Computer Vision and Image Understanding 132 (2015) 3–11

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Large margin learning of hierarchical semantic similarity for image classification ☆

Ju Yong Chang, Kyoung Mu Lee *

School of Electrical Engineering, ASRI, Seoul National University, 599 Gwanak-ro, Gwanak-gu, Seoul 151-744, Republic of Korea

ARTICLE INFO

Article history:
Available online 10 December 2014

Keywords:
Image classification
Similarity learning
Semantic representation
Large-margin framework

ABSTRACT

In the present paper, a novel image classification method that uses the hierarchical structure of categories to produce more semantic prediction is presented. This implies that our algorithm may not yield a correct prediction, but the result is likely to be semantically close to the right category. Therefore, the proposed method is able to provide a more informative classification result. The main idea of our method is two-fold. First, it uses semantic representation, instead of low-level image features, enabling the construction of high-level constraints that exploit the relationship among semantic concepts in the category hierarchy. Second, from such constraints, an optimization problem is formulated to learn a semantic similarity function in a large-margin framework. This similarity function is then used to classify test images. Experimental results demonstrate that our method provides effective classification results for various real-image datasets.

1. Introduction

Recognizing categories of objects and scenes is one of the most critical problems in computer vision. Although continuous progress has been made in this field, there still remains a large gap between machine performance and human intelligence. Unlike machines, humans can categorize at least tens of thousands of objects and scenes without any difficulty [1]. Furthermore, they can build a hierarchy of categories by simply observing images, and exploit it to produce semantically more meaningful judgement. For example, someone may mistakenly classify a dog as a cat but hardly misclassify a dog as a car. This example shows that one can produce a more informative classification result by considering the similarity between two semantic concepts. In the current work, we focus on this issue and attempt to make the image classification algorithm more semantic and human-like.

To achieve this goal, image classification algorithm should be developed under a new performance evaluation criterion. This criterion can be formulated by utilizing the hierarchical loss, which reflects the hierarchy of various semantic concepts, and not the flat 0/1 loss. Similar to [2], the hierarchical loss can be defined based on WordNet [3], a lexical semantic network for modeling human psycholinguistic knowledge. Under the hierarchical loss-based criterion, misclassifying an image as a different but semantically close category incurs a smaller loss than misclassifying it as a semantically distant category. Therefore, image classification can be significantly more informative by learning the algorithm based on the hierarchical loss. As shown in Fig. 1, the use of the hierarchical loss can provide substantial benefit to the results of classification.

In the present paper, a new image classification method is presented, which utilizes the hierarchical loss function and produces semantically more meaningful results. The key idea of this approach is a novel combination of semantic representation and similarity function learning. Several recent works are available that explicitly estimate high-level semantic attributes for various applications, such as description of generic or unfamiliar images [4,5], zero-shot transfer learning [6], and intermediate features that can aid in visual recognition [5–8]. We adopt semantic representation for the latter purpose to produce low-dimensional semantic feature vectors. The low dimensionality of our feature vector helps in the subsequent similarity learning being performed very efficiently compared with the conventional similarity or distance function learning [9–11] that usually directly handles high-dimensional low-level feature vectors. Moreover, because our feature vector is semantic, enforcing constraints among semantic concepts for the minimization of hierarchical loss is easy. More specifically, learning the similarity function, which is the core learning problem of our approach, is formalized within *a large-margin framework* and is guaranteed to minimize empirical hierarchical loss.

☆ This paper has been recommended for acceptance by Chung-Sheng Li.

* Corresponding author. Fax: +82 2 878 1452.
E-mail addresses: jangbon@snu.ac.kr (J.Y. Chang), kyoungmu@snu.ac.kr (K.M. Lee).

http://dx.doi.org/10.1016/j.cviu.2014.11.006
1077-3142/

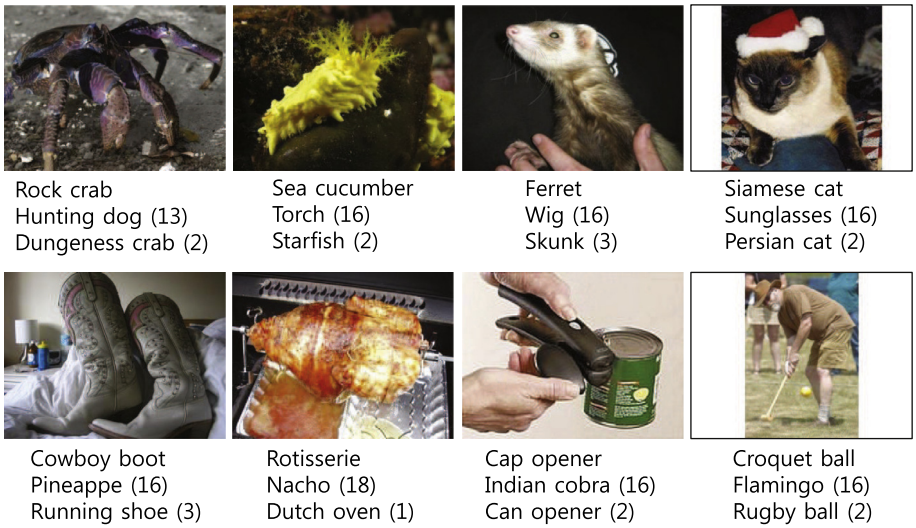

Fig. 1. Image classification results by exploiting class hierarchy versus those without considering hierarchy. The first label is the ground truth category. The second and third labels are the results of flat prediction (one-versus-all SVM) and hierarchical prediction (our method), respectively. The numbers indicate the hierarchical loss.

The contributions of the present work include the following:

- A novel large-margin formulation for semantic similarity learning is proposed. This learning problem can be viewed as an instance of a semidefinite program (SDP) [12], and an efficient optimization algorithm is developed.
- A thorough experimental study is conducted for comparing the performances of several algorithms for hierarchical image classification [13–15]. For this purpose, the Caltech [16,17] and ImageNet [18] datasets are used.
- The proposed method is shown to achieve a state-of-the-art classification result under the hierarchical-loss criterion. Furthermore, a noticeable gain in the conventional measures, such as accuracy and precision, can also be obtained.

The rest of the current paper is organized as follows. In Section 2, some related works are discussed. Section 3 presents the framework of the proposed method, followed by the presentation of the experimental results in Section 4. Finally, Section 5 concludes this paper.

2. Related works

The document categorization method proposed by Cai and Hofmann [13] treats the category structure above *flat* and considers the relationships among categories, which are commonly expressed in concept hierarchy or taxonomies. The task of exploiting these pre-determined taxonomies as additional information for classification fits well into the popular **structured learning framework** [19,20]. This idea enables us to use not just the flat 0/1 loss but also the more general loss function. In [13], the hierarchical loss function between two categories is defined and then minimized to provide the hierarchical classifier based on the structural support vector machines (S-SVMs). The experimental results of [13] on the document categorization show that S-SVM outperforms flat support vector machines (SVMs) in terms of hierarchical loss.

