Learning Visual Scene Attributes

Vazheh Moussavi
vmoussav@cs.brown.edu

1 A Glance at Attribute-Centric Scene Representations

Take a look around you. How would you describe your surroundings to best give an idea of what everything looks like to someone not there? Maybe you will give a category to the scene, say, 'bedroom'. You might try to list some of the objects around you, like 'bed', 'lamp', and 'desk'. Or perhaps you'll describe it with adjectives like 'indoors', 'cozy', and 'cluttered'. In computer vision, (or more specifically, in scene understanding), the most effective way to describe a visual scene is also a major question.

Of the these three ways of describing a scene, (commonly referred to as categorization, scene parsing, and attribute-based representation respectively), categories have historically been the method of choice. In categorization, an image (scene) is allowed to fall into exactly one of an arbitrary number of buckets. Attribute representations, however, are typically composed of several sets of buckets each of which will have a value associated with that scene. For instance, a simple category-based model would place an image in one of urban/rural/room, whereas a binary attribute-based model would have as attributes indoors and warm, each of which are marked as either present or not. In larger models, this leads to high dimensionality for attribute-based models, which has been a large disincentive for its use. In addition, classifying a scene's entire attribute set non-trivially falls under *multi-label learning*, for which there exist very few learning algorithms in popular use. Lastly, there is scene parsing[5], which involves using object detectors, possibly in conjunction, to build distributions over objects to define scenes.

Despite all this, attribute-centric models are becoming more popular. As seen in Figure 1, attribute-based representations give much more realistic partitions of the space of visual scenes than categories. We not only explicitly receive a fuller description of the scene, but there is the added benefit of a potential way to measure a distance between two scenes that does not require storing means of other images in the same category.

The authors of [2] further define visual attributes, introducing a dichotomy of *discriminative* and *semantic* attributes. Discriminative attributes do not have any pre-defined meaning. Instead, they are each built from trials where selected combinations of image classes (i.e. labels of categories and semantic attributes) are used to accordingly place instances from the dataset in two partitions. For a given partition, classifiers are trained on different lower-level (texture/color-based) features, and the resulting models that confidently and accurately discriminate among the selected classes are kept as the discriminative attributes.

Semantic attributes, on the other hand, have an interpretable meaning (several examples of classes are shown in Figure 1). Semantic attribute classes may themselves be categorized differently by a general visual property relevant to modeling. As an example, let us consider spatial persistence. An attribute such as "rusty" is local: it is clear that it applies only to certain parts of a scene (the rusty ones). Conversely, "open area" is a global attribute: its presence (according to our perception) can not be attributed to any specific part of a scene, and it applies entirely. Or, an attribute may be spatially ambiguous: either it can exist in both local/global forms or we might not consciously know ("driving" and "competing" being two examples). Clearly, modeling assumptions can affect the understanding of local and global attributes differently.

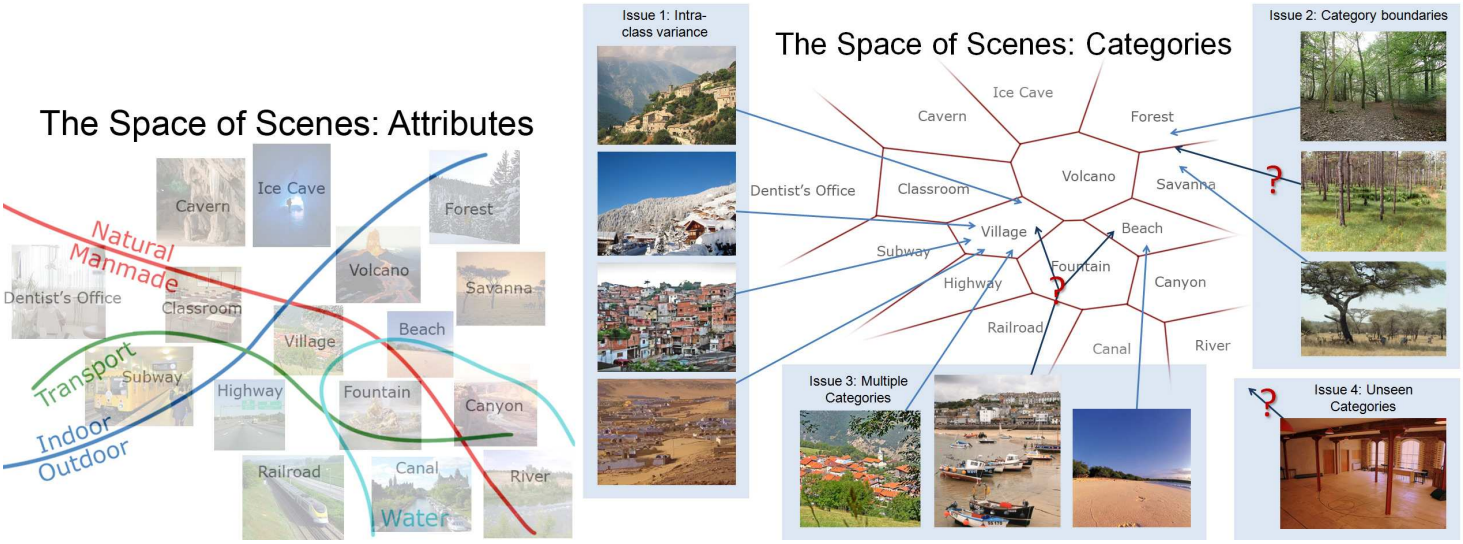

Figure 1: A comparison of scene understanding with attributes (left), and categories (right)[8].

For large-scale, data-driven computer vision systems, an obvious factor in determining success is the quality of the dataset used in training. We will focus primarily on two datasets made specifically for scene understanding. The SUN database[11] contains over 100,000 images, each annotated with one of 899 scene category labels, the first dataset to have such fine definitions of their scene categories. With this data, the authors achieve new benchmarks in scene classification. Of course, we are interested in scene attributes, and for that we look at the SUN Attributes dataset[8], a subset of the SUN database containing over 14,000 images. Each image is annotated with a binary label for each of 102 semantic scene attribute classes indicating that class's presence. The set of attribute classes forms four groups: functional affordances, materials, surface properties, and the spatial envelope (eg. "praying", "grass", "glossy", "no horizon", respectively). Labels were collected via crowd-sourcing.

The remainder of this paper is as follows: in section 2, work on weakly-supervised attribute detection for semantic attributes is reviewed, and in section 3, a small project on learning discriminative attributes in a bayesian nonparametric framework.

