Discovering objects and their location in images with Latent Dirichlet Allocation

Bryan C. Russell
Research Supervisor: William T. Freeman

Abstract

We seek to discover object categories and their locations in a set of unlabelled images. We achieve this using probabilistic models developed in the text understanding community to discover interesting topics in a corpus of text documents. We hope that the application of these models to a set of images will discover visual topics corresponding to object categories. We show how to form the visual analogue of text documents by vector quantizing SIFT-like regions at points found through two types of interest point detectors. We primarily focus on the Latent Dirichlet Allocation (LDA) model and compare with the very similar probabilistic Latent Semantic Analysis (pLSA) model. We describe the LDA model and the Gibbs sampling solution for Bayesian inference. We fit the LDA model to image sets for which we know ground-truth labels and investigate the visual topics that are discovered. In addition, for images with primarily a single object category depicted, we use the inferred LDA parameters to assign these images to the discovered visual topics and evaluate the goodness of these assignments through confusion matrices. Furthermore, we evaluate assignments to topics on unseen images and compare with an existing supervised method. Finally, we formulate a method for localizing objects in an image and evaluate the resulting segmentations. Since the LDA model assumes that words are exchangeable in a document, we show how to improve the segmentations by introducing a new vocabulary called ‘doublets’, which encode spatially local co-occurring regions.

1. Introduction

Humans effortlessly reason about complicated scenes. In particular, we can point to the different objects in a given image and relate them to objects that we have seen in other images. Teaching a computer how to “point” to the objects in an image and relate them to previously seen objects is not an easy task. A common approach to teaching a computer about objects is to provide some form of supervision. For face detection, often the location or boundary outline is provided [22, 31]. Recent work on multi-class object recognition involves specifying the object identity for cropped images containing a single object [9, 32] or providing keywords that name the objects contained in a given image [1, 2].

While such supervision is valuable, there are two main drawbacks. First, to effectively learn about all of the intra-class variation that an object class contains and to avoid overfitting, we need a large amount of labelled training data. This fact has been observed for the task of speech recognition and machine translation. However, obtaining large amounts of annotated data is expensive. Almost all existing annotated datasets contain only a few categories with around 100 annotations per category (see [19] for some examples). There has been recent progress with building datasets containing a larger number of categories with many images in each category [7, 9, 29]. Second, providing annotations may introduce unforeseen biases. Furthermore, when we sit down to provide labelled data, we encounter the question, What is an object anyway?

Keeping in mind that the number of available images is several orders of magnitude larger than those with annotations, the above observations lead us to ask the question, Is it possible to learn visual object classes simply from looking at images? In an attempt to answer this question, we will apply probabilistic models that have been used in the natural language processing community to discover interesting topics in a corpus of text documents. To apply these models, we will need a way to “convert” an image to a text document. We will develop the notion of a “visual word” by quantizing local appearance descriptors [5, 25] that are found at interesting points in an image and adjusted for affine invariance [14, 15, 21]. In this paper,

For tractability reasons, both of these models make the simplifying assumption that words in a given document are exchangeable, where the spatial relationship between words are ignored. This is also known as the bag-of-words assumption. While this assumption seems to throw away a lot of useful information, the two models cope with the loss of information in the text domain because certain words are highly discriminative and because of the redundancy of natural language.

For the visual domain, at first glance it seems that we cannot effectively exploit regularity as in the natural language domain since spatial relationships are just as crucial as local features themselves to detecting and recognizing objects. However, the local appearance descriptors that we use have been used for matching regions in two separate images [13] and have been shown to be quite discriminative and able to encode complex visual stimuli. Furthermore, the affine-invariant regions detected by the interest point operators overlap significantly, as shown in Figure 11(b) and hence provide much redundancy. The overlap also preserves spatial information since a reshuffling of the pixels will yield different interest points and descriptors.

In addition to discovering topics, we also show how to localize objects in a given image using the detected visual words. While these visual words provide surprising top-down segmentations, we attempt to improve the results by introducing an additional vocabulary, which we call doublets, that is formed from spatially neighboring word pairs. We show that doublets provide a cleaner segmentation of the objects in an image.

We wish to highlight a few recent papers that use models similar to the LDA model. As mentioned above, [1, 2] use text from captions and are able to extend the LDA model by including these observations. In [8], a variation of the author-topic model [20] is used where a scene label (e.g. street, office, etc.) is specified for each image and visual topics are discovered. Sudderth et al. extend the LDA model by modelling the joint spatial relationship between shared parts across the image corpus [27] and by incorporating a modified version of the hierarchical Dirichlet process [28] to count the number of objects in an image [26].

