Knowledge Transfer with Interactive Learning of Semantic Relationships

Jonghyun Choi
University of Maryland
College Park, MD
jhchoi@umiacs.umd.edu

Sung Ju Hwang
UNIST
Ulsan, Korea
sjhwang@unist.ac.kr

Leonid Sigal
Disney Research
Pittsburgh, PA
lsigal@disneyresearch.com

Larry S. Davis
University of Maryland
College Park, MD
lsd@umiacs.umd.edu

Abstract

We propose a novel learning framework for object categorization with interactive semantic feedback. In this framework, a discriminative categorization model improves through human-guided iterative semantic feedbacks. Specifically, the model identifies the most helpful relational semantic queries to discriminatively refine the model. The user feedback on whether the relationship is semantically valid or not is incorporated back into the model, in the form of regularization, and the process iterates. We validate the proposed model in a few-shot multi-class classification scenario, where we measure classification performance on a set of 'target' classes, with few training instances, by leveraging and transferring knowledge from 'anchor' classes, that contain larger set of labeled instances.

Introduction

Semantic information has been exploited extensively in recent years to improve object category recognition accuracy since object categories are essentially semantic entities that are human-defined. Various types of semantic sources have been exploited such as attributes (Hwang, Sha, and Grauman 2011; Akata et al. 2013), taxonomies (Weinberger and Chapelle 2008; Zhou, Xiao, and Wu 2011), and analogies (Hwang, Grauman, and Sha 2013), as auxiliary information to aid categorization. These methods often require a large knowledge base. However, construction of such knowledge bases could be expensive as it takes human effort to obtain, and such knowledge base might be especially challenging to obtain for highly specific sets of object categories, *e.g.*, recognizing the specific year/model of a vehicle, or cartoon characters from animation database.

Further, not all knowledge is equally useful in the discriminative classification sense. For example, knowing that an *apple* is more similar to a *pear* than a *dragon*, while semantically meaningful, may not be helpful in distinguishing *apple* from other fruits. Thus, finding and using relevant semantic information (*e.g.*, that an *apple* is more similar to a *pear* than a *melon*) to enhance recognition accuracy is challenging. In addition, it is difficult to identify an appropriate vocabulary of semantic information without prior knowledge about the object category itself and/or other categories that are potential confusers. This is evident from the game of 20-questions. Every question asked (and answered) has a significant effect on the distribution of questions a player may ask next. It is clear that if a player needed to ask all the questions at once, upfront, he may need considerably more than 20 questions to identify the object in question.

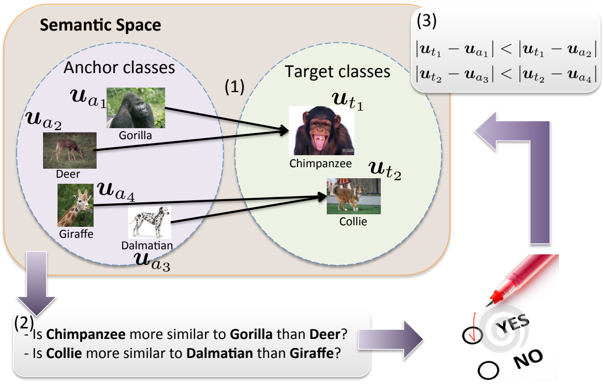
Semantic Space
Anchor classes
Target classes
(1)
$\boldsymbol{u}_{a_1}$
Gorilla
$\boldsymbol{u}_{a_2}$
Deer
$\boldsymbol{u}_{a_4}$
Giraffe
Dalmatian
$\boldsymbol{u}_{a_3}$
$\boldsymbol{u}_{t_1}$
Chimpanzee
$\boldsymbol{u}_{t_2}$
Collie
(3)
$|\boldsymbol{u}_{t_1} - \boldsymbol{u}_{a_1}| < |\boldsymbol{u}_{t_1} - \boldsymbol{u}_{a_2}|$
$|\boldsymbol{u}_{t_2} - \boldsymbol{u}_{a_3}| < |\boldsymbol{u}_{t_2} - \boldsymbol{u}_{a_4}|$
(2)
- Is **Chimpanzee** more similar to **Gorilla** than **Deer**?
- Is **Collie** more similar to **Dalmatian** than **Giraffe**?
YES
NO

Figure 1: **Approach Overview.** Our model is a discriminative manifold with embedded semantic entities. The discriminative categorization model is refined by iteratively generating semantic questions and obtaining user feedback. Thumbnail images denote category prototypes in the embedding space.

To address such challenges, we propose a method to obtain and leverage a focused set of semantic queries by examining a discriminatively learned model for object categorization in an interactive learning framework. Starting from a base model with no semantic information, we iteratively improve it by generating semantic queries for human(s) to answer, then in turn update the existing model with feedback. Such an interactive learning system effectively transfers knowledge from *anchor* categories, that are well learned, to *target* categories that have very few labeled training samples.

Figure 1 shows the overview of our approach. The categories are partitioned into two sets: *Anchor* classes, that have reasonable number of samples per class, and *Target* classes, that have few labeled instances, and to which the semantic knowledge is transferred. From a semantic embedding space in which both *Anchor* and *Target* samples are embedded, we detect relational hypotheses based on classification confusion among target and anchor classes. The approach consists

This work was done during J. Choi's internship at Disney Research. J. Choi is now at Comcast Labs, DC.

of three steps: (1) finding confusing classes in the target set and confident classes in the anchor set and generating triplet-based relationships (*e.g.*, target class $Chimpanzee$ is closer to anchor class $Gorilla$ than to anchor class $Deer$); (2) translating the detected relational hypotheses into a ranked list of semantic questions to obtain human judgement concerning their validity; (3) translating validated geometric relations into regularizers for the objective function and retraining the model.

Our contributions are threefold: (1) We propose an interactive learning framework that can be incrementally improved, by asking for verification of semantic queries from humans and taking their responses into account. (2) As part of the learning framework, we present an active selection method that automatically generates semantic queries from a learned model by detecting relational regularities, and ranking them by their expected impact on recognition performance. (3) We empirically validate that our method transfers knowledge for better classification via relational semantics to *target* categories, and thus improves their classification performance.

