CO-OCCURRENCE MODELS FOR IMAGE ANNOTATION AND RETRIEVAL

Nikhil Garg

Idiap-RR-22-2009

AUGUST 2009

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Co-occurrence Models for Image Annotation and Retrieval

Nikhil Garg
Ecole Polytechnique Fédérale de Lausanne, Switzerland
nikhil.garg@epfl.ch

Master Thesis
August 2009

Submitted to:
Computer Science Section
The School of Computer & Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Supervisors:
Prof. Hervé Bourlard
Dr. Daniel Gatica-Perez

Abstract

We present two models for content-based automatic image annotation and retrieval in web image repositories, based on the co-occurrence of tags and visual features in the images. In particular, we show how additional measures can be taken to address the noisy and limited tagging problems, in datasets such as Flickr, to improve performance. As in many state-of-the-art works, an image is represented as a bag of visual terms computed using edge and color information. The cooccurrence information of visual terms and tags is used to create models for image annotation and retrieval. The first model begins with a naive Bayes approach and then improves upon it by using image pairs as single documents to significantly reduce the noise and increase annotation performance. The second method models the visual terms and tags as a graph, and uses query expansion techniques to improve the retrieval performance. We evaluate our methods on the commonly used 150 concept Corel dataset, and a much harder 2000 concept Flickr dataset.

Acknowledgments

This work was done as a part of my Master thesis at Idiap Research Institute, Martigny, Switzerland. I would like to acknowledge the support of the Swiss National Science Foundation (SNSF) through the National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

I would like to thank my supervisors, Prof. Hervé Bourlard and Dr. Daniel Gatica-Perez for their guidance and support. I would also like to thank Florent Monay and Radu Negoescu for providing data and technical support.

Contents

Chapter 1

Introduction

With the increasing availability of large image collections on the web, content-based automatic image annotation and retrieval have gained significant interest to enable indexing and retrieval of unannotated or poorly annotated images [Barnard et al., 2003, Blei and Jordan, 2003, Li and Wang, 2003, Monay and Gatica-Perez, 2007]. The annotation problem is defined as follows: given an image, produce a ranked list of tags that describe the content of the image. Retrieval is the reverse problem, defined as follows: given a set of query tags, produce a ranked list of images whose content relate to the query tags. Content-based retrieval would benefit not only image search engines such as *Google Image Search*[1] and *Yahoo Image Search*[2], but also photo sharing websites such as *Flickr*[3] and *Picasa*[4]. In particular, Flickr allows users to write *descriptions* and attach *tags* to their photos. These features are used to enable image search on the site. Content-based automatic annotation may be used to suggest tags to users, and retrieval may be used to expand the search beyond the user generated annotations. Large scale image collections such as Flickr present a special challenge for these tasks due to the vast variety of content in these images that results in a huge number of "visual concepts", and the often poor or limited annotation done by users that results in "noisy" labels for supervised learning methods. In this work, we propose novel algorithms for image annotation and retrieval tasks that aim to address these challenges in noisy datasets. Our first method describes an improvement over a basic naive Bayes algorithm by considering pairs of images as single documents. The hypothesis is that co-occurrence at the image pair level helps reducing the ambiguity about the relation of the tags with the actual image content. This method reduces the annotation noise by using only the common tags and visual features in image pairs to construct an improved naive Bayes model which gives a better annotation performance. The second method is used to improve the retrieval performance. It uses a

[1] http://images.google.com/

[2] http://images.search.yahoo.com/

[3] http://www.flickr.com/

[4] http://picasaweb.google.com/

graph-based approach to first perform a query expansion and then uses the expanded query to weight the *visual terms*, which are then used further to rank the images. Here, the hypothesis is that a single tag is often insufficient to generate a relevance score for visual features because of the noisy training and high diversity in the image content.

A wide variety of datasets have been used in the research community for image analysis experiments. The Corel image collection is a publicly available and widely used dataset that has images with carefully done manual annotations. To facilitate comparison among the different approaches, we use data from both the Corel and Flickr collections. The main contributions of this work are the exploration of simple co-occurrence based algorithms that include measures to address the noisy and limited annotation problem, and an objective evaluation on Corel and Flickr data.

The rest of the report is organized as follows: Chapter 2 gives an overview of related work. Chapter 3 describes the image representation that we use in this work. Chapter 4 details the proposed algorithms. Chapter 5 describes the datasets used, experiments and results. We conclude in Chapter 6 and discuss some future directions for research.

