Image and Vision Computing 31 (2013) 982–991

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Hierarchical classification of images by sparse approximation ☆

Byung-soo Kim [a],*, Jae Young Park [a], Anna C. Gilbert [b], Silvio Savarese [a]

[a] EECS Building, 1301 Beal Avenue Ann Arbor, MI 48109-2122, United States
[b] East Hall, 530 Church St. Ann Arbor, MI 48109-1043, United States

ARTICLE INFO

Article history:
Received 24 October 2012
Received in revised form 8 July 2013
Accepted 10 October 2013
Available online 21 October 2013

Keywords:
Sparse approximation
Sparse sensing
Sparsity
Image classification
Hierarchy
Structured data classification

ABSTRACT

Using image hierarchies for visual categorization has been shown to have a number of important benefits. Doing so enables a significant gain in efficiency (e.g., logarithmic with the number of categories [16,12]) or the construction of a more meaningful distance metric for image classification [17]. A critical question, however, still remains controversial: would structuring data in a hierarchical sense also help classification accuracy? In this paper we address this question and show that the hierarchical structure of a database can be indeed successfully used to enhance classification accuracy using a sparse approximation framework. We propose a new formulation for sparse approximation where the goal is to discover the sparsest path within the hierarchical data structure that best represents the query object. Extensive quantitative and qualitative experimental evaluation on a number of branches of the Imagenet database [7] as well as on the Caltech-256 [12] demonstrate our theoretical claims and show that our approach produces better hierarchical categorization results than competing techniques.

1. Introduction

Recent advances in computer vision and machine learning have enabled the design of recognition methods that are capable of classifying images into large number of visual categories (typically, hundreds) [11,8,6,14]. In one of the current paradigms for image categorization, image classes are organized in a flat structure and the problem is to discover the class (among all those in the flat structure) that best represents (in term of a distance function) the visual content of a given query image.

Recently, researchers have explored the idea of organizing visual data in a hierarchical structure rather than in a flat one. This paradigm addresses some of the limitations of the flat structure: i) it allows for a significant gain in efficiency, typically logarithmic with the number of categories, as addressed by Marszalek and Schmid [16] and Griffin and Perona [12]; ii) it enables the construction of a more meaningful distance metric for image classification; and iii) it echoes the way how humans organize data, as addressed by Palmer [17]. However, a critical question still remains controversial: would structuring data in hierarchical sense also help classification accuracy? Up to date there is no definite answer to that question. For instance, top-down classification schemes (applied on hierarchical structures) proposed by Marszalek and Schmid [16] and Griffin and Perona [12] have produced inconclusive evidence as for whether hierarchy has a beneficial effect on classification accuracy. Classification methods based on Hierarchical Support Vector Machines can be used to trade off accuracy against speed as in Griffin and Perona [12] or employed to increase classification accuracy as originally proposed by Tsochantaridis et al. [21] and utilized for image classification, as suggested by Binder et al. [2]. Although [2] has shown promising results, it is computationally very demanding as the number of categories becomes larger than 30 ~ 50. Finally, methods based on combining models from different levels of the hierarchy proposed by Zweig and Weinshall [23] have also shown positive results but are yet to be validated on deeper and larger hierarchical structures.

In this paper we attempt to address the issues discussed above and show that the hierarchical structure of a database can be successfully used to enhance classification accuracy using a sparse approximation framework. The key idea is to introduce a distance function that takes into account the hierarchical structure of the visual categories (Fig. 1) and to identify two images to be similar if they share a similar path in the hierarchy. We show that this distance function (or similarity metric) is equivalent to the Hamming Distance (HD) for vectors that encode the hierarchy. This allows us to cast the categorization problem as the one of discovering the category in the tree structure that has the smallest HD from the query category label. We solve this problem via sparse approximation and introduce a new formulation of the sparse approximation problem which we call hierarchical sparse approximation. In the typical sparse approximation problems [22,5,20], a query image can be identified as the sparsest representation over the set of training images, as proposed by Wright et al. [22] or basis functions, as proposed by Mairal et al. [15] for all object classes; that is, the sparsest solution is one (or a combination of a few) image out of all possible images in the

☆ This paper has been recommended for acceptance by Y. Aloimonos.

* Corresponding author at: EECS Building 4338, 1301 Beal Ave., Ann Arbor, 48109-2122, United States.

E-mail addresses: bsookim@umich.edu (B. Kim), jaeypark@umich.edu (J.Y. Park), annacg@umich.edu (A.C. Gilbert), silvio@eecs.umich.edu (S. Savarese).

URL: http://www.umich.edu/~bsookim (B. Kim).

0262-8856/$ – see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.imavis.2013.10.005

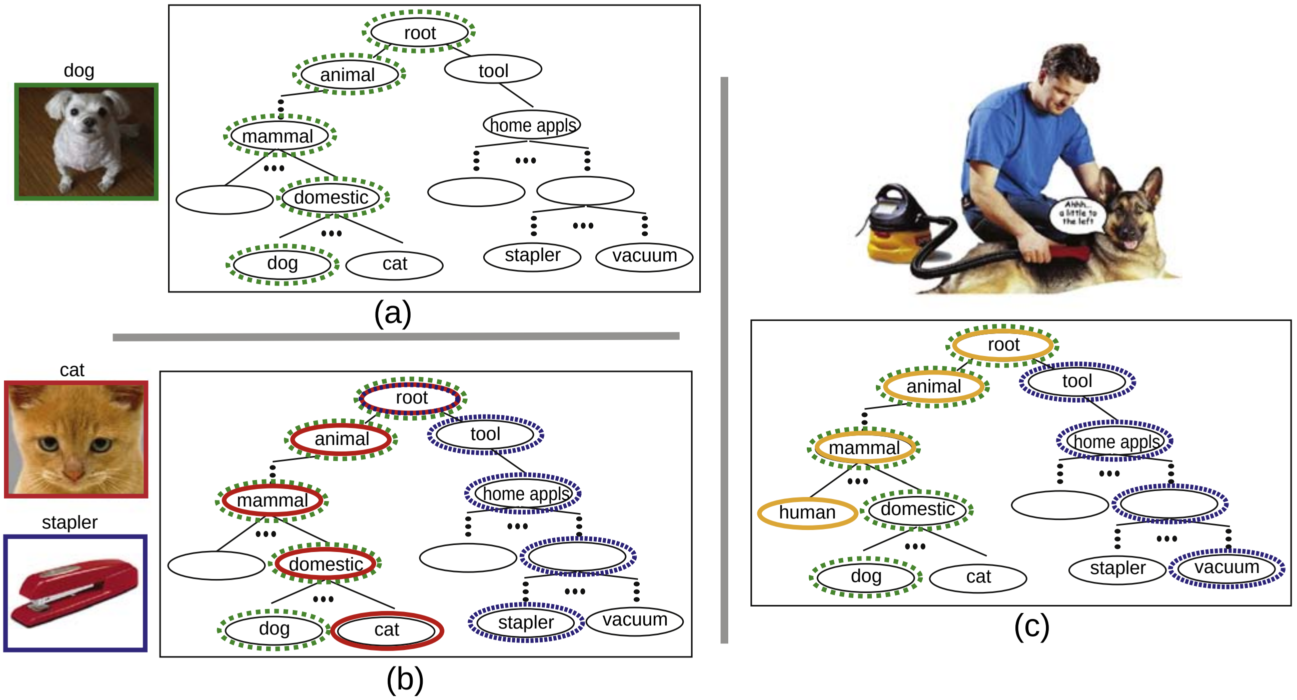


Fig. 1. (a) Organizing images in a hierarchical structure (tree) enables more descriptive methods for characterizing images: the image of a dog can be described by class labels associated to each node of the (green) path in the tree. (b) Misclassifying a dog with a cat is not as bad as misclassifying a dog with a stapler. If data are organized in a tree, it is possible to relate object classification errors with objects with locations in the tree. For instance dog, cat and stapler categories are associated with the green, red and blue paths (respectively) in the tree. The error in misclassifying a dog with a cat can be measured as the Hamming Distance (HD) between the corresponding paths. HD captures the similarity between two paths in the tree (see Section 3 for details). The HD is 1 in this case. Note that misclassifying a dog with a stapler leads to a larger HD (that is, 5). (c) It is desirable to classify multiple objects at the same time. If an image contains a dog, a human and a vacuum, our algorithm can discover three paths (green, orange and blue respectively) in the tree, one for each query object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

