Predicting the Popularity of Web 2.0 Items Based on User Comments*

Xiangnan He[1] Ming Gao[2] Min-Yen Kan[1] Yiqun Liu[3] Kazunari Sugiyama[1]
[1]School of Computing, National University of Singapore
[2]School of Information Systems, Singapore Management University
[3]Department of Computer Science & Technology, Tsinghua University
{xiangnan, kanmy, sugiyama}@comp.nus.edu.sg minggao@smu.edu.sg

ABSTRACT

In the current Web 2.0 era, the popularity of Web resources fluctuates ephemerally, based on trends and social interest. As a result, content-based relevance signals are insufficient to meet users' constantly evolving information needs in searching for Web 2.0 items. Incorporating future popularity into ranking is one way to counter this. However, predicting popularity as a third party (as in the case of general search engines) is difficult in practice, due to their limited access to item view histories.

To enable popularity prediction externally without excessive crawling, we propose an alternative solution by leveraging user comments, which are more accessible than view counts. Due to the sparsity of comments, traditional solutions that are solely based on view histories do not perform well. To deal with this sparsity, we mine comments to recover additional signal, such as social influence. By modeling comments as a time-aware bipartite graph, we propose a regularization-based ranking algorithm that accounts for temporal, social influence and current popularity factors to predict the future popularity of items. Experimental results on three real-world datasets — crawled from YouTube, Flickr and Last.fm — show that our method consistently outperforms competitive baselines in several evaluation tasks.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *information filtering, retrieval models;*

Keywords

Popularity Prediction; Item Ranking; Bipartite Graph Ranking; Comments Mining; BUIR

1. INTRODUCTION

The era of static webpages has been surpassed over a decade ago with the advent of Web 2.0. Users now not only post content in Web 2.0, but also participate in a spectrum of means – commenting, voting, forwarding, and tweeting – among other social actions. This characteristic of the new Web has led to a more dynamic and immediate sense of popularity, as the demand for items is influenced by such social actions in groups and by real world events. To satisfy the constantly evolving information needs of users in the Web 2.0 setting, it is indispensable for ranking engines to properly account for these temporal dynamics. However, we find that current search engines do not utilize popularity as effectively as they could. Figure 1 gives an illustrative example, which shows the YouTube results from Google for the query "The Voice of China" on the 24th of July, 2013. *The Voice of China* is a popular Chinese reality talent show that premiered in 2012, and kicked off its second season on 12th July 2013. Given this background, we believe that most users who queried "The Voice of China" in July 2013 are looking for videos of the second season. However, almost all the top results returned by Google are past popular videos from the first season. The first relevant result of the second season is ranked 16th, "below the fold" for many searchers. Inspecting the view count of these search results over the next three days, we find that the top three results receive less than 10,000 views, while the top-ranked video of the second season (ranked 16th) was viewed over 100,000 times. This indicates that at least for this particular query, Google did not make optimal use of future popularity, and hence did not satisfy many users' search expectations.

Along similar lines, Gonçalves *et al.* [13] previously observed that popularity had not been well utilized in the blog search. They found that popularity is quite different from the importance measured by PageRank [27] on a Web graph. Importantly, they show that search effectiveness and user satisfaction are significantly improved when incorporating popularity into ranking. Therefore, we believe that user experience can be improved, particularly for time-sensitive queries, if search engines can predict which items will become more popular in the near future and rank them accordingly.

Modeling and predicting the popularity of Web content can benefit many downstream applications such as online marketing [20], cache managing [1], search ranking [13], social network modeling [4], and so on. We equate estimating *popularity* with the task of predicting future view count, a direct and objective means to assess the interest of users. Though previous works have focused on popularity prediction, their primary strategy is to mine the view history of items [32, 28, 1]. However, for some external services which are not content providers, previous solutions are infeasible because they require full access to the item's view count histories [13]. While many Web 2.0 sites often provide a current view count for items, repeated crawling to build and maintain such view histories is expensive. This method also does not allow prediction for newly crawled items, due to insufficient view history.

*This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609558.

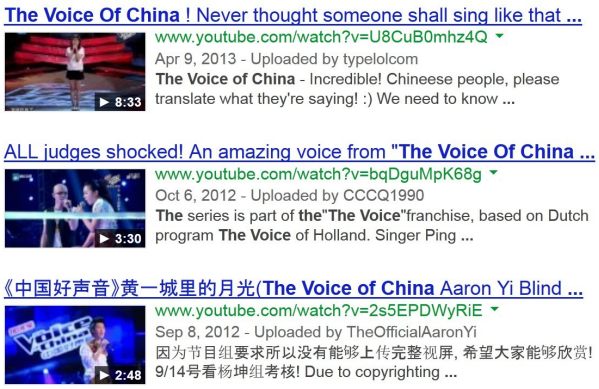


Figure 1: The top three search results for the query "The Voice of China" on 24th July 2013, from Google, restricted to the YouTube.com domain.

To address these challenges faced by external observers, we propose an alternative approach by exploiting user comments, which are more easily accessible than view counts. Comments are a rich source of information, containing not only the opinions of users, but also *timestamps* which allow us to deduce the view history, and *usernames* for mining potential social influence. As commenting (or interchangably, "reviewing") is a basic social action enabled for most Web 2.0 sites, our solution is generally applicable for Web 2.0 sites. Although many works have studied the use of comments, including summarization [16], ranking [30], clustering [15] and recommendation [31, 38], there have been little work that use comments to predict item popularity. However, comments are much sparser than views: a user viewing an item often does not comment on it. As such, simply using comment counts in a time series approach is insufficient, especially for less popular items with few user comments. Thus, it is important to mine and incorporate additional popularity signals from the comments in prediction.