Chapter 2

Related Work

A wide range of image analysis and content matching methods have been used in image annotation and retrieval research. The methods usually differ in the kind of visual features used, the modeled relationship between visual features and tags, and the kind of annotations and datasets used. Typically, the algorithms associate the tags with either the whole image or a specific region/object in the image. Using the former approach, in [Mori et al., 1999], an image is divided into a fixed grid and visual feature vectors from each block are quantized into a finite set of visual terms (visterms). All visterms of an image are associated with all the tags, and aggregating this information from all the images, an empirical distribution of a tag given a visterm is calculated. A new image is annotated by calculating the average likelihood of a tag given the visterms of the image. In contrast to this approach, a region naming approach is adopted in [Duygulu et al., 2002] by first segmenting the image into regions using the normalized cuts segmentation algorithm [Shi and Malik, 2000]. These regions are then classified into region types using a variety of visual features. A mapping between region types and keywords is learned using an EM approach. This model assumes a one-to-one correspondence between image regions and tags. An improvement over this model is suggested in [Jeon et al., 2003] by applying a cross-media relevance model for image annotation. This model also segments image into regions but does not assume a one-to-one correspondence between regions and tags. The conditional probability of a tag given an image is computed from the training data empirically. A new image is annotated by computing the likelihood of potential tags and image regions using the learned probabilities. Corr-LDA [Blei and Jordan, 2003] uses a region naming approach by first segmenting the image into regions using normalized cuts segmentation algorithm [Shi and Malik, 2000]. Next, Latent Dirichlet Allocation (LDA) [Blei et al., 2003] is used to build a combined generative model for regions and tags. For each tag, one of the regions is selected and the corresponding tag is drawn conditioned on the latent topic that generated the region. The latent topics in this case model the correspondence between visual features and tags. Also using a latent topic approach, the work in [Monay and Gatica-Perez, 2007] first constructs a bag-of-visual terms using a variety of visual features. The bag-of-visual terms and tags are both mapped to a common latent semantic space using Probabilistic Latent Semantic Analysis (PLSA) [Hofmann, 1999]. This approach associates the whole image to all the tags rather than a region naming approach. PLSA is also used in [Sivic et al., 2005] to derive latent topics for visual features but those topics are used as image categories. An image d_j is then classified as containing object k according to the maximum of $P(z_k|d_j)$ over k, where $P(z_k|d_j)$ represents the probability of latent topic z_k given the document d_j as given by the PLSA model. A diverse density multiple instance learning approach is demonstrated in [Yang and Lozano-Perez, 2000] by first dividing the image into several overlapping regions and constructing a feature vector from each. The training process then determines which features vectors in an image best represent the user's concept and which dimensions of the feature vectors are important. The work in [Li and Wang, 2003] builds a 2-D Multiresolution Hidden Markov Model (2D MHMM) [Li et al., 2000] for each image category that clusters the visual feature vectors at multiple resolutions and models spatial relations between the clusters. A new image is annotated by computing its likelihood of being generated by a category, and then tags are selected from the highest likelihood category. The work in [Hardoon et al., 2006] uses Kernel Canonical Correlation Analysis (KCCA) [Lai and Fyfe, 2000] to learn a mapping from image descriptors to tags. A graph based approach is adopted in [Pan et al., 2004] that models the visual features and tags as nodes in a graph and discovers correlations between visual and tag nodes via random walks with restarts. Table 2.1 gives an overview of different methods.

While many advanced models have been proposed, most of the existing research has used reasonably well annotated datasets such as Corel. Limited vocabulary and "simple" images in Corel also help in developing more efficient models. Annotation noise in real world datasets such as Flickr presents additional challenges that we aim to address in this work. Flickr datasets have been used more recently in numerous other studies. Tagging patterns in Flickr images are used in [Dubinko et al., 2007, Rattenbury et al., 2007] to extract events over time. Tags and location information along with image analysis is used in [Kennedy et al., 2007] to retrieve images of landmarks from Flickr. The work in [Wu et al., 2008] constructs a similarity network of tags based on the visual correlation between regions in the image. Tag recommendation systems [Garg and Weber, 2008, Sigurbjörnsson and van Zwol, 2008] have also been proposed that suggest related tags based on some query tags, using the co-occurrence patterns of tags in Flickr. Content based image annotation can be used either to enhance such tag recommendation systems or as an alternative when no query tags are present.

Paper	**Image representation**	**Visual-Tag association**	**Dataset(s)**
[Mori et al., 1999]	Divide image into a uniform grid. Features: color (RGB), edge (Sobel).	Probability distribution of tag given visual term.	Mypaedia (9681 images)
[Yang and Lozano-Perez, 2000]	Divide image into overlapping regions. Features: grayscale pixel values.	Multiple instance learning to weight visual features for a tag.	Corel (500 images), another downloaded collection (228 images)
[Duygulu et al., 2002]	Segment image into regions. Features: color, orientation energy, size, position, etc.	EM algorithm to learn mapping between region types and tags.	Corel (5000 images)
[Jeon et al., 2003]	Segment image into regions. Features: color, orientation energy, size, position, etc.	Probability distribution of tag given a set of visual terms.	Corel (5000 images)
[Blei and Jordan, 2003]	Segment image into regions. Features: color, texture, size, shape, position, etc.	LDA to map visual terms and tags to a common latent space.	Corel (7000 images)
[Li and Wang, 2003]	Divide image into uniform grid at multiple resolutions. Features: color (LUV), texture (Daubechies-4 wavelet transform).	2D MHMM to model each image category, mapping between 2D MHMM and tags.	Corel (28600 images)
[Pan et al., 2004]	Segment image into regions Features: color (RGB), texture, position, shape.	Visual features and tags as nodes in a graph. Correlation discovery via random walks with restarts.	Corel (16000 images)
[Sivic et al., 2005]	Detect interest points. Features: edge (SIFT)	PLSA to map visual features to object categories.	Caltech 101 (4090 images), MIT image dataset (2873 images)
[Hardoon et al., 2006]	Detect interest points. Features: edge (SIFT)	Kernel Canonical Correlation Analysis.	University of Washington Ground Truth Image Database (697 images)
[Monay and Gatica-Perez, 2007]	Detect interest points. Features: color (HSV), edge (SIFT)	PLSA to map visual terms and tags to a common latent space.	Corel (16000 images)
[Kennedy et al., 2007]	Detect interest points. Features: color (LUV), texture (Gabor), edge (SIFT)	Both tags and visual features used to retrieve images of landmarks.	Flickr (110000 images)
This work	Detect interest points. Features: color (HSV), edge (SIFT)	Probability distribution of visual terms given tags.	Corel (16000 images) , Flickr (65000 images)

