Columbia TAG System - Transductive Annotation by Graph Version 1.0

Jun Wang and Shih-Fu Chang
{*jwang, sfchang*}*@ee.columbia.edu*

Columbia University ADVENT Technical Report #225-2008-3
Oct 15, 2008

1 Introduction

The Columbia TAG (Transductive Annotation by Graph) system is deigned to facilitate rapid retrieval and exploration of large image and video collections. It incorporates novel graph-based label propagation methods and intuitive graphic user interfaces (GUI) that allow users quickly browse and annotate a small number of images/videos, and then in real or near real time receive refined labels for all remaining unlabeled data in the collection. Using such refined labels, additional positive images/videos matching user's interest can be quickly discovered. It can be used as a fast search system alone, or a bootstrapping system for developing additional target recognition tools needed in critical image application domains such as intelligence, surveillance, consumer, biomedical, and Web.

TAG system differs from the traditional approaches that are based on automatic image classification. These methods usually require a sufficiently large number of labeled samples to train classifiers - a method referred to as supervised learning. Instead, TAG minimizes the burden of manual labeling on users. The objective is to leverage the best use of whatever user input available (as few as one or two samples per class) and propagate such information in the most effective way to all the remaining data in the database. Specifically, we use novel graph-based transductive learning methods developed in our prior works [1, 2] to address several challenging issues, such as unbalanced, noisy, and biased labels, and achieve promising performance in several application domains, including bimolecular images, web documents, and satellite images.

TAG system is different from prior works using semi-supervised learning, which utilize both labeled and unlabeled data in learning the classifier or inferring labels of new data. Most semi-supervised techniques focus on separation of labeled samples of different classes while taking into account distribution of the unlabeled data. The performance of such methods often suffers from the scarcity of labeled data, invalid assumptions about classification models or distributions, and sensitivity to non-ideal label conditions. To overcome these issues, we adopt the graph-based label propagation paradigm that makes least assumptions about the data and classifier. One central principle of this paradigm is that data share and propagate their labels with other data in their proximity, defined in the context of a graph. Data are represented as nodes in a graph and the structure and edges in the graph define the relation among data. Propagation of labels among data in a graph is intuitive, flexible, and effective, without requiring complex models for the classifier and data distributions. Moreover, our graph inference method improved the existing graph learning approach in terms of the sensitivity to weak labels, graph construction, and noisy situations [2].

Starting with the small number of labels given by users, the graph-based transductive learning method propagates the initial labels to the remaining data and predicts the most likely labels (or scores) for each data in the graph. The propagation process is optimized with respect to several criteria. How well do the predictions fit the already known labels given by the user? What's the regularity of the predictions over data in the graph? Is the propagation process amicable to addition of new labels? Are the results sensitive to quality of the initial labels and specific ways the labeled data are selected?

The TAG system can be used in different modes - interactive and automatic. The interactive mode is designed for applications in which a user uses the GUI to interact with the system in browsing, labeling, and providing feedback. The automatic mode takes the initial labels or scores produced by other processes and then output refined scores or labels for all the data in the collection. The processes providing the initial labels may come from various sources, such as other classifiers using different modalities, models, or features (EEG signals, computer vision model, etc), rank information of the data from other search engines, or even other manual annotation tools. When dealing with labels/scores from imperfect sources (e.g., EEG classifiers and search engines), special care is needed to filter the initial labels and assess their reliability before using them as input for propagation.

The output of TAG system consists of refined or predicted labels (or scores indicating likelihood of positive detection) of all the images in the collection. Such output can be used to identify additional positive samples matching targets of interest, which in turn can be used to train more robust classifiers, arrange the best presentation order for image browsing, or rearrange image presentations for EEG-based image visualization.

In this report, we present summary of prior arts, system overview, usage modes, summary of propagation process, and some of sample applications we have tested.

2 Comparison with Prior Work

There have been prior works exploring use of user feedback in improving the image retrieval experience. In [3], relevance feedback provided by the user is used to indicate which images in the returned results are relevant or irrelevant to the search target user has in mind. Such feedback can be indicated explicitly (by marking labels of relevance or irrelevance) or implicitly (by tracking specific images viewed by the user). Given such feedback information, the initial query (either in the form of keywords or example images) can be modified. For example, the following equation describes a simple implementation that generates a new query based on linear combination of the original query and samples of relevant and irrelevant images. Alternatively, the underlying features and distance metrics used in representing and matching images can be refined using the above relevance feedback information [4, 5]. Though such ideas are intuitive and easy to implement, applications in practical domains have not shown effective results. There is no guarantee that the refined query, feature, or metric will improve the capability of retrieving additional targets that have been missed in the initial results.

$$\{\text{n}ew\ \ query\} = \alpha \cdot \{\text{i}nitial\ \ query\} + \beta \cdot \{\text{p}ositive\ \ feedback\} + \gamma \cdot \{\text{n}egative\ \ feedback\} \quad (1)$$

In another thread of research, researchers attempt to answer the question that given a set of labels, the remaining data in the collection that have not been labeled, and the existing model learned by machines using the existing information, what will be the best data sample in the next iteration of user inspection or observation? The objective is to actively select the most beneficial sample for observation so that the uncertainty about the classification model can be reduced to the largest extent. In contrast with the conventional machine learning methods that passively sample data for labeling, such approaches select sample data in an active way, therefore referred to as active learning in the literatures [6, 7, 8]. Active learning methods have shown very promising results in interactive multimedia retrieval. However, in most cases supervised learning techniques are used and a non-trivial number of labeled data are needed in order to learn a classifier with reasonable quality. Such requirements make them non-competitive when there are only very few labeled samples available. In addition, most active learning methods select data that are difficult to classify, aiming at resolving the uncertainty near the local point. However, such methods ignore the impact of additional labels to a larger extent, including other data in the unlabeled collection.