dataset. We call this the flat sparse approximation problem. The key novelty of our approach relies on the idea that the sparse representation is not constructed over a flat structure of object classes (as in the classic sparse sensing problem) but rather by enforcing that the solution must be one (or a combination of a few) path out of all possible paths on a given hierarchy of object classes (training set). Moreover, classification accuracy is measured in hierarchical sense (that is, by considering the HD between the query path and the ground truth one). Since our method relies on the sparsity of the representation, our approach is suitable for large scale classification problems; i.e., the conditions underlying the sparsity assumptions are best verified when the dataset is large and distribution of visual categories is diversified. In this work we present sufficient conditions under which our hierarchical sparse formulation can be used with success and small error bounds are guaranteed. Furthermore, a crucial property of our classification framework is that it is capable of classifying multiple object categories at the same time if more than one (dominant) object appears in the query image (Fig. 1 (c)).

We have carried out extensive quantitative and qualitative experimental evaluation on a number of branches of the Imagenet database [7] as well as Caltech-256 [11]. Each branch comprises hundreds of visual categories organized in the hierarchical structure. All the experiments demonstrate that our hierarchical approximation framework yields much better hierarchical classification accuracy over flat sparse approximation. Evaluation was carried out by comparing average precision measured in terms of HD as well as by measuring the actual classification accuracy at each level of the hierarchy. Our method achieves a performance increase ranging from 5% to 10% for the most critical levels of the hierarchy. Additional experiments on multi-category classification also show very promising results.

The rest of this paper is organized as follows. In Section 2, we will briefly review how sparse approximation can be applied to image classification problem. The formal definition of hierarchical classification and our proposed embedding is provided in Section 3. A number of experiments are carried out to validate our scheme in Section 4. Finally, we summarize our work in Section 5.

2. Image classification using sparse approximation

In this section, we describe our image representation and introduce the basic formulation of the flat image classification problem based on sparse approximation. We assume a database of images is available. Furthermore, we assume that such a database comprises a large number of categories and each category has a large number of image instances. We assume that each image has a dominant object instance with some level of background clutter as in Caltech-256 [11] or the ImageNet [7]. In classification, we assume that the query image (with unknown category label) contains one (or multiple) dominant object(s) whose category label is represented by the dataset. Of course, the query object instance itself is not necessarily included in the dataset. The classification problem can be solved by seeking, among all the images (object instances) in the database, the one that is closest to the query object(s). The category such image belongs to is the classification result. If the query image contains multiple dominant objects, the classifier must return multiple category labels associated to all of the dominant objects in the query image.

2.1. Object representation and distance function

Assessing whether an image is “close” to another one relies on the construction of a distance function which depends on the way how the visual content of an image is represented. Following a common representation used in computer vision, we describe an image using a normalized histogram of codewords (i.e., the bag of words representation, also named BOW) [6] or, equivalently, a histogram capturing a spatial pyramid of codewords [14,10]. In either cases, we denote such

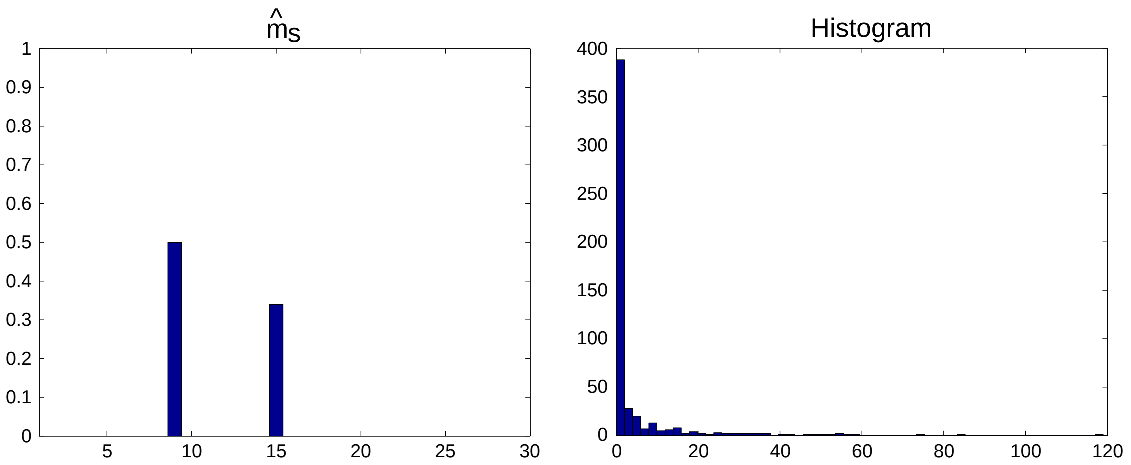


Fig. 2. (left) An example of a reconstruction of $\hat{m}_s$. In this example, $\hat{m}_s$ has only two non-zero coefficients and $\|x - H_s\hat{m}_s\|_2 = 0.58$. (right) Histogram of the number of categories that provided more sparse and accurate representations than the true category for 512 trials.

histogram by a vector x. Codewords are drawn from a learnt dictionary of vector quantized features as described in [6,14,10]. The size of the dictionary is denoted by K. Thus x is a column vector of size K, if we use a simple histogram of codewords to represent the image. Notice that other types of representations are also possible. The similarity between two images represented by x_i and x_j can be measured by computing the l_n norm distance between x_i and x_j, where n can be 0, 1, etc. Similar images will have small distances.

2.2. Model matrix

Let us stack all the histograms of images in the database as columns of the matrix H. Thus, H will be $K \times N$, where N is the number of images in the dataset. We call this matrix H the flat model matrix. Under the assumption that the database is sufficiently large, any query image can be represented as a superposition of one or more images in the training data with small error e such that $x = Hm + e$. Note that $N \times 1$ vector m is called the mixing vector and consists of a few non-zero entries associated to the images in the database that contribute to represent the query image by superposition. Note that the error e captures background clutter and the intra-class variability. A similar representation was introduced in [22] and was shown to be suitable for face recognition problems.

We argue that is also a reasonable model for the generic object classification problem. As long as the training set is large enough the image representation will yield satisfactorily discriminative features for classifying object classes as demonstrated in [11,14]. In order to further justify the model, we show empirical evidence that mixing vectors m are both fairly sparse and concentrated using a number of datasets in the following Section 2.2.1.

2.2.1. Empirical evidence for sparse approximation

In this section, we provide empirical evidence of the assumption that a query image x can be both sparsely and accurately represented by a few linear combinations of BOW descriptors of the same category. The following experimental evaluation is carried out by using the hierarchical Caltech-256 dataset with 'dog' category. See Section 4 for more details about the structure of this dataset. Let us denote the $K \times N$ matrix H_s as the matrix that is formed by taking the columns in H that correspond to the same category as x. Thus, N is the number of images in a category. Note that, $K = 4200$ and $N = 30$ for this particular dataset and also that $K > N$. Then, we empirically show that $x = H_s m_s + e$ has a solution $\hat{m}_s$ that is sparse and gives a small approximation error $\|x - H\hat{m}_s\|_2$.