Table 2.1: An overview of related work

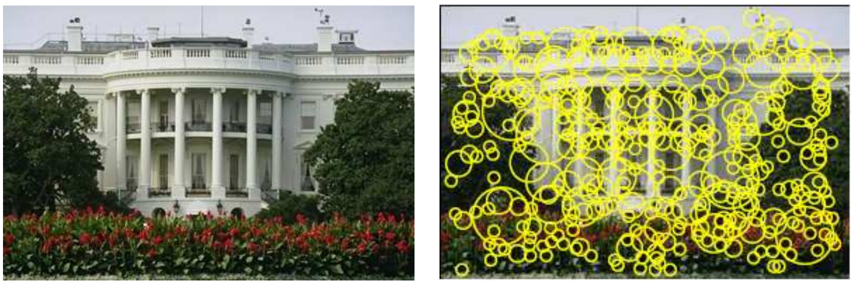
Chapter 3

Image Representation

We use the same image representation as in [Monay and Gatica-Perez, 2007], which we briefly describe here. A vocabulary of visual features or visterms is created from the training images as follows. Given a training image, Difference of Gaussians (DOG) point detector [Lowe, 2004] is used to identify regions where a maximum or minimum of intensity occurs in the image, and it is invariant to translation, scale, rotation and constant illumination variations. Figure 3.1 shows an example image and interest regions identified by the DOG point detector. Edge and color features are then computed from each interest region. For edge features, Scale Invariant Feature Transform (SIFT) descriptors [Lowe, 2004] are used to compute a histograms of edge directions over different parts of the interest region. Eight edge orientation directions and a grid size of 4x4 are used to form a feature vector of size 128. Orientation invariance is achieved by estimating the dominant orientation of the local image patch using the orientation histogram of the keypoint region. All direction computations in the elaboration of the SIFT feature vector are then done with respect to this dominant orientation. Figure 3.2a shows an illustration of the SIFT grid corresponding to a single interest region. This grid represents a single SIFT feature vector of size 128. Figure 3.2b shows the histogram of SIFT feature vectors obtained from all the interest regions in the image.

For color features, we use the Hue-Saturation-Value (HSV) color space. An image is divided into a uniform grid and a 2D Hue-Saturation (HS) histogram is computed using the color distribution from the resulting regions. Brightness values are discarded for illumination invariance. The HS histogram is used as a color feature vector.

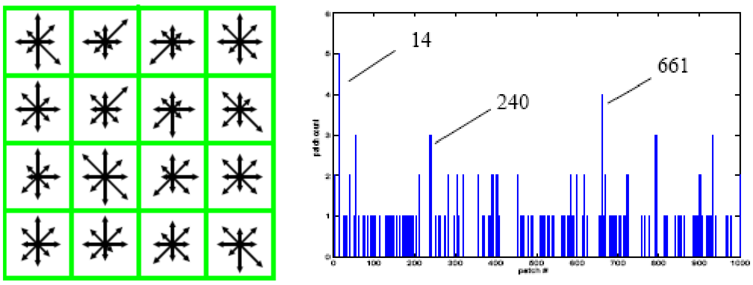
Both the edge and the color feature vectors aggregated from all the training images are then quantized into 1000 centroids each using the K-means clustering algorithm [Lloyd, 1982]. This gives us a discrete set of 1000 edge features and 1000 color features that we call visterm vocabulary of size 2000. Next, the edge and color feature vectors of each image are mapped to the corresponding closest feature vector in the visterm vocabulary. This gives us an image representation in the form of a bag of visterms. Both training and test images are represented by bags of visterms using the same visterm vocabulary.



(a) Given Image

(b) Interest Points

Figure 3.1: (a) given image, (b) interest regions obtained by applying Difference of Gaussians point detector. [Figures taken from [Monay et al., 2009]]



(a) SIFT Grid

(b) Visterm Histogram

Figure 3.2: (a) SIFT grid for an interest region. Each square represents a bin in the grid. For each bin, a histogram of edge directions is computed. The histogram is shown with the help of arrows of different sizes and directions. (b) Visterm histogram of the edge (SIFT) features computed from all the interest points in the image. [Figures taken from [Monay et al., 2009]]

Chapter 4

Co-occurrence Models

We propose two models for the annotation and retrieval tasks. Both models are based on co-occurrence of visterms and tags in the images, though the co-occurrence information is used in a different fashion. The first model is an extension of a simple naive Bayes approach, while the second model is a graph based approach.

4.1 Naive Bayes model

We first describe a basic naive Bayes model and then make improvements to address the noisy tagging problem in Flickr.

4.1.1 Basic Naive Bayes model

A simple naive Bayes model can be trained by calculating conditional probabilities $P(v_i|t_j)$ for all combinations of visterm v_i and tag t_j in the corpus,

$$P(v_i|t_j) = \frac{n_I(v_i, t_j)}{n_I(t_j)},$$

where $n_I(v_i, t_j)$ denotes the number of training images with visterm v_i and tag t_j, and $n_I(t_j)$ denotes the number of training images with tag t_j.

For image annotation, given a new image I, we first calculate its set of visterms $\{v_1, v_2, \ldots, v_k\}$. Annotation can be modeled as a classification problem by treating visterms as inputs and each of the tags in the vocabulary as a separate class. We compute the annotation score for a tag t_j as $S(t_j) = P(t_j|v_1, v_2, \ldots, v_k)$. Using Bayes rule:

$$S(t_j) = P(t_j|v_1, v_2, \ldots, v_k) = \frac{P(v_1, v_2, \ldots, v_k|t_j) * P(t_j)}{P(v_1, v_2, \ldots, v_k)}.$$

Next, we assume that given a tag, visterms occur in an image independently of each other. That is,

$$P(v_1, v_2, \ldots, v_k|t_j) = P(v_1|t_j) * P(v_2|t_j) * \ldots * P(v_k|t_j).$$

Such a conditional independence assumption is usually adopted in naive Bayes algorithms to simplify the model. We can also drop the term $P(v_1, v_2, \ldots, v_k)$ from $S(t_j)$ as it is common to all the tags, then

$$S(t_j) \propto P(v_1|t_j) * P(v_2|t_j) * \ldots * P(v_k|t_j) * P(t_j).$$

Multiplying a large number of probability terms might make the score computationally intractable. Therefore, we actually compute the logarithm of the score above, preserving the relative ranking of the tags,

$$log(S(t_j)) = log(P(v_1|t_j)) + \ldots + log(P(v_k|t_j)) + log(P(t_j)).$$

To solve the inverse problem of image retrieval, given a query tag t_j, we compute the conditional probability $P(I_n|t_j)$ for each image in the database. Let I_n be composed of visterms $\{v_1, v_2, \ldots, v_k\}$. The score of I_n is given by:

$$S(I_n) = P(I_n|t_j) = P(v_1, v_2, \ldots, v_k|t_j).$$

Again using the conditional independence assumption,

$$S(I_n) = P(v_1|t_j) * P(v_2|t_j) * \ldots * P(v_k|t_j).$$

An important point to note here is that the images with a large number of visterms will tend to get lower scores as more probabilities are multiplied. One way to address this bias is to take the geometric mean of all the conditional probabilities as the score of an image,

$$S(I_n) = (P(v_1|t_j) * P(v_2|t_j) * \ldots * P(v_k|t_j))^{1/k}.$$

Finally, for computational reasons, we actually compute the log of the score above,

$$log(S(I_n)) = (1/k) * (log(P(v_1|t_j)) + \ldots + log(P(v_k|t_j))).$$

4.1.2 Improved Naive Bayes model

The naive Bayes model works reasonably well on the Corel dataset. However, the Flickr dataset is not as well annotated as the Corel database. For instance, an image of a car might be tagged as {'john', 'car', 'san francisco'} on Flickr. As users tag photos according to their own wishes, such "annotation noise" is quite frequent on Flickr. Indeed, as the experiments will show, the performance of the basic naive Bayes algorithm is quite poor on the Flickr dataset, which calls for additional measures to counter the annotation noise. Let us take an example to illustrate how we aim to address this problem.

