Hierarchical Semantic Indexing for Large Scale Image Retrieval

Jia Deng[1,3]
Princeton University[1]

Alexander C. Berg[2]
Stony Brook University[2]

Li Fei-Fei[3]
Stanford University[3]

Abstract

This paper addresses the problem of similar image retrieval, especially in the setting of large-scale datasets with millions to billions of images. The core novel contribution is an approach that can exploit prior knowledge of a semantic hierarchy. When semantic labels and a hierarchy relating them are available during training, significant improvements over the state of the art in similar image retrieval are attained. While some of this advantage comes from the ability to use additional information, experiments exploring a special case where no additional data is provided, show the new approach can still outperform OASIS [6], the current state of the art for similarity learning. Exploiting hierarchical relationships is most important for larger scale problems, where scalability becomes crucial. The proposed learning approach is fundamentally parallelizable and as a result scales more easily than previous work. An additional contribution is a novel hashing scheme (for bilinear similarity on vectors of probabilities, optionally taking into account hierarchy) that is able to reduce the computational cost of retrieval. Experiments are performed on Caltech256 and the larger ImageNet dataset.

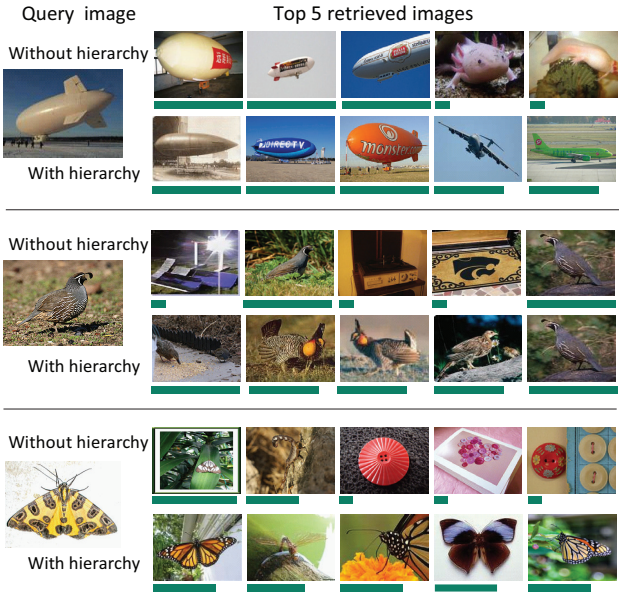

Figure 1. Images retrieved by exploiting hierarchy versus those without considering hierarchy. Green bars show ground truth similarity to the query, defined based on the category hierarchy(see Sec. 5.2). Longer bars indicate more similarity.

1. Introduction

This paper addresses the problem of similar image retrieval – given a query image, find similar images in a large image collection – as depicted in figure 1. As illustrated there, results show that exploiting hierarchical relationships can significantly improve retrieval accuracy. Incorporating hierarchical relationships is becoming more important as datasets grow larger. The potential benefit is largest when categories are sampled "densely" and fine grained distinctions must be made (e.g. [4, 7]). In order to handle such large scale data, computational efficiency and scalability is a critical aspect to effective use of hierarchy in retrieval.

Our approach demonstrates how to effectively incorporate prior human knowledge in the form of a hierarchical structure defined on semantic attributes of images. For instance given semantic attributes like containing a horse, dog, or windmill, a predefined hierarchy might let us know that an image containing a horse would be more similar to one containing a dog than to an image containing a windmill. It is feasible to specify a hierarchical structure in terms of semantic attributes, but may be quite difficult to do so directly in terms of low level features.

The current state of the art for similar image retrieval stems from a strong line of work on learning the underlying similarity function used for retrieval [6, 33, 15]. In that work, the goal is to learn a function that computes similarity directly from low level feature vectors from images, and does not allow variable measures of similarity that could encode hierarchical structure. It may be possible to adapt some of those strategies to take into account variable similarities for hierarchical structure, but would require modification of the techniques, and would not necessarily improve the scalability or parallelism of the approaches.

Our approach takes a different track, **learning to recognize semantic attributes of images, and then using a**

predefined comparison function – based on a known hierarchal structure – to produce a similarity score for retrieval. Learning to recognize semantic attributes can be easily parallelized, making this approach very scalable. That this approach requires labeled data for semantic attributes is potentially limiting, but in practice almost every single experiment in the related work on similarity learning begins from data with labels, such as the categories in Caltech256, or queries that produced the images in OASIS [6]. Furthermore, for non-overlapping categories, it is possible to reconstruct the category labels directly from the training data used for OASIS [6], LMNN [33], MCML [15], and other techniques. We show that significant improvements over the state of the art are possible when labels and a hierarchy are known or when labels can be inferred but hierarchy is not available [1]. Nevertheless, when labels are truly un-available and cannot be inferred, the proposed technique will not be appropriate or optimal.

