Latent Task Adaptation with Large-scale Hierarchies

Yangqing Jia Trevor Darrell
UC Berkeley EECS & ICSI
{jiayq, trevor}@eecs.berkeley.edu

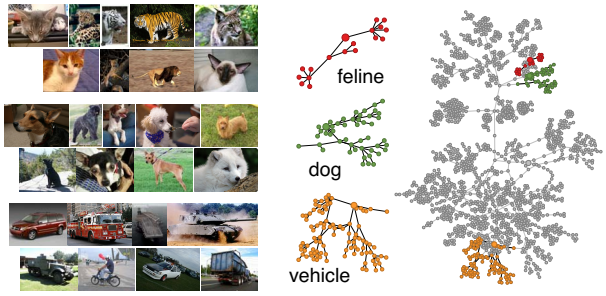
Abstract

Recent years have witnessed the success of large-scale image classification systems that are able to identify objects among thousands of possible labels. However, it is yet unclear how general classifiers such as ones trained on ImageNet can be optimally adapted to specific tasks, each of which only covers a semantically related subset of all the objects in the world. It is inefficient and suboptimal to retrain classifiers whenever a new task is given, and is inapplicable when tasks are not given explicitly, but implicitly specified as a set of image queries. In this paper we propose a novel probabilistic model that jointly identifies the underlying task and performs prediction with a linear-time probabilistic inference algorithm, given a set of query images from a latent task. We present efficient ways to estimate parameters for the model, and an open-source toolbox to train classifiers distributedly at a large scale. Empirical results based on the ImageNet data showed significant performance increase over several baseline algorithms.

1. Introduction

Recent years have witnessed a growing interest in object classification tasks involving specific sets of object categories, such as fine-grained object classification [6, 12] and home object recognition in visual robotics. Existing methods in the literature generally describe algorithms that are trained and tested on exactly the same task, *i.e.* we assume the training data and testing data share the same set of object labels. A dog breed classifier is trained and tested on dogs and a cat breed classifier done on cats, without the use of out-of-task images.

However, two observations may render this "one (multi-class) classifier per task" approach suboptimal. First, it's known that using images of related tasks is often beneficial to build a better model for the general visual world [18], which serves as a better regularization for the specific task as well. Second, object categories in the real world are often organized in, or at least well modeled by, a nested taxonomical hierarchy (*e.g.* Figure 1), with classification tasks corresponding to intermediate subtrees in this hierarchy, and recent efforts on the ImageNet challenge [2, 16, 21, 13] have leveraged the use of large-scale data to learn such information. While it is reasonable to train separate classifiers for specific tasks, this quickly becomes infeasible as there are a huge number of possible tasks - any subtree in the hierarchy may be a latent task requiring one to distinguish object categories under the subtree.





golden retriever (ice bear) **tabby cat** (dungeness crab) **garbage truck** (boathouse)

Figure 1: Top: Visualization of specific object classification tasks of interest in daily life, which are often subtrees in a large scale object taxonomy, *e.g.* the ImageNet hierarchy. Bottom: Adapting the ImageNet classifier allows us to perform accurate prediction (bold), while the original classifier prediction (in parentheses) suffers from a higher confusion.

Thus, it would be beneficial to have a system which learns a large number of object categories in the world, and which is able to quickly adapt to specific incoming classification tasks (subsets of all the object categories) once deployed. We are particularly interested in the scenario where tasks are not explicitly given, but implicitly specified with a set of query images, or a stream of query images in an online fashion. This has practical importance: for example, one may want to have a single mobile app that adapts to

plant recognition on a field trip after a few image queries, and that shifts to grocery recognitions when one stops by the grocery store. This is a new challenge beyond simple classification - one needs to discover the latent task using the context given by the queries, a problem that has not been tackled in previous classification problems.

To this end, we propose a novel probabilistic framework that generatively models a latent classification task and test time image queries, built on top of the success of classical, large-scale one-vs-all classifiers. The framework allows efficient inference to be carried out to both identify the latent task from query images and adapt the classifier for the specific task. We instantiate an experimental testbed with the benchmark ImageNet large scale visual recognition challenge (ILSVRC) data using a series of latent fine-grained tasks sampled from the taxonomy, and show promising performance over conventional classification methods.