The main drawback of the structured learning approach is its high computational complexity, which renders the approach inherently slow even for the ordinary case of several dozen categories. To address this problem, Binder et al. [14] proposed an efficient alternative to the structured approach by decomposing the problem into several local tasks. The idea is to learn a binary SVM for each node in the taxonomy tree instead of solving the whole problem at once using the structured learning approach. At the final stage of their approach, **the ensemble of local binary SVMs** from all nodes is appropriately assembled by reflecting the taxonomy. They applied their classification method to real-world image data, such as Caltech256 [17] and VOC2006 [21], and reported that their local approach performed at par with the structured approach in terms of hierarchical loss while being considerably faster in training. However, the number of nodes in the taxonomy tree rapidly increases as the number of categories increases, which makes the method not free from computational burden.

To move from the flat classification to settings that utilize the category hierarchy information, Weinberger and Chapelle [15] proposed a very different approach from structured learning approaches. Instead of learning a classifier, their method solves a regression problem where images are mapped into a latent semantic space. This semantic space is learned in a supervised manner and underlies the category taxonomy, which is the reason why this method is referred to as **taxonomy embedding** (abbreviated to *taxem*). It first performs ridge regression to embed input features into a low-dimensional semantic space and then learns the distance function in the semantic space based on the large-margin framework. This two-step approach inspired our current work. Nevertheless, our method significantly differs in its essential features, such as the methods used in obtaining the semantic space and comparing two semantic vectors. Moreover, our approach is extensively tested on image data, whereas taxem was only applied to document categorization.

In [22], **hierarchical similarity** among semantic representation of images for large-scale image retrieval was proposed. Their approach incorporates prior hierarchy information and achieves significant improvements over state-of-the-art image retrieval methods. It performs learning to recognize the semantic attributes of images and then computes a similarity score by using a predefined comparison function. This comparison function is determined only based on a known hierarchical structure and does not utilize the training data, causing the hierarchical similarity method to produce suboptimal classification results compared with our method where the similarity function is determined by a large-margin learning framework with the training data. Further, they applied additional probabilistic calibration [23] to the semantic attributes after SVM, which is not necessary for our method.

A number of studies have been conducted, which considered learning and exploiting class hierarchies for visual recognition [24–27]. In those approaches, the hierarchical structure of classes is automatically learned from the image data, whereas our approach utilizes the predetermined hierarchy from WordNet. The main concern of these approaches is the improvement of computational efficiency by using the learned hierarchy. Although some of these works reported performance gains in the final classification, the evaluation criterion used was not the hierarchical loss but the flat 0/1 loss.

3. Proposed method

Our training data are assumed to consist of images, represented as a set of high-dimensional vectors $x_1, \ldots, x_n \in \mathcal{X}$ of dimensionality d. In addition, the images are accompanied by image category labels $y_1, \ldots, y_n \in \{1, \ldots, c\}$ that lie in a certain taxonomy $\mathcal{T}$ with total c categories. This taxonomy $\mathcal{T}$ gives rise to some cost matrix $C \in \mathbb{R}^{c \times c}$, where $C_{\alpha\beta} \geqslant 0$ defines the cost of misclassifying an image of category α as β and $C_{\alpha\alpha} = 0$. Among the various methods used in defining the cost matrix from the taxonomy [28,2], we follow that of [2] and define

$$C_{\alpha\beta} = \frac{H(\alpha, \beta)}{H^*}, \tag{1}$$

where $H(\alpha, \beta)$ denotes the height of the lowest common ancestor of classes α and β, and H^* denotes the height of the root node in the taxonomy tree. By dividing $H(\alpha, \beta)$ by $H^*, C_{\alpha\beta}$ is normalized to $[0, 1]$. This definition is equivalent to predicting a path along the hierarchy and evaluating where the ground truth path and the predicted path diverge, which can be viewed as a direct measure of the semantic level at which a misclassification occurs.

Assuming that given the input vectors $x_1, \ldots, x_n$, our classifier estimates their category labels as $\hat{y}_1, \ldots, \hat{y}_n$. Ideally, the classifier should then be learned by minimizing the following empirical training error:

$$\varepsilon = \frac{1}{n} \sum_{i=1}^{n} C_{y_i \hat{y}_i}. \tag{2}$$

Fig. 2 shows the overview of our approach. A low-dimensional semantic feature space $\mathcal{F}$ is first created where each image vector $x_i \in \mathcal{X}$ is represented as a low-dimensional vector $z_i \in \mathcal{F}$. In the current work, this semantic space is defined as c-dimensional Euclidean space $\mathbb{R}^c$. Its unit vector e_α $(\alpha = 1, \ldots, c)$, codirectional with each axis, denotes the prototype, which is the most representative member within the category α similar to the prototype theory by Rosch [29]. A similarity function $Sim(a, b)$ is then introduced, which indicates the degree of resemblance between two semantic vectors a and b. This similarity function is learned by minimizing the empirical training error (2) in a large-margin sense. Final inference is done by computing the test image's similarity to all prototypes based on the learned similarity and by selecting the most similar category.

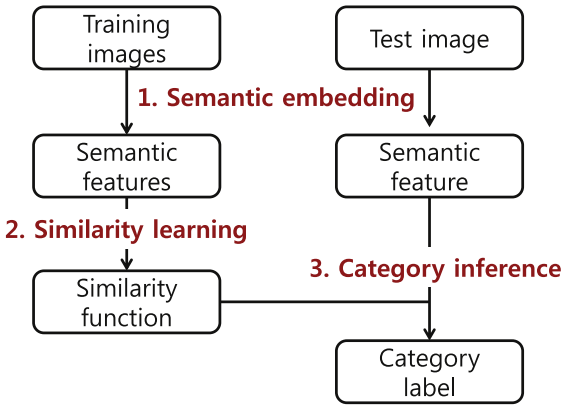

Fig. 2. Overview of our proposed method.

3.1. Semantic embedding via SVM

The first step of our algorithm is the embedding of the input image into a Euclidean vector space, which is semantic and low dimensional. For this purpose, a representation similar in spirit to the semantic representation based on the classifier's output as in [7] is considered. To obtain the semantic representation, a binary classifier for each category is independently learned and one-versus-all linear SVM is chosen for its high classification performance and efficiency. The score of the SVM classifier can be denoted as $h_\alpha(x) = w_\alpha^T \tilde{x}$ $(\alpha = 1, \ldots, c)$, where $\tilde{x}$ is a reparameterized vector $\tilde{x} = [x^T, 1]^T$. The c binary SVMs are trained as follows:

$$\min_{w_\alpha} \frac{1}{2} \|w_\alpha\|^2 + \lambda \sum_{i=1}^{n} [1 - y_i^\alpha w_\alpha^T \tilde{x}_i]_+, \tag{3}$$

where $y_i^\alpha = 1$ if $y_i = \alpha$; otherwise $y_i^\alpha = -1$, and the term $[x]_+ = \max(x, 0)$ denotes the standard hinge loss. The semantic representation $z \in \mathcal{F}$ corresponding to the input image $x \in \mathcal{X}$ is then defined as

$$z = [h_1(x), \ldots, h_c(x)]^T. \tag{4}$$

This step is easily parallelized as the classifiers can be independently learned.