2 Weakly-Supervised Attribute Detection

2.1 Background and Relevant Work

Now that we have a dataset of scenes annotated with attribute class labels, it is appropriate to determine the capabilities of that dataset in scene/image understanding. We first consider examples from the two aforementioned dataset publications. In [8], the authors experiment with image class recognition, the task of predicting the class label of a test image, given training images annotated with a label for each class (in their case, the label corresponds to attribute presence). In addition to other potential applications, the labels learned from attribute class recognition can be used to boost the performance of a previously existing computer vision model, (eg. an object detector), by providing contextual information from which scene correlations can be established. In [11], the authors introduce *Scene Detection*, where scene categories are assigned to image *regions*, as opposed to just entire images (the authors take an approach based on sliding windows, common in object detection). This prompts an interesting question: what about an image and what part(s) of it define a scene to a computer?

What we would like to explore combines ideas from both of these publications. Extending the previous question to attributes, we wish to learn what pixels in the image of a scene correspond to a certain attribute. For instance, we want to know what areas in the scene are "rusty", or suggest affordance of "sunbathing". An ideal model for achieving this would be able to learn not just basic

features (color, texture, etc.) of an attribute, but high-level features that presumably humans also perceive and use to distinguish attributes (continuity/locality, size, shape, location, relation to surroundings, etc.). If successful, we could then construct a dictionary of visual definitions for an entire set of attributes.

Such a dictionary of semantic attributes has several possible applications. For instance, it could increase the range of output in artificial scene construction systems, by allowing users to apply attributes to areas of a scene, modifying them according to the learned attribute definitions. Also, there is the ability to query images by attribute, which would be a useful feature for image search engines. If the dictionary is particularly dense, one could imagine querying combinations of attributes to solve other vision tasks. As an example, a 'potted plant' detector could exhaustively search over every attribute combination known to correspond to potted plants, ('leaves', 'indoors', and 'soil' being one example), and give a result based on the overlap of the attribute detections.

We now consider how to build such a model, by first considering the approaches in the examples that inspired our goal. In attribute recognition, we model not the appearance of an attribute itself but the appearance of images in which the attribute is present. Nor do we model attribute location, continuity, or size. For global scene attributes, it may be fair to assume that ignoring these distinctions has a negligible effect in recognition; for local or space-ambiguous attributes, it is not. Following Scene Detection and using a window-based approach is also undesirable. There is no attempt to isolate the presence of an attribute, because it only considers crops of the original image. Thus, we only achieve approximations of attribute size, location, and continuity, and still do not model shape at all.

Having ruled out other options, it would seem that in order to accommodate modeling all of the desired attribute characteristics mentioned before, we need to take an approach based on segmenting the image. Most importantly, we would like for the segments to be relatively homogenous in their domains of description (a local attribute typically ceases to apply only where there is a pronounced change in texture/color). Also, it is by using regions without any predefined structure that we can hope to model the shape of an attribute. Lastly, we have the capability of modeling size, location, and continuity of an attribute by interacting neighboring segments.

2.2 Problem Description and Approach

The problem we wish to solve falls under a category of machine learning known as *Weakly-Supervised Learning*. It is supervised, in that our training data comes with class labels (scene attribute presence), but in a "weak" way, due to the fact that the labels lack part of the information we desire (per-pixel attribute detection). We will base much of our experimental work on a system proposed for object detection[6]. (For clarity, we will refer to our application of the model in terms of attributes instead.)

In the paper, the authors propose a very elegant to approach weakly-supervised learning in the context of computer vision. They take a segment-based approach, for many of the same reasons mentioned earlier. Each individual segment is described by a feature F. They classify the segment to an attribute based on the probability that the attribute is present in the containing image, which they name $\tilde{R}$. They define an image as O if the attribute in question is present, and $\overline{O}$ if not, which are the training labels that we are given. For a given segment with feature F_i, they compute its score as:

$$\tilde{R}(F_i) \triangleq P(O \mid F_i) = \frac{P(F_i \mid O)}{P(F_i \mid O) + P\left(F_i \mid \overline{O}\right)}$$

where $P(F_i \mid O)$ is the computed frequency of F_i over the training data. In other words, they use Bayes' Rule to obtain a posterior distribution among the set of segment feature values, setting the prior probabilities $P(O)$ and $P\left(\overline{O}\right)$ to be equal.

While this model is very clean and easy to understand, it does have some shortcomings. By considering each segment individually, it is unable to consider attributes that are not bound by the segments. Namely, for any global attribute (eg. electric light), the classifier would have to be convinced strongly enough by each individual segment suggesting its presence (such as the area nearby a lamp and the distinct soft shadows it creates). Ideally, a classifier would look around to piece together similar nearby segments and use the greater collective presence to boost its score. In addition to the

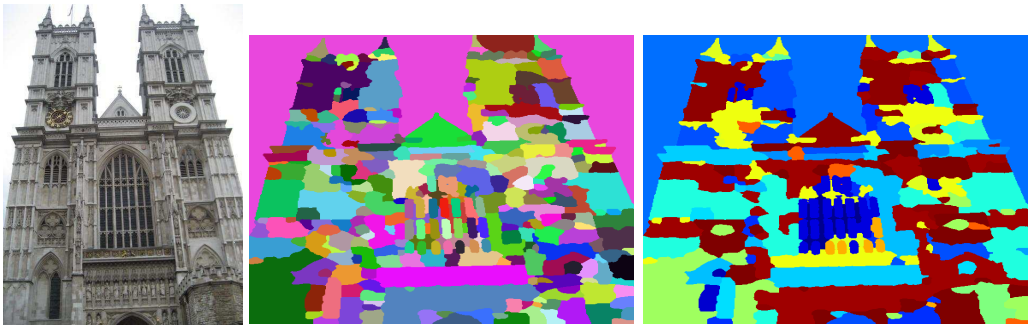

Table 1: A simple visualization of the pipeline. An image is segmented (center), and each segment is given a probability related to the presence of an attribute (right, rock/stone).

issues with spatial persistence and size, we are also unable to model shape and location with this setup.

We now detail the pipeline used in our experiments, closely following the one described in [6]. A visual description is provided in Table 1. The training data was the SUN Attributes dataset, which was broken into five equally-size 'splits' for the purpose of analyzing variance. We begin by segmenting the image as the authors with a mean-shift based segmentation algorithm. The Edison[1] system was chosen not just for its free availability, but use and approval by the same authors in prior work[7]. Parameters of the system were set to give larger and more robust segments. Nearest neighbor smoothing was applied for the same reasons.

Region description was texture-based; a filter bank of two scales and 16 orientations was used to compute responses. A texton vocabulary (100 words) is built from clustering a random sample from the responses of a subset of the images in the SUN Attributes dataset. For every pixel in a segment, its filter bank response is then matched to a word from the texton vocabulary, accumulating in a histogram. The k-means clustering and k-nearest neighbors matching software was taken from the VLFEAT[9] website.

