In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

Inferring Analogous Attributes

Chao-Yeh Chen and Kristen Grauman
University of Texas at Austin
chaoyeh@cs.utexas.edu, grauman@cs.utexas.edu

Abstract

The appearance of an attribute can vary considerably from class to class (e.g., a "fluffy" dog vs. a "fluffy" towel), making standard class-independent attribute models break down. Yet, training object-specific models for each attribute can be impractical, and defeats the purpose of using attributes to bridge category boundaries. We propose a novel form of transfer learning that addresses this dilemma. We develop a tensor factorization approach which, given a sparse set of class-specific attribute classifiers, can infer new ones for object-attribute pairs unobserved during training. For example, even though the system has no labeled images of striped dogs, it can use its knowledge of other attributes and objects to tailor "stripedness" to the dog category. With two large-scale datasets, we demonstrate both the need for category-sensitive attributes as well as our method's successful transfer. Our inferred attribute classifiers perform similarly well to those trained with the luxury of labeled class-specific instances, and much better than those restricted to traditional modes of transfer.

1. Introduction

Attributes are visual properties that help describe objects or scenes [6, 12, 4, 13, 16], such as "fluffy", "glossy", or "formal". A major appeal of attributes is the fact that they appear across category boundaries, making it possible to describe an unfamiliar object class [4], teach a system to recognize new classes by zero-shot learning [13, 19, 16], or learn mid-level cues from cross-category images [12].

But are attributes really category-independent? Does fluffiness on a dog look the same as fluffiness on a towel? Are the features that make a high heeled shoe look formal the same as those that make a sandal look formal? In such examples (and many others), while the *linguistic* semantics are preserved across categories, the *visual* appearance of the property is transformed to some degree. That is, some attributes are specialized to the category.[1] This suggests

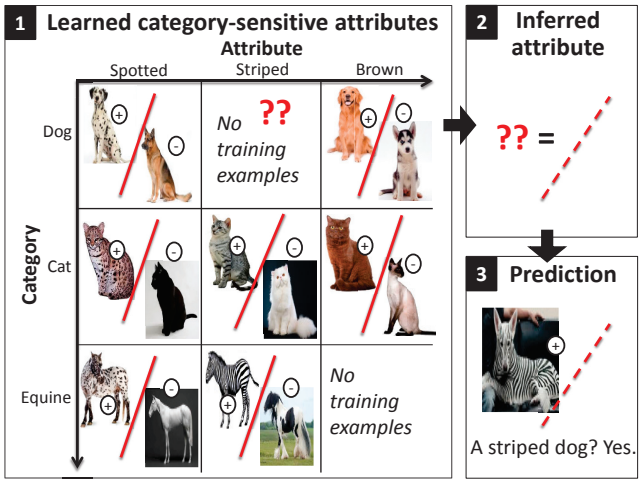

Figure 1. Having learned a sparse set of object-specific attribute classifiers, our approach infers *analogous attribute classifiers*. The inferred models are object-sensitive, despite having no object-specific labeled images of that attribute during training.

that simply pooling a bunch of training images of any object/scene with the named attribute and learning a discriminative classifier—the status quo approach—will weaken the learned model to account for the "least common denominator" of the attribute's appearance, and, in some cases, completely fail to generalize.

Accurate category-sensitive attributes would seem to require category-sensitive training. For example, we could gather positive exemplar images for each category+attribute combination (e.g., separate sets of fluffy dog images, fluffy towel images). If so, this is a disappointment. Not only would learning attributes in this manner be quite costly in terms of annotations, but it would also fail to leverage the common semantics of the attributes that remain in spite of their visual distinctions.

To resolve this problem, we propose a novel form of transfer learning to infer category-sensitive attribute models. Intuitively, even though an attribute's appearance may be specialized for a particular object, there likely are latent variables connecting it to other objects' manifestations of the property. Plus, some attributes *are* quite similar across

[1] We use "category" to refer to either an object or scene class.

some class boundaries (e.g., spots look similar on Dalmatian dogs and Pinto horses). Having learned some category-sensitive attributes, then, we ought to be able to predict how the attribute might look on a new object, *even without labeled examples depicting that object with the attribute*. For example, in Figure 1, suppose we want to recognize striped dogs, but we have no separate curated set of striped-dog exemplars. Having learned "spotted", "brown", etc. classifiers for dogs, cats, and equines, the system should leverage those models to infer what "striped" looks like on a dog. For example, it might infer that stripes on a dog look somewhat like stripes on a zebra but with shading influenced by the shape dogs share with cats.