If the category hierarchy is not considered, the category label of the input image x is usually inferred by finding the maximum classifier score as $\hat{y} = \arg\max_\alpha h_\alpha(x)$. This equation can be rewritten as $\hat{y} = \arg\max_\alpha e_\alpha^T z$, where $e_\alpha = [0, \ldots, 1, \ldots, 0]^T$ is a c-dimensional unit vector with all zeros and a single one in the α^{th} position. Therefore, in this case, $e_\alpha^T z$ can be understood to denote the similarity function between the category α's prototype e_α and the semantic vector z. In the current paper, this dot product similarity $Sim(a, b) = a^T b$ is generalized to the bilinear form:

$$Sim(a, b) = a^T S b, \tag{5}$$

where $S \in \mathbb{R}^{c \times c}$ denotes the similarity matrix. Given an input vector x_t, the inference rule in estimating its label can then be defined as finding the category with maximum similarity as follows:

$$\hat{y}_t = \arg\max_\alpha Sim(z_t, e_\alpha) = \arg\max_\alpha z_t^T S e_\alpha. \tag{6}$$

The goal of the next step is to learn the similarity matrix S by exploiting the category hierarchy $\mathcal{T}$.

3.2. Optimization problem for semantic similarity learning

Ideally similarity matrix S should be learned to minimize directly the empirical training error (2). However, because this function is non-continuous and non-differentiable, a surrogate loss function that strictly bounds (2) is derived and subsequently minimized. To construct such a loss function, the large-margin framework is followed, so the semantic vector z_i should be similar to the correct prototype e_{y_i} than any other prototype e_α by a large margin. The hierarchical information can also be exploited by enforcing that the prototypes incurring a smaller misclassifying cost would further put close to the semantic vector than those incurring a large cost. More explicitly, a margin of $C_{y_i\alpha}$ is used. This condition is expressed as a set of soft inequality constraints, i.e.,

$$\forall i, \alpha \neq y_i \; Sim(z_i, e_{y_i}) + \xi_{i\alpha} \geqslant Sim(z_i, e_\alpha) + C_{y_i\alpha}, \tag{7}$$

where the slack variable $\xi_{i\alpha} \geqslant 0$ is responsible for the violation of prototype e_α into the margin of z_i. From this formulation, an upper bound on the empirical training error (2) can be obtained, as in the following theorem.

Theorem 3.1. *Given a similarity matrix S, the empirical training error (2) is bounded above by* $\frac{1}{n}\sum_{i,\alpha}\xi_{i\alpha}$.

Proof. Assume that $\hat{y}_i$ denotes the inferred category label according to (6). It follows that $Sim(z_i, e_{\hat{y}_i}) - Sim(z_i, e_{y_i}) \geqslant 0$ for all i (equality holds for $\hat{y}_i = y_i$). We therefore obtain $\xi_{i\hat{y}_i} \geqslant Sim(z_i, e_{\hat{y}_i}) + C_{y_i\hat{y}_i} - Sim(z_i, e_{y_i}) \geqslant C_{y_i\hat{y}_i}$. It then follows from $\xi_{i\alpha} \geqslant 0$ that $\sum_{i,\alpha}\xi_{i\alpha} \geqslant \sum_i \xi_{i\hat{y}_i} \geqslant \sum_i C_{y_i\hat{y}_i}$. □

By combining Theorem 3.1 with the constraints in (7), an optimization problem that minimizes the upper bound of the empirical training error in (2) with maximum margin constraints can be constructed:

$$\begin{aligned} \min_S \quad & \textstyle\sum_{i,\alpha}\xi_{i\alpha} \\ s.t. \quad & Sim(z_i, e_{y_i}) + \xi_{i\alpha} \geqslant Sim(z_i, e_\alpha) + C_{y_i\alpha} \\ & \xi_{i\alpha} \geqslant 0. \end{aligned} \tag{8}$$

The above optimization problem is convex because the inequality constraints in (7) are linear with respect to S.

A number of previous works on the similarity or distance function learning focused on a metric, such as in the case of a positive semidefinite matrix that defines a Mahalanobis distance [9,30,10]. In these works, adding positivity constraints to enforce metric properties appears to be helpful in reducing overfitting and improving generalization. Such strategy is adopted in the current paper; thus, positive semidefiniteness of S by adding the constraint $S \succeq 0$ is enforced. This positivity constraint is empirically found to be essential in order for the learned semantic similarity to work well.

To further avoid the overfitting problem, a regularization term is added to the objective function. $\|S - \bar{S}\|_F^2$ is used as a regularization term, which enforces the similarity matrix S not to deviate too much from the prior similarity $\bar{S}$. The form of this prior matrix can be chosen by exploiting the information of the predefined semantic hierarchy as follows. $Sim(z, e_\alpha)$ returns the similarity of input vector z to prototype e_α, so that it can be understood as a classifier for category α. It can be expressed as $Sim(z, e_\alpha) = z^T S e_\alpha = [h_1(x), \ldots, h_c(x)][S_{\alpha 1}, \ldots, S_{\alpha c}]^T = \sum_{\beta=1}^{c} h_\beta(x) S_{\alpha\beta}$. This equation means that the classifier is the weighted sum of c SVM's scores ($h_\beta(x)$) with their corresponding weights ($S_{\alpha\beta}$). Let the prior matrix be defined as

$$\bar{S}_{\alpha\beta} = 1 - C_{\alpha\beta}. \tag{9}$$

By this definition, the weight ($\bar{S}_{\alpha\beta}$) increases as the semantic distance between categories α and β decreases, and vice versa. Therefore, our new hierarchical classifier is reasonably regularized by the information of neighboring categories' classifiers based on their semantic distance to the current category.

By combining the above-mentioned items, the final convex optimization problem of our method becomes the following:

$$\begin{aligned} \min_S \quad & \mu\sum_{i,\alpha}\xi_{i\alpha} + (1-\mu)\|S - \bar{S}\|_F^2 \\ s.t. \quad & Sim(z_i, e_{y_i}) + \xi_{i\alpha} \geqslant Sim(z_i, e_\alpha) + C_{y_i\alpha} \\ & \xi_{i\alpha} \geqslant 0 \\ & S \succeq 0, \end{aligned} \tag{10}$$

where constant $\mu \in [0, 1]$ controls the strength of the regularization term. The optimization problem in (10) is an instance of SDP [12].

3.3. Optimization algorithm

The standard SDP solver solves the problem in (10) very slowly. Thus, we follow the approach in [31] to propose an algorithm that can efficiently solve our problem using the subgradient method. Our algorithm can handle the problem with tens of thousands of images and several hundred categories based on the fact that the subgradient can be efficiently computed from the previous subgradient by checking the margin violation constraints in (7).