Consider two images of cars on Flickr: I_1 tagged as {'john', 'car', 'san francisco'}, I_2 tagged as {'autoshow', 'geneva', 'car', 'black'}. In the basic naive Bayes algorithm, the visterms of I_1 will contribute to the conditional probabilities with tags 'john', 'car' and 'san francisco', that is $P(v_{car}|\text{john})$, $P(v_{car}|\text{car})$ and $P(v_{car}|\text{san francisco})$. Here v_{car} denotes a visual feature related to the 'car' object. Similarly, visterms of I_2 will be associated with 'autoshow', 'geneva', 'car', and 'black', that is $P(v_{car}|\text{autoshow})$, $P(v_{car}|\text{geneva})$, $P(v_{car}|\text{car})$ and $P(v_{car}|\text{black})$. If both I_1 and I_2 are pictures of just cars, $P(v_{car}|\text{san francisco})$ might be adding noise to the model. Therefore, the visterms of I_1 could be considered as "noise" for the tags 'john', 'san francisco', and the visterms of I_2 could be considered as noise for the tag 'geneva'. One possible way to reduce such noise is to consider both I_1 and I_2 together as a "pair". We calculate the common visterms and tags in images I_1 and I_2, and then associate only the common visterms with the common tags. Assuming that both images will have some visterms corresponding to the 'car' object as common, those visterms will now only be linked to the tag 'car', and not to the other "noisy" tags. In other words, the new model will only consider $P(v_{car}|\text{car})$, eliminating $P(v_{car}|\text{san francisco})$, $P(v_{car}|\text{john})$, $P(v_{car}|\text{autoshow})$, $P(v_{car}|\text{geneva})$ and $P(v_{car}|\text{black})$. There is a possibility that some relevant tags will also get eliminated when considering image pairs. For example, if the car in I_2 is *black* in color, eliminating $P(v_{car}|\text{black})$ when considering the pair $\{I_1, I_2\}$ might appear to be removing useful information from the training set. However, note that since we consider all possible image pairs, the tag 'black' would be considered whenever I_2 is paired with any other image I_n that also has the tag 'black'. Further, if a tag is not common in any image pair, it means that its tag frequency is 1. Such a low frequency tag is very likely a "personal tag", or some other rare tag that is not very useful for the purpose of annotation.

Based on the intuition of the example above, we consider pairs of images as a single document rather than each image as a document for calculating the conditional probabilities in the naive Bayes algorithm. Concretely, for each image pair $\{I_n, I_m\}$, we define two terms, namely visual-similarity $sim_V(I_n, I_m)$ and tag-similarity $sim_T(I_n, I_m)$, calculated as the cosine similarity of visterms and tags respectively.

$$sim_V(I_n, I_m) = \frac{V_n.V_m}{norm(V_n) * norm(V_m)}$$

$$sim_T(I_n, I_m) = \frac{T_n.T_m}{norm(T_n) * norm(T_m)}$$

$$sim(I_n, I_m) = sim_V(I_n, I_m) * sim_T(I_n, I_m)$$

where V_x and T_x denote the visterm vector and the tag vector of image I_x respectively, and *norm* denotes the L_2 norm.

The conditional probability of a visterm given a tag is computed using all possible image pairs as single documents, each pair $\{I_n, I_m\}$ weighted by $sim(I_n, I_m)$.

$$P(v_i|t_j) = \frac{\sum_{\{m,n:m \neq n, v_i \in I_m, v_i \in I_n, t_j \in I_m, t_j \in I_n\}} sim(I_m, I_n)}{\sum_{\{m,n:m \neq n, t_j \in I_m, t_j \in I_n\}} sim(I_m, I_n)}.$$

This way of computing $P(v_i|t_j)$ gives more weight to image pairs which have higher similarity in terms of visterms and tags. Next, the annotation and retrieval tasks are performed in the same fashion as in the basic naive Bayes method. As shown later in results, the improved naive Bayes method gives better annotation results on the Flickr dataset. It also improves the results on the Corel dataset, though by a smaller margin. Additionally, this method tends to downweight low frequency tags as they are less likely to be found in a pair of similar images. Overall, it benefits the system as the low frequency tags are more often very "personal" tags that might be considered as noise for the purpose of automatic annotation.

4.2 Graph based model

The improved naive Bayes model helps in the annotation performance for the Flickr dataset but the retrieval performance is still quite low. The increase in annotation performance can be largely attributed to the removal of annotation noise found in images. However, the problem of "limited tagging" is still there, which is one of the main reasons for low retrieval performance. For example, in the training set, if the images tagged as 'bay area' are not also tagged as 'san francisco', the visterms related to 'bay area' will not have a high conditional probability w.r.t. 'san francisco'. Now, in the test set, if the images of 'bay area' are tagged as 'san francisco', it would be very difficult for the naive Bayes model to retrieve them for the query 'san francisco' based on the visual context only. This "limited tagging" illustration provides the intuition that it might be useful to first perform a query expansion and then retrieve images for the expanded query. If the query 'san francisco' is expanded to also include 'bay area', it would now become easier to retrieve images using the trained model. Query expansion is a commonly used technique in text retrieval to enhance the performance for queries that might be insufficient to retrieve the relevant documents due to variety of reasons [Xu and Croft, 1996]. Term co-occurrence in documents is often used for query expansion to find related terms given some input terms. In our case, query expansion should also look beyond the immediate tag co-occurrence as the tags 'san francisco' and 'bay area' might not occur together very often in the training set. We aim to build a graph model that captures these notions to enhance the retrieval performance.
In our formulation, each tag and visterm contributes a node to a graph. Weighted directed edges between nodes represent the conditional probabilities. Concretely, there are three kinds of edges:

tag-to-tag edges An edge from tag t_i to tag t_j, $e(t_i, t_j)$ is weighted by $P(t_j|t_i)$.

tag-to-visterm edges An edge from tag t_i to visterm v_j, $e(t_i, v_j)$ is weighted by $P(v_j|t_i)$.

visterm-to-visterm edges An edge from visterm v_i to visterm v_j, $e(v_i, v_j)$ is weighted by $P(v_j|v_i)$.