To compute $\hat{m}_s$ for a given x we solve,

$$\min_{m_s} \|x - H_s m_s\|_2 + \lambda\|m_s\|_1,$$

which is also known as the least absolute shrinkage and selection operator (LASSO) [19]. The first term of the cost function ensures that the approximation error is small and the second term ensures that the solution is sparse. Fig. 2 shows an example of a plot of $\hat{m}_s$ obtained by solving the above minimization problem. We can see that $\hat{m}_s$ is indeed sparse with only two non-zero coefficients and has a small approximation error of 0.58.

In order to demonstrate that such behavior is common across most queries x, we repeat the above for 512 queries x that belong to different

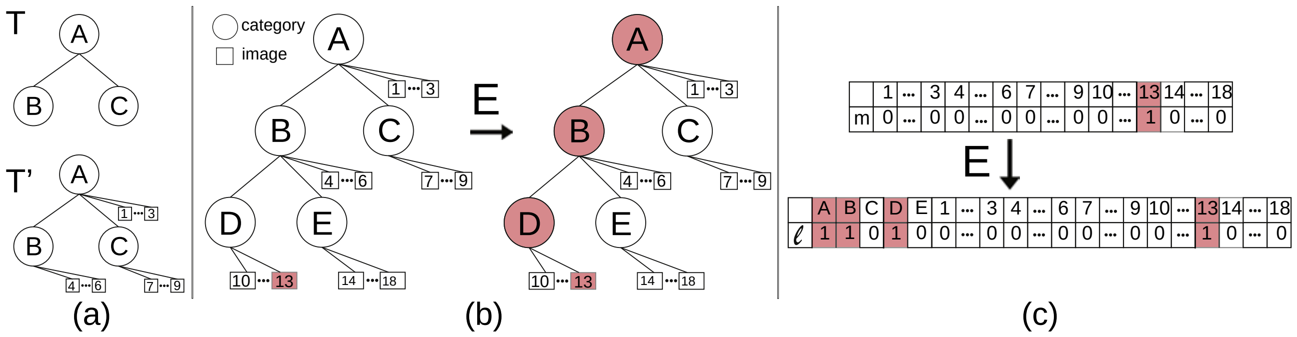


Fig. 3. Visualization of the embedding. (a) Examples of T and T′. (b,c) As a result of the embedding E, the flat mixing matrix m is mapped into ℓ. In this example, when m shows a nonzero entry corresponding to image 13, the embedded ℓ shows non-zero entries corresponding to image 13 as well as to its ancestors categories (nodes) A, B and D. These are on the path to the root from image 13.

categories and evaluate how sparse and accurate the solutions are by computing $\frac{\|\hat{m}_s\|_1}{\|\hat{m}_s\|_2}$ and $\|x - H_s\hat{m}_s\|_2$, respectively. We note that $1 \leq \frac{\|m_s\|_1}{\|m_s\|_2} \leq \sqrt{30}$ and the closer this fraction is to 1 the sparser the $\hat{m}_s$ and vice versa. The average value of $\frac{\|\hat{m}_s\|_1}{\|\hat{m}_s\|_2}$ and $\|x - H_s\hat{m}_s\|_2$ for a large number of trials are 2.41 and 0.52, respectively, which show that x can indeed be sparsely and accurately represented by the columns of the same category.

Next we show that for a large number of queries, x is best represented by the columns of the same category than by those of other categories. In the Caltech-256 dataset there are in total 256 categories. For each query x, we solve the above minimization problem for all 256 categories, where each category has a different H_s that is constructed by taking the appropriate columns in H. Then for all 256 solutions, we evaluate $\frac{\|\hat{m}_s\|_1}{\|\hat{m}_s\|_2}$ and $\|x - H_s\hat{m}_s\|_2$ as a measure of sparsity and accuracy. To assess whether or not x is better represented by the columns of the true category, we compute how many other categories resulted in a representation $\hat{m}_s$ that gave 10% better performance in terms of the two measures simultaneously. We repeat this procedure for 512 different query images that belong to different categories and plot in Fig. 2 the histogram of the number of categories that resulted in a better representation than the true category. Out of 512 trials for exactly327 query images, the true category was able to better represent x than others. This and the fact that this histogram exhibits a high concentration close to zero shows that for most queries, the true category provides more sparse and accurate representations than other categories.

2.3. Classification

Clearly m contains the information that allows us to estimate the class label of the query image. Therefore, the classification problem (what is the object class?) is recast into the problem of estimating the vector m (where is a non-zero entry?). Furthermore, this formulation allows us to discover multiple dominant object categories in the image. Suppose the image contains three objects as in Fig. 1 (c), then the query image may be expressed as a superposition of $s = 3$ training histograms and the non-zero entries of m will return the 3 classes appearing in x (i.e., dog, human and vacuum). Solving m is challenging because the system is under-determined ($N \gg K$) and has an infinite number of solutions. Because we postulate or seek an s-sparse mixing vector m, we can formulate this problem as a sparse approximation problem and seek to find the sparsest solution that best approximates (in ℓ_0 error) the observed instance. Notice that the pseudo-norm $|| \cdot ||_0$ counts the number of non-zero entries in a vector.

Problem 0. $min||m||_0$ subject to $||Hm - x||_2 \leq \epsilon$.

Unfortunately, the above problem is an NP-hard problem in general (given an arbitrary matrix H and an arbitrary vector x). We can, however, solve this problem in polynomial time with appropriate geometric assumptions on H; if the maximum entry of the matrix $|H^*H\text{-}I|$, or the coherence[1] $\mu(H)$, of the matrix is small, then there are several algorithmic solutions. Let us assume for now that the training set contains the query image x. As proposed by [4,22], one method is to observe that Problem 0 is an optimization problem with a non-convex objective function and that a convex relaxation of this problem yields a problem which can be solved efficiently with standard optimization techniques [5],

Problem 1. $\min ||m||_1$ subject to $||Hm - x||_2 \leq \epsilon$.

A second algorithmic approach is to use a greedy algorithm, one that identifies image instances iteratively, such as Orthogonal Matching Pursuit (OMP). See [20] and the references therein for details on this algorithm. In Section 4.3 we show that the coherence between individual images decreases as a function of their hierarchical distance; thus, while the overall coherence $\mu(H)$ is high, with high probability, the coherence between any two images is quite small and OMP can distinguish among these images and choose a representation close to the ground truth.

3. Hierarchical classification with sparse approximation

While the model $x = Hm + e$ is reasonable and empirical evidence suggests that it is fairly accurate, it fails to take into account any hierarchical information amongst the classes. Furthermore, the error metrics for typical sparse approximation algorithms [20,18] do not take into account structural relationships amongst the columns of H. Indeed, a small error in the mixing vector $\|\hat{m}_s - m\|_2$ or in the reconstruction of the observation x does not necessarily guarantee hierarchical similarity between $\hat{m}$ and m. For instance, suppose the ground truth label of a query image is "dog". Assume two possible classification results are generated: "stapler" and"cat". These two results would be associated to the same flat classification error $\|\hat{m} - m\|_2$ if the model in $x = Hm + e$ were employed, whereas the classification error associated to "cat" would be smaller than that associated to "stapler" if the error function was defined in hierarchical sense (Fig. 1).

In this section, we assume that object categories are structured in a (rooted, labeled, recursive) tree T that reflects the semantic (parental) relationships among object categories. Note that each node of T contains all of the images representative of the visual category label associated to that node. A schematic illustration of such data structure is given in Figs. 1 and 3. We define T′, the data structure induced by the semantic tree, that contains two types of nodes, category labels and individual column vectors of H (images) (Fig. 3). It encodes the semantic relationship amongst the categories and the assignment of columns of H to those categories, but, unlike the tree T, both categories and individual columns of H make up the nodes. A key contribution of our work is to introduce a suitable encoding matrix E that embeds the flat model matrix H into a hierarchical (tree) model matrix Φ and to show that the resulting hierarchical sparse approximation is solvable and appropriate for classification.