We propose to predict item future popularity from comments based on three hypotheses about the 1) temporal, 2) social, and 3) current popularity factors. We model comments as a time-aware bipartite graph, on which we propose a regularization-based algorithm *Bipartite User-Item Ranking* (BUIR) to rank items by capturing the three hypotheses. We evaluate BUIR extensively on three real-world datasets that represent a spectrum of different media: YouTube (videos), Flickr (photos) and Last.fm (music). Experimental results show that BUIR consistently outperforms competitive baselines in predicting item future popularity. We further analyze the prediction quality on specific item subsets – on per-query and tiered popularity bases – to deconstruct the efficacy of BUIR in complementing Web search ranking, and to provide more insights into how to use comments for popularity prediction.

This paper is organized as follows. We first review related work in Section 2. In Section 3, we show the feasibility of using comments for popularity prediction in an initial analysis over YouTube data. In Section 4, we detail the BUIR method and evaluate it in Section 5. We conclude the paper in Section 6.

2. RELATED WORK

In this section, we review work on popularity prediction first, then discuss Web 2.0 user comment mining.

2.1 Popularity Prediction of Online Content

Popularity prediction can be classified into three broad types: statistics-based, classification-based, and model-based approaches.

Statistics-based Prediction. These approaches assume that past popularity is a good predictor of future popularity. Szabo and Huberman [32] analyzed the popularity growth of YouTube videos and Digg stories, finding a strong correlation between the logarithmically transformed past popularity and current popularity. They proposed a univariate linear model to capture this correlation. Later, Pinto *et al.* [28] extended the univariate model to a multivariate one by incorporating additional historical points and features. Radinsky *et al.* [29] proposed several time series prediction methods of user behaviors based on state-space models. All of these techniques require access to the view histories of items, which are difficult for third parties to obtain in practice, as described earlier in Section 1.

Classification-based Prediction. These approaches transform the popularity prediction problem into a discrete classification task by using different classifiers such as k-nearest neighbors [17], decision tree [17, 34], and support vector machine [17, 34]. Various features derived from the textual content, time series, and community structure are distilled as input features for the classifiers. The output of such methods are unfortunately, too coarse-grained for many applications, such as Web search ranking.

Model-based Prediction. Model-based approaches are difficult to formulate, but often yield more insight and higher accuracy. Yin *et al.* [37] ranked potentially popular items from early votes. They modeled each user's voting behavior as a constrained random process. Recently, Ahmed *et al.* [1] predicted popularity by modeling the temporal evolution of online content. They first split an item's history into time windows, and then generated clusters of items in each window. Based on the clusters, they built a transition graph to predict the most promising cluster for an item in the future.

The above methods, however, only model a user's past behavior or an item's history individually, and do not account for social signals, an important criterion in Web 2.0. Lerman *et al.* [22] analyzed friending actions in Digg as a way of propagating user behavior to influence other users. They modeled user voting behavior as a stochastic model, considering both social influence and website layout to predict story popularity. This work lends evidence that users exert varying levels of influence on others, and that such social factors need to be taken into account to predict popularity. However, their work is too specific to Digg; parts their model do not easily transfer to other sites (requiring Digg specific page view distribution and other internal factors), making the lessons drawn from their study difficult to port to other Web 2.0 sites. For items with longer lifecycles where external events may exert a strong influence, such as unexpected view bursts (i.e., YouTube videos), their approach may not work well.

2.2 Mining User Comments

User comments, as one of the most common sources for user-generated content, have received much attention in recent years.

Descriptive Information Mining. The descriptive information contained in user comments are leveraged in many applications. Mishne *et al.* [25] found that incorporating comments into blog search improved recall by 5–15%. Hu *et al.* [16] integrated comments to improve the summarization of blogs. Noise in comments is a well-known source of difficulty, yet the content of comments still shows utility in IR applications when properly handled: Filippova *et al.* engaged comments for video classification [12], while He *et al.* [15] did similarly for Web 2.0 item clustering.

Sentiment Information Mining. Sentiment latent in user comments can also be utilized to rank items. Wijaya and Bressan [35] ranked movies based on the sentiment (*positive* or *negative*) of reviews. Their obtained ranking was highly correlated with the gross

income of movies. Pedro *et al.* [30] ranked images from an aesthetic perspective by extracting image features and opinions from comments. In recent work, Zhang *et al.* [38] performed phrase-level sentiment analysis of user comments to improve the accuracy and explainability of recommender system.

Although these works have mined comments, they focused exclusively on textual content. In particular, timestamps and usernames are additional important features that can be extracted from comments, which have been neglected in existing work. We believe that there is important knowledge contained in the timestamps and user communities (evidenced by usernames), and that popularity prediction can be refined to utilize these relevance signals for search ranking. We thus propose to exploit user comments for popularity prediction. There are two works most closely aligned to our proposal, but they also have shortcomings. In [33], due to lack of ground truth, they used comments as the popularity metric, which we show differs from actual view count in Section 5.2.1. In Jamali *et al.*'s work [17] on Digg, they transformed the prediction problem into one of classification tasks, using *Digg-score* as the popularity index, rather than the views.

3. PRELIMINARIES

We now conduct a feasibility study on a YouTube dataset to validate our idea of using comments for popularity prediction.

