Learning Hierarchical Linguistic Descriptions of Visual Datasets

Roni Mittelman†, **Min Sun**‡, **Benjamin Kuipers**†, **Silvio Savarese**†
† Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor
‡ Department of Computer Science and Engineering, University of Washington, Seatle
rmittelm,kuipers,silvio@umich.edu, sunmin@cs.washington.edu

Abstract

We propose a method to learn succinct hierarchical linguistic descriptions of visual datasets, which allow for improved navigation efficiency in image collections. Classic exploratory data analysis methods, such as agglomerative hierarchical clustering, only provide a means of obtaining a tree-structured partitioning of the data. This requires the user to go through the images first, in order to reveal the semantic relationship between the different nodes. On the other hand, in this work we propose to learn a hierarchy of linguistic descriptions, referred to as attributes, which allows for a textual description of the semantic content that is captured by the hierarchy. Our approach is based on a generative model, which relates the attribute descriptions associated with each node, and the node assignments of the data instances, in a probabilistic fashion. We furthermore use a nonparametric Bayesian prior, known as the tree-structured stick breaking process, which allows for the structure of the tree to be learned in an unsupervised fashion. We also propose appropriate performance measures, and demonstrate superior performance compared to other hierarchical clustering algorithms.

1 Introduction

With the abundance of images available both for personal use and in large internet based datasets, such as Flickr and Google Image Search, hierarchies of images are an important tool that allows for convenient browsing and efficient search and retrieval. Intuitively, desirable hierarchies should capture similarity in a semantic space, i.e. nearby nodes should include categories which are semantically more similar, as compared to nodes which are more distant. Recent works that are concerned with learning image hierarchies (Bart et al., 2011; Sivic et al., 2008), have relied on a bag of visual-words feature space, and therefore have been shown to provide unsatisfactory results with respect to the latter requirement (Li et al., 2010).

A recent trend in visual recognition systems, has been to shift from using a low-level feature based representation to an attribute based feature space, which can capture higher level semantics (Farhadi et al., 2009; Lampert et al., 2009; Parikh & Grauman, 2011; Berg et al., 2011; Ferrari & Zisserman, 2007). Attributes are detectors that are trained using annotation data, to identify particular properties in an instance image. By evaluating these detectors on a query image, one can obtain a linguistic description of the image. Therefore, learning a visual hierarchy based on an attribute representation can allow for an improved semantic grouping, as compared to the previous use of low-level image features.

In this work we wish to utilize an attribute based representation to learn a hierarchical linguistic description of a visual dataset, in which few attributes are associated with each node of the tree. As is illustrated in Figure 1, such an attribute hierarchy is tightly related to a category hierarchy, in which the instances associated with every node are described using all the attributes associated with all the nodes along the path leading up to the root node (the instances in Figure 1 are described by the corresponding photographs). This “duality” between the at-

Proceedings of the NAACL HLT Workshop on Vision and Language (WVL '13), pages 20–28,
Atlanta, Georgia, 14 June 2013. ©2013 Association for Computational Linguistics

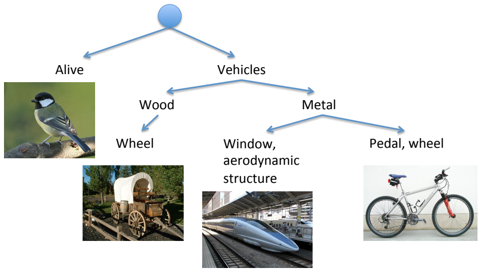


Figure 1: Attribute and category hierarchies.

tribute and category hierarchies, offers an important advantage when characterizing the dataset to the end-user, since it eliminates the need to visually inspect the images assigned to each node, in order to reveal the semantic relationship between the categories that are associated with the different nodes. Exploratory data analysis methods for learning hierarchies, such as agglomerative hierarchical clustering (AHC) (Jain & Dubes, 1988 p. 59), only assign instances to different nodes in the tree, whereas our approach learns an attribute hierarchy which is used to assign the instances to the nodes of the tree. The attribute hierarchy provides a linguistic description of a category hierarchy.

We develop a generative model, which we refer to as the attribute tree process (ATP), and which ties together the attribute and category hierarchies in a probabilistic fashion. The tree structure is learned by incorporating a nonparametric Bayesian prior, known as the tree-structured stick breaking process (TSSBP) (Adams et al., 2010), in the probabilistic formulation. An important observation which we make about the attribute hierarchies which are learned using the ATP, is that attributes which are related to more image instances tend to be associated with nodes which are closer to the root, and vice versa, attributes which are associated with fewer instances tend to be associated with leaf nodes. A hierarchical clustering algorithm that is based on the TSSBP for binary feature vectors was developed in (Adams et al., 2010), and is known as the factored Bernoulli likelihood model (FBLM). However, similarly to AHC, it does not produce the attribute hierarchy in which we are interested.