3.1. Hierarchical embedding

The encoding matrix E is constructed so as to map the mixing vector m into an embedded mixing vector $\ell = Em$, whose non-zero entries correspond to the paths in T′ from the image to the root of the tree (Fig. 3). More concretely, for C object categories along with N images, we define E to be the $(N + C) \times N$ matrix that embeds a column of H and its path to the root in the tree T′. Without loss of generality, we can permute the rows of E so that E has the following structure $E = [I\ L^T]^T$ where I is the $N \times N$ identity matrix and the $C \times N$ matrix L consists of the hierarchical labels of each image. Each row of L corresponds to a category and each column to a training image; $L_{i,j} = 1$ if category i is on the path to the root from training image j. Each row encodes which training images are descendants of category j. Note that the length of ℓ is $N + C$. If we denote $E^\dagger$ the pseudo-inverse of E, then we define $\Phi = HE^\dagger$.

3.2. Hierarchical sparse approximation

The hierarchical embedding allows to reformulate Problem 1 as a hierarchical sparse approximation problem and find a solution for ℓ given x:

Problem 2. $min\|\ell\|_1$ subject to $\|\Phi\ell - x\|_2 \leq \epsilon$

Unlike the original sparse approximation problem, in this problem, the sparsity pattern of the vector ℓ is constrained to lie on a single path (or subtree) of the tree T′. While the embedding $Em = \ell$ increases

[1] An equivalent definition of $\mu(H)$ is the maximum dot-product of different columns of H, $\mu(H) = max_{i \neq j}| < H_i, H_j > |$.

the number of non-zeros in ℓ (as compared to that of m), it also enforces a model that these non-zero entries must follow; they must lie on paths from individual columns of H to the root of the tree T′. Because the sparsity of ℓ follows a model and Φ has more columns than rows, this problem has the structure of a model-based compressive sensing problem [1].

Problem 2 can be solved efficiently by a greedy algorithm called TREE-OMP [13], which is a special case of the more general algorithm MODEL-COSAMP [1], assuming that Φ satisfies a geometric condition, referred to as model-Restricted Isometry Property (model RIP). (See Algorithm 1) TREE-OMP is similar to the OMP algorithm with the additional step that for all non-zero components in the vector ℓ, the algorithm enforces that all the components that correspond to ancestors in the tree are non-zero. This constraint guarantees that the estimated solution $\hat{\ell}$ corresponds to one (or more) physical path(s) in the tree.

Algorithm 1: TREE-OMP [13]

```
**Input**: $\Phi$, $x$, $\epsilon$, $\theta$
**Output**: $\widehat{\ell}$
Initialize the the counter $k = 0$, the vector $\widehat{\ell_k} = 0$, the residual $r_k = x$, and
the index set $\Lambda_k = \emptyset$.;
**while** *!done* **do**
    $z = \Phi^* r_k$;
    $S_k = \{\lambda \big| |z_\lambda| > \theta\}$;
    // Form a candidate set of columns of $\Phi$ which have inner
       products with the residual larger than the threshold
       $\theta$.
    $i_k = \arg\min_{i \in S_k} \|x - P_{span\{\phi_l : l \in F_i\}}\|_2$;
    // Search among the candidate set $S_k$ for the item that
       together with its family $F_i$ (i.e., all ancestor items)
       maximizes the reduction of residual.
    Set $\Lambda_{k+1} = \Lambda_k \cup F_{i_k}$.;
    $\ell_{k+1} = P_{span\{a_l : l \in \Lambda_{k+1}\}} x$;
    Update the residual, $r_{k+1} = x - \ell_{k+1}$.;
    **if** $\|r_{k+1}\|_2^2 < \epsilon$ **then** done;
    **else** $k = k + 1$.;
**end**
```

3.3. Theoretical analysis

In this subsection, we show that the hierarchical embedding in Section 3 produces a matrix Φ that, on average, satisfies the model RIP. We also show that $\hat{\ell}$, the output of TREE-OMP, is close to the ground truth embedded vector $\ell = Em$ not only in l_2 error, but, more importantly, in HD. These results are summarized in the following theorem. Moreover these results enable the construction of a classification algorithm that we call SPARSE PATH SELECTION (SPS) (see Algorithm 2).

Algorithm 2: Sparse Path Selection (SPS)

```
**TRAINING**
**Input**: Images with known hierarchy
**Output**: Encoded model matrix $\Phi$, threshold value $\eta$
Form the matrix $H$ of training vectors collected from all images in the
dataset;
Encode: $\ell = Em$, $\Phi = HF$, where $F = E^\dagger$;
Normalize the columns of $\Phi$ to have unit $l^2$-norm.;
Learn the threshold value $\eta$;

**TESTING**
**Input**: Encoded model matrix $\Phi$, query image, threshold value $\eta$
**Output**: Class labels
Form a vector $x$ from the query image.;
Estimate mixing vector $\widehat{\ell}$ =TREE-OMP($\Phi, x, \epsilon, \theta$);
Truncate small elements in $\widehat{\ell}$ by learned threshold values $\eta$ and return the
labels of the remaining non-zero entries; i.e., the classification results.;
```

Theorem 1. *Given a normalized test image x ($\|x\|_2 = 1$) which is sd-sparse with background "noise" n, we can solve $\Phi\ell = x + n$ for the embedded mixing vector ℓ with* TREE-OMP. *After $T > \log(sd)$ iterations, the output vector $\hat{\ell}$ has at most Td non-zero entries and satisfies*

$$\left\|\ell - \hat{\ell}\right\|_2 \leq 2^{-T} + C\|n\|_2.$$

In addition, if the noise $\|n\|_2 \leq \sqrt{Td}\left(\eta - 2^{-T}\right)$ is small enough compared to a learnt threshold η (See SPS algorithm), then $HD\left(\hat{\ell}, \ell\right) = 0$; i.e., we correctly identify all the categories on the ground-truth hierarchical path.

Proof. First, we note that the embedded vector $\ell = Em$ follows a model-sparse pattern as defined in [1].

Lemma 1. *If m is a s-sparse vector, then $\ell = Em$ has a sparse tree structure; that is, it encodes a rooted tree with s leaves.*

Proof. From [1], a signal model M_k is the union of m_{k} canonical k-dimensional subspaces $M_k = \cup_{m=1}^{m_k} \chi_m$ where each k-dimensional subspace $\chi_m = \left\{y | y|_{\Omega_m^c} = 0\right\}$ contains all signals y with support in Ω_m. The model M_k is defined by the set of possible k-sparse supports $\Omega_1, \ldots, \Omega_{m_k}$ and, if we restrict ourselves to those sets that are defined by a rooted tree structure, we have a model-sparse signal. Our embedding, by construction, yields such a vector ℓ; it is model $k \leq sd$ sparse (where d is the depth of the tree T′).