Given a mixture of labeled and unlabeled data, better machine learning models can be learned in order to discriminate labeled data from different classes, and simultaneously considering the distribution structures of the rest of data that are not labeled yet. Such techniques, referred to as semi-supervised learning or transductive learning, have attracted a lot of attention from researchers due to its major advantage that the manual labeling cost can be greatly reduced. Among the various options, transductive graph-based diffusion has shown great promises in predicting classification labels in challenging cases such as those have very few initial labels only [9, 10, 11]. Such methods take as input the initial labels of few samples and propagate them to the rest of data. The propagation process is done via a graph which describes the similarity between each sample and its neighbors, and the connectivity structures among the samples in the collection. Several methods have been developed in this area, such as local and global consistency [9], the method based on Gaussian fields and harmonic functions [10], and other related methods using manifold regularization framework proposed in [12, 13] where graph Laplacian regularization terms are combined with regularized least squares (RLS) or support vector machine (SVM) optimization criteria. These methods lead to graph-regularized variants called Laplacian RLS (LapRLS) and Laplacian SVM (LapSVM) respectively.

The existing graph-based transductive learning techniques, though promising, are still inadequate under several challenging conditions in practice. For example, the interactive retrieval process often lead to unbalanced situations in which labeled samples from one class often significantly outnumber those from different classes. Such conditions often cause inaccurate results from label propagation. In addition, the data samples and their observed features may be subject to a large level of noise, causing confusing and ambiguous cases for classification. Furthermore, data labeled by users may be sampled from the underlying data set in a biased way, leading to biased coverage of the data set and thereby incorrect classification results.

In the TAG system, we implement several novel ideas developed in our prior works in [1, 2] to address the problems mentioned above. Specifically, we use an iterative optimization method to improve the label propagation accuracy. During each iteration of the process, the most informative label is automatically selected and its class label is automatically predicted. The added label sample is then added to the existing labeled pool and the optimal predicted labels for all of the rest of the unlabeled data are then computed. Such techniques improve the quality of the label propagation results by avoiding an aggressive step of predicting a large number of labels from a small number of labels. Instead, it implements a judicious procedure to predict new labels incrementally, starting from the most informative ones.

In addition, we apply a novel graph regularization method to effectively address the class imbalance issue. Specifically, each class is assigned an equal amount of weights and each member of a class is assigned a weight proportionally to its connection density and inversely proportional to the number of samples sharing the same class.

Finally, the TAG system includes a novel incremental learning method that allows addition of new labeled samples efficiently. Each time when user labels more data, the results can be quickly updated using a superposition process without repeating the entire propagation process. Influence by the new labels can be easily added to the original predicted labels. Such incremental learning capabilities are important for achieving real-time responses in user's interaction with the system.

3 TAG System Overview

We present the system diagram of TAG system in Figure 1. Given a collection of images or video clips, TAG system builds an affinity graph to capture the relationship among individual images or videos. The graph is also used to propagate information from labeled data to the large number of data in the same collection. In the following, we will walk through the main processed involved in building the graph and using the graph for label propagation.

3.1 Feature Extraction and Graph Construction

Each node in the graph represents a basic entity (data sample) of retrieval and annotation. It can be an image, a video clip, a multimedia document, or an object contained in an image or video. In the ingestion process, each data sample is first pre-processed (e.g., scaling, partitioning, noise reduction, smoothing, quality enhancement etc). Some pre-filter may also be used to filter likely candidates of interest (e.g., images that are likely to contain targets of interest). After

