Link analysis techniques for object modeling and recognition

Gunhee Kim

CMU-RI-TR-08-14

May 2008

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

Abstract

This paper proposes a novel approach for unsupervised modeling and recognition of object categories in which we first build a large-scale complex network which captures the interactions of all unit visual features across the entire training set and we infer information, such as which features are in which categories, directly from the graph by using link analysis techniques. The link analysis techniques are based on well-established graph mining techniques used in diverse applications such as WWW, bioinformatics, and social networks. The techniques operate directly on the patterns of connections between features in the graph rather than on statistical properties, e.g., from clustering in feature space. We argue that the resulting techniques are simpler, and we show that they perform similarly or better compared to state of the art techniques on both common and more challenging data sets.

Also, we extend this link analysis idea to combine it with the statistical framework of topic contents. By doing so, our approach not only dramatically increases performance but also provides feasible solutions to some persistent problems of statistical topic models based on bag-of-words representation such as modeling of geometric information, computational complexity, and the inherent ambiguity of visual words. Our approach can be incorporated in any generative models, but here we consider two popular models, pLSA and LDA. Experimental results show that the topic models updated by adding link analysis terms significantly outperform the standard pLSA and LDA models. Furthermore, we presented competitive performances on unsupervised modeling, classification, and localization tasks with datasets such as MSRC and PASCAL2005.

This research was performed in part for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Program funded by the Ministry of Commerce, Industry and Energy of Korea.

Thesis supervisor: Prof. Martial Hebert

Contents

1 Introduction

This paper presents novel approaches to unsupervised modeling and recognition of object categories inspired by recent success in the research on a large-scale network. Our approach is unique that all low-level visual information of image sets are represented by a single large-scale network and then link analysis techniques are applied in order to mine the visual models in an unsupervised way and perform classification and localization of unseen images. Experimentally, our approaches showed almost best performances over the previous work for several different image datasets.

Starting from mathematical graph theory, the study of networks has been recently moved to the focus on statistical analysis of a large-scale network [29]. Examples include the Internet, the World Wide Web, human social networks, metabolic networks, and food webs. In particular, the success of search engines of World Wide Webs clearly suggests that the research on a large-scale network is promising in real world applications. The searching in the WWW is a challenging task since the WWW is a complex system in the sense that it is totally decentralized and uncontrollable. That is, anybody can add any information without any notice to others. Also, the searching requires fast responses to queries by users in a matter of milliseconds. However, several real search engines give us quite satisfactory results in a very short time and their success is largely based on the fact that the WWW is a network and recent significant progresses on the understanding the properties of a large-scale network. According to [29], some of the primary focus on the network study include (1) What properties do different real-world networks share? and (2) How do we discover those properties? In this work, we take advantage of two common properties of networks - *Non Uniform Degree Distributions* and *Community Structures in Network*, which are detected by *link analysis techniques*.

This paper contains two independent approaches to unsupervised modeling and recognition of object categories although they share the same line of thought, *link analysis techniques for visual tasks*. In the section 2, we propose a method of construction of visual network and an unsupervised modeling based on two different types of link analysis techniques. In the section 3, the link analysis idea is combined with statistical topic models widely used in computer vision.

2 Unsupervised Modeling Using Link Analysis Techniques

2.1 Introduction

Unsupervised visual modeling of object categories involves the extraction of parts (regions or sets of features) from unlabeled images corresponding to different, unknown object categories. This problem, typically addressed with statistical machine leaning tools, remains one of the most challenging visual perception problems. Here, we propose a novel representation and formulation of the problem inspired by the tools commonly used for the analysis of complex networks such as the WWW and social networks.

Specifically, we construct a single large graph which captures the interactions of all visual features in the training set, called the *visual similarity network*. The basic idea of the approach is that (1) if feature i has a significant number of consistent matching from others, then the feature i should be important and (2) if feature i from one image matches features in the other images that are mostly the same as those matched by another feature j, then i and j are more likely to belong to the same object. This type of reasoning relies only on the configuration of the graph of features (*i.e.* which feature matches which features in the entire training set) and powerful link analysis techniques can be brought to bear to mine global matching consistency from the graph for the inference of visual models.

Our approach is analogous to the Web search engines, which are successful examples of extracting information by analyzing the interactions of data elements with each other. In this case, despite the large amount of data, it is possible to retrieve accurate information within a very short time by analyzing the interactions between the web pages (specifically, which page *links* which other pages) without explicitly describing their content. In particular, we will extract visual models from unlabeled data by using primarily the PageRank [6] and the vertex similarity algorithms [5] as link analysis techniques.

Because we need to represent in principle all pairwise relations between all visual features, the representation could rapidly grow to become intractable. In practice, however, the resulting representation is compact and efficient because the graphs involved are very sparse and, like other complex networks such as those found in social networks, WWW, or epidemiology, the graph used in our approach follows the power law (*i.e.* it is a "scale-free" network) [1]. In other words, in practice, there is some small number of features acting as *hubs* compared to the total number of features, and most features are less important. Intuitively, when we gather images of object classes and generate a graph of all the features, there is a small number of class representative visual information, and most of the features are likely to be irrelevant (such as background) to the discovery of category models. Therefore, by applying link analysis techniques, we can quickly filter out large amount of non-essential visual information. Another reason for the efficiency of the approach is that, once the network is constructed, we do not need to access the feature data. All inference for unsupervised modeling is based on link analysis of the graph, and thus the computation is quite fast.

2.2 Related Work

Our approach is unique in describing all of the visual interactions explicitly in a single view, by formulating low-level visual information as a complex network and by applying link analysis techniques for visual modeling. Here, we introduce some previous work which is closely related to ours.

A common way to approach the unsupervised modeling problem is to first quantize the input into a discrete set of words, which are then used for extracting groups of features corresponding to categories [15, 12, 36, 14]. However, unlike the text analysis domain from which it originated, this identification process is not straightforward. Since, unlike words in text, there are no natural boundaries, orders, and clear semantic meaning of visual words, the definition of visual words and their assignments to instances of local features are in themselves challenging. For this reason, there is no dominant methods for dictionary formation (*e.g.* hierarchical agglomerative clustering[32] or k-means [7, 36]), the optimal selection of dictionary sizes, and the assignment of codewords to each feature instance (*e.g.* soft or hard assignment). In contrast, we do not try to identify each visual entity (*i.e.* the definition and assignment to *codewords*) but focus on its *interactions with others*. In this approach, the data describing each feature, such as its position, orientation, scale, and descriptor, is used only for defining its relationship with other features.

Grauman and Darrell [17] applied the pyramid match kernels to unsupervised modeling. Their work is similar to ours in that they use image-based matching and spectral clustering for final classification results. However, their algorithm relies on the results of image matching but it does not take advantage of explicit interactions between individual low-level features. We compare experimentally with their approach.

Todorovic and Ahuja [38] proposed an unsupervised modeling method based on tree matching. Segmented regions are represented as nodes in a tree, inference of models is performed by tree matching. Although the tree structure can support complex hierarchical relationships, the complexity of the approach is on the order of the fourth power of the number of nodes.

Data mining and link analysis techniques have been used in computer vision tasks. Quack *et al.*'s work [32] applies well-known data mining techniques, termed *frequent itemsets* and *association rules*, to the feature selection. However, their work differs from ours in that it requires bounding box annotations, do not use any networks, and only applied two-class cases (positive and negative image sets). Pan *et al.*'s work [31] applies the PageRank algorithm to image data. Their task is auto-captioning in which, given a novel image, the most probable caption words are assigned by using the PageRank algorithm. However, their work requires labeled training sets (*i.e.* caption words should be annotated to the segmented regions in training images). Also, their task is a labeling problem rather than a visual perception task.

2.3 Construction of *Visual Similarity Networks*

The basic representation on which we will operate is a weighted directed graph, termed the *visual similarity network* (VSN). The nodes of the VSN are the features extracted from all of the training images, and the edges of the graph link features that have been

matched across images.

We denote the set of training images by $\mathbb{I} = \{I_a\}_{a=1,\ldots,m}$, from which we wish to extract K categories. We denote image indices by $a, b, \ldots$ and feature indices by $i, j, \ldots$. The VSN is a graph $G = (V, E, W)$, where V is the set of vertices, E is the set of edges, and W is the set of edge weights. We also denote the adjacency matrix of the VSN by M. Each node in V is denoted by a_i, representing the i-th feature in image I_a. We denote the total number of features by n, and the number of features in image a by n_a.

In general, a node in the VSN can be any unit of local visual information. Here, we use the standard Harris-Affine interest point detector [27] and the SIFT descriptor [26]. That is, all affine covariant regions extracted from all the training images form the set of nodes (V) of the VSN. We now describe the procedure used to form the links in E, and finally the way the weights in W are computed.

2.3.1 Establishing edges in the VSN

The links are established by finding matches between features in different images. In our case, we apply the spectral matching of [22, 23] to each pair of images (I_a, I_b). This approach combines matching based on local data with geometric consistency constraints by finding a combination of correspondences that is globally most consistent, based on pairwise relations between features. The algorithm requires an initial set of potential correspondences based on local appearance, which we obtain by using the L_2 distance between SIFT descriptors. The second-order geometric affinity is calculated by the over-complete set of translation invariant geometric relations proposed in [24]. The advantage of this particular matching technique is that it is fast and simple and it has been shown to have good limit properties in the context of inference in problems defined by first- and second-order potentials. Other similar matching techniques such as the deformable matching of [2] or pyramid matching of [17] could be used as well.

After matching all the pairs of images, each correspondence between features a_i and b_j forms a new edge between the two corresponding nodes in V: $e = (a_i, b_j)$. It is important to note that, at this stage, we do not require the matching to be accurate. The resulting graph may be quite noisy, especially because of extra edges between images that do not contain any common objects. The link analysis algorithm will be responsible for dealing with the (possibly large number of) incorrect correspondences. In fact, we do not want the matching to be *too strict*, in which case it might find very few correspondences which will not be enough to populate the graph. To further relax the matching, we allow many-to-many correspondences between a pair of images, by iterating the one-to-one algorithm of [22]. Also, we use a fixed model of the pairwise geometric relations, rather than a more accurate model learned from training data as in [24]. The final output of this process is a set of potential correspondences E.

The resulting graph is not necessarily symmetric: If I_a (the *query* image) is matched to I_b (the *reference* image), then the initial correspondences are obtained by retrieving the k-nearest features b_j to each feature a_i (by using the L_2 distance between SIFT descriptors). If the order of the query and reference images is reversed, we will instead retrieve the k-nearest features to each feature in I_b, which will yield a different set of initial correspondences. As a result, the edge (a_i, b_j) does not necessarily exist if the

(b_j, a_i) exists. If they both exist, their weights (described in the next section) will be different in general.