Once a similarity function is determined the next challenge is efficient retrieval of the most similar database images for a query, with respect to the hierarchical similarity. This paper presents **a novel hashing strategy that provides a sub-linear time solution for retrieval and forms a generally usable component on its own.** When combined with the training for the semantic classifiers that is linear in the input data and inherently parallelizable, the overall system is very scalable. To make this concrete, our semantic index structure can be built on 600,000 images in 14 days on a single cpu, or because of the easy parallelizability, in 20 minutes of wall clock time using 1000 cpus. Using hashing, retrieval of similar images in the resulting index can be performed in 3 milliseconds per query with accuracy close to 90% of brute force search and computational cost less than 0.001 times that of brute force.

2. Related Work

We review closely related work on hierarchy, similarity learning, semantic indexing, and hashing for retrieval.

Work on recognition in computer vision has reached the scale – in terms of number of classes – where hierarchical relationships between classes begin to 1) be non-trivial, 2) have an impact on recognition performance, and 3) have the potential to improve recognition accuracy. This has been demonstrated by work putting existing datasets into hierarchies [17], and building large new datasets – e.g. TinyImages [29] and ImageNet [8] – based on the hierarchical semantic structure in WordNet [12] a major project of the linguistics and natural language processing community. This line of work has begun to reveal both the effects of hierarchical structure on classification accuracy [4, 29, 7],and hints at the promise of exploiting such structure for classification when evaluated in terms of the hierarchy [7] as well as showing improved classification given very small amounts of training data [13].

In this paper we demonstrate that it is possible to exploit hierarchical structure for a different but related task – similar image retrieval – gaining significant improvements in accuracy. This complements recent work advancing the learning of similarity functions, especially for retrieval, *e.g.* [15, 33, 6], that does not yet address hierarchy. Some of our experiments compare with OASIS from Chechik *et al*. [6], the current state of the art in learning similarity functions for retrieval [2], and we demonstrate significant improvement by adding hierarchy. Furthermore the proposed techniques are easily parallelizable, allowing better scaling than [6] (even without hierarchy) which already improved computational efficiency significantly over other techniques [15, 33].

Many of the improvements shown stem from exploiting high level knowledge in the form of a semantic hierarchy. This is related to recent research in explicitly estimating high level semantic attributes for recognition [21, 22, 11, 23, 31]. In particular [21] allows retrieval queries using language to refer to the semantic attributes of faces. We consider queries specified by an image, and add a hierarchical relationship between semantic attributes. Recent work [31] considers a representation similar in spirit to the semantic representation we use, but focuses on using the representation for classification – using multiple training examples for a class specific query, as opposed to a single example as a query – instead of retrieval. There is also related work from the multimedia and information retrieval community, especially on TrecVid *e.g.* [2, 18] (and references therein), that explicitly train object or concept "detectors" and use their output as features for retrieval based on high level (or textual) queries.

Efficient retrieval with respect to a similarity function is important for very large scale settings. Significant work has been done on hashing for the related problem of finding approximate nearest neighbors [1, 9]. In our setting, retrieval using bilinear similarity on vectors of probabilities is a core subroutine, and we introduce a novel hashing scheme to accomplish this. Note that this is a data independent hashing approach in contrast to recent approaches based on learning hashing functions for vision [20, 28, 30].

3. Exploiting Hierarchy for Retrieval

In order to exploit hierarchical knowledge in retrieval, we consider the core subroutine of evaluating the similarity Sim between two images. Retrieval consists of finding the images from a large database with greatest similarity to a query image. In previous work, much of the

[1] This is actually the case in most work on similarity learning [33, 15, 6].

[2] Closely related in technique to [14] for classification.

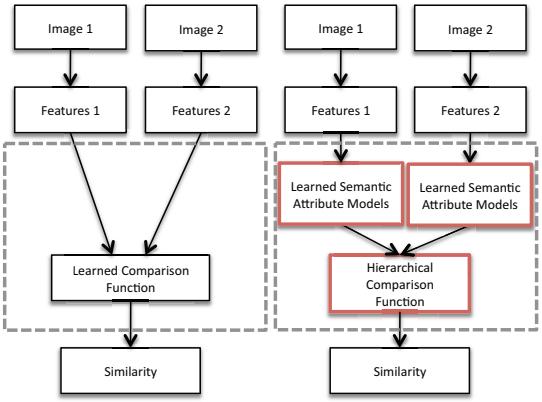

Figure 2. Compared to previous work (left) our approach to learning similarity functions (right) separates the learned similarity function into two parts, an estimate of probabilities for semantic labels, and a hierarchical comparison function. The hierarchical comparison function is deterministically built from prior knowledge and only the semantic models are learned.

effort in building a system for retrieval was in learning the similarity function that mapped low-level image features computed from image to a similarity value. For images a,b,c, and d, let $f(\cdot)$ denote their low level features, then training can be performed by considering constraints on pairs, *e.g.*, a and b are more similar than c and d so $Sim(f(a), f(b)) > Sim(f(c), f(d)) + 1$ as in [33]. The state of the art OASIS [6] considers triples of images, where a was supposed to be more similar to b than to c, yielding the constraint $Sim(f(a), f(b)) < Sim(f(a), f(c)) + 1$. These constraints were used to *learn* a matrix L so that $Sim(f(a), f(b)) = f(a)'Lf(b)$.