In our experiments, we tried two different attribute classifiers. First, the posterior-based approach as previously discussed. This is again accomplished by clustering the texton histograms of segments into a feature vocabulary and matching to the words (vocabulary sizes of 50, 100, and 300 were considered). The conditional frequencies $P(F_i \mid O)$ of the training set are then computed, leading to the posterior probabilities necessary for classification. Another approach is to use a Support Vector Machine. The main benefit of using an SVM here is that because the texton histograms are used directly by the model, there is no loss of information due to quantization, which occurs in the posterior approach as a result of assigning the segment descriptors to feature words. On the other hand, when using an SVM we no longer receive probabilities, but a classification score that is only able to make relative distinctions without setting an arbitrary threshold. (A linear SVM was used in the experiments.)

2.3 Experiments

2.3.1 Results

Results from the posterior probability model were mixed, and general results are shown in Table 5 (probabilities are displayed on a relative 'jet' colormap from blue to red). As previously hypothesized, the classifier performed much better for local attributes such as materials than global and ambiguously-spatially persistent attributes. A comparison of the size of the feature vocabulary is shown in Table 3, and results are inconclusive, although there may be reason to believe that using more feature words could be beneficial (see the discussion section). Results among the different splits are shown in Table 2, and it can be seen that the data seems to give pretty consistent results among them.

An SVM classifier was additionally built, primarily for reasons of comparison discussed in the previous subsection. Table 6 shows visual results of the SVM classifier over multiple attributes and

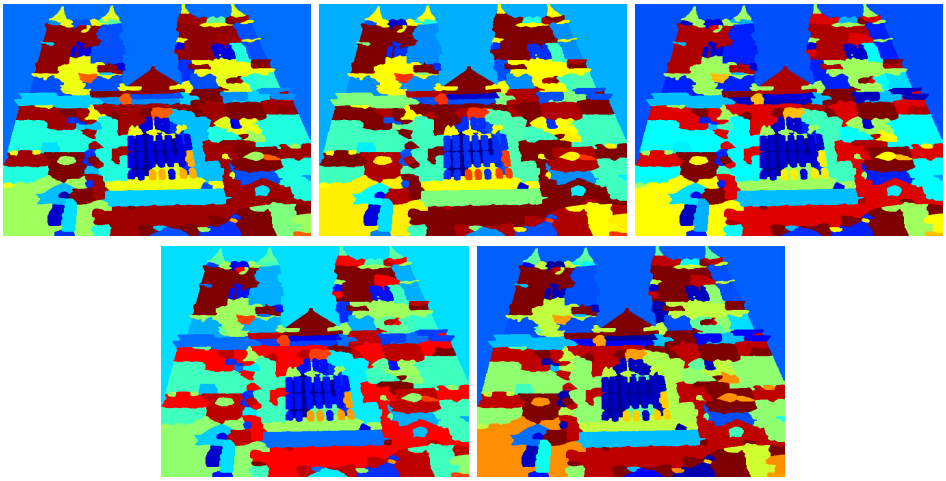

Table 2: A comparison of the results for the rock/stone attribute among each of the five splits. Results are fairly consistent, as the same strips in the abbey are mapped to similar probabilities.

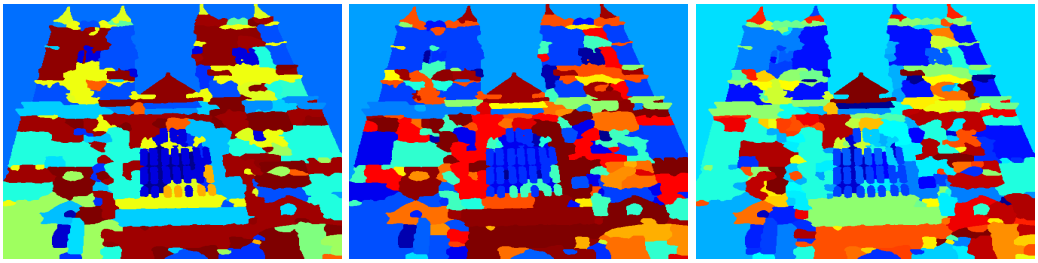

Table 3: A comparison of the posterior probability model using a vocabulary size of 50, 100, and 300 words, respectively.

images, and we can see that the results are perhaps slightly worse. Specifically, results tended to be oversmoothed. Analysis of the model output suggests again that the classifier is almost invariably negative. The weights and bias variable varied significantly among attributes, The SVM slack parameter was varied in experiments to see possible effects it would have on the classifier ($\lambda = 0.1, 1, 10$), and an example is shown in Table 4.

2.3.2 Discussion

Perhaps the biggest issue with the results is the lack of absolute discriminatory power. As stated before, the colormaps of the result figures are all relative, which (at least in some cases) show that the classifier does a good job of ordering segments in terms of their likelihood to indicate the

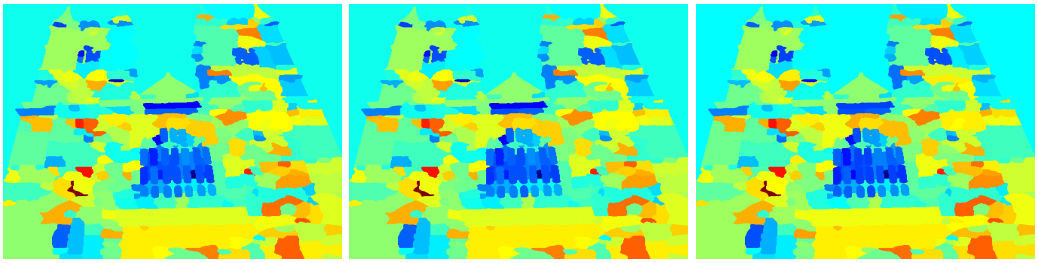

Table 4: A comparison of the svm model with different values for the slack parameter. Clearly, there is very little difference in this range.