Lemma 2. *The matrix Φ is well-approximated by an iid (sub-)Gaussian random matrix.*

Proof. We model[2] the label matrix L as an iid random Bernoulli matrix; each entry $L_{i,j} = 1$ with probability p and 0 with probability *1-p*. Let

$$\widetilde{E} = \frac{1}{2}\left[I \quad \widetilde{L}^T\right]^T$$

where $\widetilde{L}_{i,j} = \frac{1}{Cp(1-p)}\left(L_{j,i} - p\right)$ is a centered version of the transpose of L. Observe that, on average, $\widetilde{E} = E^\dagger$, as

$$\mathbb{E}\left(\left(\widetilde{L}L\right)_{j,l}\right) = \mathbb{E}\left(\sum_{k=1}^{C} \widetilde{L}_{j,k} L_{k,l}\right) = \sum_{k=1}^{C} \frac{1}{Cp(1-p)} \mathbb{E}\left(L_{k,j} - p\right) \mathbb{E}\left(L_{k,l}\right) = 0$$

and

$$\mathbb{E}\left(\left(\widetilde{L}L\right)_{j,j}\right) = \mathbb{E}\left(\frac{1}{Cp(1-p)} \sum_{k=1}^{C} \left(L_{k,j} - p\right)\left(L_{k,j}\right)\right) = \frac{1}{Cp(1-p)} \sum_{k=1}^{C} p(1-p) = 1.$$

Then, on average,

$$\Phi = H\widetilde{E} = \frac{1}{2}\left[H \quad H\widetilde{L}^T\right]^T$$

and the entries in the columns corresponding to the $H\widetilde{L}$ block are

$$\left(H\widetilde{L}\right)_{j,l} = \sum_{k=1}^{N} H_{j,k} \widetilde{L}_{k,l} = \sum_{k=1}^{N} H_{j,k} \frac{1}{Cp(1-p)} \left(L_{k,l} - p\right)$$

approximately iid Gaussian random variables as they are (large) sums of bounded random variables with mean 0.

This analysis describes the average behavior of Φ only. Any instance of $E^\dagger$ has non-zero entries in the off-diagonal terms. These entries are

[2] In practice, the assignment of labels to training images is deterministic and we have more descendant images for a category the higher in the tree it is. The indexing of the columns is, however, arbitrary so we can order them at random initially and fixed throughout the remainder of the algorithm. A more realistic model is to change the probability p as a function of the depth of the category in the tree. The root has $p = 1$ and a deep category has p close to 0.

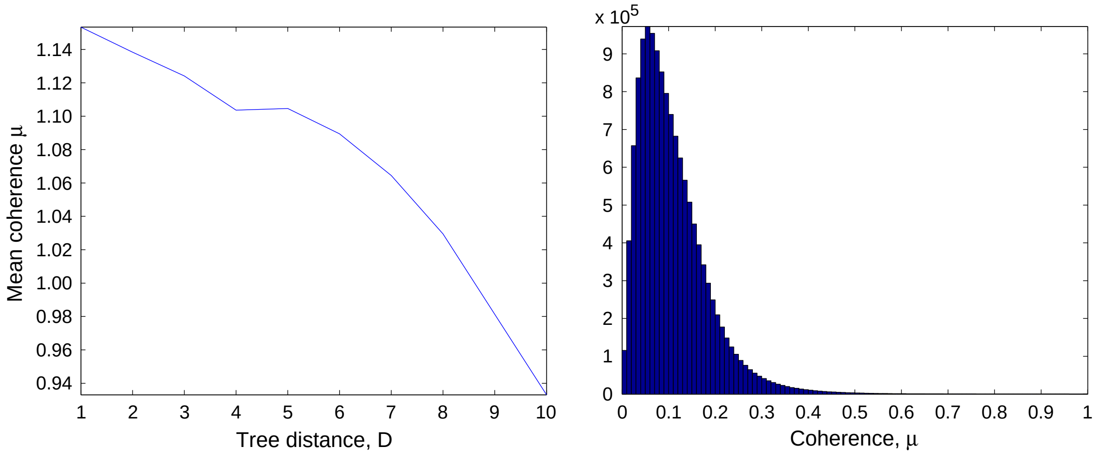


Fig. 4. (left) Mean coherence for objects chosen uniformly at random as a function of their distance in the tree; (right) Distribution of coherence values for objects with distance 6 in the tree.

also bounded random variables, and, hence, the product $\Phi = HE^{\dagger}$ consists of entries that are approximately Gaussian random variables.

From Lemma 1 and 2, we can conclude that Φ satisfies a model-RIP property [3]. Furthermore, we can use the result in [1] to conclude that after T iterations of TREE-OMP, the output $\hat{\ell}$ contains at most Td non-zero entries and satisfies $\left\|\ell-\hat{\ell}\right\|_2 \leq 2^{-T}+C\|n\|_2$. While the l_2 distance between two vectors is meaningful, it does not tell us how close the path(s) corresponding to the vector $\hat{\ell}$ are compared to the ground-truth vector ℓ, it conflates the paths with the coefficients on those paths. The error bound tells us what the average error in $\hat{\ell}$ is and, as long as it is below our learned threshold, $\frac{1}{\sqrt{Td}}\left(2^{-T}+\|n\|_2\right)<\eta$, we will not introduce spurious nodes in the path nor miss them and hence, $HD\left(\hat{\ell},\ell\right)=0$.

3.4. Sparse path selection algorithm (SPS)

After solving Problem 2, we obtain an estimate of the path ℓ in the hierarchical database associated to the query image. However, ℓ cannot be used directly for image classification. Ideally, the sparsest solution of Problem 2 should return a vector of “1” and “0” where the non-zero elements in ℓ allow to estimate the category labels of the query object as well as its parents. Unfortunately, this is not always the case and values between “0” and “1” can be also found because of the estimation noise. To solve this issue, we perform a post processing step. The idea is to introduce a threshold η and interpret it as a positive response any value that is above η (and as negative response, otherwise). Finding this threshold, however, is not trivial as it may vary with different datasets. Thus, in our experiments, we propose to automatically learn these thresholds using a binary MAP estimator trained using a validation set. Such evaluation set is then removed from the dataset so as to avoid contamination during testing. Our entire classification scheme is summarized in the Algorithm 2. We call this algorithm SPS.

3.5. Classifying multiple categories

As discussed in the previous sections, if the input vector x describes an image comprised of s categories, the mixing vector m is an s-sparse vector and the corresponding embedded mixing vector ℓ defines a subtree composed of s paths. Each of these paths is associated to one of the categories in x. (Fig. 1) Thus, solving Problem 2 and obtaining an estimate $\hat{m}$ of m allows us to simultaneously discover the presence of multiple categories in the image. Even if this appears to be an appealing property, one critical question must be addressed. How many categories s can we simultaneously handle until the conditions (i.e. sparsity, etc.) underlying the solution of Problem 2 are violated? The bounds in [1] suggest that we need at least $O(sd)$ rows in the histograms, where d is depth of hierarchical tree and Section 4.7 gives some empirical evidence that multiple category classification is possible with these algorithms.

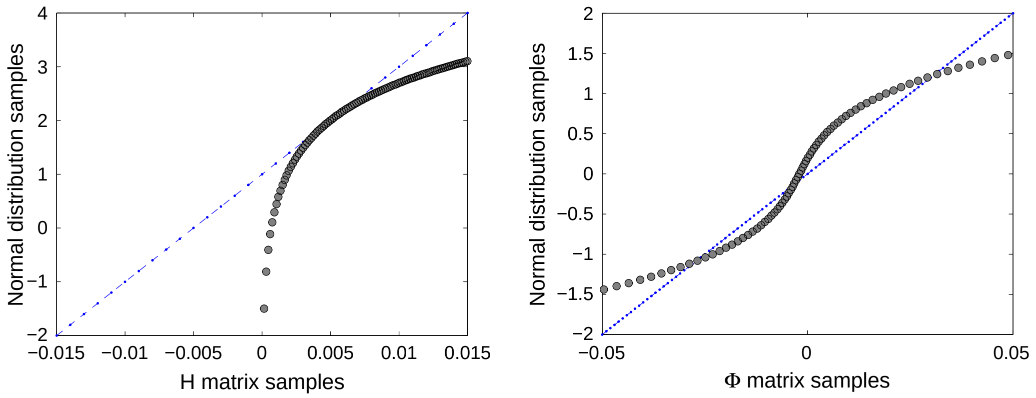


Fig. 5. The QQ-plot can be used to verify that the matrix Φ obtained from embedding the Caltech-256 dataset is consistent with our theoretical observation that Φ is well-approximated by an iid random Gaussian matrix. The plot on the left is the QQ-plot for the original matrix H; the plot on the right is the QQ-plot for the matrix Φ. Observe that Φ is more consistent with a Gaussian random matrix than H is, although somewhat skewed compared to the normal.