2. The LDA model

Suppose we are given an ordered set \(\mathcal{D}\) of size \(D\) whose elements \(\{w_1^d, \ldots, w_{N_d}^d\} \in \mathcal{D}\) are themselves ordered sets of size \(N_d\) containing discrete elements \(w_i^d \in V\), where \(V = \{v_1, \ldots, v_V\}\). Let us call the set \(\mathcal{D}\) a corpus, \(d \in \mathcal{D}\) a document in the corpus, \(w_i^d\) the \(i\)th word in the \(d\)th document, and \(V\) the set of all vocabulary words that can appear in the corpus. We would like to describe this data with a probablistic generative model \(p(D|\eta)\) for some parameters \(\eta\).

To simplify the model, we will make a few relaxations. First, we will assume that the order of the words in a particular document does not matter. In other words, if \(\pi\) is a permutation on the index set \(\mathcal{I} = \{1, \ldots, N_d\}\), then \(p(w_1^d, \ldots, w_{N_d}^d) = p(w_{\pi(1)}^d, \ldots, w_{\pi(N_d)}^d)\). When this condition is satisfied, the random variables \(w_1^d, \ldots, w_{N_d}^d\) are said to be ‘exchangeable’. An infinite sequence of random variables is said to be ‘infinitely exchangeable’ if the joint distribution of all finite subsequences are exchangeable. De Finetti showed that an infinitely exchangeable infinite sequence of random variables is independently and identically distributed from a distribution conditioned on a random parameter drawn from another distribution [6]. Application of this theorem to the finite case results in

\[
p(w_1^d, \ldots, w_{N_d}^d) = \int p(\theta^d) \prod_{i=1}^{N_d} p(w_i^d|\theta^d) d\theta^d
\]

for some latent random parameter \(\theta^d\).

We will assume that each word \(w_i^d\) is associated with one of \(T\) finitely discrete topics. Let the latent random variable \(z^d_i\) denote the assignment of the topic to the \(i\)th word in the \(d\)th document. We will assume that each word is drawn conditioned on its topic assignment. These assumptions yield

\[
p(D|\beta) = \int p(\phi|\beta) \sum_{z^d} \int p(\theta^d) \prod_{i=1}^{N_d} p(w_i^d|z_i^d, \phi) p(z_i^d|\theta^d) d\theta^d d\phi
\]

where \(\phi\) is a set of parameters, \(p(\phi|\beta)\) is a prior distribution over these parameters, and where we assume that \(w_i^d\) is independent of \(\theta^d\) given \(z_i^d\).

\[\text{We define the random variables } \bar{w}_i^d : V \rightarrow \{1, \ldots, V\} \text{ and let } p(D|\eta) \equiv p(\bar{w}_1^d, \ldots, \bar{w}_{N_D}^D, \ldots, \bar{w}_1^D, \ldots, \bar{w}_{N_D}^D). \text{ For notational convenience, we let } w_i^d \equiv \bar{w}_i^d.\]

\[\text{This is also known as the bag-of-words assumption.}\]
Similar to the word order in a particular document, we will relax the condition on the order of the documents in the corpus. Namely, we will assume that the sets \( \{w_i^1, \ldots, w_i^T\} \) of random variables are exchangeable. If we condition on the parameter from which each set of random variables are drawn, we get

\[
p(D|\alpha, \beta) = \prod_{d=1}^{D} \int p(\phi|\beta) \sum_{\phi} p(\theta^d|\alpha) \prod_{i=1}^{N_d} p(w_i^d|z_i^d, \phi)p(z_i^d|\theta^d)d\theta^d d\phi
\]

where \( \alpha \) is the parameter and where we assume that \( w_i^d \) and \( z_i^d \) are independent of \( \alpha \) given \( \theta^d \).

The graphical model for this family of distributions is shown in Figure 1. To generate a document from the LDA model, we use

\[
\sum_{s=1}^{V} \phi_{t,v} = 1 \quad \forall t \in \{1, \ldots, T\}
\]

and

\[
\phi_{t,v} \geq 0 \quad \forall t \in \{1, \ldots, T\} \quad \text{and} \quad \forall v \in \{1, \ldots, V\}.
\]