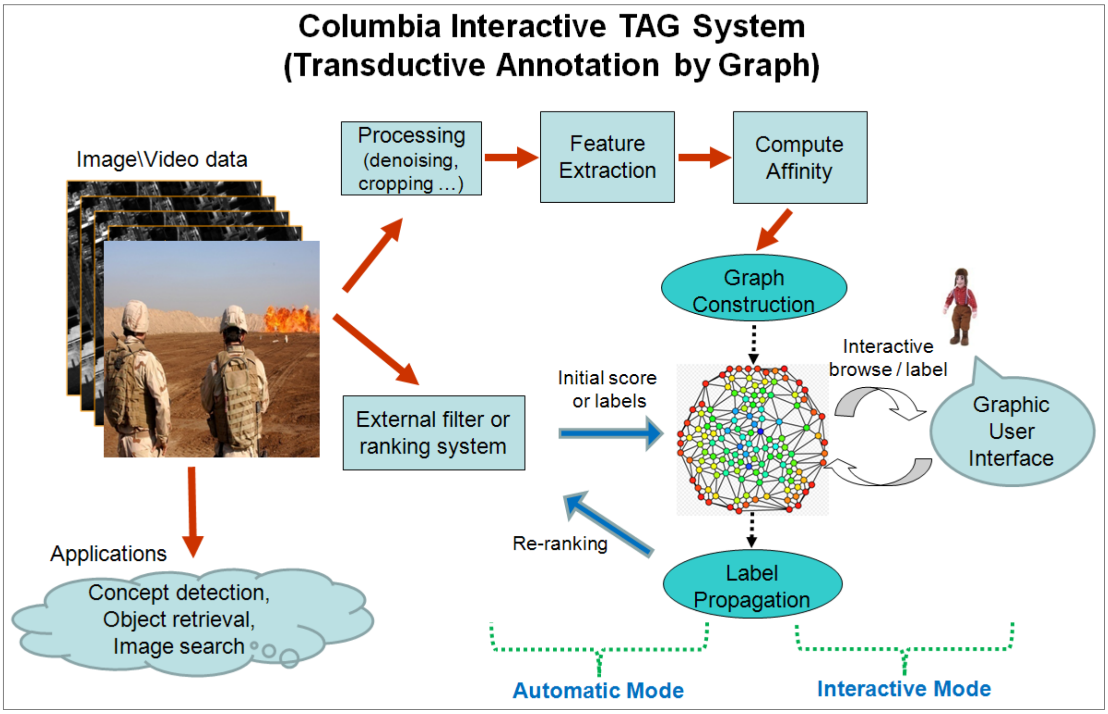

Columbia Interactive TAG System
(Transductive Annotation by Graph)
Image\Video data
Processing (denoising, cropping ...)
Feature Extraction
Compute Affinity
Graph Construction
Interactive browse / label
Graphic User Interface
Initial score or labels
External filter or ranking system
Re-ranking
Label Propagation
Applications
Concept detection, Object retrieval, Image search
Automatic Mode
Interactive Mode

Figure 1: The system diagram and usage modes of Columbia TAG (Transductive Annotation by Graph) System.

pre-processing and filtering, features are extracted from each sample. TAG does not dictate usage of specific features. Any feature set preferred by practical applications may be used, such as global features (color, texture, edge), local features (such as local interest points), and spatial information (such as layout). Multiple types and modalities of features may also be aggregated or combined. Given the extracted features, affinity (or similarity) between each pair of samples is computed. Again, no specific metrics are required by TAG, though judicious choices of features and similarity metrics often play a critical role in determining the quality of the final label prediction results. The pair-wise affinity values are then assigned to be weights of the corresponding edges in the graph. Usually, weak edges with small weights are pruned to reduce the complexity of the affinity graph. Alternatively, a fixed number of edges may be set for each node by finding a fixed number of nearest neighbors for each node.

3.2 Annotation and Browsing

With the affinity graph in place, TAG system is ready to be used for retrieval and annotation. TAG currently provides two different modes for such processes. In the Interactive Mode, users browse, view, inspect, and label images or videos through a graphic user interface (GUI) described later in this document. Initially before any label is assigned, a subset of data may be shown in the browsing window by using certain metadata (time, ID, etc) or simply random sampling of the collection. Using the GUI, user may view any image of interest and then provide feedback about relevance of the result (e.g., marking the image as relevant or irrelevant). Such labels can then be encoded as labels and assigned to the corresponding nodes in the graph.

In the Automatic Mode, the initial labels of a subset of nodes in the graph may be provided by some external filters, classifiers, or ranking systems. For example, if the target of interest is "helipad" for military intelligence analysis in satellite imagery, an external classifier using image features and computer vision classification models may be used to predict whether the target is present in an image and assign the image to the most likely class (positive vs. negative). If the target is product image search for Web images (say "automobile"), external Web image search engines may be used to retrieve most likely images using keyword search. The rank information of each returned image can then be used to estimate the likelihood of detecting the target in the image and approximate the class scores which can be assigned to the corresponding node in the graph. As mentioned above, each node in the graph is associated with either a binary label (positive vs. negative) or a continuous-valued score approximating the likelihood of detecting the target.

3.3 Graph-Based Label Propagation

Given the assigned labels or scores for some subset of the nodes in the graph (usually a very small portion of the entire graph), a key function of the TAG system is to propagate the labels to other nodes in the graph in the most accurate and efficient way. Such propagation process needs to be fast, completed in the real time or near-real time in order to keep users engaged. After the propagation process is completed, the predicted labels of all the nodes of the graph are used to determine the best order of presenting the results to the user. One typical option is to rank the images in the database in the descending order of likelihood so that user can quickly find additional relevant images. An alternative is to determine the most informative data to show to the user so that human inspection and labels may be collected for such critical samples. The objective is to maximize the utility of the user interaction so that the best prediction model and classification results can be obtained with the least amount of manual user input.

The graph propagation process may also be applied to predict labels for new data that are not yet included in the graph. Such processes may be based nearest neighbor voting or some forms of extrapolation from existing graph to external nodes.

4 Summary of Label Propagation Process

Here we briefly describe the graph based label propagation algorithm used in the TAG system. Comparing with existing graph transduction approaches, there are two major innovations of our methods. First, to handle the interactive and real time requirements, we use a novel graph superposition method to incrementally update the label propagation results, without the need of repeating computation associated with prior labeled samples. Second, to solve noisy and uninformative label problem, we use an alternate optimization technique to achieve a greatly improved accuracy from graph label propagation. The detailed algorithms can be found in [1, 2].

Consider the image set $\mathcal{X} = (\mathcal{X}_l, \mathcal{X}_u)$ consisting of labeled samples $\mathcal{X}_l = \{\mathbf{x}_1, \cdots, \mathbf{x}_l\}$ and unlabeled samples $\mathcal{X}_u = \{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$. The corresponding labels for the labeled data set are denoted as $\{y_1, \cdots, y_l\}$, where $y_i \in \mathcal{L} = \{1, \cdots, c\}$. For transductive learning, the objective is to infer the labels $\{y_{l+1}, \cdots, y_n\}$ of the unlabeled data $\{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, where typically $l << n$, namely only a very small portion of data are labeled. The graph transduction methods define an undirected graph represented by $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where the set of node or vertices is $\mathcal{X} = \{\mathbf{x}_i\}$ and the set of edges is $\mathcal{E} = \{e_{ij}\}$. Each sample $\mathbf{x}_i$ is treated as the node on the graph and the weight of edge e_{ij} is w_{ij}. Typically, one uses a kernel function $k(\cdot)$ over pairs of points to recover weights, in other words $w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ with the RBF kernel being a popular choice. The weights for edges are used to build a weight matrix which is denoted by $\mathbf{W} = \{w_{ij}\}$. Similarly, the node degree matrix $\mathbf{D} = diag\left([d_1, \cdots, d_n]\right)$ is defined as $d_i = \sum\limits_{j=1}^{n} w_{ij}$. The binary label matrix $\mathbf{Y}$ is described as $Y \in \mathcal{B}^{n \times c}$ with $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ has label $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise.

