Journal of Machine Learning Research 11 (2010) 1109-1135 Submitted 2/09; Revised 9/09; Published 3/10

Large Scale Online Learning of Image Similarity Through Ranking

Gal Chechik GAL@GOOGLE.COM
Google
1600 Amphitheatre Parkway
Mountain View CA, 94043

Varun Sharma* VASHARMA@GOOGLE.COM
Google, RMZ Infinity
Old Madras Road, Bengalooru
Karnataka 560016, India

Uri Shalit*† URI.SHALIT@MAIL.HUJI.AC.IL
The Gonda Brain Research Center
Bar Ilan University
52900, Israel

Samy Bengio BENGIO@GOOGLE.COM
Google
1600 Amphitheatre Parkway
Mountain View CA, 94043

Editor: Soeren Sonnenburg, Vojtech Franc, Elad Yom-Tov, Michele Sebag

Abstract

Learning a measure of similarity between pairs of objects is an important generic problem in machine learning. It is particularly useful in large scale applications like searching for an image that is similar to a given image or finding videos that are relevant to a given video. In these tasks, users look for objects that are not only visually similar but also semantically related to a given object. Unfortunately, the approaches that exist today for learning such semantic similarity do not scale to large data sets. This is both because typically their CPU and storage requirements grow quadratically with the sample size, and because many methods impose complex positivity constraints on the space of learned similarity functions.

The current paper presents OASIS, an *Online Algorithm for Scalable Image Similarity* learning that learns a bilinear similarity measure over sparse representations. OASIS is an online dual approach using the passive-aggressive family of learning algorithms with a large margin criterion and an efficient hinge loss cost. Our experiments show that OASIS is both fast and accurate at a wide range of scales: for a data set with thousands of images, it achieves better results than existing state-of-the-art methods, while being an order of magnitude faster. For large, web scale, data sets, OASIS can be trained on more than two million images from 150K text queries within 3 days on a single CPU. On this large scale data set, human evaluations showed that 35% of the ten nearest neighbors of a given test image, as found by OASIS, were semantically relevant to that image. This suggests that query independent similarity could be accurately learned even for large scale data sets that could not be handled before.

Keywords: large scale, metric learning, image similarity, online learning

∗. Varun Sharma and Uri Shalit contributed equally to this work.
†. Also at ICNC, The Hebrew University of Jerusalem, 91904, Israel.

1. Introduction

Large scale learning is sometimes defined as the regime where learning is limited by computational resources rather than by availability of data (Bottou, 2008). Learning a pairwise similarity measure is a particularly challenging large scale task: since pairs of samples have to be considered, the large scale regime is reached even for fairly small data sets, and learning similarity for large data sets becomes exceptionally hard to handle.

At the same time, similarity learning is a well studied problem with multiple real world applications. It is particularly useful for applications that aim to discover new and relevant data for a user. For instance, a user browsing a photo in her album may ask to find similar or related images. Another user may search for additional data while viewing an online video or browsing text documents. In all these applications, similarity could have different flavors: a user may search for images that are similar visually, or semantically, or anywhere in between.

Many similarity learning algorithms assume that the available training data contains real-valued pairwise similarities or distances. However, in all the above examples, the precise numerical value of pairwise similarity between objects is usually not available. Fortunately, one can often obtain information about the *relative* similarity of different pairs (Frome et al., 2007), for instance, by presenting people with several object pairs and asking them to select the pair that is most similar. For large scale data, where man-in-the-loop experiments are prohibitively costly, relative similarities can be extracted from analyzing pairs of images that are returned in response to the same text query (Schultz and Joachims, 2004). For instance, the images that are ranked highly by one of the image search engines for the query "cute kitty" are likely to be semantically more similar than a random pair of images. The current paper focuses on this setting: similarity information is extracted from pairs of images that share a common label or are retrieved in response to a common text query.

Similarity learning has an interesting reciprocal relation with classification. On one hand, pairwise similarity can be used in classification algorithms like nearest neighbors or kernel methods. On the other hand, when objects can be classified into (possibly overlapping) classes, the inferred labels induce a notion of similarity across object pairs. Importantly however, similarity learning assumes a form of supervision that is weaker than in classification, since no labels are provided. OASIS is designed to learn a *class-independent* similarity measure with no need for class labels.

A large number of previous studies have focused on learning a similarity measure that is also a metric, like in the case of a positive semidefinite matrix that defines a Mahalanobis distance (Yang, 2006). However, similarity learning algorithms are often evaluated in a context of ranking. For instance, the learned metric is typically used together with a nearest-neighbor classifier (Weinberger et al., 2006; Globerson and Roweis, 2006). When the amount of training data available is very small, adding positivity constraints for enforcing metric properties is useful for reducing over fitting and improving generalization. However, when sufficient data is available, as in many modern applications, adding positive semi-definitiveness constraints consumes considerable computation time, and its benefit in terms of generalization are limited. With this view, we take here an approach that avoids imposing positivity or symmetry constraints on the learned similarity measure.

The current paper presents an approach for learning semantic similarity that scales up to an order of magnitude larger than current published approaches. Three components are combined to make this approach fast and scalable: First, our approach uses an unconstrained bilinear similarity. Given two images p_1 and p_2 we measure similarity through a bilinear form $p_1^T \mathbf{W} p_2$, where the matrix $\mathbf{W}$ is not required to be positive, or even symmetric. Second we use a sparse representation of the images, which allows to compute similarities very fast. Finally, the training algorithm that we developed, OASIS, *Online Algorithm for Scalable Image Similarity learning*, is an online dual approach based on the passive-aggressive algorithm (Crammer et al., 2006). It minimizes a large margin target function based on the hinge loss, and already converges to high quality similarity measures after being presented with a small fraction of the training pairs.

We find that OASIS is both fast and accurate at a wide range of scales: for a standard benchmark with thousands of images, it achieves better (but comparable) results than existing state-of-the-art methods, with computation times that are shorter by orders of magnitude. For web-scale data sets, OASIS can be trained on more than two million images within three days on a single CPU, and its training time grows linearly with the size of the data. On this large scale data set, human evaluations of OASIS learned similarity show that 35% of the ten nearest neighbors of a given image are semantically relevant to that image.

The paper is organized as follows. We first present our online algorithm, OASIS, based on the Passive-aggressive family of algorithms. We then present the sparse feature extraction technique used in the experiments. We continue by describing experiments with OASIS on problems of image similarity, at two different scales: a large scale academic benchmark with tens of thousands of images, and a web-scale problem with millions of images. The paper ends with a discussion on properties of OASIS.

2. Learning Relative Similarity

We consider the problem of learning a pairwise similarity function S, given data on the relative similarity of pairs of images.

Formally, let $\mathcal{P}$ be a set of images, and $r_{ij} = r(p_i, p_j) \in \mathbb{R}$ be a pairwise relevance measure which states how strongly $p_j \in \mathcal{P}$ is related to $p_i \in \mathcal{P}$. This relevance measure could encode the fact that two images belong to the same category or were appropriate for the same query. We do not assume that we have full access to all the values of r. Instead, we assume that we can compare some pairwise relevance scores (for instance $r(p_i, p_j)$ and $r(p_i, p_k)$) and decide which pair is more relevant. We also assume that when $r(p_i, p_j)$ is not available, its value is zero (since the vast majority of images are not related to each other). Our goal is to learn a similarity function $S(p_i, p_j)$ that assigns higher similarity scores to pairs of more relevant images,

$$S(p_i, p_i^+) > S(p_i, p_i^-)\ , \quad \forall p_i, p_i^+, p_i^- \in \mathcal{P} \text{ such that } r(p_i, p_i^+) > r(p_i, p_i^-). \tag{1}$$

In this paper we overload notation by using p_i to denote both the image and its representation as a column vector $p_i \in \mathbb{R}^d$. We consider a parametric similarity function that has a bi-linear form,

$$S_{\mathbf{W}}(p_i, p_j) \equiv p_i^T\ \mathbf{W}\ p_j \tag{2}$$

with $\mathbf{W} \in \mathbb{R}^{d \times d}$. Importantly, if the images p_i are represented as sparse vectors, namely, only a number $k_i \ll d$ of the d entries in the vector p_i are non-zeroes, then the value of Equation (2) can be computed very efficiently even when d is large. Specifically, $S_{\mathbf{W}}$ can be computed with complexity of $O(k_i k_j)$ regardless of the dimensionality d.

2.1 An Online Algorithm

We propose an online algorithm based on the Passive-Aggressive (PA) family of learning algorithms introduced by Crammer et al. (2006). Here we consider an algorithm that uses triplets of images $p_i, p_i^+, p_i^- \in \mathcal{P}$ such that $r(p_i, p_i^+) > r(p_i, p_i^-)$.

We aim to find a parametric similarity function S such that all triplets obey

$$S_{\mathbf{W}}(p_i, p_i^+) > S_{\mathbf{W}}(p_i, p_i^-) + 1 \tag{3}$$

which means that it fulfills Equation (1) with a safety margin of 1. We define the following hinge loss function for the triplet:

$$l_{\mathbf{W}}(p_i, p_i^+, p_i^-) = \max\left\{0, 1 - S_{\mathbf{W}}(p_i, p_i^+) + S_{\mathbf{W}}(p_i, p_i^-)\right\}. \tag{4}$$

Our goal is to minimize a global loss $L_{\mathbf{W}}$ that accumulates hinge losses (4) over all possible triplets in the training set:

$$L_{\mathbf{W}} = \sum_{(p_i, p_i^+, p_i^-) \in \mathcal{P}} l_{\mathbf{W}}(p_i, p_i^+, p_i^-) .$$

In order to minimize this loss, we apply the Passive-Aggressive algorithm iteratively over triplets to optimize $\mathbf{W}$. First, $\mathbf{W}$ is initialized to some value $\mathbf{W}^0$. Then, at each training iteration i, we randomly select a triplet (p_i, p_i^+, p_i^-), and solve the following convex problem with soft margin:

$$\begin{aligned} \mathbf{W}^i \quad &= \quad \operatorname*{argmin}_{\mathbf{W}} \frac{1}{2}\|\mathbf{W} - \mathbf{W}^{i-1}\|^2_{Fro} + C\xi \\ \text{s.t.} \qquad & l_{\mathbf{W}}(p_i, p_i^+, p_i^-) \leq \xi \quad \text{and} \quad \xi \geq 0 \end{aligned} \tag{5}$$

where $\|\cdot\|_{Fro}$ is the Frobenius norm (point-wise L_2 norm). Therefore, at each iteration i, $\mathbf{W}^i$ is selected to optimize a trade-off between remaining close to the previous parameters $\mathbf{W}^{i-1}$ and minimizing the loss on the current triplet $l_{\mathbf{W}}(p_i, p_i^+, p_i^-)$. The *aggressiveness* parameter C controls this trade-off.