Figure 1 shows an example of edge construction by comparing the same image on the left to two different images from different object classes on the right. There is a substantial number of wrong correspondences in the bottom pair because the object classes are different between the two images (*i.e.* an *giraffe* on the left and a *car* on the right.) However, if the matching behavior is consistent, the link analysis techniques introduced below will be able to extract the major trends from noisy correspondences.

2.3.2 Computing the edge weights

The weight w_e of the edge $e = (a_i, b_j)$ should reflect how consistent that correspondence is with all the other correspondences obtained in matching I_a and I_b. A higher weight would indicate more confidence in the correspondence, meaning that many other correspondences would agree with it. In order to describe how w_e is computed, we need to look more closely at the spectral matching approach [22]: For the matching of a pair of images, a matrix Q is first created with one row and one column for each potential correspondences (a_i, b_j). Here, we abbreviate the notation (a_i, b_j) to simply ij since we are dealing with a single pair of images I_a and I_b in this paragraph. $Q(ij, i'j')$ contains the pairwise geometric consistency between correspondences ij and $i'j'$ such that the more deformation is needed to map the pair ij to the pair $i'j'$, the lower $Q(ij, i'j')$ is. The solution is found essentially by computing the principal eigenvector of Q and binarizing it by following the procedure described in [22]. We denote by $\mathbb{C}^*$ the set of correspondences that are selected at the end of the matching. An estimate of the confidence of $(a_i, b_j) \in \mathbb{C}^*$ is given by: $C_{ij} = \sum_{i'j' \in \mathbb{C}^*} Q(ij, i'j')/|\mathbb{C}^*|$. C_{ij} measures how well the correspondence ij agrees with all the other correspondences. In practice, Q is constructed such that $0 \leq C \leq 1$ for all the correspondences.

One problem is that we cannot simply set w_{ij} to be equal to C_{ij} since we do not know in advance what, in absolute terms, is a *good* value for C. To address this problem, we take into account the histogram of the confidence values and the rank ordering of each C_{ij} rather than the absolute values. More precisely, let $\{t_l\}, l = 1, \ldots, N_t$ be a set of thresholds such that $t_1 < \ldots < t_{N_t}$. We define $\mathbb{C}_l$ as the set of surviving correspondences ij such that $C_{ij} > t_l$. Clearly, $\mathbb{C}_{N_t} \subset \mathbb{C}_{N_t - 1}, \ldots, \subset \mathbb{C}_1 \subset \mathbb{C}^*$. In practice, we use five thresholds regularly spaced between 0.8 and 0.4. For example, in Fig.1, the red features belong to a subset of strongly geometrically consistent features for $t = 0.8$ while the blue ones have weaker consistency for $t = 0.4$.

Finally, the weight w_e of the edge $e = (a_i, b_j)$ is defined as:

$$w_e = \frac{1}{n_a} \sum_{l=1}^{N_t} S_l \mathbf{1}_{\mathbb{C}_l}(ij) \tag{1}$$

where S_l is the normalized score $S_l = \frac{\sum_{ij \in \mathbb{C}_l} C_{ij}}{|\mathbb{C}_l|}$ which measures the global consistency of the set of correspondences that are over a particular threshold. $\mathbf{1}_{\mathbb{C}_l}(ij)$ is the indicator function which is defined by 1 if the correspondence ij is in the set $\mathbb{C}_l$. The division by n_a is intended for the normalization of irregular number of features in

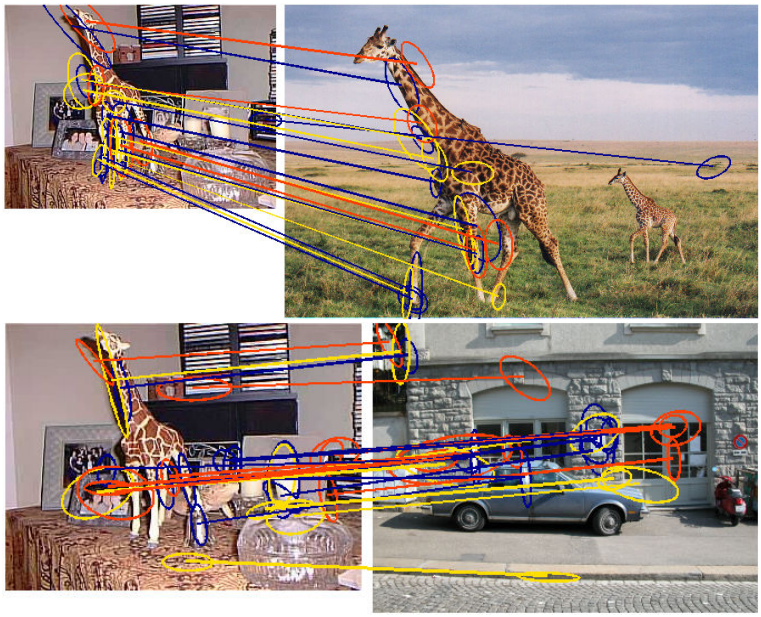


Figure 1: Example of link generation on two pairs of images. The features are matched by using a liberal spectral matcher. The jet colormap is used from red (strong) to blue (weak geometric consistency). (These figures are best viewed in color.)

each image. Intuitively, ten correspondences from an image with a total of 50 features should be weighted more than ten votes by an image with 100 features.

Consequently, by iterating the matches between all pairs of image, we obtain a sparse $n \times n$ matrix M (*i.e.* the adjacency matrix of the VSN G) where $M(a_i, b_j)$ is the value of the weight w_e of the edge $e = (a_i, b_j)$.

2.4 Inference of Object Models from Networks

2.4.1 Ranking of Visual Information

Since we use all the features in the training images, and since the links are very noisy, we need a mechanism to estimate the relative importance of nodes in the graph. This can be done by treating the weights associated with the links in the VSN as *votes* for the importance casted by other nodes. Even though there will be a lot of false links from different classes or even background, they are highly likely to have higher variations in those linking behaviors than the links between the nodes of the same objects, which will vote more consistently. In order words, *hubs* in a given class are likely to be formed through consistent matches with features in the same class.

Well-known ranking algorithms such as PageRank [6] and Kleinberg's HITS algorithms [21] can estimate the ranked importance of nodes in a graph using only the graph

connectivity. In our experiments, PageRank slightly outperforms the Kleinberg's HITS and it is the one that we use as the baseline algorithm. In its most general form, the ranking algorithm generates the $n \times 1$ PageRank vector P by solving the equation [6]:

$$P = (1 - \alpha)(M + D)P + \alpha v, \tag{2}$$

where M is the weight matrix of the graph (the VSN in our case), α is a constant close to one (in all of our experiments $\alpha = 0.9$), v is the *transport* vector ($= [\frac{1}{n}]_{n \times 1}$, uniform probability distribution over all nodes), and $D = vd^{\mathrm{T}}$ (d is the n-dimensional indicator vector identifying the nodes with outdegree 0). Intuitively, the definition of ranking can be viewed recursively in the sense that components of P with high values are nodes connected to many nodes with high values.

We obtain the PageRank vector P_a for each image I_a in $\mathbb{I}$ by considering the portion of the VSN M obtained by considering the links between the nodes a_i and all of the other nodes from the other images. In other words, when computing P_a, we use the modified M_a for Eq.2 by enforcing $M_{ij} = 0$ *if* $i \notin I_a$ *and* $j \notin I_a$. This eliminates the interactions between the features irrelevant of the image I_a for the computation of P_a. Intuitively, a large $P_a(i)$ means (1) if $i \in I_a$, i is an relatively important feature in the image I_a (*i.e.* this information is valuable for *localization* of object in the image) or (2) if $i \notin I_a$, i is an highly relevant feature with respect to I_a (*i.e.* useful to clustering of the images according to object classes).

2.4.2 Structural similarity

In constructing the edges of the VSN (Section 2.3.1), we already consider two types of similarities, appearance similarity and geometric consistency. However, once we have a global representation of all the interactions between the features, we can infer another type of similarity termed *structural similarity*. The underlying observation is that similar nodes are highly likely to exhibit similar link structures in the graph. This observation is illustrated in Figure 2, in which both node i and node j are *wheel* features. Both nodes are highly likely to point out to and to be pointed to similar sets of features (*e.g.*, other *wheel* nodes), and at the same time both are highly unlikely to be linked to the same set of entities from different objects or background clutters.

Brondel *et al.* propose an algorithm which provides a generalized method to compute structural similarities between vertices of two directed graphs by using only link analysis [5]. The simplified version of the algorithm which we use here is the same as the algorithm used for automatic discovery of synonyms in a dictionary [5]. Given the VSN G, the *neighborhood graph* G_{ai} of a node a_i is the subgraph of G whose vertices are pointed to by a_i or are pointing to a_i. Let M_{ai} be the adjacency matrix of G_{ai}. M_{ai} is of dimension $N_{ai} \times N_{ai}$, where N_{ai} is the number of nodes in G_{ai}.

The algorithm of [5] defines the *central score* which is the similarity scores between the vertexes of G_{ai} and the vertex 2 of the path graph of length 3 of Eq.3. B is the incident matrix of the path graph. Intuitively, if a vertex $b_j \in G_{ai}$ has a high *central score*, then b_j and a_i are likely synonyms such that they contain the same words in their definitions and at the same time they are included in the definitions of the same words.

Operationally, the structural similarity values between G_{ai} and the graph of Eq.3 are computed by iterating Eq.4.

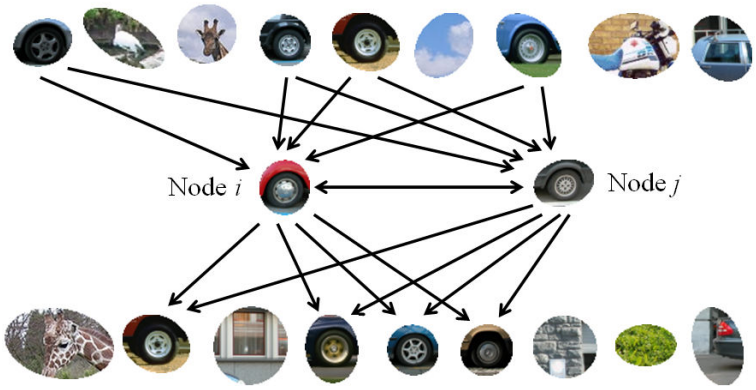


Figure 2: An example of structural similarity in a small part of the VSN G. Each link with its weight represents its local similarity measure (appearance and geometric consistency). The structural similarity captures the degree of similarity of the link structures of two nodes. The structural similarity is high if the two nodes match similar nodes like the *wheel* patches in this example.

$$1 \rightarrow 2 \rightarrow 3, \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{3}$$

$$U_{k+1} = \frac{BU_k M_{ai}^T + B^T U_k M_{ai}}{\|BU_k M_{ai}^T + B^T U_k M_{ai}\|_F}, \tag{4}$$

where U_k is a $3 \times N_{ai}$ matrix, initially set to $\mathbf{1}$, and $\|\cdot\|_F$ is the Frobenius norm. Upon convergence, $U_{ai} = \lim_{k\to\infty} U_k$ has the property that $U_{ai}(2, b_j)$ is the structural similarity value for each node b_j in the neighborhood of a_i [5]. In other words, a large value $U_{ai}(2, b_j)$ indicates that b_j and a_i share a lot of common nodes both in the incoming and outgoing directions.