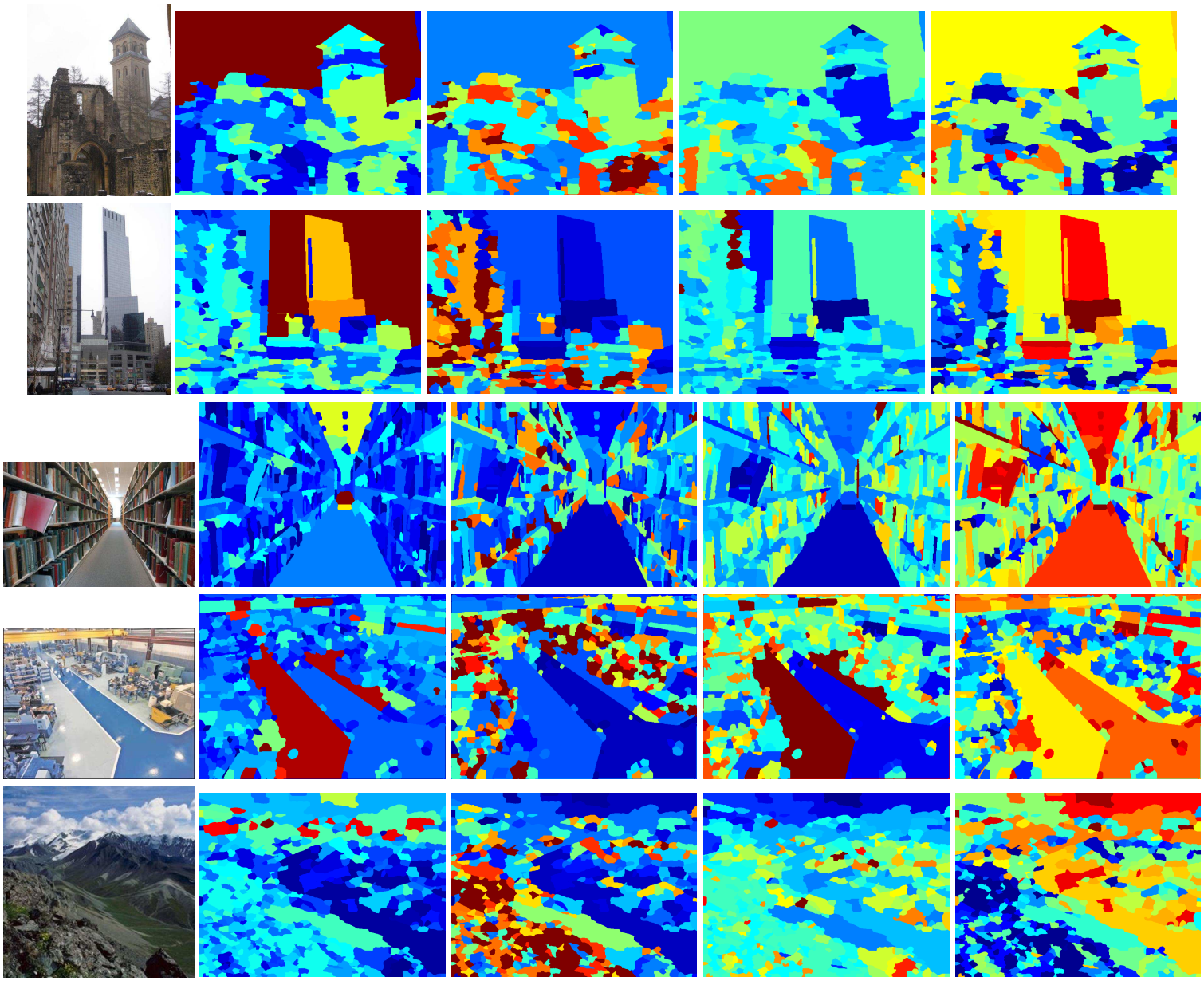

Table 5: Relative colormaps of the posterior model (100 words) for several images. Attributes from left to right: clouds, rugged scene, praying, glossy.

attribute in question. However, for the vast majority of results, the posterior probabilities are all in a very small range, meaning that the classifier is not learning the attributes with enough confidence. A hypothetical ROC curve of our classifier would have a sharp spike towards the top-left in one small segment, but stick close to the line $y = x$ everywhere else, leading to an unsatisfactory area under curve.

There is reason to believe that increasing the size of the feature vocabulary would be necessary in order to boost the classifier's discriminatory power. A likely explanation is that because of the large number of segments assigned to the same image attribute label (300-500 in most cases), there are not enough distinct descriptors to vary along with the number of images in a given split, effectively dividing the size of the training set by hundreds. While this is all well and good, due to the resultant increase in dimensionality, we would then need even more training images to accommodate the

