10-2013

Annotation for free: Video tagging by mining user search behavior

Yao TING

Tao MEI

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Shipeng LI

Citation

Annotation for Free: Video Tagging by Mining User Search Behavior [*]

Ting Yao [†], Tao Mei [‡], Chong-Wah Ngo [†], Shipeng Li [‡]
[†] City University of Hong Kong, Kowloon, Hong Kong
[‡] Microsoft Research Asia, Beijing, P. R. China
tingyao.ustc@gmail.com; {tmei, shipeng.li}@microsoft.com; cscwngo@cityu.edu.hk

ABSTRACT

The problem of tagging is mostly considered from the perspectives of machine learning and data-driven philosophy. A fundamental issue that underlies the success of these approaches is the visual similarity, ranging from the nearest neighbor search to manifold learning, to identify similar instances of an example for tag completion. The need to searching for millions of visual examples in high-dimensional feature space, however, makes the task computationally expensive. Moreover, the results can suffer from robustness problem, when the underlying data, such as online videos, are rich of semantics and the similarity is difficult to be learnt from low-level features. This paper studies the exploration of user searching behavior through click-through data, which is largely available and freely accessible by search engines, for learning video relationship and applying the relationship for economic way of annotating online videos. We demonstrated that, by a simple approach using co-click statistics, promising results were obtained in contrast to feature-based similarity measurement.

Furthermore, considering the long tail effect that few videos dominate most clicks, a new method based on polynomial semantic indexing is proposed to learn a latent space for alleviating the sparsity problem of click-through data. The proposed approaches are then applied for three major tasks in tagging: tag assignment, ranking, and enrichment. On a bipartite graph constructed from click-through data with over 15 million queries and 20 million video URL clicks, we showed that annotation can be performed for free with competitive performance and minimum computing resource, representing a new and promising paradigm for video tagging in addition to machine learning and data-driven methodologies.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

General Terms

Algorithms, Performance, Experimentation.

Keywords

Video Tagging, Tag Assignment, Tag Ranking, Tag Enrichment, Annotation, Video Search, Click-through Data.

1. INTRODUCTION

The advent of video sharing sites and rapid development of video technologies have led to the unprecedented Internet delivery of video content. Millions of users are searching, browsing, and sharing online videos as a source of information and entertainment daily. While the research on content-based video retrieval has been studied for decades, most commercial video search engines to date still heavily rely on user-provided tags for search. Therefore, tagging, or in another word, annotation, is the key not only to video search, but also to many applications such as video browsing and recommendation.

However, manually annotating video content is an intellectually expensive and time-consuming process. As a result, the tags provided by human subjects are often noisy, incomplete, subjective (biased), and sparse. This is particularly true for video, as video is a sequence of images with large content variance and complexity. Automatic annotation of videos with relevant and complete tags has attracted extensive research attentions in recent years.

The research on video annotation (tagging) has proceeded along two dimensions, i.e., model-based [8][14][23] and data driven [27][30] approaches. The model-based method always relies on pre-trained classifiers, while data driven approach aims to exploit video similarity for annotation. These approaches highly rely on computing the pair-wise video similarity in a high dimensional feature or semantic space. However, compared with the image, a video is typically associated with more complicated semantics. Furthermore, it is difficult to represent a video sequence using simple visual or aural features. As a result, the video similarity (or distance) computed in these spaces is usually not robust and computationally expensive, limiting the capacity of the existing approaches in scaling up to real data.

On the other hand, popular video search engines provide rich connection between users' search intent and video content. We are investigating in this paper if users' searching behavior can be exploited for measuring video similarity and
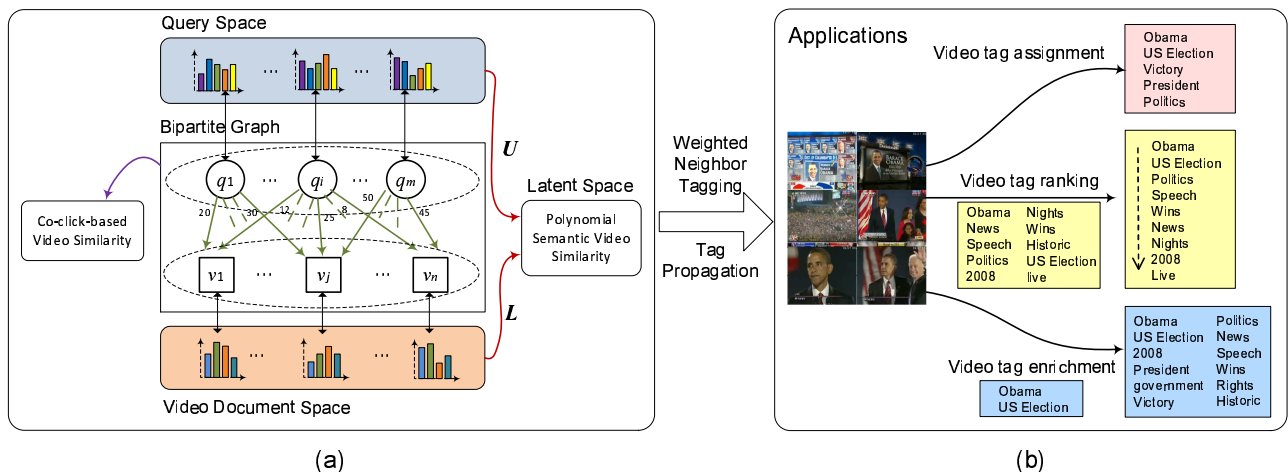
Figure 1: Unified video tagging framework. (a) Relationships mining between videos: the click-through bipartite enclosed in the rectangle is extracted from video search logs. Co-click-based video similarity captures video relationship only on the structure of the bipartite graph. Polynomial semantic video similarity combines video document features and click-through bipartite to learn two mappings. With this, video document features are projected into a latent space and then the dot product in the latent space is taken as the similarity function. (b) Based on the two proposed video tagging methods, i.e., weighted neighbor tagging and tag propagation, various applications on video tag analysis can be formulated within a unified framework. For better viewing, please see original color pdf file.

in turn resulting in scalable and efficient video tagging with million scale. This provides a new paradigm for video tagging by leveraging the "latent" and "crowdsourcing" human intelligence which is the underlying user click-through. As we all know that manual tagging of videos is expensive, but the search logs are not that expensive to obtain.