Graph based semi-supervised learning methods propagate label information from labeled data to unlabeled data by treating all samples as nodes in a graph and using edge-based affinity functions between all pairs of nodes to estimate the weight of each edge. Most methods then define a continuous classification function $F \in \mathcal{R}^{n \times c}$ that is estimated on the graph to minimize a cost function. The cost function typically enforces a tradeoff between the smoothness of the function on the graph of both labeled and unlabeled data and the accuracy of the function at fitting the label information for the labeled nodes. In trading off smoothness for accuracy, some recently proposed approaches attempt to preserve label consistency on the graph [9, 10]. In both these two methods, the loss function involves the additive contribution of two goodness terms the global smoothness Q_{smooth} and local fitness Q_{fit} as shown below:

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \mathcal{Q} = \arg\min_{\mathbf{F}} \{Q_{smooth} + Q_{fit}\} \tag{2}$$

For instance, in [14], a random walk was defined on $\mathcal{G}$ with transition probabilities p and stationary distribution π, both of which can be derived from the above graph setting [15]. Thus, the classification function can be obtained by solving the following optimization problem

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \left\{ \sum_{\mathbf{x}_i \in \mathcal{X}, \mathbf{x}_j \in \mathcal{X}} \pi(\mathbf{x}_i) p(\mathbf{x}_i, \mathbf{x}_j) (\mathbf{F}(\mathbf{x}_i) - \mathbf{F}(\mathbf{x}_j))^2 + \mu \sum_{\mathbf{x}_i \in \mathcal{X}} \pi(\mathbf{x}_i) (\mathbf{F}(\mathbf{x}_i) - y_i)^2 \right\} \tag{3}$$

where $p(\mathbf{x}_i, \mathbf{x}_j)$ is the transition probability from node $\mathbf{x}_i$ to $\mathbf{x}_j$. In the above formulation, the first term enforces function $\mathbf{F}$ to smoothly change in the densely connected subgraph. The second term is so called local fitness, which measures how close the predicted values are to the given labels.

TAG implements several additional novel ideas to improve the quality of label propagation results. First, the optimization process can be decomposed into a series of parallel problems since the cost function can be formulated as component terms that only depend on individual columns of the matrix $\mathbf{F}$. Because each column of $\mathbf{F}$ encodes the label information of each individual class, such a decomposition reveals that biases may arise if the input labels are disproportionately imbalanced. Therefore, conventional propagation algorithms often fail in this unbalanced case as the results tend to be biased towards the dominant class. To overcome this problem, we apply a novel graph regularization method to effectively address the class imbalance issue. Specifically, each class is assigned an equal amount of weights and each member of a class is assigned a weight proportionally to its connection density and inversely proportional to the number of samples sharing the same class.

In addition, we use an iterative optimization method to improve the label propagation accuracy. During each iteration of the process, the most informative label is automatically selected and its class label is automatically predicted. The added label sample is then added to the existing labeled pool and the optimal predicted labels for all of the rest of the unlabeled data are then computed. Such techniques improve the quality of the label propagation results by avoiding an over aggressive step of propagating information from a very small number of labeled samples to a much larger set of samples in one single step. Instead, it implements a judicious procedure to predict new labels incrementally, starting from the most informative ones.

Finally, the TAG system includes a novel incremental learning method that allows addition of new labeled samples efficiently. Each time when user label more data, the results can be quickly updated using a superposition process without repeating the computation associated with the labeled samples already used in the previous iterations of propagation. Contributions from the new labels can be easily added to update the final prediction results. Such incremental learning capabilities are important for achieving real-time responses in users interaction. Figure 2 describes the conceptual flow of the label propagation process.

5 User Interfaces and Functionalities

As shown in Figure 3, the GUI of TAG includes the following components.

1. Image browsing area as shown in the upper left corner of the GUI, allows users to browse and label images and provide feedback. During the incremental annotation procedure, the image browsing area presents the top ranked images from left to right and from top to bottom.

2. System status bar is located in the bottom left area. It shows information about machine learning model used, the status of current propagation process, etc.. The system processing status is shown as '*Ready*', '*Updating*' or '*Re-ranking*', and so on.

3. The top right area show the name of current target class, e.g., "state of liberty" as shown in Figure 3. Note for semantic targets that do not have prior definition, this field may be left blank or use a general name such as "target of interest".

4. Annotation function area is below the area of target name. User can choose from the label of '*Positive*', '*Negative*', and '*Unlabeled*'. Also the statistical information, like the number of positive, negative and unlabeled samples are shown. The function button includes '*Next Page*', '*Previous Page*', '*Model Update*', '*Clear Annotation*', and '*System Info*'.

Input: image set $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, labeled sample $Y_l = \{\mathbf{y}_1, \cdots, \mathbf{y}_l\}$, class $\mathcal{L} = \{1, \cdots, c\}$.