The conditional probabilities are calculated in the same way as in the naive Bayes method.

$$P(t_j|t_i) = \frac{n_I(t_j, t_i)}{n_I(t_i)}; P(v_j|t_i) = \frac{n_I(v_j, t_i)}{n_I(t_i)}; P(v_j|v_i) = \frac{n_I(v_j, v_i)}{n_I(v_i)}.$$

However, to limit the number of edges and reduce noise, we propose to calculate "*support*" and "$confidence$" for each edge, and keep only those edges for which *support* $\geq \alpha$, where α depends on the type of edge. For instance,

$$\textit{support} = P(t_j, t_i) = \frac{n_I(t_j, t_i)}{\#\textit{documents}},$$

$$confidence = P(t_j|t_i) = \frac{n_I(t_j, t_i)}{n_I(t_i)}.$$

Here, the *confidence* values are the weights of the edges, and *support* values are just used for pruning the edges. A low *support* value indicates that we do not have enough training data for that particular edge. This approach is commonly used in association rule mining [Agrawal and Imielinski, 1993]. Once we build such a graph from the training set, there are three steps for retrieving images. A query expansion step, a cross-mapping step, and an image ranking step. Each of these steps are described in the following sections.

4.2.1 Query expansion

Let us illustrate the concept with a toy-example. Consider that the tag subgraph obtained from the training data looks like the one in Figure 4.1. If the query is

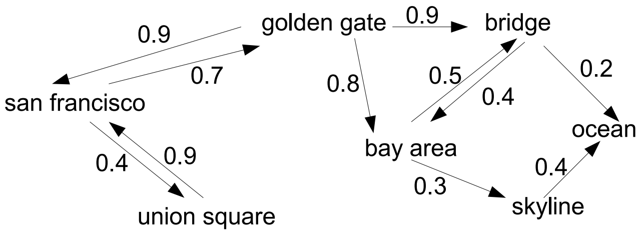
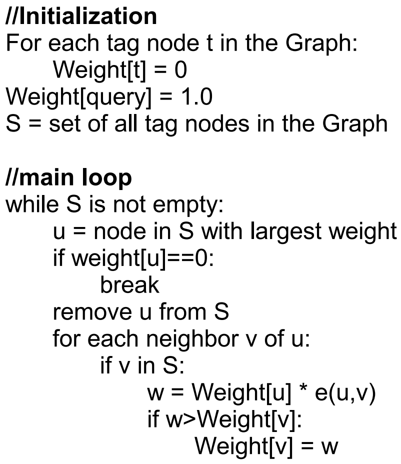


Figure 4.1: Subgraph showing tag nodes and edges.

'san francisco', we give a weight of 1.0 to the tag node 'san francisco'. The rest of the nodes are weighted by a heuristic method. Following the edges, 'golden gate' can be given a weight of Weight(san francisco)*e(san francisco, golden gate) = 1.0*0.7 = 0.7. Similarly, 'union square' will get a weight of 0.4 but we also need to reach the other tags such as 'bay area', 'skyline', etc. Missing edges could arise due to the limited number of images and tagging information in the training set. To calculate the score for the tag 'bay area', one possibility is to "chain" the probabilities along a path from 'san francisco' to 'bay area'. For

instance, Weight(bay area) = Weight(san francisco) * e(san francisco, golden gate) * e(golden gate, bridge) * e(bridge, bay area) = 1.0*0.7*0.9*0.4 = 0.252. Observe that there exists another path to calculate the same score. Weight(bay area) = Weight(san francisco) * e(san francisco, golden gate) * e(golden gate, bay area) = 1.0*0.7*0.8 = 0.560. The path that gives the highest score for a tag best represents the “cohesiveness” of the tag with the query tag. In this example, we would take the score of ‘bay area’ as 0.560.
The above example illustrates that a variation of the well-known Dijkstra’s shortest path algorithm [Dijkstra, 1959] can be used to calculate the scores for all the tags in the graph. Figure 4.2 gives the algorithm. In our modified version, instead of adding edge weights and keeping the minimum path value as the label of each node, we multiply the edge weights and keep the maximum path value as the label of each node. The rest of the algorithm remains the same. In case of multiple tags in the query, we make $Weight(q) = 1.0$ during initialization for each tag q in the query.



```
//Initialization
For each tag node t in the Graph:
    Weight[t] = 0
Weight[query] = 1.0
S = set of all tag nodes in the Graph

//main loop
while S is not empty:
    u = node in S with largest weight
    if weight[u]==0:
        break
    remove u from S
    for each neighbor v of u:
        if v in S:
            w = Weight[u] * e(u,v)
            if w>Weight[v]:
                Weight[v] = w
```

Figure 4.2: Algorithm for calculating tag weights during query expansion.

Using the visterm-visterm edges, we can also do query expansion for visterms in a similar fashion for the annotation task. In practice, however, we did not find it useful as we typically had enough visterms from the query image and adding any other visterms led to an increase in noise. This was a somewhat expected result due to the large number of visterms usually present in an image compared to typically small number of tags.