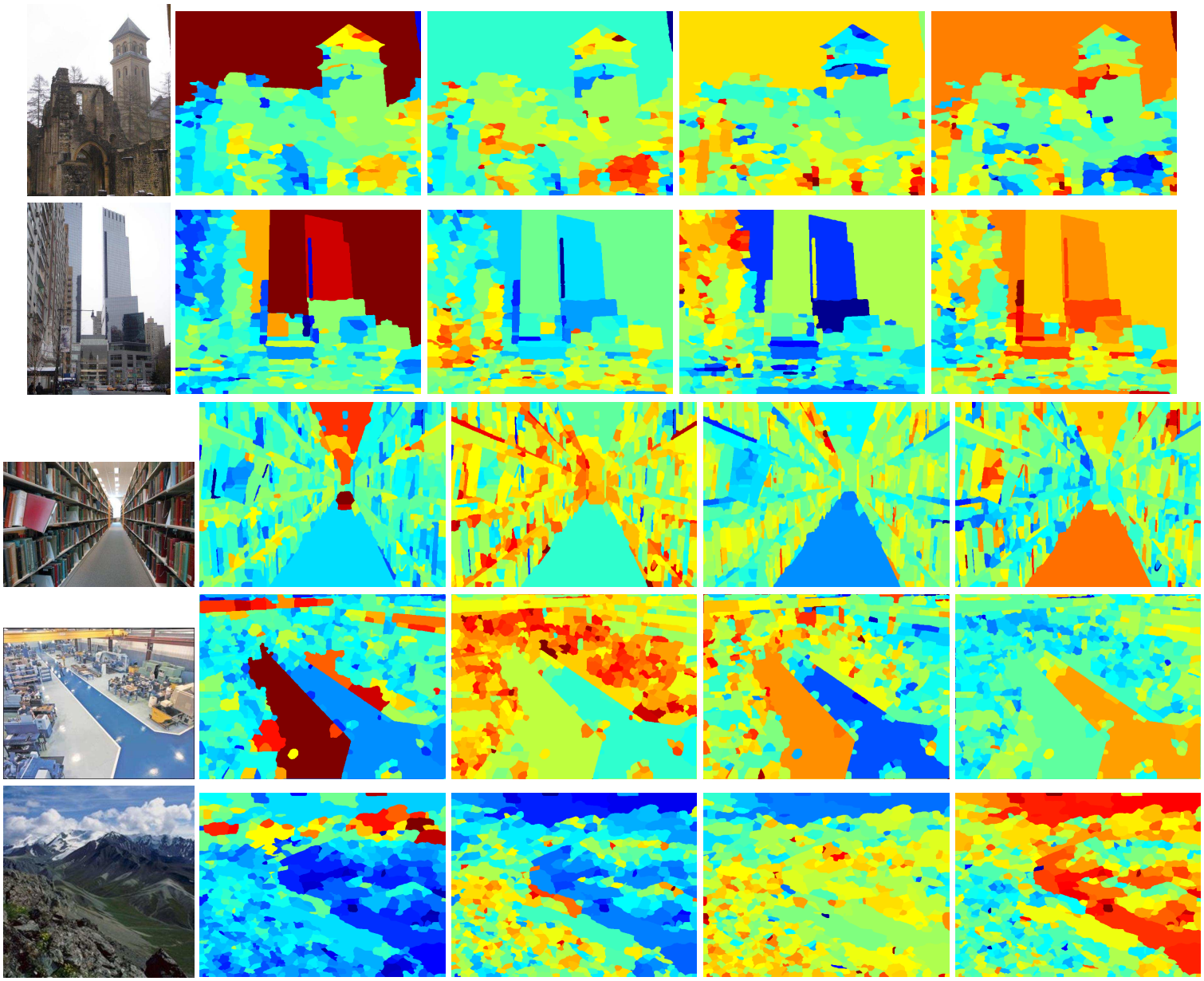

Table 6: Relative colormaps of the svm classifier for several images. Attributes from left to right: clouds, rugged scene, praying, glossy.

learning. The belief that more images are needed is further supported by evidence in Figure 2, which shows the average texton distributions among each of the training data splits for the 'sailing/boating' attribute (with similar results for other attributes). The histograms are nearly identical for each split, so it is clear that this is not an issue of variance from the textons. This would also seem to suggest that there is not enough of a distinction being made among region descriptors in assigning them to attributes, and thus there needs to be more data to further discriminate among them.

Additionally, the probabilities given by the classifier were overwhelmingly negative ($p < 0.5$). While this should be expected to a certain extent due to the fact that most attributes are more likely to be not present in an image, this result again suggests that there are not enough images to 'break through' the natural variance of data. Oddly enough, this was not an issue for the authors of [6], but

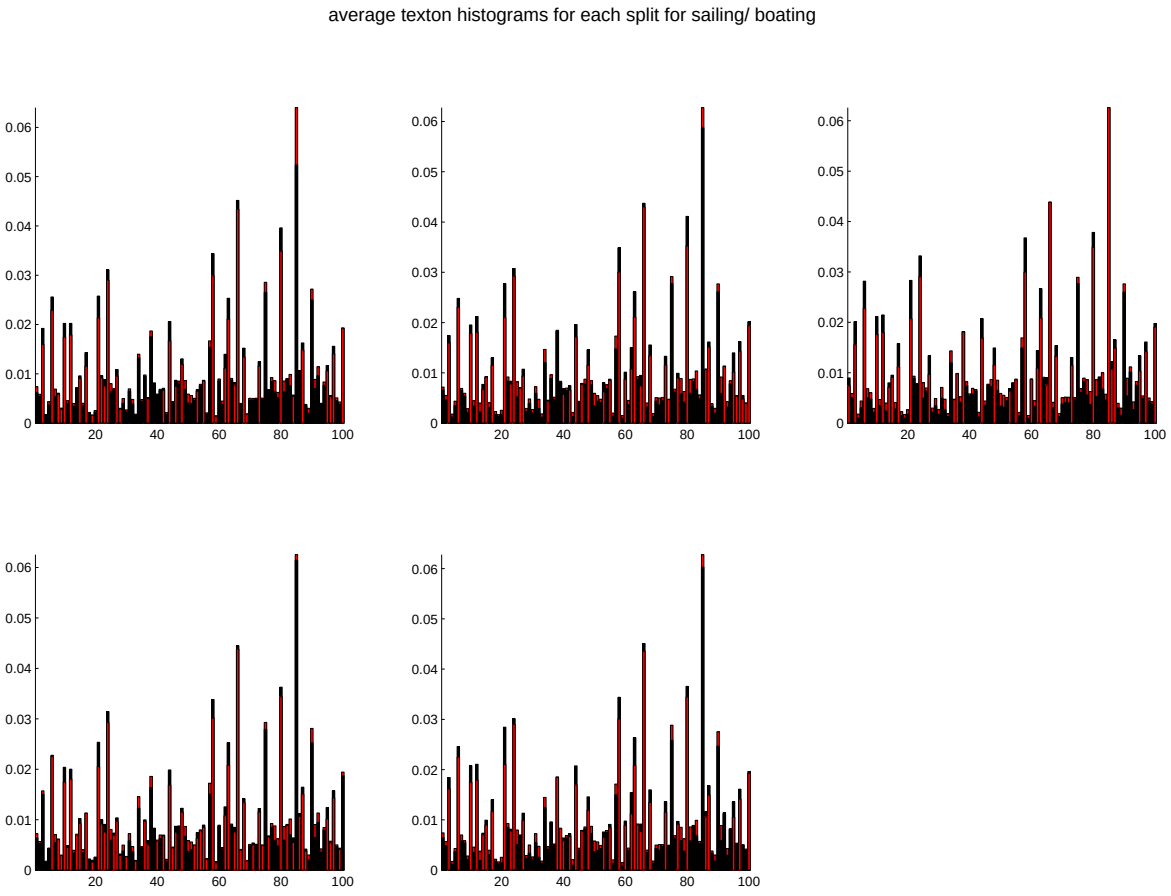

Figure 2: +/- texton histogram for each split for attribute sailing/boating

judging from a glance of their data, it appears that they were working with incredibly positive-biased training data, which may be an explanation.

At least one theoretical issue exists as well, that the posterior probabilities we compute are subtly different from what we ideally want. Recall that we model $P(O \mid F_i)$, the probability that the attribute is present in the *image*, given the feature F_i. Of course, we wish to model the probability that the attribute is in the segment. For small, local attributes it is difficult for the attribute-present segments to collectively influence the whole image enough, leading to problems with negative bias, as discussed before. For global and ambiguously-persistent attributes, the attribute-present segments are quite varied in their feature description, which leads to copious noise for the classifier.

2.4 Future Work

There are several additions we can make that would improve the quality of our results, as well as give a greater understanding of the effectiveness of the solution to this problem. Firstly, we could easily improve our segment description. There was no attempt to include color in our features, which means that our model is obviously nowhere near a complete representation of what we as humans use to distinguish what we see. Using color histograms alongside the textons would help the system to distinguish, say, a desert from an ocean (similar textures, very different colors). The features are also very simplistic, only using one texton bin to map to a probability. Modeling a joint distribution

over the features would allow for the classifier to learn and/or relationships among different feature types for a given attribute.

Next, our model does not consider using information from neighboring segments to determine likelihoods of attribute presence in that segment. Using energy-based graphical models for weakly-supervised tasks has been previously explored, and this could be an important step in trying to learn spatial persistence of attribute classes, as proposed earlier. Lastly, our experiments clearly lack a quantitative analysis. Ideally, a labeling of each segment would be available, but even this would be seemingly too large a task to be crowd-sourced.