The contribution of this paper is two-fold. We show that with a large-scale image source where object labels are organized in a taxonomical structure, it is almost always beneficial to learn the classifier on the whole dataset even for tasks involving only subtrees of the overall taxonomy. More importantly, we examine a novel task adaptation paradigm that is beyond recognizing individual images, and propose an algorithm to easily adapt a general classifier to unknown latent tasks during testing time, yielding a significant performance boost.

Finally, our pipeline will be made open-source, including a toolbox for distributed classifier learning with quasi-Newton stochastic algorithms [5], which allows one to train large-scale classifiers (such as ILSVRC) without the need of huge clusters or sophisticated infrastructure support.

2. Related Work

The problem of task adaptation is analogous to, but essentially distinctive from domain adaptation [19, 14]. While domain adaptation aims to model the *perceptual* difference of the training and testing images from the same labels, task adaptation focuses on modeling the *conceptual* difference: different label spaces during training and testing. Additionally, as one is often able to use large amounts of data during training, we assume that the testing tasks involve subsets of labels encountered during training time.

Predicting the intermediate concept in a hierarchy with a set of examples has been discussed in psychology [26, 1, 23]. These methods often make a simplified assumption that labels (leaf nodes in the hierarchy) are given for the input images. We believe our paper is the first to connect such psychological study with computer vision research by directly taking perceptual inputs, allowing one to perform generalization with images of unknown category.

There are several algorithms in image classification that use label hierarchy or structured regularizations to learn better classifiers [20, 10, 8], or to leverage the accuracy and information gain from classifiers [4]. These methods still assume an identical label space for training and testing. The ultimate goal thus remains to be better accuracy on classifying individual images, not to adapt to different tasks during testing time by utilizing contextual information. Better classifiers presented in these papers could, of course, be incorporated in our model to improve the end-to-end performance of task adaptation.

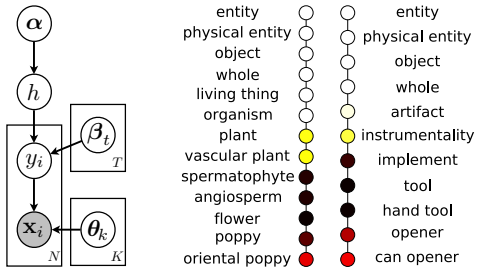


Figure 2: Left: the generative model for the latent task and corresponding query images. Right: the prior probabilities of the latent tasks from psychological study, along the path leading to the synsets *oriental poppy* and *can opener* respectively, with darker color indicating higher probability.

Finally, it is known that context information, such as scene context and co-occurring context within a image, could be adopted for better detection [25] or scene understanding [15]. In this paper we utilize a novel type of context - task context - that is implied by a semantically related group of images.

3. A Generative Model for Task Adaptation

Formally, we define a classification *task* to be a subset of all the possible object labels that are semantically related (such as all breeds of dogs in ImageNet). During testing time, a number of query images are randomly sampled from the labels belonging to a task, and the learning algorithm needs to give predictions on these images. In this section we propose a probabilistic framework that models the generation of latent tasks and the test time query images.

As stated in the previous section, we are interested in the scenario when the task is *latent*, *i.e.* only implicitly specified by the query images. We introduce two key components for modeling the generative process of query images: a latent task space that defines possible tasks and their probability, and a procedure to sample query images given a specific latent task. Specifically, we propose the graphical model in Figure 2 which generates a set of N query images when given T possible tasks and K object categories:

1. Sample a latent task h from the task priors $P(h)$ with hyperparameter $\boldsymbol{\alpha}$;

2. For the N query images:
   (a) Sample an object category y_i from the conditional probability $P(y_i|h;\boldsymbol{\beta}_h)$;
   (b) Sample a query image from category y_i with $P(x_i|y_i;\boldsymbol{\theta}_{y_i})$.

We will elaborate each component in the subsections below.

3.1. Latent Task Space

Determining the tasks and their prior is a high-level problem that essentially asks "what scenarios do people encounter in daily life, and how often do they appear". To this end, we take advantage of the existing research in cognitive science to construct the latent task space and the prior distribution.

For the structure of the latent task space, we adopt the WordNet hierarchy [7], which models the semantic relations in a psychologically justified tree structure [17]. The use of WordNet in cognitive science has shown promising results in identifying latent concepts (semantically related sets from the universe of objects) for human concept learning [1, 24]. In our work, we follow the existing classification protocols [2] by considering the set of leaf nodes in the tree as the object labels that we need to classify images into. Every intermediate node then serves as a possible task, which requires the computer to identify object labels, *i.e.* leaf nodes under the subtree rooted at the node, *e.g.* "what animal this is", or "what breed of dog this is".