The objective function in (10) can be reformulated by removing the slack variables as follows:

$$E(S) = \mu\sum_{i,\alpha}[Sim(z_i, e_\alpha) - Sim(z_i, e_{y_i}) + C_{y_i\alpha}]_+ + (1-\mu)\|S - \bar{S}\|_F^2. \tag{11}$$

This function should be minimized over the positive semidefinite matrix $S \succeq 0$. For such purpose, an iterative subgradient projection method is implemented. Let the similarity matrix at the tth iteration be denoted by S_t. At each iteration, the optimization algorithm takes a step along the subgradient to reduce the objective function (11) and then projects S_t onto the cone of all positive semidefinite matrices.

At the tth iteration, the objective function in (11) can be rewritten as

$$E(S_t) = \mu \sum_{(i,\alpha)\in\mathcal{N}_t} \left(z_i^T S_t e_\alpha - z_i^T S_t e_{y_i} + C_{y_i\alpha}\right) + (1-\mu)\|S_t - \bar{S}\|_F^2, \tag{12}$$

where $\mathcal{N}_t$ denotes a set of pairs, such that $(i, \alpha) \in \mathcal{N}_t$ if and only if the indices (i, α) trigger the hinge loss in (11). The gradient G_t of $E(S_t)$ can then be computed as

$$G_t = \frac{\partial E}{\partial S_t} = \mu \sum_{(i,\alpha)\in\mathcal{N}_t} (z_i e_\alpha^T - z_i e_{y_i}^T) + 2(1-\mu)(S_t - \bar{S}). \tag{13}$$

The computational complexity mainly depends on the first term, i.e., computation of outer products ($z_i e_\alpha^T$ and $z_i e_{y_i}^T$) for all indices in $\mathcal{N}_t$. However, this term can be efficiently computed by considering only the differences between the sets $\mathcal{N}_t$ and $\mathcal{N}_{t-1}$. From this fact, the gradient equation in (13) can be rewritten as

$$\begin{aligned} G_t = {} & \mu \sum_{(i,\alpha)\in\mathcal{N}_{t-1}} \left(z_i e_\alpha^T - z_i e_{y_i}^T\right) - \mu \sum_{(i,\alpha)\in\mathcal{N}_{t-1}-\mathcal{N}_t} \left(z_i e_\alpha^T - z_i e_{y_i}^T\right) \\ & + \mu \sum_{(i,\alpha)\in\mathcal{N}_t-\mathcal{N}_{t-1}} \left(z_i e_\alpha^T - z_i e_{y_i}^T\right) + 2(1-\mu)(S_t - \bar{S}). \end{aligned} \tag{14}$$

The second and third terms represent the contributions from pairs that are no longer active and the contributions of those that just become active. Because the first term is already computed at the previous iteration and the set $\mathcal{N}_t$ usually changes very little from one iteration to the next, the computation of the gradient G_t in (14) can be done very efficiently.

Note that unlike the general case in [31], the proposed algorithm in (14) can have the additional benefit in the subgradient computation step. In general, the complexity of the outer products computation in (14) is $O(mc^2)$, where m is the number of (i, α) pairs that contribute the subgradient update and c is the dimensionality of the semantic vector. However, in our case, one of two operands in the outer product is always the prototype vector, that is, the unit vector, and therefore the computational complexity reduces to $O(mc)$. This gives us the substantial gain in the learning time and it is clearly shown in our experiments compared to the taxonomy embedding approach in [15] that is also based on [31].

The minimization of the objective function in (12) should enforce the constraint that the matrix S_t remains a positive semidefinite matrix. To enforce this constraint, S_t is projected onto the cone of all positive semidefinite matrices after each step. This projection is computed from the diagonalization of S_t. Let

$S_t = Q\Lambda Q^T$ denote the eigendecomposition of S_t, where Q is the orthonormal matrix of eigenvectors and Λ is the diagonal matrix of the corresponding eigenvalues. Let $\Lambda^+ = max(\Lambda, 0)$ denote the diagonal matrix containing all the positive eigenvalues. The projection of S_t onto the cone of positive semidefinite matrices is then given by

$$\mathcal{P}(S_t) = Q\Lambda^+ Q^T. \tag{15}$$

3.4. Inference

The learned embedding function in Section 3.1 and the learned similarity function in Section 3.2 can be used to classify the test images. Test image x_t is first transformed into z_t using (4) and then compared with all prototypes e_α with the similarity matrix S using (6), giving the category estimate $\hat{y}_t$.

4. Experimental results

4.1. Datasets and evaluation criteria

Caltech256 [17] and ILSVRC [32], a subset of ImageNet [18] with 1000 classes and 1.2 million images, are used in our experiments. For the Caltech256 dataset, according to [14], the *Animals13* dataset is constructed by considering only 13 animal classes, specifically, all Protostomia, from 256 classes of objects. This small dataset serves as a toy dataset for comparing our method with others, including the structured approach [13] that generally requires much training time. Although another dataset that includes 52 animal classes is also introduced in [14], this is excluded because training the structured method takes 30 h aside from the cross-validation in our machine. The predefined taxonomy of biological systematics is exploited in constructing the taxonomy tree, which contains 13 leaf nodes and 22 internal nodes for the Animals13 dataset. Each of all the animal classes has at least 80 images, hence 60 training and 20 test images are selected for each class, resulting in 780 training and 260 test images. The random selection of the training/test images is repeated 10 times, and the results are averaged for final evaluation.

For the ILSVRC dataset, a subset of its training dataset is considered by randomly choosing 60 and 100 classes to construct the *ILSVRC60* and *ILSVRC100* datasets. As a large-scale dataset, we construct the *ILSVRC1000* dataset by considering all 1000 classes. From each class, 300 training and 100 test images are then selected, hence 18,000 training and 6000 test images are used in our experiments as ILSVRC60 dataset (30,000 training and 10,000 test images as ILSVRC100 dataset and 300,000 training and 100,000 test images as ILSVRC1000 dataset). To construct several datasets with different category and image configurations, this process is repeated five times for ILSVRC60/100 and two times for ILSVRC1000. The categories of ILSVRC are hierarchically organized according to WordNet.

Suppose for a set of test examples $\{(x_i, y_i)\}_{i=1}^n$, the classification method produces a set of output classes $\{\hat{y}_i\}_{i=1}^n$. To evaluate the effectiveness of image classification methods, three measures are employed: *accuracy*, *hierarchical loss*, and *precision*. Accuracy evaluates how often the prediction is correct and it is defined as

$$acc = \frac{1}{n}\sum_{i=1}^{n}[y_i = \hat{y}_i], \tag{16}$$

where $[\cdot]$ is one if the predicate inside is true; otherwise it is zero. Note that $1 - acc$ is the conventional flat 0/1 loss. The hierarchical loss has the same form as the empirical training error (2) except that it is further multiplied by H^*, the height of the root node:

$$hier = \frac{H^*}{n}\sum_{i=1}^{n} C_{y_i\hat{y}_i}. \tag{17}$$

Thus, the hierarchical loss can be interpreted as a degree of misclassification in terms of distance in the taxonomy tree. The above two measures evaluate the quality of the top-ranked category. However, the prediction results are sometimes viewed as a ranked list. Precision is a measure for evaluating the quality of this ranking result:

$$prec = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\{y : Sim(z_i, e_y) \geqslant Sim(z_i, e_{y_i})\}|}, \tag{18}$$

where $|\cdot|$ denotes the set's cardinality and z_i is a semantic representation of x_i as in (4). Roughly, $1/prec$ can be regarded as the average position of the true category.