Related Work

Encoding semantics for object recognition. The most popular semantic sources of information that have been explored for improving recognition accuracy are attributes and taxonomies (Marszalek and Schmid 2008; Hwang, Sha, and Grauman 2011; Akata et al. 2013; Zhou, Xiao, and Wu 2011; Hwang and Sigal 2014). While most previous work leverages taxonomies and attributes by focusing on shallow properties such as similarity between the semantic entities, some recent works focus on their geometrical relationships. (Mikolov, Tau Yih, and Zweig 2013) showed that there exist regularity among word vectors trained on the skip-gram models (*e.g.*, the words form analogies). The same analogical relations were explored in (Hwang, Grauman, and Sha 2013) and (Law, Thome, and Cord 2013) to regularize the geometry of the learned category embeddings for categorization, such that category embeddings associated with an analogy form a parallelogram. More similar to our design, (van der Maaten and Weinberger 2012) took advantage of relative closeness, encoded by triplets of entities.

The limitation of all these methods is that they require a pre-constructed knowledge base, which often takes a lot of human effort and expertise to create. Such knowledge bases may not be readily available for atypical classes, *e.g.*, specific dog breeds or exotic car models. Our method does not require a predefined knowledge base, and is designed to ascertain the most informative, from the model's point of view, semantic relationships from human users/expert(s).

Active/Interactive/Self-paced learning. Our method, which actively selects a few important relational patterns to validate through user feedback, is an instance of active/interactive learning. Generally, active learning focuses on selecting instances based on an estimated contribution that the selected instances can make towards improving classification and asking for corresponding category labels from a human annotator. Recently, non-class label type queries have been also explored for active-learning, such as in (Kovashka, Vijayanarasimhan, and Grauman 2011), which presents an active learning algorithm that can either ask for attribute or category labels, while learning a joint object categorization model. Pairwise similarity, which forms our relational patterns, has also been explored in (Karbasi, Ioannidis, and Massoulie 2012). However, in this work, the queries are selected to better search for the target in a fixed metric space, while our method iteratively retrains the metric space given the answered queries.

The closest work to ours, in terms of motivation, is (Eaton, Holness, and McFarlane 2010), which generates active queries considering the geometry of the manifold, and retrains the model with the newly annotated samples. However, (Eaton, Holness, and McFarlane 2010) focuses on the instance(sample)-level geometry while we focus on *semantically* important geometrical patterns among category prototypes. (Bilgic, Mihalkova, and Getoor 2010) makes use of graph structure to select instances from groups for queries; instances whose collective label prediction disagrees with instance label prediction are preferred. Our method also makes use of structural relationships, but focuses on the geometry of category prototypes rather than instances.

(Parikh and Grauman 2011) also closely share our motivation of building a semantic model by iteratively selecting semantically meaningful hypotheses from a pool of candidates. They generate discriminative visual attribute hypotheses and then present human subjects a set of images with and without such attributes, and ask them to name the attributes that differentiate between the two, only if the difference is nameable. A model with such semantic refinement was shown to outperform the non-semantic initial variant.

Self-paced learning, or curriculum learning, (Kumar, Packer, and Koller 2010; Lee and Grauman 2011) is a learning paradigm that incrementally learns from subsets of labeled instances instead of learning in a batch. Self-paced learning iteratively builds the model using samples that are discovered adaptively, based on the model at the previous iteration. Our approach is an instance of self-paced learning, but discovers semantic constraints rather than instances. Further, since semantics are latent, and do not directly correlate with recognition performance, a useful criterion for iterative selection of such entities is much more difficult to identify.

Lifelong learning. Lifelong learning (Thrun 1995) is a learning paradigm that continuously learns from a stream of incoming inputs, while transferring knowledge obtained from earlier stages to later ones. It has gained popularity due to its scalability and applications that deal with long streams of inputs, *e.g.*, in web-scale data, wearable cameras and autonomous vehicles. Since the inception of the idea by the seminal work of (Thrun 1995), many researchers have worked on such continuous learning systems. Recent work includes (Eaton and Ruvolo 2013), which efficiently learns latent shared bases for all tasks in an online learning framework. The model was later expanded, in (Ruvolo and Eaton 2013), to allow active selection of tasks at each iteration. We hope that our interactive learning paradigm, that learns semantic information online, can serve as a module in such lifelong learning frameworks to mitigate *semantic drift* through intermittent, but focused, human feedback.

Knowledge transfer. When little labeled data is available for certain categories, transferring knowledge from related categories can be helpful. (Tommasi, Orabona, and Caputo 2010) adapt classifiers for classes with few training instances by utilizing information from classifiers of classes with sufficiently large numbers of training instances. However, they transfer information in a batch, where as our method focuses on incremental transfer and improvements. (Qi et al. 2011) similarly use cross-category knowledge to improve image classification accuracy in a batch.

Approach

Given a labeled dataset $D = \{(\boldsymbol{x}_i, y_i) \in (\mathbb{R}^d, \mathcal{Y})\}_{i=1}^N$, where $\boldsymbol{x}_i$ is a d-dimensional feature vector of i^{th} sample, y_i is its class label and N is the number of samples, we learn a model that minimizes classification error for new, unknown, sample $\boldsymbol{x}^*$ at test time. We adopt an efficient and scalable discriminative embedding approach (Bengio, Weston, and Grangier 2010) for classification, where both the samples, $\boldsymbol{x}_i$, and their labels, y_i, are projected into a common low dimensional space $\mathbb{R}^m$, where $m \ll d$. We denote the projected version of $\boldsymbol{x}_i$ as $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ and class label $y_i = c \in \mathcal{Y}$ as $\boldsymbol{u}_c$. The goal is then to learn both the embedding function $f(\cdot)$ and the location of the prototypes $\boldsymbol{u}_c$ for all classes such that the projected test instance $f(\boldsymbol{x}^*)$ would be closer to the correct class prototype than to others.

Given semantic information, this model can be further improved (Hwang, Grauman, and Sha 2013; Law, Thome, and Cord 2013) through graph-based regularization, *i.e.*, semantic relationships constrain the placement of prototypes in the embedding space. However, as the number of entities increases, the number of possible relationships between them increases rapidly, making it impractical to consider all semantic relationships offline. Further, even if one has a complete set of semantic information, not only does using all of the semantic relationships lead to an exorbitant computational expense, but also not all semantics are equally useful for discriminative classification. This suggests that incorporating all of the semantics may even degrade classification performance. One often needs to trade off discriminative classification accuracy with the ability to encode all the semantics in the knowledge set due to limited capacity of a fixed dimensional manifold. To address this, we aim to actively identify a small subset of semantic relations that are most helpful in learning a discriminative classification model. We make use of semantics in the form of relative distances: "class a is more similar to class b than to class c." However, the total number of such triplet relationships is cubic in the number of category labels. To avoid the cost of constructing a complete semantic knowledge base, we propose an interactive approach to acquire an informative subset of them. Specifically, we repeat the following three steps. 1) detect geometric patterns that constitute potential semantic triplet queries with respect to the current model, 2) obtain 'yes' or 'no' answers to these semantic questions from a human and 3) retrain the model by imposing structural regularizers based on the obtained semantic knowledge. We summarize the overall procedure in Algorithm 1 and describe the steps in the following subsections.