In order to evaluate the ATP quantitatively, we compare its performance to other hierarchical clustering algorithms. If the ground truth of the category hierarchy is available, we propose to use the semantic distance between the categories, that is given by the ground truth hierarchy, to evaluate the degree to which the semantic distance between the categories is preserved by the hierarchical clustering algorithm. If the ground truth is not available, we use two criteria, which as we argue, capture the properties that should be demonstrated by desirable semantic hierarchies. The first is the “purity criterion” (Manning et al., 2009 p. 357) which measures the degree to which each node is occupied by instances from a single class, and the second is the “locality criterion” which we propose, and which measures the degree to which instances from the same class are assigned to nearby nodes in the hierarchy. Our experimental results show that when compared to AHC and FBLM, our approach captures the ground truth semantic distance between the categories more accurately, and without significant dependence on hyperparameters.

The remaining of this paper is organized as follows. In Sec. 2 we provide background on agglomerative hierarchical clustering, and on the TSSBP, and in Sec. 3 we develop the generative model for the ATP. In Sec. 4 we propose evaluation metrics for the attribute hierarchy, and in Sec. 5 we present the experimental results. Sec. 6 concludes this paper.

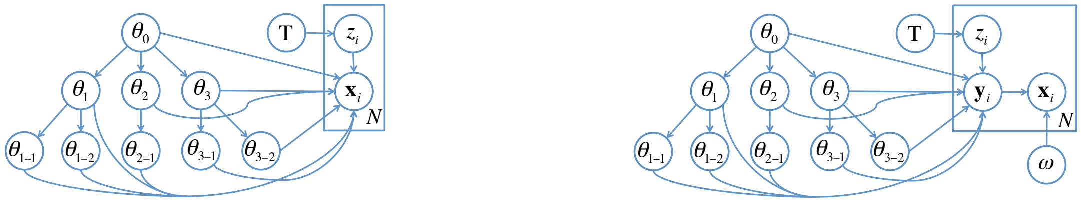
2 Background

2.1 Agglomerative hierarchical clustering

AHC uses a bottom up approach to clustering. In the first iteration, each cluster includes a single instance of the dataset, and at each following iteration, the two clusters which are closest to each other are joined into a single cluster. This requires a distance metric, which measures the distance between clusters, to be defined. The algorithm concludes when the distance between the farthest clusters is smaller than some threshold.

2.2 Tree structured stick breaking process

The TSSBP is an infinite mixture model, where each mixture component has one-to-one correspondence with one of the nodes of an infinitely branching and infinitely deep tree. The weights of the infinite mixture model are generated by interleaving two stick-breaking processes (Teh et al., 2006), which allows the number of mixture components to be inferred



(a) TSSBP

(b) ATP

Figure 2: The graphical model representations of the probability distribution functions for the (a) TSSBP, and (b) ATP (ours). The parameter θ_ϵ is associated with node ϵ in the tree, and T denotes the parameters $\{\pi_\epsilon\}_{\epsilon\in\mathcal{T}}$ from the TSSBP construction, where $\mathcal{T}$ is the set of all the nodes in the tree.

from the data in a Bayesian fashion. Since each mixture component is associated with a unique node in the infinite tree, this is equivalent to inferring the structure of the tree from the data. Let $\mathcal{T}$ denote the infinite set of node indices, and let π_ϵ denote the corresponding weight of the mixture component associated with node $\epsilon \in \mathcal{T}$, then one can sample a node z in the tree using

$$z \sim \sum_{\epsilon\in\mathcal{T}} \pi_\epsilon \delta_\epsilon(z), \tag{1}$$

where $\delta_\epsilon()$ denotes a Dirac delta function at ϵ.

Since the cardinality of the set $\mathcal{T}$ is unbounded, sampling from (1) is not trivial, however, an efficient sampling scheme was presented in (Adams et al., 2010). Similarly, an efficient scheme for sampling from the posterior of $\{\pi_\epsilon\}_{\epsilon\in\mathcal{T}}$ given the node assignments of all the data instances, was also developed in (Adams et al., 2010).

2.2.1 Factored Bernoulli likelihood model

The FBLM was used in (Adams et al., 2010) to perform hierarchical clustering of color images using binary feature vectors. Let $\mathrm{x}_i \in \{0,1\}^D$, $i = 1,\ldots,N$, denote a set of binary training vectors that are available for learning the hierarchy. The graphical model representation of the probability distribution function is shown in Figure 2a, where for the sake of clarity of the exposition, the tree that is shown here is finite. The parameters $\theta_\epsilon = [\theta_\epsilon^{(1)},\ldots,\theta_\epsilon^{(D)}]^T$ satisfy

$$\theta_\epsilon^{(d)} = \theta_{\mathrm{Pa}(\epsilon)}^{(d)} + n_\epsilon^{(d)},\ d = 1,\ldots,D, \tag{2}$$

where $n_\epsilon^{(d)} \sim \mathcal{N}(0,\sigma^2)$, and $\mathrm{Pa}(\epsilon)$ denotes the parent of node ϵ. The indicator variable z is sampled using (1), and the likelihood of a binary observation vector $\mathrm{x} = [x^{(1)},\ldots,x^{(D)}]^T$ follows a Bernoulli distribution whose parameter is a logistic function

$$f(\mathrm{x}|\theta_z) = \prod_{d=1}^{D}(1+\exp\{-\theta_z^{(d)}\})^{-x^{(d)}} \times (1+\exp\{\theta_z^{(d)}\})^{-(1-x^{(d)})}. \tag{3}$$