1. Graph Construction:
   Calculate the graph elements, such as weight matrix $W = \{w_{ij}\}$, node degree matrix D;
2. Initialization:
   Calculate the optimal propagated function: $\mathbf{F} = \arg\min \mathcal{Q}$;
3. Obtain initial labels or scores over a subset of nodes in the graph via the Interactive Mode or Automatic Mode. Compute the new classification function $\hat{\mathbf{F}}$ using the initial subset of assigned labels or scores.
4. Compute the weights for graph superposition λ, γ;
5. Update the classification function:

   $\mathbf{F}^{new} = \lambda\mathbf{F} + \gamma\hat{\mathbf{F}}$;
6. Update image labels or likelihood scores and arrange the optimal presentation order for image browsing.
7. Repeat step 3-6 or output final retrieval/annotation results.

Output: New labels or prediction scores for data those are not labeled initially; refined labels for data that are labeled in advance; ranking order of data.

Figure 2: The conceptual flow chart of the label propagation process used in Columbia TAG system.

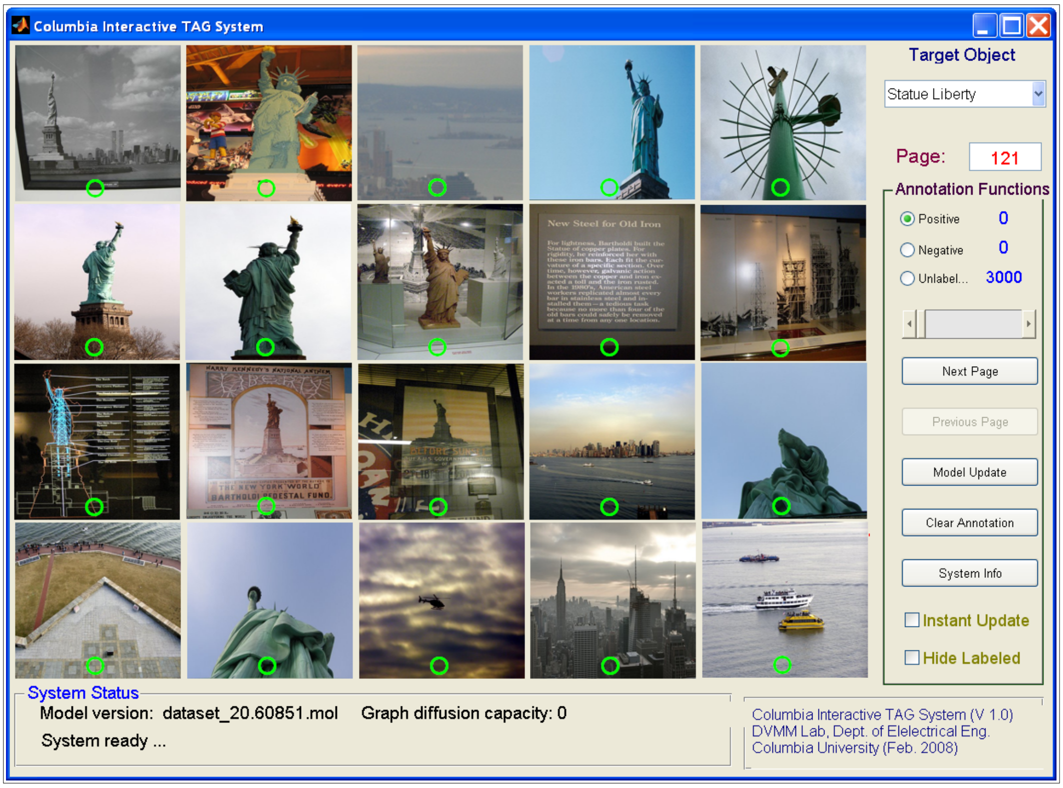
The corresponding functions of the above components are as follows.

Image browsing functions: After reviewing the current ranking results or initial ranking, user can browse additional images image by clicking the buttons '*Next Page*' and '*Previous Page*'. Users may also use the sliding bar to move through more pages at once.

Manual annotation functions: First user need to select the name of target for annotation. Then user can annotate a certain image to be positive or negative by clicking on the images. For ambiguous labeled images, user can change it back to be unlabeled samples. The positive images are with check mark ✓ and negative images with cross mark ×, and unlabeled images is marked as circle ○.

Automatic propagation functions: After user inputs some labels, clicking the button '*Model Update*' will trigger the label propagation process and the system will automatically infer the labels and generate a refined ranking score for each image. User can reset the system to the inital status by clicking the button '*Clear Annotation*'. Clicking the button '*System Info*' will generate the system information, and output the ranking results in MATLAB formate.

There are two auxiliary functions, which are controlled by checking boxes '*Instant Update*' and '*Hide Labels*'. When user select '*Instant Update*', the system will respond to each individual labeling operation and instantly update the ranking list. User can hide the labeled images and only show the ranking results of unlabeled images by checking *Hide Labels*.



Target Object: Statue Liberty

Page: 121

● Positive 0
○ Negative 0
○ Unlabel... 3000

□ Instant Update
□ Hide Labeled

Model version: dataset_20.60851.mol Graph diffusion capacity: 0
System ready ...

Columbia Interactive TAG System (V 1.0)
DVMM Lab, Dept. of Elelectrical Eng.
Columbia University (Feb. 2008)

Figure 3: The graphic user interface (GUI) of Columbia TAG System. Images shown in this example are from image sharing website flickr.com using a text search "*statue of liberty*". Original sites containing these images are listed in Appendix A.