OASIS

<u>Initialization:</u>

Initialize $\mathbf{W}^0 = I$

<u>Iterations</u>

repeat

Sample three images p, p_i^+, p_i^-, such that $r(p_i, p_i^+) > r(p_i, p_i^-)$.

Update $\mathbf{W}^i = \mathbf{W}^{i-1} + \tau_i \mathbf{V}^i$

where $\tau_i = \min\left\{C, \frac{l_{\mathbf{W}^{i-1}}(p_i, p_i^+, p_i^-)}{\|\mathbf{V}^i\|^2}\right\}$

and $\mathbf{V}^i = [p_i^1(p_k^+ - p_k^-), \ldots, p_i^d(p_k^+ - p_k^-)]^T$

until (stopping criterion)

Figure 1: Pseudo-code of the OASIS algorithm.

We follow Crammer et al. (2006) to solve the problem in Equation (5). When $l_{\mathbf{W}}(p_i, p_i^+, p_i^-) = 0$, it is clear that $\mathbf{W}^i = \mathbf{W}^{i-1}$ satisfies Equation (5) directly. Otherwise, we define the Lagrangian

$$\mathcal{L}(\mathbf{W}, \tau, \xi, \lambda) = \frac{1}{2}\|\mathbf{W} - \mathbf{W}^{i-1}\|^2 + C\xi + \tau(1 - \xi - p_i^T \mathbf{W}(p_i^+ - p_i^-)) - \lambda\xi \tag{6}$$

where $\tau \geq 0$ and $\lambda \geq 0$ are Lagrange multipliers. The optimal solution is such that the gradient vanishes $\frac{\partial \mathcal{L}(\mathbf{W},\tau,\xi,\lambda)}{\partial \mathbf{W}} = 0$, hence

$$\frac{\partial \mathcal{L}(\mathbf{W}, \tau, \xi, \lambda)}{\partial \mathbf{W}} = \mathbf{W} - \mathbf{W}^{i-1} - \tau \mathbf{V}_i = 0$$

where the gradient matrix $\mathbf{V}_i = \frac{\partial \mathcal{L}_W}{\partial \mathbf{W}} = [p_i^1(p_i^+ - p_i^-), \ldots, p_i^d(p_i^+ - p_i^-)]^T$. The optimal new $\mathbf{W}$ is therefore

$$\mathbf{W} = \mathbf{W}^{i-1} + \tau \mathbf{V}_i \tag{7}$$

where we still need to estimate τ. Differentiating the Lagrangian with respect to ξ and setting it to zero also yields:

$$\frac{\partial \mathcal{L}(\mathbf{W}, \tau, \xi, \lambda)}{\partial \xi} = C - \tau - \lambda = 0 \tag{8}$$

which, knowing that $\lambda \geq 0$, means that $\tau \leq C$. Plugging Equations (7) and (8) back into the Lagrangian in Equation (6), we obtain

$$\mathcal{L}(\tau) = \frac{1}{2}\tau^2\|\mathbf{V}_i\|^2 + \tau(1 - p_i^T(\mathbf{W}^{i-1} + \tau\mathbf{V}_i)(p_i^+ - p_i^-)) \ .$$

Regrouping the terms we obtain

$$\mathcal{L}(\tau) = -\frac{1}{2}\tau^2\|\mathbf{V}_i\|^2 + \tau(1 - p_i^T \mathbf{W}^{i-1}(p_i^+ - p_i^-)) \ .$$

Taking the derivative of this second Lagrangian with respect to τ and setting it to 0, we have

$$\frac{\partial \mathcal{L}(\tau)}{\partial \tau} = -\tau\|\mathbf{V}_i\|^2 + (1 - p_i^T \mathbf{W}^{i-1}(p_i^+ - p_i^-)) = 0$$

which yields

$$\tau = \frac{1 - p_i^T \mathbf{W}^{i-1}(p_i^+ - p_i^-)}{\|\mathbf{V}_i\|^2} = \frac{l_{\mathbf{W}^{i-1}}(p_i, p_i^+, p_i^-)}{\|\mathbf{V}_i\|^2} \ .$$

Finally, Since $\tau \leq C$, we obtain

$$\tau = \min\left\{C, \frac{l_{\mathbf{W}^{i-1}}(p_i, p_i^+, p_i^-)}{\|\mathbf{V}_i\|^2}\right\} \ . \tag{9}$$

Equations (7) and (9) summarize the update needed for every triplets (p_i, p_i^+, p_i^-). It has been shown (Crammer et al., 2006) that applying such an iterative algorithm yields a cumulative online loss that is likely to be small. It was furthermore shown that selecting the best $\mathbf{W}_i$ during training using a hold-out validation set achieves good generalization. We also show below that multiple runs of the algorithm converge to provide similar precision (see Figure 7).

2.2 Loss Bounds

Following closely the analysis of loss bounds for passive aggressive (PA) algorithms developed by Crammer et al. (2006) we state similar *relative* bounds for the OASIS framework. We do this by rewriting OASIS as a straightforward linear classification problem. Denote by $\overrightarrow{w_i}$ the vector obtained by"unfolding" the matrix $\mathbf{W}$ (concatenating all its columns into a single vector) and similarly $\overrightarrow{x_i}$ the unfolded matrix $p_i(p_i^+ - p_i^-)^T$. Using this notation, the constraint in Equation (3) becomes

$$\overrightarrow{w_i} \cdot \overrightarrow{x_i} > 1 \quad ,$$

with $\cdot$ denoting the standard inner product. This is equivalent to the formulation of PA when the label y_i is always 1. The introduction of slack variables in Equation (5) brings us to the variant denoted by Crammer et al. (2006) as PA-I.

The loss bounds in Crammer et al. (2006) rely on $\overrightarrow{w_0}$ being the zero vector. Since here we initialize with $W^0 = I$ (the identity matrix) we need to adapt the analysis slightly. Let $\overrightarrow{u}$ be a vector in $\mathbb{R}^{d^2}$ obtained by unfolding an arbitrary matrix $\mathbf{U}$. We define

$$l_i = 1 - \overrightarrow{w_i} \cdot \overrightarrow{x_i} \quad \text{and} \quad l_i^* = 1 - \overrightarrow{u} \cdot \overrightarrow{x_i} \quad ,$$

where l_i is the instantaneous loss at round i, and l_i^* is the loss suffered by the arbitrary vector $\overrightarrow{u}$. The following two theorems rely on Lemma 1 of Crammer et al. (2006), which we restate without proof:

$$\sum \tau_i (2l_i - \tau_i \|x_i\|^2 - 2l_i^*) \leq \|\overrightarrow{u} - \overrightarrow{w_0}\|^2 \, .$$

While in Crammer et al. (2006) $\overrightarrow{w_0}$ is the zero vector, in our case $\overrightarrow{w_0}$ is the unfolded *identity matrix*. We therefore have

$$\|\overrightarrow{u} - \overrightarrow{w_0}\|^2 = \|\mathbf{U}\|_{Fro}^2 - 2trace(\mathbf{U}) + n \, .$$

Using this modified lemma we can restate the relevant bound:

Theorem 1 *Let $(\overrightarrow{x_1}),...,(\overrightarrow{x_M})$ be a sequence of examples where $\overrightarrow{x_i} \in \mathbb{R}^{d^2}$, $\|\overrightarrow{x_i}\| \leq R$ for all $i = 1...M$. Then, for any matrix $\mathbf{U} \in \mathbb{R}^{n2}$, the number of prediction mistakes made by OASIS on this sequence of examples is bounded from above by,*

$$\max\{R^2, 1/C\} \left(\|\mathbf{U}\|_{Fro}^2 - 2trace(\mathbf{U}) + n + 2C \sum_{i=1}^{M} l_i^* \right)$$

where C is the aggressiveness parameter provided to OASIS.

2.3 Sampling Strategy

For real world data sets, the actual number of triplets (p_i, p_i^+, p_i^-) is typically very large and cannot be stored in memory. Instead, we use the fact that the number of relevant images for a category or a query is typically small, and keep a list of relevant images for each query or category. For the case of single-labeled images, we can efficiently retrieve an image that is relevant to a given image, by first finding its class, and then finding another image from that class. The case of multi-labeled images is described in Section 5.2.

Specifically, to sample a triplet (p_i, p_i^+, p_i^-) during training, we first uniformly sample an image p_i from $\mathcal{P}$. Then we uniformly sample an image p_i^+ from the images sharing the same categories or queries as p_i. Finally, we uniformly sample an image p_i^- from the images that share no category or query with p_i. When the set $\mathcal{P}$ is very large and the number of categories or queries is also very large, one does not need to maintain the set of non-relevant images for each image: sampling directly from $\mathcal{P}$ instead only adds a small amount of noise to the training procedure and is not really harmful.

When relevance feedbacks $r(p_i, p_j)$ are provided as real numbers and not just $\in \{0,1\}$, one could use these number to bias training towards those pairs that have a higher relevance feedback value. This can be done by considering $r(p_i, p_j)$ as frequencies of appearance, and sampling pairs according to the distribution of these frequencies.

3. Image Representation

The problem of selecting an informative representation of images is still an unsolved computer vision challenge, and an ongoing research topic. Different approaches for image representation have been proposed including by Feng et al. (2004); Takala et al. (2005) and Tieu and Viola (2004). In the information retrieval community there is wide agreement that a bag-of-words representation is a very useful representation for handling text documents in a wide range of applications. For image representation, there is still no such approach that would be adequate for a wide variety of image processing problems. However, among the proposed representations, a consensus is emerging on using *local descriptors* for various tasks, for example, Lowe (2004); Quelhas et al. (2005). This type of representation segments the image into *regions of interest*, and extracts visual features from each region. The segmentation algorithm as well as the region features vary among approaches, but, in all cases, the image is then represented as a set of feature vectors describing the regions of interest. Such a set is often called a *bag-of-local-descriptors*.

In this paper we take the approach of creating a sparse representation based on the framework of local descriptors. Our features are extracted by dividing each image into overlapping square blocks, and each block is then described with edge and color histograms. For edge histograms, we rely on *uniform Local Binary Patterns* (uLBPs) proposed by Ojala et al. (2002). These texture descriptors have shown to be effective on various tasks in the computer vision literature (Ojala et al., 2002; Takala et al., 2005), certainly due to their robustness with respect to changes in illumination and other photometric transformations (Ojala et al., 2002). Local Binary Patterns estimate a texture histogram of a block by considering differences in intensity at circular neighborhoods centered on each pixel. Precisely, we use $LBP_{8,2}$ patterns, which means that a circle of radius 2 is considered centered on each block. For each circle, the intensity of the center pixel is compared to the interpolated intensities located at 8 equally-spaced locations on the circle, as shown on Figure 2, left. These eight binary tests (lower or greater intensity) result in an 8-bit sequence, see Figure 2, right. Hence, each block pixel is mapped to a sequence among $2^8 = 256$ possible sequences and each block can therefore be represented as a 256-bin histogram. In fact, it has been observed that the bins corresponding to non-uniform sequences (sequences with more than 2 transitions $1 \rightarrow 0$ or $0 \rightarrow 1$) can be merged, yielding more compact 59-bin histograms without performance loss (Ojala et al., 2002).