As illustrated in Figure 2, our approach computes similarity by first estimating probabilities of semantic attributes for an image $s(f(a))$, based on low level image features. Then we use the prior knowledge of the semantic hierarchical relationship to deterministically compute a hierarchical similarity matrix, S, and the similarity function is $Sim(a, b) = s(f(a))^T Ss(f(b))$ (Sec. 3.1 & 3.2). Not only does this allow exploiting hierarchical knowledge, but learning is only needed to build the models for semantic attributes – a process that is much easier to parallelize than previous approaches to learning similarity.

3.1. Encoding hierarchy in semantic similarity

The core of our approach is to use prior knowledge of a hierarchy between semantic attributes to compute similarity for retrieval. We start by discussing a non-probabilistic version of such a similarity, and describe an image a by a set of binary semantic attributes $\{1 \dots K\}$. The attributes can be object categories("is dog"), part relations("has legs"), visual descriptions("is black") or in fact any predicate about the image. We will mainly focus on object class category attributes as they are the dominant type of attributes currently used and have been extensively studied, but the approach will extend to arbitrary attributes.

Given the attributes, the similarity between two images a and b can then be measured as how well their attributes match. Specifically, let $\delta_i(a) \in \{0, 1\}$ be the indicator function of image a having attribute i. We define the similarity as $Sim^*(a, b) = \sum_{i,j} \delta_i(a) S_{ij} \delta_j(b)$, where $S \in \mathbb{R}^{K \times K}$ and S_{ij} is a "matching score" between attribute i and j, *i.e.*, the semantic similarity based on prior human knowledge. We refer to S as *the prior matrix*.

This is a very general form and encompasses a large class of semantic similarities. For object category attributes, a dominate relationship is the "is a" relation that naturally organizes them into a semantic hierarchy. In this case, S can be derived by measuring the closeness of categories relative to the hierarchy. For instance, let $S(i, j) = \xi(\pi(i, j))$, where $\pi(i, j)$ is the lowest common ancestor of category i and j and $\xi(\cdot) : \{1 \dots K\} \rightarrow \mathbb{R}$ is some real function that is non-decreasing going down the hierarchy,*i.e.* the lower the lowest shared ancestor, the more similar categories i and j are. For example, "is donkey" is much more similar to "is horse" than "is keyboard" because "donkey" shares a lower level common ancestor "equine" with "horse" than "object" with "keyboard". More concretely $\xi(\cdot)$ can be based on the height of the node [32]. Note that there are other possible ways to obtain similarity between attributes such as automatic text mining[27] when such a manually constructed hierarchy is not available.

A special case is when the attributes are mutually exclusive categories and S is the identity matrix, so $Sim^*(a, b)$ simply indicates whether a and b belong to the same category. Refer to this as a "flat" setting as there is no hierarchical relationship between the attributes. The attributes are treated as either identical or different and a retrieval system optimized for this similarity would be incapable of ranking "horse" higher than "keyboard" given a query "donkey". This is setting where most existing techniques were developed and evaluated [15, 33, 6].

So far our similarity employs hard assignment of binary attributes. However there is often uncertainty in representing images with semantic attributes. On one hand, natural language is inherently ambiguous and categories overlap. There will always be objects that evade exact categorization. Also perfect classification of semantic attributes is unrealistic. Thus instead of using binary indicators, we represent an image a as a vector $s(f(a)) = x \in \mathbb{R}^K$ where $x_i = \Pr(\delta_i(a) = 1|a)$, *i.e.* the probability that image x has attribute i. Given image a, b and their vector of probabilities x, y, we redefine the similarity to be the expectation of the non-probabilistic version, *i.e.* $Sim(a, b) = \mathbb{E}\, Sim^*(a, b) = \sum_{i,j} x_i S_{ij} y_j$, or simply $Sim(a, b) = x^T Sy$. This is assuming that image a and b are drawn independently, as is

valid for most retrieval settings. We will refer to this form of similarity $x^T Sy$ as *bilinear similarity*.

Note that although we have mainly used mutually exclusive object categories as attributes in our discussion and will also focus on this case in the experiments due to availability of datasets, our formulation is not restricted to mutually exclusive categorization of images. An image can have multiple objects and thus any number of attributes "turned on".

3.2. Learning semantic attributes

Once the prior matrix S is given, to compute the bilinear similarity $x^T Sy$, a critical step is to learn models of semantic attributes and obtain probabilities. For large scale retrieval, important considerations are scalability and efficiency of learning, as real world retrieval systems need to handle tens of thousands of semantic attributes and to train from very large datasets.

To obtain the probabilistic attribute representation, we first learn binary classifiers for each semantic attribute independently. For example, in the case of category attributes, we train 1-vs-all linear SVM for each category. Then we calibrate the outputs of the classifiers into probabilities. In our experiments, we fit a sigmoid function to each SVM classifier [25] to convert the output into a probability. For non-overlapping categories, we further normalize the probabilities to form a vector whose entries sum to one [3].

