets used for training the SVM and testing.

The pair-matching scores obtained by computing patch distances using L2-norm and OSS are 0.6872 (standard error, SE, of 0.0059) and 0.7637 (SE of 0.0065) respectively. Thus, the OSS considerably outperforms the SSD similarity. ROC curves comparing these and other methods are presented in Fig. 3. A Comparison to published results indicates that the OSS applied to random sub-windows outperforms every single descriptor method in [38], and any other published system, except for the multi-descriptor, multi-similarity measure, "hybrid" method of [38]. Note that the best score reported in [38] for a single descriptor was obtained by using OS similarities between LBP descriptors. This score was 0.7463 (SE of 0.0048), considerably less than what we get here, also using only LBP and OSS. For brevity, we omit from the ROC plots other benchmarks al-

[1] http://vis-www.cs.umass.edu/lfw/results.html

Method	5	10	50
Nearest Neighbor	0.2750 ± 0.1372	0.1725 ± 0.0550	0.0530 ± 0.0258
1-vs-all Linear SVM	0.3300 ± 0.1418	0.2500 ± 0.0918	0.1140 ± 0.0272
1-vs-all Gaussian SVM	0.2800 ± 0.1473	0.1875 ± 0.0510	0.0680 ± 0.0226
1-vs-all χ^2 SVM	0.3600 ± 0.1635	0.2575 ± 0.1017	0.1025 ± 0.0281
1-vs-all SVM with extra neg. examples	0.4100 ± 0.1629	0.2625 ± 0.1398	0.1270 ± 0.0266
RCA followed by 1-vs-all SVM	0.3850 ± 0.1538	0.3000 ± 0.1046	0.1335 ± 0.0283
LDA followed by 1-vs-all SVM	0.3800 ± 0.1795	0.2300 ± 0.1069	0.0945 ± 0.0221
1-vs-all SVM with LDA OSS kernel	0.3900 ± 0.1447	0.2875 ± 0.1134	0.1285 ± 0.0281
1-vs-all SVM with free-scale LDA OSS kernel	0.3250 ± 0.1209	0.2425 ± 0.0963	0.1110 ± 0.0261
1-vs-all SVM with exponential LDA OSS kernel	0.4300 ± 0.1559	0.3075 ± 0.1398	0.1430 ± 0.0301
1-vs-all SVM with exponential free-scale LDA OSS kernel	0.4400 ± 0.1501	0.3200 ± 0.1271	0.1380 ± 0.0302

Table 1. Classification performance and standard errors for the insect identification experiments. Each experiment was repeated 100 times, and the average recognition rate and the standard deviation of the rate are reported. Columns represent the number of insect classes.

gorithms that did not perform well: an analog construct with LDA, using the same set A (with the extra label information), performs worse than the L2-norm (0.6683 ± 0.0053); RCA does not perform better in this task (0.6588 ± 0.0057).

4.3. Multi-person identification using OSS kernels

We next repeated the classification tests from [38] on the LFW data set, this time comparing the performance of their 1-vs-all classifier (with 1,000 extra negative examples), to that of 1-vs-all SVM with an OSS kernel and LDA as the underlying OSS classifier. We use only subjects having enough images to contribute to both "probe" and "gallery" sets. Taking two images per person as probes and two as gallery, we thus employ a subset of the LFW image set consisting of the 610 subjects having at least four images. This subset contains a total of 6733 images. For the negative set $\mathbf{A}$ we take 1,000 images selected at random from individuals having only one image. All our images were aligned using a commercial face alignment software and represented using the LBP descriptor [30, 31].

We compare the performance of the two methods as a function of the number of subjects N, testing 5, 10, 20, and 50 subject identities. We perform 20 repetitions per experiment. In each, we select N random subjects and choose two random gallery images and a disjoint set of two random probes from each. The results reported in Table 4.3 indicate that using OSS as the basis of a kernel matrix outperforms the use of the extra negative examples as part of the negative training in a 1-vs-all multiclass classification scheme, as was done in [38]. Note that RCA is irrelevant to this scenario since the set A contains no groups ("chunklets").