Color histograms are obtained by K-means clustering. We first select a palette or typical colors by training a color codebook from the Red-Green-Blue pixels of a large training set of images using K-means. The color histogram of a block is then obtained by mapping each block pixel to the closest color in the codebook palette.

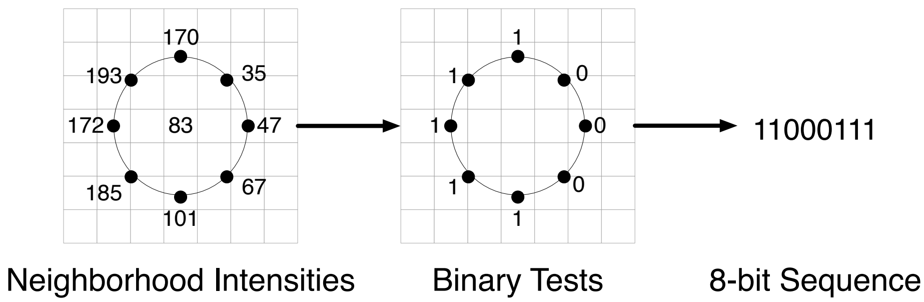

Figure 2: An example of Local Binary Pattern ($LBP_{8,2}$). For a given pixel, the Local Binary Pattern is an 8-bit code obtained by verifying whether the intensity of the pixel is greater or lower than its 8 neighbors.

Finally, the histograms describing color and edge statistics of each block are concatenated, which yields a single vector descriptor per block. Our local descriptor representation is therefore simple, relying on both a basic segmentation approach and simple features. Naturally, alternative representations could also be used with OASIS, (Feng et al., 2004; Grangier et al., 2006; Tieu and Viola, 2004) However, this paper focuses on the learning model, and a benchmark of image representations is beyond the scope of the current paper.

As a final step, we use the representation of blocks to obtain a representation for an image. For computation efficiency we aim at a high dimensional and sparse vector space. For this purpose, each local descriptor of an image p is represented as a discrete index, called *visual term* or *visterm*, and, like for text data, the image is represented as a *bag-of-visterms* vector, in which each component p_i is related to the presence or absence of visterm i in p.

The mapping of the descriptors to discrete indexes is performed according to a codebook C, which is typically learned from the local descriptors of the training images through k-means clustering (Duygulu et al., 2002; Jeon and Manmatha, 2004; Quelhas et al., 2005). The assignment of the weight p_i of visterm i in image p is as follows:

$$p_i = \frac{f_i\, d_i}{\sqrt{\sum_{j=1}^{d} (f_j\, d_j)^2}} \,,$$

where f_i is the term frequency of i in p, which refers to the number of occurrences of i in p, while d_j is the inverse document frequency of j, which is defined as $-log(r_j)$, r_j being the fraction of training images containing at least one occurrence of visterm j. This approach has been found successful for the task of content based image ranking described by Grangier and Bengio (2008).

In the experiments described below, we used a large set of images collected from the web to train the features. This set is described in more detail in Section 5.2. We used a set of 20 typical RGB colors (hence the number of clusters used in the k-means for colors was 20), the block vocabulary size $d = 10000$ and our image blocks were of size 64x64 pixels, overlapping every 32 pixels. Furthermore, in order to be robust to scale, we extracted blocks at various scales by successively down scaling images by a factor of 1.25 and extracting the features at each level, until there were less than 10 blocks in the resulting image. There was on average around 70 non-zero values (out of 10000) describing a single image. Note that no other information (such as meta-data) was added in the input vector representation each image.

4. Related Work

Similarity learning can be considered in two main setups, depending on the type of available training labels. First, a regression setup, where the training set consists of pairs of objects x_i^1, x_i^2 and their pairwise similarity $y_i \in \mathbf{R}$. In many cases however, precise similarities are not available, but rather a weaker notion of similarity order. In one such setup, the training set consists of triplets of objects x_i^1, x_i^2, x_i^3 and a ranking similarity function, that can tell which of the two pairs (x^1, x^2) or (x^1, x^3) is more similar. Finally, multiple similarity learning studies assume that a binary measure of similarity is available $y_i \in \{+1, -1\}$, indicating whether a pair of objects is similar or not.

For small-scale data, there are two main groups of similarity learning approaches. The first approach, learning Mahalanobis distances, can be viewed as learning a linear projection of the data into another space (often of lower dimensionality), where a Euclidean distance is defined among pairs of objects. Such approaches include Fisher's Linear Discriminant Analysis (LDA), relevant component analysis (RCA) (Bar-Hillel et al., 2003), supervised global metric learning (Xing et al., 2003), large margin nearest neighbor (LMNN) (Weinberger et al., 2006) and Metric Learning by Collapsing Classes (Globerson and Roweis, 2006). A Mahalanobis distance learning algorithm which uses a supervision signal identical to the one we employ in OASIS is Rosales and Fung (2006), which learns a special kind of PSD matrix via linear programming. See also a review by Yang (2006) for more details.

The second family of approaches, learning kernels, is used to improve performance of kernel based classifiers. Learning a full kernel matrix in a non parametric way is prohibitive except for very small data. As an alternative, several studies suggested to learn a weighted sum of pre-defined kernels (Lanckriet et al., 2004) where the weights are being learned from data. In some applications this was shown to be inferior to uniform weighting of the kernels (Noble, 2008). The work of Frome et al. (2007) further learns a weighting over local distance function for every image in the training set. Non linear image similarity learning was also studied in the context of dimensionality reduction, as in Hadsell et al. (2006).

Finally, Jain et al. (2008a,b), based on work by Davis et al. (2007), aim to learn metrics in an online setting. This work is one of the closest work with respect to OASIS: it learns a linear model of a [dis-]similarity function between documents in an online way. The main difference is that the work of Jain et al. (2008a) learn a true distance throughout the learning process, imposing positive definiteness constraints, and is slightly less efficient computationally. We argue in this paper that in the large scale regime, such a constraint is not necessary given the amount of available training examples.

Another work closely related to OASIS is that of Rasiwasia and Vasconcelos (2008), which also tries to learn a semantic similarity function between images. In their case, however, semantic similarity is learned by representing each image by the posterior probability distribution over a predefined set of semantic tags, and then computing the distance between two images as the distance between the two underlying posterior distributions. The representation size of images in this approach is therefore equal to the number of semantic classes, hence it will not scale when the number of semantic classes is very large as in free text search.

5. Experiments

Evaluating large scale learning algorithms poses special challenges. First, current available benchmarks are limited either in their scale, like 30K images in Caltech256 as described by Griffin et al. (2007), or in their resolution, such as the tiny images data set of Torralba et al. (2007). Large scale methods are not expected to perform particularly well on small data sets, since they are designed to extract limited information from each sample. Second, many images on the web cannot be used without explicit permission, hence they cannot be collected and packed into a single database. Large, proprietary collections of images do exist, but are not available freely for academic research. Finally, except for very few cases, similarity learning approaches in current literature do not scale to handle large data sets effectively, which makes it hard to compare a new large scale method with the existing methods.

To address these issues, this paper takes the approach of conducting experiments at two different scales. First, to demonstrate the scalability of OASIS we applied OASIS to a web-scale data with 2.7 million images. Second, to investigate the properties of OASIS more deeply, we compare OASIS with small-scale methods using the standard Caltech256 benchmark.

5.1 Evaluation Measures

We evaluated the performance of all algorithms using standard ranking precision measures based on nearest neighbors. For each query image in the test set, all other test images were ranked according to their similarity to the query image. The number of same-class images among the top k images (the k nearest neighbors) was computed. When averaged across test images (either within or across classes), this yields a measure known as precision-at-top-k, providing a precision curve as a function of the rank k.

We also calculated the *mean average precision* (mAP), a measure that is widely used in the information retrieval community. To compute average precision, the precision-at-top-k is first calculated for each test image. Then, it is averaged over all positions k that have a positive sample. For example, if all positives are ranked highest, the average-precision is 1. The average-precision measure is then further averaged across all test image queries, yielding the *mean average precision* (mAP).

5.2 Web-Scale Experiment

Our first set of experiments is based on Google proprietary data that is two orders of magnitude larger than current standard benchmarks. We collected a set of $\sim$150K text queries submitted to the Google Image Search system. For each of these queries, we had access to a set of relevant images, each of which is associated with a numerical relevance score. This yielded a total of $\sim$2.7 million images, which we split into a training set of 2.3 million images and a test set of 0.4 million images (see Table 1).

Set	Number of Queries	Number of Images
Training	139944	2292259
Test	41877	402164

Table 1: Statistics of the Web data set.

5.2.1 EXPERIMENTAL SETUP

We used the query-image relevance information to create an image-image relevance as follows. Denote the set of text queries by $\mathcal{Q}$ and the set of images by $\mathcal{P}$. For each $q \in \mathcal{Q}$, let $\mathcal{P}_q^+$ denote the set of images that are relevant to the query q, and let $\mathcal{P}_q^-$ denote the set of irrelevant images. The query-image relevance is defined by the matrix $\mathbf{R}_{QI} : \mathcal{Q} \times \mathcal{P} \rightarrow \mathbb{R}^+$, and obeys $\mathbf{R}_{QI}(q, p_q^+) > 0$ and $\mathbf{R}_{QI}(q, p_q^-) = 0$ for all $q \in \mathcal{Q}$, $p_q^+ \in \mathcal{P}_q^+$, $p_q^- \in \mathcal{P}_q^-$. We also computed a normalized version of $\mathbf{R}_{QI}$, which can be interpreted as a joint distribution matrix, or the probability to observe a query q and an image p for that query,

$$Pr(q, p) = \frac{\mathbf{R}_{QI}(q, p)}{\sum_{q', p'} \mathbf{R}_{QI}(q', p')} \quad .$$

In order to compute the image-image relevance matrix $\mathbf{R}_{II} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^+$, we treated images as being conditionally independent given the queries, $Pr(p_1, p_2|q) = Pr(p_1|q)Pr(p_2|q)$, and computed the joint image-image probability as a relevance measure

$$Pr(p_1, p_2) = \sum_{q \in \mathcal{Q}} Pr(p_1, p_2|q) Pr(q) = \sum_{q \in \mathcal{Q}} Pr(p_1 \mid q) Pr(p_2 \mid q) Pr(q) \; .$$

To improve scalability, we used a threshold over this joint distribution, and considered two images to be related only if their joint distribution exceeded a cutoff value θ

$$\mathbf{R}_{II}(p_1, p_2) = [Pr(p_1, p_2)]_\theta \tag{10}$$

where $[x]_\theta = x$ for $x > \theta$ and is zero otherwise. To set the value of θ we have manually inspected a small subset of pairs of related images taken from the training set. We selected the largest θ such that most of those related pairs had scores above the threshold, while minimizing noise in $\mathbf{R}_{II}$.