Note that both steps are easily parallelizable as the classifiers and sigmoid functions can be learned independently. Also learning the semantic attributes is decoupled from the specification of the prior matrix S. In contrast to existing similarity learning algorithms that learn similarity from low level features, our scheme can be adapted to new similarity measures by simply replacing the prior matrix in retrieval time, without relearning of the attribute models.

4. Efficient indexing

Efficiency is a major challenge for large scale retrieval Merely considering object categories as attributes may result in a probability vector of tens of thousands dimensions [3]. Computing the similarity between a query and each database image thus becomes prohibitively expensive.

We introduce a novel technique based on locality sensitive hashing(LSH) [1] to achieve sublinear retrieval time for bilinar similarity. The key is to construct on a family of hash functions $\mathcal{H}$ such that $\Pr_{h\in\mathcal{H}}(h(x) = h(y)) = Sim(x, y)$ [5] or $\Pr(h_1(x) = h_2(y)) = Sim(x, y)$ with h_1 and h_2 drawn independently [9]. For a query point y, one retrieves the database points from the bin of $h(y)$ and rerank them to produce the final results. In practice, one may concatenate multiple hash functions to reduce false collision and use multiple hash tables to increase recall.

For our bilinear similarity $Sim(x, y) = x^T Sy$ where x and y are vectors of probabilities, we provide theoretical results on sufficient conditions of S and corresponding construction techniques, informally: *(1)If S is element-wise non-negative, symmetric and diagonally dominant, then there exists a construction(Lemma A.2); (2)If S is derived from a hierarchy such that classes sharing lower common ancestors have higher similarity, then there exists a construction(Lemma A.7).*

Constructions and proof sketches are in the appendix with detailed proofs in supplementary material. The hashing construction for a special case, where the semantic attributes are non-overlapping category labels and S is identity matrix, is especially simple: $h(x)$ is an integer from 1 to K sampled according to the multinomial distribution x. In implementation, h is parametrized by a uniformly drawn real number $p \in [0, 1)$ and returns the index of the interval where p falls in x.

One closely related existing technique is the random hyperplane LSH [5] for approximating cosine similarity $Sim(x, y) = \frac{x^T y}{\|x\|\|y\|}$ that measures the angle between x and y, different from ours due to the L2 normalization. We compare empirical performance with it in Sec. 5.5.

5. Experiments

5.1. Datasets and evaluation criteria

We use Caltech256 [16] and ILSVRC [32], a subset of ImageNet [8] with 1000 classes and 1.2 million images [4]. We use Caltech256 to compare with existing similarity learning algorithms and use ILSVRC for large scale experiments. Both datasets assign one class label per image. The categories of ILSVRC are hierarchically organized.

For both datasets, we split the data into training and test, use the training set to learn semantic models and use the test to evaluate retrieval performance. For retrieval we obtain the top k neighbors by brute force scan except in Sec. 5.5. Unless otherwise noted, all evaluation is done by using each of the test images to query against the rest of the test images and reporting the average.

We use a ranking based criteria for evaluation. Given a similarity function, $Sim(a_i, a_j) \in [0, \infty)$, between images a_i and a_j, it can be used assign a rank, $r_i^q \in \{1, \ldots, n\}$ to n images $\{a_i\}_{i=\{1,\ldots,n\}}$ in a dataset with respect to a query image q so that $r_i^q <= r_j^q$ iff $Sim(q, a_i) >= Sim(q, a_j)$. Let $Sim_g(x, y)$ be "ground truth" or desired values of the similarity function. We can evaluate a ranking of k data

[3] Note that the probabilities can be made more accurate by joint calibration. For example, for non-overlapping categories one can use random forest of probability estimation trees(PET) [26] to obtain more accurate multiclass probabilities. We find that for large training data the simplicity and efficiency outweighs the marginal accuracy gain from PETs.

[4] we do not use the newly collected validation and test sets as they are too small for retrieval evaluation.