This structural similarity algorithm is applied to each node a_i independently and the resulting similarity values are combined in a single $n \times n$ matrix Z, such that $Z(a_i, b_j)$ is the structural similarity of node b_j to a_i: $Z(a_i, b_j) = U_{ai}(2, b_j)$. Although n can be large, Z is very sparse in practice. We row-normalize Z to make the sum of vertex similarities of all the other features with respect to a feature to 1.

2.5 Unsupervised Modeling

From the link analysis described above, we have now two pieces of information: the PageRank vectors P_a for all the images $I_a \in \mathbb{I}$, which characterize how strongly the features of each image is related to all the other features from the other images, and the vertex similarity matrix Z, which characterizes how structurally similar the nodes are

with respect to each other. We will now use these two pieces of information in a two-step approach to unsupervised modeling. First, we will estimate which image belongs to which category. Roughly speaking, this step is the counterpart of the *topic discovery* step used in other approaches [36]. Second, for each category, we will estimate which features from the training images are relevant to that category. This is similar to the localization step used in other approaches with the critical difference that we do not attempt any clustering of the original features; we use directly the original features and we merely assess which feature is important for a given category. We argue that this can be done in large part by direct link analysis of the VSN without any clustering, statistical modeling, or other difficult manipulation of the actual feature values.

2.5.1 Category discovery

Our first objective is to partition $\mathbb{I}$ into K groups corresponding to the K categories. Of course, this is not optimal because it prevents the correct handling of images containing multiple categories (a case that is generally not handled by unsupervised techniques) and because it would be better to not make a hard decision on the partition of $\mathbb{I}$ before the next step. However, we feel that this is still an effective approach for demonstrating the feasibility of using the link analysis techniques for this problem.

The basic idea is to combine the m PageRank vector P_a and the $n \times n$ matrix Z into a single $m \times m$ affinity matrix A. $A(a, b)$ measures the affinity of I_b with respect to I_a and by combining 1) the total sum of $P_a(b_j)$ for the features in I_b, and 2) the total sum of the $P_a(a_i)$ of the features in I_a distributed proportionally to vertex similarities. $A(a, b)$ takes into account the entire graph structure to evaluate how confident we are that I_a and I_b contain a consistent collection of features corresponding to the same category. Intuitively, if they do, then they should be linked to similar groups of images, and many of their features should be structurally similar. In practice, A is computed as:

$$A(a,b) = \sum_{b_j \in I_b} P_a(b_j) + \sum_{a_i \in I_a, b_j \in I_b} P_a(a_i) Z(a_i, b_j), \tag{5}$$

for all pairs of images I_a and I_b. The groups of images corresponding to the K categories are estimated by partitioning A by using spectral clustering. Following [39], we use the Shi and Malik's Normalized spectral clustering [35] on the k-nearest neighbor graph. The k-nearest neighbor graph is easy to work with because it is a sparse matrix, and known to be less sensitive parameter settings [39]. In practice, we use $k = 10 \log(m)$ since [39] recommends that k be chosen in the order of $\log(m)$. After measuring the accuracy for different values of k, the variation of performance is less than 2% over the experiments reported below. This means that the matrix has strongly separated blocks in practice and that, therefore, the affinity measure is effective at separating the different categories. Figure 3 shows the affinity matrix computed from one dataset. This example shows that the groups of images corresponding to different categories are clearly separated by using the definition of affinity above.

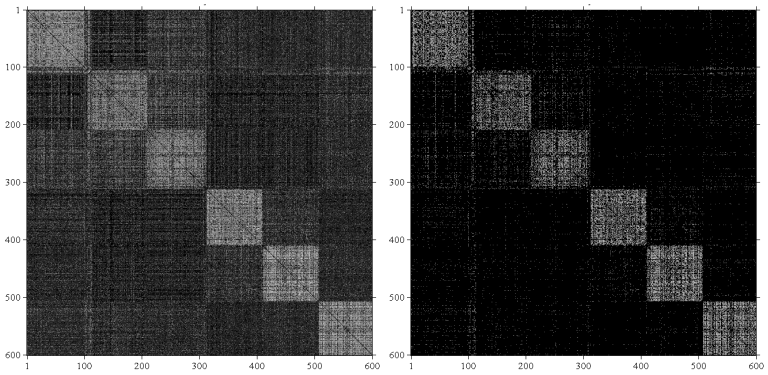


Figure 3: The raw affinity matrix A computed from 600 images of 6 categories of the Caltech-101 dataset (left) and the same matrix after retaining the $10\log(m)$ largest values for each node (right). The rows and columns have been ordered so that images in the same category are grouped together.

2.5.2 Localization

The objective of the localization step is to establish which features are relevant to each category. To do this, we first apply again the page rank algorithm, but this time to all the features in each category. More precisely, for each class c, we compute the page rank matrix P_c based on Eq. 2, replacing M by M_c, that is, the graph matrix obtained by using only the features from images in category c. The relative importance of each feature a_i from an image I_a in category c should be estimated by combining its own rank, $P_c(a_i)$ with the sum of all the other features' rank, weighted by their structural similarity to a_i, so that the importance of a feature will increase if many other features agree with it:

$$I_c(a_i) = P_c(a_i) + \sum_{b_j \in c} P_c(b_j) Z(b_j, a_i). \tag{6}$$

$I_c(a_i)$ can be interpreted as a confidence measure that each feature a_i belongs to the class c. As the last final localization step, we select the features whose importance are close enough to the maximum in the image: $I_c(a_i) \geq \rho \times max_{a_i} I_c(a_i)$. Different operating points are obtained by varying ρ as shown in the localization experiments below. All the examplar results shown later in this paper used $\rho = 0.8$ to be consistent with the top 20% rule used in [32].

2.6 Experiments

The input of our algorithm is a set of m unlabeled images with a single piece of information (*i.e.* the number of object categories K). The outputs are the classification of images according to object classes, and the ranked importance of all features with respect to their object categories, from which we can easily estimate the most probable locations of the objects in the images.

We evaluate the proposed unsupervised modeling method using two different datasets, which are Caltech101-dataset [11] and TUD/ETHZ dataset [1]{ETHZ *Giraffes*, TUD *Motorbikes*, TUD *Cars*}. By following the experimental setup proposed by Grauman and Darrell [17], we iterate the same experiment ten times, in which 100 and 75 images per object are randomly picked in each object class for the Caltech-101 and TUD/ETHZ dataset, respectively. We select only 75 images for the TUD/ETHZ experiments because there are only 83 images for the *giraffe* class.

2.6.1 Category discovery

For Caltech-101, we selected six object classes which have more than 100 training images – {*airplane*, *rear cars*, *faces*, *motorbikes*, *watches*, *ketches*}. We measure how well the unlabeled training images are clustered according to their categories by measuring the agreement of topic discovery with ground truth labels. Table 1 shows the confusion matrices for Caltech-101 classes. As shown in the results, our performance is competitive compared to previous work. In the case of four object classes, our results achieve 98.55% success ratio (compared to 98% in [36]). We outperform the Grauman and Darrell [17]'s method (86%) by more than 10% by using the same experimental as theirs. While related prior work generally goes up to four classes, we show that we can increase the number of classes with only a slow degradation in performance: 97.30% and 95.42% for five and six object classes, respectively. Also, for the TUD/ETHZ dataset, our method achieved 95.47% classification success ratio. Unlike the Caltech-101, this dataset has a lot of class variations and clutter in the background.

2.6.2 Localization

Localization is in general harder to measure and, in fact, most prior work evaluates classification and localization performance in separate experiments. For example, [7, 36, 38] designed simpler experimental setups for evaluating localization performance such as limiting the experiments to two category cases. Here, we evaluate the localization on the same setup as we used for evaluating classification, including up to six categories in the training set.

We use two metrics proposed by [32] - bounding box hit rates (BBHR) and false positive rates (FPR). Some papers use the segmentation ratios of intersections of detected regions and ground truth [7, 36] for the localization. But we feel BBHR and FPR would be better because we use Harris-Affine interest regions as our unit visual information instead of segmented patches that are more amenable to pixel-wise error. The bounding box hit (BBH) number is incremented for every ground truth box in which more than h features fall. We use $h = 5$, following [32]. The BBHR is the number of BBH divided by the total number of object instances in the dataset (BBHR=1 for perfect localization). The FPR is defined as the number of selected features lying outside the bounding box, divided by the total number of selected features (FPR=0 for perfect localization). In general, the FPR is a fairly severe measure because it counts the number of features without accounting for their spatial extent. For example, 10

[1]The TUD *Motorbikes* and *Cars* dataset is available at http://www.pascal-network.org/challenges/VOC/ and ETHZ *Giraffes* at http://www.vision.ee.ethz.ch/datasets.