3.1 YouTube Dataset

YouTube captures detailed video statistics, which include the view, comment and favoriting history up to the current date, as charts (Figure 2). YouTube creates these charts via the Google Chart API, which exposes the data points in the request URL. This allows us to obtain all of the data points used to create the charts, following the methodology in [11].

Since we want to study the feasibility of comment-based popularity prediction, a general sample of videos is sufficient to form a proof-of-concept. We use ten general queries, drawing from the most popular tags at collection time (9th August 2012), to generate a corpus of videos: "animal", "car", "food", "football", "game", "movie", "music", "nba", "olympic" and "people". We collect the YouTube pages containing the videos using the YouTube API, requesting the top 1,000 videos using three different order-by sorting criteria: ranking by relevance, view count and published time. From this preliminary corpus, we remove (1) duplicate videos, (2) videos with a low number of comments and views (thresholds set to 10 and 20, respectively), and (3) videos that lack statistics (some videos do not allow commenting, or choose to keep these statistics private). The analysis in the remainder of this section is based on this final corpus of 14,509 videos.

3.2 Correlation of Comments and Views

We conjecture that the comment history and view history are highly correlated, as exemplified by the view and comment curves in Figure 2. We wish to gauge the quality of their correlation to see whether we can use the comment history as a surrogate to predict future views. While prior works [25, 6] have shown that comments do exhibit a strong correlation with views, this is only done for a particular temporal *snapshot* (*i.e.*, on some given day d, how highly correlated are the total cumulative view count with the comment count). To ensure the feasibility of our approach, we need to analyze how the histories of views and comments on individual items evolve over time. To the best of our knowledge, there has not been any studies on such correlation.

The historical views of a video form a time series. We first calculate raw counts per time point, then measure the similarity between

Figure 2: A sample video's statistics in YouTube.

the comment history and view history using the correlation [5]:

$$cr = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}, \quad (1)$$

where $x_1, ..., x_n$ and $y_1, ..., y_n$ denote the comment series and the view series, $\overline{x}$ and $\overline{y}$ denote the mean of the two series, respectively.

The mean correlation coefficient for each item is 0.76, with a standard deviation of 0.3. Figure 3 shows the cumulative distribution function (CDF) of videos given their correlation. As the figure shows, more than 45% have a correlation greater than 0.9 (strong correlation), and more than 80% have a correlation greater than 0.5 (good correlation). We conclude that *comment history is highly correlated with view history*, which lends supports for our comment-based prediction proposal.

3.3 Comment Series Autocorrelation

The strong correlation between the comment history and view history indicates that we can substitute "view" for "comment". Can past comments be used to predict future views, as we propose?

To answer this question, we perform an autocorrelation analysis of the comment series. The autocorrelation coefficient measures the correlation of a time series with itself over different lags. Given the time series $x_1, ..., x_n$ and a lag k, the autocorrelation coefficient of the series $\{x_i\}$ at lag k is the correlation of series $x_1, ..., x_{n-k}$ and series $x_{k+1}, ..., x_n$. It is usually approximated as follows [5]:

$$acr_k = \frac{\sum_{t=1}^{n-k}(x_t - \overline{x})(x_{t+k} - \overline{x})}{\sqrt{\sum_{t=1}^{n}(x_t - \overline{x})^2}}. \quad (2)$$

Figure 4 shows the mean autocorrelation of the comment series at different lag k ($0 \leq k \leq 97$). The figure exhibits a short-term correlation characterized by a large value, $acr_1 = 0.64$, followed by a few further coefficients which are successively smaller ($acr_2 = 0.51, acr_3 = 0.43$). Values of acr_k for longer lags ($k \geq 40$) are approximately zero. We thus conclude that *comment histories can reflect future comment in the near-term*, and that *its predictive ability decreases with a larger lag*.

4. PROPOSED METHOD

Most applications seek to determine an item's ranking relative to other items. We thus focus on the relative ranking of items — rather than exact popularity prediction (*cf.* Section 5.2.1) — to reflect their potential popularity in the future (as in [37]).

Having shown a strong correlation between the comment and view series, an intuitive solution is to apply any time series prediction approach on the comment series. However, we notice that comments are relatively sparse compared with views, *e.g.*, many items do not have any comments in a particular time unit at all. This has an adverse impact on regression. We argue thus that in case of comments sparsity, just using the counts of comments and applying traditional time series prediction approach is insufficient; it is essential to incorporate other factors for prediction, such as social influence. To account for such latent signals in user comments, we first model user comments as a time-aware bipartite graph, and

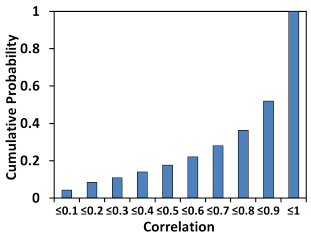


Figure 3: CDF of videos with respect to their correlation coefficient.

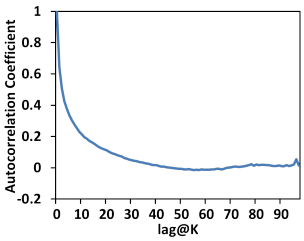


Figure 4: The average correlogram of comment series against lag k.

then predict future popularity based on this graph using a regularization framework [39], which enables the incorporation of multiple factors in a principled manner.