Algorithm 1 Interactive Learning with Semantic Feedback

```
Input: $(x_i, y_i) \in \mathbb{R}^d \times \mathcal{Y},\ \forall i \in \{1, \dots N\}$.
Output: $\boldsymbol{W} \in \mathbb{R}^{m \times d}, \boldsymbol{U} \in \mathbb{R}^{m \times C}$.
$\mathcal{R} \leftarrow \emptyset$, Initialize $\boldsymbol{W}_{prev}, \boldsymbol{U}_{prev}$ with random matrices
$\boldsymbol{W}^A$ and $\boldsymbol{U}^A \leftarrow$ Solve Eq.(1)
$\delta\boldsymbol{W} = \boldsymbol{W}^A - \boldsymbol{W}_{prev}$, $\delta\boldsymbol{U} = \boldsymbol{U}^A - \boldsymbol{U}_{prev}$
while $\delta\boldsymbol{W} > \epsilon$ and $\delta\boldsymbol{U} > \epsilon$ do
    $\boldsymbol{W}$ and $\boldsymbol{U} \leftarrow$ Solve Eq.(2) with $\mathcal{R}, \boldsymbol{W}^A$
    $\mathcal{P} \leftarrow GenerateOrderedQueries(\boldsymbol{W}, \boldsymbol{U}, \mathcal{R})$ (Section
    What Questions to Ask First?)
    $R \leftarrow Feedback(\mathcal{P})$ (Section Feedback)
    $\mathcal{R} \leftarrow \mathcal{R} \cup R$
    $\delta\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{W}_{prev}$, $\delta\boldsymbol{U} = \boldsymbol{U} - \boldsymbol{U}_{prev}$
    $\boldsymbol{U}_{prev} = \boldsymbol{U}$, $\boldsymbol{W}_{prev} = \boldsymbol{W}$
end while
```

Discriminative Semantic Embedding

To detect patterns that can be translated into semantic queries, we use a manifold embedding approach, where both the data points (features) and the semantic entities (category labels) are embedded as points on a manifold. The semantic queries are posed and the answers used to refine the manifold. Both query generation and categorization are done in this embedding space (Weinberger and Chapelle 2008). With the relational semantics, the manifold is discriminatively learned with a large margin loss function.

Formally, we want to embed both the image features $\boldsymbol{x}_i$ and corresponding class labels y_i into a common low-dimensional space such that the projection of $\boldsymbol{x}_i$, denoted as $\boldsymbol{z}_i$, is more similar to the corresponding category embedding $\boldsymbol{u}_{y_i}$ than the embeddings of all other categories. This is accomplished by constructing a linear projection $\boldsymbol{W} \in \mathbb{R}^{m \times d}$ such that $\boldsymbol{z}_i = \boldsymbol{W}\boldsymbol{x}_i$, and $\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 + 1 \leq \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2, \forall c \neq y_i$.

For knowledge transfer, we first build a reference model with relatively well-trained *anchor* classes. Then we build model for the *target* classes by transferring semantic information from the *anchor* classes.

Semantic embedding for anchor classes The objective for categorizing semantic embeddings for the *anchor* classes is expressed as minimization of the large-margin constraints above for all anchor class instances indexed by $i \in \{1, \dots, N^A\}$ with respect to $\boldsymbol{W}^A$ and prototypes $\boldsymbol{u}_c$:

$$\min_{\boldsymbol{W}^A, \boldsymbol{U}^A} \sum_{i=1}^{N^A} \sum_{c \in \mathcal{C}^A} \mathcal{L}\left(\boldsymbol{W}^A, \boldsymbol{x}_i, \boldsymbol{u}_c\right) + \lambda_1 \|\boldsymbol{W}^A\|_F^2 + \lambda_2 \|\boldsymbol{U}^A\|_F^2,$$

$$\text{s.t. } \mathcal{L}(\boldsymbol{W}^A, \boldsymbol{x}_i, \boldsymbol{u}_c) = \max\left(\|\boldsymbol{W}^A\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 - \|\boldsymbol{W}^A\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2 + 1, 0\right), \forall i, \forall c \neq y_i, \tag{1}$$

where N^A is the number of training samples in anchor classes ($\mathcal{C}^A$), $\boldsymbol{U}^A$ is a stacked column matrix of label prototypes $\{\boldsymbol{u}_c\}$ of the anchor classes and λ_1 and λ_2 are hyperparameters for scale regularization; $\|\cdot\|_F$ is the Frobenius norm.

Knowledge transfer via relational semantics From the learned *anchor* class categorization model with $\boldsymbol{W}^A$ and $\boldsymbol{U}^A$, we transfer the knowledge to the *target* classes that have only a few training samples. Specifically, we use interactively provided semantic relationships $R \in \mathcal{R}$ to regularize the objective function. Formally, learning the discriminative embeddings of target classes can be achieved by solving the following regularized optimization problem:

$$\begin{aligned}\min_{\boldsymbol{W}^T,\boldsymbol{U}^T} \sum_{i=1}^{N^T} \sum_{c \in \mathcal{C}^T} \mathcal{L}\left(\boldsymbol{W}^T, \boldsymbol{x}_i, \boldsymbol{u}_c\right) &+ \lambda_1 \|\boldsymbol{W}^T\|_F^2 + \lambda_2 \|\boldsymbol{U}\|_F^2 \\ &+ \lambda_3 \|\boldsymbol{W}^T - \boldsymbol{W}^A\|_F^2 + \gamma \sum_j \Omega\left(R_j, \boldsymbol{U}\right), \\ \text{s.t. } \mathcal{L}(\boldsymbol{W}^T, \boldsymbol{x}_i, \boldsymbol{u}_c) = \max\left(\|\boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 - \|\boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2 + 1, 0\right)&, \\ \forall i, \forall c \neq y_i,\ R_j \subset \mathcal{R}&,\end{aligned} \tag{2}$$

where N^T is the number of training samples in the target classes ($\mathcal{C}^T$), R_j is a subset of $\mathcal{R}$ (the set containing all semantic constraints), and $\boldsymbol{U} = [\boldsymbol{U}^A, \boldsymbol{U}^T]$ is the concatenation of all class prototypes. We regularize the data embedding $\boldsymbol{W}^T$ with $\boldsymbol{W}^A$, and the semantic embedding with $\Omega(R_j, \boldsymbol{U})$, which is a regularizer defined on the relationship R_j, described subsequently.