Based on this intuition, we show how to infer an *analogous attribute*—an attribute classifier that is tailored to a category, even though we lack annotated examples of that category exhibiting that attribute. Given a sparse set of category-sensitive attribute classifiers, our approach first discovers the latent structure that connects them, by factorizing a tensor indexed by categories, attributes, and classifier dimensions. Then, we use the resulting latent factors to complete the tensor, inferring the "missing" classifier parameters for any object+attribute pairings unobserved during training. As a result, we can create category-sensitive attributes with only partial category-sensitive labeled data. Our solution offers a middle ground between completely category-independent training (the norm today [12, 4, 13, 19, 16, 17]) and completely category-sensitive training. We don't need to observe all attributes isolated on each category, and we capitalize on the fact that some categories and some of their attributes share common parameters.

Compared to existing forms of transfer learning, our idea has three key novel elements. First, performing transfer jointly in the space of two labeled aspects of the data—namely, categories and attributes—is new. Critically, this means our method is not confined to transfer along same-object or same-attribute boundaries; rather, it discovers analogical relationships based on some mixture of previously seen objects and attributes. Second, our approach produces a discriminative model for an attribute with zero training examples from that category. Third, while prior methods often require information about which classes should transfer to which [2, 29, 26, 1] (e.g., that a motorcycle detector might transfer well to a bicycle), our approach naturally discovers where transfer is possible based on how the observed attribute models relate. It can transfer easily between multiple classes at once, not only pairs, and we avoid the guesswork of manually specifying where transfer is likely.

We validate our approach on two large-scale attribute datasets, SUN [17] and ImageNet [19], to explore both object-sensitive and scene-sensitive attributes. We first demonstrate that category-sensitive attributes on the whole outperform conventional class-independent models. Then we show that our method accurately infers analogous attribute models, in spite of never seeing labeled examples for that property and class. Furthermore, we show its advantages over applying traditional forms of transfer learning that fail to account for the intrinsic 2D nature of the object-attribute label space.

2. Related Work

The standard approach to learn an attribute is to pool images regardless of their object category and train a discriminative classifier [12, 4, 13, 19, 16, 17]. While this design is well-motivated by the goal of having attributes that transcend category boundaries, it sacrifices accuracy in practice, as we will see below. We are not aware of any prior work that learns category-sensitive attributes, though class-specific attribute training is used as an intermediate feature generation procedure in [4, 27], prior to training class-independent models.

While attribute learning is typically considered separately from object category learning, some recent work explores how to jointly learn attributes and objects, either to exploit attribute correlations [27], to promote feature sharing [25, 9], or to discover separable features [30, 20]. Our framework can be seen as a new way to jointly learn multiple attributes, leveraging structure in object-attribute relationships. Unlike any prior work, we use these ties to directly infer category-sensitive attribute models without labeled exemplars.

In [8], analogies between object categories are used to regularize a semantic label embedding. Our method also captures beyond-pairwise relationships, but the similarities end there. In [8], explicit analogies are given as input, and the goal is to enrich the features used for nearest neighbor object recognition. In contrast, our approach implicitly *discovers* analogical relationships among *object-sensitive attribute classifiers*, and our goal is to generate novel category-sensitive attribute classifiers.

In vision, factorized models have been used for various problems, from bi-linear models for separating style and content [7], to multi-linear models separating the modes of face image formation (e.g., identity vs. expression vs. pose) [22, 24]. While often applied for visualization, the discovered factors can also be used to impute missing data—for example, to generate images of novel fonts [7] or infer missing pixels for in-painting tasks [15]. Tensor completion is an area of active research in machine learning, and forms the basis of modern recommender systems to infer missing labels (e.g., movie ratings) [11, 28]. In contrast, we use tensor factorization to infer *classifiers*, not data instances or labels. This enables a new "zero-shot" transfer protocol: we leverage the latent factors underlying previously trained models to create new analogous ones without

any labeled instances.[2]

Transfer learning has been explored for object recognition [5, 2, 29, 18, 26, 21, 14, 1], where the goal is to learn a new object category with few labeled instances by exploiting its similarity to previously learned class(es). While often the source and target classes must be manually specified [2, 26, 1], some techniques automatically determine which classes will benefit from transfer [21, 14, 10]. In our setting the motivation to reduce labeled data requirements is as much about data availability as labeling cost: it can be difficult to obtain sufficient category-specific images for each possible attribute, even if we did not mind the labeling effort. More importantly, as discussed above, our idea for transfer learning jointly in two label spaces is new, and, unlike the prior work, we can infer new classifiers without training examples.

3. Approach

Given training images labeled by their category and one or more attributes, our method produces as output a series of category-sensitive attribute classifiers. Some of those classifiers are explicitly trained with the labeled data, while the rest are inferred by our method. We show how to create these analogous attribute classifiers via tensor completion.

In the following, we first describe how we train category-sensitive classifiers (Sec. 3.1). Then we define the tensor of attributes (Sec. 3.2) and show how we use it to infer analogous models (Sec. 3.3). Finally, we discuss certain salient aspects of the method design (Sec. 3.4).