Using the implicit feedback from search engine users has been studied in the research, though not in the domain of web video. These efforts include web document search by query log analysis [6][15], query suggestion [17][19], query clustering [3][31], user behavior understanding [4][11][24] and image search [7][13]. Inspired by the successful use of click-through data in these domains, in this paper, we consider exploring and exploiting information from both click-through data and video document features to shed some light on video and video similarities measurement, and furthermore, demonstrating the utilization of searching behavior for three problems in video tagging. Figure 1 shows an overview of our proposed work. A bipartite graph between the user queries and videos is constructed by using the search logs of a video search engine. An edge between a query and a video is established, if the users who issue the query clicked and watched the video. The edges of the graph can capture some relations between query-query, query-video and video-video. For example, videos about "2008 US election" and "2012 US President Obama" are related to each other, since they are co-clicked with some queries such as "Obama", "election" and so on. Therefore, the basic idea of co-click is that two videos are similar if they are co-clicked by users with same queries, and thus leads to the intuition that the tags between them can be exchanged to enrich the descriptions. However, solely depending on click-through data overlooks the following challenges: on one hand, only click-through data is explored and exploited in the co-click method, thus the method can not be applied to videos with no clicks; on the other, users always looked at and clicked on the top search results for a query, which makes the click-through data with high sparsity. Therefore, in practice, the use of

co-click method is challenging due to the high sparsity of click-through data.

To address the above issues, we leverage both the click-through data and the video document features to learn video similarities, which fully explores the relations between videos. Specifically, we use two mappings to project queries and video documents in a click-through bipartite into a same latent space. The two mappings are supposed to be different since the query distribution is very different to the video document distribution. We formalize the two mappings learning as an optimization problem, in which the objective is to minimize the margin ranking loss of the observed query-video pairs on the click-through bipartite. The dot product in the latent space is taken as the similarity function between video and video, and the tags of one video will be assigned to its similar videos as extra metadata. It is worth noticing that although we only utilize the video-video similarities for the video tagging problem, the learned query-query similarities and query-document similarities can be equivalently applied to any kind of related applications such as query suggestion and video search.

In summary, this paper makes the following contributions:

- We study the problem of video tagging by leveraging user click-through data. To the best of knowledge, this paper represents the first effort towards this target in the multimedia research community.

- We propose two video similarity measures, i.e., co-click-based and polynomial semantic video similarity. The former relies on the structure of click-through bipartite, while the later learns the similarity in a latent space based on the click-through data.

- The proposed approach can solve video tag assignment, tag ranking, and tag enrichment problems in a single framework. We evaluate the proposed video tagging approach on a large scale of user click-through data collected from a commercial video search engine.

978

The remaining sections are organized as follows. Section 2 describes the related work. Section 3 presents the video similarity measures including co-click-based and polynomial semantic methods, while Section 4 formulates the problem of video tagging over the discovered video-video similarities. In Section 5, we analyze the click-through data generated from large-scale query logs. Section 6 provides empirical evaluations, followed by the discussion and conclusions in Section 7.

2. RELATED WORK

We briefly group the related works into two categories: video annotation and search by using click data. The former draws upon research in automatically assigning tags (annotations) to a video sequence, and the later investigates Web search and mining by interpreting the click-through data.

2.1 Video Annotation (Tagging)

Video annotation has received intensive research attention since the early of year 2000 [28], and is probably the "hottest" topic with a large number of published papers in the area of multimedia. The research in this direction has proceeded along two different dimensions: model-based methods [8][14][23] and data driven approaches [27][30].

Model-based methods assume that a training set of videos along with keyword annotations is provided for developing concept classifiers. Cristianini et al. employed SVM with one-against-the-other strategy to learn a set of detectors, each of which independently models the presence/absence of a certain concept in [8]. Later in [14], Jiang et al. used a Context Based Concept Fusion-based learning method. Users are involved in their approach to annotate a few concepts for extra videos, and these manual annotations were then utilized to help infer and improve detections of other concepts. In [23], Qi et al. proposed correlative multilabel method to simultaneously model both the individual concepts and their correlations in a unifying formulation and the principle of least commitment was obeyed.

Different from model-based methods, data driven approaches construct video similarity for annotation. In [21], tag propagation technique is developed by crawling tags of similar videos for annotation by using text and visual features in a graph reinforcement framework. In [27], the overlapping or duplicated content of videos was exploited for measuring video similarity. With this, tags associated with similar videos were exchanged for generating new tag assignments. In another work by Wang [30], both distance between samples and the difference of their surrounding neighborhood sample distributions were taken into account for video similarity estimation.

2.2 Search by Using Click Data

Click-through data has been studied and analyzed widely with different Web mining techniques for improving search engines' efficacy and usability in recent years. The use of the click-through data for query clustering was suggested by Befferman and Berger [3], who proposed an agglomerative clustering technique to identify related queries and Web pages. Wen et al. [31] combined query content information and click-through information and applied a density-based method to cluster queries. The click-through data has been studied for query expansion in the past [9]. Mei et al. [20] proposed an approach to query suggestion by computing the hitting time on a click graph. Li et al. [18] presented the use of click graphs in improving query intent classifiers.

There are also several approaches that have tried to model the representation of queries or documents on the click-through bipartite. In [1], the authors introduced another vectorial representation for the queries without considering the content information. Queries were represented as points in a high dimensional space, where each dimension corresponds to a unique URL. The weight assigned to each dimension was equal to the click frequency. This is one of the traditional click frequency models. Poblete et al. [22] proposed the query-set document model by mining frequent query patterns to represent documents rather than the content information of the documents.

In addition, click-through data has also been used to learn the rank function [15]. Joachims et al. [16] observed the relationship between clicked links and the relevance of the target pages by an eye tracking experiment. Radlinski et al. [25] concluded that the click-through data is not reliable for obtaining absolute relevance judgements, and is also affected by the retrieval quality of the underlying system. For image search, click-through data has been found to be very reliable [7][13]. In [7], Craswell et al. built a query-image click graph and performed backward random walks to determine a probability distribution over images conditioned on the given query which can be used for ranking. Later in [13], Jain et al. reranked the image search results so as to promote images that are likely to be clicked to the top of the ranked list. In another work by Trevisiol et al. [29], an in-depth analysis of several ranking algorithms was performed on Flickr user log data to investigate the importance of many factors, including internal and external image popularity, the overall attentions, diversity, semantic categories and visual appearance. However, to the best of our knowledge, no work leveraging click data is proposed for video search domain.

3. LEARNING VIDEO RELATIONSHIP FROM CLICK-THROUGH

In this section, we first define the bipartite graph that naturally comes from user actions in the query log, followed by the co-click method to mine video similarity on the structure of the click-through bipartite graph. Then, a polynomial semantic video similarity algorithm is proposed to solve the issues of click-through data incompleteness and no-click data for new coming queries (videos) by combining the bipartite graph and video features.