3 Learning Discriminative Attributes with Bayesian Nonparametrics

3.1 Introduction

We now turn our attention from semantic attributes to discriminative ones, as previously discussed. An interesting aspect not yet covered in much depth is developing a system for learning such attributes. Because we know so much less about discriminative attributes than the semantic ones, we will need to use a much more flexible approach. For this, we turn to bayesian nonparametrics. Using a model known as the Indian Buffet Process[3], we consider a potentially infinite number of visual scene attributes to be learned in an unsupervised fashion. We first use the infinite factorial model of binary latent features with a linear-gaussian likelihood as described by [4]. We then move on to a fully generative *Noisy-OR* model for the IBP[10].

3.2 The Indian Buffet Process and Nonparametric Latent Feature Models

The *Indian Buffet Process*[3], defines a prior over binary matrices, which can in turn be used for models where objects are represented by multiple latent features. To sample from the Indian Buffet Process, one imagines a buffet with an infinite number of dishes, where customers enter one after another so that previous samplings for each dish are known to each one. The first customer samples the first Poisson(α) dishes and all following customers sample each dish in proportion to how many previous customers have already tried that dish, in addition to Poisson($\frac{\alpha}{i}$) new dishes, where α is a parameter which gives soft control over the number of dishes. Figure 3 shows an example run of the IBP. The resulting distribution over row-N binary matrices is:

$$P(Z) = \frac{\alpha^{K_+}}{\prod_{h=1}^{2^{N-1}} K_h!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N-m_k)!\,(m_k-1)!}{N!}$$

where K_+ is the number of columns (dishes) such that $m_k > 0$, K_h is a history-based mapping to distinguish columns in the *left-ordered-form*, and $H_N = \sum_{i=1}^{N} \frac{1}{i}$.

3.2.1 Infinite Factorial Linear-Gaussian Model

[4] presents a model for statistical inference that uses a zero-mean uncorrelated matrix Gaussian as a likelihood on X such that $E[X] = ZA$, where A is also matrix Gaussian. In this model, X is an $N \times D$ matrix of observations, Z is a $N \times K$ binary matrix indicating A is a $K \times D$ matrix representing the values of each hidden binary variable. In addition to the Linear-Gaussian model presented in [4], the authors also present a collapsed Gibbs Sampler for posterior inference. Starting with the conditional likelihood,

$$P(X \mid Z, A, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{ND/2}} \exp\left\{-\frac{1}{2\sigma_X^2}\text{tr}\left((X-ZA)^T(X-ZA)\right)\right\}$$

, a collapsed likelihood can be derived by integrating out the value matrix A,

$$\begin{aligned} P(X \mid Z, \sigma_X, \sigma_A) \;=\; & \frac{1}{(2\pi)^{ND/2}\,\sigma_X^{(N-K_+)D}\sigma_A^{K_+D}\left|Z_+^TZ_+ + \frac{\sigma_X^2}{\sigma_A^2}I_{K_+}\right|^{D/2}} \cdots \\ & \exp\left\{-\frac{1}{2\sigma_X^2}\text{tr}\left(X^T\left(I - Z_+\left(Z_+^TZ_+ + \frac{\sigma_X^2}{\sigma_A^2}I_{K_+}\right)^{-1}Z_+^T\right)X\right)\right\} \end{aligned}$$

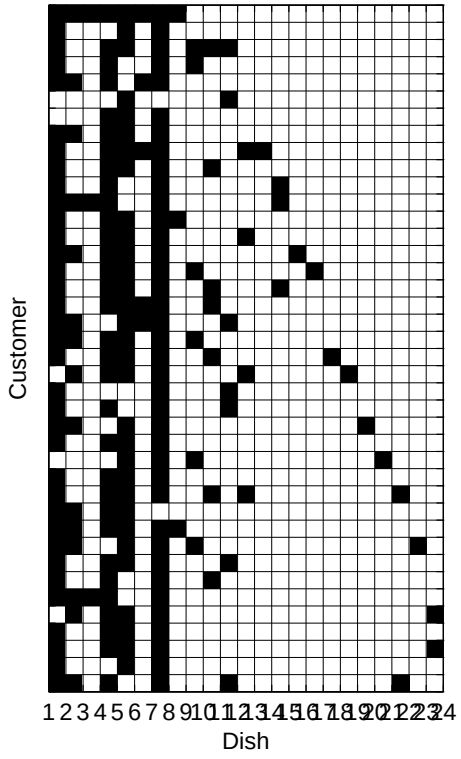

Figure 3: In the Indian Buffet Process, each customer (x_i) samples dishes (z_i) sequentially in proportion to how many previous customers have already tried each dish, with a fixed probability of sampling a new dish.

where Z_+ and K_+ respectively are Z and K with zero-sum columns removed. This, paired with an update for each element z_{ik} of Z is updated to be "on" with a probability equal to the proportion of other data points $-i$ with feature k on, gives us a way to sample from the posterior distribution for Z, given the hyperparameters α, σ_X and σ_A.

3.2.2 Noisy-OR IBP

In contrast to the discriminative model proposed above, [10] propose a generative model for learning causal structure with the IBP. The model is defined as follows: X, an $N \times T$ matrix of T observations, where each *row* represents a distinct binary random variable. Y, a $K \times T$ binary matrix indicating presence of latent causes for each observation. Lastly, Z, another binary matrix ($N \times K$) to relate the hidden causes to observed variables.

An IBP prior is placed on Z, and hidden causes are assumed to follow an iid. Bernoulli distribution:

$$P(Y) = \prod_{k,t} p^{y_{k,t}} (1-p)^{1-y_{k,t}}$$

Then, a *Noisy-OR* likelihood is placed on observations:

$$P(x_{i,t} = 1 \mid Z, Y) = 1 - (1-\lambda)^{z_i \cdot y'_t} (1-\epsilon)$$

, where z_i is the i^{th} row of Z, and y'_t is the t^{th} column of Y. Applying Bayes Rule gives a straightforward posterior distribution over the latent variables Z and Y.