3.1. Learning Category-Sensitive Attributes

In existing systems, attributes are trained in a category-independent manner [12, 4, 13, 19, 16, 17]. Positive exemplars consist of images from various object categories, and they are used to train a discriminative model to detect the attribute in novel images. We will refer to such attributes as *universal*.

In this work, we challenge the convention of learning attributes in a completely category-independent manner. As discussed above, while attributes' visual cues are often shared among *some* objects, the sharing is not universal. It can dilute the learning process to pool cross-category exemplars indiscriminately.

The naive solution to instead train *category-sensitive* attributes would be to partition training exemplars by their category labels, and train one attribute per category. Were labeled examples of all possible attribute+object combinations abundantly available, such a strategy might be sufficient. However, in initial experiments with large-scale datasets, we found that this approach is actually inferior to training a single universal attribute. We attribute this to two things: (1) even in large-scale collections, the long-tailed distribution of object/scene/attribute occurrences in the real world means that some label pairs will be undersampled, leaving inadequate exemplars to build a statistically sound model, and (2) this naive approach completely ignores attributes' inter-class semantic ties.

To overcome these shortcomings, we instead use an importance-weighted support vector machine (SVM) to train each category-sensitive attribute. Let each training example $(\boldsymbol{x}_i, y_i)$ consist of an image descriptor $\boldsymbol{x}_i \in \Re^D$ and its binary attribute label $y_i \in \{-1, 1\}$. Suppose we are learning "furriness" for dogs. We use examples from all categories (dogs, cats, etc.), but place a higher penalty on violating attribute label constraints for the same category (the dog instances). This amounts to an SVM objective for the hyperplane $\boldsymbol{w}$:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}||\boldsymbol{w}||^2 + C_s \sum_i \xi_i + C_o \sum_j \gamma_j \right) \quad (1) \\ s.t. \quad & y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1 - \xi_i; \quad \forall i \in \mathcal{S} \\ & y_j \boldsymbol{w}^T \boldsymbol{x}_j \geq 1 - \gamma_j; \quad \forall j \in \mathcal{O} \\ & \xi_i \geq 0; \gamma_j \geq 0, \end{aligned}$$

where the sets $\mathcal{S}$ and $\mathcal{O}$ denote those training instances in the same-class (dog) and other classes (non-dogs), respectively, and C_s and C_o are slack penalty constants. Note, $\mathcal{S}$ and $\mathcal{O}$ contain both positive and negative examples for the attribute in consideration.

Instance re-weighting is commonly used, e.g., to account for label imbalance between positives and negatives. Here, by setting $C_o < C_s$, the out-of-class examples of the attribute serve as a simple prior for which features are relevant. This way we benefit from more training examples when there are few category-specific examples of the attribute, but we are inclined to ignore those that deviate too far from the category-sensitive definition of the property. As we will see in results, these models typically outperform their universal counterparts.

3.2. Object-Attribute Classifier Tensor

Next we define a tensor to capture the structure underlying many such category-sensitive models. Let $m = 1, \ldots, M$ index the M possible attributes in the vocabulary, and let $n = 1, \ldots, N$ index the N possible object/scene categories. Let $\boldsymbol{w}(n, m)$ denote a category-sensitive SVM weight vector trained for the n-th object and m-th attribute using Eqn. 1.

We construct a 3D tensor $\mathbf{W} \in \Re^{N \times M \times D}$ using all available category-sensitive models. Each entry w_{nm}^d contains the value of the d-th dimension of the classifier $\boldsymbol{w}(n, m)$. For a linear SVM, this value reflects the impact of

[2] This is not to be confused with zero-shot learning in [13], where unseen objects are learned by listing their attributes.

the d-th dimension of the feature descriptor $\boldsymbol{x}$ for determining the presence/absence of attribute m for the object class n. To use non-linear SVM classifiers, we use the efficient kernel map approach of [23], which computes explicit linear embeddings for additive kernels, including the intersection and χ^2 kernels commonly used in visual recognition. This lets us maintain an explicit tensor $\mathbf{W}$ while still benefitting from more powerful non-linear classifiers.[3] In this case, D is the dimension of the feature map embedding, and all else is the same. We test both variants in our experiments.

The resulting tensor is quite sparse. We can only fill entries for which we have class-specific positive and negative training examples for the attribute of interest. In today's most comprehensive attribute datasets [19, 17], this means only $\sim 25\%$ of the possible object-attribute combinations can be trained in a category-sensitive manner. Rather than resort to universal models for those "missing" combinations, we propose to use the latent factors for the observed classifiers to synthesize analogous models for the unobserved classifiers, as we explain next.