To find the estimate \( \phi^* \), we draw samples from the posterior distribution \( p(z|D, \alpha, \beta) \) to get

\[
\phi^* \in \arg\max_{\phi} \mathcal{L}(\phi) \approx \arg\max_{\phi} \sum_{s=1}^{S} p(\phi|z_s, D, \alpha, \beta) \approx \arg\max_{\phi} \sum_{s=1}^{S} p(\phi|z_s, D, \alpha, \beta) \approx \arg\max_{\phi} \mathcal{L}(\phi, z_s)
\]

where \( S \) is the number of samples drawn and \( z_s \sim p(z|D, \alpha, \beta) \). For the last approximation, we are assuming that the modes do not overlap much.\(^4\) To get an estimate for \( \phi^* \), we need to compute \( \phi^* \in \arg\max_{\phi} p(\phi|z_s, D, \alpha, \beta) \) for each sample \( z_s \) and then choose the \( \phi^* \) that results in the highest posterior given \( z_s \). For a particular sample \( z_s \), we form the Lagrangian of the optimization problem in Equation 6 and take partial derivatives to get

\[
\phi^*_{t,v} \propto \beta_{v} + f_{t,v}(D, z_s)
\]

\(^4\)In general, this is not true. In the future, we are planning to apply a constraint-based gradient method to solve for the cost function that results after combining the different samples.
Figure 2: (a) Unigram model. There is only one topic for the entire corpus and is multinomially distributed. (b) Mixture of unigrams model. Here, we assume that each document can be described by a single topic and words are multinomially distributed conditioned on the topic. Blei et al. [4] show that this is too limiting to describe a corpus of documents. (c) pLSA model of Hofmann [11]. In this model, each document may be described by multiple topics as in LDA. However, this model assumes that the document indices are observed in addition to the words. This tends to cause overfitting problems (see text for more details).

\[
f_{t,v}(D, z) = \sum_{w^d_i \in D} \delta[z^d_i = t \text{ AND } w^d_i = v]
\]

where \( \delta[\cdot] \) is the Kronecker delta. Intuitively, \( f_{t,v}(D, z) \) is counting the number of times vocabulary word \( v \) is assigned to topic \( t \) for a particular instantiation of \( z \).

To draw samples from \( p(z|D, \alpha, \beta) \), we use the Gibbs sampling approach where we fix all but one of the random variables \( z_i \) and sample the remaining random variable \( z_i \) conditioned on \( z_{-i} \). This conditional distribution is given by

\[
p(z_i = t|z_{-i}, D, \alpha, \beta) \propto (f_{d,t}(D_{-i}, z_{-i}) + \alpha) \left( \frac{f_{t,v}(D_{-i}, z_{-i}) + \beta_v}{f_t(D_{-i}, z_{-i}) + \sum_v \beta_v} \right)
\]

for \( z_i \) in document \( d \), \( w^d_i = v \), and \( w^d = \{w^d_1, \ldots, w^d_{N_d}\} \). Notice that Equation 9 is a function of the fixed variables \( z_{-i} \) and hence does not include \( z_i \) in the counting functions. Intuitively, Equation 9 is the frequency of vocabulary word \( v \) being assigned to topic \( t \) multiplied by the number of times topic \( t \) appears in document \( d \). The hyperparameters \( \alpha \) and \( \beta \) provide smoothing for unobserved words.

If instead we are interested in inferring the latent parameters \( \theta \), then we can follow almost identically the steps outlined above to obtain

\[
\theta_{d,t} \propto \alpha_t + f_{d,t}(D, z_s)
\]

where we draw the samples \( z_s \) as before.

2.2. Comparison with Other Models

Figure 2 shows graphical models for other attempts at modelling a corpus of documents. Of particular interest is the probabilistic Latent Semantic Analysis (pLSA) model of Hofmann [11] (see Figure 2(c)). This model allows documents to contain multiple topics as in LDA, but does not have a prior over the topic probabilities for each document. Furthermore, the model explicitly indexes the documents in the training set. This has the following undesirable effects: the model is not a true generative model and hence cannot model unseen documents; to infer the multinomial weights for \( P(w|z) \), one must also infer \( P(z|d) \), which leads to overfitting. pLSA tries to overcome these issues through various heuristics (such as folding-in unseen documents to infer \( P(z|d) \) and tempering to smooth the parameters). LDA overcomes these issues because of its ability to integrate over latent parameters. See [4] for an empirical explanation of these effects.
3. Obtaining Visual Words

In this section, we will outline how to obtain visual words so that an image can be converted into a document. We will describe two types of visual words: vector-quantized SIFT features [13] obtained through two different interest point detectors and pairs of words called “doublets” that loosely encode spatial relationships. With these visual words, we can then fit the LDA model to the data and infer the latent variables.

3.1. Vector-Quantized SIFT Features

We seek a vocabulary of visual words that will be insensitive to changes in viewpoint and illumination. To achieve this, we use vector quantized SIFT descriptors [13] computed on affine covariant regions [14, 15, 21]. Affine covariance gives tolerance to viewpoint changes. SIFT descriptors, based on histograms of local orientation, gives some tolerance to illumination change. Vector quantizing these descriptors gives tolerance to morphology within an object category. Others have used similar descriptors for object classification [5, 18], but in a supervised setting.