	A	C	F	M
A	**98.4**±0.82	**1.0**±0.9	**0.1**±0.3	**0.5**±0.7
C	**0.2**±0.4	**99.8**±0.4	**0.0**	**0.0**
F	**1.9**±1.3	**0.1**±0.3	**98.0**±1.2	**0.0**
M	**1.4**±1.2	**0.6**±1.0	**0.0**	**98.0**±1.5

	A	C	F	M	W
A	**98.2**±1.2	**0.7**±0.8	**0.1**±0.3	**0.8**±0.4	**0.2**±0.4
C	**0.6**±0.7	**99.3**±0.8	**0.0**	**0.0**	**0.1**±0.3
F	**2.2**±1.3	**0.1**±0.3	**96.2**±1.7	**0.0**	**1.5**±1.5
M	**1.3**±0.8	**0.9**±1.1	**0.0**	**97.5**±1.6	**0.3**±0.7
W	**2.7**±2.1	**0.8**±0.4	**0.0**	**1.2**±1.0	**95.3**±1.9

	A	C	F	M	W	K
A	**94.5**±4.2	**0.5**±0.7	**0.0**	**0.5**±0.5	**0.3**±0.5	**4.2**±3.8
C	**1.1**±2.2	**97.1**±3.2	**0.0**	**0.0**	**0.0**	**1.8**±2.1
F	**1.5**±1.2	**0.0**	**95.6**±2.5	**0.0**	**1.8**±1.8	**1.1**±1.0
M	**1.4**±1.6	**0.4**±0.7	**0.0**	**93.5**±3.3	**0.1**±0.3	**4.6**±3.3
W	**2.2**±1.0	**0.3**±0.5	**0.0**	**0.3**±0.7	**93.4**±2.7	**3.8**±2.3
K	**1.5**±1.2	**0.0**	**0.1**±0.3	**0.0**	**0.0**	**98.4**±1.3

Table 1: Confusion tables for the Caltech-101 data set for increasing number of objects from four to six. The means and standard deviations of 10 runs for each are shown. The modeling accuracies (*i.e.* the averages of the diagonals) of four to six object categories are **98.55**% **97.30**%, **95.42**%, respectively. (A:*Airplanes*, C: *Cars*, F: *Faces*, M: *Motorbikes*, W: *Watches*, K: *Ketches*)

	M	C	G
M	**93.3**±2.7	**0.0**	**6.7**±2.7
C	**4.8**±2.6	**95.2**±2.6	**0.0**
G	**2.0**±1.1	**0.1**±0.4	**97.9**±1.4

Table 2: Confusion tables for the TUD/ETHZ shape dataset. The means and standard deviation values of 10 runs for each are shown. The classification accuracies (*i.e.* the averages of the diagonals) are **95.47**%. (M:*Motorbikes*, C: *Cars*, G: *Giraffes*)

misclassified features may give a high FPR even though they are clustered in a very small region. Unfortunately, for feature-based approaches, there is no absolutely fair measure, unlike patch-based methods for which a pixelwise error rate can be defined easily.

As proposed in [32], we generate FPR-BBHR curves by varying the relative threshold ρ (Fig.4). The plots show that our methods achieve reasonably low FPRs across the BBHRs. For some objects of caltech-101 dataset such as *watch* and *motorbikes*, the FPRs are fairly low since the objects are generally quite large in the image and only one object instance exist in most cases. The principal remaining source of errors is that, although our unsupervised classification is quite accurate, the misclassified images might produce the wrong localization result. For example, if a *face* image is misclassified into a *airplane*, the matched regions are unlikely to be on the correct object, which leads to localization errors. On the other hand, *faces* in Caltech-101 and *giraffes* in ETHZ

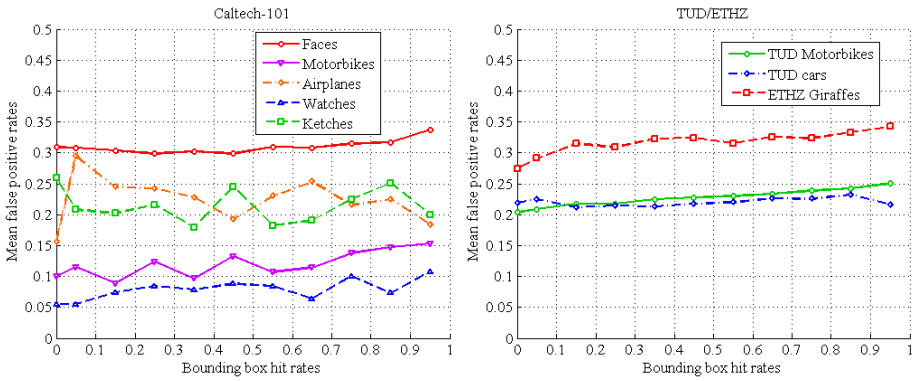


Figure 4: BBHR-FPR plots for Caltech-101(Left) and TUD/ ETHZ (Right) dataset.

dataset generate higher FPR values. This is primarily because there is relatively little background variation across some of the training images. For example, since trees in grassy plain are very often observed along with *giraffes* across the training images, it is natural that trees are also considered as important visual information for the *giraffes* class. In the case of *faces*, the higher FPRs are mainly due to the fact that the upper bodies (especially, shoulders) are always in the image with the faces, but the bounding boxes are located on the faces only.

Fig.5 shows some examples of the localization. Even though there are a lot of features in the background, the high confidence features are mostly on the objects. Some selected features on the background are low-ranked (colored blue). At the same time, the class representative features are fairly selected as *hubs* such as reddish *wheels* in the *car* class and *eyes* in the *face* class.

2.6.3 Computational issues

The VSN is represented by a $n \times n$ matrix, where n is the total number of features. However, in practice, the VSN is very sparse. For example, in the case of six object classes in Caltech-101, the number of nodes in the VSN is about 90,000. The VSN is quite sparse since the ratio of nonzero elements is about 5×10^{-4}. The sparseness of the vertex similarity matrix Z is about 0.002. However, since most of the non-zero elements have very low values, we could use an even sparser matrix by thresholding it. The basic operation used in the algorithm is the power iteration on matrices. Owing to the sparseness of the matrices involved, the complexity of the power iteration grows roughly linearly with n. This is similar to the behavior observed in other applications of the link analysis techniques [3]. In addition, motivated by the very large size of the matrices involved in Web applications, there has been a lot of work in optimizing the power method by taking advantage, among other things, of latent block structure and convergence acceleration techniques [3]. Although we did not use them in this work, these methods would enable scaling to much larger graphs.

Figure 5: Some examples of localization for the Caltech-101 and TUD/ETHZ dataset. In each image pair, the left image represents original extracted features with yellow, and the right image shows top 20% high-ranked features with color variance according to the importance weights. The jet colormap is used from red(high) to blue(low). (These figures are best viewed in color.)

3 Statistical Modeling and Recognition with Combination of Topic Contents and Link Analysis

3.1 Introduction

Generative topic models based on the *bag-of-words* representation have been a successful modeling and recognition tools in computer vision [13, 36, 34]. These models originated from statistical text analysis to automatically discover latent topics (i.e., object categories in most cases) in the training images based on the distribution of visual words.

This paper introduces the use of one general idea, *link analysis*, which has been used extensively in other research areas but which has been largely ignored before in computer vision. By combining link analysis with conventional topic models such as pLSA [18] and LDA [4], we not only propose some reasonable solutions to several key drawbacks of current topic models but we also report significant improvement of modeling and recognition performances. This work is inspired by some notable success in combining topic contents and link analysis in web applications such as intelligent surfers [33], web crawlers [25], and the analysis of blog topic influences [28]. The underlying observation is that all documents in a corpus have referential relationships with one another, which are as valuable clues as the contents of documents. By analogy, images that contain similar object categories tend to generate a larger number of

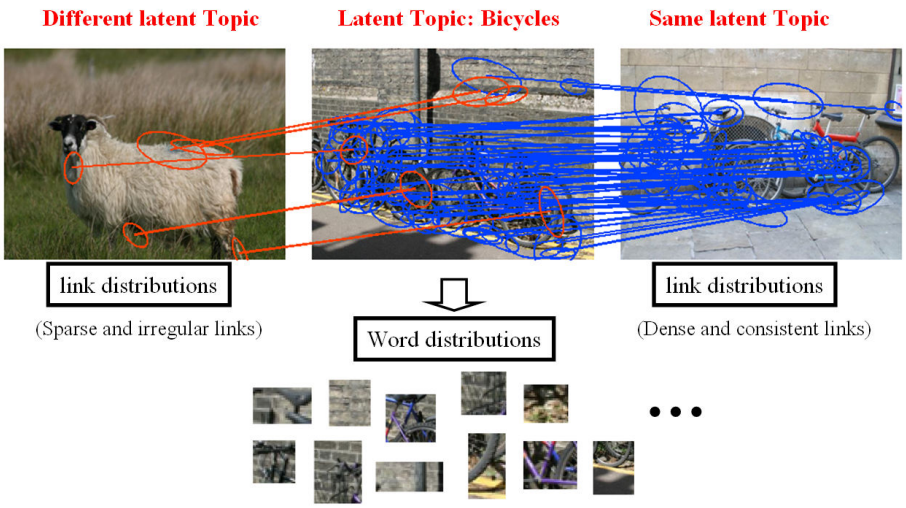


Figure 6: Intuition of the proposed approach. This figure only shows a small part of networks of the image set. No geometrically consistent image matcher is perfect, but it can discover reasonable relations between images in most cases. The latent topics are involved in the generations of not only visual words but also links between images.

correspondences when compared by a matching algorithm. We will call such images as "stongly linked" with each other. As a result, analyzing which images is strongly linked with which other image (the equivalent of documents referencing each other) provides useful information in addition to the statistics of visual words (the equivalent of statistics on the content of each document). In this paper, we propose an approach to implementing this analogy for category discovery, classification, and detection applications.

Fig. 6 illustrates the intuition of our approach. The topic models are based on the observation that samples of visual words are generated from a mixture model of latent topics. Assuming that we have an image matcher which is reasonable in most cases, the distributions of the links generated by the matcher are highly likely to be governed by the same latent topics as well. In other words, if two images share similar latent topics, then a matcher should generate a large number of consistent correspondences between features in the two images. Otherwise, the correspondences between the images may be sparse and irregular.

Recently, Kim et al [20] showed that link analysis techniques can be used for unsupervised inference of object category models. They represented visual information in the form of a large-scale complex network and formulated the unsupervised classification and localization in the modeling as the problem of finding *hubs* and *communities*. The *hubs* behave like important class-specific visual information and the *communities* map to a set of object categories. We extend their work to combine the statistical framework of topic content with link analysis. The basic differences of this work with

[20] are: 1) [20] is a purely link analysis method, but this work combines it with topic models which are popular in computer vision; 2) This work can be thought of a more principled approach based on a solid statistical foundation.

By combining the topic contents and link analysis, the proposed approach has three significant advantages over conventional topic models as follows.

(1) *More supporting evidence for better performance*: We augment the visual content of each image by taking advantage of the *relationships between other images*. Obviously, this information is valuable to describe the image, and the comparative experimental results clearly show significant improvement of the performance in unsupervised modeling, classification, and recognition (See Section 3.6.2.)

(2) *Easy plug-in of geometric information into topic models*: One persistent problem of the bag-of-words representation is that it is not straightforward to incorporate geometric information into the model, primarily because the computational complexity of the model increases exponentially with the number of parts [30]. We indirectly formulate the geometric information in the form of geometric consistency matches between a pair of images, which requires only a very simple modification of any generative models. We take advantage of the recent successes of a lot of off-the-shelf geometric consistency matching such as spectral matching of [22, 23], deformable matching of [2], or Pyramid matching [16]. Although there is no limitation on the choice of matching algorithms, our approach is based on the spectral matching of [22, 23].