3.3. Inferring Analogous Attributes

Having learned how certain attributes look for certain object categories, our goal is to transfer that knowledge to hypothesize how the same attributes will look for other object categories. In this way, we aim to infer analogous attributes: category-sensitive attribute classifiers for objects that lack attribute-labeled data. We pose the "missing classifier" problem as a tensor completion problem. We recover the latent factors for the 3D object-attribute tensor $\mathbf{W}$, and use them to impute the unobserved classifier parameters.

Let $\mathbf{O} \in \Re^{K \times N}$, $\mathbf{A} \in \Re^{K \times M}$, and $\mathbf{C} \in \Re^{K \times D}$ denote matrices whose columns are the K-dimensional latent feature vectors for each object, attribute, and classifier dimension, respectively. We assume that w^d_{nm} can be expressed as an inner product of latent factors,

$$w^d_{nm} \approx \langle O_n, A_m, C_d \rangle, \tag{2}$$

where a subscript denotes a column of the matrix. In matrix form, we have $\mathbf{W} \approx \sum_{k=1}^{K} O^k \circ A^k \circ C^k$, where a superscript denotes the row in the matrix, and $\circ$ denotes the vector outer product.

The latent factors of the tensor $\mathbf{W}$ are what affect how the various attributes, objects, and image descriptors covary. What might they correspond to? We expect some will capture mixtures of two or more attributes, e.g., factors distinguishing how "spots" appear on something "flat" vs. how they appear on something "bumpy". The latent factors can also capture useful clusters of objects, or supercategories, that exhibit attributes in common ways. Some might capture other attributes beyond the M portrayed in the training images—namely, those that help explain structure in the objects and other attributes we have observed.

We use Bayesian probabilistic tensor factorization [28] to recover the latent factors. Using this model, the likelihood for the explicitly trained classifiers (Sec. 3.1) is

$$p(\mathbf{W}|\mathbf{O}, \mathbf{A}, \mathbf{C}, \alpha) = \Pi_{n=1}^{N} \Pi_{m=1}^{M} \Pi_{d=1}^{D} \left[\mathcal{N}(w^d_{nm}|\langle O_n, A_m, C_d \rangle, \alpha^{-1})\right]^{I_{nm}},$$

where $\mathcal{N}(w|\mu, \alpha)$ denotes a Gaussian with mean μ and precision α, and $I_{nm} = 1$ if object n has an explicit category-sensitive model for attribute m, and $I_{nm} = 0$ otherwise. For each of the latent factors O_n, A_m, and C_d, we use Gaussian priors. Let Θ represent all their means and covariances. Following [28], we compute a distribution for each missing tensor value by integrating out over all model parameters and hyper-parameters, given all the observed attribute classifiers:

$$p(\hat{w}^d_{nm}|\mathbf{W}) = \int p(\hat{w}^d_{nm}|O_n, A_m, C_d, \alpha) p(\mathbf{O}, \mathbf{A}, \mathbf{C}, \alpha, \Theta|\mathbf{W}) \; d\{\mathbf{O}, \mathbf{A}, \mathbf{C}, \alpha, \Theta\}.$$

After initializing with the MAP estimates of the three factor matrices, this distribution is approximated using Markov chain Monte Carlo (MCMC) sampling:

$$p(\hat{w}^d_{nm}|\mathbf{W}) \approx \sum_{l=1}^{L} p(\hat{w}^d_{nm}|O^{(l)}_n, A^{(l)}_m, C^{(l)}_d, \alpha^{(l)}). \tag{3}$$

Each of the L samples $\{O^{(l)}_n, A^{(l)}_m, C^{(l)}_d, \alpha^{(l)}\}$ is generated with Gibbs sampling on a Markov chain whose stationary distribution is the posterior over the model parameters and hyper-parameters. We use conjugate distributions as priors for all the Gaussian hyper-parameters to facilitate sampling. See [28] for details.

We use these factors to generate analogous attributes. Suppose we have no labeled examples showing an object of category n with attribute m (or, as is often the case, we have so few that training a category-sensitive model is problematic). Despite having no training examples, we can use the tensor to directly infer the classifier parameters

$$\hat{\boldsymbol{w}}(n, m) = [\hat{w}^1_{nm}, \ldots, \hat{w}^D_{nm}], \tag{4}$$

where each $\hat{w}^d_{nm}$ is the mean of the distribution in Eq. (3).

Our method is quite efficient. For the datasets in Sec. 4, training all explicit category-sensitive models takes around 5 minutes. Factorizing the tensor with $M = 59$ and $N = 280$ and $D = 512$ takes around 180 seconds. Then inferring a new attribute classifier takes 0.05 seconds.