For the large-scale datasets, it is useful to allow the classification algorithm to produce multiple predictions. In such a case, the evaluation measure is required to assess the quality of multiple predictions. For example, let us assume that the classification method produces k predictions $\hat{y}_{ij}, j = 1, \ldots, k$ for a test sample (x_i, y_i). These can be obtained by selecting k classes in descending order of confidence. The accuracy can then be defined as

$$acc_k = \frac{1}{n}\sum_{i=1}^{n}\max_{j=1}^{k}[y_i = \hat{y}_{ij}]. \tag{19}$$

The idea is that the ground truth label y_i should be matched by one of the outputs $\hat{y}_{ij}, j = 1, \ldots, k$ to be regarded as the correct prediction. The hierarchical loss can be similarly defined as

$$hier_k = \frac{H^*}{n}\sum_{i=1}^{n}\min_{j=1}^{k} C_{y_i\hat{y}_{ij}}. \tag{20}$$

4.2. Implementation details

In the following experiments, the bag of words (BoW) representation is used based on the densely sampled SIFT features. Each image is first resized to have a maximum side length of no more than 300 pixels. SIFT descriptors are then computed at points on a regular grid with a 10 pixel spacing. At each grid point, the descriptors are extracted at three different patch sizes ($8 \times 8, 16 \times 16$, and 32×32) to handle scale variation between images. The dense features are further processed to form a visual vocabulary of 400 visual words for the Animals13 dataset (1000 visual words for ILSVRC60/100/1000 datasets) using K-means clustering. The explicit feature map approach [33] that approximates various useful kernels, including the intersection and χ^2 kernels, is then adopted, enabling the use of the efficient linear SVM with high performance comparable to the nonlinear SVMs with implicit kernels. Specifically, the χ^2 kernel is adopted with approximation order $N = 1$ and sampling step $L = 0.7$. From all these steps, an image of the Animals13 dataset is converted to a 1200-dimensional feature vector (3000-dimensional vector for the ILSVRC60/100/1000 datasets).

Table 1
Performance analysis on the Animals13 dataset. Bold and italic numbers indicate the best and second-best results, respectively.

method	*acc* (%)	*hier*	*prec* (%)
Flat-Nolearning	35.88	2.61	53.34
Hier-Nolearning	23.46	2.98	39.97
Flat-Learning	*36.73*	*2.58*	*54.46*
Hier-Learning	**37.00**	**2.57**	**54.59**
Dist-Learning	36.23	2.60	53.53

Table 2
Performance comparison on the Animals13, ILSVRC60, and ILSVRC100 datasets. Bold and italic numbers indicate the best and second-best results, respectively.

Method	Animals13			ILSVRC60			ILSVRC100		
	acc (%)	*hier*	*prec* (%)	*acc* (%)	*hier*	*prec* (%)	*acc* (%)	*hier*	*prec* (%)
Ridge	*36.92*	*2.58*	*54.21*	38.17	7.53	51.61	31.66	8.20	44.55
Taxem [15]	36.58	*2.58*	54.20	*40.72*	*7.00*	*54.48*	*34.64*	*7.61*	*48.09*
SVM	35.88	2.61	53.34	38.76	7.47	51.44	32.47	8.08	44.33
SVMtax [14]	35.19	2.59	52.24	33.29	7.95	45.62	26.55	8.53	37.72
SSVMtax [13]	35.15	*2.58*	52.23	–	–	–	–	–	–
HPS [22]	35.65	2.61	51.71	40.03	7.20	50.19	34.20	7.72	43.62
Ours	**37.00**	**2.57**	**54.59**	**41.82**	**6.96**	**55.47**	**36.05**	**7.48**	**49.34**

Our algorithm has two parameters. One is penalty parameter λ of the binary SVMs (3) used for semantic embedding in Section 3.1, and the other is regularization parameter μ for the large-margin formulation (10) in Section 3.2. $\mu = 0.1$ provides good results, hence it is used for all our experiments. The parameter λ is determined by 3-fold cross-validation. Finally, the open source VLFeat library [34] is used to compute the image features and train the binary SVMs.

4.3. Performance analysis

Several variations of the proposed method are tested to justify that all parts of our algorithm are necessary for its performance. In the first variation (**Flat-Nolearning**), the category hierarchy is not exploited, and large-margin learning is also not performed, which can be easily tested using identity matrix as the similarity matrix S. This approach is equivalent to the plain binary SVMs. The second variation (**Hier-Nolearning**) exploits the hierarchical information using the prior similarity $\bar{S}$ in (9) but still does not incorporate the learning process. The third variation (**Flat-Learning**) is then used, where our large-margin learning framework described in Section 3.2 is used to optimize the similarity matrix. However, the flat 0/1 cost matrix for flat learning is used rather than the hierarchical cost matrix (1). Finally our algorithm (**Hier-Learning**) is tested.

The evaluation results of the Animals13 dataset are summarized in Table 1. The simple use of hierarchical information using the fixed prior similarity (Hier-Nolearning) degrades the results compared with even the plain SVMs (Flat-Nolearning). Its performance is dramatically improved with the application of our large-margin framework to the similarity matrix learning (Hier-Learning), which achieves the best performance for all criteria. Adopting the large-margin learning without considering class hierarchy (Flat-Learning) also results in significant performance improvements. Conclusion can be made that our similarity learning framework is crucial for performance gain, and the results can be further enhanced by exploiting the class hierarchy.

Another variation of our method is then tested to justify that learning the similarity function is better fitted to our semantic representation than the distance function learning. To learn the Mahalanobis distance, the algorithm in [15] is implemented, where the semantic vector is constructed via ridge regression rather than SVM. Ridge regression enforces the semantic vectors to be close to their correct category prototypes in terms of the Euclidean distance, which makes the subsequent distance learning problem well posed. However, the semantic vectors produced by our SVM embedding fail to meet such conditions, hence they must be further normalized. Therefore, normalized vectors are used, and the distance function learning in [15] is applied to them. The results are shown in Table 1, where **Dist-Learning** denotes such an approach. Dist-Learning does not perform well compared with our similarity learning approach, and this result supports the validity of our argument.