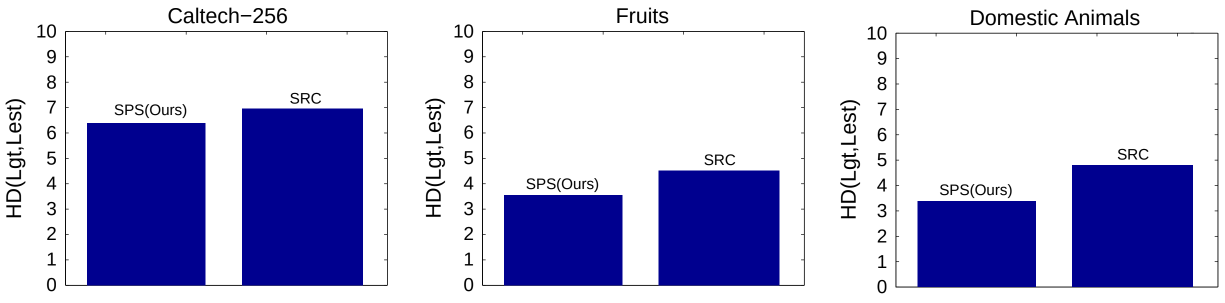


Fig. 6. Average Hamming Distance (HD) for different subcategories is drawn.

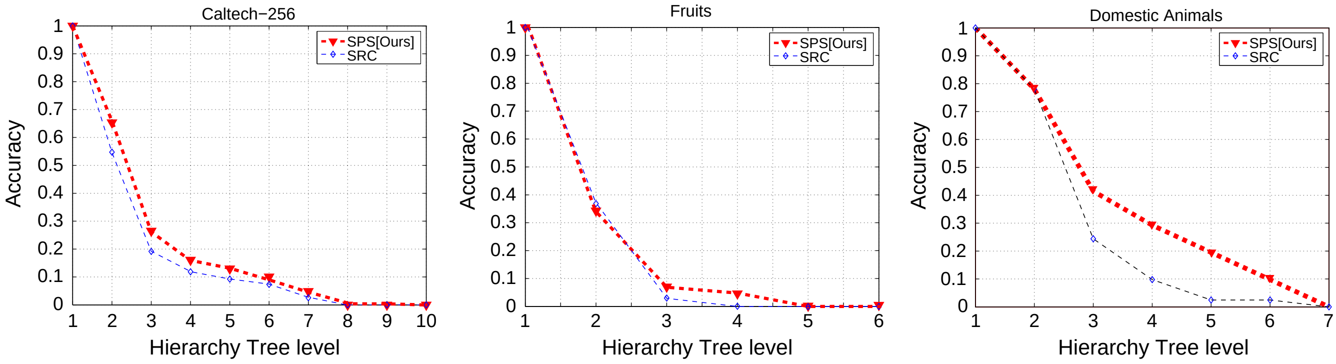


Fig. 7. Average accuracy of classification for different hierarchical levels. We tested on five different categories, Caltech-256, Fruits, Domestic animals and Domestic animals (leaves only). Average accuracy captures the average number of correctly estimated nodes (categories) for each level (x-axis) for all testing images. A node *j* is estimated correctly if the ground truth path evaluated at *j* is equal to the estimated path at *j* for a given test image. Consider the example in Fig. 3 (b). In this case, the accuracy is calculated over 3 levels. Suppose the ground truth query image is 13 and the ground truth *m* is associated to class labels *A*, *B*, *D*. If the estimated ⁄ returns class labels *A*, *B*, *E*, the accuracy for three levels is 1, 1, 0. If the estimated *m* returns the class labels *A*, *C*, the accuracy for three levels is 1, 0, 0. If the estimated *m* just returns the class labels *A*, the accuracy for three levels is still 1, 0, 0.

4. Experiments

In this section, we present quantitative and qualitative experimental results to validate our theoretical claims. We test our algorithm using different hierarchical databases. These are: i) 3 branches of the ImageNet [7] each comprising hundreds of categories; ii) The hierarchical Caltech-256 dataset [12]. We use different metrics to evaluate the performances of our algorithm: i) Overall average Hamming Distance (HD); ii) Average classification accuracy for each levels of the hierarchy. We benchmark our results using competitive classification methods SRC, the sparse approximation technique introduced by [22]. Our experiments include classification of a single dominant object category as well as multiple categories. In each of the single category classification experiments we used 16 patches on a grid with step 8 pixels to generate SIFT descriptors. BOW histograms are constructed using 500 codewords generated from K-means clustering. Finally, we used SPH (Spatial Pyramid Histogram) up to the resolution level 4 to represent each image. In each experiment we sample (at most) 100 images for each node of the working database and use these for learning (i.e. to build the *H* matrix). Forexample, for the domestic Animal sub-tree of ImageNet, we collected about 21,000 images for training. We sample an additional 10 images per node for testing. This way, testing images are guaranteed to be different from those in the training set.

4.1. ImageNet subsets

ImageNet [7] is a hierarchical image database with 10 million images and over 10,000 categories. It organizes different classes of images according to the WordNet [9] structure, and “IS-A” relationship exists between parents and children. Images are collected for leaf nodes as well as internal nodes. In consequence, a test image can be chosen either from an internal node or from a leaf node. In the experiments, we used 2 different branches from the ImageNet: Domestic Animals and Fruits. These subsets are chosen so as to observe the effect of different dataset sizes (48, 21, 320 respectively) and structures (domestic animal is a deeper tree than fruits) on the classification results. The hierarchical structure of both subsets extensively diverge; for example, for ‘domestic animals’ subset, it splits from 1 to 5 in the first level, and then splits into 18 in the second level.

4.2. Hierarchical Caltech-256

The Caltech-256 is rearranged in a hierarchy according to best matches in the WordNet. In this Hierarchical Caltech-256, all images are associated to a leaf node, hence there are no images in the internal nodes.

4.3. Dataset coherence properties

In this subsection, we analyze the coherence properties of different subsets of the datasets. We do not necessarily use all instances of every category but instead we pick two instances uniformly at random to obtain a statistical perspective on the coherence values of the derived *H* matrices.

Experimental results show that if we use all object instances, the matrix *H* is quite coherent and that the value of $\mu(H)$ is close to 1. A straightforward application of the previous theoretical results would suggest that neither the greedy algorithm nor the convex relaxation is appropriate for identifying a single instance of an object category. Notice that the case of multiple categories would be even more problematic. These theoretical results are, however, too pessimistic and do not