Encoding relational semantics by geometric topologies. The semantic relationships are used to regularize the embedding space for better classification generalization (Hwang, Grauman, and Sha 2013; Law, Thome, and Cord 2013; Hwang and Sigal 2014). As mentioned previously, we use the *triplet-based relationships* in which human feedback is of the form: 'object a is more similar to b than to c.' Triplet-based relationships have the favorable property of minimal need to reconcile feedback scale (Tamuz et al. 2011; Kendall and Gibbons 1990). Even though the relationships are local with respect to the associated entities, solving the optimization using the relationships, Eq.(2), changes the topology of the class prototype embeddings globally, which results in a semantically more meaningful model overall.

Suppose a target entity, $\boldsymbol{u}_t$, is semantically closer to the anchor entity $\boldsymbol{u}_{a_1}$ than to another anchor entity $\boldsymbol{u}_{a_2}$; we denote such a relationship by $R = (t, (a_1, a_2))$ and define its geometric regularizer as a hinge loss type of regularizer that encourages moving $\boldsymbol{u}_t$ closer to $\boldsymbol{u}_{a_1}$ and farther from $\boldsymbol{u}_{a_2}$:

$$\max\left(1 - \|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|_2^2 / \|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2, 0\right). \tag{3}$$

Eq.(3), however, is neither differentiable nor convex in terms of $\boldsymbol{u}_*$'s thus making the optimization difficult if it is used as a regularization term. So, we relax the regularizer by introducing a scaling constant σ_1 as a proxy of $\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2$ by the distance between the sample means of classes a_1 and t. In addition, the $\max(x, 0)$ is not continuous at $x = 0$, thus not differentiable. So, we use a differentiable smooth proxy of the $\max(x, 0)$ function, $h_\rho(\cdot)$, to make the regularizer differentiable everywhere:

$$\Omega(R, \boldsymbol{U}) = \sigma_1 h_\rho\left(\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2 - \|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|_2^2\right) \tag{4}$$

where $h_\rho(x)$ is a differentiable proxy for $\max(x, 0)$ as in (Amit et al. 2007). A detailed description of $h_\rho(\cdot)$ can be found in the supplementary material[1].

Numerical optimization. The optimization problems in Eq. (1) and Eq. (2) are not convex in both $\boldsymbol{W}$ and $\boldsymbol{U}$. However, we can use alternating optimization to obtain a reasonable local minimum, where we alternate between optimization of $\boldsymbol{W}$ and $\boldsymbol{U}$ while fixing the other. We use the stochastic sub-gradient method to optimize for each variable.

What Questions to Ask First?

To reduce the number of semantic relationships in the regularizer, while aiming for better classification, we discover candidate semantic questions that are helpful for improving classification accuracy.

Generating a pool of queries. We first generate a pool of candidate triplet-based semantic relationships; $\mathcal{R} = \{R | R = (t, (a_1, a_2))\}$. R has three entities; target, $\boldsymbol{u}_t$, and two anchors $(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2})$. We want to improve the classification of the target class by transferring knowledge from the anchor classes, which are more confidently classified than the target. To generate the pool of triplets, we find the target classes that are highly confused (*i.e.*, classification accuracy in the current model is low) and the anchor classes that are highly confident (*i.e.*, classification accuracy in the current model is high). Specifically, for each $R = (t, (a_1, a_2))$, we define a scoring function, $S(R, \boldsymbol{U})$, for querying semantic relationship by favoring the most confusing (the least confident) target class and the least confusing (the most confident) anchor classes. For the measure of confusion of each class, we regard label's prototype in the projected space as a random variable for class label and use its entropy, $H(\boldsymbol{u}_c)$. The entropy of an entity can be written as:

$$H(\boldsymbol{u}_c) = -\sum_{j \in \mathcal{C}} P_{\boldsymbol{u}_c}(j) \log P_{\boldsymbol{u}_c}(j), \tag{5}$$

where $\mathcal{C}$ is a set of all class labels. For joint and conditional entropy, we derive joint and conditional probability mass function of multiple label entities in the supplementary material[1]. The higher the entropy, the higher the confusion. We then define the scoring function as the conditional entropy of a target entity, $\boldsymbol{u}_t$, given anchor entities $(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2})$ as:

$$S(R, \boldsymbol{U}) = H(\boldsymbol{u}_t | \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) = H(\boldsymbol{u}_{t_1}, \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}), \tag{6}$$

Given the label of the target entity $\boldsymbol{u}_t$ of the candidate relationship R, we want the anchor entities to be even more certain. In other words, we assume the uncertainty of anchor entities given the target entity label, $H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2} | \boldsymbol{u}_t)$, is 0. Then, we can reduce (6) to:

$$S(R, \boldsymbol{U}) = H(\boldsymbol{u}_t) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}). \tag{7}$$

Intuitively, the score favors choosing target entities that have high classification confusion and the anchor entities that have low classification confusion. Detailed descriptions of how to compute the probability mass function for the entropy, and the derivation of the conditional entropy function can be found in the supplementary material.

Scoring metric to prioritize the queries. Given the pool of queries, we prioritize the queries to reduce the number of questions to be answered for efficiency. Note that in the

[1] http://umiacs.umd.edu/∼jhchoi/paper/aaai16salsupp.pdf

interactive setting, in principle, it is optimal to ask one question at a time. However, this can be expensive as it requires frequent re-training of the model. An alternative is to ask mini-batches of questions at a time. In both cases the scoring scheme is crucial for selecting one (or a few) most useful questions from the pool to maximize the effect of the knowledge transfer. To this end, we consider several scoring metrics.

• *Entropy based score.* The entropy based score uses the conditional entropy scores computed in the pool generation process to prioritize the queries (Eq.(7)). Although this metric is good for generating a potential set of queries that could improve accuracy the most, it cannot directly predict the potential accuracy improvement from enforcement of the corresponding relational semantics. For example, when *Deer* is the confused target class and *Elephant* and *Killer Whale* are confident anchor classes, the entropy is going to be high, but the actual accuracy improvement that may result by enforcing the relational semantics of *Deer* is closer to *Elephant* than *Killer Whale* may not be.