3 The attribute tree process

In this section, we develop a new generative model that is based on the TSSBP, however unlike the FBLM it also reveals a hierarchy of attributes, which allows for a linguistic description of the image dataset. This is achieved by relating the attribute hierarchy, and the assignment of image instances to nodes, in a probabilistic manner.

3.1 The attribute hierarchy

In order to allow for a probabilistic description of the attribute hierarchy, we associate a parameter vector $\theta = [\theta_\epsilon^{(1)},\ldots,\theta_\epsilon^{(D)}]^T$ with each node $\epsilon \in \mathcal{T}$, where D denotes the number of attributes. The attributes $y_i^{(d)}$, $d = 1,\ldots,D$ that are associated with a data instance i that is assigned to node ϵ, are generated using the following scheme:

1. For each $\epsilon' \in \mathrm{A}(\epsilon)$, draw $\xi_{\epsilon',i}^{(d)} \sim \mathrm{Bernoulli}(\theta_{\epsilon'}^{(d)})$, $d = 1,\ldots,D$,
2. Set $y_i^{(d)} = \bigoplus_{\epsilon'\in\mathrm{A}(\epsilon)} \xi_{\epsilon',i}^{(d)}$, $d = 1,\ldots,D$,

where $\xi_{\epsilon',i}^{(d)}$ is an auxiliary random variable, $\bigoplus$ denotes the logical or operation, and $\mathrm{A}(\epsilon)$ denotes the set composed of all the ancestors of node ϵ. By marginalizing with respect to $\xi_{\epsilon',i}^{(d)}$, we obtain a simplified representation: first set

$h^{(d)}_{\epsilon} = 1 - \prod_{\epsilon' \in \mathrm{A}(\epsilon)}(1 - \theta^{(d)}_{\epsilon'})$, and then sample $y^{(d)}_i \sim \mathrm{Bernoulli}(h^{(d)}_{\epsilon})$ for every $d = 1, \ldots, D$.

We use the parameters $h^{(d)}_{\epsilon}$ to define the attribute hierarchy, since they represent the probability of an attribute being associated with an instance assigned to node ϵ. Furthermore, they satisfy the property that the likelihood of any attribute can only increase when moving deeper into the tree. We can obtain an attribute hierarchy, similar to that in Figure 1, by thresholding $h^{(d)}_{\epsilon}$, and only displaying attributes that have not been detected at any ancestor node.

In order to complete our probabilistic formulation for the attribute hierarchy, we need to specify the prior for the node parameters θ_{ϵ}. We use a finite approximation to a hierarchical Beta process (Thibaux & Jordan, 2007; Paisley & Carin, 2009). This choice promotes sparsity, and therefore only few attributes will be associated with each node. Specifically, the parameters at the root node follow

$$\theta^{(d)}_0 \sim \mathrm{Beta}(a/D, b(D-1)/D),\ d = 1, \ldots, D, \tag{4}$$

and the parameters in the other nodes follow

$$\theta^{(d)}_{\epsilon} \sim \mathrm{Beta}(c^{(d)}\theta^{(d)}_{\mathrm{Pa}(\epsilon)}, c^{(d)}(1-\theta^{(d)}_{\mathrm{Pa}(\epsilon)})),\ d = 1, \ldots, D, \tag{5}$$

where Pa(ϵ) denotes the parent of node ϵ, and where a, b, and $c^{(d)}$, $d = 1, \ldots, D$ are positive scalar parameters.

In this work we used a uniform prior for the precision hyper-parameter $c^{(d)} \sim U[l, u]$ with $\ell = 20$, and $u = 100$. We also used the hyper-parameter values $a = 10$, and $b = 5$ (unless otherwise stated). In Section 5.1.1 we demonstrate that the performance of our algorithm depends only weakly on the choice of these parameters.