(3) *Relaxation of the ambiguous definition of visual words*: In the visual domain, the definition of visual words is not intuitive. There are no natural boundaries, orders, and clear semantic meaning of visual words unlike in the text analysis domain from which the bag-of-words representation originated. Because of this difficulty, some under/over-clustering is unavoidable during the dictionary generation. For instance, two words A and B may be quantized into different clusters even though they are semantically similar. In our approach, this effect can be relaxed by similarity voting across the visual words in different images. Without quantization, the matching is based on the appearance affinity (*i.e.* L_2-norm of differences between feature vectors) in the context of geometric consistency.

Although a lot of generative models have been proposed, this work only considers two standard models in computer vision such as pLSA [18] and LDA [4] although, in principle, any generative models would do.

The link analysis techniques are very popular in other research areas such as text analysis, web applications, and bioinformatics. Therefore, the pLSA and LDA based models which combine topic contents and link analysis are already used in other research communities [8, 9] and the statistical models used here are inspired from this earlier work.

3.2 Related Work

A lot of generative topic models have been proposed for modeling and recognition of object categories [13, 36, 34, 7, 10]. Due to the vast amount of previous work in this area, here we limit ourselves to topic models based on bag-of-words with *spatial*

information. A general overview of this line of research can be found in the excellent Fei-Fei's tutorial [10].

The modeling of spatial information can be roughly classified into two classes of approaches. The first approach is to simply impose a spatial coherency constraint such as Spatial-LTM[7] and Spatial LDA[40]. The basic idea is that neighboring visual words are labeled by the same latent topics if they have similar appearance. Therefore, this approach is appropriate for a segmentation task since it can obtain consistent regions in the image and does not suffer from computational complexity.

The other approach is to explicitly represent the spatial relations between parts in the model such as [30] and [37]. Although this approach maybe be computationally expensive, it can achieve more discriminative modeling power. In order to reduce the complexity issue, [30] proposes a hierarchical model which consists of a part layer and a feature layer. In [37], the theoretical inference cost linearly increases with the number of parts, but they still used a small number of parts (i.e. less than ten) per an object and the training images were manually aligned for learning.

Our method for incorporating geometric information is different from those two classes of approaches. Rather than explicitly plugging geometric information into the model, we indirectly model the spatial evidence in the form of links with weights across the visual features by using the output of geometric consistent matching algorithms, which are *independent* of the model. By doing so, our approach can benefit from the performance of the matching algorithms with no increase of complexity with respect to the number of parts. Also, we do not need to rely on a definition of neighborhood between the parts, and, at the same time, our approach does not sacrifice the discriminative power of geometric information of whole parts.

More fundamentally, almost all of the previous approaches treat each document in isolation and focus on only the contents of each image for the inference of visual tasks. In other words, they do not explicitly express the *relationships between documents* into their models, even though they are quite informative in light of the success of search engines in WWW. This concept is helpful not only to model geometric information but also to obtain better performance.

3.3 Definition of Visual Words and Links

3.3.1 Links

For the representation of linking information, we adopt Kim et al [20]'s *Visual Similarity Network* (VSN), which explicitly expresses pairwise similarities across all n features in a set of training images $\mathbb{I}$. The VSN is a weighted graph, in which the nodes are the features extracted from all of the training images, and the edges link features that have been matched across images. The weights of the edges measure the degree of appearance and geometric similarities between features. Mathematically, the VSN $\mathbf{V}$ is represented by a $n \times n$ sparse matrix in which $\mathbf{V}_{ij}$ is a non-negative similarity value. If $\mathbf{V}_{ij} = 0$, there is no similarity links between node i and j. This representation is quite effective due to the fact that the VSN is *scale-free* [1]. In other words, given an image set only small number of features act as important *hubs* and large parts of them are of low importance. Therefore, we can quickly shrink our representation by

discarding trivial information such as backgrounds. The VSN is built by first matching all of the training images against each other by using a standard image matcher and by recording all the pairs of features that are matched by this procedure. The details of the construction of the VSN are described in [20].

3.3.2 Visual words

Following the standard approach to obtain visual words, we apply the Harris-Affine interest point detector [27] and the SIFT descriptor [26] to each image. In turn, the codebook of size W is created by K-means clustering to all the descriptors.

Once the visual words are computed for all M images, the $W \times M$ term-image co-occurrence matrix (**N**) is generated. Instead of using simple counts of words, we *weighted* each word before adding it to a term. Intuitively, if a feature has more inlinks (*i.e.* more matches), it receives more weight in **N**. Therefore, the weigth of a word w in image j, $\mathbf{N}_{wj}$, is not simply the number of occurences of w in image j, but instead: $\sum_j \mathbf{V}_{jw}$ (*i.e.* the sum of weights of inlinks to node w). Each column of **N** (*i.e.* a word histogram of each image) is normalized such that the sum of original word counts of an image is preserved.

3.3.3 Geometrically consistent links

Similar to the term-image matrix **N**, we define a $M \times M$ link-image co-occurrence matrix **A** as the other input to our algorithm. Since the VSN describes pairwise relationships at the feature level, we need to summarize them at the image level. Given the VSN **V**, $\mathbf{A}_{ab}$ (*i.e.*, the weights of the similarity inlinks from the image a to image b) is obtained as follows; $\mathbf{A}_{ab} = \sum_{i \in a} \sum_{j \in b} \mathbf{V}_{ij}$. It is a simple sum of all weights of links associated with the image a and b. In short, the link-image matrix **A** can be thought of an affinity matrix between all pairs of images.

In general, the term-image matrix **N** is quite sparse since only a small number of words in the dictionary occurs in an image. In order to make the link-image matrix **A** be similarly sparse, we limit the maximum number of nonzero elements in each column of **A** to $10 \log(M)$. In other words, we only consider k-neighbor neighbors for each image. (*i.e.* inlinks with top-k largest weights). k is set to $10 \log(M)$ by following the recommendation of [39, 20]. In practice, this heuristic dramatically decreases the computation time and it is known to be less sensitive parameter setting (as shown in [39]). Finally, each column of **A** is normalized to be $\sum_a \mathbf{A}_{ab} = \sum_a \mathbf{N}_{ab}$. This is intended for the two factors of content and links to have the same influences on the model.

Intuitively, our representation of links and their weights are *exchangeable* in the sense that their orders are not important, which is a necessary condition for using the LDA model [4].

3.4 The Proposed Generative Models

In this section, we describe the updated models which combine topic contents and links based on pLSA and LDA. The underlying assumption is that the image-specific topic

distribution not only generates visual words in the image but also governs geometrically consistent similarity matching between images as shown in Fig. 6. Intuitively, the images that share similar topics tend to share similar visual appearances and matching behaviors.

3.4.1 pLSA-based Model

Our pLSA-based model is based on the joint probabilistic model of [8], which combines term-based pLSA and link-based pHITS. As shown in Eq.7, pLSA and pHITS have similar mathematical forms. The only difference is that the pLSA models the distribution of terms w_n in an image d_j as $P(w_n|d_j)$, whereas the pHITS models the probability of in-links c_l (i.e., the citation to the image l) by an image d_j as $P(c_l|d_j)$. In our application, the citation to the image l by the image j means how well the image l is matched by the image j. These two equations share the same image-topic term $P(z_i|d_j)$, which is assumed to generate terms in an image $P(w_n|z_i)$ and links with other images $P(c_l|z_i)$, respectively.

$$P(w_n|d_j) = \sum_i P(w_n|z_i)P(z_i|d_j), \quad P(c_l|d_j) = \sum_i P(c_l|z_i)P(z_i|d_j) \quad (7)$$

The parameters we are interested in are $P(z_i|d_j)$, $P(w_n|z_i)$, $P(c_l|z_i)$. They can be obtained by EM iterations to maximize the log-likelihood function (Eq.8), which is a simple extension of that of pLSA [18] by introducing the relative weight α between the two contributions. In the following experiments, we use $\alpha = 0.5$, which means the two contributions are equally weighted.

$$\begin{aligned} \mathcal{L} = \sum_j \Bigg[& \alpha \sum_n \frac{\mathbf{N}_{nj}}{\sum_{n'} \mathbf{N}_{n'j}} \log \sum_i P(w_n|z_i)P(z_i|d_j) \\ & + (1-\alpha) \sum_l \frac{\mathbf{A}_{lj}}{\sum_{l'} \mathbf{A}_{l'j}} \log \sum_i P(c_l|z_i)P(z_i|d_j) \Bigg] \end{aligned} \quad (8)$$

Here, $\mathbf{N}_{nj}$ denotes how often a term w_n occurs in image d_j and $\mathbf{A}_{lj}$ indicates the frequencies of links c_l in image d_j.

3.4.2 LDA-based Model

We introduce the key equations that define our LDA-based model. They can be derived directly by following the procedures proposed by the original LDA paper [4]. We consistently follow the notations of [4] for readability. For the inference and parameter estimation, we use the variational approximation method [4]. Conceptually, our model resembles the mixed-membership models of [9] which are used for the field and subtopic classifications of papers in PNAS using *words* in abstract and *references* in bibliographies.

The joint distribution of $\{\theta, \mathbf{z}, \mathbf{w}, \mathbf{c}\}$ given the parameters $\{\lambda, \alpha, \beta\}$ and its variational distribution are given in Eq.9. These are direct extensions to the standard LDA

model by introducing the link-topic distribution which are almost identical to the word-topic distributions. As shown in Eq.9, the first term is the standard expression used in topic models based on distributions of words, the second term is the similar term obtained by using link distributions instead of word distributions. This parallel between the two terms can be carried over in the rest of the model, including, in particular in the update equations below.

$$
\begin{aligned}
P(\theta, \mathbf{z}, \mathbf{w}, \mathbf{c}|\lambda, \alpha, \beta) &= P(\theta|\lambda)\Big(\prod_{n=1}^{N} P(z_n|\theta)P(w_n|z_n, \alpha)\Big)\Big(\prod_{l=1}^{L} P(z_l|\theta)P(c_l|z_l, \beta)\Big) \\
q(\theta, \mathbf{z}|\gamma, \phi, \varphi) &= q(\theta|\gamma)\prod_{n=1}^{N} q(z_n|\phi_n)\prod_{l=1}^{L} q(z_l|\varphi_l) \qquad (9)
\end{aligned}
$$

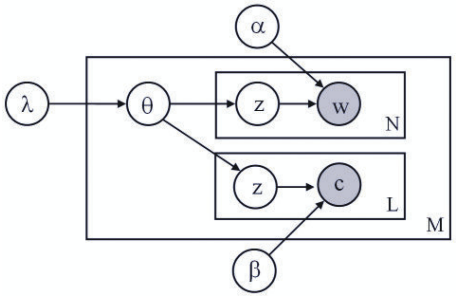
In Eq.9 the Dirichlet parameter γ, the multinomial parameters $(\phi_1, ...\phi_N)$ and $(\varphi_1, ...\varphi_L)$ are the variational parameters.