• *Classification accuracy.* To obtain a good scoring function of the relational semantics, we use classification accuracy of each candidate constraint computed using a validation set. Validation set accuracy is a direct proxy of expected classification gain of each relation. Further, since we only order questions from a pool of a small number of queries, this is still computationally viable (not so if considering all possible semantic relationships as the pool).

• *Predicting the classification accuracy by a regression model.* Computing the classification accuracy of each constraint even within the pool at every iteration is still computationally expensive; thus we introduce a method to approximate it by regressing over multiple types of *features*, which are proxies for estimating the classification improvement by a vector of various scores ($\boldsymbol{c}$) to the validation accuracy (s). Suppose the relationship consists of target class t and two anchor classes a_1 and a_2. We use a score vector to estimate the validation accuracy. The details of score vectors can be found in the supplementary material[1]. Using a set of features ($\boldsymbol{C}$) and corresponding validation accuracies ($\boldsymbol{s} = \{s\}$), we obtain a linear regression model, with a bias, by solving $\hat{\boldsymbol{R}} = \arg\min_{\boldsymbol{R}} \|\boldsymbol{R}^T[\boldsymbol{C}; 1] - \boldsymbol{s}\|_2^2$, where we can use $\hat{\boldsymbol{R}}$ to regress the validation accuracy.

Feedback

We can obtain feedbacks from human expert(s). We simulate human feedbacks by an oracle that provides answers based on the distance of attribute descriptions. Since the attribute description is an agglomerative score of different criteria from a number of human annotators, it is a reasonable measure for the semantic decision regarding validity of relational queries. Specifically, for each triplet-based relationships, we compute the distance of attribute description of $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_1}$ and $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_2}$. If the semantic distance between $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_1}$ is smaller than the distance between $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_2}$, the oracle answers 'Yes', and 'No' otherwise. We only use the relationships that are answered as 'Yes' as constraints.

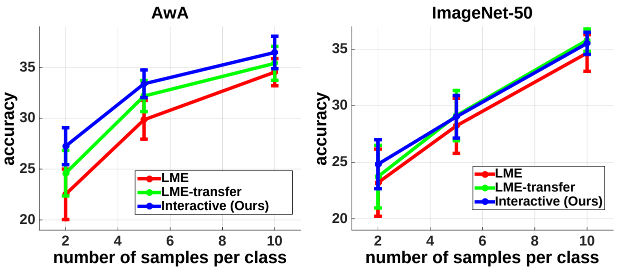

Figure 2: Classification accuracy of target classes on AWA and ImageNet-50 dataset. Results are average accuracies over five random splits with standard error shown at 95% confidence interval.

Interactive Learning

The key to our approach is to adaptively update the query generation. We refer to this as the 'interactive' model. So far, we describe one iteration of human interaction. We iterate the process multiple times, updating the embedding manifold (model) and use the updated model to generate a new pool of queries and prioritize them for the next iteration. The adaptive query generation and prioritization scheme achieves better classification accuracy with a fewer number of relational constraints, compared to a single iteration model, which we refer to as an 'active' model. In other words, the interactive model is more efficient in terms of utilizing human feedback.

Experiments

Datasets and Experimental Details

We use two object categorization datasets: 1) Animals with Attributes (AWA) (Lampert, Nickisch, and Harmeling 2009), which consists of 50 animal classes and 30,475 images, 2) ImageNet-50 (Hwang, Grauman, and Sha 2013), which consists of 70,380 images of 50 categories. In both datasets, we configure 40 classes as anchor classes and 10 classes as target classes. For testing and validation set, we use a 50/50 split of the remaining samples, excluding the training samples. Details of the datasets including information about anchor classes and target classes can be found in the supplementary material[1].

We evaluate the performance of knowledge transfer by measuring the classification accuracy of each model on the target classes in a challenging set-up that has only a few training samples (2, 5 and 10 samples per class, few-shot learning) with a prior learned with anchor classes that have a larger numbers of training samples (30 samples per class). We use test sets that are much larger (300 (AWA) or 700 (Imgnet-50) per category) than the training set.

Classification Accuracy

Fig. 2 shows the classification accuracy on target classes for the two datasets. Our interactive model (Interactive) with the scoring metric described in *Comparison of Different Query-Scoring Metrics of the Proposed Method* section outperforms the baseline transfer models (LME-transfer) without semantic constraints and the large margin model without knowledge transfer (LME). Specifically, 'LME' refers to the

Dataset	Animals with Attribute			ImageNet-50		
# samples/class	2	5	10	2	5	10
LME	22.51±2.48	29.85±1.90	34.52±1.33	23.20±2.97	28.22±2.43	34.67±1.62
LME-Transfer	24.59±2.23	32.17±1.53	35.39±1.67	23.47±2.66	28.78±2.05	34.94±1.03
Random	24.75±2.11	31.32±1.31	35.96±1.66	24.23±1.92	28.72±2.26	34.74±2.26
Entropy	24.96±2.24	31.81±1.27	35.92±1.91	24.60±2.80	28.88±2.43	35.64±0.99
Active-Regression	25.43±1.90	32.49±1.58	36.18±0.88	23.34±2.76	28.99±2.34	35.49±0.89
Active	26.62±1.67	32.42±1.45	36.40±1.33	24.35±2.42	28.55±2.07	35.60±1.01
Interactive	**27.24±1.82**	**33.31±1.28**	**36.46±1.60**	**24.95±2.20**	**29.08±1.88**	**35.62±1.01**
Interactive-UB	28.57±1.85	33.61±2.15	36.86±1.83	25.15±2.13	29.23±1.85	35.95±1.53

Table 1: Classification accuracy (%) of the proposed method using different scoring functions . For comparison, we also provide two baselines, LME and LME-Transfer, and the upper-bound of our interactive model (Interactive-UB), which uses the test set to score the constraints.

model learned using Eq.(2) with $\lambda_3 = 0, \gamma = 0$, and 'LME-Transfer' refers to the model learned using Eq.(2) without the semantic constraints ($\gamma = 0$). For 'Interactive', we add 20 semantic constraints per iteration and run 5∼6 iterations, so add 100∼120 semantic constraints in total.

Effect of interaction. Our interactive learning scheme continuously updates the model to select a better set of questions in terms of classification accuracy. We use a mini batch size of 10 for the interactive setting. The interactively mined constraints provide better classification accuracy over an equivalent sized set of constraints produced in a batch. The left plot in Fig. 3 shows the classification accuracy as a function of number of constraints added by the iteratively updated model and by a batch model. In both cases, the same measure for selection and ordering was used. Interestingly, as iterations continue, the accuracy starts to drop. We believe it is because there are not helpful semantic relationships to be added for classification past certain iterations.