3.2 Assigning images to nodes

In order to assign every image instance to one of the nodes, we combine the attribute hierarchy with the TSSBP. The resulting graphical model representation of the probability distribution function is shown in Figure 2b. For every data instance i, a node z_i in the tree is sampled from the TSSBP. The observed attribute vector x_i is obtained by sampling $y^{(d)}_i \sim \mathrm{Bernoulli}(h^{(d)}_{z_i})$, and flipping $y^{(d)}_i$ with probability ω, which models the effect of the noisy attribute detectors. By marginalizing over $y^{(d)}_i$, we have that

$$p(x^{(d)}_i = 1|-) = 1 - ((1 - h^{(d)}_{z_i})(1 - \omega^{(d)}). + h^{(d)}_{z_i}\omega^{(d)}). \tag{6}$$

The prior for ω is $\omega \sim \mathrm{Beta}(\rho_0, \rho_1)$, where in this work we used $\rho_0 = 5$, and $\rho_1 = 20$, which promotes smaller values for ω. Our algorithm is highly insensitive to the choice of ρ_0, ρ_1, as long as they are chosen to promote small values of ω.

3.3 Inference

Inference in the ATP is based on Gibbs sampling scheme. In order to sample from the posterior of the node parameter $\theta^{(d)}_{\epsilon}$, we note that

$$\begin{aligned} &p(\theta^{(d)}_{\epsilon}|-) \propto \\ &\prod_{\epsilon' \in \epsilon \cup \mathrm{D}(\epsilon)} (1 - ((1 - h^{(d)}_{\epsilon'})(1 - \omega^{(d)}) + h^{(d)}_{\epsilon'}\omega^{(d)}))^{n^{(1,d)}_{\epsilon'}} \\ &\times ((1 - h^{(d)}_{\epsilon'})(1 - \omega^{(d)}) + h^{(d)}_{\epsilon'}\omega^{(d)})^{n^{(0,d)}_{\epsilon'}} \\ &\times \prod_{\epsilon'' \in \mathrm{Ch}(\epsilon)} \mathrm{Beta}(\theta^{(d)}_{\epsilon''}; c^{(d)}\theta^{(d)}_{\epsilon}, c^{(d)}(1 - \theta^{(d)}_{\epsilon})) \\ &\times \mathrm{Beta}(\theta^{(d)}_{\epsilon}; a^{(d)}_{\epsilon}, b^{(d)}_{\epsilon}), \end{aligned} \tag{7}$$

where $n^{(j,d)}_{\epsilon} = \sum_{i|z_i=\epsilon} \delta_j(x^{(d)}_i)$ for $j = 0, 1$, $\mathrm{D}(\epsilon)$ denotes the set composed of all the descendants of node ϵ, $\mathrm{Ch}(\epsilon)$ denotes the child nodes of node ϵ, and $a^{(d)}_{\epsilon} = a/D$, $b^{(d)}_{\epsilon} = b(D-1)/D$, for $\epsilon = 0$ (the root node), and for any other node: $a^{(d)}_{\epsilon} = c^{(d)}\theta^{(d)}_{\epsilon}$, $b^{(d)}_{\epsilon} = c^{(d)}(1 - \theta^{(d)}_{\epsilon})$. The expression in (7) is a highly complicated function of $\theta^{(d)}_{\epsilon}$, and therefore we use slice-sampling (Neal, 2000) in order to sample from the posterior. The slice-sampler is very much a "black-box" algorithm, which only requires the log likelihood of (7) and very few parameters, and returns a sample from the posterior. We sample the node parameters using a two-pass approach, starting from the leaf nodes and moving up to the root, and subsequently moving down the tree from the root to the leaves.

In order to sample from ω, we first sample the binary random variables $y^{(d)}_i$ using

$$p(y^{(d)}_i = j|-) \propto p(y^{(d)}_i = j|-)(\delta_j(x^{(d)}_i)(1 - \omega^{(d)}) + \delta_{1-j}(x^{(d)}_i)\omega^{(d)}),\ j = 0, 1, \tag{8}$$

and then sample ω using

$$\omega^{(d)}|- \sim \text{Beta}\big(\rho_0 + \sum_{i=1}^{N} \delta_1(y_i^{(d)} \text{xor } x_i^{(d)}),\\ \rho_1 + \sum_{i=1}^{N} \delta_0(y_i^{(d)} \text{xor } x_i^{(d)})\big). \tag{9}$$

Sampling from the posterior of the hyper-parameter $c^{(d)}$ was also performed using slice sampling. We note that slice sampling each of the parameters $\theta_\epsilon^{(d)}$ for $d = 1, \dots, D$, and each of $c^{(d)}$ for $d = 1, \dots, D$, can be implemented in a parallel fashion. Therefore, the computational bottleneck in the ATP is the number of nodes in the tree, rather than the number of attributes. Sampling from the posterior of the TSSBP parameters is performed using the algorithms developed in (Adams et al., 2010). The parameters of the stick-breaking processes involved in the TSSBP construction are also learned from the data using slice-sampling, by assuming a uniform prior on some interval (as was also performed in (Adams et al., 2010)).

4 Evaluating the attribute hierarchy

In order to quantify the performance of the attribute hierarchies, we evaluate the performance of the ATP as a hierarchical clustering algorithm. We consider two cases, in the first, the ground truth category hierarchy is available and can be used to compare different hierarchies quantitatively. In the second case, the ground truth is unavailable.