In this paper we are mainly interested in modeling the frequency of various tasks in a general, large-scale setting[1]. Prior research on psychology and Bayesian generalization [22, 23] have shown that people favor basic-level concepts, which could be well modeled by a Erlang prior with respect to the size $|h|$ of each latent task, defined as the number of leaf nodes in the subtree rooted at the task:

$$P(h) = \alpha_h \propto (|h|/\sigma^2)\exp(-|h|/\sigma), \tag{1}$$

which favors medium-sized tasks corresponding to basic level concepts.

Figure 2 shows two such examples along the paths to the ImageNet synsets *can opener* and *oriental poppy*. It could be observed that basic level tasks have higher probability than overly general tasks such as "entity", which means that our bias is for the computer to assist us in more specific tasks, *e.g.* classifying flowers on a field trip or tools in a Robotics scenario; this is more desired than vaguely asking "what entity is this".

3.2. Generating Query Images

Given a task, we assume that the query images we encounter are then randomly sampled from the object classes that belong to the given task. For each query image, we first sample the object class label from the set of possible labels that belong to the task. The conditional probability $P(y_i|h)$ follows from assuming strong sampling [23]: labels are generated uniformly at random using the corresponding parameter $\boldsymbol{\beta}_h$ as follows:

$$P(y_i|h) = \beta_{hy_i} = \begin{cases} 1/|h|, & \text{if task } h \text{ contains label } y_i \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $|h|$ is the size of the task - the number of leaf node classes in the task. The size principle plays a critical role in inferring the latent task, as larger tasks will generate lower probabilities for each individual object class. Thus, when we observe a Dalmatian, a corgi and a Shih-Tzu, the latent task "dog" is more probable than task "animal" since the former yields higher conditional probability for the detailed dog breeds.

To generate an actual image $\mathbf{x}_i$ from a given class label y_i, it is relatively difficult to fully model the conditional probability $P(\mathbf{x}_i|y_i)$ to the pixel level of the images. Thus, we use a mixed approach by having a classifier trained on all the leaf node objects, and obtain the classifier prediction

$$f(\mathbf{x}_i) = \operatorname{argmax}_j \quad \boldsymbol{\theta}_j^\top \mathbf{x}_i, \tag{3}$$

where we simplify the notation by using $\mathbf{x}_i$ as both the image and the feature extracted from it, and assuming that a linear classifier with parameter $\{\boldsymbol{\theta}_j\}_{j=1}^K$ is used. The conditional probability is then defined as

$$P(\mathbf{x}_i|y_i) = C_{y_i f(\mathbf{x}_i)}, \tag{4}$$

where $\mathbf{C}$ is the confusion matrix of the classifier, and C_{ij} is the probability that an image of object class i is classified as class j.

With the probabilistic model given as Figure 2, given a set of testing images $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, our goal is to jointly identify the hidden task h and the hidden labels $\mathcal{Y} = \{y_1, y_2, \cdots, y_N\}$ that maximizes the posterior probability

$$(\hat{h}, \hat{\mathcal{Y}}) = \operatorname*{argmax}_{h,\mathcal{Y}} P(h, \mathcal{Y}|\mathcal{X}). \tag{5}$$

We will discuss in the next section how the various parameters, especially the parameters $\boldsymbol{\theta}$ for the classifiers and the confusion matrix $\mathbf{C}$, can be estimated from training data, and how to carry out efficient inference to find the solution to Eqn. 5.

4. Efficient Learning and Inference

The probabilistic model involves multiple parameters to be estimated and nested hidden variables during the inference phase. In this section, we present a novel approach to estimate the confusion matrix for the classifier, and a linear-time inference algorithm that jointly identifies the latent task and predictions for individual images.

[1] We note that in specific settings, the task prior probabilities for each latent task could also be learned via human behavioral study that evaluates the popularity of various tasks.