Two types of affine covariant regions are computed for each image. The first is constructed by elliptical shape adaptation about an interest point. The method is described in [15, 21]. The second is constructed using the maximally stable procedure of Matas et al. [14] where areas are selected from an intensity watershed image segmentation. For both of these we use the binaries provided at [30]. Both types of regions are represented by ellipses. These are computed at twice the originally detected region size in order for the image appearance to be more discriminating.

Each ellipse is mapped to a circle by appropriate scaling along its principal axes and a SIFT descriptor is computed. There is no rotation of the patch, i.e. the descriptors are rotation variant (alternatively, the SIFT descriptor could be computed relative to the the dominant gradient orientation within a patch, making the descriptor rotation invariant [13]; since this invariance gives slightly worse performance, we do not carry out this procedure in this work). The SIFT descriptors are then vector quantized into the visual ‘words’ for the vocabulary. The vector quantization is carried out here by \(k\)-means clustering computed from about 300K regions. The regions are those extracted from a random subset (about one third of each category) of images of airplanes, cars, faces, motorbikes and backgrounds (see Expt. (2) in section 4). About 1K clusters are used for each of the Shape Adapted and Maximally Stable regions, and the resulting total vocabulary has 2,237 words. The number of clusters, \(k\), is clearly an important parameter. The intention is to choose the size of \(k\) to determine words which give some intra-class generalization. This vocabulary is used for all the experiments throughout this paper.

3.2. Doublet Visual Words

For the task of segmentation, we seek to increase the spatial specificity of object description while at the same time allowing for configurational changes. We thus augment our vocabulary of words with “doublets” – pairs of visual words which co-occur within a local spatial neighborhood. As candidate doublet words, we consider only the 100 words (or less) with high probability in each topic after an initial run of LDA. To avoid trivial doublets (those with both words in the same location), we discard those pairs of ellipses with significant overlap. We then form doublets from all pairs of the remaining words that are within five nearest neighbors of each other. There is a preference for ellipses of similar sizes, and this is achieved by using a distance function (for computing the neighbors) that multiplies the actual distance (in pixels) between candidate ellipses by the ratio of the larger to smaller major axis of the two ellipses. Figure 3 illustrates the geometry and formation of the doublets. Figures 11(d),(e) show examples of doublets on a real image.
4. Experiments

Now that we are able to convert images into “documents” containing visual words, we can perform Bayesian inference to recover the latent parameters. This will allow us to explore four interesting tasks: (i) discovering visual topics along with the visual words that best describe the topics, (ii) clustering the images used to find the visual topics based on the image’s most likely topic, (iii) classification of unseen images based on the topics found by another set of images, and (iv) topic detection and localization.

We will see that the visual topics will often correspond to object categories or types of texture. To ensure that we understand the topics that are discovered, we consider image sets for which we know the desired visual topics. We consider the Caltech 101 dataset [7, 9] and the MIT Objects and Scenes dataset [29]. The Caltech dataset contains 101 different object categories with one category prominently featured in a given image. This dataset contains large intra-category variation. The MIT dataset is more difficult as it contains multiple object categories in a given image. Example images from both of these datasets and a description of the preprocessing on these images are shown in Figure 4. We will show results on these two datasets and compare with the output from pLSA and a baseline k-means method.

Baseline Method – K-means (KM) To understand the contributions of the topic discovery model to the system performance, we also implemented an algorithm that finds clusters based on the word frequency vector for each image (i.e. we compute a vector $f$ of length $V$ that is the normalized histogram count of the visual words for a given image). This is very similar to the mixture of unigrams model except that we assume that $p(f|z_d)$ is Gaussian distributed where $z_d$ is the topic assignment for the document. The standard $k$-means procedure is used to determine $T$ clusters from the word frequency vectors by hard-assigning each document to exactly one cluster based on the Euclidean distance of the feature vector to the cluster center.

Model Learning For LDA, the hyperparamters essentially perform Laplacian smoothing by controlling the mixing of the multinomial weights (lower values give less mixing). This prevents degeneracy for the case where there are no observed words for a given topic. As in [10], we specialize to scalar hyperparameters (i.e. $\alpha_i = a \forall i$). For all of our experiments, we used $\alpha_i = 0.5$ and $\beta_j = 0.5$. We run the Gibbs sampler 10 times with 100 iterations (every $z_i$ is sampled in each iteration) for each run. For each run, we randomly assign the initial topic settings. We compute the latent parameters based on the sample $z_s$ that gives the highest posterior according to Equation 6 across all of the iterations for the 10 runs. Empirically, the sampler reaches a plateau of the posterior after approximately 50 iterations with each iteration taking a few seconds on a 2GHz PC running a Matlab implementation.