3.1 Notation

Let $\mathscr{G} = (\mathcal{V}, \mathcal{E})$ denote a click-through bipartite as shown in the left part of Figure 2. $\mathcal{V} = Q \cup V$ is the set of vertices, which consists of a query set $Q = \{q_k\}_{k=1}^{m} \subset \mathbb{R}^D$ and a video set $V = \{v_k\}_{k=1}^{n} \subset \mathbb{R}^D$, and D is the feature dimension. $\mathcal{E}$ is the set of edges between query vertices and video vertices. In addition, there exists rich metadata on the vertices of the click-through bipartite and the metadata consists of queries (videos) content features.

3.2 Co-click-based Video Similarity

Given the click-through bipartite graph $\mathscr{G}$, we can define a co-click matrix from the structure of the bipartite graph. The element of the matrix stands for the number of queries
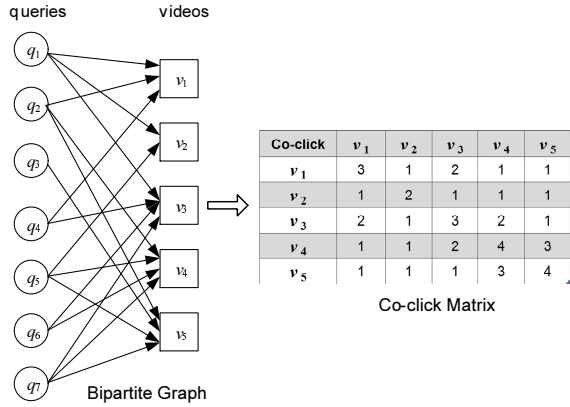
Figure 2: Co-click Method. A click-through bipartite with seven queries and five videos is illustrated and the co-click matrix on this bipartite is given accordingly.

with which the two video pages are both clicked. The co-click value of a video with itself is the number of clicks received by the video from all the queries. Figure 2 shows an example of one bipartite graph and its co-click matrix. For simplicity, we assume that each video is clicked by queries only once in Figure 2. Take v_1 as an example, the video is clicked by q_1, q_2 and q_4, thus the co-click number of v_1 with itself will be 3. Similarly, v_1 and v_2 are both clicked only by q_1, hence the co-click value of v_1 and v_2 is 1.

Next we describe how to measure the similarity of two videos based on the co-click matrix. The co-click number of a video v_i with itself is denoted as C_i, while the co-click value between v_i and v_j is denoted as C_{ij}. With these notations, the similarity $\omega(v_i, v_j)$ between two videos v_i and v_j based on the co-click relationship can be defined as

$$\omega(v_i, v_j) = \frac{C_{ij}}{\sqrt{C_i C_j}}. \qquad (1)$$

The rationale underlying this formula is that, if two videos are clicked by mostly the same queries, the two videos are likely to be similar to each other. The similarity is offset by the number of video clicks from all the queries, which moderates the similarity score in the case when some videos received more clicks than others. These videos are generally more common than others. With the normalization, the similarity measure is in the range of $[0, 1]$.

3.3 Polynomial Semantic Video Similarity

It is easy to demonstrate that co-click method could achieve promising performance if the click-through data is complete, i.e. each query is associated with all the related documents. But unfortunately, in real-world search, the key challenge in doing so arises from the fact that the click-through data is very sparse. For many of the queries, a handful of relevant videos had received a large number of clicks but the total number of clicked videos ranged typically from ten to a hundred. Generalizing co-click method to all the videos becomes very challenging in such scenarios. Deriving from the ideas of Polynomial Semantic Indexing (PSI) [2], and related methods such as Partial Least Squares (PLS) [26] and Latent Semantic Indexing (LSI) [10], we develop a polynomial semantic video similarity (PSVS) approach to model the relations in a click-through bipartite by projecting queries and videos into a latent space. With this, the video similarities

can be calculated by the dot product of their mappings in the latent space.

Given a query q and a video v, we wish to learn two mappings, which can map query q from query space and video v from video document space into a common latent space. With this, the learned ranking function $f(q, v)$ that returns a score measuring the relevance of v given q can be better satisfied with the query-video click-through observations. Let us first consider the function $f(q, v)$ as

$$f(q, v) = \sum_{i,j=1}^{D} W_{ij} q_{(i)} v_{(j)} = q^T W v, \qquad (2)$$

where $q_{(i)}$ $(v_{(j)})$ stands for the ith (j^{th}) dimension of the feature vector and W is the learning matrix. As discussed in [2], the huge memory requirement and large amount number of parameters for W make the model training hardly be possible for realistic tasks. Thus, a low-rank approximation of W which will lead to capacity control, smaller memory and less computational cost is given by

$$\bar{W}_{ij} = (U^T L)_{ij} + I_{ij}. \qquad (3)$$

After plugging the approximation matrix into ranking function $f(q, v)$, we can derive that

$$f(q, v) = q^T (U^T L + I) v = \sum_{i=1}^{Y} (Uq)_{(i)} (Lv)_{(i)} + q^T v, \qquad (4)$$

where U and L are $Y \times D$ matrices and induce a Y-dimensional latent space.

The training of U and L could be many forms. From our click-through data, we can easily get a set of triplets $\mathcal{T}$, where each tuple (q, v^+, v^-) consists of the query q, a video v^+ with higher click and a lower clicked video v^-. Deriving from the idea of "learning to rank" [12][15], it aims to optimize U and L which makes $f(q, v^+) > f(q, v^-)$, i.e. video v^+ should be ranked higher than video v^-. The margin ranking loss [12] which has been used in several information retrieval methods [5][15] is employed and the optimization problem is defined as

$$minimize: \sum_{(q, v^+, v^-) \in \mathcal{T}} \max(0, 1 - f(q, v^+) + f(q, v^-)). \qquad (5)$$

Algorithm 1 presents the learning procedure. We train this using stochastic gradient descent [5] for our model due to its efficiency and capability of applying to highly scalable problems. The algorithm of polynomial semantic video similarity has several characteristics in the following:

- Preference Relations: the proposed polynomial semantic video similarity advocates the training using preference relations, which are mined from user implicit feedback. Roughly speaking, if video A has significantly more clicks than video B when given a query q, then users should have a preference for A. These preferences are used as supervised signals for our training.

- Low Rank: the matrices U and L are $Y \times D$ dimensions, where $Y \ll D$. Thus, it will sharply reduce the memory and capacity cost, and speed up the training procedure. That makes the algorithm highly applicable and flexible when applied to the practical Web applications.