3.4. Discussion

We stress that while tensor completion itself is certainly not new, prior work in vision [15, 7, 22, 24] and data mining (e.g., [11, 28]) focuses on inferring missing *data* instances

[3] Alternatively, kernelized factorization methods could be applied.

or missing *labels*. For example, for data problems, the tensor could be a corrupted video in which one wants to inpaint missing voxels [15]; for missing label problems, the tensor could be the movie ratings given by different users for various films over time, and one wants to guess how a user would rate a new movie [28].

In contrast, we propose to use factorization to infer *classifiers* within a tensor representing two inter-related label spaces. Our idea has two key useful implications. First, it leverages the interplay of both label spaces to generate new classifiers without seeing any labeled instances. This is a novel form of transfer learning. Second, by working directly in the classifier space, we have the advantage of first isolating the low-level image features that are informative for the observed attributes. This means the input training images can contain realistic (un-annotated) variations. In comparison, existing data tensor approaches often assume a strict level of alignment; e.g., for faces, examples are curated under n specific lighting conditions, m specific expressions, etc. [22, 24].

Our design also means that the analogous attributes can transfer information from multiple objects and/or attributes simultaneously. That means, for example, our model is not restricted to transferring the fluffiness of a dog from the fluffiness of a cat; rather, its analogous model for dog fluffiness might just as well result from transferring a mixture of cues from carpet fluffiness, dog spottedness, and cat shape.

In general, transfer learning can only succeed if the source and target classes are related. Similarly, we will only find an accurate low-dimensional set of factors if some common structure exists among the explicitly trained category-sensitive models. Nonetheless, a nice property of our formulation is that even if the tensor is populated with a variety of classes—some with no ties—analogous attribute inference can still succeed. Distinct latent factors can cover the different clusters in the observed classifiers. For similar reasons, our approach naturally handles the question of "where to transfer": sources and targets are never manually specified. Below, we consider the impact of building the tensor with a large number of semantically diverse categories versus a smaller number of closely related categories.

4. Experimental Results

The experiments analyze four main aspects: (1) how category-sensitive attributes compare to standard universal attributes (Sec. 4.1), (2) how well our inferred attributes compete with the upper bound category-sensitive attributes trained explicitly with images, and compared to a traditional transfer approach (Sec. 4.2), (3) the impact of focusing the tensor on closely related classes (Sec. 4.3), and (4) the feasibility of inferring non-linear models (Sec. 4.4).

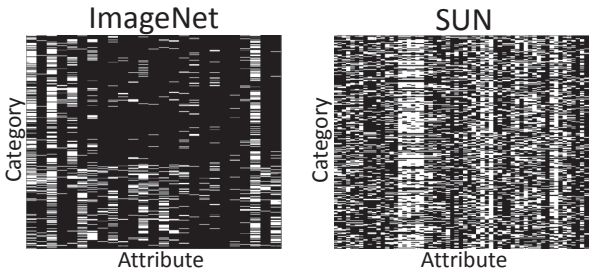

Figure 2. Data availability: white entries denote category-attribute pairs that have positive and negative image exemplars. In ImageNet, most vertical stripes are color attributes, and most horizontal stripes are man-made objects. In SUN, most vertical stripes are attributes that appear across different scenes, such as vacationing or playing, while horizontal stripes come from scenes with varied properties, such as airport and park.

Datasets and features We evaluate our approach on two datasets: the attribute-labeled portion of ImageNet [19] and SUN Attributes [17]. ImageNet contains 9,600 total images, with 384 object categories and 25 attributes describing color, patterns, shape, and texture. SUN contains 14,340 total images, with 717 scene categories and 102 attributes describing global properties, activity affordances, materials, and basic textures. We use all 280 categories and 59 attributes for which SUN contains both positive and negative examples for the scene-attribute pair. For both datasets, we use features provided by the authors. For ImageNet, we concatenate color histograms, SIFT bag of words, and shape context ($D = 1550$). For SUN, we use GIST ($D = 512$).

The datasets do not contain data for all possible category-attribute pairings. Figure 2 shows which are available: there are 1,498 and 6,118 pairs in ImageNet and SUN, respectively. The sparsity of these matrices actually underscores the need for our approach, if one wants to learn category-sensitive attributes.

We split both datasets in half for training and testing. When explicitly training an attribute, we randomly sample $S\%$ of the images from all other categories ($S = 50\%$ for ImageNet and $S = 10\%$ for SUN, proportional to their sizes). We use $L = 100$ samples and fix the number of latent factors $K = 30$, following [28]. We set the slack penalties $C_o = 0.1$ and $C_s = 1$. We did not tune these values. Unless otherwise noted, all methods use linear SVMs.