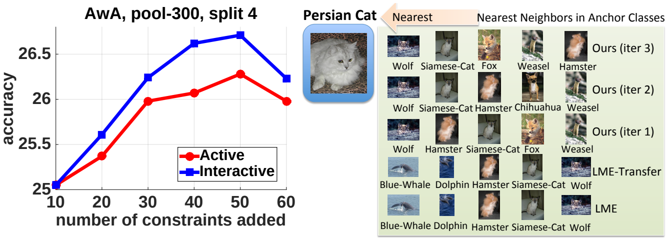

Figure 3: Benefit of interaction over batch-mode active criterion. **left:** Classification accuracy as a function of number of constraints added by active (batch) or interactive scoring. **right:** Qualitative result of nearest neighbor of target class over iterations.

As a qualitative result, we present the nearest neighbors of a target class in the anchor set in Fig. 3. As baseline models (LME, LME-Transfer) do not explicitly enforce the semantic relationships of categories, the nearest neighbors obtained by the baseline models are not semantically meaningful. The nearest neighbors obtained using our model, however, are semantically meaningful from the first iteration onward. As iterations proceed, the nearest neighbor is further refined to be semantically more meaningful, *e.g.*, *Siamese-cat* appears as the second nearest neighbor in iteration 2 and 3 while it is a third-nearest neighbor at the first iteration.

As interaction proceeds, the embedding space becomes semantically more meaningful as do the generated queries.

# Iteration	Positively answered query at its highest rank
1	|fox - persian cat| < |blue whale - persian cat|
2	|grizzly bear - persian cat| < |horse - persian cat|
3	|dalmatian - persian cat| < |beaver - persian cat|
4	|dalmatian - persian cat| < |german shepherd - persian cat|

Table 2: Top ranked query as interaction proceeds. As interactions continue, top ranked query whose target class is 'Persian Cat' becomes semantically more meaningful.

Table 2 shows the top positively answered queries related to the *Persian-cat* category as a function of iterations. In early iterations, the questions try to relate *Persian-cat* to *fox* and *blue whale*. But in the later iterations, the question becomes more semantically meaningful - comparing *Persian-cat* with *dalmatian* and *german shepherd*.

Comparison of different query-scoring metrics of the proposed method. The scoring metric for a query is one of the most important components in the interactive framework. In Table 1, we compare the accuracy obtained by adding the constraints with various scoring schemes that we have presented in *Scoring Metric to Prioritize the Queries* section. The number of constraints added and other hyperparameters are determined by cross validation. The scoring schemes include 'Random'–random ordering of query from the selected pool, 'Entropy'–Entropy-based scores, 'Active'–classification accuracy based score by a batch-mode model, 'Active-Regression'–regressed score of the classification accuracy obtained by a batch-mode linear regression model, and 'Interactive'–classification accuracy based score by an adaptively updated model, which is our proposal. 'Interactive-UB' refers to the upper bound that our framework can achieve: we score and add the queries based on classification accuracy with the test set itself in our interactive model. Note that except 'Interactive', all other scoring metrics are in a batch-mode. The interactive model outperforms the batch mode model, which we denote as 'Active', and other scoring schemes, and is tight to the upper bound. We also present the baseline results of 'LME' and 'LME-Transfer' for reference.

Note that all methods use the same validation set to tune parameters. Our scoring metric in 'Active' and 'Interactive', in addition, uses it to prioritize queries to the user as this is the most direct way to measure the effect of adding a particular constraint on the recognition accuracy without using the testing set. While this perhaps makes direct comparison to the baselines slightly less transparent, the comparison of 'Active' and 'Interactive' variants, which both use this criterion, clearly points to the fact that 'Interactive' learning is much more effective in selecting and ordering of constraints.

Conclusion

We proposed an interactive learning framework that takes human feedback to iteratively refine a learned model. Our method detects recurring relational patterns from a semantic manifold and translates them into semantic queries to be answered. We then retrain the model by imposing the constraints obtained by positively feeding back the semantic relationships. We validate our method against batch learning methods on classification accuracy of target classes with transferred knowledge from anchor classes via relational semantics.

Acknowledgement

J. Choi and L.S. Davis were partially supported by MURI from the Office of Naval Research under the Grant N00014-10-1-0934.

References

[Akata et al. 2013] Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2013. Label-Embedding for Attribute-Based Classification. In *CVPR*.

[Amit et al. 2007] Amit, Y.; Fink, M.; Srebro, N.; and Ullman, S. 2007. Uncovering Shared Structures in Multiclass Classification. In *ICML*.

[Bengio, Weston, and Grangier 2010] Bengio, S.; Weston, J.; and Grangier, D. 2010. Label Embedding Trees for Large Multi-Class Tasks. In *NIPS*.

[Bilgic, Mihalkova, and Getoor 2010] Bilgic, M.; Mihalkova, L.; and Getoor, L. 2010. Active learning for networked data. In *International Conference on Machine Learning (ICML)*.

[Eaton and Ruvolo 2013] Eaton, E., and Ruvolo, P. L. 2013. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning (ICML)*, 507–515.

[Eaton, Holness, and McFarlane 2010] Eaton, E.; Holness, G.; and McFarlane, D. 2010. Interactive learning using manifold geometry. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[Hwang and Sigal 2014] Hwang, S. J., and Sigal, L. 2014. A unified semantic embedding: Relating taxonomies and attributes. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 271–279.

[Hwang, Grauman, and Sha 2013] Hwang, S. J.; Grauman, K.; and Sha, F. 2013. Analogy-preserving semantic embedding for visual object categorization. In *International Conference on Machine Learning (ICML)*, 639–647.

[Hwang, Sha, and Grauman 2011] Hwang, S. J.; Sha, F.; and Grauman, K. 2011. Sharing features between objects and their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1761–1768.

[Karbasi, Ioannidis, and Massoulie 2012] Karbasi, A.; Ioannidis, S.; and Massoulie, L. 2012. Comparison-based learning with rank nets. In *International Conference on Machine Learning (ICML)*, 855–862.

[Kendall and Gibbons 1990] Kendall, M., and Gibbons, J. D. 1990. *Rank Correlation Methods*. 5 edition.