6 Sample Applications

We have used the TAG system in several applications, such as biomedical microcopy image analysis [16], satellite image annotation, and TRECVID video concept annotation [17]. Here, we present a case study in searching images downloaded from Internet photo sharing site **Flickr**. In this application, users are given a collection of images that have been filtered using keywords, and would like to quickly retrieve images of a specific class (for example *Statue of Liberty*) through interactive browsing and relevance feedback. We assume that no prior defined recognition models have been trained to simulate the scenarios in which users may change their targets of interest dynamically depending on the contexts and tasks. Using the TAG system, users are able to quickly zero in on the images matching their specific interest by browsing and annotating returned results as positive (relevant to target) or negative (irrelevant to target). The TAG system uses the label propagation method described earlier to infer likelihood scores for each image in the collection indicating whether the image contains the desired target. User can repeat the procedure of labeling and propagation to refine the results until the output results satisfy the requirements.

6.1 Data

For this proof-of-concept experiment, we acquired an image collection from **Flickr** first by a a simple text search "Statue of Liberty". Images on such web sharing sites usually are already associated with certain textual tags, assigned by users who upload the images. However, it has been well recognized that such manually assigned tags are inaccurate - the error rate could be as high as 50%. Such discrepancy may be due to the ambiguity of labels or lack of control of the labeling process. Here, in this experiment, we show how TAG system can be used to quickly refine the accuracy of the labels. In the specific experiment, we use "*statue of liberty*" as the keyword and download the top 3000 returned images, which are fed as input to the TAG label propagation system. As shown in Figure 3, many of the initial returned images from **Flickr** are not correct - the visual content in the images does not actually show the scene or object of the statue. Using the initial 3000 images from **Flickr**, we extract features and construct a TAG graph. Users then interact with the TAG system to browse images and provide relevance labels for a few images, from which additional images showing the object/scene of the statue are predicted and retrieved.

6.2 Features and Graph

Each candidate image is processed to extract features, such as wavelet-based features and Zernike moments. The former is a popular feature used in image indexing and retrieval. It has been shown effective in approximate the texture property of objects, such as smoothness and structure. Different forms of texture features are shown in Table 1. Additionally, we include Zernike moments as the region descriptor. Moreover, we applied the soft weight version of bag of visual word (BoW) feature, which has been shown effective in improving robustness of object and scene recognition [18]. Note TAG system is scalable in terms of feature representation. Therefore, other application specified features can also be utilized to improve the graph propagation.

With the extracted image features, we can compute the pair wise affinity between samples to construct the undirected and weighted graph. The procedure of building a efficient and robust graph is the key part of the graph based methods. Most researchers prefer to use RBF kernel matrix [9, 23, 24, 25]. However, the determination of the kernel size δ is not learnable if the labeled data set is small. The previous paper has show that the propagation results highly dependents on the kernel parameter selection. The are some way to improve the graph construction, such as local scalling [26] and adaptive kernel size selection [25]. However, this fixed size of kernel is not feasible to real data since the samples may not be sampled evenly and uniformly. Since, in our experiments on graph construction, we use an adaptive kernel size approach to compute node affinity, similar to [25].

$$w_{ij} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{(mean\{knnd(\mathbf{x}_i), knnd(\mathbf{x}_j)\})^2}\} \tag{4}$$

Table 1: Image features used in satellite image retrieval.

Features	Dimensionality	Specification
Gabor wavelet [19]	70-d	Five scales and seven orientations. The mean and standard variance of Gabor wavelet coefficients in each are used.
CDF97 wavelet [20]	15-d	Three levels wavelet transformation. Minimum value, maximum value, mean value, the median value, and the standard derivation are computed from each band.
Zernike moments [21]	49-d	The degree is set as 12, the magnitude values of output complex moments are used.
Haralick co-occurrence [22]	13-d	angular second moment, contrast, correlation, sum of squares, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, information measures of correlation, and maximal correlation coefficient.

where $mean\{knnd(\mathbf{x}_i), knnd(\mathbf{x}_j)\}$ represents the mean distance of the K-Nearest Neighbor distance of the sample $\mathbf{x}_i, \mathbf{x}_j$.

6.3 Results

In this experiment, we use a data set that includes 3000 **Flickr** images in response to a search keyword ”*statue of liberty*”. We formulated the problem as a two-class problem - distinguishing images containing Statue of Liberty from those not containing Statue of Liberty. We use TAG to label a small number of samples and then conduct graph-based label propagation to predict the likelihoods and re-rank the images in the database. We evaluate the performance of the TAG system by measuring the accuracy of the first page of results (precision of the top 20 returned images). Specifically, we compute the top-20 precision (average over multiple runs) and evaluate the influence of the number of labeled images on the accuracy. We will demonstrate that a very high precision can be achieved by using the TAG system even only a very small number of labels are given by users.

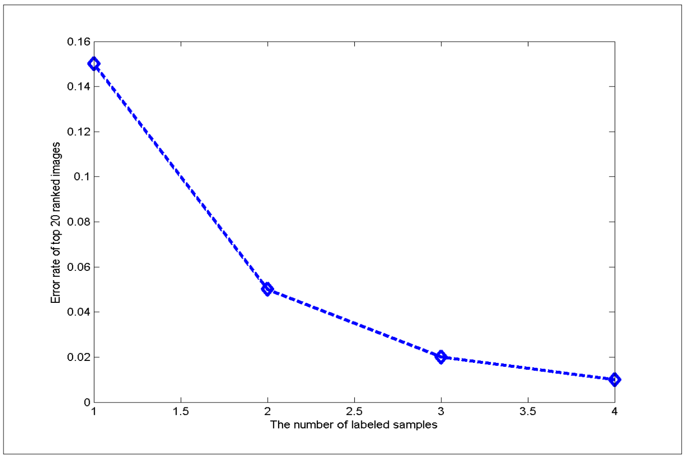


Figure 4: The top-20 accuracy of TAG label propagation results with different number of manual labels.