4.1. Category-Sensitive vs. Universal Attributes

First we test whether category-sensitive attributes are even beneficial. We explicitly train category-sensitive attribute classifiers using importance-weighted SVMs, as described in Sec. 3.1. This yields 1,498 and 6,118 classifiers for ImageNet and SUN, respectively. We compare their predictions to those of universal attributes, where we train one model for each attribute ($M = 25$ for ImageNet

	Datasets		Trained explicitly		Trained via transfer			
	# Categ (N)	# Attr (M)	Category-sens.	Universal	Inferred (Ours)	Adopt similar	One-shot	Chance
ImageNet	384	25	**0.7304**	0.7143	**0.7259**	0.6194	0.6309	0.5183
SUN	280	59	**0.6505**	0.6343	**0.6429**	N/A	N/A	0.5408

Table 1. Accuracy (mAP) of attribute prediction. Category-sensitive models improve over standard universal models, and our inferred classifiers nearly match their accuracy with no training image examples. Traditional forms of transfer (rightmost two columns) fall short, showing the advantage of exploiting the 2D label space for transfer, as we propose. These results are averages over thousands of attributes; category-sensitive attributes achieve an average gain of 0.15 in AP in 76% of the cases.

and $M = 59$ for SUN). When learning an attribute, both models have access to the exact same images; the universal method ignores the category labels, while the category-sensitive method puts more emphasis on the in-category examples.[4] We evaluate both methods on the same test set.

Table 1 (cols 4 and 5) shows the results, in terms of mean average precision across all 84 attributes and 664 categories. Among those, our category-sensitive models meet or exceed the universal approach 76% of the time, with average increases of 0.15 in AP, and gains of up to 0.83 in AP for some attributes. This indicates that the status quo [12, 4, 13, 19, 16, 17] pooling of training images across categories is indeed detrimental.

4.2. Inferring Analogous Attributes

The results so far establish that category-sensitive attributes are desirable. However, the explicit models above are *impossible to train for 18K of the ∼26K possible attributes in these datasets.* This is where our method comes in. It can infer all remaining $18K$ attribute models even without class-specific labeled training examples.

We perform leave-one-out testing: in each round, we remove one observed classifier (a white entry in Figure 2), and infer it with our tensor factorization approach. Note that even though we are removing one at a time, the full tensor is always quite sparse due to the available data. Namely, only 16% (in ImageNet) and 37% (in SUN) of all possible category-sensitive classifiers can be explicitly trained.

Table 1 (cols 4 to 6) shows this key result. In this experiment, the explicitly trained category-sensitive result is the "upper bound"; it shows how well the model trained with real category-specific images can do. We see that our inferred analogous attributes (col 6) are nearly as accurate, yet use zero category-specific labeled images. They approximate the explicitly trained models well. Most importantly, our inferred models remain more accurate than the universal approach. Our inferred attributes again meet or exceed the universal model's accuracy 79% of the time, with gains averaging 0.13 in AP.

We stress that our method infers models for *all* missing attributes. That is, using the explicitly trained attributes, it infers another $8,064$ and $10,407$ classifiers on ImageNet and SUN, respectively. While the category-sensitive method would require ~ 20 labeled examples per classifier to train those models, our method uses zero. *That amounts to saving* $348K$ *total labeled images.* That in turn means saving $17,400 in labeling costs, if we were to pay $0.05 per image for MTurkers to both track down and label images exhibiting all those class-attribute pairings. (Due to ground truth availability, though, we can only validate against the held-out attributes.)

The results so far presume we know which category's attribute model to apply to a novel image. If we further require the category to be predicted automatically—by marginalizing over the category label to estimate the attribute probability—our results remain similar. In particular, the explicit category-sensitive results (col 4 of Table 1) become 0.7249 and 0.6419, and the inferred results (col 6) become 0.7218 and 0.6401—still better than universal.

Table 1 also compares our approach to conventional transfer learning. The first transfer baseline infers the missing classifier simply by adopting the category-sensitive attribute of the category that is semantically closest to it, where semantic distance is measured via WordNet using [3] (not available for SUN). For example, if there are no furry-dog exemplars, we adopt the wolf's "furriness" classifier. The second transfer baseline additionally uses one category-specific image example to perform "one-shot" transfer (e.g., it trains with both the furry-wolf images plus a furry-dog example).[5] Unlike the transfer baselines, our method uses neither prior knowledge about semantic distances nor labeled class-specific examples. We see that our approach is substantially more accurate than both transfer methods. This result highlights the benefit of our novel approach to transfer, which leverages both label spaces (categories and their attributes) simultaneously.

Which attributes does our method transfer? That is, which objects does it find to be analogous for an attribute? To examine this, we first take a category j and identify its neighboring categories in the latent feature space, i.e., in terms of Euclidean distance among the columns of $\mathbf{O} \in \Re^{K \times N}$. Then, for each neighbor i, we sort its attribute classifiers ($\boldsymbol{w}(i,:)$, real or inferred) by their maximal cosine similarity to any of category j's attributes $\boldsymbol{w}(j,:)$. The resulting shortlist helps illustrate which attribute+category pairs our method expects to transfer to category j.