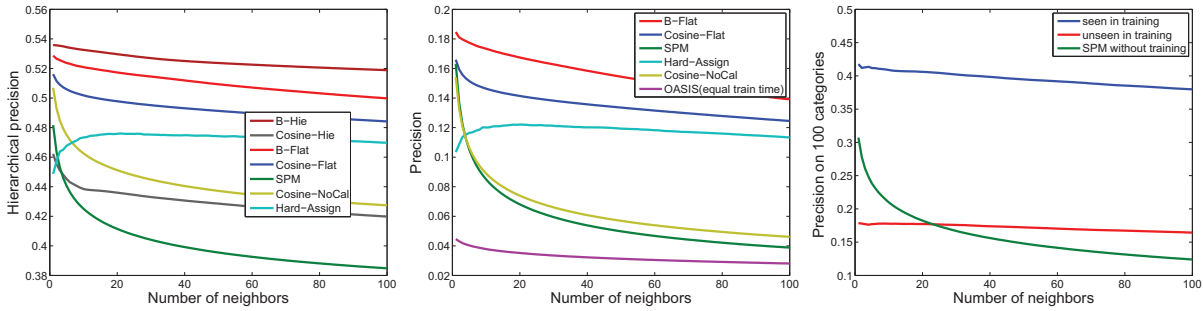

Figure 3. Precision vs. rank for similarity based retrieval on ILSVRC images from 1000 categories. **Left:** Hierarchical precision of our bilinear similarity with hierarchy encoded prior matrix (B-Hie) against others as described in Sec. 5.2. For all curves, stdev by swapping training and test is too small to show (≤ 0.002). **Middle:** Flat precision of our bilinear similarity with identity prior matrix(B-Flat) against others as described in Sec. 5.3. Stdev by swapping training and test is too small to show (≤ 0.001). **Right:** Flat precision on a subset 100 categories(Sec. 5.4). Using training data from the 100 categories (*"seen in training"*) performs the best, but training on 900 categories not including any of the 100 test categories (*"unseen in training"*) compares favorably to directly using SPM without any training.

items with respect to a query q by a precision,

$$p(Sim, q, Sim_g, k) = \frac{\sum_{r_i^q=1}^{k} Sim_g(q, a_i)}{\max_o \sum_{i=1}^{k} Sim_g(q, a_{o_i})} \quad (1)$$

The numerator is the sum of ground truth similarities for the most similar k database items based on similarity s for a query q. The denominator is the sum of ground truth similarities for the best possible k database items. The complexity of the evaluation function allows it to represent the standard "precision at k" for category labels when $Sim_g(a_i, a_j) \in \{0, 1\}$ is 1 for a_i and a_j with the same category label and 0 otherwise, as well as more general scores when $Sim_g(a_i, a_j) \in [0, 1]$ is a more nuanced measure of similarity, for instance based on hierarchical relationships between semantic categories.

5.2. Hierarchical retrieval

We define the ground truth similarity in a similar way the hierarchical cost is defined in ILSVRC [32]. Let $h(\pi(i, j))$ be the height of the lowest common ancestor $\pi(i, j)$ between class i and class j on the category hierarchy. The height of a node is the length of the longest path to one of its leaf node (leaf nodes have height 0). Similarity between class i and class j is then defined as $1 - h(\pi(i, j))/h^*$, where h^* is the height of the root node(19 for ILSVRC). All classes have similarity 1 to itself. We can then define the ground truth similarity between image a with ground truth class c_a and image b with c_b as $Sim_g(a, b) = 1 - h(\pi(c_a, c_b))/h^*$. Precision at top k as in Eqn. 1 is then the average class similarity between the query and top k returned images divided by the maximum possible similarity from the dataset (perfect would be 1). We refer to this criteria as *hierarchical precision*.

We evaluate our similarity on hierarchical precision on ILSVRC dataset. We split the ILSVRC images 50%-50% as training and test. To learn the semantic attributes, we train binary linear SVMs using LIBLINEAR [10] on sparse 21k dimensional vectors formed by a three level SPM [24] on the published SIFT visual words from a 1000 word codebook [32]. We use 2-fold cross validation to determine the parameter C. Probability calibration is done using Platt's scaling [25] during cross validation. In retrieval, we set the prior matrix S such that S_{ij}=1 $- h(\pi(i, j))/h^*$, matching the definition of hierarchical precision.

We compare our bilinear similarity with hierarchy encoded prior matrix (**B-Hie**) with various baselines: (1)**SPM**: ranking the images by intersection kernel on SPM histograms of visual words, representing low level feature based methods that do not use learning; (2) **Hard-Assign**: classifying the query image to the most likely class and ranking others by their probabilities of belonging to this class, equivalent to retrieval by annotation. (3)**Cosine-NoCal**: using cosine similarity of the raw outputs of semantic classifiers without probability calibration; (4)**Cosine-Flat**: using cosine similarity of the probabilities, same as Cosine-NoCal excpet with probability calibration; (5)**Cosine-Hie**: same as bilinear similarity with hierarchy encoded S except with L2 normalized probability vectors; (6)**B-Flat**: using bilinear similarity but without encoding the hierarchy in the prior matrix, *i.e.* with S set to identity.

We present the results in Fig. 3(left). Bilinear similarity with hierarchy encoded(B-Hie) achieves significantly better precision than all others. It also demonstrates that all components of our similarity are essential: (1)learning semantic attributes is important as SPM(directly using low level features without learning) is quite effective for the first few nearest neighbors but for the rest of the curve performs significantly worse than those that use semantic classifiers; (2)probabilistic representation significantly improves over hard assignment(B-Flat versus Hard-Assign); (3)probability calibration is important as using raw classifier outputs