Equation 10 is written as if one needs to calculate the full joint matrix $\mathbf{R}_{II}$, but this matrix grows quadratically with the number of images. In practice, we can use the fact that $\mathbf{R}_{QI}$ is very sparse, to quickly create a list with images that are relevant to a given image. To do this given an image p_i, we go over all the queries for which it is relevant $\mathbf{R}_{QI}(q, p_i)$, and for each of these queries, collect the list of all images that are relevant to that query. The average number of queries relevant for an image in our data is small (about 100), and so is the number of images relevant for a given query. As a result, $\mathbf{R}_{II}$ can be calculated efficiently even for large image sets.

We trained OASIS over 2.3 million images in the training set using the sampling mechanism based on the relevance of each image, as described in Section 2.3. To select the number of training iterations, we used as a validation set a small subset of the training set to trace the mean average precision of the model at regular intervals during the training process. Training was stopped when the mean average precision had saturated, which happened after 160 million iterations (triplets). Overall, training took a total of $\sim$4000 minutes on a single CPU of a standard modern machine. Finally, we evaluated the trained model on the 400 thousand images of the test set.

5.2.2 RESULTS

We start with specific examples illustrating the behavior of OASIS, and continue with a quantitative analysis of precision and speed. Table 2 shows the top five images as ranked by OASIS on four examples of query-images in the test set. The relevant text queries for each image are shown beneath the image. The first example (top row), shows a query-image that was originally retrieved

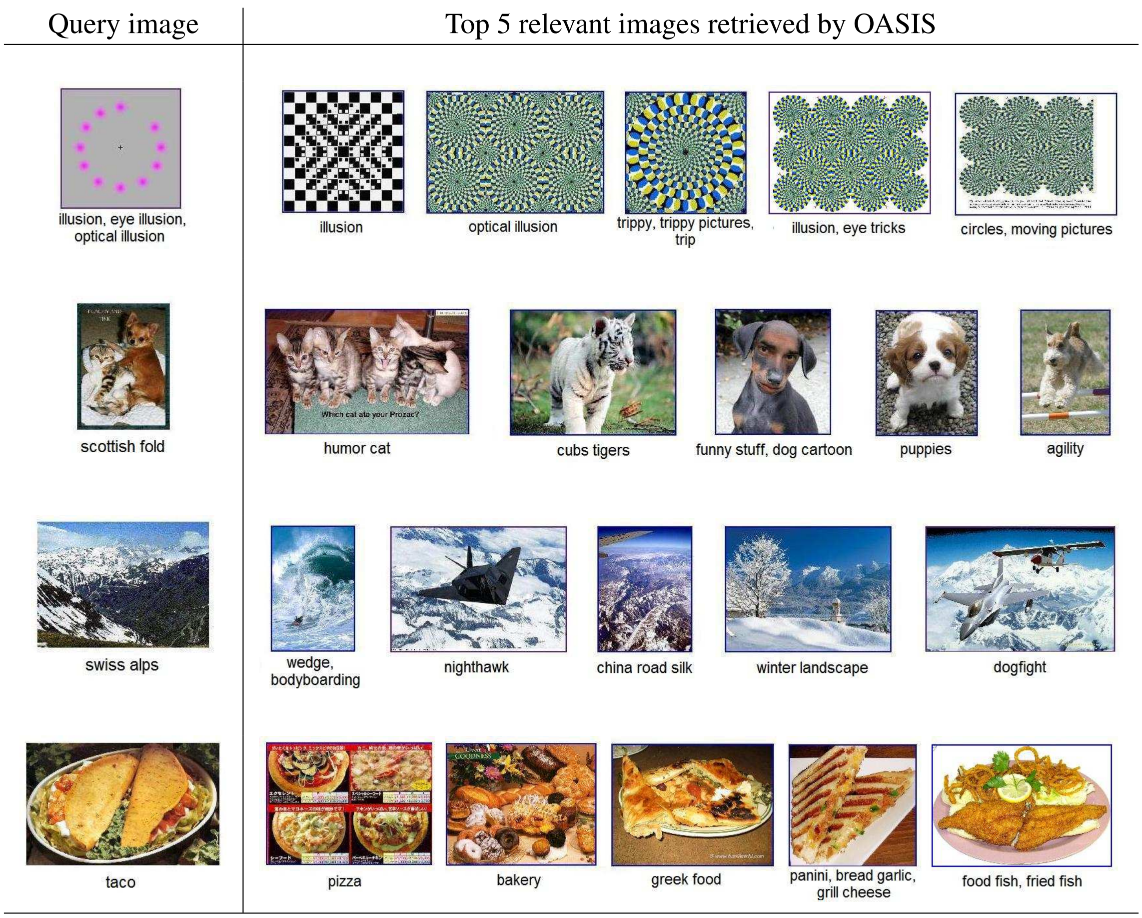

Table 2: OASIS: Successful cases from the Web data set

in response to the text query “illusion”. All five images ranked highly by OASIS are semantically related, showing other types of visual illusions. Similar results can be observed for the three remaining examples on this table, where OASIS captures well the semantics of animal photos (cats and dogs), mountains and different food items.

In all these cases, OASIS captures similarity that is both semantic and visual, since the raw visual similarity of these images is not high. A different behavior is demonstrated in Table 3. It shows three cases where OASIS was biased by visual similarity and provided high rankings to images that were semantically non relevant. In the first example, the assortment of flowers is confused with assortments of food items and a thigh section (5th nearest neighbor) which has visually similar shape. The second example presents a query image which in itself has no definite semantic element. The results retrieved are those that merely match texture of the query image and bear no semantic similarity. In the third example, OASIS fails to capture the butterfly in the query image.

To obtain a quantitative evaluation of OASIS we computed the precision at top k, using a threshold $\theta = 0$, which means that an image in the test set is considered relevant to a query image, if there exists at least one text query to which they were both relevant to.

The obtained precision values were quite low, achieving 1.5% precision at the top ranked image. This is drastically lower than the precision described below for Caltech256, and could be the result

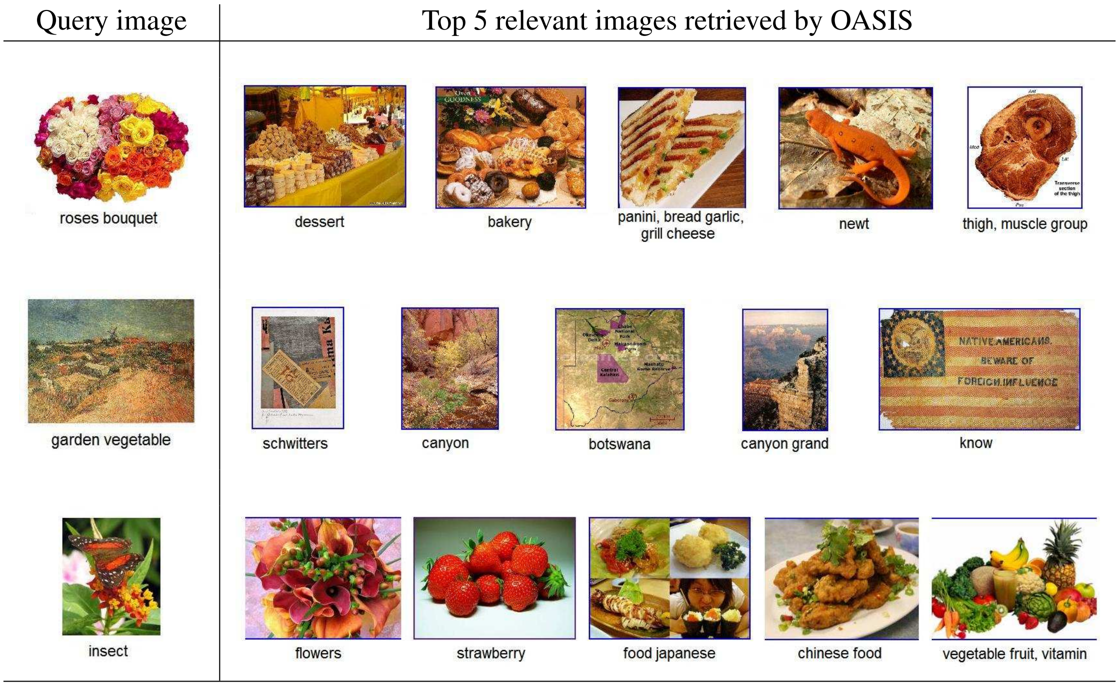

Table 3: OASIS: Failure cases from the Web data set

of multiple reasons. First, the number of unique textual queries in our data is very large (around 150K), hence the images in this data set were significantly more heterogeneous than images in the Caltech256 data.

Second, and most importantly, our labels that measure pairwise relevance are very partial. This means that many pairs of images that are semantically related are not labeled as such. A clear demonstration of this effect is observed in Tables 2 and 3. The query images (like "*scottish fold*") have labels that are usually very different from the labels of the retrieved images (as in "*humor cat*", "*agility*") even if their semantic content is very similar. This is a common problem in content-based analysis, since similar content can be described in many different ways. In the case discussed here, the partial data on the query-image relevance $\mathbf{R}_{QI}$ is further propagated to the image-image relevance measure $\mathbf{R}_{II}$.

5.2.3 HUMAN EVALUATION EXPERIMENTS

In order to obtain a more accurate estimate of the real semantic precision, we performed a rating experiment with human evaluators. We chose the 25 most relevant images[1] from the test set and retrieved their 10 nearest neighbors as determined by OASIS. We excluded query-images which contained porn, racy or duplicates in their 10 nearest neighbors. We also selected randomly a set of 10 negative images p^- that were chosen for each of the query images p such that $\mathbf{R}_{II}(p, p^-) = 0$. These negatives were then randomly mixed with the 10 nearest neighbors.

All 25 query images were presented to twenty human evaluators, asking them to mark which of the 20 candidate images are *semantically relevant* to the query image.[2] Evaluators were volunteers

1. The overall relevance of an image was estimated as the sum of relevances of the image with respect to all queries.
2. The description of the task as given to the evaluators is provided in Appendix A.

selected from a pool of friends and colleagues, many of which had experience with search or machine vision problems. We collected the ratings on the positive images and calculated the precision at top k.