[4] So the universal model also uses category-specific images. We find it performs similarly whether it uses them or not.

[5] We also tried an Adaptive SVM [29] for the transfer baseline, but it was weaker than the results reported above.

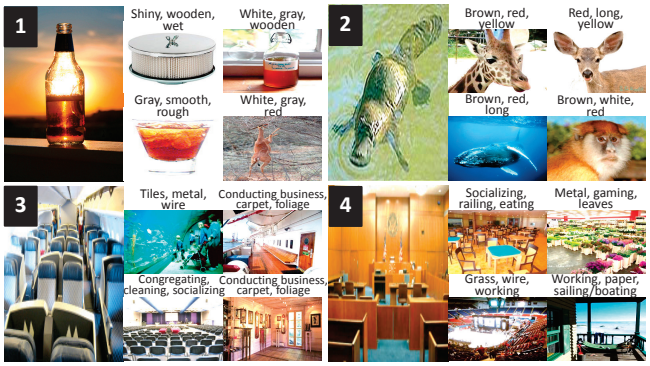

Figure 3. Analogous attribute examples for ImageNet (top) and SUN (bottom). Words above each neighbor indicate the 3 most similar attributes (learned or inferred) between leftmost query category and its neighboring categories in latent space. Query category:neighbor category = 1.Bottle: filter, syrup, bullshot, gerenuk. 2.Platypus: giraffe, ungulate, rorqual, patas. 3.Airplane cabin: aquarium, boat deck, conference center, art studio. 4.Courtroom: cardroom, florist shop, performance arena, beach house.

Figure 3 shows 4 such examples, with one representative image for each category. We see neighboring categories in the latent space are often semantically related (e.g., syrup/bottle) or visually similar (e.g., airplane cabin/conference center); although our method receives no explicit side information on semantic distances, it discovers these ties through the observed attribute classifiers. Some semantically more distant neighbors (e.g., platypus/rorqual, courtroom/cardroom) are also discovered to be amenable to transfer. The words in Figure 3 are the neighboring categories' top 3 analogous attributes for the numbered category to their left (*not* attribute predictions for those images). It seems quite intuitive that these would be suited for transfer.

Next we look more closely at where our method succeeds and fails. Figure 4 shows the top (bottom) five category+attribute combinations for which our inferred classifiers most increase (decrease) the AP, per dataset. As expected, we see our method most helps when the visual appearance of the attribute on an object is quite different from the common case, such as "spots" on the killer whale. On the other hand, it can detract from the universal model when an attribute is more consistent in appearance, such as "black", or where more varied examples help capture a generic concept, such as "symmetrical".

Figure 5 shows qualitative examples that support these findings. We show the image for each method that was predicted to most confidently exhibit the named attribute. By inferring analogous attributes, we better capture object-specific properties. For example, while our method correctly fires on a "smooth wheel", the universal model mistakes a Ferris Wheel as "smooth", likely due to the smoothness of the background, which might look like other classes' instantiations of smoothness.

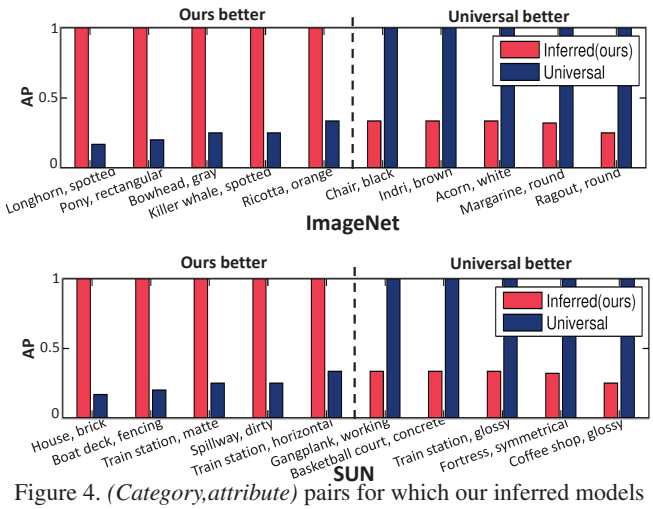

Figure 4. *(Category,attribute)* pairs for which our inferred models most improve (left) or hurt (right) the universal baseline.

Figure 5. Test images that our method (top row) and the universal method (bottom row) predicted most confidently as having the named attribute. (✓ = positive for the attribute, X = negative, according to ground truth.)