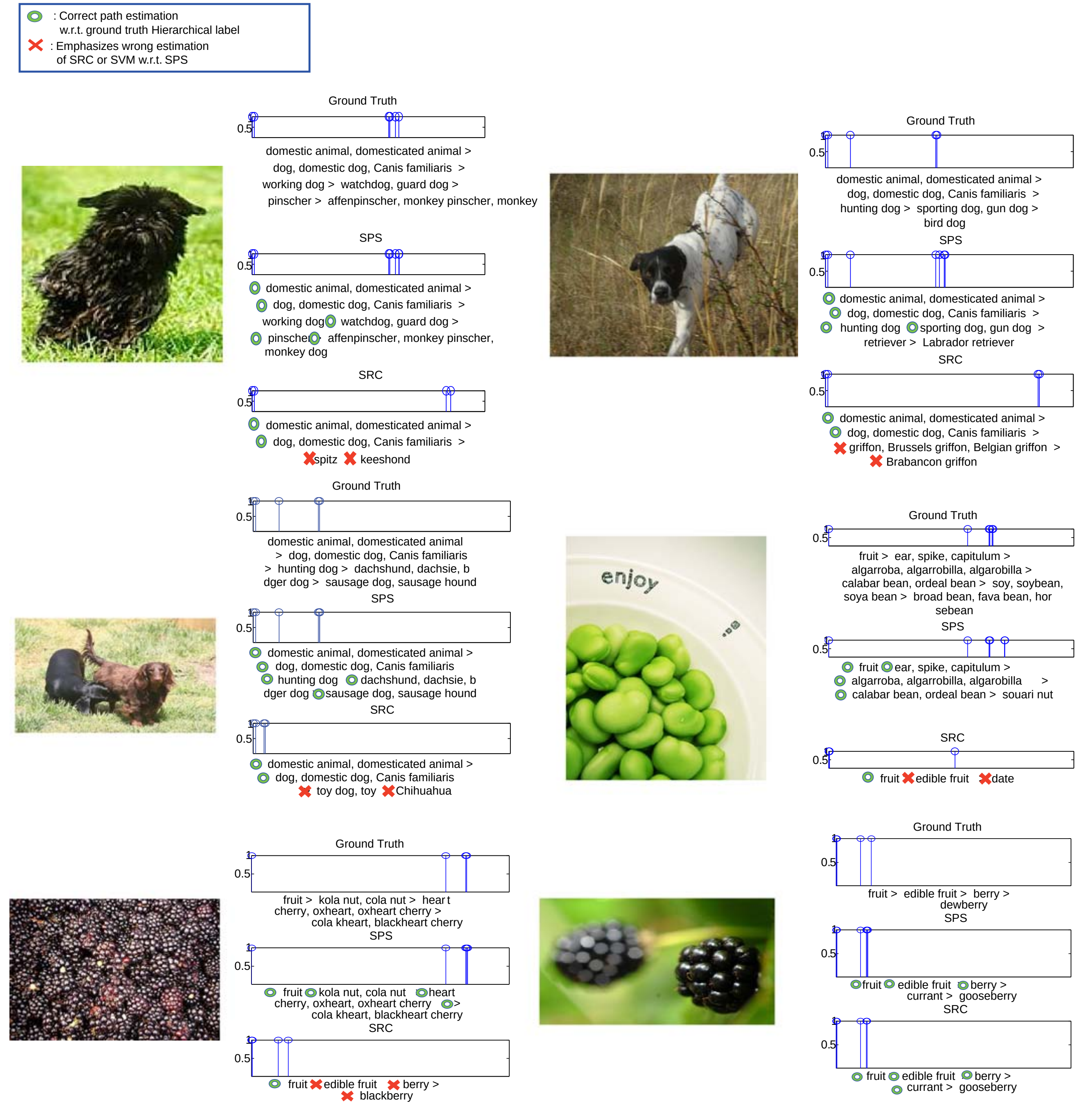


Fig. 8. The hierarchical path is estimated as non-zero entries in the encoded mixing vector ℓ. Note that the path estimated by SPS (ours) is closer to ground truth path than SRC is. Each response (non zero entry) corresponds to a classified category label for different levels of hierarchy. All these responses define the path ℓ (=mixing matrix). For instance, look at the top left example. Ground truth path is domestic animal, domesticated animal > dog, domestic dog, *Canis familiaris* > working dog > watchdog, guard dog > pinscher > affenpinscher, monkey pinscher, monkey dog. Each category label, e.g., working dog, corresponds to a non-zero entry in the mixing matrix ℓ; e.g., working dog is a category within the third level of the hierarchy. Notice that the SPS (our algorithm) returns the correct path (each category for each level is correctly estimated). Conversely, SRC estimates the first two levels correctly, but it returns the wrong estimation starting from the third level.

explain all of our empirical observations.[3] We note that if we choose two objects independently and uniformly at random from the learned database, the coherence between these two objects decreases as a function of their distance in the hierarchical tree.

Fig. 4 shows the relationship between the coherence of two objects (on average for objects chosen uniformly at random) and their distance (path length) in the hierarchical tree for the ImageNet dataset. This analysis suggests that instances of the same object category are similar while instances of different categories are, with high probability, dissimilar. Instead of tweaking the parameters of the "flat" sparse approximation in hopes of a small improvement, we should search for a sparse approximation that

[3] In our experimental results, we can see that a sparse representation that does not take into account any structure amongst the instances is surprisingly successful, albeit far from the best solution.

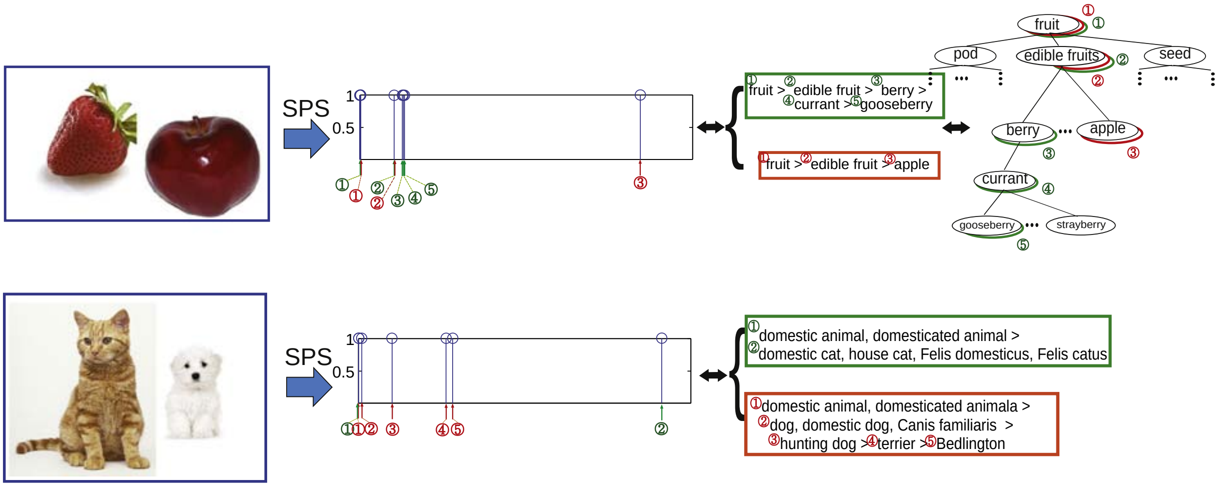

SPS
1
0.5
①
②
③
④
⑤
fruit > edible fruit > berry > currant > gooseberry
fruit > edible fruit > apple
fruit
pod
edible fruits
seed
berry
apple
currant
gooseberry
strayberry
SPS
1
0.5
domestic animal, domesticated animal >
domestic cat, house cat, Felis domesticus, Felis catus
domestic animal, domesticated animala >
dog, domestic dog, Canis familiaris >
hunting dog > terrier > Bedlington

Fig. 9. Multi objects recognition examples. Each test image contains two object categories. A level-zero pyramid histogram of codewords is used to represent the image in this case. Our SPS algorithm returns a mixing matrix where two paths can be identified. Each path is associated to a different object category. In top example, the estimated path associated to apple (i.e., fruit > edible fruit > apple) is indicated in red; the estimated path associated to gooseberry (i.e. fruit > edible fruit > berry > currant > gooseberry) is indicated in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

takes into account the hierarchical structure amongst the objects (and their categories).