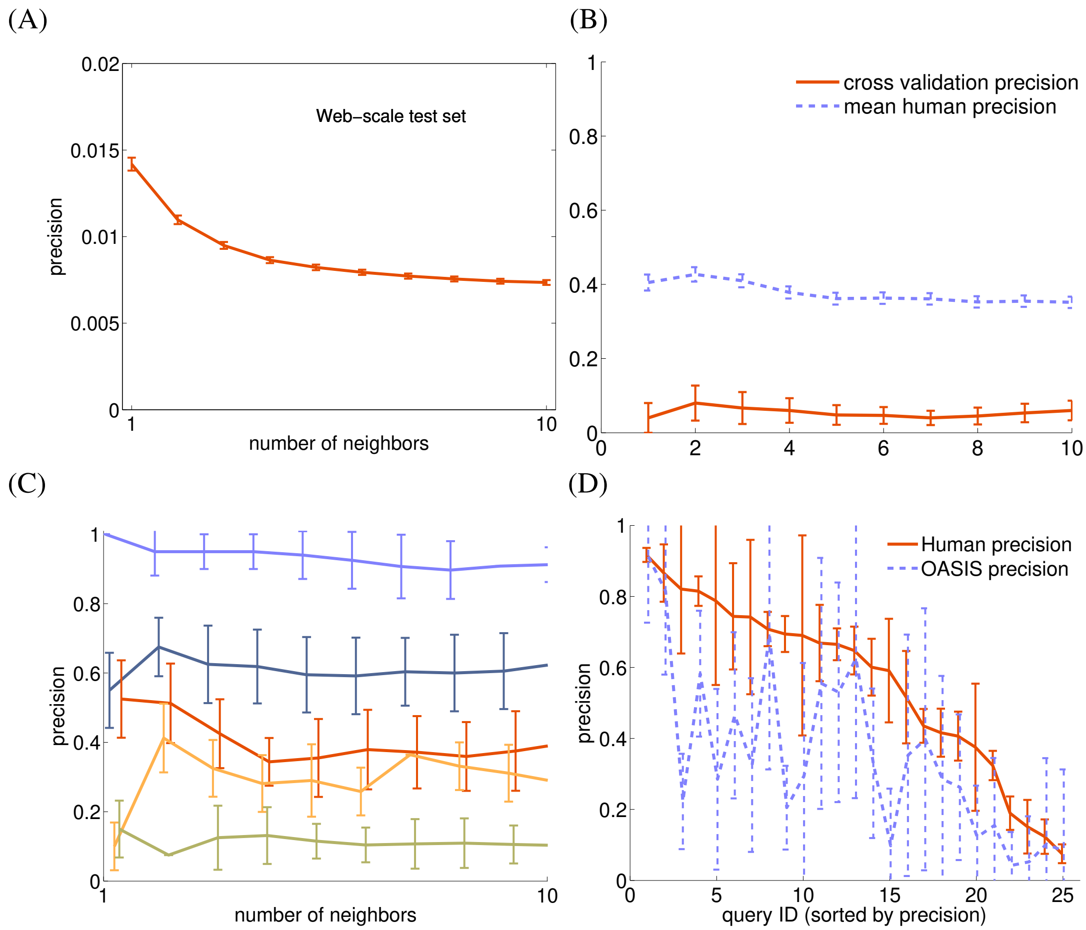

Figure 3: **(A)** Precision at top k as a function of k neighbors computed against $\mathbf{R}_{II}$ ($\theta = 0$) for the web-scale test set. **(B)** Precision at top k as a function of k neighbors for the human evaluation subset. **(C)** Mean precision for 5 selected queries. Error bars denote the standard error of the mean. To select the queries for this plot, we first calculated the mean-average precision per query, sorted the queries by their mAP, and selected the queries ranked at position 1, 6, 11, 16, and 21. **(D)** Precision of OASIS and human evaluators, per query, using rankings of all (remaining) human evaluators as a ground truth.

Figure 3(B) shows the average precision across all queries and evaluators. Precision peaks at 42% and reaches 35% at the top 10 ranked image, being significantly higher than the values calculated automatically using $\mathbf{R}_{II}$.

We observed that the variability across different query images was also very high. Figure 3(C) shows the precision for 5 different queries, selected to span the range of average-precision values. The error bars at each curve show the variability in the responses of different evaluators. The precision of OASIS varies greatly across different queries. Some query images were "easy" for OASIS, yielding high scores from most evaluators, while other queries retrieved images that were consistently found to be irrelevant by most evaluators.

We also compared the magnitude of variability across human evaluators, with variability across queries. We first calculated the mAP from the precision curves of every query and evaluator, and then calculated the standard deviation in the mAP of every evaluator and of every query. The mean standard deviation over queries was 0.33, suggesting a large variability in the difficulty of image queries, as observed in Figure 3(C) . The mean standard deviation over evaluators was 0.25, suggesting that different evaluators had very different notions of what images should be regarded as "semantically similar" to a query image.

Finally, to estimate an "upper bound" on the difficulty of the task, we also computed the precision of the human evaluators themselves. For every evaluator, we used the rankings of all other evaluators as ground truth, to compute his precision. As with the ranks of OASIS, we computed the fraction of evaluators that marked an image as relevant, and repeated this separately for every query and human evaluator, providing a measure of "coherence" per query. Figure 3(D) shows the mean precision obtained by OASIS and human evaluators for every query in our data. For some queries OASIS achieves precision that is very close to that of the mean human evaluator. In many cases OASIS achieves precision that is as good or better than some evaluators.

5.2.4 SPEED AND SCALABILITY

We further studied how the runtime of OASIS scales with the size of the training set. Figure 4 shows that the runtime of OASIS, as found by early stopping on a separate validation set, grows linearly with the train set size. We compare this to the fastest result we found in the literature, based on a fast implementation of LMNN by Weinberger and Saul (2008). LMNN learns a Mahalanobis distance for k-nearest neighbor classification, aiming to have the nearest neighbors of a sample belong to the same class, and samples from different classes separated by a large margin. The LMNN algorithm is known to scale quadratically with the number of objects, although their experiments with MNIST data show that the active set of constraints grows linearly. This could be because MNIST has 10 classes only. In many real world data however, the number of classes typically grows almost linearly with the number of samples.

5.3 Caltech256 Data Set

To compare OASIS with small-scale methods we used the *Caltech256* data set (Griffin et al., 2007). This data set consists of 30607 images that were obtained from Google image search and from *PicSearch.com*. Images were assigned to 257 categories and evaluated by humans in order to ensure image quality and relevance. After we have pre-processed the images as described in Section 3 and filtered images that were too small, we were left with 29461 images in 256 categories. To allow comparisons with other methods in the literature that were not optimized for sparse representation, we also reduced the block vocabulary size d from 10000 to 1000. This processed data is available online at *http://ai.stanford.edu/∼gal/Research/OASIS*.

Using the Caltech256 data set allows us to compare OASIS with existing similarity learning methods. For OASIS, we treated images that have the same labels as similar. The same labels were used for comparing with methods that learn a metric for classification, as described below.

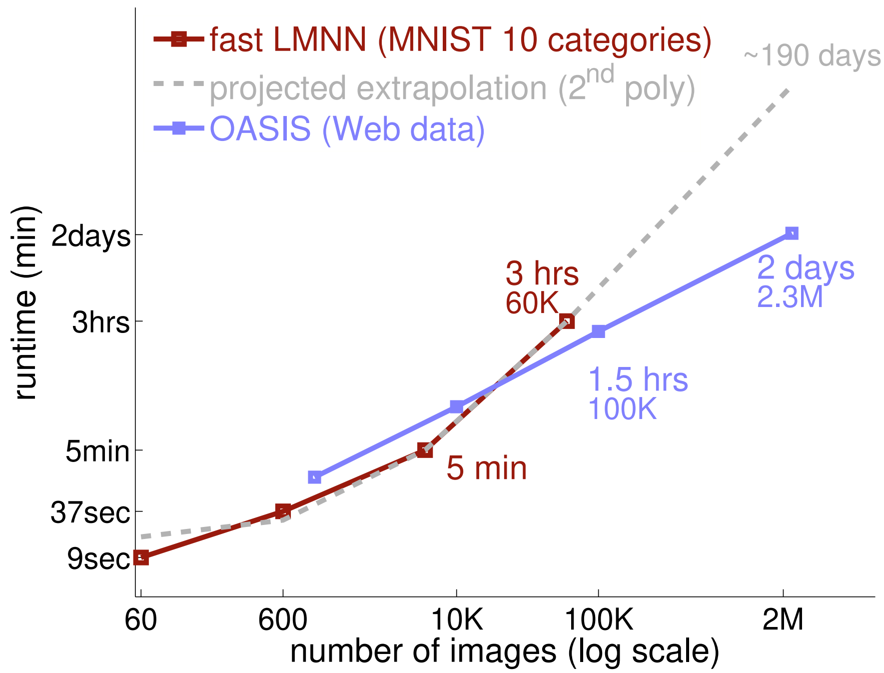

Figure 4: Comparison of the runtime of OASIS and fast-LMNN by Weinberger and Saul (2008), over a wide range of scales. LMNN results (on MNIST data) are faster than OASIS results on subsets of the web data. However LMNN scales quadratically with the number of samples, hence is three times slower on 60K images, and may be infeasible for handling 2.3 million images.

5.3.1 COMPARED METHODS

We compared the following approaches:

1. **OASIS.** - The algorithm described above in Section 2.1.

2. **Euclidean.** - The standard Euclidean distance in feature space. The initialization of OASIS using the identity matrix is equivalent to this distance measure.

3. **MCML** - Metric Learning by Collapsing Classes (Globerson and Roweis, 2006). This approach learns a Mahalanobis distance such that samples from the same class are mapped to the same point. The problem is written as a convex optimization problem, and we have used the gradient-descent implementation provided by the authors.

4. **LMNN** - Large Margin Nearest Neighbor Classification (Weinberger et al., 2006). This approach learns a Mahalanobis distance for k-nearest neighbor classification, aiming to have the k-nearest neighbors of a given sample belong to the same class while examples from different classes are separated by a large margin. As a preprocessing phase, images were projected to a basis of the principal components (PCA) of the data, with no dimensionality reduction, since

this improved the precision results. We also compared with a fast implementation of LMNN, that uses a clever scheme of maintaining a set of active constraints (Weinberger and Saul, 2008). We used the web data discussed above to compare with previously published results obtained with fast-LMNN on MNIST data (see Figure 4).