4.1 Bipartite User-Item Temporal Graph

Let $G = (U \cup P, E)$ be a bipartite graph, where U and P represent users and items respectively, and the edges E represent comments (Figure 5). Each edge carries a weight w, modeling its contribution towards an item's future popularity. As our analysis shows a strong near-term correlation, we assign w based on temporal considerations. We model recent (older) comments as contributing more (less) to an item's future popularity, by assigning edge weight as a monotonically decreasing exponential decay function:

$$w_{ij} = \delta^{a(t_0 - t_{ij}) + b}, \tag{3}$$

where δ is the decay parameter that controls the rate at which w_{ij} changes with time, t_0 is the ranking time and t_{ij} is the commenting time of user u_i on item p_j. a and b are constants, to be tuned for the particular media and site. Time units are arbitrary; they can be assigned as minutes, hours, days, weeks or other units, depending on the temporal resolution and the domain of items to rank. If no edge exists between u_i and p_j, then w_{ij} is zero.

Exponential functions have been widely used to model the diminishing impact of past behavior over time (*e.g.*, [10]). Due to its simplicity and interpretability, we have purposefully chosen it as a proof-of-concept of our BUIR solution in this work; however, more accurate decay functions do exist [8] (*i.e.*, polyexponential decay and sliding window function) and may further improve our model. We leave this possibility for future work.

4.2 Bipartite User-Item Ranking (BUIR)

We now present our proposed ranking method in the bipartite user-item graph. We describe the hypotheses that form the basis for our regularization function first before presenting our solution.

4.2.1 Hypotheses on Comment-based Prediction

Generally speaking, we have three hypotheses about the future popularity of an item, and wish to incorporate into our model:

H1. Temporal Factor: If an item receives many recent comments, it is more likely to be popular in the next time step (*cf.* our study of YouTube).

H2. Social Influence Factor: If the users commenting on an item are more influential, the item is more likely to receive more views in the future. This is enabled by the Web 2.0 social interfaces that propagate a user's comments to friends and followers. Such social factors have been shown to be useful in popularity prediction and recommendation [22, 23].

H3. Current Popularity Factor: If an item is already popular (*i.e.*, has accumulated a large amount of views), it is likely to garner more views in the future. This is effected by the ranking functions

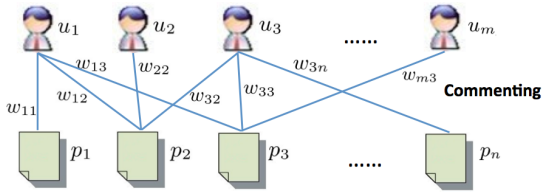


Figure 5: Bipartite User-Item Structure.

and recommendation interfaces in Web 2.0: the more views an item has, the more likely it will be suggested by the system. This "rich-get-richer" effect has been observed in some Web 2.0 systems [6].

H1 has been studied in our initial analysis of YouTube dataset. We further validate *H2* and *H3* through experiments in Section 5.3.

4.2.2 Regularizing the Hypotheses

We now devise regularizers to capture these three hypotheses. Our goal is to devise a ranking function $f : P \cup U \to \mathbb{R}$, which maps each vertex in G to a real number such that the value reflects the vertex's popularity (for items) or influence (for users).

Capturing H1 and H2. Combining *H1* and *H2* together yields an equivalent formulation: *if an item is reviewed by many influential users recently, it should be given a high score*. We model this through the regularized term $R_1(f)$:

$$R_1(f) = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{ij}\left(\frac{f(p_j)}{\sqrt{d_j^p}} - \frac{f(u_i)}{\sqrt{d_i^u}}\right)^2, \tag{4}$$

where w_{ij} is the edge weight defined in Eq. (3); n and m denote the number of items and users, respectively; d_j^p and d_i^u are the weighted degrees (*i.e.*, sum of edge weights) of item p_j and user u_i for normalization, respectively.

We now discuss the relationship between $R_1(f)$ and our hypotheses. First, minimizing $R_1(f)$ forces p_j's normalized score (*i.e.*, $f(p_j)/\sqrt{d_j^p}$) to be similar to the normalized scores of all its connected users. Thus, if p_j is commented on by influential users, its normalized score will be large (as in *H2*). Second, note that the score of p_j is normalized by $\sqrt{d_j^p}$, which is proportional to the degree of p_j. Hence, in constraining the normalized score of p_j to be similar to the scores of its neighbors, $f(p_j)$ is large when the degree of p_j is large (as in *H1*). Therefore, minimizing Eq. (4) simultaneously captures both *H1* and *H2*.

Capturing H2. We have enforced the social influence of user commenting behaviors on the popularity of items, however, we have not distinguished influential users. Intuitively, if a user has more friends, his behavior is likely to influence more users. Thus, we set a user's initial influence score proportional to the log value of his number of friends:

$$u_i^0 = \frac{\log(1+g_i)}{\sum_{k=1}^{m}\log(1+g_k)}, \tag{5}$$

where g_i is user u_i's number of friends at the ranking time. We use add-1 smoothing to address the case where a user has no friends. We can now define the regularized term $R_2(f)$ to encode initial user influence:

$$R_2(f) = \sum_{i=1}^{m}(f(u_i) - u_i^0)^2. \tag{6}$$

We note that more accurate social influence models do exist (*e.g.*, [3]), but we have purposefully chosen to rely just on the single feature of the number of friends to make our method easily generalizable to a wide range of Web 2.0 systems.