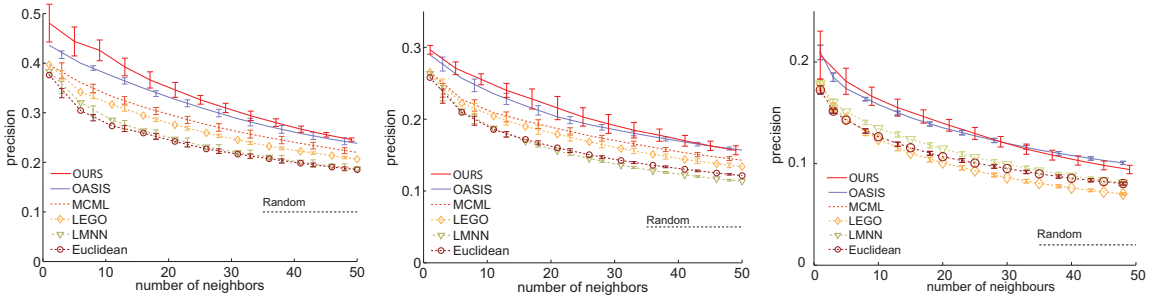

Figure 4. Comparison with (dis-)similarity learning methods including OASIS [6], MCML [15], LEGO [19] and LMNN [33] on 10 classes(left), 20 classes(middle) and 50 classes(right) from Caltech256. All curves except ours are from [6].

(Cosine-NoCal) is significantly worse than the calibrated counterparts(Cosine-Flat & B-Flat) (4)cosine similarity performs significantly worse than bilinear similarity due to L2 normalization of probability vectors(Cosine-Flat versus B-Flat & Cosine-Hie versus B-Hie); (5)most importantly, encoding hierarchy into the prior matrix significantly improves precision, as quantitatively shown by B-Hie versus B-Flat and qualitatively by Fig. 1, where images from nearby classes also rank higher.

5.3. Flat retrieval

Our method is motivated by hierarchical retrieval. However, most existing work is optimized for a special case where the ground truth similarity between two images are 1 if they are from the same categories and 0 otherwise. The retrieval precision at top k as defined in Eqn. 1 is then percentage of the images from the same class of the query image. We refer to this criteria as *flat precision*.

To effectively compare with existing work we adapt our bilinear similarity to this special case by simply setting the prior matrix to identity in retrieval.

We experimented with our method on Caltech256 [16] in the same setting as in evaluating OASIS [6]. We use linear SVMs as the semantic classifiers, trained on the same features published by the authors of [6], parameter C determined by by 5-fold cross validation. We report results from 5 random splits of training and testing data (40 training images and 25 test images per class), as in [6]. Figure 4 shows our method is on par with OASIS and significantly outperforms all other algorithms. Moreover, our scheme is more efficient. For 50 classes, a single run of OASIS takes 96 seconds to converge [6], while learning all 50 linear SVMs and sigmoid functions for probability estimation (including 5 trials of parameter C and 5 fold cross validation for each trial) take 12 seconds in total, on a single CPU.

Next we compare our method with OASIS on the much larger ILSVRC data and present results in Fig. 3(middle). We run multiple instances of OASIS with aggressiveness parameters $C = \{0.001, 0.01, \ldots, 1000\}$ and report the best after 14 days of training (our method takes 14 days in total on one CPU). We also include a subset of the baselines

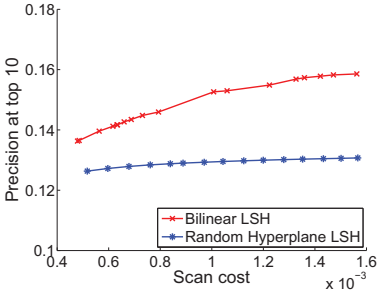

Figure 5. Precision at top 10 vs scanning cost for bilinear LSH and random hyperplane LSH as described in Sec. 5.5

from Fig. 3(left) excluding those optimized for hierarchy. OASIS seems slower to converge with the larger dataset and higher feature dimensionality (21K versus 1K for Caltech256) as it is still worse than SPM. Note that OASIS is inherently sequential while our method can be easily parallelized by training each semantic classifier independently.

5.4. Cross-category generalization

A potential advantage of using semantic attributes is the ability to generalize to new categories, as demonstrated in [23, 27] in a classification setting. We evaluate how our method can generalize to unseen categories in a retrieval setting. Only 900 of the 1000 ILSVRC semantic attributes are used to build the semantic representation and retrieval is *only* evaluated for categories for which no images are seen during training ("unseen in training" curve in Fig 3 right). While performance is lower than then when example images from those categories are seen in training ("seen in training" curve in Fig. 3 right) it is still better than the raw feature comparison baseline for much of the retrieval list. Note that, as in all other experiments, training and test image sets are disjoint.

5.5. Indexing efficiency

We evaluate the effectiveness of our bilinear LSH by measuring retrieval precision versus scanning cost. Scanning cost is the percentage of the data points needed to be scanned for top k retrieval, the dominating computation for retrieval. Brute force linear scan would result 1.0 scan cost

and maximum precision. One can adjust the number of hash tables and the number of hash function concatenations to trade off between precision and scanning cost. We compare our hashing technique with the widely used random hyperplane LSH [5] on retrieval. Random hyperplane LSH approximates the cosine similarity $\frac{x^T y}{\|x\|_2 \|y\|_2}$ by repeatedly selecting a random hyperplane and project vector x to one bit depending on which side x is on. To compare fairly with random hyperplane LSH, we set our prior similarity matrix S to identity and use flat precision to evaluate. We set the length of hash code for both methods to be the same(20 bits). In Fig. 5, we vary the number of hash tables to produce the precision versus scanning cost curves. Fig. 5 shows that bilinear LSH achieves a precision of 0.15 for top 10 images, very close to the precision (0.17) of linear scan, while examining only 0.1% of the database points. This is significantly better than random hyperplane LSH.