5. **LEGO** - Online metric learning (Jain et al., 2008a). LEGO learns a Mahalanobis distance in an online fashion using a regularized per instance loss, yielding a positive semidefinite matrix. The main variant of LEGO aims to fit a given set of pairwise distances. We used another variant of LEGO that, like OASIS, learns from relative distances. In our experimental setting, the loss is incurred for same-class examples being more than a certain distance away, and different class examples being less than a certain distance away. LEGO uses the LogDet divergence for regularization, as opposed to the Frobenius norm used in OASIS.

For all these approaches, we used an implementation provided by the authors. Algorithms were implemented in Matlab, with runtime bottlenecks implemented in C for speedup (except LEGO). We test below two variants of OASIS applied to the Caltech256 data set: a pure Matlab implementation, and one that has a C components. We used a C^{++} implementation of OASIS for the web-scale experiments described below.

We have also experimented with the methods of Xing et al. (2003) and RCA (Bar-Hillel et al., 2003). We found the method of Xing et al. (2003) to be too slow for the sets in our experiments. RCA is based on a per-class eigen decomposition that is not well defined when the number of samples is smaller than the feature dimensionality. We therefore experimented with a preprocessing phase of dimensionality reduction followed by RCA, but results were inferior to other methods and were not included in the evaluations below. RCA also did not perform well when tested on the full data, where dimensionality was not a problem, possibly because it is not designed to handle well sparse data.

5.3.2 EXPERIMENTAL PROTOCOL

We tested all methods on subsets of classes taken from the Caltech256 repository. Each subset was built such that it included semantically diverse categories, spanning the full range of classification difficulty, as measured by Griffin et al. (2007). We used subsets of sizes 10, 20, 50 and 249 classes (we used 249 classes since classes 251-256 are strongly correlated with other classes, and since class 129 did not contain enough large images). The full lists of categories in each set are given in Appendix B. For each set, images from each class were split into a training set of 40 images and a test set of 25 images, as proposed by Griffin et al. (2007).

We used cross-validation to select the values of hyper parameters for all algorithms except MCML. Models were learned on 80% of the training set (32 images), and evaluated on the remaining 20%. Cross validation was used for setting the following hyper parameters: the early stopping time for OASIS; the ω parameter for LMNN ($\omega \in \{0.125, 0.25, 0.5\}$), and the regularization parameter η for LEGO ($\eta \in \{0.02, 0.08, 0.32\}$). We found that LEGO was usually not sensitive to the choice of η, yielding a variance that was smaller than the variance over different cross-validation splits. Results reported below were obtained by selecting the best value of the hyper parameter and then training again on the full training set (40 images). For MCML, we used the default parameters supplied with the code from the authors, since its very long run time and multiple parameters made it non-feasible to tune hyper parameters on this data.

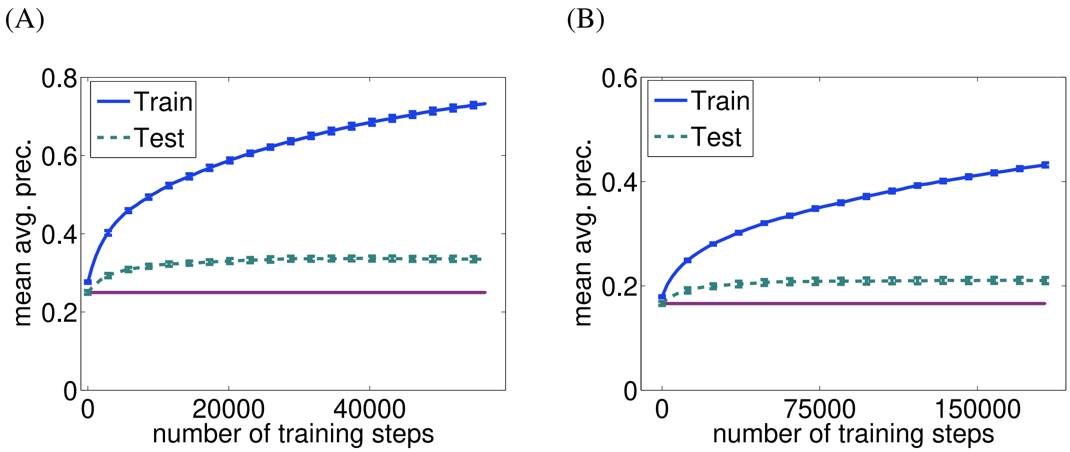

Figure 5: Mean average precision of OASIS as a function of the number of training steps. Error bars represent standard error of the mean over 5 selections of training (40 images) and test (25 images) sets. Performance is compared with a baseline obtained using the naïve Euclidean metric on the feature vector. C=0.1 **(A)** 10 classes. Test performance saturates around 30K training steps, while going over all triplets would require 2.8 million steps. **(B)** 20 classes.

5.3.3 RESULTS

Figure 5 traces the mean average precision over the training and the test sets as it progresses during learning. For the 10 classes task, precision on the test set saturates early (around 35K training steps), and then decreases very slowly.

Figure 6 and Table 4 compare the precision obtained with OASIS, with four competing approaches, as described above (Section 5.3.1). OASIS achieved consistently superior results throughout the full range of k (number of neighbors) tested, and on all four sets studied. Interestingly, we found that LMNN performance on the training set was often high, suggesting that it overfits the training set. This behavior was also noted by Weinberger et al. (2006) in some of their experiments.

OASIS achieves superior or equal performance, with a runtime that is faster by about two orders of magnitudes than MCML, and about one order of magnitude faster than LMNN. The run time of OASIS and LEGO was measured until the point of early stopping.

Table 5 shows the total CPU time in minutes for training each of the algorithms compared (measured on a standard 1.8GHz Intel Xeon CPU). For the purpose of a fair comparison with competing approaches, we tested two implementations of OASIS: The first was fully implemented Matlab. The second had the core of the algorithm implemented in C and called from Matlab.[3] LMNN code and MCML code were supplied by the authors and implemented in Matlab, with core parts implemented in C. LEGO code was supplied by the authors and fully implemented in Matlab.

Importantly, we found that Matlab does not make full use of the speedup that can be gained by sparse image representation. As a result, the C/C^{++} implementation of OASIS that we tested is significantly faster.

3. The OASIS code is available online at *http://ai.stanford.edu/∼gal/Research/OASIS*

10 classes	OASIS Matlab	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	Euclidean -
Mean avg prec	**33** $\pm$ 1.6	29 $\pm$ 1.7	27 $\pm$ 0.8	24 $\pm$ 1.6	23 $\pm$ 0.9
Top 1 prec.	**43** $\pm$ 4.0	39 $\pm$ 5.1	39 $\pm$ 4.8	38 $\pm$ 5.4	37 $\pm$ 4.1
Top 10 prec.	**38** $\pm$ 1.3	33 $\pm$ 1.8	32 $\pm$ 1.2	29 $\pm$ 2.1	27 $\pm$ 1.5
Top 50 prec.	**23** $\pm$ 1.5	22 $\pm$ 1.3	20 $\pm$ 0.5	18 $\pm$ 1.5	18 $\pm$ 0.7
20 classes	OASIS	MCML	LEGO	LMNN	Euclidean
Mean avg. prec	**21** $\pm$ 1.4	17 $\pm$ 1.2	16 $\pm$ 1.2	14 $\pm$ 0.6	14 $\pm$ 0.7
Top 1 prec.	**29** $\pm$ 2.6	26 $\pm$ 2.3	26 $\pm$ 2.7	26 $\pm$ 3.0	25 $\pm$ 2.6
Top 10 prec.	**24** $\pm$ 1.9	21 $\pm$ 1.5	20 $\pm$ 1.4	19 $\pm$ 1.0	18 $\pm$ 1.0
Top 50 prec.	**15** $\pm$ 0.4	14 $\pm$ 0.5	13 $\pm$ 0.6	11 $\pm$ 0.2	12 $\pm$ 0.2
50 classes	OASIS	MCML	LEGO	LMNN	Euclidean
Mean avg. prec.	**12** $\pm$ 0.4	$*$	9 $\pm$ 0.4	8 $\pm$ 0.4	9 $\pm$ 0.4
Top 1 prec.	**21** $\pm$ 1.6	$*$	18 $\pm$ 0.7	18 $\pm$ 1.3	17 $\pm$ 0.9
Top 10 prec.	**16** $\pm$ 0.4	$*$	13 $\pm$ 0.6	12 $\pm$ 0.5	13 $\pm$ 0.4
Top 50 prec.	**10** $\pm$ 0.3	$*$	8 $\pm$ 0.3	7 $\pm$ 0.2	8 $\pm$ 0.3

Table 4: Mean average precision and precision at top 1, 10, and 50 of all compared methods. Values are averages over 5 cross validation folds; $\pm$ values are the standard deviation across the 5 folds. A '*' denotes cases where a method took more than 5 days to converge.

classes	OASIS Matlab	OASIS Matlab+C	MCML Matlab+C	LEGO Matlab	LMNN (naive) Matlab+C	fast-LMNN Matlab+C
10	42 $\pm$ 15	0.12 $\pm$ .03	1835 $\pm$ 210	143 $\pm$ 44	337 $\pm$ 169	247 $\pm$ 209
20	45 $\pm$ 8	0.15 $\pm$ .02	7425 $\pm$ 106	533 $\pm$ 49	631 $\pm$ 40	365 $\pm$ 62
50	25 $\pm$ 2	1.6 $\pm$ .04	$*$	711 $\pm$ 28	960 $\pm$ 80	2109 $\pm$ 67
249	485 $\pm$ 113	1.13 $\pm$ .15	$*$	$**$	$**$	$**$

Table 5: Runtime (minutes) of all compared methods. Values are averages over 5 cross validation folds, $\pm$ values are the standard deviation across the 5 folds. A '$*$' denotes cases where a method took more than 5 days to converge. A '$**$' denotes cases where performance was worse than the Euclidean baseline.