Capturing H3. To capture the potential "rich-get-richer" effect, we define the initial score of an item as:

$$p_j^0 = \frac{\log v_j}{\sum_{k=1}^{n} \log v_k}, \tag{7}$$

where v_j is the total view count of item p_j at the ranking time. Similarly, the corresponding regularizer to capture the current popularity factor of items is defined as:

$$R_3(f) = \sum_{j=1}^{n} (f(p_j) - p_j^0)^2. \tag{8}$$

Regularization function. Having defined the regularizer for each hypothesis, we combine them linearly to obtain a final regularization function:

$$\begin{aligned} Q(f) = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} (\frac{f(p_j)}{\sqrt{d_j^p}} - \frac{f(u_i)}{\sqrt{d_i^u}})^2 \\ + \alpha \sum_{j=1}^{n} (f(p_j) - p_j^0)^2 + \beta \sum_{i=1}^{m} (f(u_i) - u_i^0)^2, \end{aligned} \tag{9}$$

where the regularization parameters α and β determine the trade-off among these three terms. The first term is a *smoothness* term, that helps to rank items such that high scores are assigned to items that have been recently reviewed by many influential users (*H1* and *H2*). The second and third terms are for *consistency*, that assert that the final rankings should not overly deviate from their initial scores, which encode our hypotheses *H2* and *H3*.

4.2.3 Solving the Regularization

The regularization function $Q(f)$ defined by Eq. (9) needs to be solved (minimized) to obtain the final ranking. As two types of variables (p_j and u_i) exist in the function, we can find the solution using alternating optimization. Differentiating $Q(f)$ with respect to p_j and u_i, respectively, and letting the derivatives be 0, we have:

$$\begin{aligned} f(p_j) &= \frac{2\alpha}{1+2\alpha} p_j^0 + \frac{1}{1+2\alpha} \sum_{i=1}^{m} \frac{w_{ij} f(u_i)}{\sqrt{d_j^p}\sqrt{d_i^u}}, \\ f(u_i) &= \frac{2\beta}{1+2\beta} u_i^0 + \frac{1}{1+2\beta} \sum_{j=1}^{n} \frac{w_{ij} f(p_j)}{\sqrt{d_j^p}\sqrt{d_i^u}}. \end{aligned} \tag{10}$$

This is the iterative solution of the objective function $Q(f)$. It is guaranteed to find the global minimum, as $Q(f)$ is strictly convex in both the p_j and u_i variables (the Hessian is positive semi-definite). We show the proof in the Appendix. Other standard optimization techniques (*e.g.*, gradient descent) can also be used; alternating optimization has the advantage of quick convergence.

As the updating rules shown in Eq. (10) are linear transformations of $f(p_j)$ and $f(u_i)$, they can be equivalently written in matrix form. Let the ranking vectors be $\mathbf{p} = [f(p_j)]_{n\times 1}$ and $\mathbf{u} = [f(u_i)]_{m\times 1}$, and the initial vectors be $\mathbf{p}_0 = [p_j^0]_{n\times 1}$ and $\mathbf{u}_0 = [u_i^0]_{m\times 1}$. Let matrix $\mathbf{S}_w$ be $[\frac{w_{ij}}{\sqrt{d_j^p}\sqrt{d_i^u}}]_{m\times n}$. We then obtain the neat matrix form of Eq. (10):

$$\begin{aligned} \mathbf{p} &= \frac{1}{1+2\alpha} \mathbf{S}_w^T \mathbf{u} + \frac{2\alpha}{1+2\alpha} \mathbf{p}_0, \\ \mathbf{u} &= \frac{1}{1+2\beta} \mathbf{S}_w \mathbf{p} + \frac{2\beta}{1+2\beta} \mathbf{u}_0. \end{aligned} \tag{11}$$

By further reducing Eq. (11), we obtain a nice closed-form solution:

$$\mathbf{p}^* = [(1+2\alpha)\mathbf{I} - \frac{1}{1+2\beta} \mathbf{S}_w^T \mathbf{S}_w]^{-1} \cdot (\frac{2\beta}{1+2\beta} \mathbf{S}_w^T \mathbf{u}_0 + 2\alpha \mathbf{p}_0). \tag{12}$$

Although the closed form can be obtained, in practical cases – especially when there is a large number of items to rank – the iterative solution is preferable, as the matrix to inverse is $n \times n$. In our experiments, the iterative solution usually converges in fewer than 30 iterations, which is sufficiently efficient. Therefore, in subsequent experiments, we implement the iterative solution of Eq. (11) and adopt the name BUIR (*Bipartite User-Item Ranking*) to refer this specific instance.

4.3 Time Complexity Analysis

It is easy to show that a direct implementation of the iterative solution in Eq. (11) has a $O(mn)$ time complexity, mainly due to the multiplication of $\mathbf{S}_w^T\mathbf{u}$ and $\mathbf{S}_w\mathbf{p}$. However, note that $\mathbf{S}_w$ is typically sparse, as a non-zero entry denotes a comment by a user on an item. A representation of sparse matrix only needs to account for non-zero entries, instead of all mn entries. As such, the whole time cost of BUIR is $O(lc)$, where c denotes the number of comments, and l denotes the number of iterations executed to converge.