4.2.2 Cross-mapping

The expanded query has a weight for each tag. Next, we calculate the weight of each visterm as:

$$Weight(v_i) = \sum_{t_j} Weight(t_j) * IDF(t_j) * e(t_j, v_i)$$

where $IDF(t_j)$ denotes the inverse document frequency of tag t_j calculated as

$$IDF(t_j) = log\left(\frac{n_I}{n_I(t_j)}\right)$$

where n_I is the total number of images and $n_I(t_j)$ is the number of images with tag t_j. The aim here is to normalize the weights of high frequency tags to avoid a bias. $Weight(v_i)$ is computed such that more weight is given to visterms that have higher conditional probabilities $P(v_i|t_j)$ with a large number of high weight query tags.

4.2.3 Image Ranking

Once we have a weight of each visterm, we need to rank the images. We use the traditional TF*IDF setup here similar to text document retrieval. Each image I_n has a weight vector V_n of visterms.

$$V_n(v_i) = TF_n(v_i) * IDF(v_i)$$

where $TF_n(v_i)$ is the term frequency of v_i in I_n normalized by the total number of visterms, and $IDF(v_i)$ is the inverse document frequency calculated as $log(n_I/n_I(v_i))$.
Let Q represent the vector of visterm weights obtained from the cross-mapping step. To generate a ranked list of images, the score of an image is calculated as:

$$S(I_n) = V_n.Q$$

Images are shown in the order of decreasing scores and precision-recall is calculated using the ground truth available in the test set.

It is possible to construct a similar method for the image annotation task. However, in our experiments, we did not find much improvement in annotation due to the reason explained in the query expansion section.

Chapter 5

Experiments

We will first describe the datasets used in Section 6.1. The sections following that will describe the aggregated data strategy in Flickr, evaluation setup and the results respectively.

5.1 Data Sets

We performed our experiments on two datasets:

5.1.1 Corel Dataset

The first dataset is constructed from the publicly available *Corel Stock Photo Library*. This dataset is well annotated manually using a limited vocabulary size and has offered a good testbench for algorithms. [Barnard et al., 2003] organized images from this collection into 10 different samples of roughly 16,000 images, each sample containing training and test sets. We use the same 10 sets in our experiments and report the performance numbers averaged over all the sets (standard deviation was around 1%). Each set has on average 5240 training images, 1750 test images, and a vocabulary size of 150 tags.

5.1.2 Flickr Dataset

We use a subset of the Flickr data used in [Negoescu and Gatica-Perez, 2008]. This subset consists of roughly 65k images by 4k randomly chosen users. We used the top 2k tags out of 10k tags, in terms of frequency, as the vocabulary. While Corel may be considered as an artificially constructed dataset, Flickr represents images and annotations by real world users. Flickr images are usually very rich in terms of content, often containing multiple objects. A few tags with each image is quite restrictive to describe the image completely or to build effective models. In our experiments, instead of considering each image as a single document, we aggregated the visterms and tags from all the images for a particular user, and considered that as a single document. In this way, each user contributes a single document to the corpus, and then users are partitioned into training and test sets. The average number of images per user was 12. The motivation for doing such an aggregation will become clear from the Canonical Correlation Analysis (CCA) described in Section 5.2.

5.2 Canonical Correlation Analysis (CCA)

We work with the complete set of 65k Flickr images and the 10k tag vocabulary in this analysis. An image I has a set of visterms $S^V : \{v_1, v_2, \ldots, v_{N_v}\}$ and a set of tags $S^T : \{t_1, t_2, \ldots, t_{N_t}\}$. For this analysis, we first map visterms and tags to a lower dimensional concept space using Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. S^V is mapped to a probability distribution over 100 latent topics. Each topic is a probability distribution over 2000 visterms:

$$p(S^V|\alpha_v, \beta_v) = \int p(\theta_v|\alpha_v) \left(\prod_{i=1}^{|S^V|} \sum_{k=1}^{100} p(z_k^{(v)}|\theta_v) p(v_i|z_k^{(v)}, \beta_v) \right) d\theta_v,$$

where α_v, β_v are corpus level parameters, θ_v is the topic distribution for a document, and $p(v_i|z_k^{(v)}, \beta)$ is the probability distribution of visterms for topic $z_k^{(v)}$ as described in [Blei et al., 2003]

Similarly, S^T can be mapped to a probability distribution over 100 latent topics. Each topic in this case is a probability distribution over 10k tags:

$$p(S^T|\alpha_t, \beta_t) = \int p(\theta_t|\alpha_t) \left(\prod_{j=1}^{|S^T|} \sum_{k=1}^{100} p(z_k^{(t)}|\theta_t) p(t_j|z_k^{(t)}, \beta_t) \right) d\theta_t.$$

For image annotation and retrieval to work, the image content should be correlated to its tag annotations. For our purposes, we would like to measure correlation between topic distribution for visterms θ_v and topic distribution for tags θ_t. Canonical Correlation Analysis (CCA) [Hotelling, 1936] is a method to measure correlation between two multi-dimensional variables. It finds bases for each variable such that the correlation matrix between the basis variables is diagonal and the correlations on the diagonal are maximized. Concretely, for two multi-dimensional variables X and Y, CCA first finds basis vectors α_1 and β_1 such that the correlation between the scalar quantities $\alpha_1.X$ and $\beta_1.Y$ is maximized. The entities $u_1 = \alpha_1.X$ and $v_1 = \beta_1.Y$ are called the *first pair of canonical variables* and their correlation $\rho_1 = correlation(u_1, v_1)$ is called the *first canonical correlation coefficient*. Next, CCA finds a second pair of basis vectors α_2 and β_2 ($u_2 = \alpha_2.X$, $v_2 = \beta_2.Y$) such that the correlation $\rho_2 = correlation(u_2, v_2)$ is maximized subject to the constraint that the *second pair of canonical variables* u_2 and v_2 is uncorrelated with the first pair u_1 and v_1. This procedure is continued such that the r^{th} pair of canonical variables u_r and v_r is uncorrelated with the first $(r-1)$ canonical variables. $\rho_r = correlation(u_r, v_r)$ is called the r^{th} *canonical coefficient*. The dimensionality of the canonical variables is equal

Measure	Flickr images		Corel images
	Individual	Aggregated	
max	0.25 (0.01)	0.35 (0.12)	0.53 (0.07)
sum	1.54 (0.25)	4.70 (3.05)	6.47 (1.72)