4.1 Using the ground truth category hierarchy

The category hierarchy should capture the distance between the categories in a semantic space. For instance, since car and bus are both vehicles, they should be assigned to nodes which are closer, compared to the categories car and sheep, which are semantically less similar. Given the ground truth category hierarchy, we can "measure" the semantic distance between different categories by counting the number of edges that separate any two categories in the graph.

In order to compare the hierarchies learned using different hierarchical clustering algorithms, we propose a criterion which measures the degree to which the semantic distance which a hierarchy assigns to different image instances, diverges from the semantic distance that is given by the ground truth category hierarchy. Let $d_{GT}(c_1, c_2)$ denote the number of edges separating categories c_1 and c_2 in the ground truth category hierarchy, and let $d_H(i,j)$ denote the number of edges separating instances i and j in a hierarchy that is learned using a hierarchical clustering algorithm. Our proposed criterion, which we refer to as the average edge error (AEE), takes the form

$$\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |d_H(i,j) - d_{GT}(c(i), c(j))|, \tag{10}$$

where $c(i)$ denotes the category of instance i, and N denotes the number of image instances.

4.2 Without ground truth hierarchy

When the ground truth of the category is unavailable, we propose to use the following two criteria in order to evaluate the hierarchies. The first is known as the purity criterion (Manning et al., 2009 p. 357), and the second is the locality criterion which we propose. The purity criterion measures the degree to which each node is occupied by instances from a single class, and takes the form

$$\text{Purity} = \frac{1}{N} \sum_{\epsilon \in \mathcal{T}} \sum_{i=1}^{N_\epsilon} \delta_{c_\epsilon^*}(c(i)), \tag{11}$$

where N_ϵ denotes the number of instances in node ϵ, and c_ϵ^* is the class which is most frequent in node ϵ.

The locality criterion measures the degree to which each class is concentrated in few adjacent nodes. Quantitatively we define the category locality for class c as

$$\text{CL}_c = -\frac{2}{(|C|-1)|C|} \sum_{\substack{i,j \in C, \\ i \neq j}} \text{dist}(\epsilon_i, \epsilon_j), \tag{12}$$

where $|\cdot|$ denotes the cardinality of a set, $C = \{i | c_i = c\}$ where c_i is the class associated with instance i, and $\text{dist}(\epsilon_i, \epsilon_j)$ denotes the number of edges along the path separating nodes ϵ_i and ϵ_j. The category locality is negative or equal to zero. Values that are closer to 0 indicate that the instances of category

c are concentrated in a few adjacent nodes, and negative values indicate that the category instances are more dispersed in the tree. We define the locality as the weighted average of the category locality, where the weights are the category instance frequencies.

We note that each of these objectives can generally be improved on the account of the other: locality can usually be improved by joining nodes (which in general makes purity worse), and purity can usually be improved by splitting nodes (which in general makes locality worse). Therefore, we argue that a desirable hierarchy should offer an acceptable compromise between these two performance measures.

5 Experimental results

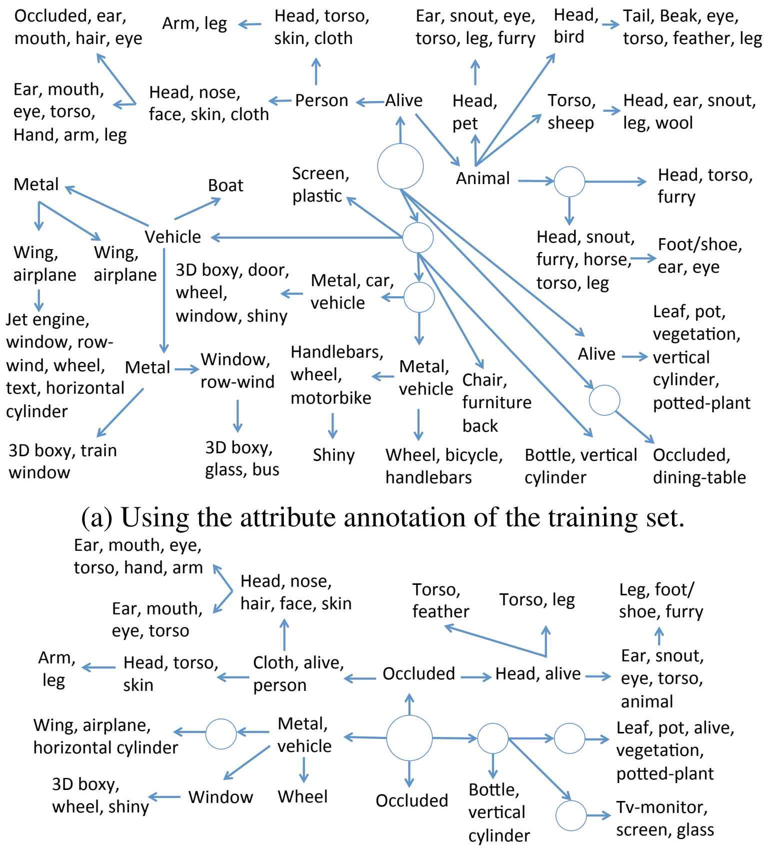
In this section we learn the attribute hierarchy using our proposed ATP algorithm. In order to evaluate the performance we evaluate the ATP as a hierarchical clustering algorithm, and compare it to the FBLM and AHC. We use subsets of the PASCAL VOC2008, and SUN09 datasets, for which attribute annotations are available. We learn hierarchies using the ground truth attribute annotation of the training set, and using the attribute scores obtained for the image instances in the testing set, where the attribute detectors are trained using the training set. We used the FBLM implementation which is available online. Our implementation of the ATP is based the TSSBP implementation which is available online, where we extended it to implement our ATP generative model. We used the AHC implementation available at (Mullner,), where we used the average distance metric, which is also known as the *Unweighted Pair Group Method with Arithmetic Mean* (UMPGA) (Murtagh, 1984).