If one only aims to rank items without requiring a completed ranking for users, then the time cost can be further reduced through improved implementation. Embedding the update rule of $\mathbf{u}$ into the update rule for $\mathbf{p}$ in Eq. (11), we obtain:

$$\mathbf{p} = \frac{\mathbf{S}_w^T \mathbf{S}_w}{(1+2\alpha)(1+2\beta)} \mathbf{p} + \frac{2\beta \mathbf{S}_w^T \mathbf{u}_0 + 2\alpha(1+2\beta)\mathbf{p}_0}{(1+2\alpha)(1+2\beta)}. \tag{13}$$

The above can then be solved with simply iterating the update rule Eq. (13) until convergence. Note that transition matrix in the first term ($\mathbf{S}_w^T\mathbf{S}_w$) and the entire second term remain unchanged between iterations, thus they can be pre-computed offline. As such, the online ranking reduces to the straightforward power iteration algorithm for computing the stationary distribution of a Markov chain. Without any optimization, the time complexity is $O(ln^2)$. Our experiments on our largest dataset (YouTube) with over 7M comments took 7.4 seconds to complete on a modest commodity desktop (Intel quad-core 3.40GHz CPU and 8GB RAM). And in our Flickr and Last.fm datasets, BUIR only takes 0.2 seconds to finish ranking. Coupled with previous work [24] that can further accelerate computation, we believe that BUIR can be applied in real-world large-scale online item ranking.

4.4 Interpretation of BUIR

It is instructive to interpret how BUIR ranks items in relation to other graphical algorithms that have been adopted in IR.

From the iterative solution in Eq. (11), we can see that BUIR essentially captures the mutual reinforcement between users and items. The first term shows that the comment by a user will increase the target item's score; and in return, the target item increases the user's score. The number of items a user has commented on reflects his engagement, and indirectly his influence. The second term shows that the score of items and users is partially determined by prior belief. To sum up, BUIR determines a user's social influence based on two source of evidence: his level of activity and his number of friends. Analogously, BUIR determines an item's future popularity based on four aspects: the quantity of its comments, their timing (recency), the influence of its commenting users, and its current accumulated popularity.

Our proposed BUIR can be seen as a variant of PageRank [27] that treats the two types of vertices in the bipartite graph differently. To show the connection, we first focus on the first equation of Eq. (11). Assuming $\mathbf{p}$ and $\mathbf{u}$ represent the same set of vertices (removing the bipartite graph property), then the equation is equivalent to the personalized PageRank algorithm [14], where $\mathbf{S}_w^T$ (after normalization) serves as the transition matrix, and $\mathbf{p}_0$ as the

Table 1: Statistics of our three Web 2.0 datasets. Avg C:I denotes the average number of comments per item.

Dataset	#Item	#User	#Comment	Avg C:I
YouTube	21,653	3,620,487	7,246,287	334.7
Flickr	26,815	37,690	169,150	6.3
Last.fm	16,284	77,996	530,237	32.6

personalized vector. As PageRank was originally proposed for homogeneous graphs, where vertices uniformly represent entities of the same type, direct use of PageRank in our bipartite scenario will mix the weights of two types of entities, and may lead to unexpected results. There are other graph ranking algorithms that are specifically designed for bipartite graphs which are more relevant, such as HITS [19], SALSA [21], Co-HITS [9] and CoRank [36]. It is known that HITS and SALSA will fall short when the graph is disconnected (tightly knit community effect [21]). As our user-item graph is built from sparse user comments, resulting in many disconnected components, direct use of either HITS or SALSA may not lead to expected results. Co-HITS and CoRank are designed for different semantics – although they work on bipartite graphs, they consider the influence of like vertices of the same type, which is explicitly not used in BUIR (no influence among items or users).

4.5 Extensions

Our solution forms a general framework, easily extendable to incorporate other factors beyond what we have described. For new features related to individual comments, such as content and sentiment relevance, we can estimate their weight in contributing to each item's popularity and integrate them into the definition of w_{ij} in Eq. (3). For new features related to individual items or users, we can model them within BUIR's bipartite regularization framework (Eq. (9)), by adding corresponding regularized terms.

5. EXPERIMENTS

As BUIR is a general method which does not require any domain-specific knowledge, we provide a comprehensive assessment of its prediction quality over a wide variety of different Web 2.0 media. We crawl three real-world datasets from well-known Web 2.0 sites (demographics in Table 1) to assess our proposed BUIR solution.

1. YouTube (21,653 videos): The dataset used is identical to the one used in the preliminary analysis in Section 3.1, but omitting the third filter that drops videos that lack statistics.

2. Flickr (26,815 images): We follow the same collection method as in the YouTube case, using the same ten queries. We do not apply any frequency filter as this dataset is more sparse than YouTube.

3. Last.fm[1] (16,284 artists): As Last.fm's search API differs from the two other datasets, we collect this dataset by obtaining data about artists: obtaining at most 100 similar artists for each of the top 1,000 most popular artists. For the query-specific evaluation, we query on the top 10 tags that describe a music style: "classical", "country", "electronic", "folk", "hip-hop", "indie", "jazz", "metal", "pop" and "rock". We assign each artist to the single tag that is used most often to describe the artist by Last.fm users.

We choose the three datasets for ease of evaluation, as these all provide item view count. Our datasets are crawled on two different dates: for graph construction (t_0) and for obtaining ground truth (GT) for evaluation (t_3), which is 3 days after t_0. As we have observed that ephemeral trends are important to capture, we specifically aim to evaluate short-term prediction and chose 3 days as the target period to evaluate. The initial crawl t_0 for YouTube, Flickr and Last.fm is on 9th August 2012, 3rd September 2012 and 24th October 2012, respectively. For items in Flickr and Last.fm, we crawl the view count, the number of friends and the list of comments on the two dates. For YouTube, due to its privacy policy, we cannot obtain a user's number of friends, so we set the initial score for users uniformly. As older items may have accumulated many past comments but which would not significantly contribute in BUIR, we discard comments older than five months before t_0 for efficiency. If an item is commented by the same user multiple times, we only keep the most recent comment when calculating the edge weight. This also helps to avoid problems when users have tangential conversations via comments.