For pLSA, the EM algorithm is initialized randomly and typically converges in 40–100 iterations. One iteration takes about 2.3 seconds on 4K images with 7 fitted topics and ~300 non-zero word counts per image (Matlab implementation on a 2GHz PC). For more details, see [24].

Note that for each model used in this paper, it is necessary to specify the number of topics $T$ to discover. However, Bayesian [28] or minimum complexity methods [3] can be used to infer the number of topics implied by a corpus. We do not investigate the problem of inferring the number of topics in this paper.

4.1. Topic Discovery

We carry out two experiments of increasing complexity and recover the visual words that correspond to the discovered topics. For each experiment, we specify the number of topics to discover and then fit an LDA model to the data. To visualize the visual words that correspond to each discovered topic, we use the recovered $\phi$ multinomial parameters to display those words with high probability for a given topic.

Expt. (1) Images of four object categories with cluttered backgrounds. We use images from the faces, motorbikes, airplanes, and cars rear object categories from the Caltech 101 dataset. All four of these categories have cluttered backgrounds and vary significantly in scale (especially in the case of car rear). Figure 5 shows the two top words for each of the discovered topics when $T = 4$. Notice that the top words for each topic correspond to semantically meaningful regions for the four object categories (e.g. eyes for faces, wheel parts for motorbikes, nose tips for airplanes, and license plates for cars rear). It appears that we are discovering the four object categories!

\footnote{As noted before, in the future we plan to apply a constraint-based gradient method to Equation 6. For this, we will properly perform Gibbs sampling by requiring burn-in and lag time between drawn samples.}
Examples from Caltech 101

Examples from MIT Objects and Scenes

Figure 4: We pool images from five object categories from the Caltech 101 dataset (top row). The categories and number of images used are: faces, 435; motorbikes, 800; airplanes, 800; cars rear, 1155; background (indoor and outdoor scenes around the Caltech campus), 900. We chose to experiment with these categories since they offer the greatest number of examples per category, thereby increasing the chance of success. All images have been converted to grayscale before processing. No other alterations were made with the exception of removing a white border around a number of motorbike images since this was providing an artifactual cue. The MIT Objects and Scenes dataset (bottom row) contains 2,873 images and indoor and outdoor scenes, with annotations consisting of polygonal outlines. Again, these images were converted to grayscale before processing.

**Expt. (2) Images of four object categories plus “background” category.** We add to Expt. (1) the set of “background” images from the Caltech 101 dataset. Since the set of images belonging to the other four object categories contain some background, we would like to discover a background topic and better model those images containing foreground and background. We fit an LDA model with $T = 3$ to just the set of background images to get the most likely words (based on $\phi$) for each discovered topic as shown in Figure 6. These words show the type of scenes contained in the dataset.

4.2. Clustering Images

Since it appears that we are discovering topics corresponding to object categories, we would like to assign each image to one of the discovered topics. We do this by using the recovered $\theta$ multinomial parameters to assign each document to the topic with the highest probability. To evaluate the goodness of these assignments, we compute a confusion matrix where each column corresponds to the distribution of the images in an object category to the discovered topics. For example, if there are $C$ object categories in the image set and $T$ topics are discovered, then the resulting confusion matrix will be of dimension $T \times C$ with columns summing to one. An ideal confusion matrix, up to some permutation of the rows, would be the identity matrix. For a given confusion matrix, we show an intensity plot of the matrix for the best permutation of the rows and report the average along the diagonal and the number of images that appear off of the diagonal.

Figure 7 shows confusion matrices and Table 1 show summary statistics for Expt. (1) as the $T$ parameter is increased from 4 to 7. For $T = 4$, we get an almost perfect assignment score for the four categories. As the number of found topics is increased, certain object categories seem to cluster into subtopics. This seems to happen mostly with the motorbike and car rear object categories. Upon looking at the clustered images, we find that these subtopics correspond to motorbikes containing background versus no background and the car rear set containing many almost identical examples of two particular cars. These discovered subtopics agree well with the structure of the data.

Figure 8 and Table 1 show confusion matrices and summary statistics for Expt. (2) as the $T$ parameter is increased from 5 to 7. Notice that for both this and the earlier experiments the LDA and pLSA models significantly outperform the baseline $k$-means method and that there is little difference in the LDA and pLSA performance. For LDA, there is some confusion between the background and the desired object categories. For example, for $T = 5$ a significant number of background images are incorrectly assigned as faces. Another example is for $T = 7$ where some motorbikes and cars are incorrectly assigned to background topics. This seems to occur because, as noted above, many images contain a mixture of visual words corresponding to foreground and background. Examples include faces occurring in office scenes and cars appearing in road
Table 1: Confusion matrix summary statistics of the experiments comparing LDA, pLSA, and the k-means baseline method. The ‘%’ column is the average along the diagonal of the confusion matrix corresponding to a given experiment and the ‘#’ column is the number of misclassified images. For the case of (2)*, the two/three background topics are allocated to one category. See the text and Figures 7 and 8 for more details. Notice that both LDA and pLSA have comparable results and outperforms the baseline k-means method.