Eq.10-12 show the parameters to be estimated by variational EM iteration. These are iteratively updated until convergence. The detailed procedures are described in [4].

$$
\begin{aligned}
\phi_{ni} &\propto \alpha_{iw_n}\exp\{\Psi(\gamma_i)\}, \quad \varphi_{li} \propto \beta_{ic_l}\exp\{\Psi(\gamma_i)\}, \qquad (10) \\
\gamma_i &= \lambda_i + \sum_{n=1}^{N}\phi_{ni} + \sum_{l=1}^{L}\varphi_{li}, \qquad (11) \\
\alpha_{ij} &\propto \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}w_{dn}^{j}, \quad \beta_{ij} \propto \sum_{d=1}^{M}\sum_{l=1}^{L_d}\varphi_{dli}c_{dl}^{j}. \qquad (12)
\end{aligned}
$$

3.5 Modeling, Classification, and Detection

The main tasks are to 1) automatically generate object models in an unsupervised way, 2) classify the unseen images , and 3) localize the probable regions of the object in a novel image. Thereafter, we refer to each of them as the unsupervised modeling,



1. Sample $\theta \sim Dir(\theta|\lambda)$.
2. For each word $w_n, n \in \{1, ...N\}$
   (a) Sample $z_n \sim Mult(\theta)$
   (b) Sample $w_n \sim Mult(w|z_n, \alpha)$.
3. For each link $c_l, l \in \{1, ...L\}$
   (a) Sample $z_l \sim Mult(\theta)$
   (b) Sample $c_l \sim Mult(c|z_l, \beta)$.

Figure 7: The LDA based model [9]. This model is a simple extension of the LDA [4] by adding the link generation process which shares the same topic distributions θ with the word generation.

classification, and localization tasks, respectively. All related equations of pLSA and LDA-based models for theses tasks are summarized in Table 3.

In the unsupervised modeling task, a set of M unlabeled images is classified into object classes with a single piece of information (*i.e.* the number of object categories K). In other words, it assigns the most probable class membership to each unlabeled training image. In the pLSA model, the modeling is intuitive since we can easily obtain the distribution of latent topics, $P(z_i|d_j)$, for all images. In the LDA model, the modeling is obtained from one of variational parameters, γ_k, which is proportional to the posterior probability that each image contains topic k.

The classification task involves discovering the object classes of unseen images. Generally, it is done by running the same process by using the trained word-topic distributions: $P(w|z)$ and $P(c|z)$ in the pLSA model and λ, α, β. In the fold-in heuristics [36, 7], these values are fixed during the EM inference. However, we just use the learnt parameters for the initialization and allow the updates. Experimentally, the classification performances did not change whether they are updated or not, but for localization in the new image, the update may be helpful because it allows more opportunity to fit to new image data. Since the learnt parameters should be almost same to the parameters for the test image set, the EM iterations were quickly converged.

Our algorithm requires pairwise image matching to generate **A** as an input. Since it is inefficient to match each test image to all training images, we select 30 examplar images per class in the training set. As a measure of ranking of images with respect to each topic, we use $P(c_l|z_i)$ in the pLSA model and β_{ij} in the LDA model. They indicate how likely the image is to be cited (*i.e.* matched) from within the community of topic i . In other words, if a image has high value of $P(c_l|z_i)$, then it can be interpreted as an influential (*i.e.* authoritative) image with respect to its object category i.

Since we use interest region detectors as our unit visual information, the localization consists essentially in the selection of features which are most probable on the object in the image. Following [36], we select the feature by using $P(z_i|w_n, d_j)$ in the pLSA. In the LDA, the corresponding measure is ϕ_{ni}, which is the posterior probability that the word w_n in an image is generated from topic i. For each image, we select the features whose $P(z_i|w_n, d_j)$ are close enough to the maximum of the image. In other words, we choose the features with $P(z_i|w_n, d_j) \geq \rho \times \max_j P(z_i|w_n, d_j)$. In the experiments, $\rho = 0.8$ is used to be consistent with [20].

Table 3: Equations of pLSA and LDA-based models for ranking, unsupervised modeling, classification, and localization.

	pLSA-based model	LDA-based model
Ranking	$P(c_l\|z_i)$	β_{ij}
Modeling	$i^* = \arg\max_i P(z_i\|d_j)$	$i^* = \arg\max_i \gamma_i$
Classification	$i^* = \arg\max_i P(z_i\|d_{test})$	$i^* = \arg\max_i \gamma_{i,test}$
Localization	$P(z_i\|w_n, d_{test})$	ϕ_{ni}

3.6 Experiments

We designed two different experiments to evaluate the proposed methods. First, in order to justify the usefulness of link analysis, we performed comparison tests between the standard pLSA and LDA models and their linked versions for the unsupervised modeling task. Second, we present results of unsupervised modeling, ranking of training images, classification, and localization of unseen images with more challenging datasets such as MSRC [19] and PASCAL2005[1].

For better comparison tests, we use publicly available pLSA and LDA software[2].

3.6.1 Comparison tests

For comparison tests, we used one of the experimental setups of [20]. Specifically, we randomly selected 100 images per object for the five object classes of Caltech-101 - {*airplane*, *rear cars*, *faces*, *motorbikes*, *watches*}. The task is the unsupervised modeling, in which 500 training images are classified according to the categories with only the number of object classes (K=5) is given.

We compared the performances of three different versions of topic models - (1) Standard pLSA and LDA models, (2) pLSA and LDA models with matching weighted co-occurrence matrices (Section.3.3.2), and (3) Linked pLSA and LDA with weighted co-occurrence matrices. Fig.8.(a) represents variations of learned category accuracies of the three different versions with the 1000 codebook size. It clearly shows that the proposed approach (*i.e.* matching weighted counts of words and the combination between topic contents and links) leads to significant performance increase. Fig.8.(b) to Fig.8.(d) plots the accuracy of the three versions as the size of the codebook is varied through $\{500, 750, 1000, 1250, 1500\}$. However, the changes of codebook sizes showed little change in performance.

3.6.2 Results of modeling, Ranking, Classification, and Detection

We evaluate the proposed unsupervised modeling and recognition method using two different datasets, which are MSRC dataset {272 Bicycles, 505 Cars, 166 Doors, 190 Sheep, 165 Signs} and PASCAL05 {95 ETHZ motorbikes, 100 ETHZ cars, 168 TU-Graz person, 88 ETHZ Giraffes}. Since all images in the MSRC dataset are 640×480, they are resized to 320×240 for better computational speed. However, the PASCAL05 consists of diverse sizes of images and they are used without rescaling.

We ran two different experiments for each of two image sets by changing the number of object classes - (1)Three object case in PASCAL dataset: {Motorbike, Cars, Person}, (2)Four object case in PASCAL: {Giraffes, Motorbike, Cars, Person}, (3)Four object case in MSRC: {Bicycles, Cars, Doors, Sheep}, and (4)Five object case in MSRC: {Bicycles, Cars, Doors, Sheep, Signs}. We randomly selected the same numbers of images for training and test sets - 75 images for (1) and (2), 45 images for (3), and 40 images for (4). The number of images are decided according to the minimum

[1]The PASCAL dataset is available at http://www.pascal-network.org/challenges/ VOC/

[2]The pLSA code is available at http://people.csail.mit.edu/fergus/iccv2005/bagwords.html
The LDA code is at http://chasen.org/~daiti-m/dist/lda/

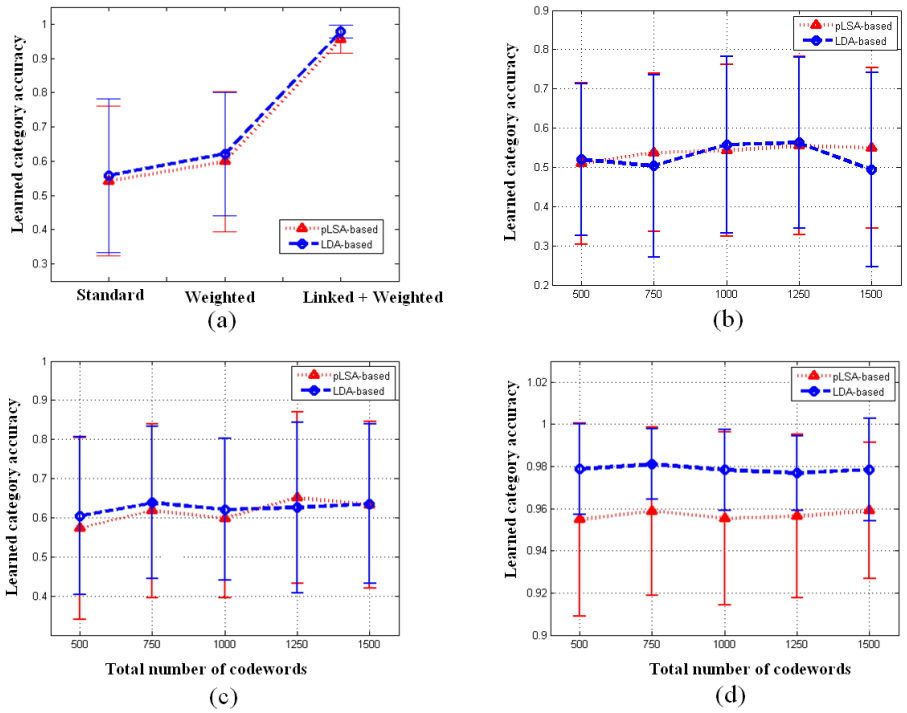


Figure 8: Comparison test results on the five objects of Caltech-101 dataset. The means and standard deviations of 10 runs for each are shown. (a) Performance comparison between three different versions with a codebook size of 1000. The accuracies of (pLSA, LDA) are (1) **Standard**: $54.2 \pm 21.8, 55.8 \pm 22.5$, (2) **Weighted**: $59.9 \pm 20.4, 62.1 \pm 18.1$, (3) **Linked**: $95.5 \pm 4.1, 97.8 \pm 1.9$. The LDA-based methods showed slightly outperformed the pLSA-based ones. (b), (c), (d) are accuracy variations of **Standard**, **Weighted**, and **Linked** versions according to the size of codebook, respectively.

size of object classes in the dataset. The models learned using the training images in an unsupervised way are used for classification and localization of test images. We iterated ten runs of experiments.