Algorithm 1 Polynomial Semantic Video Similarity

Input:
⋆ Click-through bipartite $\mathscr{G} = (\mathcal{V}, \mathcal{E})$.
⋆ Query feature q and video feature v.
Initialization:
⋆ Generate a set of triplets (q, v^+, v^-) as labeled data based on the click-through data.
⋆ Initialize the matrices U and L using a normal distribution with mean zero and standard deviation one.
⋆ Initialize the learning rate α.
For all the triplets:
   Update the new matrices U and L:
   $U = U + \alpha L(v^+ - v^-)q^T, \quad if \quad 1 - f(q, v^+) + f(q, v^-) > 0$
   $L = L + \alpha U q(v^+ - v^-)^T, \quad if \quad 1 - f(q, v^+) + f(q, v^-) > 0$
End
Output:
Similarity functions:
Video similarity: $\forall v, v', \quad \omega(v, v') = v^T L^T L v'$
Query similarity: $\forall q, q', \quad \omega(q, q') = q^T U^T U q'$

- Similarity Measurement: matrices U and L are the two mappings from query space and video document space into a Y-dimensional latent space. These two mappings are learned on the observations of query-video click-through bipartite. When obtaining the two mappings, the relations of query-query and video-video can be modeled by projecting them into the latent space. Although only video-video relations are used for the following video tagging problem, query-video and query-query relations can be equivalently used in their related applications.

- Incremental: the complexity of polynomial semantic video similarity algorithm is linear to the number of training triplets, which makes it easy to implement and update. When the new click-through data is coming, new query video preference triplets will be firstly generated and then used to update the two matrices U and L accordingly. Moreover, early stopping can be assessed with a validation set for further reducing the computational cost.

4. VIDEO TAGGING

After we get the relationships between videos, how to use them for automatic tagging? In this section, two video tagging approaches, i.e., weighted neighbor tagging and tag propagation, will be described. The weighted neighbor tagging method once only takes immediately neighbor video into account for automatic tag assignment, while the tag propagation approach further propagates the video relationships and considers all the related videos simultaneously.

4.1 Weighted Neighbor Tagging

Let $V_r = \{v_i : 1 \leq i \leq N\}$ be the video collection where each video has one or more relationships to others and let $Tag = \{t_1, \ldots, t_k, \ldots, t_M\}$ be the set of tags initially assigned to the videos in V_r. Let $I(t_k, v_i)$ be an indicator function, with $I(t_k, v_i) = 1$ if v_i was manually tagged with tag t_k, otherwise $I(t_k, v_i) = 0$. The tag relevance score $rel(t_k, v_i)$ of a tag t_k from neighbor videos is computed as

$$rel(t_k, v_i) = \sum_{\substack{v_i, v_j \in V_r \\ t_k \in Tag}} \omega(v_i, v_j) I(t_k, v_j). \quad (6)$$

In this way, we compute a weighted sum of influences of the related videos containing tags. Then we can automatically assign new tags for each video v_i. Basically, there are two major ways of utilizing the result. The simplest way is by sorting the tags according to their relevance scores and popping top k tags as the final result. An alternative way is by producing a threshold δ for tag relevancy. Tags whose relevance scores are above δ will be set as the new tags.

4.2 Tag Propagation

So far, we have just considered the direct relationships between videos. Next, all the relationships between videos are deployed holistically in a random walk framework to better compute the tag relevancy. It is worth noticing that the objective of tag propagation is not to assign relevance values for the videos, but it is an approach for computing relevance values of a tag t_k for a given video v_i.

Denote p_{ij} as the transition probability from video v_i to video v_j, and $I(t_k, v_i)$ and $TR(t_k, v_i)$ as the initial and updated relevance scores of a tag t_k to a video v_i, respectively. The tag propagation is formulated as following

$$\begin{pmatrix} TR(t_k, v_1) \\ TR(t_k, v_2) \\ \vdots \\ TR(t_k, v_N) \end{pmatrix} = \lambda \begin{pmatrix} p_{11} & \cdots & p_{1N} \\ p_{21} & \cdots & p_{2N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \cdots & p_{NN} \end{pmatrix}^T \begin{pmatrix} TR(t_k, v_1) \\ TR(t_k, v_2) \\ \vdots \\ TR(t_k, v_N) \end{pmatrix} + (1-\lambda) \begin{pmatrix} I(t_k, v_1) \\ I(t_k, v_2) \\ \vdots \\ I(t_k, v_N) \end{pmatrix}, \quad (7)$$

where the tradeoff parameter λ ($0 \leq \lambda \leq 1$) weights the importance of the propagated and initial scores. Note that the first term in the equation represents information exchange from neighboring videos, while the second term is the original tag relevance score. The element p_{ij} is given by

$$p_{ij} = \frac{\omega_{ij}}{\sum_k \omega_{ik}}, \quad (8)$$

where ω_{ij} is the video similarity between video v_i and v_j, as discussed in Section 3.

The spirit of tag propagation is to give a higher relevance score for tag t_k to a video v_i if the video is in close proximity with other videos that are also highly relevant to the tag t_k. For each tag in the collection Tag, the above iterative propagation will be repeated. To the end, the refined relevance between tags and videos are produced and tags assigned to the video can be generated and ranked by their relevance scores.

4.3 Discussion

The proposed work can handle different practical scenarios in a unified framework. For a no-tag video, tagging can be carried out by crawling relevant tags from similar videos (tag assignment). For a video with only few tags, the coverage of tag list can be expanded by including additional tags from other similar videos not in the original tag list (tag enrichment). Finally, for videos with abundant tags, tags can be sorted according to their relevancy by getting clue from the appearance of the tags in similar videos (tag ranking). In all the tasks, tagging is for free as the click-through data, which plays the essential role in mining similar videos, are largely and freely available for access by search engines.
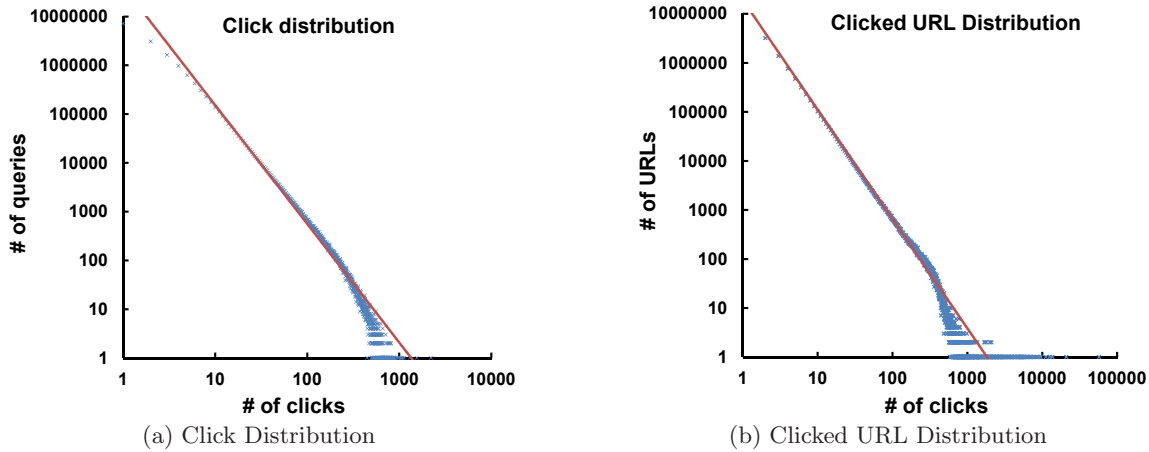
(a) Click Distribution