5.4 Parallel Training

We presented OASIS as optimizing an objective function at each step. Since OASIS is based on the PA framework, it is also known to minimize a global objective of the form

$$\|\mathbf{W}\|_{Fro}^2 + C\sum_i l_i$$

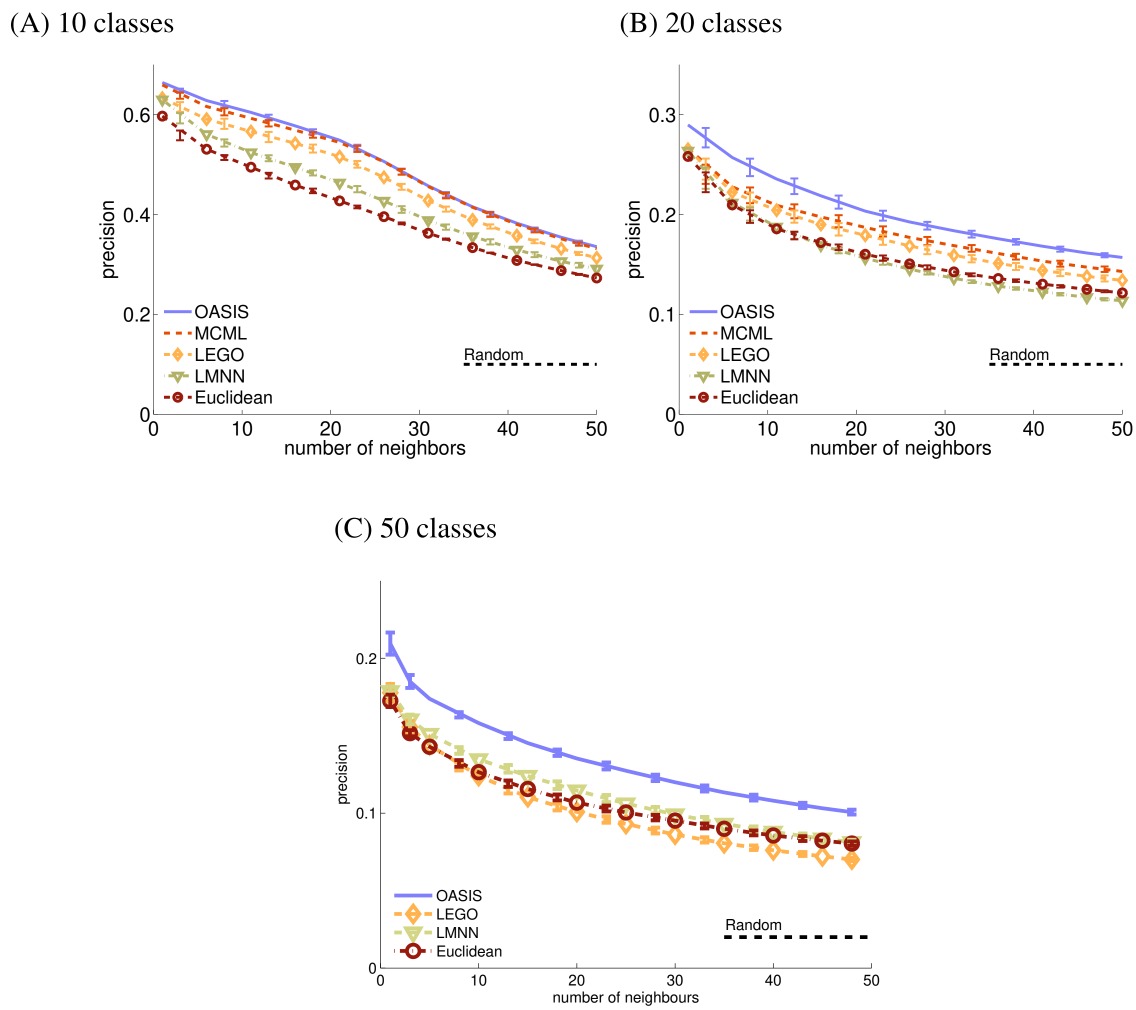

Figure 6: Comparison of the performance of OASIS, LMNN, MCML, LEGO and the Euclidean metric in feature space. Each curve shows the precision at top k as a function of k neighbors. The results are averaged across 5 train/test partitions (40 training images, 25 test images), error bars are standard error of the means (s.e.m.), black dashed line denotes chance performance. **(A)** 10 classes. **(B)** 20 classes. **(C)** 50 classes.

as shown by Crammer et al. (2006) This objective is convex since the losses l_i are linear in $\mathbf{W}$. For such convex functions, it is guaranteed that any linear combination of solutions is superior than each of the individual solutions. This property suggests another way to speed up training, by training multiple rankers in parallel and averaging the resulting models. Each of the individual models can be trained with a smaller number of iterations. Note however that there is no guarantee that the total CPU time is improved.

Figure 7 demonstrates this approach; we trained 5 or 10 rankers in parallel and plot the test set mean average precision as a function of the number of training iterations.

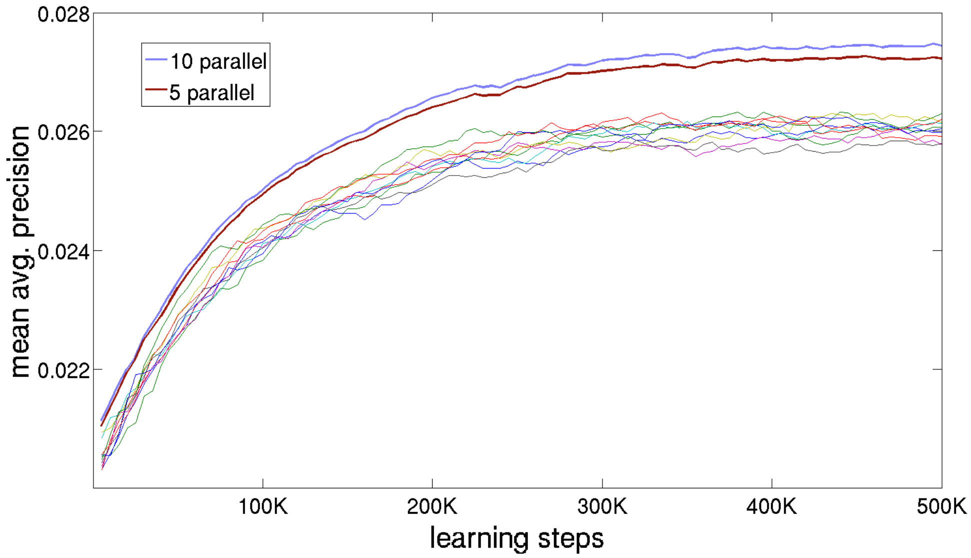

Figure 7: Comparing individual rankers and a linear combination of 5 and 10 rankers. Results are for an experiment with 249 classes of the Caltech256 data set.

.

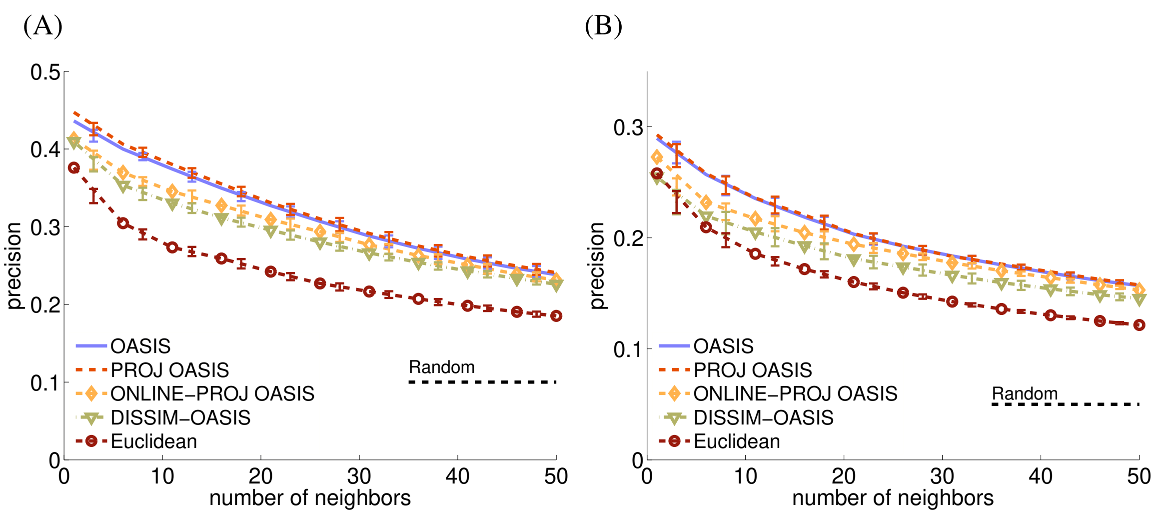

Figure 8: Comparison of Symmetric variants of OASIS. **(A)** 10 classes. **(B)** 20 classes.

6. Symmetry and Positivity

The similarity matrix **W** learned by OASIS is not guaranteed to be positive or even symmetric. Some applications, like ranking images by semantic relevance to a given image query are known to be non-symmetric when based on human judgement (Tversky, 1977). However, in some applications symmetry or positivity constraints reflect a prior knowledge that may help avoiding overfitting.

Furthermore positive $\mathbf{W}$ impose a Mahalanobis metric over the data, that can be further factorized to extract a linear projection of the data into a Euclidean space: $\mathbf{x}^T\mathbf{W}\mathbf{y} = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{y})$ such that $\mathbf{A}^T\mathbf{A} = \mathbf{W}$. Such projection $\mathbf{A}$ of the data can be useful for visualization and exploratory analysis of data for example in scientific applications. We now discuss variants of OASIS that learn a symmetric or positive matrices.

6.1 Symmetric Similarities

A simple approach to enforce symmetry is to project the OASIS model $\mathbf{W}$ onto the set of symmetric matrices $\mathbf{W}' = sym(\mathbf{W}) = \frac{1}{2}\left(\mathbf{W}^T + \mathbf{W}\right)$. The update procedure then consists of a series of gradient steps followed by projection to the feasible set (of symmetric matrices). This approach is sometimes called projected gradient, and we denote it here *Online-Proj-Oasis*. Alternatively, projection can also be applied after learning is completed (denoted here *Proj-Oasis*).

Alternatively, the asymmetric score function $S_{\mathbf{W}}(p_i, p_j)$ in the loss l_W can be replaced with a symmetric score

$$S'_{\mathbf{W}}(p_i, p_j) \equiv -(p_i - p_j)^T\ \mathbf{W}\ (p_i - p_j)\ .$$

and derive an OASIS-like algorithm (which we call *Dissim-Oasis*). The optimal update for this loss has a symmetric gradient $\mathbf{V}'^i = (p_i - p_i^+)(p_i - p_i^+)^T - (p_i - p_i^-)(p_i - p_i^-)^T$. Therefore, if $\mathbf{W}^0$ is initialized with a symmetric matrix (for example, the identity matrix) all $\mathbf{W}^i$ are guaranteed to remain symmetric. *Dissim-Oasis* is closely related to LMNN (Weinberger et al., 2006). This can be seen be casting the batch objective of LMNN, into an online setup, which has the form $err(W) = -\omega \cdot S'_{\mathbf{W}}(p_i, p_i^+) + (1-\omega) \cdot l'_{\mathbf{W}}(p_i, p_i^+, p_i^-)$. This online version of LMNN becomes equivalent to *D*issim-Oasis for $\omega = 0$.