6. Conclusion

We have presented an approach that can exploit prior knowledge of a semantic hierarchy for similar image retrieval, and is scalable to very large retrieval problems. Experiments show that adding hierarchical knowledge significantly increases retrieval performance. In addition we show that a handicapped version of our system – without prior hierarchical information – can start with the same training information as the state of the art for similarity function learning (OASIS) and learn a more accurate similarity function with less total computation, and much less "wall clock time" due to significantly better inherent parallelism. We note that our technique should be seen to complement previous work on similarity learning as it is most useful when some explicit labels are available at training time (a common case). Our final contribution is a hashing scheme for bilinear similarity on probability distributions that is shown to provide efficient (sub-linear) retrieval in our setting, and may be useful in a wide range of applications. This completes an end-to-end system for very large scale hierarchical retrieval that has inherent parallelization, linear time training, sub-linear time retrieval, and better accuracy and scalability than the state of the art.

Acknowledgments. L. F-F is partially supported by an NSF CAREER grant (IIS-0845230), the DARPA CSSG grant, and a Google research award.

A. Proofs

We outline two of the proofs for non-overlapping categories(for probability vector x, $\sum_i x_i = 1$). We present the general case and more details in supplementary materials.

Definition A.1. A matrix $S \in \mathbb{R}^{K \times K}$ is *hashable*, if there exists a $\lambda_S > 0$ and, for any $\epsilon > 0$, a distribution on a family $\mathcal{H}(S, \epsilon)$ of hash functions $h(\cdot; S, \epsilon)$ such that for any probability vectors $x, y \in \mathbb{R}^K$

$$0 \leq \Pr\left(h_1(x; S, \epsilon) = h_2(y; S, \epsilon)\right) \; - \; \lambda_S \cdot x^T S y \leq \epsilon$$

where h_1 and h_2 are drawn independently from $\mathcal{H}(S, \epsilon)$.

Lemma A.2. *If S is symmetric, element-wise non-negative and diagonally dominant, that is, $\forall i = 1, \ldots, K, \; s_{ii} \geq \sum_{j \neq i} s_{ij}$, then S is hashable.*

Proof. Define a $K \times (K+1)$ matrix $\Theta = (\theta_{ij})$, where $\forall i \leq K, \; j \leq K, \; i \neq j, \; \theta_{ij} = \sqrt{\hat{s}_{ij}}$ and $\forall i \leq K, \; \theta_{ii} = \sqrt{\hat{s}_{ii} - \sum_{j \neq i} \hat{s}_{ij}}, \; \theta_{i,K+1} = 1 - \sum_{j=1}^{K} \theta_{ij}$ where $\hat{S} = \lambda_S \cdot S$ with λ_S chosen to ensure $\theta_{i,K+1} \geq 0$. Each row of Θ sums to one, and Θ without last column is symmetric.

Consider hash functions of the form $h : \Delta^{K-1} \rightarrow 2^{\mathbb{N}}$, where $2^{\mathbb{N}}$ is all subsets of natural numbers. Note that $h(x) = h(y)$ is defined as *set equality*, that is, the ordering of elements does not matter.

To construct $\mathcal{H}(S, \epsilon)$, let $N \geq 1/\epsilon$. Then $h(x; S, \epsilon)$ is computed as follows: (1) Sample $\alpha \in \{1, \ldots, K\} \sim multi(x)$; (2) Sample $\beta \in \{1, \ldots, K+1\} \sim multi(\theta_\alpha)$ where θ_α is the α^{th} row of Θ; (3) If $\beta \leq K$, return $\{\alpha, \beta\}$; (4) Randomly pick γ from $\{K+1, \ldots, K+N\}$, return $\{\gamma\}$.

Full details are provided in the supplementary materials, but it can be shown that

$$\Pr(h(x) = h(y)) = \lambda_S x^T S y + \frac{1}{N} \sum_{i,j} x_i y_j \theta_{i,K+1} \theta_{j,K+1}$$

where $0 \leq \frac{1}{N} \sum_{i,j} x_i y_j \theta_{i,K+1} \theta_{j,K+1} \leq \epsilon$. □

Lemma A.3. *If S is a matrix of all ones, then S is hashable.*

Lemma A.4. *If Q is a* zero padded extension *of S (i.e., Q is obtained by symmetrically inserting rows and columns of zeros into S) and S is hashable, then Q is hashable.*

Lemma A.5. *If S is hashable, then aS is hashable for any $a > 0$.*

Lemma A.6. *If $Q = \sum_{l=1}^{L} S_l$ and S_l is hashable for $l = 1, \ldots, L$, then Q is hashable.*