[Kovashka, Vijayanarasimhan, and Grauman 2011] Kovashka, A.; Vijayanarasimhan, S.; and Grauman, K. 2011. Actively selecting annotations among objects and attributes. *IEEE International Conference on Computer Vision (ICCV)* 1403–1410.

[Kumar, Packer, and Koller 2010] Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-Paced Learning for Latent Variable Models. In *NIPS*.

[Lampert, Nickisch, and Harmeling 2009] Lampert, C.; Nickisch, H.; and Harmeling, S. 2009. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Law, Thome, and Cord 2013] Law, M. T.; Thome, N.; and Cord, M. 2013. Quadruplet-wise Image Similarity Learning. In *CVPR*.

[Lee and Grauman 2011] Lee, Y. J., and Grauman, K. 2011. Learning the Easy Things First: Self-Paced Visual Category Discovery. In *CVPR*.

[Marszalek and Schmid 2008] Marszalek, M., and Schmid, C. 2008. Constructing category hierarchies for visual recognition. In *European Conference on Computer Vision (ECCV)*.

[Mikolov, Tau Yih, and Zweig 2013] Mikolov, T.; Tau Yih, W.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 746–751.

[Parikh and Grauman 2011] Parikh, D., and Grauman, K. 2011. Interactively building a discriminative vocabulary of nameable attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1681–1688.

[Qi et al. 2011] Qi, G.-J.; Aggarwal, C.; Rui, Y.; Tian, Q.; Chang, S.; and Huang, T. 2011. Towards Cross-Category Knowledge Propagation for Learning Visual Concepts. In *CVPR*.

[Ruvolo and Eaton 2013] Ruvolo, P., and Eaton, E. 2013. Active task selection for lifelong machine learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[Tamuz et al. 2011] Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. T. 2011. Adaptively Learning the Crowd Kernel. In *ICML*.

[Thrun 1995] Thrun, S. 1995. A lifelong learning perspective for mobile robot control. In Graefe, V., ed., *Intelligent Robots and Systems*. Elsevier.

[Tommasi, Orabona, and Caputo 2010] Tommasi, T.; Orabona, F.; and Caputo, B. 2010. Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer. In *CVPR*.

[van der Maaten and Weinberger 2012] van der Maaten, L., and Weinberger, K. 2012. Stochastic Triplet Embedding. In *IEEE Int'l Workshop on Machine Learning for Signal Processing*.

[Weinberger and Chapelle 2008] Weinberger, K., and Chapelle, O. 2008. Large Margin Taxonomy Embedding with an Application to Document Categorization. In *NIPS*.

[Zhou, Xiao, and Wu 2011] Zhou, D.; Xiao, L.; and Wu, M. 2011. Hierarchical Classification via Orthogonal Transfer. In *ICML*.

Knowledge Transfer with Interactive Learning of Semantic Relationships

Supplementary Material

Smoothed Hinge Loss function $h_\rho(\cdot)$

In order to use the gradient descent optimization method at the peak points, we approximate them by smoothed versions as shown by the blue curves in Fig. 1 as in (Amit et al. 2007).

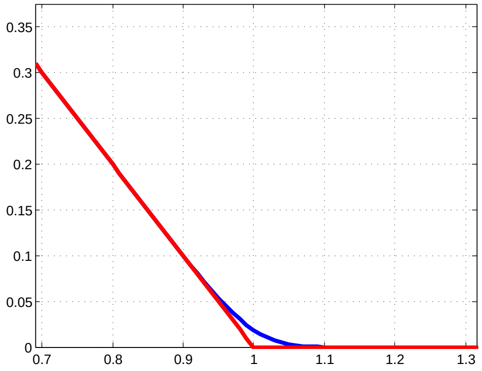

Figure 1: **Smoothed hinge loss.**

$h_\rho(\cdot)$ is the approximate hinge loss function that has no discontinuity:

$$h_\rho(z) = \begin{cases} 1 - z & z < 1 - \rho \\ \frac{-(1-z)^4}{16\rho^3} + \frac{3(1-z)^2}{8\rho} + \frac{(1-z)}{2} + \frac{3\rho}{16} & |1 - z| \leq \rho, \\ 0 & z > 1 + \rho. \end{cases} \quad (1)$$

and its derivative with respect to z is

$$\frac{\partial h_\rho(z)}{\partial z} = \begin{cases} -1 & z < 1 - \rho \\ \frac{(1-z)^3}{4\rho^3} - \frac{3(1-z)}{4\rho} - \frac{1}{2} & |1 - z| \leq \rho. \\ 0 & z > 1 + \rho. \end{cases} \quad (2)$$

In our experiments we use $\rho = \sigma = 10^{-7}$.

Probability Mass Function

To compute the score by the entropy (Sec. 3.2 in the main paper), we define each entity's probability mass function by its classification confusion on validation set. Specifically, the probability of a label entity $\boldsymbol{u}_i$ to be a class label j is definedas:

$$P_{\boldsymbol{u}_i}(j) = \frac{\sum_{\boldsymbol{x}_k \in \mathcal{V}} \mathbb{1}(g(\boldsymbol{x}_k) = j)}{|\mathcal{V}|}, \quad (3)$$

where $g(\cdot)$ is the current classification model learned with $\boldsymbol{u}_i$ and $\boldsymbol{x}_k$ and $\mathcal{V}$ is a set of feature embeddings, $g(\boldsymbol{x}_k)$, in validation set. Thus $\mathbb{1}(g(\boldsymbol{z}_k) = j)$ equals to the number of feature embeddings whose obtained label by the current model is j. $|\cdot|$ denotes cardinality of a set. The ideal PMF is a delta function when $c = j$; $\delta(c = j)$. Note that the measure depends on the sample distribution under the current model. Thus, the entropy of an entity can be written as:

$$H(\boldsymbol{u}_i) = -\sum_{j \in \mathcal{C}} P_{\boldsymbol{u}_i}(j) \log P_{\boldsymbol{u}_i}(j), \quad (4)$$

where $\mathcal{C}$ is a set of all class labels. For the joint entropy, we need to derive a joint probability mass function of multiple label entities.