(b) Clicked URL Distribution

Figure 3: Query click and clicked URL distribution for the click-through data. Red lines denote the fitting power law curves.
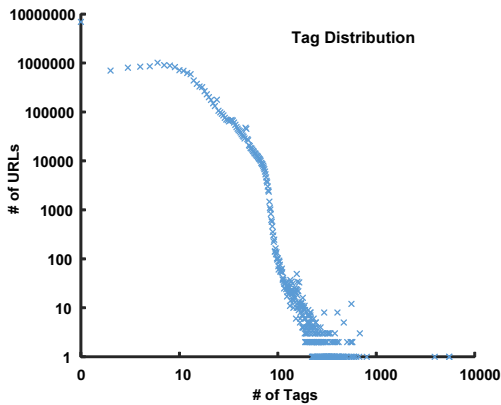

Figure 4: Tag distribution for the click-through data.

Table 1: Coefficients of Power laws of Click Distribution and Clicked URL Distribution, respectively.

Power laws: Ax^B	A	B
Click Distribution	41427273	-2.43
Clicked URL Distribution	19082457	-2.23

5. DATA ANALYSIS

We have collected one month query logs in November 2012 from a commercial video search engine. The query logs are represented as plain text files that contain a line for each HTTP request satisfied by the Web server. For each record, the following fields are used in our data collection:

$$< Query, ClickedURL, ClickCount, Title, Description, Tags >,$$

the $ClickedURL$ and $ClickCount$ represent the URL and number of clicks on this URL when user submit the $Query$, respectively. Note that $ClickCount$ is cumulated over all identical queries. $Title$, $Description$ and $Tags$ denote the corresponding textual information associated with the clicked video.

For building the click-through bipartite, we used all the queries in the log with at least one click. There are 15,697,027 queries and 21,000,433 URLs on the bipartite graph. Figure 3 gives the main characteristics of the query and URL distribution. Figure 3(a) shows the query click distribution.

Each point represents the number of queries (y axis) with a given number of clicks (x axis). The plot on Figure 3(b) shows the clicked URL distribution. Each point denotes the number of URLs with a given number of clicks. We can see that these two distributions clearly follow power laws. The observation is similar to [1], which also states that the user search behavior follows a power law. For details, the associated law is plotted in Figure 3, and the law coefficients are listed in Table 1. According to the statistics, each query has on average 4.08 clicked URLs and each URL was clicked by 3.12 times on average.

Figure 4 further shows the statistics of the video tags. Each point represents the number of videos with a given number of tags. Among all videos, more than 65% videos have 1 to 100 tags, followed by around 33% videos having no tags, and less than 0.1% videos having more than 100 tags.

6. EXPERIMENT

We conducted our experiments on the aforementioned click-through data and evaluated our approaches on video tag assignment, video tag ranking and video tag enrichment.

6.1 Experimental Settings

Compared Approaches: We compare the following approaches for performance evaluation:

- Initial tags. The original tag list associated with the video. We name this run as *Initial*. We simply treat the tag order in the original list, giving that the first tag is usually more representative than the second and so on.

- Cosine similarity. The video-video relationship is measured by the cosine similarity on the video features. Then two video tagging runs based on weighted neighbor tagging (WN) and tag propagation (TP) are performed, and named as $Cos + WN$ and $Cos + TP$ respectively.

- Partial least squares [26]. A typical method which aims to model the relations between two or more feature spaces by projecting them into a latent space. We employed the Partial Least Squares method on our
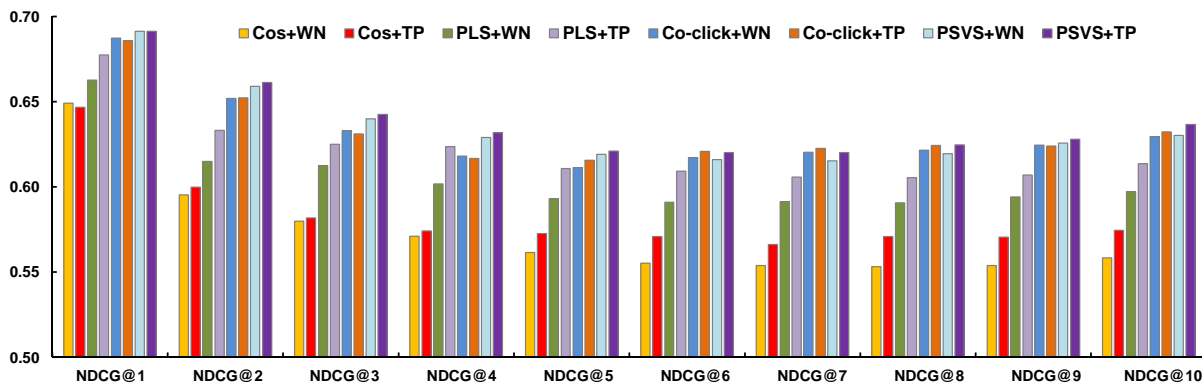
Figure 5: The NDCG of different approaches for video tag assignment.

click-through bipartite with query and video document features. We name the two runs as $PLS + WN$ and $PLS + TP$ by using the two video tagging methods, respectively.

- Co-click-based video similarity. We design two runs for co-click based video similarity described in Section 3.2: $Co-click + WN$ and $Co-click + TP$.

- Polynomial semantic video similarity. Similarly, two runs, i.e., $PSVS + WN$ and $PSVS + TP$, are experimented based on the polynomial semantic video similarity with the proposed two tagging approaches.