Lemma A.7. *Let $T = G(V, E)$ be a rooted tree and define $\pi_{m,n}$ to be the lowest common ancestor between node m and n for any $m, n \in V$. Let $V_r \subseteq V$ be subtree rooted at r (i.e., the set of all nodes descending from node $r \in V$ including r itself). Let $\Omega_r \subseteq V_r$ be all the leaf nodes of r and let $K_r = |\Omega_r|$. Let $f_r : \Omega_r \rightarrow \{1, \ldots, K_r\}$ be a one-to-one mapping of the leaf nodes of r to a set of integers. Let $\xi(\cdot) : V \rightarrow \mathbb{R}$ be any function defined on V. Let $S^{(r,\xi)} \in \mathbb{R}^{K_r \times K_r}$ be a similarity matrix induced by r and ξ, where $S_{ij}^{(r,\xi)} = \xi(\pi_{f_r^{-1}(i), f_r^{-1}(j)}), \forall i = 1, \ldots, K_r, j = 1, \ldots, K_r$.*

For any $r \in V$, if $\xi(\cdot)$ is non-negative and downward non-decreasing in the subtree of r, that is, $\xi(q) \geq 0$ for any $q \in V_r$ and $\xi(q) \geq \xi(p)$ for any $p, q \in V_r$ such that q is a child of p, then $S^{(r,\xi)}$ is hashable.

Proof. Let $r \in V$. We prove the claim by induction on the tree. If r is a leaf node, then $S^{(r,\xi)}$ is a scalar and thus hashable. Suppose r is an internal node. Let $\sigma(r)$ be its direct children. Our inductive hypothesis is that for any $c \in \sigma(r)$, the similarity matrix $S^{(c,\xi')}$ induced by c and any $\xi' : V_c \to \mathbb{R}$, which is non-negative and downward non-decreasing, is hashable.

Observe that the columns and rows of $S^{(r,\xi)}$ can be partitioned by the direct children of r. Also for any $c, d \in \sigma(r)$ and $c \neq d$, $S^{(r,\xi)}_{f_r(\Omega_c), f_r(\Omega_d)} = \xi(\pi_{\Omega_c,\Omega_d}) = \xi(r) \cdot \mathbf{1}$, where $\mathbf{1}$ is a matrix of all ones. Thus $S^{(r,\xi)} = \xi(r) \cdot \mathbf{1} + \sum_{c \in \sigma(r)} Q^{(c)}$ where $Q^{(c)}_{ij} = S^{(r,\xi)}_{ij} - \xi(r)$ if $i, j \in f_r(\Omega_c)$ and 0 otherwise

Let $\xi'(\cdot) = \xi(\cdot) - \xi(r)$. The lowest common ancestor of the leaf nodes of r cannot be higher than r and ξ is downward non-decreasing, hence $\xi'(d) \geq 0$ for any $d \in V_r$ and $\xi'(d)$ is downward non-decreasing. By the inductive hypothesis, given any $c \in \sigma(r)$, the similarity matrix $S^{(c,\xi')}$ induced by c and ξ' is hashable. We conclude the proof by showing that $Q^{(c)}$ is a zero padded extension of $S^{(c,\xi')}$. It follows from Lemmas A.3-A.6 that $S^{(r,\xi)}$ is hashable. □

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS06*, pages 459–468, 2006.

[2] Y. Aytar, M. Shah, and J. Luo. Utilizing semantic word similarity measures for video retrieval. In *CVPR*, 2008.

[3] I. Biederman. Recognition by components: A theory of human image understanding. *PsychR*, 94(2):115–147, 1987.

[4] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV10*, pages IV: 438–451, 2010.

[5] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC '02*, 2002.

[6] G. Chechik, U. Shalit, S. Bengio, S. Sonnenburg, V. Franc, E. Yom-tov, and M. Sebag. Large scale online learning of image similarity through ranking. *JMLR*, 2010.

[7] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV10*.

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR09*, 2009.

[9] W. Dong, M. Charikar, and K. Li. Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces. In *SIGIR '08*, 2008.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[12] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[13] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV10*.

[14] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, pages 1–8, 2007.

[15] A. Globerson and S. Roweis. Metric learning by collapsing classes. *NIPS*, 18:451–458, 2006.

[16] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, Caltech, 2007.

[17] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. *CVPR08*, 2008.

[18] A. G. Hauptmann, M. G. Christel, and R. Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602–622, April 2008.

[19] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *NIPS08*.

[20] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *PAMI*, 31:2143–2157, 2009.

[21] N. Kumar, P. N. Belhumeur, and S. K. Nayar. FaceTracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, pages 340–353, Oct 2008.

[22] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV09*.

[23] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[24] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR06*, 2006.

[25] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[26] F. Provost and P. Domingos. Well-trained pets: Improving probability estimation trees, 2000.

[27] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What Helps Where – And Why? Semantic Relatedness for Knowledge Transfer. In *CVPR*, 2010.

[28] R. Salakhutdinov and G. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009.

[29] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 30(11):1958–1970, November 2008.

[30] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR08*.

[31] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV10*, pages 776–789.

[32] http://www.image-net.org/challenges/LSVRC/2010/.

[33] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*. MIT Press, 2006.