4.4. Visualization of OSS distances

In Fig. 4 we visualize the performance of the OSS as a distance function, applied to face images from the LFW dataset. We compare the OSS measure to the standard Euclidean norm between vectors. We picked, at random, five individuals from the LFW set having at least five images each, and five images from each individual. Dissimilarities between all 300 pairs of LBP encoded images were then computed using both the Euclidean norm and OSS scores. The negative training set $\mathbf{A}$ for the OSS scores consisted of 1,000 images selected at random from individuals having just one image each. The images were then positioned on the plane by computing the 2D Multidimensional-Scaling of these distances (MATLAB's `mdscale` function).

The LFW data set is considered challenging due to its unconstrained nature. Not surprising, no method achieved perfect separation. However, both OSS scores appear to perform better at discriminating between individuals than the L_2 similarity.

5. Conclusions

It is a consensus that collecting unlabeled images in a specific domain is much easier than collecting labeled ones. Therefore, methods that can employ unlabeled data to improve the learning process of labeled examples are valuable. Indeed, much work has been done in this domain. Examples include methods which learn better image representations (e.g., codeword learning schemes) based on unlabeled data, and methods that are able to learn more effectively by employing unlabeled data (semi-supervised learning).

Here we are able to show that the OSS is versatile – it can be used to learn better image representation, as is done in the random sub-windows experiment, it can be used directly as a similarity measure, and it can be used as the basis of a kernel which is employed within SVM.

As shown, when employed with LDA or SVM the OSS is not a PD nor a CPD kernel. However, experiments with the kernel which is the exponent of the LDA OSS score demonstrate that in practice it behaves well. A simple variant of the LDA OSS is shown to be CPD, and performs equally well in the experiments.