<table>
<thead>
<tr>
<th>Expt.</th>
<th>Categories</th>
<th>T</th>
<th>LDA</th>
<th>pLSA</th>
<th>KM baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>4</td>
<td>4</td>
<td>97</td>
<td>86</td>
<td>98</td>
</tr>
<tr>
<td>(2)</td>
<td>4 + bg</td>
<td>5</td>
<td>78</td>
<td>931</td>
<td>78</td>
</tr>
<tr>
<td>(2)*</td>
<td>4 + bg</td>
<td>6</td>
<td>84</td>
<td>656</td>
<td>76</td>
</tr>
<tr>
<td>(2)*</td>
<td>4 + bg-fxd</td>
<td>7</td>
<td>78</td>
<td>1007</td>
<td>83</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrices for unseen test images in Expt. (3) for LDA and pLSA. The test images comprise examples from four object categories (there are no background images). Notice that there is little confusion between the different categories.

<table>
<thead>
<tr>
<th>True Class →</th>
<th>Faces</th>
<th>Motorbikes</th>
<th>Airplanes</th>
<th>Cars rear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic 1 - Faces</td>
<td>97.70</td>
<td>0.50</td>
<td>3.75</td>
<td>1.00</td>
</tr>
<tr>
<td>Topic 2 - Motorbikes</td>
<td>1.38</td>
<td>96.50</td>
<td>2.25</td>
<td>0.00</td>
</tr>
<tr>
<td>Topic 3 - Airplanes</td>
<td>0.00</td>
<td>0.75</td>
<td>93.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Topic 4 - Cars rear</td>
<td>0.92</td>
<td>2.25</td>
<td>1.00</td>
<td>99.00</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrices for unseen test images in Expt. (3) for LDA and pLSA. The test images comprise examples from four object categories (there are no background images). Notice that there is little confusion between the different categories.

4.3. Classifying New Images

After fitting an LDA model to a corpus of images, we can use the learned \( \phi \) parameters to classify unseen images. To find the \( \theta \) parameters for these novel images, we follow a similar path as in Section 2.1, except we now condition on the \( \phi \) parameters found during training. We obtain a similar Gibbs sampling solution:

\[
p(z_i = t | \phi, z_{-i}, D, \alpha, \beta) \propto (f_{d,t}(D_{-i}, z_{-i}) + \alpha_t) \phi_{t,v} \quad \text{if } t \in \text{FIXED}
\]

\[
\left( f_{d,t}(D_{-i}, z_{-i}) + \alpha_t \right) \left( f_{i,v}(D_{-i}, z_{-i}) + \beta_v \phi_{t,v} \right)
\]

where \( \text{FIXED} \) denotes the set of topic indices that are held fixed. Figure 8 and Table 1 show the improvement in the confusion matrix and summary statistics. Again, there is little difference in performance between this semi-supervised version of LDA and an equivalent version of pLSA (described in [24]).
Table 3: Equal error rates for the classification task of object versus background for LDA, pLSA, and [9]. The object category test images were classified against (a) the set of 500 testing background images (this test was performed in [9]) and (b) the set of testing background images and testing images of all other object categories. The small confusion between the different categories explains the improvement in performance in column (b). (*) For the cars rear category, we classify against a separate set of background images consisting of road scenes (as in [9]). For training, we experimented with using 400/900 background images respectively.

**Expt. (4) Binary classification of category against background.** We investigate binary classification of unseen images containing objects versus unseen background images and compare with [9]. For training, we use the same set of images containing objects as in Expt. (3) but use only 400 (out of 900) background images. We then fit an LDA model with $T = 7$ on this set of images. The test set (the remaining images containing objects and the 500 background images) is used to produce ROC curves for each discovered object topic. We report the equal error rates (the point where the probability of detecting the object category equals one minus the false alarm rate) for LDA, pLSA, and [9] in Table 3. For the cars rear category, to be comparable with [9], for testing we classify against a separate set of road scenes. Also, for the cars rear category, we experimented with training on all of the background images. Notice that LDA and pLSA obtains comparable performance with [9] even though we do not supply any training labels for the images.