Figure 4 shows the performance curve of the TAG system for the experiment on the statue of liberty dataset. The horizontal axis is the number of manually labeled samples and the vertical axis is the error rate among the top 20 ranked images. As shown in this figure, TAG system demonstrates very good performance (error rate as low as 1%) with only 4 labels on the average. In Figure 5, we show samples of the TAG retrieval results (top 20 images) with only one label given by the user. Two different scenarios are shown - one targeting at the far view of the location and the other focusing on the near view. Excellent accuracies are achieved, 95% for the far-view statue of liberty and 100% for the near view. These confirm the effectiveness of using the TAG system to rapidly refine the search results and retrieve relevant images from large collections for any arbitrary targets of interest, without depending on pre-defined target classes and time consuming labeling and learning processes.

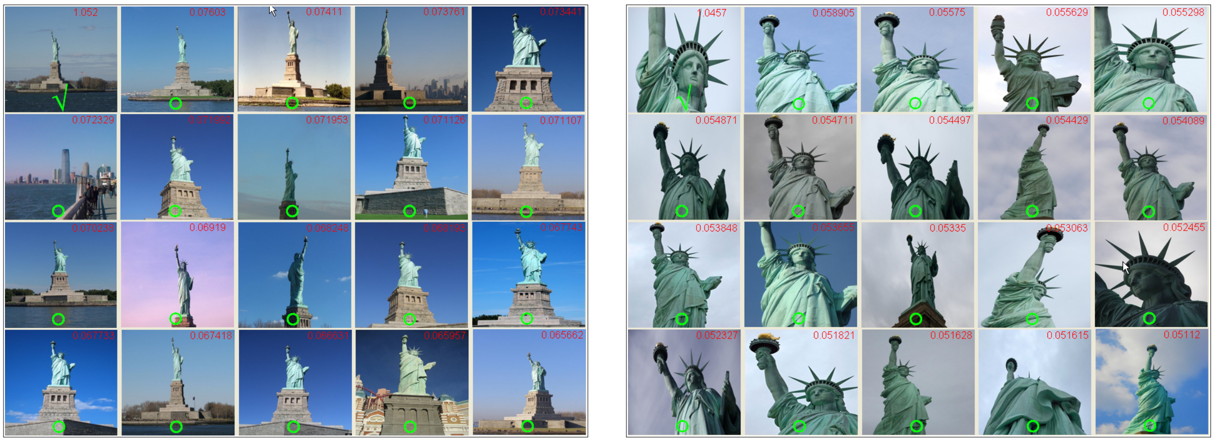


Figure 5: Performance evaluation of TAG system, measured in terms of the accuracy among the top 20 ranked images after TAG label propagation. The left and the right figure show the TAG propagation results of far-view and close-view of *statue of liberty*, respectively. With only one manual label by user, TAG successfully propagates the labels to correct samples with 95% and 100% accuracy.

7 Conclusion

Columbia TAG system version 1.0 implements novel graph-based label propagation methods for rapid searching of images and videos that match user interest related to either predefined categories or arbitrary targets without prior definition. Its unique features include

- a new framework for real-time interactive image search and label propagation;
- novel graph-based transductive learning methods for solving the challenging issues, such as small labeled data set, unbalanced class sizes, noisy locations of labeled data, and unreliable labels;
- a label regularizing method to handle unbalanced label class sizes;

. a superposable graph transduction method for decomposing the label propagation process into subprocesses, each of which involves only a single or subset of label inputs;

. a novel graph transduction method that jointly optimize the binary labels and predicted scores via an alternating cost minimization process;

. implementation of the above label propagation algorithms in an interactive image retrieval system, in which user labels are used as input for propagation in each iteration;

. implementation of the above label propagation algorithms in a fully automatic label refinement system without user interaction.

References

[1] J. Wang, S.-F. Chang, X. Zhou, and T. C. S. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA, June 2008.

[2] J. Wang, T. Jebara, and S.-F. Chang, "Graph transduction via alternating minimization," in *International Conference on Machine Learning (ICML)*, Helsinki, Finland, July 2008.

[3] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, 1998.

[4] J. Peng, B. Bhanu, and S. Qing, "Probabilistic feature relevance learning for content-based image retrieval," *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 150–164, 1999.

[5] S. Hoi, W. Liu, and S. Chang, "Semi-Supervised Distance Metric Learning for Collaborative Image Retrieval," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–7.

[6] D. Cohn, Z. Ghahramani, and M. Jordan, "Active Learning with Statistical Models," *Arxiv preprint cs.AI/9603104*, 1996.

[7] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM New York, NY, USA, 2001, pp. 107–118.

[8] T. Huang, C. Dagli, S. Rajaram, E. Chang, M. Mandel, G. Poliner, and D. Ellis, "Active Learning for Interactive Multimedia Retrieval," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 648–667, 2008.

[9] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Proc. NIPS*, vol. 16, 2004, pp. 321–328.

[10] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. 20th ICML*, 2003.

[11] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th ICML*, 2001, pp. 19–26.

[12] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," *Proc. of ICML*, 2005.

[13] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[14] D. Zhou and C. Burges, "Spectral clustering and transductive learning with multiple views," *Proceedings of the 24th international conference on Machine learning*, pp. 1159–1166, 2007.