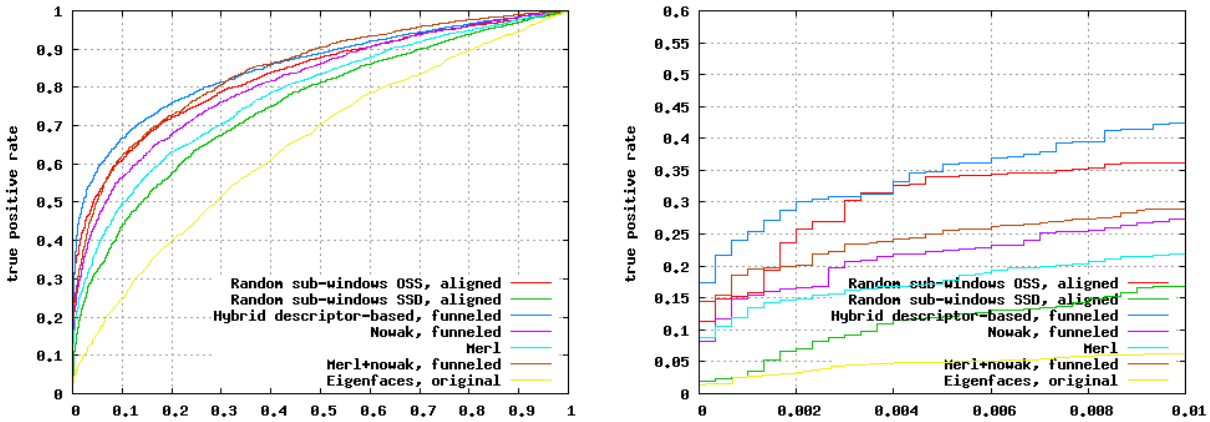
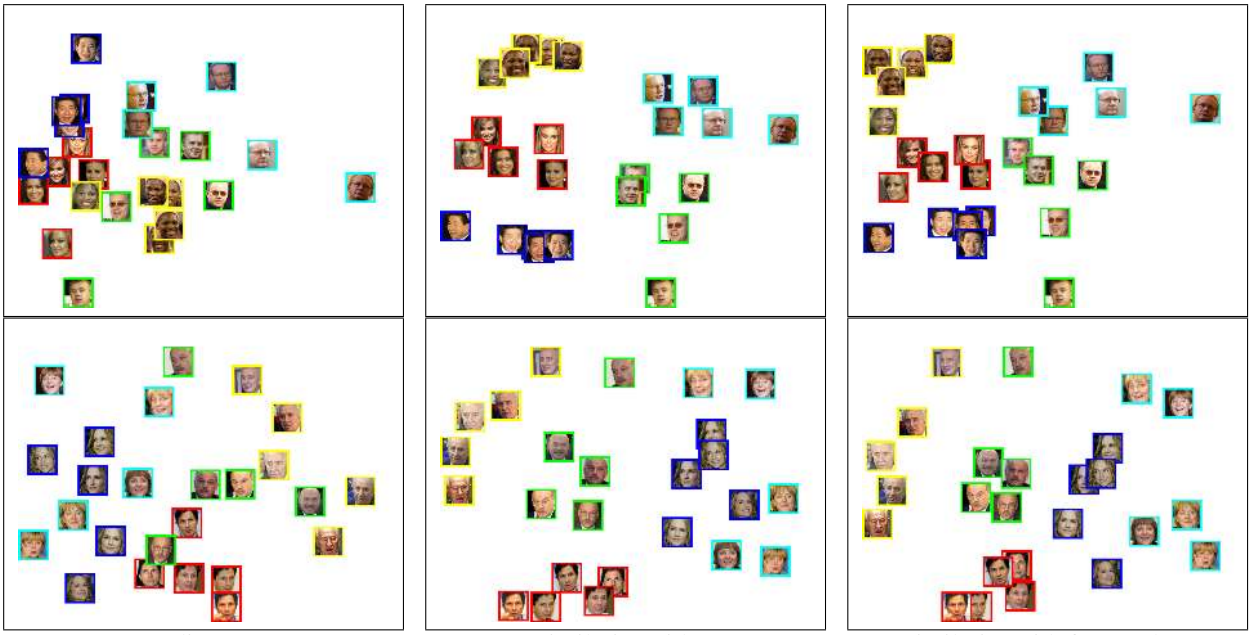


Figure 3. ROC curves averaged over 10 folds of View 2 of the LFW data set. Points on the curve represent averages over the 10 folds of (false positive rate, true positive rate) for a fixed threshold. **Left:** Full ROC curve. **Right:** A zoom-in onto the low false positive region. The methods shown are: classical Eigenfaces [36], combined nowak+Merl system [18], Merl face recognition system [23], the randomized trees approach of [29], Hybrid descriptor based method [38], and our random-patch based representation using both SSD and OSS scores.

Method	5	10	20	50
Nearest Neighbor	0.5750 ± 0.1333	0.4300 ± 0.0979	0.4913 ± 0.0808	0.3430 ± 0.0405
1-vs-all Linear SVM	0.5500 ± 0.1147	0.4875 ± 0.1099	0.5462 ± 0.0808	0.4005 ± 0.0426
1-vs-all Gaussian SVM	0.5950 ± 0.1099	0.5200 ± 0.1174	0.5037 ± 0.0694	0.3410 ± 0.0509
1-vs-all χ^2 SVM	0.6100 ± 0.1119	0.5250 ± 0.0939	0.5737 ± 0.0845	0.4585 ± 0.0522
1-vs-all SVM with extra neg. examples	0.8050 ± 0.1050	0.7175 ± 0.0783	0.5938 ± 0.0980	0.4520 ± 0.0473
LDA followed by 1-vs-all SVM	0.6050 ± 0.1146	0.5750 ± 0.1118	0.6150 ± 0.0916	0.4925 ± 0.0518
1-vs-all SVM with LDA OSS kernel	0.7850 ± 0.1268	0.7300 ± 0.0785	0.7063 ± 0.0802	0.5865 ± 0.0431
1-vs-all SVM with free-scale LDA OSS kernel	0.7550 ± 0.1432	0.7300 ± 0.0768	0.7000 ± 0.0782	0.5855 ± 0.0365
1-vs-all SVM with exp LDA OSS kernel	0.8150 ± 0.1226	0.7225 ± 0.0716	0.6900 ± 0.0758	0.5790 ± 0.0412
1-vs-all SVM with exp FS LDA OSS kernel	0.8250 ± 0.1164	0.7225 ± 0.0716	0.6863 ± 0.0737	0.5800 ± 0.0450