[1]In lieu of view count, Last.fm provides a "scrobble" count, which is the number of times Last.fm users listen to a track by the target artist. This differs from the view count of an artist's page, but we argue more indicative of an artist's popularity. For convenience, we use "view count" to refer to scrobble count in Last.fm.

5.1 Evaluation Metrics and Baselines

To assess the predicted ranking with the ground truth ranking, we employ ranking correlation in the standard form of the Spearman coefficient [2]. It measures the agreement between two rankings defined as follows:

$$S(R_1, R_2) = 1 - \frac{6 \times \sum_{i=1}^{N} (s_{1,i} - s_{2,i})^2}{N \times (N^2 - 1)}, \tag{14}$$

where N is the number of items in the ranking, $s_{1,i}$ and $s_{2,i}$ are the positions of the i^{th} item in two rankings R_1 and R_2, respectively. It ranges from -1 to 1, where 1 (-1) means a perfect agreement (disagreement) between the two rankings and 0 means no correlation.

While the Spearman coefficient is indicative of the agreement between two rankings, it does not reflect the importance of getting the top ranks correct, which are crucial for many applications such as Web search ranking. To address this, normalized discounted cumulative gain (nDCG) [18] – which rewards relevant results in the top ranks more highly than those ranked lower – is widely used for evaluating query-dependent rankings. As such, in our query-specific evaluation (Section 5.2.2), we also employ nDCG@k to evaluate the top k rankings for each query. As nDCG takes relevance levels into account, we define the top 10% of items found in the ground truth ranking as relevant, where higher ranked positions are accorded more relevance, computing a relevance score of $1 - \frac{i}{0.1 \times N}$ [37] for the i^{th} ranked item and a score of 0 for items beyond the top 10%.

We compare our BUIR with the following five baselines:

1. **View Count (VC)**: Rank based on the current view count of items. This corresponds to our belief in Hypothesis *H3* alone.

2. **Comment Count in the Past (CCP)**: Rank based on the number of comments received in the 3-day period prior to t_0 (*i.e.*, t_{-2} to t_0), corresponding to our Hypothesis *H1*.

3. **Comment Count in the Future (CCF)**: Rank based on the number of new comments received in the three days after t_0 (t_1 to t_3). This is an oracular method with access to future comments.

4. **Multivariate Linear model (ML) [28]**: We implement this method on the comment series of the 30 days prior to t_0, aggregating comment counts into 3-day windows, each contributing a feature, for a total of 10 features. This is the state-of-the-art statistical method for predicting the popularity of Web content.

5. **PageRank (PR) [27]**: Our temporal user-item graph is bipartite, which could cause the random walk to become periodic and non-stationary [26]. To work around this, we use the standard method to set a uniform self-transition weight $w_{ii} = 1$ for all nodes, and then convert the weight matrix to a probabilistic one for use with PageRank. For the damping factor, we vary its value from 0 to 1 with step size 0.05. Experimental results show consistently good performance when the damping factor is in the range 0.1 to 0.9; we set it to 0.85 as suggested in [27].

Table 2: Spearman coeff. (%) of overall evaluation.

Method	YouTube	Flickr	Last.fm
VC	73.39	58.42	67.31
CCP	83.35	59.43	67.21
CCF	84.53	59.41	67.20
ML [28]	78.24	58.00	38.09
PR [27]	80.72	28.15	10.24
BUIR	**87.72**	**64.60**	**70.43**

In our BUIR solution, there are two sets of parameters to be specified: 1) ones for assigning edge weights, and 2) ones for the regularization. Edge weights are assigned intuitively: for the time unit of YouTube and Last.fm, as comments are rich and reflective of popularity, we set it to 1 day; for Flickr, we find that the comments are posted less frequently, thus we set it to 3 days; for the time decay function in Eq. (3), we empirically set $\delta = 0.85$, $a = 1$, and $b = 0$ for all datasets. As for the regularization parameters α and β in Eq. (11), we randomly held out 10% of the dataset as development for parameter tuning. We use grid search to set the best parameters on the development portion, and then evaluate all methods on the remaining 90% test portion.

Note that although the first two baselines are heuristic and simple, they do produce reasonable results for short-term popularity prediction, thus forming competitive baselines (see [29]). For all methods, if items receive the same score, we break ties by ranking based on their current view count.

5.2 Performance Evaluation

We first evaluate the prediction of all items in each dataset. As the overall ranking of all items does not tell the whole story, we then further dissect the results through evaluating on subsets of items, in order to better understand the task and results. Specifically, we assess the performance on individual queries, and study the performance over different popularity levels.

5.2.1 Overall Evaluation

Table 2 shows that BUIR achieves the highest fidelity in ranking items of the test datasets, among all methods. Further experimentation of 10-fold cross validation shows that BUIR obtains very consistent performance, significantly outperforming all other methods ($p < 0.01$, via one-sample paired t-test). BUIR is followed by CCF and CCP, where the difference between CCP and CCF are insignificant. VC also obtains a good performance in general, indicating the effectiveness of *H3*. PageRank (PR) performs poorly for Flickr and Last.fm, indicating that just the centrality of an item in the user-item temporal graph is insufficient for prediction. We also used BUIR's initial vector $\mathbf{p}_0$ and $\mathbf{u}_0$ as the personalized vector of PageRank, which also results in poor performance. This lends evidence that separately handling the two vertex types (users and items) in the bipartite graph is important.