4.1. Confusion Matrix Estimation with One-step Unlearning

Given a classifier, evaluating its behavior (including accuracy and confusion matrix) is often tackled with two approaches: using cross-validation or using a held-out validation dataset. In our case, we note that both methods have significant shortcomings. Cross-validation requires retraining the classifiers multiple rounds, which may lead to high re-training costs. A held-out validation dataset usually estimates the accuracy well, but not for the confusion matrix $\mathbf{C}$ due to insufficient number of validation images. For example, the ILSVRC challenge has only 50K validation images versus 1 million confusion matrix entries, leading to a large number of incorrect zero entries in the estimated confusion matrix (see supplementary material).

Instead of these methods, we propose to approximate its leave-one-out (LOO) error on the training data with a simple gradient descent step to "unlearn" each image to estimate its LOO prediction, similar to the early unlearning ideas [9] proposed for neural networks. We will focus on the use of multinomial logistic regression, which minimizes $\mathcal{L}(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_2^2 - \sum_{i=1}^{M} \mathbf{t}_i \log \mathbf{u}_i$, where $\mathbf{t}_i$ is a 0-1 indicator vector where only the y_i-th element is 1, and $\mathbf{u}_i$ is the softmax of the linear outputs $u_{ij} = \exp(\boldsymbol{\theta}_j^\top \mathbf{x}_i) / \sum_{j'=1}^{K} \exp(\boldsymbol{\theta}_{j'}^\top \mathbf{x}_i)$, with $\mathbf{x}_i$ being the feature for the i-th training image.

Specifically, given the trained classifier parameters $\boldsymbol{\theta}$, it is safe to assume that the gradient $\mathbf{g}(\boldsymbol{\theta}) = \mathbf{0}$. Thus, the gradient for the logistic regression loss when removing a training image $\mathbf{x}_i$ could be computed simply as $\mathbf{g}_{\setminus \mathbf{x}_i}(\boldsymbol{\theta}) = (\mathbf{u}_i - \mathbf{t}_i)\mathbf{x}_i^\top$. Given the Hessian matrix $\mathbf{H}$ at $\boldsymbol{\theta}$, one can perform one-step quasi-Newton least-square update as[2]

$$\boldsymbol{\theta}_{\setminus \mathbf{x}_i} = \boldsymbol{\theta} - \rho' \mathbf{H}^{+} \mathbf{g}_{\setminus \mathbf{x}_i}. \tag{6}$$

Note that we put an additional step size ρ' instead of $\rho' = 1$ as would be the case for exact least squares. We set ρ' to the value that yields the same LOO approximation accuracy as the validation accuracy. We use the new parameter $\boldsymbol{\theta}_{\setminus \mathbf{x}_i}$ to perform prediction on $\mathbf{x}_i$ as if $\mathbf{x}_i$ has been left out during training, and accumulate the approximated LOO results to obtain the confusion matrix. We then applied Kneser-Ney [11] smoothing on the confusion matrix for a smoothed estimation.

4.2. Linear Time MAP Inference

A conventional way to do probabilistic inference with nested latent variables is to use variational inference or Gibbs sampling to find a lower bound of the posterior probability. This, however, may involve multiple iterations over the hidden variables and may be slow. We show that when the latent task space is organized in a DAG structure, the exact MAP solution (Eqn. (5)) could be found with an efficient dynamic programming algorithm that has complexity linear to the number of possible tasks.

We first note that the logarithm of posterior probability in Eqn. 5 could be expanded as

$$\log P(h, \mathcal{Y}|\mathcal{X}) \propto \log \alpha_h + \sum\nolimits_{i=1}^{N} \log(\beta_{h y_i} C_{y_i f(\mathbf{x}_i)}). \tag{7}$$

Notice that the size constraint defining the latent task space gives us $\beta_{h y_i} = \frac{1}{|h|} I(y_i \in h)$, the equation above could further be written as

$$\log \alpha_h - N \log |h| + \sum\nolimits_{i=1}^{n} (\log C_{y_i f(\mathbf{x}_i)} + \log I(y_i \in h)),$$

where one can observe that h and $\mathcal{Y}$ decouples except for the $I(y_i \in h)$ term. As the latent tasks are organized as a tree-based hierarchy, we can define auxiliary functions

$$q_i(h) = \max_{\mathcal{Y}} \left[\log C_{y_i f(\mathbf{x}_i)} + \log I(y_i \in h)\right], \tag{8}$$

which could be computed recursively as

$$q_i(h) = \max\nolimits_{\{h' \in child(h)\}} q_i(h'), \tag{9}$$

where $child(h)$ is the set of children of h in the tree. Finally, the latent task could be estimated as

$$\hat{h} = \operatorname*{argmax}_{h} \left[\log(\alpha_h) - N \log |h| + \gamma \sum\nolimits_{i=1}^{N} q_i(h)\right], \tag{10}$$

and the corresponding $\hat{y}_i$s could be identified by taking the argmax of the corresponding $q_i(h)$.