Table 2. Classification performance and SE for the person identification experiments. Columns represent the number of subjects (classes).



L2 distances | OS similarity with LDA | OS similarity with free-scale LDA

Figure 4. Visualizing Euclidean distance vs. OSS scores for LFW images. Images positioned according to pairwise Euclidean distances (left), OSS with LDA scores (middle), and OSS with free-scale LDA scores (right). Color frames encode subject IDs.

Acknowledgments

LW is supported by the Israel Science Foundation (grants No. 1214/06), the Colton Foundation, and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation.

References

[1] E. Bart and S. Ullman. Cross-generalization: learning novel classes from a single example by feature replacement. In *CVPR*, 2005.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, 2001.

[3] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, 2004.

[4] A. Bordes and L. Bottou. The huller: A simple and efficient online svm. In *ECML*, 2005.

[5] S. Boughorbel, J.-P. Tarel, and N. Boujemaa. Conditionally positive definite kernels for svm based image recognition. In *ICME*, 2005.

[6] L. Breiman. Bagging predictors. *Machine Learning*, pages 123–140, 1996.

[7] C.Berg, J. Christensen, and P.Ressel. *Harmonic analysis on semigroups. Theory of positive definite and related functions.* Springer-Velarg, New-York, 1994.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *NIPS*, 2002.

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification, second edition*. Wiley, 2001.

[11] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006.

[12] M. Fink. Object classification from a single example utilizing class relevance pseudo-metrics. In *NIPS*, 2004.

[13] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188, 1936.

[14] P. Geurts, D.Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[15] T. Hastie, , R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2001.

[16] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *ICML*, 2004.

[17] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[18] G. Huang, M. Jones, and E. Learned-Miller. Lfw results using a combined nowak plus merl recognizer. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[19] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. UMASS, TR 07-49, 2007.

[20] C. image collection. Website. `http://calphotos.berkeley.edu/`.

[21] M. B. insect photo collection. Website. `http://bscit.berkeley.edu/biocode/`.

[22] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search. In *ECCV*, 2008.

[23] M. Jones and P.Viola. Face recognition using boosted local features, 2003.

[24] W. Krzanowski, P. Jonathan, W. McCarthy, and M. Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Applied statistics*, 41:101–115, 1995.

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

[26] R. Marée, P. Geurts, and L. Wehenkel. Random subwindows and extremely randomized trees for image classification in cell biology. In *Int. Workshop Multiscale Biological Imaging, Data Mining and Informatics*, pages 611–620, 2006.

[27] R. Marée, P. Geurts, and L. Wehenkel. Content-based image retrieval by indexing random subwindows with randomized trees. In *ACCV*, pages 611–620, 2007.

[28] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müler. Fisher discriminant analysis with kernels. In *IEEE Workshop on Neural Networks for Signal Processing*, 1999.

[29] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[30] T. Ojala, M. Pietikainen, and D. Harwood. A comparative-study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[31] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[32] B. Scholkopf. The kernel trick for distances. In *NIPS*, 2000.

[33] B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2001.

[34] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *ECCV*, 2002.

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

[36] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[37] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *NIPS*, 2006.

[38] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[39] E. Xing, A. Y. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2003.

[40] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.