4.4. Performance comparison

Our method is compared with several related algorithms for image/document classification. The first is the structured approach (**SSVMtax**) [13], which can minimize the empirical training error in (2) using S-SVM. For implementation, the SVMstruct code of Joachims [20] is used. Second, the ensemble of local binary SVMs (**SVMtax**) [14] is considered, whose implementation is straightforward. The binary SVM for each node of taxonomy tree is learned, and the scores of all nodes lying on the path from the root to the leaf category are subsequently averaged. The scores are further scaled into [0,1] using a logistic function and then averaged by the geometric mean. The third algorithm is the taxonomy embedding (**Taxem**) approach [15], which has two parameters. One parameter is the weight of the regularization term in linear ridge regression, and the other is the regularization parameter μ for large-margin formulation. $\mu = 0.7$ is set for all experiments, which gives good performance. The hierarchical probabilistic similarity (**HPS**) approach in [22] is considered, originally proposed for image retrieval problem. Although [22] did not mention how to use their method for image classification, the process is similar to our framework, i.e., by computing the image's similarities to category prototypes. Finally, the two flat classification methods: one-versus-all **SVM** and ridge regression (**Ridge**), are considered. For fair comparison, the same image features as in our method are used to test the above-mentioned algorithms, and all parameters, except for μ of taxem, are determined by 3-fold cross validation.

The performance comparisons for our small and middle-scale datasets are shown in Table 2. The results of the Animals13 dataset are considered first. The methods exploiting category hierarchy produce lower hierarchical loss than the flat methods. Unlike other hierarchy-based methods, our algorithm achieves the best performance in both accuracy and precision measures, as well as in the hierarchical loss. For the test with middle-scale datasets, all the considered methods except for the SSVMtax are applied to five ILSVRC60 and five ILSVRC100 datasets, which have different category configurations from each other; the classification results are then averaged. The SSVMtax method is excluded because it requires too much time for training. For these middle-scale datasets, our algorithm performs best, and its performance gain becomes bigger than that of the Animals13 dataset. This result shows that our method can take advantage of the complex taxonomy structure for hierarchically more meaningful prediction. Figs. 3 and 4 present the evaluation results along with the dataset number, showing that our method consistently outperforms other methods in all cases.

For the large-scale ILSVRC1000 dataset, we compare three approaches: SVM, HPS, and ours. The evaluation results are summarized in Table 3 and Fig. 5. In this large-scale experiments, multiple predictions are allowed and the measures in (19) and (20) are used to evaluate the performance. Our algorithm produces the best results except the $hier_1$ case, where the HPS approach shows a slightly better output. However, our method readily catch up with

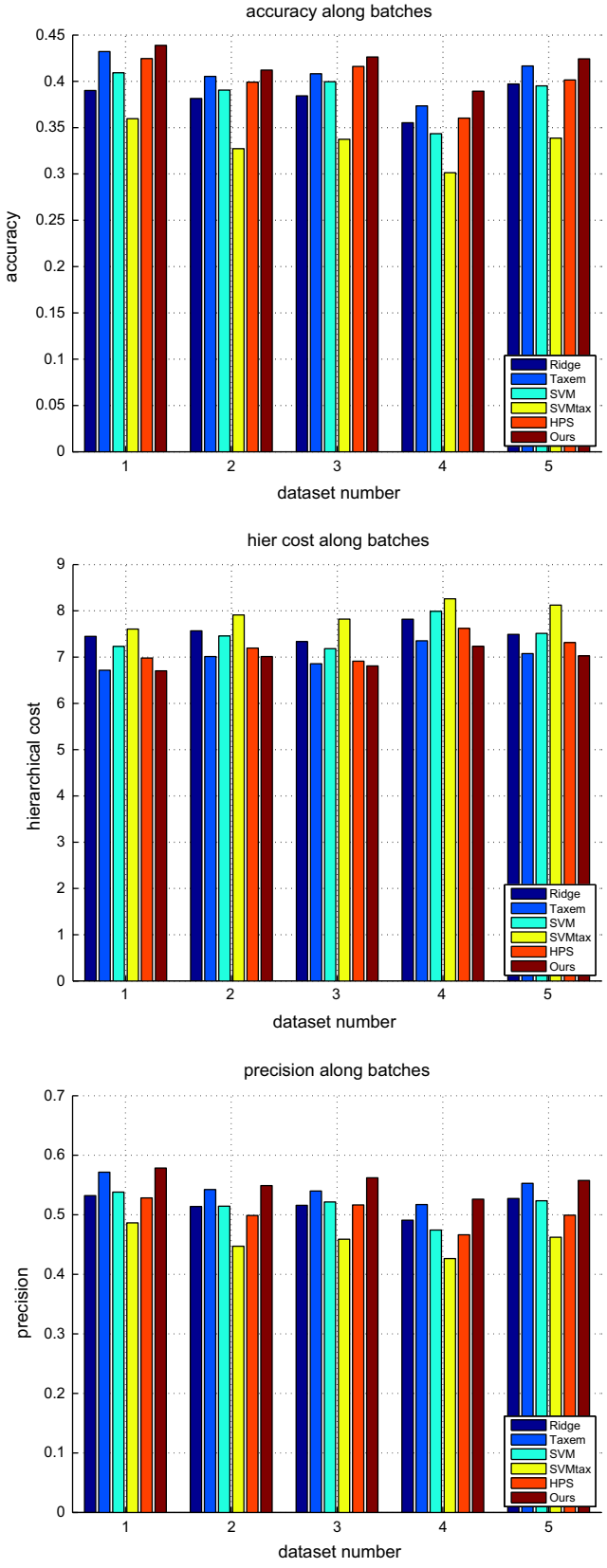

Fig. 3. The classification results of our method against others are plotted along with the dataset number for five ILSVRC60 datasets.