Parameter Setting: We take the word in queries and video URLs including title, description, and tags as features. Words are stemmed and stop words are removed. With word features, each query is represented by a $tf-idf$ vector in the query space, and each video is represented by a $tf-idf$ vector in the video document space. In our experiments, we use only the top 100,000 most frequent words. Moreover, we randomly select 2 million triplets as the training data for our polynomial semantic indexing. The learning rate α is fixed using a validation set of 0.5 million triplets and set as $\alpha = 0.06$ in the experiment. The dimension Y of latent space is empirically set to 200. For the tradeoff parameter λ in the tag propagation, we set $\lambda = 0.9$ for all the TP runs, making the similarities output by different approaches comparable.

Ground Truth: To facilitate the evaluation and comparison with other methods, we randomly selected 2,500 URLs as the test samples, in which 500 URLs without tags are evaluated for video tag assignment and the remaining 2,000 URLs with initial tags are tested in video tag ranking and enrichment tasks. For each URL, the top 20 tags in the ranked lists obtained by different approaches are all annotated during the final evaluation. We invited nine evaluators from different education backgrounds, including computer science, linguistics, physics, industry, business, and design. All evaluators are familiar with video sharing websites. Every URL-tag pair was annotated on a three point ordinal scale: 2-Highly Relevant; 1-Relevant; 0-Non-relevant. Note that obtaining just these annotations was very time-consuming. The evaluators are requested to watch the whole video before assigning labels. Whenever the relevancy judgments cannot be made by visual inspection alone, the evaluators have to read the text description and even refer to exter-

nal sources (e.g., Web pages, Wikipedia) for understanding the background behind the video.

Evaluation Metrics: For the evaluation of video tag assignment and video tag ranking, we adopted Normalized Discounted Cumulative Gain ($NDCG$) which takes into account the measure of multi-level relevancy as the performance metric. Given a tag ranked list, the $NDCG$ score at the depth of d in the ranked list is defined by:

$$NDCG@d = Z_d \sum\nolimits_{j=1}^{d} \frac{2^{r^j} - 1}{\log(1 + j)} \qquad (9)$$

where r^j represents the rating of a tag in the ground-truth, Z_d is a normalization constant such that $NDCG@d = 1$ for perfect ranking and 0 otherwise. In the evaluation of video tag enrichment, the metrics of Precision, Recall and F1-Measure are employed.

6.2 Evaluation of Video Tag Assignment

The problem of video tag assignment is to assign a set of tags to a given video for describing the video content. As described in section 6.1, 500 video URLs without tags are tested.

Figure 5 shows the $NDCG$ performances of eight runs. Note that the *Initial* run is excluded because the test video contains no tags. Overall, the results across different depths of $NDCG$ consistently indicate that using click-through data leads to a performance boost against cosine similarity, which relies on only video document features. Furthermore, $PSVS$ utilizing click-through data as relative relevance judgements also exhibits better performance than PLS, which uses the absolute click numbers for learning. The result basically indicates that click-through data does not convey absolute relevance judgements, but partial relative relevance judgements.

Comparing $PSVS$ to $Co-click$, performance improvement is observed especially in the top few ranked tags. Basically $PSVS$ has better capability in recalling relevant tags, for leveraging video document features to compensate the sparsity of click-through data. Compared to the weighted neighbor tagging, tag propagation can constantly lead to better performance gain. This somewhat reveals the weakness of linear fusion behind the weighted neighbor tagging, where the relationships between videos are considered individually. Tag propagation, in contrast, is benefited from the way of formulating the video tag assignment as a random
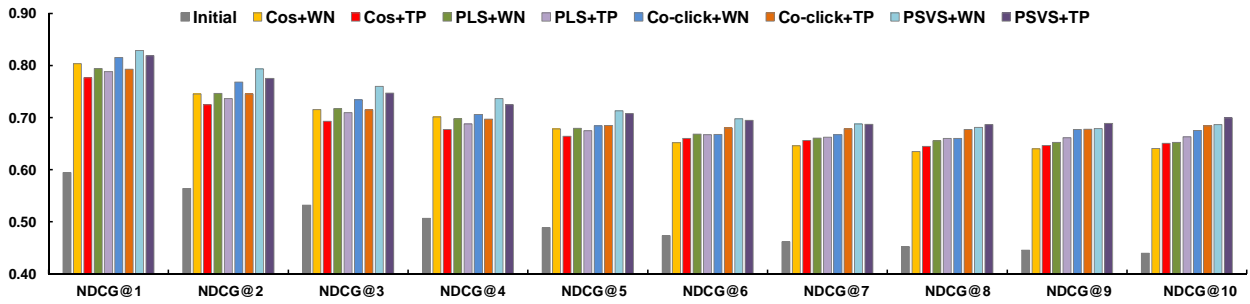
Figure 6: The NDCG of different approaches for video tag ranking.

Table 2: Quality comparison of the tags obtained by each method. Each measurement has been averaged over all evaluated URLs.

| Method	Precision	Recall	F1-Measure
Initial	0.198	0.33	0.247
Cos+WN	0.376	0.72	0.453
Cos+TP	0.40	0.778	0.484
PLS[26]+WN	0.378	0.718	0.453
PLS[26]+TP	0.407	0.782	0.49
Co-click+WN	0.399	0.764	0.479
Co-click+TP	0.415	0.803	0.50
PSVS+WN	0.403	0.766	0.484
PSVS+TP	**0.419**	**0.808**	**0.504**

Figure 7: Performance improvement of test videos returned by head and tail queries, respectively. The performance are compared against the results w.r.t their tag ranking at NDCG@10.

walk over the whole video similarity graph. Figure 8(a) lists the tags generated by different methods on two exemplary videos. $PSVS$ and $co-click$ achieve better qualitative results than others, and quantitatively $PSVS+TP$ recalls the most number of relevant tags than others.

6.3 Evaluation of Video Tag Ranking

In addition to tagging the no-tag videos, we also evaluated the performance of tag ranking on 2,000 videos with initial tags supplied by users. Based on the scores output by a run, the initial tags are ranked according to their relevancy to a video. Tag ranking is an important task that helps browsing of videos by prioritizing the tags to be displayed and indexed.

Figure 6 shows the performance of different runs. All the methods exhibit significantly better performance than the $Initial$ run, clearly showing the advantages of exploiting video relationship for tag ranking. Particularly, by building the relationship based upon click-through data that underlies the latent human feedback, the similarities of videos are more objectively measured. This basically facilitates the evaluation of tag importance as evidenced by the good performance of $PSVS$ and $Co-click$ in comparison to others. Similar to the results of video tagging, $PSVS$ shows the strongest performance followed by $Co-click$.