We note that we added a hyperparameter γ in the equation above. In practice, simply finding the MAP solution (using $\gamma = 1$) often involves a task that is smaller than the ground truth, as there are two ways to explain the predicted labels: assuming correct prediction and a task of larger size, or assuming wrong prediction and a task of smaller size. The latter is preferred by the size principle, especially for classes with low classification accuracy. We found it beneficial to explicitly add a weight term that favors the classifier outputs using $\gamma > 1$ learned on validation data.

In general, our dynamic programming method runs in $O(TNb)$ time where T is the number of tasks, N is the number of query images, and b is the branching factor of the tree (usually a small constant factor). This complexity is linear to the number of testing images and to the number of latent tasks, and is usually negligible compared to the basic classification algorithm, which runs $O(KND)$ time where K is the number of classes and D is the feature dimension (usually very large).

[2] In practice we used the accumulated matrix $\mathbf{H}$ obtained from Adagrad [5] as a good approximation of the Hessian matrix. See supplementary material for details. We tested the Adagrad $\mathbf{H}$ matrix and the exact Hessian computed at $\boldsymbol{\Theta}$, and found the former to actually perform better, possibly due to its overall robustness.

Finally, one may prefer an online algorithm that could take new images as a stream, performing classification sequentially while discovering the latent task on the fly. We note that our method could be easily adapted to this end. Specifically, $q_i(h)$ serves as the sufficient statistics for the task discovery, and we only need to keep record of the accumulated auxiliary function values seen so far as

$$q_{:n}(h) = \sum_{i=1}^{n-1} q_i(h) \tag{11}$$

for the n-th image for each task candidate h. This allows us to perform online classification with $O(M)$ memory without storing the full history of images.

5. Distributed Implementation Details

Recent image classification tasks often involve large amounts of images, making the training of classifiers increasingly difficult. To address this issue, we have developed a distributed, stochastic optimization toolbox to train large-scale image classifiers. In particular, we used the minibatch approach to perform stochastic gradient descent updates, and utilized the Adagrad [5] algorithm to achieve quasi-Newton performance by accumulating the statistics of the per-iteration gradient estimations, a mechanism shown to work particularly well with vision tasks [3].

We further took advantage of parallel computing by distributing the data as well as the gradient computation over multiple machines. As datasets are often too large to fit into the memory of even a medium-sized cluster, we only keep the minibatch in memory at each iteration, with a background process pre-fetching the next minibatch from disk during the computation of the current one, which enables us to perform efficient optimization with arbitrarily large datasets.

For the image features, we followed the pipeline in [16] to obtain over-complete features for the images. Specifically, we extracted dense local SIFT features, and used Local Coordinate Coding (LCC) to perform encoding with a dictionary of size 16K. The encoded features were then max pooled over 10 spatial bins: the whole image and the 3×3 regular grid. This yielded 160K feature dimensions per image, and a total of about 1.5TB for the training data in double precision format. The overall performance is 41.33% top-1 accuracy and a 61.91% top-5 accuracy on the validation data, and 41.28% and 61.69% respectively on the testing data. For the computation time, training with our toolbox took only about 24 hours with 10 commodity computers connected on a LAN. Our toolkit is implemented in Python and will be publicly available[3], and we refer to the supplementary materials for more technical details.

[3] http://www.eecs.berkeley.edu/~jiayq/

6. Experiments

We conduct our experiment on the ILSVRC 2010 dataset [2], where both validation and test data are available. For all the experiments, we learn the parameters of the model on the training and validation data, and report the performance on the test images.

We note that more comprehensive features and better classification pipelines may lead to better 1-vs-all accuracy on ImageNet, but it is not the main goal of the paper, as we focus on the adaptation on top of the base classifiers. Recent efforts on learning better classifiers, such as the ones presented in [21, 13] could be seamlessly incorporated into our learning framework for general performance increases.