Joint Probability Mass Function of Multiple Entities

For computing a joint entropy, deriving a joint probability mass function (PMF) of multiple entities from Eq.(3) is straightforward. We start from the joint PMF of two entities, $P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2)$. Since the probability of $\boldsymbol{u}_i$ being label c_1 is dependent on the obtained labels of neighboring feature embeddings, $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$, $P_{\boldsymbol{u}_i}(c_1)$ is actually a conditional probability as:

$$\begin{aligned} P_{\boldsymbol{u}_i}(c_1) &= P_{\boldsymbol{u}_i}(c_1|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) \\ &= P_{\boldsymbol{u}_i}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \mathcal{N}^i\}). \end{aligned} \quad (5)$$

We can write the joint PMF of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ as:

$$\begin{aligned} P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2) &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)P_{\boldsymbol{u}_j}(c_2) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2)P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \mathcal{N}^j\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2), \end{aligned} \quad (6)$$

the second to third line is because if $\boldsymbol{u}_j$ is given (or known), $\mathcal{N}^j$ are not necessary as conditioned variables; $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2) =$

$P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})$. Then the conditional probability of $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)$ and the joint probability of $(\boldsymbol{u}_i, \boldsymbol{u}_j)$ can be written as:

$$P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \tag{7}$$

$$P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \cdot \frac{\sum_{\boldsymbol{z}_i \in \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_2)}{|\mathcal{N}^j|}. \tag{8}$$

A joint PMF of more than three variables can be straightforwardly obtained by the chain rule.

Conditional Entropy

By the independence of variable for conditional entropy, we have the following equation:

$$\begin{gathered} H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}|\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) - H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = 0, \\ H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}). \end{gathered} \tag{9}$$

Using Eq.(9) here, we can derive Eq.(6) in the main paper from Eq.(5) in the main paper as:

$$\begin{aligned} S(R, \boldsymbol{U}) &= H(\boldsymbol{u}_t, \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) \\ &= H(\boldsymbol{u}_t) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}). \end{aligned} \tag{10}$$

Computational Complexity

The computational complexity of Algorithm 1 depends on the complexity of training the model on the anchor and target classes, and generating a query pool.

First, the complexity of training the model on the anchor classes in Eq.(1) in the main paper for each $\boldsymbol{W^A}$ and $\boldsymbol{U^A}$ is $O(md(N^A + 1))$ and $O(m(dN^A + C^A))$ respectively, and the complexity of training the model for target categories in Eq.(2) in the main paper is $O(md(N^T + 2))$ and $O(m(dN^T + C^T + |\mathcal{R}|))$ for $\boldsymbol{W}$ and $\boldsymbol{U}$. It is dominated by $O(mdN^T)$ as $dN^T \gg C^T + |\mathcal{R}|$.

To generate a query pool (Sec.3.2), we first compute the probability mass function (PMF) for each label entity by $O(NC)$, where $N = N^A + N^T, C = C^A + C^T$ and a confusion matrix of label entities by its PMF with the complexity of $O(N^A C^A + N^T C^T)$. A naive way of enumerating all possible constraints takes $O(C^T {C^A}^2)$ but we generate a decent sized subset (k_p) to consider the most confusing entities' nearest neighboring label embeddings (C_r) by $O(k_p C_r^2)$. Thus, the complexity of generating the pool is $O(NC + N^A C^A + N^T C^T + k_p C_r^2)$. Re-scoring the pool using cross validation takes $O(k_p mdN^T)$. Finally, the outer loop of algorithm usually iterates few times and thus the total complexity of Algorithm 1 is $O(N^A(md + C^A) + N^T(k_p md + C^T) + NC + k_p C_r^2)$.

Test time complexity is $O(m(C^T + d))$, which is the same for all linear embedding methods.

Score Vector for Estimating Classification Improvement

The score vector $\boldsymbol{c}$ consists of confidence/confusion of t, a_1 and a_2 on both training set and validation set, geometric fitness $\left(\frac{\|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|^2}{\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|^2}\right)$ and ball radius of sample distribution with respect to each class label prototype for t, a_1 and a_2.

Dataset Details

Low-Level Features. For visual features, we use the features provided by dataset authors (Lampert, Nickisch, and Harmeling 2014; Hwang, Grauman, and Sha 2013). The low-level features of both dataset is SIFT and other texture and color descriptors with PCA. In AWA dataset, we do PCA to reduce the dimensions to 300. In ImageNet-50, we use 1000 dimensional feature of same type of low level description to AWA dataset. We center the features by the sample mean.

Embedding Space Detail. For dimension of the embedding space, we choose 75, which is slightly bigger than the number classes (50) for encoding additional semantic information.

Animals with Attribute (AwA). There are 50 classes in total in AwA dataset (Lampert, Nickisch, and Harmeling 2014). Ten of them are target classes. The target classes of AwA dataset are 'Leopard', 'Pig', 'Hippopotamus', 'Seal', 'Persian Cat', 'Chimpanzee', 'Rat', 'Humpback Whale', 'Giant Panda' and 'Racoon'. The rest of the 40 classes of AwA serves as anchor classes.

ImageNet-50. There are 50 classes in total in the ImageNet-50 dataset (Hwang, Grauman, and Sha 2013). The 50 classes are randomly chosen from the entire ImageNet dataset. The 50 classes are: 'Kitfox', 'australianterrier', 'lesserpanda', 'egyptiancat', 'persiancat', 'cougar', 'badger', 'greatdane', 'scottishdeerhound', 'jaguar', 'blackfootedferret', 'skunk', 'corgi', 'weasel', 'colobus', 'orangutan', 'chimpanzee', 'gorilla', 'greyhound', 'hare', 'patas', 'baboon', 'macaque', 'tabby', 'raccoon', 'polecat', 'lion', 'cheetah', 'otter', 'sunflower', 'bonsai', 'strawberry', 'lamp', 'pooltable', 'acorn', 'drum', 'marimba', 'daisy', 'comb', 'rule', 'ferriswheel', 'rollercoaster', 'buckle', 'button', 'barnspider', 'gardenspider', 'bridge', 'featherboa', 'bathtub', 'basketball.'

Among them, we randomly choose ten of them are target classes. The target classes of ImageNet-50 dataset are 'cougar', 'weasel', 'colobus', 'gorilla', 'tabby', 'raccoon', 'pool-table', 'comb', 'roller-coaster', 'feather-boa'. The rest of the 40 classes of ImageNet-50 serves as anchor classes.

References

[Amit et al. 2007] Amit, Y.; Fink, M.; Srebro, N.; and Ullman, S. 2007. Uncovering Shared Structures in Multiclass Classification. In *ICML*.

[Hwang, Grauman, and Sha 2013] Hwang, S. J.; Grauman, K.; and Sha, F. 2013. Analogy-preserving semantic embedding for visual object categorization. In *International Conference on Machine Learning (ICML)*, 639–647.

[Lampert, Nickisch, and Harmeling 2014] Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans. on PAMI*.