Please note that there have been very few previous work which performed experiments of unsupervised modeling, ranking, classification, and detection in a single framework. Most prior work evaluates their performance in separate experiments [36, 7]. Also, the MSRC and PASCAL datasets are challenging in the sense that they have not been much used for unsupervised modeling.

Unsupervised modeling: Table 4 and Table 5 represent the confusion matrices for the unsupervised modeling accuracies of the pLSA and LDA models. They represent how well the unlabeled training images are clustered according to their categories by

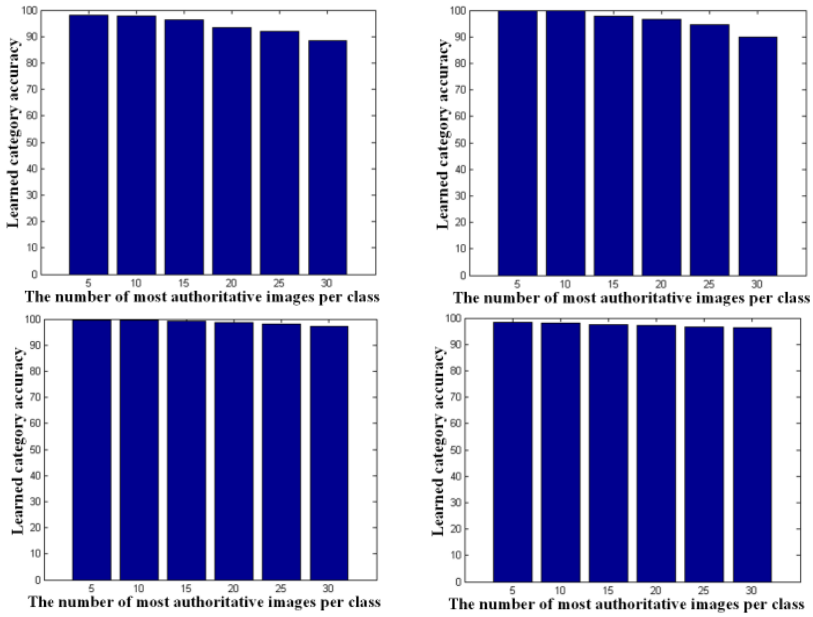


Figure 9: Accuracy of classification of top-k most authoritative images with respect to each topic. From left to right: pLSA in the 3 objects of the PASCAL, LDA in the 3 objects of the PASCAL, pLSA in the 4 objects of the MSRC, and LDA in the 4 objects of the MSRC.

measuring the agreement of topic discovery with ground truth labels. Experimental results showed that our performance is competitive since we achieved 93.85% (3 objects) and 85.44% (4 objects) for the PASCAL dataset and 96.33% (4 objects) and 90.29% (5 objects) for the MSRC dataset. Also, we observed that the LDA based model slightly outperforms the pLSA based model.

Ranking of training images: As introduced in Table 3, we can rank the training images with respect to each topic by using $P(c_l|z_i)$ in the pLSA and β_{ij} in the LDA based model. In practice, this ranking is quite useful since in many cases we need to find some representative images for each object category. For example, for each topic i, we can sort $P(c_l|z_i)$ or β_{ij} of all images and select top-k images with highest values as prototype images.

Fig.9 shows the variations of accuracy of agreement of the top-k images per an object class with ground truth category labels by varying the k from 5 to 30. Obviously, as k increases, the ratios drop slightly. One interesting observation is that for $k = 30$ the accuracy is slightly worse than the unsupervised modeling accuracy reported in Table 4. For example, for the 30 image case of pLSA based model in the 3 objects of the PASCAL dataset (*i.e.* the right most bar in the Fig.9.(a)), the accuracy is 88.4%. However, the unsupervised modeling ratio for all 45 images per class is 93.11% (See

the Table 4.(a)). This discrepancy occurs because $P(c_l|z_i)$ and β_{ij} are purely link analysis terms whereas the results of Table 4 are contributed by the combination of topic contents and link analysis. Therefore, this can be interpreted as a strong evidence of superiority of the combination over relying on a single aspect.

Classification of unseen images: For each object class, we selected the top 30 training images as exemplars as described in the previous section. In addition to words/links-topic parameters learned during the training, we can also take advantage of the matches between exemplar images, which makes our method more discriminative. Table 6 and Table 7 show the results of classifying test images. Even though this task is more challenging than the unsupervised modeling step in the sense that we do not have full comparison between images and can be affected by modeling errors, we only observed a slow degradation in performance.

Localization of unseen images: Fig.10 shows some typical localization examples of MSRC and PASCAL05 dataset. For each image, we selected the features which satisfy the equation $P(z_i|w_n, d_j) \geq 0.8 \times \max_j P(z_i|w_n, d_j)$. (For LDA model, ϕ_{ni} is used instead.) The topic i is assigned to each word by $i^* = \arg\max_i P(z_i|w_n, d_j)$. We draw the features by different colors according to the assigned topic. As shown in the pictures, the majority of topics assigned to high confident features are coincident with the topic of the image.

As discussed in Section.3.3.2, our method is based on the counts of words *weighted* by an image matcher with geometric consistency. Contrary to the standard bag-of-words representation, we can take advantage of the geometric information inferred by the matcher for the localization. Consequently, the proposed formulation of geometric information as the *similarity links* circumvents the problem of plugging a geometric modeling into topic models but also eludes any additional increase of complexity of models.

4 Conclusion

This work proposes novel unsupervised modeling and recognition approaches for object categories by applying link analysis techniques to a large-scale network of visual information. Experimental results clearly showed that the proposed methods achieved better performance over the existing algorithms for several different image datasets.

In the section 2, we proposed an approach for extracting object models from unlabeled training data. Unlike prior methods, this approach extracts categories and features within the categories by analyzing a visual similarity graph, without clustering or statistical modeling. This representation provides a global view of the interactions between all features which allows us to use different types of information - ranked importance of each feature with respect to an image or an object category and structural similarity between any pair of nodes. This approach yields better results on the Caltech-101 examples used in prior work in unsupervised modeling, with a larger number of classes. We also showed competitive results for the TUD/ETHZ dataset.

In the section 3, we introduced the use of the *link analysis* idea to generative topic models. The approach is based on the observation that the relationships between visual information across images are as valuable as the visual contents in the images. Inspired by the statistical frameworks developed for web applications, experimental results showed that the combination of contents and links is a promising approach for computer vision problems, too.

We believe much remains to be done for this approach to be used in other visual tasks. In particular, even though we have a rich representation which describes all interactions between low-level visual information, in the section 2, we can certainly improve the way we integrate it in Eq.5 and Eq.6, which are first-order sums of two types of information estimated from link analysis. Similarly, in the section 3, the links are defined as *relations between images*, which is largely due to the fact that the framework used here is developed for the analysis on the documents and their references to others. However, more detailed investigation on the feature level interactions may be necessary since an image may contain semantically different objects. For example, a scene may consist of buses, humans, sky, tree, and buildings, which are not necessarily related. Second, we would like to explore more sophisticated way to generate the network for visual tasks. Here, we are mainly based on an image matcher with geometric consistency but other types of relational information can be used for the constructions of similarity links. Finally, the sparseness of the data observed in these experiments, together with the fact that the link analysis tools are routinely used in far larger applications suggest that it is possible to scale the algorithms up to a much large of classes.

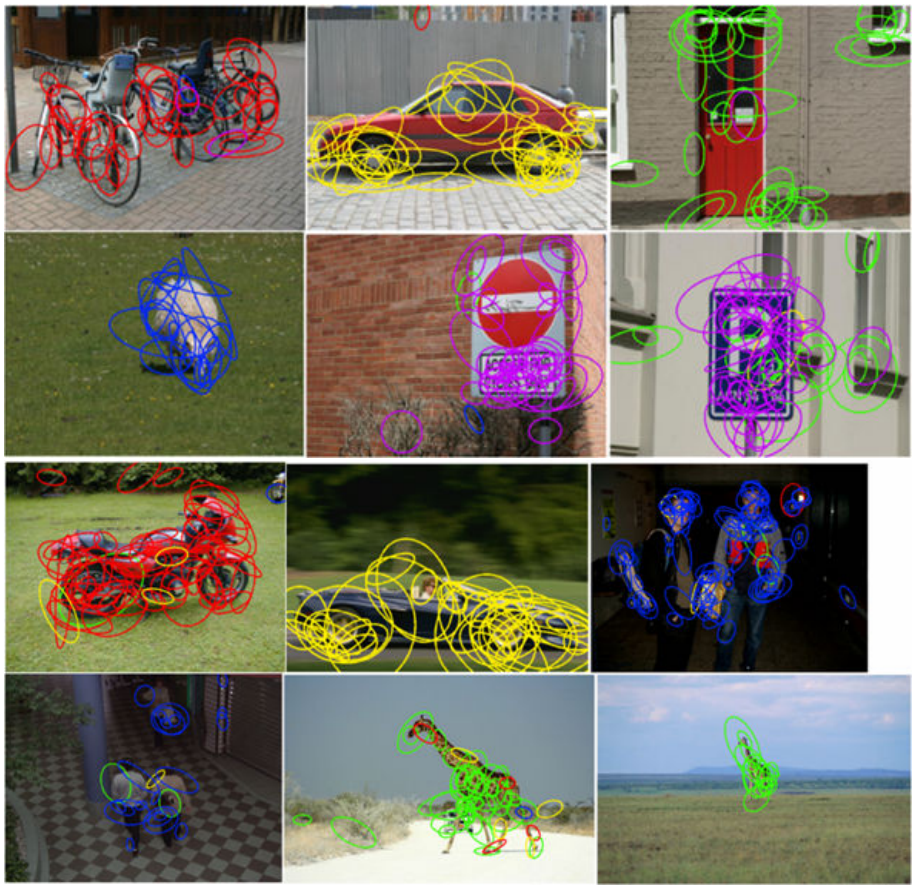


Figure 10: Localization results. The first and second rows are examples of the MSRC dataset, and the third and fourth rows are for the PASCAL05 dataset. The colors of features are assigned according to the topics. (1) For MSRC dataset, red, yellow, green, blue, and purple colors are used for *bicycles*, *cars*, *doors*, *sheep*, and *signs* topics, respectively. (2) For PASCAL05, red, yellow, blue, and green are assigned to *motorbikes*, *cars*, *people*, and *giraffes*. (These figures are best viewed in color.)