6.1. Estimating the Confusion Matrix

As stated in Section 4, an good estimation of the confusion matrix $\mathbf{C}$ is crucial for the probabilistic inference. We evaluate the quality of different estimations using the test data: for each testing pair $(y, \hat{y})$, where $\hat{y}$ is the classifier output, its probability is given by the confusion matrix entry $C_{y\hat{y}}$. The perplexity measure [11] then evaluates how "surprising" the confusion matrix sees the testing data results (a smaller value indicates a better fit):

$$perp = \text{Power}\Big(2, \big(\sum_{i=1}^{N_{te}} \log_2 C_{y_i \hat{y}_i}\big)/N_{te}\Big),$$

where N_{te} is the number of testing images. Overall, we obtained a perplexity of 46.27 using our unlearning algorithm, while the validation data gave a value of 68.36 and the training data (without unlearning) gave 94.69, both worse than our unlearning algorithm. We refer to the supplementary material for a more complete analysis of the performance of different methods.

6.2. Adapting Classifiers with Known Tasks

An important question to ask is, even if we are allowed to retrain task-specific classifiers, do we want to do the retraining? We first analyze the benefits of retraining versus our adaptation method. To this end, we specify 5 subtrees from the ILSVRC hierarchy: `building`, `dogs`, `feline` (the superset of cats), `home appliance`, and `vehicle`, the subcategories of which are often of interest. Figure 1 visualizes the corresponding subtrees for dog, feline and vehicles respectively. We explicitly trained classifiers on these three subtrees only, and compared the retrained accuracy against our adapted classifier with the given task. We also test the naive baseline that uses the raw 1000 class predictions, and the forced choice baseline (FC) which simply selects the class under the task that has the largest output from the original classifiers. Table 1 summarizes the performance of the algorithms.

It is worth pointing out that retraining the classifiers for the specific tasks does *not* help improve the classification

Task	Naive	Retraining	FC	Ours
building	55.48	78.67	81.48	**82.19**
dog	35.37	39.94	42.95	**43.76**
feline	47.13	61.07	62.67	**63.54**
home app	50.78	67.30	69.26	**70.52**
vehicle	55.62	61.43	**63.41**	63.28

Table 1: Classification accuracy on given tasks (subtrees) of the whole ILSVRC data. See subsection 6.2 for details.

Method	query size=5		query size=100	
	$s(h,\hat{h})$	Accuracy	$s(h,\hat{h})$	Accuracy
Naive	1.54	42.75	1.50	42.68
Proto	8.14	43.16	60.39	50.28
Hist	22.21	44.84	96.61	59.87
Hedging	39.12	44.81	50.34	51.83
Ours	**84.43**	**65.89**	**99.37**	**70.70**
Oracle	100.0	70.36	100.0	70.88

Table 2: The average task overlap score and the average accuracy for the algorithms, under query sizes 5 and 100 respectively. All numbers are in percentage. The last row provides the oracle performance in which the ground truth task is given.

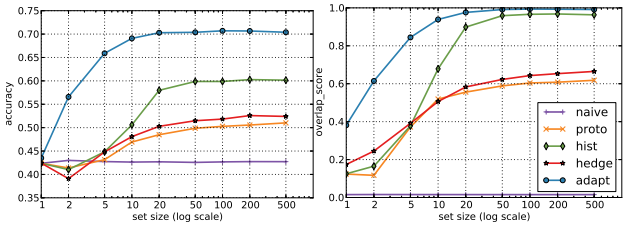


Figure 3: Classification accuracy (left) and the task overlap score (right) with different query set sizes for our method and the baselines.

accuracy, although retraining requires additional nontrivial computation cost. On contrary, it is always helpful to use out-of-task data to train a larger classifier and then take the subset with forced choice. One possible explanation is that this gives us more information about the general image statistics (similar to a better regularization term). Our method further benefits from the statistics from all the classifiers (for in-task and out-of-task classes) in the proposed probabilistic framework to achieve the best adapted accuracy in most cases (only slightly worse than the FC baseline on vehicle).

6.3. Joint Task Discovery and Classification

We next analyze the performance when we have the classifier trained on the whole ILSVRC data, and adapt it to an unknown task that is defined by a set of query images. The forced choice option is not available in this case as we do not know the latent task beforehand, and one has to use the semantic relationships between the query images to infer the latent task.

To sample the latent tasks, we used the Erlang prior defined in Section 3 from the ImageNet Tree excluding leaf nodes (as leaf nodes would contain only 1 label). We then randomly sampled N query images from the subtree of the sampled task. All query images were randomly selected from the test images of ILSVRC and had not been seen by the classifier training. We varied the value N to assess the quality of task discovery under different set sizes. For each query image size N, we created 10,000 independent tasks and reported the average performance here.