A different observation compared to evaluation of tag assignment, however, is that the runs based on weighted neighbor tagging exhibit better performance in ranking the top five tags than the runs based on tag propagation. One analysis shows that, when there are few tags that dominantly appear in the videos, the weighted neighbor tagging, which performs weighted voting of tags, is effective in boosting these tags into the top few positions. Tag propagation, on
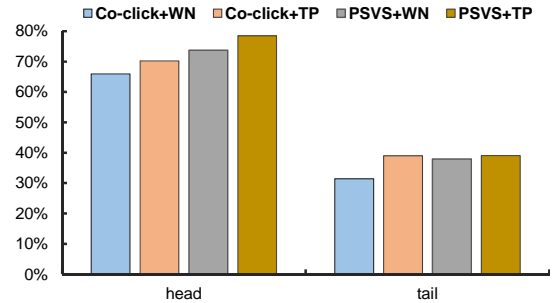
the other hand, somewhat diffuses the scores of these tags through random walk, but has a better ability in recalling more relevant tags not dominantly appear, and thus shows better performance when going deeper into the ranked list. Figure 8(b) shows the list of tags ranked by different approaches on two exemplary videos.

Furthermore, we split the test videos into two subsets, i.e., $head$ and $tail$. The former is the video set returned by head queries, which have more than 100 clicks on different URLs, while the later represents the video set returned by tail queries, which have less than 100 clicks. Figure 7 shows the degree of improvement on two sets of videos. The result indicates that improvement can be generally expected, and larger degree of improvement is attained when the videos are returned by head queries.

6.4 Evaluation of Video Tag Enrichment

We adopted Precision, Recall and F1-measure as performance metric. The evaluation requires the full set of tags that are relevant to videos. However, practically such ground truth is difficult to obtain because there are rich of words that can potentially enrich the description of a video. Instead, we adopted the pooling strategy, which evaluates the top 20 tags pooled from different runs and generate the ground truth for measure the Recall.

Table 2 lists the performance of eight different runs. Compared to the initial tags assigned to videos, more new and relevant tags are introduced. As shown in the table, by using $PSVS+TP$, the newly brought in tags enhance the coverage of the initial tags, significantly raising the Recall from 0.33 to 0.808 while improving the Precision. Comparing all runs, $PSVS$ again shows consistently better performance in both Precision and Recall.

(a) Exemplary videos with the tags generated by all the methods for video tag assignment

Video I

Keyframes of Test Video	Cos+WN	Cos+TP	PLS+WN	PLS+TP	Co-click+WN	Co-click+TP	PSVS+WN	PSVS+TP
	kidsongs	kidsongs	kidsongs	kids songs	kidsongs	kidsongs	kidsongs	kids songs
	kendincos	kids music	kendincos	kendincos	video	kendincos	kendincos	kids songs
	part	kendincos	intro	video	free videos	kids songs	kids songs	kids music
	kids music	video	outro	kids videos	online videos	kids songs	kids songs	video
	video	songs	1995	kids music	video search	kids music	kids	kendincos
	intro	free videos	kids songs	music	kids songs	free videos	songs	kids
	free videos	online videos	video	film	music	online videos	video	songs
	online videos	video search	kids	world news	film	video search	kids music	music
	video search	baby songs	1987	headlines	songs	songs	free videos	baby songs
	songs	nursery rhymes	childrens song	wn archive	part	music	online videos	free videos
	show	1990	tv	wn network	world news	part	video search	online videos
	music	part	baby songs	Newspaper	headlines	film	intro	video search
	Film	kids	film	media	wn archive	1990	baby songs	part
	1990	silly	world news	tv	wn network	world news	family	nursery rhymes
	baby songs	music	headlines			headlines	part school	film

Video II

Keyframes of Test Video	Cos+WN	Cos+TP	PLS+WN	PLS+TP	Co-click+WN	Co-click+TP	PSVS+WN	PSVS+TP
	video	spongebob	video	spongebob	spongebob	spongebob	spongebob	spongebob
	music	squarepants	music	video	squarepants	squarepants	squarepants	squarepants
	nickelodeon	video	vh1	music	video	video	video	video
	squarepants	music	nickelodeon	photos	music	music	patrick	patrick
	spongebob	nickelodeon	squarepants	audio	patrick	patrick	music	music
	funny	patrick	spongebob	square pants	photos	photos	schwammkopf	nickelodeon
	moments	funny	funny	film	patrick	audio	nickelodeon	photos
	onetruemedia	photos	moments	entertainment	nickelodeon	nickelodeon	funny	audio
	photos	audio	onetruemedia	world news	entertainment	entertainment	sandy	funny
	audio	schwammkopf	photos	headlines	film	film	online	schwammkopf
	film	film	audio	wn network	funny	funny	cartoons	film
	world news	entertainment	film	newspaper	schwammkopf	mysteries	photos	cartoons
	headlines	online	world news	journalist	cartoons	zitate	audio	entertainment
	wn network	video clip	headlines	media	world news	kurzfilme	sings	sings
	newspaper	cartoons	wn network	archives	headlines		squidward	

(b) Exemplary videos with the tags ranked by all the methods for video tag ranking

Video III

Keyframes of Test Video	Initial	Cos+WN	Cos+TP	PLS+WN	PLS+TP	Co-click+WN	Co-click+TP	PSVS+WN	PSVS+TP
	halo4	halo4	halo4	halo4	halo4	halo4	halo4	halo4	halo4
	tricks	halo	halo	halo	halo	halo	halo	halo	halo
	glitch	gameplay	xbox	xbox	xbox	gameplay	xbox	gameplay	xbox
	machinima	master chief	gameplay	gameplay	halo3	xbox360	microsoft	gameplay	gameplay
	hidden	343 industries	343	343	343	master chief	343	master chief	343
	matchmaking	microsoft	microsoft	microsoft	gameplay	microsoft	microsoft	multiplayer	multiplayer
	hiding	xbox360	multiplayer	multiplayer	multiplayer	343 industries	343 industries	microsoft	microsoft
	hlg	multiplayer	bungie	bungie	video	cortana	walkthrough	halo3	bungie
	hidden	bungie	tutorial	tutorial	elite	walkthrough	machinima	tutorial	tutorial
	league	walkthrough	halo3	tips	halo2	multiplayer	video	bungie	halo3
	easter	cortana	tips	video	tips	sniper	tips	video	video
	egg	chief	video	machinima	tutorial	master	elite	walkthrough	machinima
	multiplayer	tutorial	machinima	elite	video	chief	matchmaking	343 industries	elite
	secrets	t trailer	elite	tricks	league	trailer	tricks	tips	tricks
	myths	tips	tricks		elite	spartan		machinima	league