Figure 8 compares the precision of the different symmetric methods with the original OASIS. All symmetric variants performed slightly worse, or equal to the original asymmetric OASIS. Asymmetric OASIS is also twice faster than DISSIM-OASIS. The precision of *Proj-Oasis* was equivalent to that of OASIS. This was because the asymmetric OASIS learning rule actually converged to an almost-symmetric model (as measured by a symmetry index $\rho(\mathbf{W}) = \frac{\|sym(\mathbf{W})\|_2}{\|\mathbf{W}\|_2} = 0.94$).

6.2 Positive Similarity

Most similarity learning approaches focus on learning metrics. In the context of OASIS, when $\mathbf{W}$ is positive semi definite (PSD), it defines a Mahalanobis distance over the images. The matrix square-root of $\mathbf{W}$, $\mathbf{A}^T\mathbf{A} = \mathbf{W}$ can then be used to project the data into a new space in which the Euclidean distance is equivalent to the $\mathbf{W}$ distance in the original space.

We experimented with positive variants of OASIS, where we repeatedly projected the learned model onto the set of PSD matrices, once every t iterations. Projection is done by taking the eigen decomposition $\mathbf{W} = \mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V}^T$ where $\mathbf{V}$ is the eigenvector matrix and $\mathbf{D}$ is the diagonal eigenvalues matrix limited to positive eigenvalues. Figure 9 traces precision on the test set throughout learning for various values of t.

The effect of positive projections is complex. First, continuously projecting once every few steps helps to reduce overfitting, as can be observed by the slower decline of the blue curve (upper smooth curve) compared to the orange curve (lowest curve). However, when projection is performed after many steps (instead of continuously), performance of the projected model actually outperforms the continuous-projection model (upper jittery curve). The reason for this effect is likely to be that the

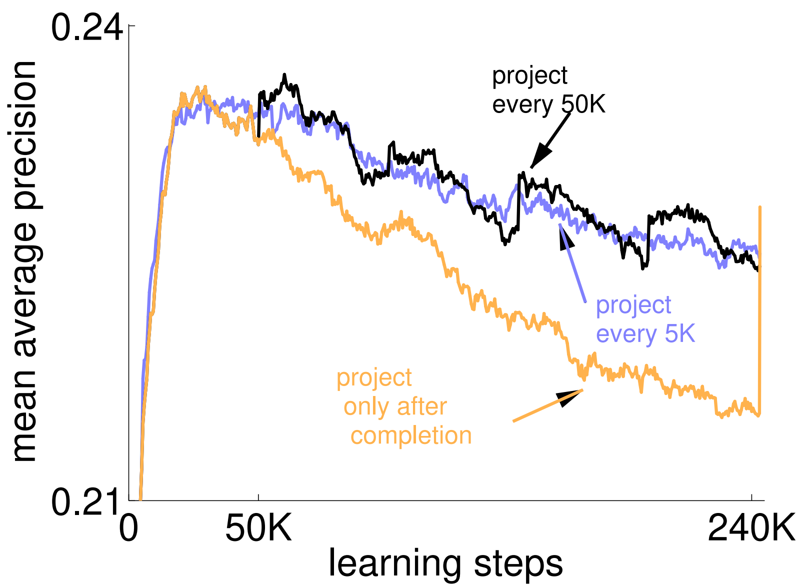

Figure 9: Mean average precision (mAP) during training for three PSD projection schemes, using the set of 20 classes from caltech256.

estimates of the positive sub-space are very noisy when only based on a few samples (see also Chen et al. 2009, Section 2.1). Indeed, accurate estimation of the negative subspace is known to be a hard problem, because small perturbations can turn a negative but small eigenvalue, into a small but positive one. As a result, the set of vectors selected based on having positive eigenvalues, is highly variable. We found that this effect was so strong, that the optimal projection strategy is to avoid projection throughout learning completely. Instead, projecting into PSD after learning (namely, after a model was chosen using early stopping) provided the best performance in our experiments.

An interesting alternative to obtain a PSD matrix was explored by Kulis et al. (2009) and Jain et al. (2008a). Using a LogDet divergence between two matrices $D_{ld}(X,Y) = tr(XY^{-1}) - log(det(XY^{-1}))$ ensures that, given an initial PSD matrix, all subsequent matrices will be PSD as well. It would be interesting to test the effect of using LogDet regularization in the OASIS setup.

7. Discussion

We have presented OASIS, a scalable algorithm for learning image similarity that captures both semantic and visual aspects of image similarity. Three key factors contribute to the scalability of OASIS. First, using a large margin online approach allows training to converge even after seeing a small fraction of potential pairs. Second, the objective function of OASIS does not require the similarity measure to be necessarily a metric during training, although it appears to naturally converge to a symmetric solution. Finally, we use a sparse representation of low level features which allows computing scores very efficiently.

We found that OASIS performs well in a wide range of scales: from problems with thousands of images, where it slightly outperforms existing metric-learning approaches, to large web-scale problems, where it achieves high accuracy, as estimated by human evaluators.

OASIS differs from previous methods in that the similarity measure that it learns is not forced to be a metric, or even symmetric. When the number of available samples is small, it is useful to add constraints that reflect prior knowledge on the type of similarity measure expected to be learned. However, we found that these constraints were not helpful even for problems with a few hundreds of samples. Interestingly, human judgements of pairwise similarity are known to be asymmetric, a property that can be easily captured by an OASIS model.

OASIS learns a class-independent model: it is not aware of which queries or categories were shared by two similar images. As such, it is more limited in its descriptive power and it is likely that class-dependent similarity models could improve precision. On the other hand, class-independent models could generalize to handle classes that were not observed during training, as in transfer learning. Large scale similarity learning, applied to images from a large variety of classes, could therefore be a useful tool to address real-world problems with a large number of classes.

Acknowledgments

This work was supported by the Israeli Science Foundation (ISF 1001/08). We thank Andrea Frome for very helpful discussions and comments on the manuscript. We thank Amir Globerson, Killian Weinberger and Prateek Jain, each providing an implementation of their method for our experiments.

Appendix A. Human Evaluation

The following text was given as instructions to human evaluators when judging the relevance of images to a query image.

```
Scenario:
     A user is searching images to use in a presentation he/she plans to
  give.  The user runs a standard image search, and selects an image,
  the ``query image''.  The user then wishes to refine the search and
  look for images that are SEMANTICALLY similar to the query image.
     The difficulty lies, in the definition of ``SEMANTICALLY''. This can
  have many interpretations, and you should take that into account.
     So for instance, if you see an image of a big red truck, you can
  interpret the user intent (the notion of semantically similar) in
  various ways:
    - any big red truck
    - any red truck
    - any big truck
    - any truck
    - any vehicle
  You should interpret ``SEMANTICALLY'' in a broad sense rather than
  in a strict sense but feel free to draw the line yourself (although
```

```
  be consistent).

Your task:
     You will see a set of query images on the left side of the screen,
  and a set of potential candidate matches, 5 per row, on the
  right. Your job is to decide for each of the candidate images if it
  is a good semantic match to the query image or not. The default is
  that it is NOT a good match. Furthermore, if for some reason you
  cannot make-up your mind, then answer ''can't say''.
```

Appendix B. Caltech256 Class Sets

- **10 classes**: *bear, skyscraper, billiards, yo-yo, minotaur, roulette-wheel, hamburger, laptop-101, hummingbird, blimp.*
- **20 classes***: airplanes-101, mars, homer-simpson, hourglass, waterfall, helicopter-101, mountain-bike starfish-101, teapot, pyramid, refrigerator, cowboy-hat, giraffe, joy-stick, crab-101, bird-bath, fighter-jet tuning-fork, iguana, dog.*
- **50 classes***: car-side-101, tower-pisa, hibiscus, saturn, menorah-101, rainbow, cartman, chandelier-101, backpack, grapes, laptop-101, telephone-box, binoculars, helicopter-101, paper-shredder, eiffel-tower, top-hat, tomato, star-fish-101, hot-air-balloon, tweezer, picnic-table, elk, kangaroo-101, mattress, toaster, electric-guitar-101, bathtub, gorilla, jesus-christ, cormorant, mandolin, light-house, cake, tricycle, speed-boat, computer-mouse, superman, chimp, pram, fried-egg, fighter-jet, unicorn, greyhound, grasshopper, goose, iguana, drinking-straw, snake, hot-dog.*
- **249 classes***: classes 1-250, excluding class 129 (leopards-101), which had less than 65 large enough images.*

References

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proc. of 20th International Conference on Machine Learning (ICML)*, page 11, 2003.

L. Bottou. Large-scale machine learning and stochastic algorithms. In *NIPS 2008 Workshop on Optimization for Machine Learning*, 2008.

Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, and L. Cazzanti. Similarity-based classification: Concepts and algorithms. *The Journal of Machine Learning Research*, 10:747–776, 2009.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, 7:551–585, 2006.

J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM Press New York, NY, USA, 2007.

P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision (ECCV)*, pages 97–112, 2002.

S.L. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *International Conference on Computer Vision*, pages 1–8, 2007.

A. Globerson and S. Roweis. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*, 18:451, 2006.

D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(8):1371–1384, 2008.

D. Grangier, F. Monay, and S. Bengio. Learning to retrieve images from text queries with a discriminative model. In *International Conference on Adaptive Multimedia Retrieval (AMR)*, 2006.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2006.

P. Jain, B. Kulis, I. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems*, volume 22, 2008a.

P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008b.

J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *International Conference on Image and Video Retrieval*, pages 24–32, 2004.

B. Kulis, M.A. Sustik, and I.S. Dhillon. Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10:341–376, 2009.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research (JMLR)*, 5:27–72, 2004.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

W.S. Noble. Multi-kernel learning for biology. In *NIPS 2008 workshop on kernel learning*, 2008.

T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002.

P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. J. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, pages 883–890, 2005.

N. Rasiwasia and N. Vasconcelos. A study of query by semantic example. In *3rd International Workshop on Semantic Learning and Applications in Multimedia*, 2008.

R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 367–373. ACM New York, NY, USA, 2006.

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. Bradford Book, 2004.

V. Takala, T. Ahonen, and M. Pietikainen. Block-based methods for image retrieval using local binary patterns. In *Scandinavian Conference on Image Analysis (SCIA)*, 2005.

K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision (IJCV)*, 56(1):17 – 36, 2004.

A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007. URL `http://dspace.mit.edu/handle/1721.1/37291`.

A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 18:1473, 2006.

K.Q. Weinberger and L.K. Saul. Fast solvers and efficient implementations for distance metric learning. In *ICML25*, pages 1160–1167, 2008.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528, Cambridge, MA, 2003. MIT Press.

L. Yang. Distance metric learning: A comprehensive survey. Technical report, Michigan State University, 2006.