5.1 Object category hierarchy

Here we consider the PASCAL VOC 2008 dataset. We use the bounding boxes and attribute annotation data that were collected in (Farhadi et al., 2009), and are available online, along with the low-level image features. Each of the training and testing sets contains over 6000 instances of the object classes: person, bird, cat, cow, dog, horse, sheep, airplane, bicycle, boat, bus, car, motorcycle, train, bottle, chair, dining-table, potted-plant, sofa, and tv/monitor. We used the annotation and features available for the training set, to train the attribute detectors using a linear SVM classifier (Fan et al., 2008). We used 88 attributes, which included 24 attributes in addition to those used in (Farhadi et al., 2009): "pet", "vehicle", "alive", "animal", and the remaining 20 attributes were identical to the object classes. The annotation for the first 4 additional attributes was inferred from the object classes. In all the experiments presented here, we ran the Markov chain for 10,000 iterations and used the tree and model parameters from the final iteration.

The attribute hierarchies for the PASCAL dataset are shown in Figure 3, when using the annotation for the training set, and when using the attribute scores obtained for the testing set. The hierarchies were obtained by thresholding $h_\epsilon^{(d)}$ with the threshold parameter 0.7 (this parameter is only used to create the visualization, and it is not used when learning the hierarchies), and only displaying the attributes that are not already associated with an ancestor node. It can be seen that the attribute hierarchies can accurately capture the semantic space that is represented by the 20 categories in the PASCAL dataset. An important observation is that attributes which are associated with more categories, such as alive or vehicle, are assigned to nodes that are closer to the root node, as compared to more specialized attributes such as eye or window.

In order to evaluate the performance of the attribute hierarchies quantitatively, we use the ground truth category hierarchy for the 20 categories in the PASCAL dataset, which is available at (Binder et al., 2012), and is shown in Figure 4. In Figure 5 we show the AEE performance measure, which we discussed in the previous section, for the different hierarchical clustering algorithms which we consider here. It can be seen that for the AHC, the AEE is very sensitive to the threshold parameter, which effectively determines the number of clusters. A poor choice of the parameter can adversely affect the performance significantly. On the other hand, the performance of the ATP and FBLM is significantly less sensitive to the choice of the hyper-parameters, since all the parameters are learned in a Bayesian fashion with weak dependence on the hyper-parameters. This is demonstrated for the ATP in Section 5.1.1. The ATP significantly improves the AEE as com-


Occluded, ear, mouth, hair, eye
Arm, leg
Head, torso, skin, cloth
Ear, snout, eye, torso, leg, furry
Head, bird
Tail, Beak, eye, torso, feather, leg
Ear, mouth, eye, torso, Hand, arm, leg
Head, nose, face, skin, cloth
Person
Alive
Head, pet
Torso, sheep
Head, ear, snout, leg, wool
Metal
Boat
Screen, plastic
Animal
Head, torso, furry
Vehicle
Wing, airplane
Wing, airplane
3D boxy, door, wheel, window, shiny
Metal, car, vehicle
Head, snout, furry, horse, torso, leg
Foot/shoe, ear, eye
Jet engine, window, row-wind, wheel, text, horizontal cylinder
Metal
Window, row-wind
Handlebars, wheel, motorbike
Metal, vehicle
Chair, furniture back
Alive
Leaf, pot, vegetation, vertical cylinder, potted-plant
3D boxy, train window
3D boxy, glass, bus
Shiny
Wheel, bicycle, handlebars
Bottle, vertical cylinder
Occluded, dining-table
Ear, mouth, eye, torso, hand, arm
Head, nose, hair, face, skin
Ear, mouth, eye, torso
Torso, feather
Torso, leg
Leg, foot/ shoe, furry
Arm, leg
Head, torso, skin
Cloth, alive, person
Occluded
Head, alive
Ear, snout, eye, torso, animal
Wing, airplane, horizontal cylinder
Metal, vehicle
Leaf, pot, alive, vegetation, potted-plant
3D boxy, wheel, shiny
Window
Wheel
Occluded
Bottle, vertical cylinder
Tv-monitor, screen, glass

(a) Using the attribute annotation of the training set.