Clearly, the biggest difference to be considered in modeling data with the Noisy-OR IBP model is the binary nature of the observations. Under a words-based representation, this would mean changing

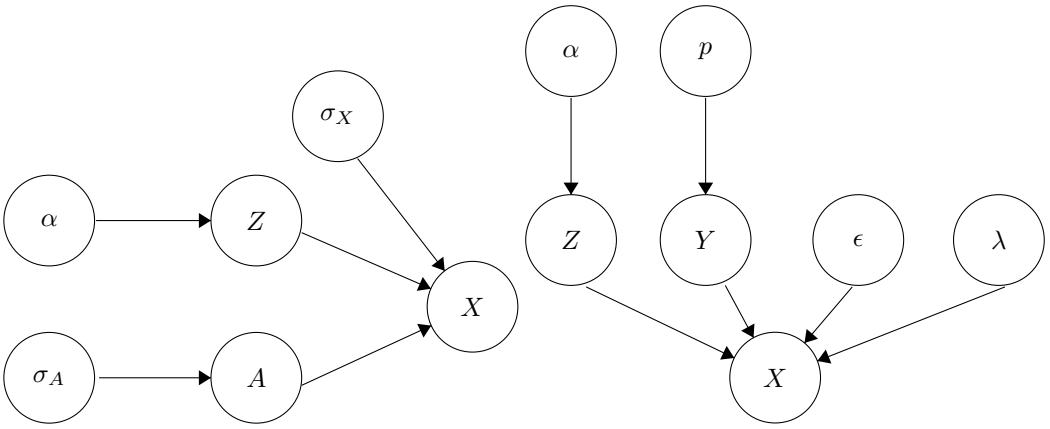

Figure 4: (left) Graphical model for linear-Gaussian model with binary features (σ_A and σ_X are the standard deviations for A and X, respectively, and α is the Poisson parameter for the basic IBP). (right) Graphical model for Noisy-OR IBP. ϵ is the the baseline probability that $x_{i,t} = 1$, λ is the prior probability of any cause affecting the observation, and p is the Bernoulli parameter for the iid distribution over all hidden causes $Y_{k,t}$.

data from a direct histogram of quantized words to instead denote simply the "presence" of a word in an image, the implementation of which is non-trivial when considering issues of bias from sampling size, among other factors.

[10] gives a Gibbs Sampler as well, although it is uncollapsed. The algorithm iterates through every latent variable ($1 \ldots N$), within which it iterates through every cause ($1 \ldots K$). First, each $z_{i,k}$ is sampled according to:

$$P\left(z_{i,k} = a \mid X, Z_{-i,k}Y\right) \propto \overline{\theta}_k^a \left(1 - \overline{\theta}_k^a\right)^{(1-a)} \prod_{t=1}^{T} \left(1 - (1-\lambda)^{z_i \cdot y_t'} (1-\epsilon)\right) |_{z_{i,k}=a}$$

, where θ_k is the proportion of other data points $-i$ with feature k "on". Then, the number of new latent features is sampled by

$$\begin{aligned} P\left(K_i^{new} \mid X_{i,1:T}, Z_{i,1:K+K_i^{new}}, Y\right) &\propto P\left(X_{i,1:T} \mid Z_{i,1:K+K_i^{new}}, Y, K_i^{new}\right) P\left(K_i^{new}\right) \\ P\left(X_{i,1:T} \mid Z_{i,1:K+K_i^{new}}, Y, K_i^{new}\right) &= \prod_{t=1}^{T} P\left(x_{i,t} \mid Z_{i,1:K+K_i^{new}}, Y, K_i^{new}\right) \\ P\left(x_{i,t} = 1 \mid Z^{new}, Y^{new}, K_i^{new}\right) &= 1 - (1-\epsilon)(1-\lambda)^{z_{i,1:K} \cdot y_{1:K,t}} (1-\lambda p)^{K_i^{new}} \end{aligned}$$

, where the prior probability of new K values is Poisson$\left(\frac{\alpha}{N}\right)$, as given by the IBP. Then each latent variable $y_{k,t}$ is sampled from:

$$P\left(y_{k,t} = a \mid Z, X, Y_{-k,t}\right) \propto p^a (1-p)^{1-a} \prod_{i=1}^{N} \left(1 - (1-\lambda)^{z_i \cdot y_t'} (1-\epsilon)\right) |_{y_{i,k}=a}$$

3.3 Experiments

3.3.1 Data

To test the empirical validity of our model, we will run experiments on the SUN Attributes dataset, with a set of 102 manually-labeled visual attributes. The set of attributes is by no means visually exclusive, and there are significantly correlated attributes (eg. foliage and leaves). It is also not a ground truth, but instead a reasonable representation of visual scenes by human aesthetics, and will be used as a means of assigning distances between images. In the following experiments, 20 images which had been categorized as "park" and 20 categorized as "indoor theater" in the SUN database[11]. These categories were selected because of their inherent difference in visual

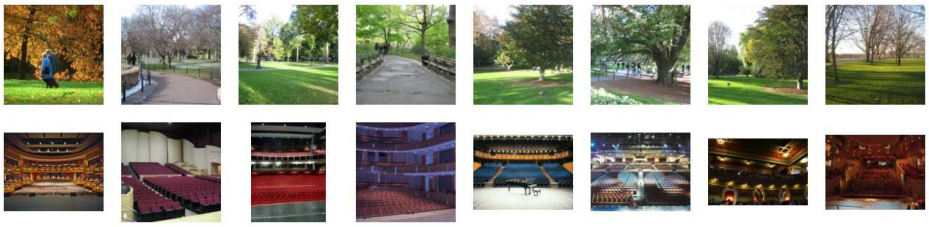

Figure 5: Example images used in experiments (top row: images of parks, bottom: images of theaters)

appearance, and we intend to show that our model should be able to discriminate between the two while finding similarities in attributes within the same category. See Figure 5 for a few examples.

3.3.2 Results

The Linear-Gaussian Model had incredible difficulty mixing for any vocabulary size larger than 20 (thousands of iterations were necessary). This is possibly due to the fact that integrating out the factor loading matrix A is the source of too much variability and does not allow for large enough steps to be made, limiting mixing from taking place. In addition, further tweaking of the M-H scheme may be necessary to avoid "killing" the σ parameters. The issues of these traits are supported in Table 7 which shows average Hamming distances for the images used for experimentation. When compared to the aforementioned SUN Attributes database, the Linear-Gaussian model does not do as good of a job in discriminating between the two image categories, with a somewhat blurred information for smaller vocabularies. For larger vocabularies, the learned model seems to find a larger difference on average among park images than it does when comparing them to images of theaters. Naturally, given the posterior mean of A, an interesting experiment with this model would be to attempt scene reconstruction. However, due to the great complexity of a natural scene, this is nearly impossible to do. In addition, a larger dataset, most likely to the order of millions, would be needed.

Also unlike the Linear-Gaussian model, the Noisy-OR IBP model seemed to perform well at discriminating between the two image categories. Figure 6 displays a colormap of similarity for each possible pair of images. By measuring the average over all iterations of the mean equality among hidden variables (Y) for each image, it is easy to see that this learned model strongly distinguishes the first 20 (park) images from the latter 20 (theatre). The reason that Y is used instead of Z is because of the changed meaning of data points under this model (observations are considered to be repeats of individual variables, as opposed to instances of sets of variables/features), so Y represents the latent space for the $T = 40$ images. It is also interesting to point out that, (roughly speaking), just as in the findings with the Linear-Gaussian model, theatre images were perceived as much closer among each other than park images.