4.3. Focusing on Semantically Close Data

In all results so far, we make no attempt to restrict the tensor to ensure semantic relatedness. The fact our method succeeds in this case indicates that it is capable of discovering clusters of classifiers for which transfer is possible, and is fairly resistant to negative transfer.

Still, we are curious whether restricting the tensor to classes that have tight semantic ties could enhance performance. We therefore test two variants: one where we restrict the tensor to closely related objects (i.e., downsampling the rows), and one where we restrict it to closely related attributes (i.e., downsampling the columns). To select a set of closely related objects, we use WordNet to extract sibling synsets for different types of dogs in ImageNet. This yields 42 categories, such as *puppy, courser, coonhound, corgi*. To select a set of closely related attributes, we extract only the color attributes.

Table 2 shows the results. We use the same leave-one-out protocol of Sec. 4.2, but during inference we only consider category-sensitive classifiers among the selected categories/attributes. We see that the inferred attributes are stronger with the category-focused tensor, raising accuracy from 0.7173 to 0.7358, closer to the upper bound. This sug-

Subset	Category-sensitive	Inferred (subset)	Inferred (all)
Categories (dogs)	0.7478	0.7358	0.7173
Attributes (colors)	0.7665	0.7631	0.7628

Table 2. Attribute label prediction mAP when restricting the tensor to semantically close classes. The explicitly trained category-sensitive classifiers serve as an upper bound.

	Category-sensitive	Inferred	Universal
linear SVM	0.7304	0.7259	0.7143
χ^2 SVM	0.7589	0.7428	0.7037

Table 3. Using kernel maps [23] to infer non-linear SVMs.

gests that among the entire dataset, attributes for which categories differ can introduce some noise into the latent factors. On the other hand, when we ignore attributes unrelated to color, the mAP of the inferred classifiers remains similar. This may be because color attributes use such a distinct set of image features compared to others (like stripes, round) that the latent factors accounting for them are coherent with or without the other classifiers in the mix. From this preliminary test, we can conclude that when semantic side information is available, it could boost accuracy, yet our method achieves its main purpose even when it is not.

4.4. Inferring Non-linear Classifiers

Finally, we demonstrate that our approach is not limited to inferring linear classifiers. We use the homogeneous kernel map [23] of order 3 to approximate a χ^2 kernel non-linear SVM. This entails mapping the original features to a space in which an inner product approximates the χ^2 kernel. Using the kernel maps, we repeat the experiment of Sec. 4.2. Table 3 shows the results on ImageNet. The non-linear classifiers boost accuracy for both the explicit and inferred category-sensitive attributes. Unexpectedly, we find the kernel map SVM decreases accuracy slightly for the universal approach; perhaps due to overfitting.

5. Conclusions

We introduced a new form of transfer learning, in which analogous classifiers are inferred using observed classifiers organized according to two inter-related label spaces. We developed a tensor factorization approach that solves the transfer problem, even when no training examples are available for the decision task of interest.

Our work highlights the reality that many attributes are not strictly category-independent. We offer a practical tool to ensure category-sensitive models can be trained even if category-specific labeled datasets are not possible. As demonstrated through multiple experiments with two large-scale datasets, the idea seems quite promising.

In future work, we will explore one-shot extensions of analogous attributes, and analyze their impact for learning relative properties.

Acknowledgements This research is supported in part by NSF CAREER IIS-0747356.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.
[2] E. Bart and S. Ullman. Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement. In *CVPR*, 2005.
[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.
[4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009.
[5] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003.
[6] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *NIPS*, 2007.
[7] W. T. Freeman and J. B. Tenenbaum. Learning bilinear models for two-factor problems in vision. In *CVPR*, 1997.
[8] S. J. Hwang, K. Grauman, and F. Sha. Analogy-preserving semantic embedding for visual object categorization. In *ICML*, 2013.
[9] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *CVPR*, 2011.
[10] L. Jacob, F. Bach, and J. Vert. Clustered multi-task learning: a convex formulation. In *NIPS*, 2008.
[11] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.
[12] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009.
[13] C. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009.
[14] J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2002.
[15] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *ICCV*, 2009.
[16] D. Parikh and K. Grauman. Relative Attributes. In *ICCV*, 2011.
[17] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.
[18] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*, 2008.
[19] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *ECCV Workshop on Parts and Attributes*, 2010.
[20] V. Sharmanska, N. Quadrianto, and C. Lampert. Augmented attributes representations. In *ECCV*, 2012.
[21] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010.
[22] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, 2002.
[23] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.
[24] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans Graphics*, 24(3):426–433, 2005.
[25] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.
[26] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *CVPR*, 2010.
[27] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.
[28] L. Xiong, X. Chen, T. Huang, J. Schneider, and J. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *SDM*, 2010.
[29] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.
[30] F. Yu, L. Cao, R. Feris, J. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.