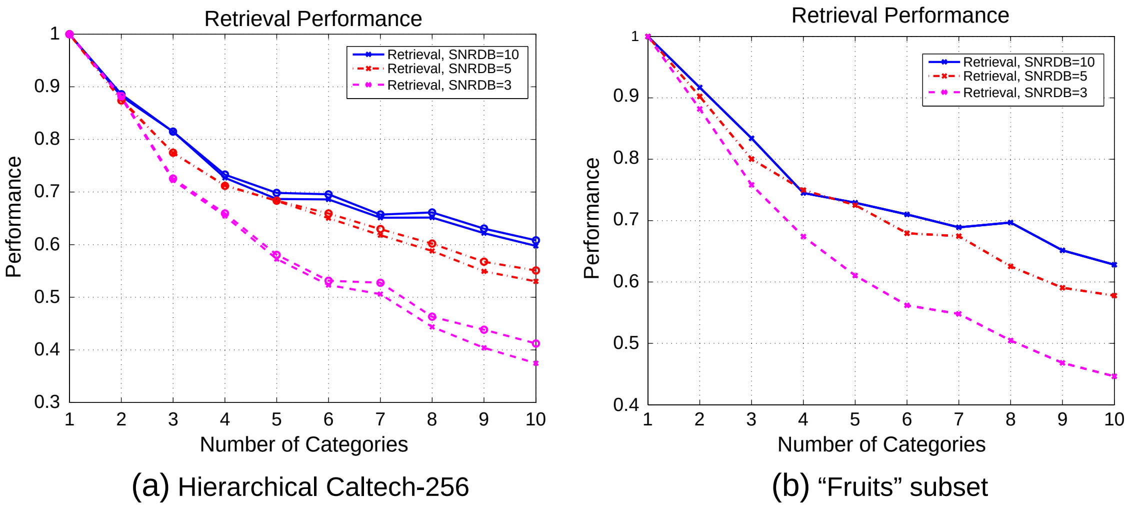
In practice, our data are not randomly generated. To test whether or not our data are consistent with our theoretical analysis, we show in Fig. 5 a QQ-plot, which shows a similarity between two probability distributions, for both a normal distribution and the entries of the matrices H (left) and Φ (right), respectively. Specifically, if samples are concentrated at diagonal lines, two distributions are similar. Thus, the plots show that the Φ distribution is closer to a normal distribution than H is but somewhat more skewed to negative values as compared to a normal distribution.

4.4. Benchmarks

The sparse approximation technique introduced by [22] (SRC) is used. We use Problem 1 (Section 2) to find the solution m via sparse approximation (similarly to [22]). We use the post-processing procedure in [22] to estimate the final class label. Notice that this method does not exploit the hierarchical structure of the database and "sees" the database as flat. Notice that SRC returns a single class label (not a path in the tree) which can be used to form the mixing vector m_{SRC}. In order to compare SRC results with ours, we embed m into its corresponding path $\ell_{\mathrm{SRC}} = Em_{\mathrm{SRC}}$. Notice that classifying ℓ correctly is as challenging as classifying m correctly since we don't know in advance the depth of the ground truth path.

4.5. Hierarchical similarity verification

In this section, we show classification results in terms of HD (which is a natural distance function to compare the similarity of two paths in a tree). Thus, if the ground truth path and the estimated path are similar, the HD will be small. In Fig. 6 we show average HD between ground truth paths and estimated path for all our testing images using our approach (SPS). In the same figure we also report the HD distance between ground truth path and path estimated by SRC (i.e., ℓ_{SRC}). Note that the HD associated to our approach is systematically smaller for all the datasets. This result supports our argument that the proposed framework yields smaller HD bounds. Also, notice that when the



(a) Hierarchical Caltech-256

(b) "Fruits" subset

Fig. 10. Numerical results showing how accurately the algorithm is capable to retrieve multiple images at the same time (in this case we assume query images are also contained in the database; we point out that in all the other classification experiments presented in this paper, test images are not contained in the database). Here x is a superposition of (up to) 10 histograms, and the goal is retrieve the ground truth paths given x. In this experiment, in order to simulate the effect of background clutter and intra-class variability, we added Gaussian noise on top of query image so as to have SNRdB from 3 to 10. As expected retrieval performances decrease as the number of categories increases, or the noise ratio increases. This analysis is interesting as it can be related with our theoretical findings.

hierarchical structure is relatively flat, the effect of encoding (and thus the advantage from our framework) becomes less significant (Fig. 8).

4.6. Effect on different hierarchy levels

HD returns a global measurement of path similarity regardless of the level and position in the tree. In this experiment we explore the performance of our framework at different levels of the tree. Fig. 7 plots the accuracy versus the levels of the hierarchy for different datasets (see caption for details). Notice that the root node is always classified correctly. As we go down toward the bottom of the tree, the likelihood of classifying nodes correctly becomes smaller and smaller. Also, note that this graph is always monotonically decreasing because whenever the estimation of the child category is correct, parent category estimation is correct too. When the hierarchical level is low, the performance of our SPS is similar to SRC. Interestingly, the plot shows that two algorithms yield equivalent performances in classifying images belonging to the leaf nodes. However, when the hierarchical level increases the gap between our SPS and SRC becomes much larger. This demonstrates the ability of our method to yield higher rates in classifying ancestors of the query object category. Anecdotal examples of paths returned by our SPS algorithm compared with those returned by SRC are shown in Fig. 5. Note that estimated parent nodes returned by SRC are much less accurate than those returned by SPS. Paths are reported in text format.

4.7. Multiple category classification

We report anecdotal examples demonstrating that our framework is able to classify images containing multiple categories. Assuming that there is no noise from background clutters, the histogram representing the query image can be expressed as a superimposition of multiple object category histograms (See examples in Fig. 9). So, as discussed in the technical section, our SPS method will return multiple paths — a path for each of category in the query image. Examples in Fig. 9 show some successful cases. Paths are reported in text format. The numerical results showing how accurately the algorithm is capable of retrieving multiple categories are shown in the Fig. 10.

5. Conclusion

In this work, we introduced a novel framework for hierarchical classification using a new formulation of the sparse approximation problem. We demonstrated, for the first time (up to our knowledge), that the hierarchical structure of a large and complex database can be indeed successfully used to enhance classification accuracy. Experimental results on several large scale dataset were used to support our claims.

References

[1] R. Baraniuk, V. Cevher, M. Duarte, C. Hegde, Model-Based Compressive Sensing, 2008. (preprint).

[2] A. Binder, M. Kawanabe, U. Brefeld, Efficient Classification of Images with Taxonomies 2003.

[3] T. Blumensath, M. Davies, Sampling theorems for signals from the union of linear subspaces, IEEE Trans. Inf. Theory (2007) 30–56.

[4] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207.

[5] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, Citeseer, 2004, p. 22.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, Proc. CVPR, 2009, pp. 710–719.

[8] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, The PASCAL visual object classes challenge 2006 (VOC2006) results, Workshop in ECCV06, Citeseer, May. Graz, Austria', 2006.

[9] C. Fellbaum, et al., WordNet: An Electronic Lexical Database, MIT press Cambridge, MA, 1998.

[10] K. Grauman, T. Darrell, The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features, Citeseer, 2005.

[11] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset2007.

[12] G. Griffin, P. Perona, Learning and using taxonomies for fast visual categorization, IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, 2008, pp. 1–8.

[13] La, C. Do, M. n.d., Tree-based orthogonal matching pursuit algorithm for signal reconstruction, *in* 'IEEE International Conference on Image Processing (ICIP)', Citeseer, pp. 1277–1280.

[14] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, Citeseer, 2006.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, M. Wood, N. Hengartner, E. Matzner-Lober, L. Rouvière, T. Burr, N. Malyshkina, et al., Online learning for matrix factorization and sparse coding, Statistics 1050 (2009) 1.

[16] M. Marszalek, C. Schmid, Semantic Hierarchies for Visual Object Recognition, 2007.

[17] S. Palmer, Vision Science: Photons to Phenomenology, MIT press Cambridge, MA, 1999.

[18] Saunders, S. Chen, D. Donoho, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1996) 33–61.

[19] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B 58 (1994) 267–288.

[20] J. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inf. Theory 50 (10) (2004) 2231–2242.

[21] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 104.

[22] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, pp. 210–227.

[23] A. Zweig, D. Weinshall, Exploiting object hierarchy: combining models from different category levels, IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.