(b) Using the attribute scores of the testing set.

Figure 3: Attribute hierarchy learned for the PASCAL dataset, using the (a) attribute annotation available for the training set, and (b) attribute scores obtained by applying the attribute detectors to the image instances of the testing set. The largest circle denotes the root node.

pared to the FBLM, both for the training and testing sets. We also note, that for the ATP, the AEE obtained for the training set is better than that obtained using the testing set's attribute scores (training: 1.76, testing: 2.82), which is consistent with our expectation. This is not the case for the FBLM (training: 6.55, testing: 5.63).

5.1.1 Sensitivity to hyper-parameters

In order to validate our claim that the ATP is highly insensitive to the choice of hyper-parameters, we performed experiments with different hyper-parameter values. In Table 1 we compare the performance when using different values for the hyper-parameters a, and b in (4). It can be seen that when comparing to AHC in Figure 5, the ATP is significantly less sensitive to the choice of hyper-parameters. When comparing to the FBLM, even for the the worst choice of a, b the AEE is still significantly better.

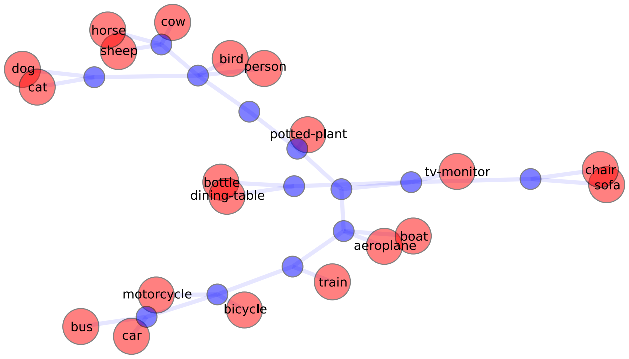
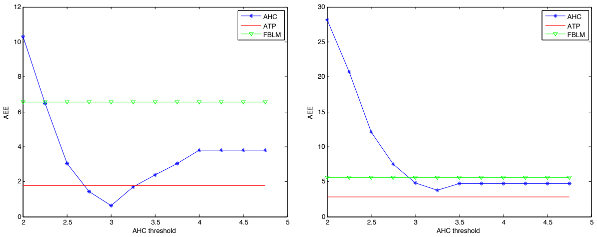


Figure 4: The ground truth category hierarchy, for the 20 categories in the PASCAL dataset.

5.2 Scene category hierarchy

Here we used the SUN09 dataset which is comprised of indoor and outdoor scenes. We use the training



(a) Training

(b) Testing

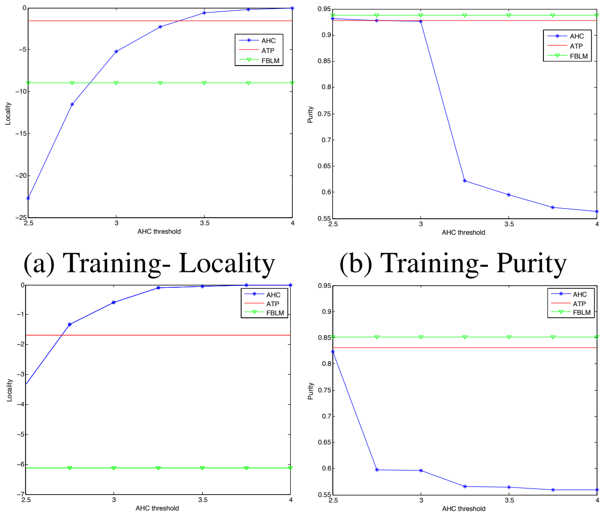
Figure 5: The average edge error (AEE) (10) vs. the AHC threshold parameter, for the hierarchies learned using the (a) training set's attribute annotations, and (b) attribute detectors applied to the testing set image instances. Smaller values indicate better performance. It can be seen that our ATP algorithm outperforms the FBLM, and unlike the AHC, it is not as sensitive to the choice of the hyper-parameters.

Table 1: Average edge error using the attribute annotation of the training set, for different hyper-parameters.

a	b	AEE
1	10	1.97
5	5	1.93
10	5	1.76
10	10	1.585
10	20	1.569

and testing sets which were used in (Myung et al., 2012), each containing over 4000 images. The annotation of 111 objects in the training set, and object detector scores for the testing set, are available online. Objects in the scene have the role of attributes in describing the scene. The object classifiers were trained using logistic regression classifiers based on Gist features that were extracted from the training set.