Video IV

Keyframes of Test Video	Initial	Cos+WN	Cos+TP	PLS+WN	PLS+TP	Co-click+WN	Co-click+TP	PSVS+WN	PSVS+TP
	naruto vs pain	naruto	naruto	naruto	naruto	naruto	naruto	naruto	naruto
	pain	anime	anime	anime	anime	animation	anime	anime	anime
	ita	sasuke	sasuke	sasuke	sasuke	sasuke	sasuke	sasuke	sasuke
	eng	shippuden	video	video	video	ninja	video	video	video
	esp	video	manga	pain	pain	sakura	opening	shippuden	manga
	sub	kakashi	pain	manga	manga	ultimate	manga	ninja	opening
	amv	sakura	opening	opening	opening	sakura	pain	manga	pain
	hd	pain	bleach	bleach	pain	video	bleach	sakura	bleach
	hq	manga	rock	piece	bleach	opening	sub	opening	rock
	hero	hinata	games	final	piece	hinata	hd	pain	games
	508	opening	naruto vs pain	rock	naruto vs pain	manga	rock	bleach	hd
	chapter	ultimate	piece	ps3	hd	animationbd	battle	rock	piece
	fith	animatioabd	sub	sub	sub	kakashi	piece	games	naruto vs pain
	final	bleach	hd	final	final	pain	ps3	hinata	sub
	episodio	naruto vs pain	final	final	rock	naruto sasuke		piece	battle

Figure 8: Examples showing the tagging results by different methods (For better viewing, please see original color pdf file). For each row, the first block shows the keyframes of a test video, followed by each block showing the top 15 tags annotated by a method. The correct tags are highlighted by yellow color.

6.5 Complexity Analysis

The complexity of our proposed $PSVS$ learning is $O(|\mathcal{T}| \times Y \times D)$, where $|\mathcal{T}|$ represents the number of the training triplets. The training of 2 million triplets in our experiment can be finished within five days on one server. More importantly, the training complexity is linear to the number of triplets, which makes the update with incremental triplets very fast. For the video tagging, take 500 videos to perform video tag assignment for example, WN and TP take less than 2.5 and 13 seconds on a regular PC (Intel dual-core 3.33GHz CPU and 8 GB RAM) to complete the whole process, respectively. In other words, tagging one video only takes 5.0 and 26 milliseconds, respectively. Clearly, the speed is fast and provides almost instant response.

7. DISCUSSION AND CONCLUSION

We have presented an economy way of video tagging, by mining the video relationship through click-through data which can be viewed as the footprints of user searching behavior. Particularly, we propose two ways of exploiting the searching behavior, by co-click and polynomial semantic similarity, where the latter addresses the sparsity problem that generally exists in video click-through. Together with two tagging methods, we demonstrated in the experiments that the proposed approaches make improvement to three tagging tasks: assignment, ranking and enrichment, by leveraging the bipartite graph constructed from tens of millions of URL clicks. Basically, utilizing click-through, which is cheap to exploit and free for search engines, as a measure

for characterizing video similarity, shows better performance than feature-based approach such as by using cosine similarity. By combining click-through and features for deriving a latent space and tackling the sparsity problem, further improvement was consistently observed in the experiments.

Our work can be enhanced by considering also partial visual near-duplicates, which frequently happen in Web videos. The visual features can be exploited together with click-through and document features for a more comprehensive manner of characterizing visual similarities. The cost, however, is the need for processing millions of videos, which is much more computationally expensive than processing the click-through data associated with videos.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] R. A. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, 2007.

[2] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, C. Cortes, and M. Mohri. Polynomial semantic indexing. In *Proceedings of Advances in Neural Information Processing Systems*, 2009.

[3] D. Beeferman and A. L. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, 2000.

[4] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *Proceedings of International World Wide Web Conference*, 2008.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of International Conference on Machine Learning*, 2005.

[6] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Proceedings of Advances in Neural Information Processing Systems*, 2007.

[7] N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of ACM conference on Research and Development in Information Retrieval*, 2007.

[8] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. *Cambridge University Press*, 2000.

[9] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proceedings of International World Wide Web Conference*, 2002.

[10] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[11] G. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of ACM conference on Research and Development in Information Retrieval*, 2008.

[12] R. Herbrich, T. Graepel, and K. Obermayer. Advances in large margin classifiers, chapter large margin rank boundaries for ordinal regression. *MIT Press, Cambridge, MA*, 2000.

[13] V. Jain and M. Varma. Learning to re-rank: Query-dependent image re-ranking using click data. In *Proceedings of International World Wide Web Conference*, 2011.

[14] W. Jiang, S.-F. Chang, and A. C. Loui. Active concetx-based concept fusion with partial user labels. In *Proceedings of International Conference on Image Processing*, 2006.

[15] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, 2002.

[16] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. on Information Systems*, 25(2), 2007.

[17] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of International World Wide Web Conference*, 2006.

[18] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *Proceedings of ACM conference on Research and Development in Information Retrieval*, 2008.

[19] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *ACM Conference of Information and Knowledge Management*, 2008.

[20] Q. Mei, D. Zhou, and K. W. Church. Query suggestion using hitting time. In *ACM Conference of Information and Knowledge Management*, 2008.

[21] E. Moxley, T. Mei, and B. S. Manjunath. Video annotation through search and graph reinforcement mining. *IEEE Trans. on Multimedia*, 12(3):184–193, 2010.

[22] B. Poblete and R. A. Baeza-Yates. Query-sets: using implicit feedback and query patterns to organize web documents. In *Proceedings of International World Wide Web Conference*, 2008.

[23] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. In *ACM International Conference on Multimedia*, 2007.

[24] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, 2007.

[25] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, 2008.

[26] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. *Subspace, Latent Structure and Feature Selection*, pages 34–51, 2006.

[27] S. Siersdorfer, J. S. Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *Proceedings of ACM conference on Research and Development in Information Retrieval*, 2009.

[28] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[29] M. Trevisiol, L. Chiarandini, L. M. Aiello, and A. Jaimes. Image ranking based on user browsing behavior. In *Proceedings of ACM conference on Research and Development in Information Retrieval*, 2012.

[30] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *IEEE Trans. on Multimedia*, 11(3):465–476, 2009.

[31] J.-R. Wen, J.-Y. Nie, and H. Zhang. Clustering user queries of a search engine. In *Proceedings of International World Wide Web Conference*, 2001.