Table 5.1: Maximum and sum of correlation values among corresponding canonical variables for visterm topics and tag topics. The number in brackets indicate the correlation values when we randomize the tag assignment to images.

to or less than the dimensionality of either of original variables. Table 5.1 shows the maximum (first) and the sum of correlation values between corresponding canonical variables for visterms and tags. To see how significant this correlation is, we randomized the tag assignment to images and then calculated the correlation. A significant drop in correlation for the randomized case is an indicator that the tags associated with images are not random but have some relation with the content of the image. Furthermore, when we aggregate the visterms and tags for all images from a single user, the assumption is that this aggregation process would preserve the association between visterms and tags while enriching the tag collection of a document. As shown in Table 5.1, the aggregation process in the Flickr data indeed increases the correlation between visterms and tags. This suggests that we might get a better performance by considering all the images from a user as a single document. The Flickr results described further have been calculated from the aggregated dataset. For comparison, we also performed CCA on Corel image collection. The aggregated Flickr model still has lower correlation values compared to Corel, primarily due to the more careful annotations, limited vocabulary and relatively “simple” images in Corel.

5.3 Evaluation Setup

The experimental setup is as follows: we train the naive Bayes and graph models from the training set. For annotation, given an image from the test set, we count the suggested tag as relevant only if it is present in the reference annotations. For retrieval, each tag in the vocabulary is used as a query and a ranked list of suggested images is obtained. An image is considered as relevant only if it contains the query tag in the reference annotations. While this setup appears reasonable for Corel dataset, it is particularly harsh for the Flickr dataset. For example, an otherwise relevant suggested tag would be considered irrelevant if the user did not add it to his/her image. Likewise for retrieval, an image showing ‘golden gate bridge’ would be considered irrelevant for the query ‘golden gate’ if the user did not tag that image with ‘golden gate’. Ideally, one would like to conduct a user study to address this issue but such studies are difficult for large datasets. In this work, we rely only on the annotations done by actual Flickr users which means that the performance numbers may be a conservative estimate of the "true" performance. The following three standard performance measures are used for both annotation and retrieval:

P@1 Precision value at position 1 in the results.

MAP Mean Average Precision. Average precision (AP) of a single query is the mean of precision scores after each relevant item is returned. MAP is the mean of individual AP scores.

$$AP = \frac{\sum_{r=1}^{N}(P(r) * rel(r))}{\text{number of relevant documents in the whole corpus}},$$

$$MAP = \frac{\text{sum of AP for all queries}}{\text{number of queries}},$$

where r is the result position, N is the number of results retrieved, $P(r)$ is the precision at position r, $rel(r)$ is 1 if position r has a relevant result and 0 otherwise.

Acc Accuracy: defined as the precision at position p where p is the number of relevant documents for the query in the whole corpus.

5.4 Results

Table 5.2 shows annotation performance on both Corel and aggregated Flickr datasets. N.B. is used as an abbreviation for Naive Bayes. The improved naive Bayes algorithm increases the performance on both Corel and Flickr datasets, the improvement being much larger on Flickr. The huge improvement for Flickr is due to the reduction in "tagging noise" when pairs of images are used as documents. Further, since the Corel dataset has much "simpler" images and much better annotations than Flickr, one might expect the same algorithm to perform better on Corel. This would mostly be true if we were considering individual images in Flickr rather than the aggregated set. However, as shown in the precision-recall graph in Figure 5.1, the precision numbers for the first few positions are higher in Flickr than in Corel. This could be explained by the fact that the aggregation process expands the set of ground truth tags for Flickr. As a result, the annotation algorithm has simply more choice of tags to predict. However, the expansion in the size of ground truth also lowers the recall values. This is the reason why MAP and Accuracy values are lower compared to Corel. Table 5.4 shows some example queries and results for the annotation task. For Flickr queries, we use all the images from a single user's profile. It was not possible to show all those images in this example, so we included a few images that looked representative of the true and suggested tags.
Table 5.3 shows the retrieval performance of the different algorithms and Figure 5.2 shows the precision-recall curve. Both the improved naive Bayes algorithm

	Measure	Basic N.B.	Improved N.B.
Corel	P@1	0.348	0.440
	MAP	0.362	0.387
	Acc	0.283	0.326
Flickr	P@1	0.001	0.430
	MAP	0.012	0.219
	Acc	0.003	0.259

Table 5.2: Annotation performance comparison.

	Measure	Basic N.B.	Improved N.B.	Graph
Corel	P@1	0.330	0.370	0.344
	MAP	0.168	0.175	0.170
	Acc	0.182	0.189	0.187
Flickr	P@1	0.005	0.033	0.165
	MAP	0.018	0.051	0.069
	Acc	0.010	0.042	0.062

Table 5.3: Retrieval performance comparison.

and the Graph based algorithm result in a modest increase in Corel performance compared to the basic model. However, since the numbers for Corel are so close, it is very hard to say which algorithm is performing better. We might be observing a “ceiling effect” here which means that these numbers could be close to the performance limit of these algorithms for the Corel dataset. The low performance numbers for the Flickr dataset are mainly due to the reason that it is very hard to rank the content rich images based on the weight of the visterms. Nevertheless, we still see an increase in performance when using the improved naive Bayes algorithm and a further increase when using the Graph based approach. Also, as mentioned earlier, the performance numbers for Flickr show only a conservative estimate of the “true” performance owing to our evaluation setup. Table 5.5 shows some retrieval examples.