Hamming Distance	SUN Attributes	Learned Attributes (Linear-Gaussian)				
		10 words	20	50	100	200
park	0.118	0.271	0.237	0.243	0.199	0.101
theater	0.069	0.377	0.236	0.149	0.104	0.049
park-theater	0.155	0.383	0.272	0.210	0.166	0.081

Table 7: Comparison of average Hamming distances using the SUN Attributes 102 attributes vs. those learned by the Linear-Gaussian model with varying vocabulary sizes of quantized visual words. Each row represents distance among park images, theater images, and cross-distances between the two categories.

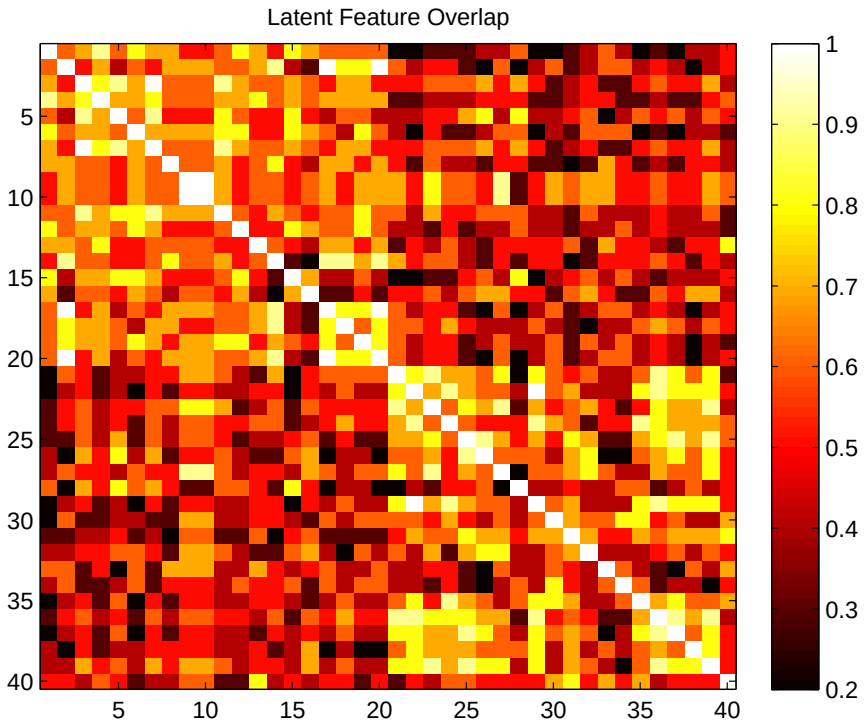

Figure 6: Average equality of hidden causes ($|Y_i = Y_j|$), over run of Gibbs Sampler for Noisy-OR IBP. The first 20 data points are park images, the last 20 are of indoor theaters.

3.3.3 Implementation Details

All code used to generate SIFT descriptors was provided by VLFEAT[9], which provides excellent interfaces and documentation for the MATLAB code used for this project. The visual words were quantized using k-means, with a euclidean distance metric, as neither VLFEAT nor built-ins provide a more appropriate histogram distance metric option (chi-squared, intersection, etc.).

For the Noisy-OR IBP model, the aforementioned "presence" of visual words was determined by placing a threshold equal to the inverse of the number of quantized words, and choosing words that were above this threshold as positive examples for observation data. This creates two flaws: first, that there is a somewhat arbitrary threshold, although this should not be an impactful issue, as natural variance among frequency of the words should cause most of the irrelevant words to be washed out anyways, and choosing a large enough vocabulary will ensure this. Second, that this introduces dependencies among the hidden variables, because of the histogram representation of the visual words. While this is certainly, true, one would hope that reduction to the simple binary case and a small subset of quantized words helps mitigate this.

All code used to obtain these stated results are adapted from Frank Wood's academic website[1]. The inference code, along with its excellent (conference-submission-left-over) display code was essentially unchanged itself. The collapsed sampler was implemented with few additional details necessary from that derived in the original paper, following a prescribed Metropolis-Hastings step for resampling α at each data point, or "customer". In addition, Metropolis-Hastings steps were also provided for the matrix Gaussian standard deviation parameters σ_A and σ_X.

The algorithmic decisions made for the Noisy-OR IBP model also was clear and made logical sense, but of course had more parameters to sample. p was initialized with a Beta(1,1), and afterwards resampled with a Beta($|Y_{k,t} = 1|$,$|Y_{k,t} = 0|$). α was initialized with a Gam(1,1) distribution, and re-

[1] `http://www.stat.columbia.edu/~fwood/Code/index.html`

sampled from Gam($1 + K_+, \frac{1}{1+H_N}$). Both ϵ and λ were initialized from a Uniform(0,1) distribution, and resampled using Metropolis-Hastings accept-reject procedures.

4 Acknowledgements

We would like to thank James Hays for his great research ideas and guidance as an advisor. We would also like to thank Genevieve Patterson for giving access to and experience with the SUN Attributes dataset.

References

[1] Christoudias, C., Georgescu, B., Meer, P., *Synergism in Low Level Vision*, International Conference on Pattern Recognition, 2002.

[2] Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.A., *Describing Objects by Their Attributes*, CVPR 2009.

[3] Ghahramani, Z., Griffiths, T., Sollich, P., *Bayesian nonparametric latent feature models*, Bayesian Statistics 8, 2007.

[4] Griffths, T. and Ghahramani, Z., *Infinite latent feature models and the Indian buffet process*, Gatsby Institute for Computational Neuroscience, University College, London, 2006.

[5] Liu, C., Yuen, J., Torralba, A., *Nonparametric Scene Parsing via Label Transfer*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[6] Pantofaru, C., Dorko, G., Schmid, C., Hebert, M., *A framework for learning to recognize and segment object classes using weakly supervised training data*, British Machine Vision Conference (BMVC), 2007.

[7] Pantofaru, C., Hebert, M., *A Comparison of Image Segmentation Algorithms*, 2005.

[8] Patterson, G., Hays, J., *SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes*, Proceeding of the 25th Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[9] Vedaldi, A., Fulkerson, B., *VLFeat: An Open and Portable Library of Computer Vision Algorithms*, `http://www.vlfeat.org/`, 2008.

[10] Wood, F., Griffiths, T., Ghahramani, Z., *A Non-Parametric Bayesian Method for Inferring Hidden Causes*, Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, 2006.

[11] Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A., *SUN database: Large-scale scene recognition from abbey to zoo*, Computer Vision and Pattern Recognition (CVPR), 2010.