Since the ground truth category hierarchy is unavailable for this dataset, we use the locality and purity criteria, which we described in the previous section. We computed both of these measures with respect to the indoor and outdoor categories. Figure 6 shows the locality and purity measures for the training and testing sets. It can be seen that the AHC is very sensitive to the threshold parameter, and can produce unsatisfactory performance for a wide range of parameter values. The FBLM slightly outperforms the ATP with respect to the purity measure, however, its locality is very poor. Therefore, we conclude that the ATP provides an improved compromise with respect to the two criteria, which shows that the ATP captures the properties of a desirable hierarchy better than the FBLM.



(a) Training- Locality

(b) Training- Purity

(c) Testing- Locality

(d) Testing- Purity

Figure 6: The locality and purity measures vs. the AHC threshold parameter, using the training set's attribute annotation, and for the testing set's attribute scores. Larger values indicate better performance. It can be seen that our ATP has significantly better locality, and only slightly worse purity, compared to the FBLM. Furthermore, the performance of the AHC depends significantly on the choice of threshold parameter.

6 Conclusions

We developed an algorithm, which we refer to as the attribute tree process (ATP), that uses an attribute based representation to learn a hierarchy of linguistic descriptions, and can be used to describe a visual dataset verbally. In order to quantitatively evaluate the performance of our algorithm, we proposed appropriate performance metrics for the cases where the ground truth category hierarchy is known, and when it is unknown. We compared the ATP's performance as a hierarchical clustering algorithm to other competing methods, and demonstrated that our method can more accurately capture the ground truth semantic distance between the different categories. Furthermore, we demonstrated that our method has weak sensitivity to the choice of hyper-parameters.

Acknowledgments

We acknowledge the support of the NSF Grant CPS-0931474.

References

[Adams et al.2010] R. P. Adams, Z. Ghahramani, and M. I. Jordan. 2010. Tree-Structured Stick Breaking for Hierarchical Data. *NIPS*.

[Bart et al.2011] E. Bart, and M. Welling, and P. Perona. 2011. Unsupervised organization of Image Collections: Taxonomies and Beyond. *IEEE Tran. on PAMI,* 33(11):2302–2315.

[Berg et al.2011] T. L. Berg, A. C. Berg and J. Shih. 2010. Automatic Attribute Discovery and Characterization from Noisy Web Data. *CVPR*.

[Binder et al.2012] A. Binder, K. R. Muller, and M. Kawanabe. 2012. On Taxonomies for Multi-class Image Categorization. *International Journal of Computer Vision,* 99:281–301.

[Fan et al.2008] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research,* 9:1871–1874.

[Farhadi et al.2009] A. Farhadi, I. Endres, D. Hoiem, and David Forsyth. 2009. Describing objects by their attributes. *CVPR*.

[Ferrari & Zisserman2007] V. Ferrari and A. Zisserman. 2010. Learning Visual Attributes. *CVPR*.

[Jain & Dubes1988 p. 59] A. K. Jain and R. C. Dubes. 1988. *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, NJ.

[Lampert et al.2009] C. H. Lampert, H. Nickisch, and S. Harmeling. 2009. Learning to detect unseen object classes by between class attribute transfer. *CVPR*.

[Li et al.2010] L. J. Li, and C. Wang, and Y. Lim, and D. M. Blei, and L. Fei-Fei. 2010. Building and using a semantivisual image hierarchy. *CVPR*.

[Manning et al.2009 p. 357] C. D. Manning, P. Raghavan, and H. Schtze. 2009. *An Introduction to information retrieval*. Available online at http://nlp.stanford.edu/IR-book/.

[Mullner] D. Mulner. fastcluster: Fast hierarchical clustering routines for R and Python. *http://math.stanford.edu/ muellner/index.html.*

[Murtagh1984] F. Murtagh. 1984. Complexities of Hierarchic Clustering Algorithms: the state of the art. *Computational Statistics Quarterly,* 1: 101–113.

[Myung et al.2012] M. J. Choi, A. Torralba and A. S. Willsky. 2012. A Tree-Based Context Model for Object Recognition. *IEEE Tran. on PAMI,* 34(2):240–252.

[Neal2000] R. Neal. 2000. Nonparametric factor analysis with beta process priors. *Annals of Statistics,* 31:705–767.

[Paisley & Carin2009] J. W. Paisley, and L. Carin. 2009. Nonparametric factor analysis with beta process priors. *ICML*.

[Parikh & Grauman2011] P. Devi and G. Kristen. 2011. Interactively building a discriminative vocabulary of nameable attributes. *CVPR*.

[Sivic et al.2008] J. Sivic, and B. C. Russel, and A. Zisserman, and W. T. Freeman, and A. A. Efros. 2008. Unsupervised Discovery of visual object class hierarchies. *CVPR*.

[Teh et al.2006] Y. W. Teh and M. I. Jordan and M. J. Beal and D. M. Blei. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association,* 101(476):1566–1581.

[Thibaux & Jordan2007] R. Thibaux and M. I. Jordan. 2007. Hierarchical Beta Processes and the Indian Buffet Process. *AISTATS*.