To evaluate the goodness of the inferred latent task and the accuracy, we compute the overlap between the ground truth task h and the predicted task $\hat{h}$ as

$$s(h,\hat{h}) = |h \cap \hat{h}|/|h \cup \hat{h}| \times 100\%, \tag{12}$$

where $\cap$ and $\cup$ are the intersection and union operations on sets, and $|\cdot|$ denotes the size of a set. For each task, we then compute the accuracy with the predicted labels $\hat{\mathcal{Y}}$ in the standard classification evaluation way. We then report the averaged overlap score and averaged per-task prediction accuracy.

To the best of our knowledge there is no published classification algorithm that is able to identify the latent task, *i.e.* the intermediate node in the taxonomy hierarchy, given a *set* of query images. Thus, we compare our algorithm against the following baselines that are natural extensions from conventional classification methods:

- **Naive approach**: simply taking the class with the highest prediction score from all the ILSVRC classes.
- **Prototype approach**: we use the conditional probability $p(y|h)$ as a vector for each task h, and use the task that yields the smallest average distance to each query image (using the classifier outputs) as the predicted latent task. Classification is then performed under this predicted task.
- **Histogram approach**: similar to the prototype approach, but instead of computing pairwise distances to individual query images, we select the task h that yields the smallest χ^2 distance between $p(y|h)$ and the histogram of predictions averaged over all queries.
- **Hedging approach**: we extend the hedging idea [4] to handle sets of query images. Specifically, we find the intermediate node that maximizes the information gain while maintaining an overall accuracy above a threshold ϵ over the set of query images. The corresponding task is then chosen as the predicted latent task. We tune the threshold ϵ on the validation data so that the averaged per-task accuracy is maximized.

We also test an oracle model, in which we explicitly tell the classifier the latent task and perform classification on the

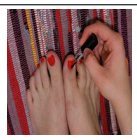
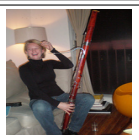
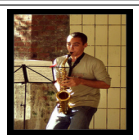
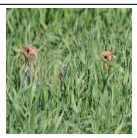
Task: **kitchen app**						Predicted task:
Label	ice maker	espresso maker	primus stove	Dutch oven	ice maker	entity
Ours	electric range	espresso maker	primus stove	Dutch oven	ice maker	artifact
Baseline	bookcase	web site	carpenter's kit	snail	scanner	artifact
						consumer goods
						kitchen app
Task: **toiletry**						Predicted task:
Label	lipstick	face powder	nail polish	lotion	hair spray	entity
Ours	lipstick	face powder	nail polish	lotion	hair spray	entity
Baseline	toothbrush	dune	bath towel	vending machine	military uniform	instrumentality
						toiletry
Task: **woodwind**						Predicted task:
Label	bassoon	flute	sax	oboe	sax	entity
Ours	bassoon	bassoon	sax	oboe	sax	artifact
Baseline	harp	prison	sax	fountain pen	turban	artifact
						device
						reed instrument
Task: **game**						Predicted task:
Label	ptarmigan	partridge	pheasant	black grouse	quail	entity
Ours	ptarmigan	partridge	pheasant	black grouse	black grouse	living thing
Baseline	giant panda	orchid	Komodo dragon	Border collie	Newfoundland	entity
						chordate
						game

Figure 4: Exemplar classification results where incorrect labels are predicted by the base classifiers, but are corrected by our method that benefits from identifying the latent task. Each row shows 5 images from a latent task, and on the right we give the predicted task by different algorithms, ordered and colored as naive, proto, hist, hedge, and adapt. The ground truth label, our prediction and the original classifier's output are provided for each image.

subset of labels with the task ground truth. This serves as an upper bound of all methods above, and helps us understand how well different algorithms perform. Regarding the classifier outputs, we used the soft output from the logistic regression for our method, and choose between the soft output and 0-1 hard output for the baseline methods, reporting the better performance of the two here.

Table 2 summarizes the performance of the methods above with a small query set size (5 images) and a relatively large size (100 image). Further, Figure 3 shows the performance when we vary the size from 1 to 500. It could be observed that when we have a reasonable amount of testing queries, identifying the latent task leads to a significant performance gain than the baseline method that does classification against all possible labels, with an increase of near 30% percent. Even with a small query size (such as 5), the performance gain is already noticably high, indicating the ability of the algorithm to perform task adaptation with very few images from the latent task.