### 4.4. Segmentation

Given a fitted LDA model, we can use the inferred $\theta$ and $\phi$ parameters to compute the probability that a given visual word in an image belongs to a particular topic

$$p(z^d_i = t| D, \phi, \theta, \alpha, \beta) \propto \theta_{d,t} \phi_{t,w_i^d}$$

Figure 9 shows two examples of images segmented using Equation 15 by displaying those words that have a probability greater than 0.88 of belonging to a particular topic for the case of Expt. (2) with $T = 7$ and the parameters of 3 background topics pre-learned and held fixed. For both examples, it appears that the topic labels assigned to the visual words align nicely with the different regions of the image.

To evaluate the goodness of the segmentations, we use a set of ground-truth bounding boxes for the set of face images [19] and compute a score that measures the percentage of overlap between the ellipses used for segmentation and the bounding box:

$$\rho_i = \frac{B_i \cap E_i}{B_i \cup E_i}$$

where $B_i$ is the set of pixels inside the bounding box for image $i$ and $E_i$ is the set of pixels inside all of the ellipses used for segmenting the face in image $i$. We compute a score $\rho$ over all of the face images by taking the average of the $\rho_i$. For all of displayed segmentations in this paper that has the set of face images in the training corpus, we set the threshold for displaying a word (e.g. 0.88 above) to be the one that empirically maximizes this score. For the segmentation experiment above, we get $\rho = 0.50$.

It seems quite remarkable that we are able to get a segmentation with the bag-of-words assumption since all spatial structure between the words is thrown away. However, it is important to note that the words themselves and the overlap that occurs between neighboring words seem to retain much of the spatial structure of the desired objects (i.e. we would probably get a completely different result if the pixels were randomly shuffled).

Notice that Equation 15 depends not only on the probability of a visual word occurring in a particular topic, but also on the probability of that topic occurring in the given image. This allows us to overcome the phenomena of polysemy. In the text domain, polysemy occurs when a word has two different meanings (e.g. ‘bank’ as in (i) a money keeping institution, or (ii) a river side). Figure 10 shows an example of a visual word with relatively high probability in two different topics. For these examples, the words are assigned to the correct topics for the segmentation task.
Expt. (5) Image segmentation for faces. We now investigate how doublets can improve image segmentation. To illustrate this clearly, we experiment with only the face and background images. We learn a set of parameters for the background class by fitting an LDA model with $T = 3$ on half of the background images (400 images in total). We then fit an LDA model with $T = 4$ on the faces and remaining background images with the learned background parameters held fixed. A doublet vocabulary is then formed from the top 100 visual words for the face topic using the inferred $\phi$ parameters. We then combine the original vocabulary (cf. singlets) and the new doublet vocabulary and repeat the above steps using the new vocabulary to fit an LDA model for $T = 4$ with fixed background parameters. We initially fit an LDA model on the singlet vocabulary to reduce the size of the doublet vocabulary. Figure 11 shows examples of doublets and compares the resulting segmentation using singlets versus doublets. Notice that we get a cleaner segmentation with doublets. Furthermore, the percentage of overlap increases from $\rho = 0.51$ for singlets to $\rho = 0.56$ for doublets.

If we perform Expt. (2) and form doublets from the top 100 words in all of the non-background topics using pre-learned fixed parameters for the background, then the overlap score decreases to $\rho = 0.51$. This suggests that there is some benefit in topic-specific doublets.

Notice the level of supervision to achieve this segmentation: the images are an unordered mix of faces and backgrounds. It is not necessary to label which is which, yet both the face objects and their segmentation are learned. The supervision provided is the number of topics $T$ to learn and the separate set of background images to pre-learn the background topics.

Expt. (6) Image Segmentation on the MIT Dataset. We consider the entire MIT Objects and Scenes dataset. We fit an LDA model with $T = 10$ and form doublets using the top 40 visual words from each topic according to the recovered $\phi$ parameters. A second LDA model is fitted using the combined singlet and doublet vocabulary. Figures 12 and 13 show examples of segmentations induced by 4 of the 10 learned topics. These topics, more so than the rest, have a clear semantic interpretation, and cover objects such as computers, buildings, trees, and bookshelves. Notice that the results clearly demonstrate that: (i) images can be accessed by the multiple objects they contain (in contrast to GIST [17], for example, which classifies an entire image); (ii) the topics induce segmentations of multiple instances of objects in each image.

5. Conclusions

We have demonstrated that it is possible to learn visual object classes simply by looking. To do this, we have introduced the Latent Dirichlet Allocation model and have shown how to convert images to the text domain by means of vector-quantized SIFT features and doublets, which form a vocabulary of visual words. For a given set of unlabelled images, we discover meaningful object parts and cluster the images into the appropriate object classes with high reliability (cf. Table 1). We are able to improve the results for those objects that cause confusion by training on side data and then fixing the learned topic parameters and learning the remaining parameters on the rest of the dataset.