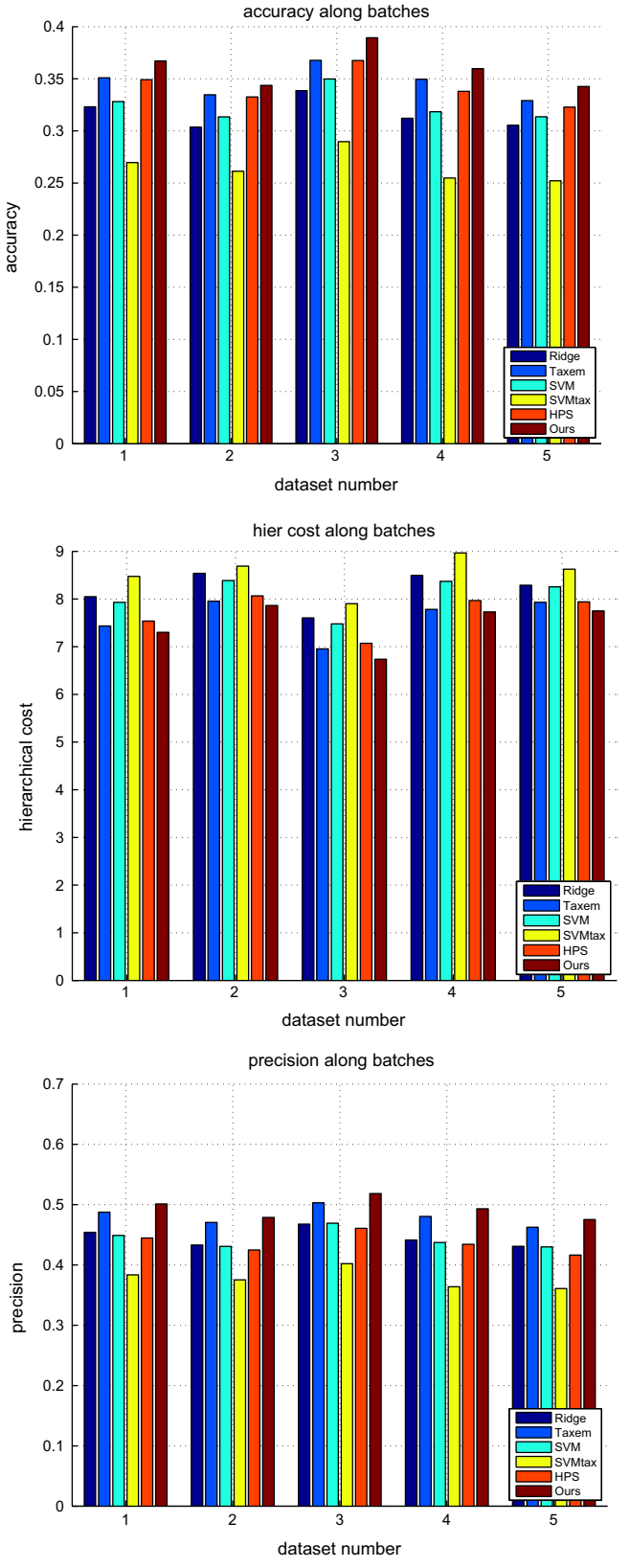

Fig. 4. The classification results of our method against others are plotted along with the dataset number for five ILSVRC100 datasets.

the HPS and actually the performance gap becomes bigger as the number of predictions k grows. It is well illustrated in the Fig. 6.

The training time results of all our datasets are summarized in Table 4. For the small and middle-scale datasets including Animals13, ILSVRC60, and ILSVRC100, we used a standard PC with

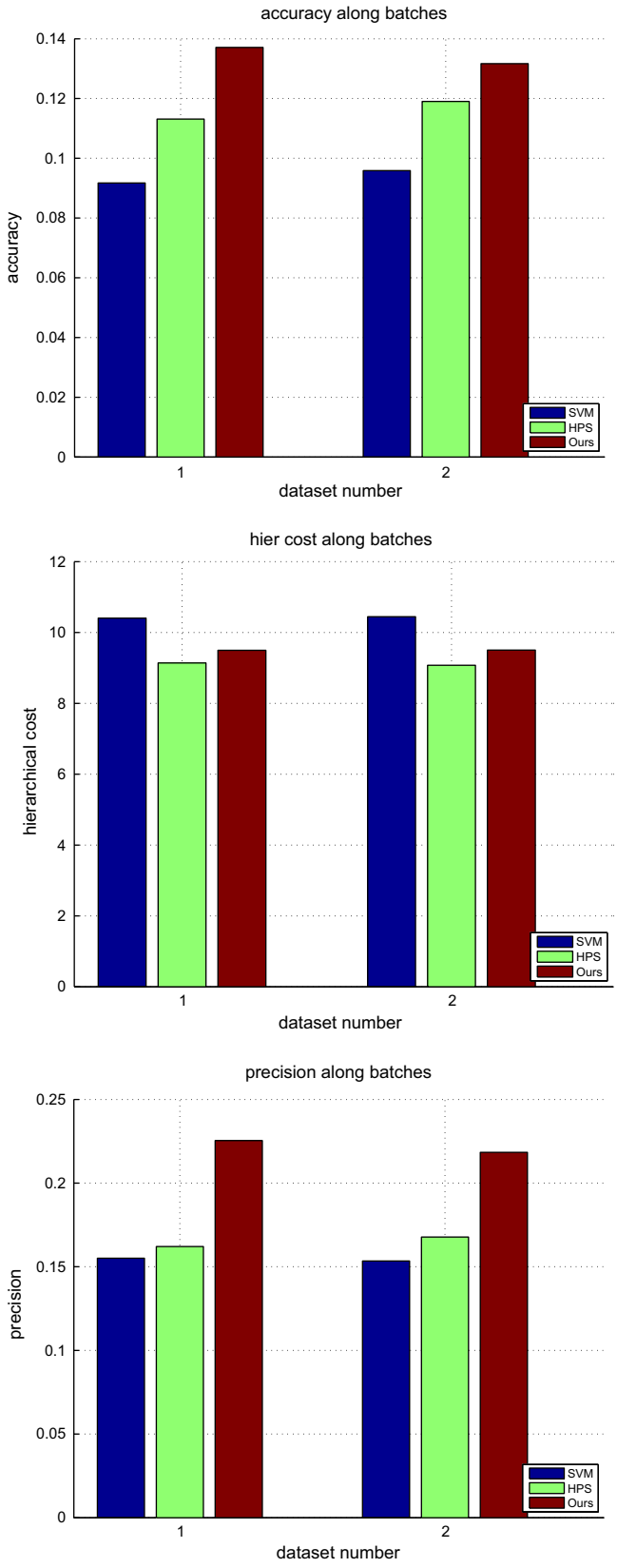

Fig. 5. The classification results of our method against others are plotted along with the dataset number for two ILSVRC1000 datasets.

3.30 GHz quadcore CPU and 16 GB RAM. And for the large-scale dataset, ILSVRC1000, a 2.40 GHz 12-core CPU machine with 1 TB RAM was utilized. The proposed method was implemented as a

Table 3
Performance comparison on the ILSVRC1000 dataset. Bold numbers indicate the best results.

Method	acc_1	acc_2	acc_3	$hier_1$	$hier_2$	$hier_3$	prec
SVM	9.38	13.49	16.28	10.43	8.79	7.93	15.43
HPS [22]	11.61	15.20	17.48	**9.11**	8.61	8.35	16.49
Ours	**13.44**	**19.63**	**24.00**	9.50	**8.12**	**7.34**	**22.19**