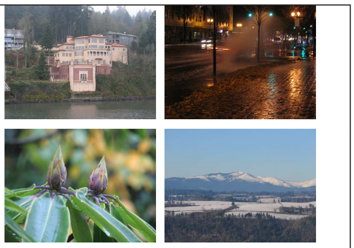
	Dataset	Corel	Flickr
example 1	Query Image(s)		
	True Tags	**beach**, **clouds**, **sky**, **water**	**brick**, **house**, **car**, **clouds**, **tree**, **polaroid**, etc.
	Basic N.B.	**clouds**, horizon, hills, mountain	rob, mexico city, cape town, orange county
	Improved N.B.	**water**, **sky**, **clouds**, tree	**people**, **street**, **tree**, **car**, **house**, sky
example 2	Query Image(s)		
	True Tags	**cat**, **ground**, **lion**, **tree**	**oslo**, **norway**, **house**, **night**, **adventure**, **blue**, etc.
	Basic N.B.	**lion**, mane, **cat**, trunk	final, stencils, republic, oc
	Improved N.B.	**lion**, **tree**, **cat**, mane	**sky**, **house**, **night**, bw, **red**, **blue**

Table 5.4: Annotation examples. Predicted tags are shown in the order of rank, that is, the first tag is suggested at position 1. Correctly predicted tags are shown in bold green, incorrectly predicted tags are shown in light red. For Flickr, a document consists of aggregated visterms and tags for a single user. The above example shows representative images and tags from a single user's profile.

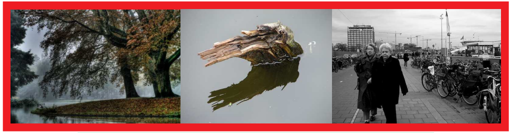
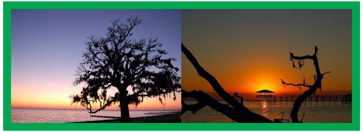
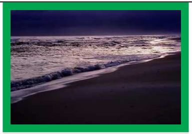
Dataset	Corel	Flickr
Query Tag	clouds	clouds
Basic N.B.	 	
Improved N.B.	 	  
Graph	 	

Table 5.5: Retrieval examples. First 3 results are shown for each algorithm in the order of rank. That is, the first result shown is retrieved at position 1. Relevant results are shown with a green background and irrelevant with a red background. For Flickr, since a single result represents all the images from a user's profile, representative images from the corresponding user's profile are shown here.

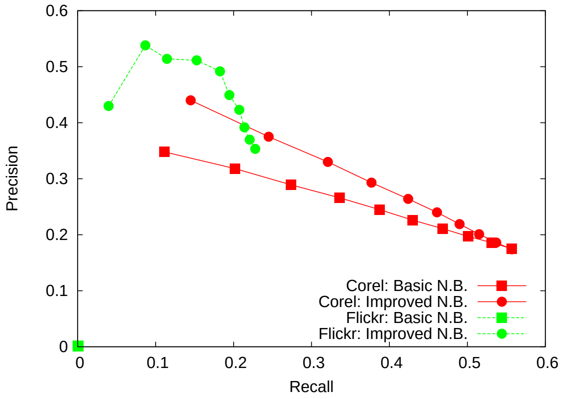


Figure 5.1: Precision-Recall curves for annotation performance.

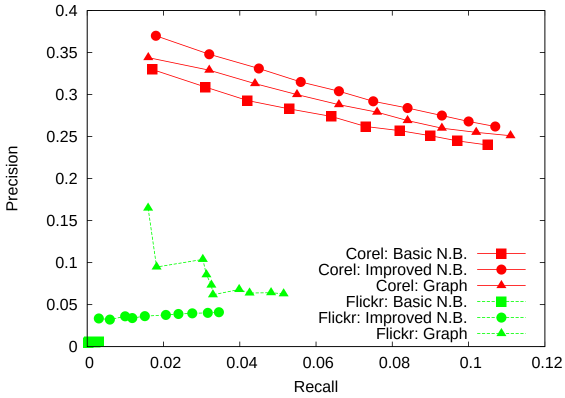


Figure 5.2: Precision-Recall curves for retrieval performance.

Chapter 6

Conclusions

We have studied two models for automatic image annotation and retrieval based on co-occurrence of visual features and tag annotations in the images. The proposed algorithms are designed to address the noise in large scale image databases such as Flickr and show gains in performance. The improved naive Bayes model suggests that it might be useful to look at “pairs of images” to reduce the annotation noise in images. The graph-based model suggests that query expansion could bring performance gains for the retrieval task.

For future work, we would like to experiment with different vocabulary sizes for visterms and tags for Flickr, to understand how that affects the performance. Expanding the visterm and tag vocabulary sizes helps to capture more information from the corpus but also makes the system more susceptible to noise and difficult to model. A different content aggregation for Flickr might also be fruitful. Aggregating all the images from a user might increase the noise if the images and/or tags are not similar or do not represent similar topics. An alternative would be to aggregate based on content, that is, aggregate only those images for which the visterm and/or tag vectors are similar. This might result in a significant performance boost for Flickr. We would also like to experiment with topic based models such as LDA and PLSA to see if using the topic distribution for visual features rather than raw visterm counts could be beneficial.

Bibliography

R. Agrawal and T. Imielinski. Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1993.

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3: 1107–1135, 2003. ISSN 1533-7928.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

D. M. Blei and M. I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3.

E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. *ACM Trans. Web*, 1(2):7, 2007. ISSN 1559-1131.

P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Lecture Notes in Computer Science*, pages 97–112, 2002.

N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 67–74, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-093-7.

D. Hardoon, C. Saunders, S. Szedmak, and J. Shawe-Taylor. A correlation approach for automatic image annotation. In *The 2'nd International Conference on Advanced Data Mining and Applications*, volume 4093, pages 681–692. Springer, 2006.

T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and* *development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1.

H. Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3-4): 321–377, 1936.

J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3.

L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5.

P. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–378, 2000.

J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003.

J. Li, R. Gray, R. Olshen, et al. Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models. *IEEE Transactions on Information Theory*, 46(5):1826–1841, 2000.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1802–1817, 2007.

F. Monay, P. Quelhas, J. Odobez, and D. Gatica-Perez. Contextual classification of image patches with latent aspect models. *EURASIP Journal on Image and Video Processing*, 2009, 2009.

Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

R. Negoescu and D. Gatica-Perez. Analyzing Flickr Groups. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 417–426. ACM New York, NY, USA, 2008.

J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 653–658, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1.

T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2.

J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 1, 2005.

L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li. Flickr distance. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 31–40, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-303-7.

J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8.

C. Yang and T. Lozano-Perez. Image database retrieval with multiple-instance learning techniques. In *Proceedings of the International Conference on Data Engineering*, pages 233–243. IEEE Computer Society Press, 2000.