Table 4: Confusion tables for unsupervised modeling of PASCAL data set using pLSA and LDA based models. The means and standard deviation values of 10 runs for each are shown. The modeling accuracies are 1) pLSA(3Obj): **93.11**%, 2) LDA(3Obj): **93.85**%, 3) pLSA(4Obj): **83.13**%, and 4) LDA(4Obj): **85.44**% on average. (PG: Giraffes, PM: Motorbikes, PC: Cars, PP: Persons of PASCAL05 dataset)

	PM	PC	PP
PM	**92.0**±3.9	**2.7**±2.7	**5.3**±3.0
PC	**0.2**±0.7	**99.8**±0.7	**0**
PP	**5.8**±3.0	**6.7**±3.8	**87.6**±5.2

	PM	PC	PP
PM	**86.2**±4.7	**0.2**±0.7	**4.6**±2.7
PC	**0**	**98.0**±1.9	**2.0**±1.9
PP	**2.2**±2.1	**0.4**±0.9	**97.3**±2.5

	PG	PM	PC	PP
PG	**71.3**±6.4	**13.7**±5.9	**2.5**±1.7	**12.5**±3.3
PM	**1.3**±1.8	**85.5**±5.0	**6.7**±6.8	**6.5**±5.3
PC	**0.0**	**0.0**	**100**	**0.0**
PP	**10.7**±6.5	**5.5**±4.4	**8.0**±5.9	**75.8**±8.2

	PG	PM	PC	PP
PG	**76.5**±5.8	**7.5**±5.5	**0.3**±0.8	**15.7**±6.1
PM	**1.8**±2.1	**86.0**±5.0	**1.8**±2.6	**10.4**±6.1
PC	**0.0**	**0.5**±1.1	**97.3**±2.5	**2.2**±2.2
PP	**13.0**±6.3	**3.2**±3.1	**1.8**±3.1	**82.0**±4.4

Table 5: Confusion tables for unsupervised modeling of MSRC data set using pLSA and LDA based models. The modeling accuracies are 1) pLSA(4obj): **91.03**%, 2) LDA(4obj): **96.33**%, 3) pLSA(5obj): **85.55**% and 4) LDA(5obj): **90.29**% on average. (MB: Bicycles, MC: Cars, MD: Doors, MS: Sheep, MG: Signs of MSRC dataset)

	MB	MC	MD	MS
MB	**74.1**±4.9	**1.9**±2.2	**3.9**±1.8	**20.1**±5.0
MC	**0.0**	**92.8**±11	**0.3**±0.6	**6.9**±10.3
MD	**0.0**	**0.0**	**98.0**±1.6	**2.0**±1.6
MS	**0.0**	**0.3**±0.8	**0.5**±0.7	**99.2**±1.3

	MB	MC	MD	MS
MB	**94.7**±3.1	**0.8**±1.3	**1.9**±1.4	**2.7**±2.7
MC	**0.4**±0.6	**95.6**±9.7	**0.1**±0.4	**3.9**±8.8
MD	**1.3**±1.4	**0.5**±0.7	**96.0**±1.7	**2.1**±2.0
MS	**0.3**±0.6	**0.5**±0.9	**0.1**±0.4	**99.1**±1.4

	MB	MC	MD	MS	MG
MB	**77.8**±3.7	**1.7**±1.9	**2.5**±1.9	**16.8**±4.7	**1.2**±1.0
MC	**0.0**	**74.4**±13.8	**0.7**±0.7	**10.8**±12.0	**14.1**±13.9
MD	**0.0**	**0.0**	**95.1**±1.7	**1.7**±1.1	**3.2**±1.6
MS	**0.0**	**0.1**±0.4	**1.6**±2.2	**98.3**±2.1	**0.0**
MG	**0.4**±0.9	**1.3**±1.4	**9.2**±2.6	**6.8**±2.4	**82.3**±4.0

	MB	MC	MD	MS	MG
MB	**96.1**±2.3	**0.3**±0.6	**1.2**±1.3	**2.4**±2.0	**0.0**
MC	**0.1**±0.4	**82.0**±18.1	**0.1**±0.4	**11.1**±15.3	**6.7**±12.1
MD	**1.2**±1.2	**0.1**±0.4	**94.5**±1.9	**1.9**±1.6	**2.3**±1.7
MS	**0.4**±0.6	**0.0**	**0.3**±0.6	**99.2**±1.1	**0.1**±0.4
MG	**3.4**±1.6	**2.8**±3.4	**8.5**±3.2	**5.7**±3.3	**79.6**±6.8

Table 6: Confusion tables for classification of test images of PASCAL data set using pLSA and LDA based models. The means and standard deviation values of 10 runs for each are shown. The classification accuracies are 1) pLSA(3Obj): **90.81**%, 2) LDA(3Obj): **89.19**%, 3) pLSA(4Obj): **83.63**% and 2) LDA(4Obj): **80.5**% on average.

	PM	PC	PP
PM	**89.1**±3.5	**3.6**±2.4	**7.3**±3.0
PC	**0.7**±1.1	**98.2**±1.8	**1.1**±1.6
PP	**6.9**±2.9	**8.0**±3.7	**85.1**±3.2

	PM	PC	PP
PM	**86.0**±5.0	**7.8**±3.8	**6.2**±3.3
PC	**0.2**±0.7	**99.3**±1.5	**0.4**±0.9
PP	**6.9**±2.7	**10.9**±5.4	**82.2**±4.7

	PG	PM	PC	PP
PG	**72.8**±3.2	**11.5**±4.7	**4.2**±2.1	**11.5**±4.9
PM	**1.7**±2.1	**86.8**±5.8	**5.5**±3.1	**6.0**±4.3
PC	**0.0**	**2.0**±3.1	**95.8**±3.1	**2.2**±2.5
PP	**11.0**±3.4	**3.5**±2.1	**6.2**±6.5	**79.3**±5.9

	PG	PM	PC	PP
PG	**66.5**±6.3	**12.2**±7.7	**10.3**±5.1	**11.0**±6.7
PM	**0.5**±1.1	**80.0**±9.1	**12.7**±6.3	**6.8**±4.4
PC	**0.0**	**0.5**±1.1	**97.8**±2.8	**1.7**±2.6
PP	**9.2**±3.7	**5.3**±3.4	**7.7**±8.4	**77.8**±8.1

Table 7: Confusion tables for classification of test images of MSRC data set using pLSA and LDA based models. The modeling accuracies are 1) pLSA(4Obj): **89.03**% and 2) LDA(4Obj): **90.17**%, 3) pLSA(5Obj): **82.19**%, and 4) LDA(5Obj): **82.16**% on average.

	MB	MC	MD	MS
MB	**73.9**±1.3	**2.8**±1.3	**4.5**±2.4	**18.8**±4.8
MC	**0.0**	**87.3**±9.0	**2.4**±1.2	**10.3**±9.3
MD	**0.0**	**0.3**±0.6	**97.2**±2.2	**2.5**±2.0
MS	**0.0**	**0.8**±1.7	**1.5**±1.2	**97.7**±2.0

	MB	MC	MD	MS
MB	**83.3**±4.5	**2.9**±1.9	**5.5**±2.5	**8.3**±4.2
MC	**0.0**	**84.3**±15	**1.7**±1.5	**14.0**±15.6
MD	**0.0**	**0.5**±0.7	**97.3**±1.9	**2.1**±1.7
MS	**0.8**±0.9	**1.2**±1.3	**2.3**±2.3	**95.7**±2.9

	MB	MC	MD	MS	MG
MB	**76.5**±6.1	**3.5**±3.0	**5.5**±3.5	**12.8**±5.0	**1.7**±0.9
MC	**0.0**	**69.3**±11.4	**1.1**±1.2	**22.7**±14.4	**6.9**±6.6
MD	**0.0**	**0.5**±0.7	**88.2**±3.7	**3.7**±1.1	**7.6**±2.7
MS	**0.5**±0.7	**0.1**±0.4	**1.2**±1.0	**97.9**±2.0	**0.3**±0.6
MG	**0.7**±0.7	**2.1**±0.9	**7.5**±1.3	**10.6**±4.1	**79.1**±5.1

	MB	MC	MD	MS	MG
MB	**79.1**±8.2	**4.4**±3.0	**5.5**±3.0	**10.1**±5.4	**0.9**±0.9
MC	**0.0**	**69.1**±10.0	**1.5**±1.3	**25.7**±12.3	**3.7**±4.1
MD	**0.0**	**0.4**±0.6	**91.2**±3.3	**3.5**±1.4	**4.9**±2.4
MS	**0.7**±0.9	**0.7**±1.1	**1.3**±2.4	**96.9**±3.3	**0.4**±0.6
MG	**0.7**±0.9	**2.9**±2.5	**11.5**±3.2	**10.4**±3.4	**74.5**±5.1

References

[1] A.-L. Barabási. Scale-free networks. *Scientific American*, 288:60–69, 2003.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence, 2005. CVPR.

[3] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2(1):73–120, 2005.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. V. Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, 46(4):647–666, 2004.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine, 1998. WWW.

[7] L. Cao and L. Fei-Fei. Spatial coherent latent topic model for concurrent object segmentation and classification, 2007. ICCV.

[8] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity, 2001. NIPS.

[9] E. Erosheva, S. Fienberg, , and J. Lafferty. Mixed-membership models of scientific publications. *PNAS*, 101(1):220–5227, 2004.

[10] L. Fei-Fei. Bag of words models: Recognizing and learning object categories, 2007. CVPR Short Courses.

[11] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006.

[12] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories, 2007. Short Courses for CVPR.

[13] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories, 2005. CVPR.

[14] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search, 2005. ICCV.

[15] M. Fritz1 and B. Schiele. Towards unsupervised discovery of visual categories, 2006. DAGM-Symposium.

[16] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features, 2005. ICCV.

[17] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features, 2006. CVPR.

[18] T. Hofmann. Probabilistic latent semantic analysis, 1999. NIPS.

[19] A. C. John Winn and T. Minka. Object categorization by learned universal visual dictionary, 2005. ICCV.

[20] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques, 2008. CVPR (To appear).

[21] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[22] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints, 2005. ICCV.

[23] M. Leordeanu and M. Hebert. Efficient map approximation for dense energy functions, 2006. ICML.

[24] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features, 2007. CVPR.

[25] H. Liu, E. Milios, and J. Janssen. Probabilistic models for focused web crawling, 2004. WIDM.

[26] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[27] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[28] R. Nallapati and W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence in blogs, 2008. ICWSM.

[29] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[30] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification, 2007. CVPR.

[31] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery, 2004. KDD.

[32] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations, 2007.

[33] M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank, 2002. NIPS.

[34] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections, 2006. CVPR.

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[36] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images image features, 2005. ICCV.

[37] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts, 2005. ICCV.

[38] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images, 2006. CVPR.

[39] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[40] X. Wang and E. Grimson. Spatial latent dirichlet allocation, 2007. NIPS.