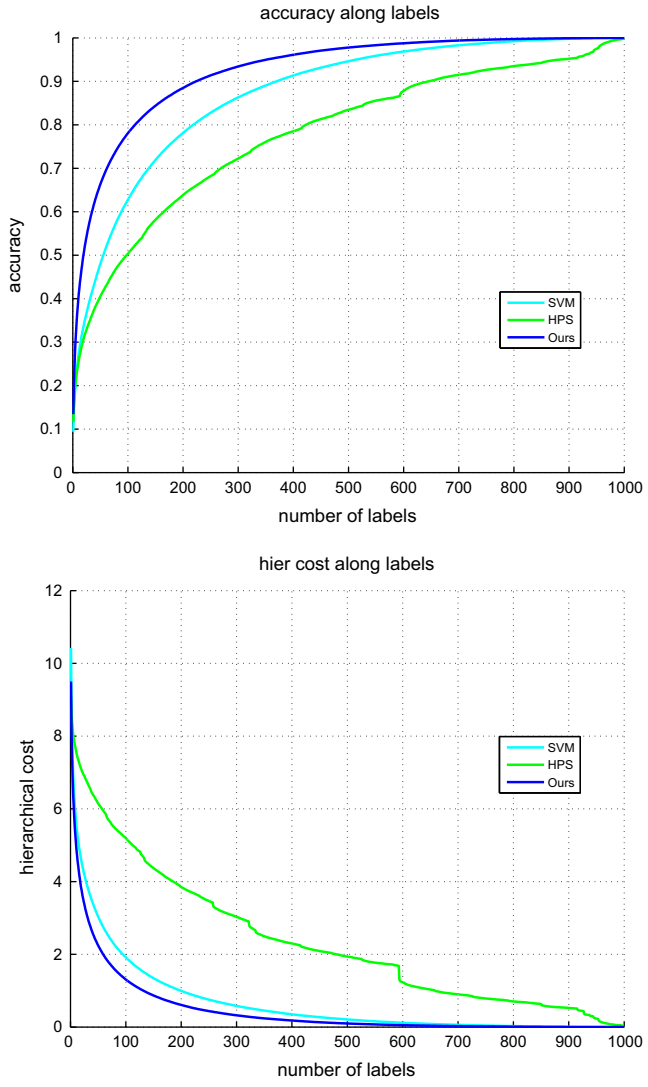

Fig. 6. The classification results of our method against others are plotted along with the number of predictions for the ILSVRC1000 datasets.

Table 4
Training time comparison on all datasets.

Method	Animals13	ILSVRC60	ILSVRC100	ILSVRC1000
Ridge	0.07 s	10.74 s	18.09 s	–
Taxem	1.55 s	48.93 min	5.75 h	–
SVM	1.13 s	27.77 s	56.03 s	25.71 min
SVMtax	2.07 s	89.71 s	5.22 min	–
SSVMtax	9.67 min	–	–	–
HPS	3.57 s	3.03 min	7.98 min	15.93 h
Ours	2.95 s	4.03 min	17.08 min	64.46 h

MATLAB code without any parallelization efforts. Although training our model is relatively slow compared with the other methods, it takes a reasonable time (approximately 64 h) even for the large-scale ILSVRC1000 dataset.

5. Conclusion

We presented an approach that can exploit prior knowledge of a category hierarchy to produce more semantic prediction for image classification. Experimental results showed that the proposed method achieved a convincing performance in terms of the hierarchical loss criterion. Moreover significant improvements for the flat measures, such as accuracy and precision, were also obtained. Our experiments do not include comparison result between our method and other state-of-the-art algorithms for flat image classification, such as sparse coding based approach [35]. The sparse coding based SVM can be easily integrated into our framework, because it can replace the plain SVM of the semantic embedding step in Section 3.1. Therefore, utilizing sparse coding to increase the performance of the proposed method will be our first future work. In our approach, the learned semantic similarity matrix has been used only to compare the input semantic vector with several category prototypes. Utilizing such a similarity function to compare any two semantic vectors for similar image retrieval problem will be our another future work.

Acknowledgments

This work was partly supported by the ICT R&D program of MSIP/IITP, Korea [14-824-09-006, Novel Computer Vision and Machine Learning Technology with the Ability to Predict and Forecast, and the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (No. 2009-0083495)].

References

[1] I. Biederman, Recognition-by-components: a theory of human image understanding, Psychol. Rev. 94 (1987) 115–147.
[2] J. Deng, A.C. Berg, K. Li, L. Fei-Fei, What does classifying more than 10,000 image categories tell us?, in: Proc. European Conf. on Computer Vision, 2010, pp. 71–84.
[3] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, 1998.
[4] V. Ferrari, A. Zisserman, Learning visual attributes, in: Proc. of Advances in Neural Information Processing Systems, 2007, pp. 433–440.
[5] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009, pp. 1778–1785.
[6] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009, pp. 951–958.
[7] L. Torresani, M. Szummer, A. Fitzgibbon, Efficient object category recognition using classemes, in: Proc. European Conf. on Computer Vision, 2010, pp. 776–789.
[8] Y. Wang, G. Mori, A discriminative latent model of object classes and attributes, in: Proc. European Conf. on Computer Vision, 2010, pp. 155–168.
[9] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Proc. of Advances in Neural Information Processing Systems, 2005, pp. 451–458.
[10] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[11] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking, J. Mach. Learn. Res. 11 (2010) 1109–1135.
[12] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[13] L. Cai, T. Hofmann, Hierarchical document categorization with support vector machines, in: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, 2004, pp. 78–87.
[14] A. Binder, K.-R. Mller, M. Kawanabe, On taxonomies for multi-class image categorization, Int. J. Comput. Vision (2011) 1–21.
[15] K.Q. Weinberger, O. Chapelle, Large margin taxonomy embedding for document categorization, in: Proc. of Advances in Neural Information Processing Systems, 2008, pp. 1737–1744.
[16] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 594–611.
[17] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Ann. Phys. 7694 (2007) 1–20.
[18] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, L. Fei-fei, ImageNet: a large-scale hierarchical image database, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[19] B. Taskar, C. Guestrin, D. Koller, Max-margin markov networks, in: Proc. of Advances in Neural Information Processing Systems, 2003, pp. 25–32.
[20] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, J. Mach. Learn. Res. 6 (2005) 1453–1484.
[21] M. Everingham, A. Zisserman, C.K.I. Williams, L. Van Gool, The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results, <http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf>, 2006.
[22] J. Deng, A. Berg, L. Fei Fei, Hierarchical semantic indexing for large scale image retrieval, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2011, pp. 785–792.
[23] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, 1999, pp. 61–74.
[24] M. Marszalek, C. Schmid, Semantic hierarchies for visual object recognition, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2007, pp. 1–7.
[25] M. Marszalek, C. Schmid, Constructing category hierarchies for visual recognition, in: Proc. European Conf. on Computer Vision, 2008, pp. 479–491.
[26] G. Griffin, P. Perona, Learning and using taxonomies for fast visual categorization, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[27] C.H. Lampert, M.B. Blaschko, A multiple kernel learning approach to joint multi-class object detection, in: Proceedings of the 30th DAGM Symposium on Pattern Recognition, 2008, pp. 31–40.
[28] A. Budanitsky, G. Hirst, Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures, in: Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics, 2001.
[29] E. Rosch, Principles of categorization, Cognition Categorization (1978) 27–48.
[30] L. Yang, Distance metric learning: a comprehensive survey, Technical report, Michigan State University, 2006.
[31] K.Q. Weinberger, L.K. Saul, Fast solvers and efficient implementations for distance metric learning, in: Proc. International Conf. on Machine Learning, 2008, pp. 1160–1167.
[32] <http://www.image-net.org/challenges/LSVRC/2011/>, 2011.
[33] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2010, pp. 3539–3546.
[34] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, <http://www.vlfeat.org/>, 2008.
[35] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.