6.4. Online Evaluation

Our final evaluation tests the performance of the proposed method in an online fashion - when images of an unknown task come as a streaming sequence. Intuitively, our algorithm obtains better information about the unknown task as new images arrive, which would in turn increase the classification accuracy. We test such conjecture by evaluating the averaged accuracy of the n-th image, over multiple independent test query sequences that are generated in the same way as described in the previous subsection.

Figure 5 shows the average accuracy of the n-th query image, as well as the overlap between the identified task so far and the ground truth task. With the joint probabilistic inference, we obtain a significant performance increase after only a few images. This has particular practical interest, as one may want the computer to quickly adapt to a new task

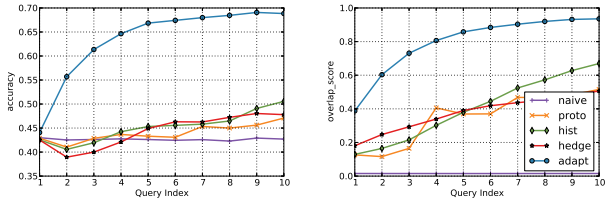


Figure 5: Classification accuracy (left) and task overlap score (right) of our online algorithm against baselines. See subsection 6.4 for details.

/ environment with only a small number of queries. It is worth pointing out that with heuristic task estimation methods (see the baselines in Figure 5 left), one may incorrectly assert the latent task, which then hurts classification performance for the first few query images.

7. Conclusion

We addressed a novel challenge when the classification problem involves latent tasks corresponding to semantically related subsets of all the objects in the world. We proposed a novel framework that is able to adapt to latent tasks to achieve a significant performance gain given a relatively small set of query images. We hope our efficient learning algorithms and the distributed toolbox that we will release will significantly contribute to the research of computer vision with large-scale data.

Acknowledgement This research is funded by NSF IIS-1116411 and IIS-1212928, DARPA's Minds Eye and MSEE programs, and Toyota Corporation.

References

[1] J. T. Abbott, J. L. Austerweil, and T. L. Griffiths. Constructing a hypothesis space from the web for large-scale bayesian word learning. In *Annu. Conf. Cog. Sci. Soc.*, 2012. 2, 3

[2] A. Berg, J. Deng, and L. Fei-Fei. ILSVRC 2010. http://www.image-net.org/challenges/LSVRC/2010/, 2008. 1, 3, 5

[3] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng. Large scale distributed deep networks. In *NIPS*, 2012. 5

[4] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*, 2012. 2, 6

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2010. 2, 4, 5

[6] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 1

[7] C. Fellbaum. Wordnet. *Theory and Applications of Ontology: Computer Applications*, pages 231–243, 2010. 3

[8] T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*. IEEE, 2011. 2

[9] L. K. Hansen and J. Larsen. Linear unlearning for cross-validation. *Advances in Computational Mathematics*, 5(1):269–280, 1996. 4

[10] Z. Harchaoui, M. Douze, M. Paulin, M. Dudik, and J. Malick. Large-scale image classification with trace-norm regularization. In *CVPR*, 2012. 2

[11] D. Jurafsky and J. H. Martin. *Speech & Language Processing*. Pearson Prentice Hall, 2000. 4, 5

[12] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR FGVC workshop*, 2011. 1

[13] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 5

[14] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 2

[15] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009. 2

[16] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: fast feature extraction and svm training. In *CVPR*, 2011. 1, 5

[17] E. M. Markman. *Categorization and naming in children: Problems of induction*. MIT Press, 1991. 3

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007. 1

[19] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2

[20] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 2

[21] J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *CVPR*, 2011. 1, 5

[22] R. N. Shepard et al. Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323, 1987. 3

[23] J. B. Tenenbaum, T. L. Griffiths, et al. Generalization, similarity, and bayesian inference. *Behavioral and brain sciences*, 24(4):629–640, 2001. 2, 3

[24] J. B. Tenenbaum, T. L. Griffiths, C. Kemp, et al. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006. 3

[25] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 2

[26] F. Xu, J. B. Tenenbaum, et al. Word learning as bayesian inference. *Psychological review*, 114(2):245, 2007. 2