[15] A. Agarwal and S. Chakrabarti, "Learning random walks to rank nodes in graphs," *Proceedings of the 24th international conference on Machine learning*, pp. 9–16, 2007.

[16] J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon, and S. T.C. Wong, "Cellular Phenotype Recognition for High-Content RNAi Genome-Wide Screening," *Journal of Biomolecular Screening*, vol. 13, no. 1, pp. 29–39, February 2008.

[17] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. New York, NY, USA: ACM Press, 2006, pp. 321–330.

[18] Y. Jiang, C. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," *Proceedings of the 6th ACM international conference on Image and video retrieval*, pp. 494–501, 2007.

[19] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, 1996.

[20] A. COHEN, I. DAUBECHES, and J. FEAUVEAU, "Biorthogonal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 45, no. 5, pp. 485–560, 1992.

[21] F. Zernike, "Beugungstheorie des schneidencerfarhens undseiner verbesserten form, der phasenkontrastmethode," *Physica*, vol. 1, pp. 689–704, 1934.

[22] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, pp. 610–620, 1973.

[23] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Proc. NIPS*, vol. 14, no. 2, pp. 849–856, 2001.

[24] O. Chapelle, J. Weston, and B. Scholkopf, "Cluster kernels for semi-supervised learning," *Proc. NIPS*, vol. 15, p. 1, 2003.

[25] M. Hein and M. Maier, "Manifold denoising," *Proc. NIPS*, vol. 19, 2006.

[26] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Proc. NIPS*, vol. 17, pp. 1601–1608, 2004.

Appendix

A: URLs Images Shown in Figure 3

The following lists show the URLs for the original sites that contain the images used in Figure 3.
http://static.flickr.com/2090/2237223845_0a311c07bc.jpg
http://static.flickr.com/2339/2237286622_f4bc4716fc.jpg
http://static.flickr.com/2268/2237291924_62e9df7635.jpg
http://static.flickr.com/2282/2237338327_110db2b1da.jpg
http://static.flickr.com/2246/2237361496_0ae6444e54.jpg
http://static.flickr.com/2185/2237518147_9c6e0cf6a3.jpg
http://static.flickr.com/2316/2237519015_8d1d953a8c.jpg
http://static.flickr.com/2208/2237521687_5aed81e73e.jpg
http://static.flickr.com/2416/2237523679_9c0c147c04.jpg
http://static.flickr.com/2221/2237524271_550cdd912f.jpg
http://static.flickr.com/2300/2237526591_5096ede42b.jpg
http://static.flickr.com/2364/2237527105_c25b5ef8fb.jpg
http://static.flickr.com/2262/2237527523_24b0aacc89.jpg
http://static.flickr.com/2028/2237531579_01f3f0e8ff.jpg
http://static.flickr.com/2065/2237534331_b43961f683.jpg
http://static.flickr.com/2227/2237536173_8d400ee4ec.jpg
http://static.flickr.com/2215/2237536589_1218b8218e.jpg
http://static.flickr.com/2389/2237537601_445cc96367.jpg

http://static.flickr.com/2236/2237538403_4e87b7e60b.jpg
http://static.flickr.com/2298/2237538691_c0d20a4704.jpg

B: URLs Images Shown in Figure 5

The following lists show the URLs for the original sites that contain the images used in Figure 5.
http://static.flickr.com/2264/2183540775_cb2894245e.jpg
http://static.flickr.com/2001/2206183455_fbf6a19dfd.jpg
http://static.flickr.com/2171/2188056437_2186fba2f3.jpg
http://static.flickr.com/2168/2217667001_2a421e5baf.jpg
http://static.flickr.com/2191/2188124861_4ff5d5d624.jpg
http://static.flickr.com/2348/2227613568_bf3f297835.jpg
http://static.flickr.com/2120/2195091833_3643af5d4f.jpg
http://static.flickr.com/2140/2224270947_c2ab751e45.jpg
http://static.flickr.com/2170/2242282761_cc053236fc.jpg
http://static.flickr.com/2416/2195080775_5d148fb028.jpg
http://static.flickr.com/2238/2206184065_a0676b71f1.jpg
http://static.flickr.com/2381/2204229091_fa7b13d862.jpg
http://static.flickr.com/2116/2207492068_d8aa23de7b.jpg
http://static.flickr.com/2066/2195091005_ce85a49167.jpg
http://static.flickr.com/2198/2216745106_69e688c6ab.jpg
http://static.flickr.com/2181/2206888865_0cc79307f8.jpg
http://static.flickr.com/2208/2190191607_10f70b6966.jpg
http://static.flickr.com/2134/2236067579_80443ed1b1.jpg
http://static.flickr.com/2349/2185050854_288174456c.jpg
http://static.flickr.com/2396/2195866300_b29b96a6e0.jpg
http://static.flickr.com/2373/2233492235_8468ae13e3.jpg
http://static.flickr.com/2353/2220879580_e639e5496c.jpg
http://static.flickr.com/2081/2229662870_8c49933789.jpg
http://static.flickr.com/2289/2186766373_d249b5bab9.jpg
http://static.flickr.com/2038/2238352056_2eaf7f2560.jpg
http://static.flickr.com/2074/2221579246_bd61fd08b2.jpg
http://static.flickr.com/2182/2233494291_a3c44acd37.jpg
http://static.flickr.com/2262/2225087732_5a42b043cb.jpg
http://static.flickr.com/2010/2233485809_c857482cb1.jpg
http://static.flickr.com/2379/2204107443_61fde5abcb.jpg