We reproduced the experiments of [9] using a set of training images to fit an LDA model and then using the learned parameters to classify a set of unseen images. For the training phase, we did not provide any class labels; we only specified the number of topics to discover. We obtained very competitive performance without much supervision.

Finally, we showed that we are able to localize objects remarkably well, in spite of the bag-of-words assumption, using those visual words with the highest posterior probability. Since this posterior depends on the likelihood of an object class occurring in an image in addition to the likelihood of a visual word occurring in an object class, we showed that this allows us to overcome polysemy. We improved the segmentation results by incorporating the doublet visual words.

Acknowledgements

Financial support was provided by the National Geospatial-Intelligence Agency, NEGI-1582-04-0004, and a grant from BAE Systems. We would like to also thank Josef Sivic, Alyosha Efros, and Andrew Zisserman for their collaboration.

References


[30] [http://www.robots.ox.ac.uk/~vgg/research/affine/](http://www.robots.ox.ac.uk/~vgg/research/affine/).


Figure 5: The two most likely words (shown by 5 examples in a row) for four learned topics in Expt. (1): (a) Faces, (b) Motorbikes, (c) Airplanes, (d) Cars.
Figure 6: The most likely words (shown by 5 examples in a row) for the three background topics learned in Expt. (2): (a) mainly local feature-like structure (b) mainly corners and edges coming from the office/building scenes, (c) mainly textured regions like grass and trees.
Figure 7: LDA confusion tables for Expt. (1) and increasing number of topics discovered ($T=4,5,6,7$). Brightness indicates number, with the ideal being bright down the diagonal. The rows/columns correspond from top/left to bottom/right as faces, motorbikes, airplanes, and cars rear. Notice that we learn interesting “subtopics” in the car and motorbike categories. For the motorbike case, these subtopics correspond to motorbikes with or without background. For the cars, there appears to be a couple of particular cars with many repeated examples in the dataset, with topics assigned to these cases.

Figure 8: Confusion tables for Expt. (2) for LDA (top row) and for pLSA (bottom row). The number of discovered topics is increased from $T=5$ to $T=7$, with the last column depicting $T=7$ and fixed background parameters. Brightness indicates number, with the ideal being bright down the diagonal. The rows/columns correspond from top/left to bottom/right as faces, motorbikes, airplanes, cars rear, and the set of background images. Notice how we remove some of the confusion between the background images and that we discover background “subtopics” when we increase the $T$. 
Figure 9: Image as a mixture of visual topics (Expt. (2)) using 7 learned topics with fixed background parameters. (a),(c) Original images. (b),(d) Images as mixtures of visual topics. Visual words with topic posterior given in Equation 15 greater than 0.88 are shown. In total, there are 926 and 391 elliptical regions in the top and bottom row images respectively. A key to the colors is given in (e) along with the number of visual words assigned to each topic and the probability of each topic occuring in the two images. Notice that the displayed visual words agree well with the object categories.
Figure 10: Example of a polysemic visual word (this word has high probability in two different topics). Each row depicts examples of the visual word occurring in two different topics. For the segmentation task, we are able to overcome polysemy since the probability that a given visual word in an image belongs to a topic, as given in Equation 15, depends on the likelihood of the topic occurring in the image in addition to the likelihood of the visual word occurring in the topic. For each of these examples, the word was assigned to the correct topic and hence induced the correct segmentation.

Figure 11: Improving object segmentation. (a) The original frame with ground truth bounding box. (b) All 416 detected elliptical regions superimposed on the image. (c) Resulting segmentation after fitting an LDA model to the “singlet” visual words in Expt. (5). (d) and (e) show examples of ‘doublets’. (f) Segmentation obtained using doublets. Notice that we get a cleaner segmentation using doublets. For (c) and (f) respectively, the threshold that we use for displaying the singlets is 0.89 and doublets is 0.70.
Figure 12: Example segmentations induced by four (out of 10) discovered topics on the MIT dataset. Examples from the first 20 most probable images for each topic are shown. For each topic the top row shows the original images and the bottom row shows visual words (doublets) belonging to that particular topic in that image. Note that we can give semantic interpretation to these topics: (a) covers computers; (b) covers building regions; (c) covers bookshelves; (d) covers trees and grass.
Figure 13: Example segmentations on the MIT dataset for the 10 topic decomposition. Left: the original image. Middle: all detected regions superimposed. Right: the topic induced segmentation. The topics depicted are from Figure 12. The color key is: a-cyan, b-red, c-magenta, d-green. Notice that each image is segmented into several ‘topics’.