It is surprising that the state-of-the-art ML approach underperforms CCP, as ML leverages more information: comments in the recent 30 days compared with CCP's access to only three days. There are two possible reasons for this: 1) short-term prediction, and 2) ML's optimization criterion. As the prediction task is a short-term one, the most recent data carries the most signal – "What happened yesterday will happen tomorrow." Radinsky *et al.* [29] concurs with this observation, showing that in the short-term prediction of query and URL clicks, considering only the last value of the time series generally outperforms other regression methods, such as using power weighting function and linear weighting function. This also indicates the effectiveness and competitiveness of simple baselines in near-term popularity prediction. The second cause may stem from ML's use of minimizing the mean Relative Squared Error (mRSE) [28]) as its optimization criterion. We note

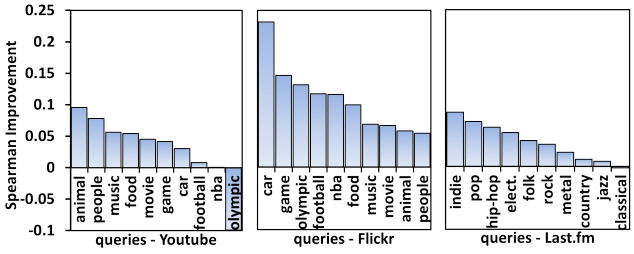


Figure 6: Improvement in Spearman coefficient between BUIR and the best baselines of query-specific evaluation.

that using mRSE as the optimization metric may favor evaluations on items with a small number of current views, as the relative popularity growth to learn are larger compared to items with a large current views[2]. As a result, the parameters learned may not be meaningful: we find that optimized weights are sometimes non-sensical (*i.e.*, negative) and that the weights for recent time units can be smaller than the earlier ones, also contradicting intuition. We also find that when we decrease the number of features to learn, performance increases. Thus, although ML does provide a better estimation of future popularity than CCP in terms of mRSE, we believe this criterion does not fit well with the goal of relative ranking. This also highlights the difference between the task of predicting the exact popularity and ranking items by the predicted popularity. Although the (exact) popularity prediction problem is more challenging compared with the relative ranking problem, we believe that the ranking problem is more suited for applications where the ordering (and not the exact numeric quantity) is important: such as search ranking, recommendation and online advertising.

It is worth noting that correlation levels dramatically differ in each dataset. YouTube shows the highest correlation while Flickr is the lowest. This indicates that comments in YouTube are generally richer and thus better reflect trending and popularity growth. Flickr users, as a whole, are less active than YouTube users (as can be seen from the comment statistics in Table 1). More specifically, many items do not receive sufficient comments to reflect their future popularity; some items even do not receive any comment within our 5-months window. In these cases, *H1* does not hold, which leads to the degraded performance of comment-based prediction methods.

Let us dissect the ranking lists to gain additional insight. In Last.fm, we notice that BUIR incorrectly ranks two items very high, while their GT ranks are low. Looking into the data, we find that the two abnormal items are two well-known artists – Lady Gaga and Madonna – ranked 4th and 7th, while their GT rank is 170th and 178th, respectively. After observing the comments, we find that the two artists receive many recent comments, but do not receive a proportional play count. Many comments are about two artists as a persona or just express praise, rather than their music. In Flickr, a similar phenomenon occurs with a few images that are ranked high but have low GT ranks. One[3] has 1,891 comments but only 4,115 views; the other[4] has 1,276 comments but only 3,299 views. Examining the details, we find that many users leave comments for participating in Flickr group activities (*"Good work! I like it!! This photo definitely deserves a Bronze Trophy! FLICKR BRONZE TROPHY GROUP Post"*), which is the cause for the excessive ratio of comments to views. In both the Last.fm and Flickr cases, the items are ranked incorrectly as the comments are not reflective of their intrinsic popularity.

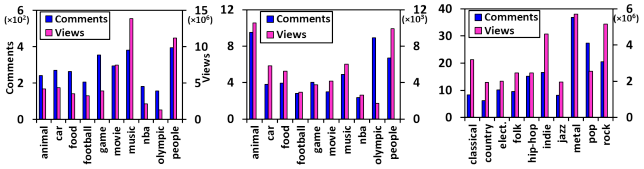
[2]The results of tiered popularity evaluation (Section 5.2.3) reflect this: ML performs better on less popular items in general.

[3]http://www.flickr.com/photos/jabitxu/7402395070/

[4]http://www.flickr.com/photos/jabitxu/6967289760/

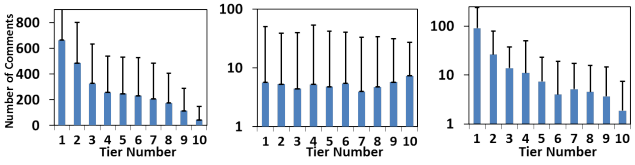
Table 3: Results (mean±standard deviation) of query-specific evaluation. "*" denotes the statistical significance for $p < 0.05$.

Metric	Spearman coefficient (%)			nDCG@10 (%)		
Method	**YouTube**	**Flickr**	**Last.fm**	**YouTube**	**Flickr**	**Last.fm**
VC	71.98±14.14	46.72±7.82	67.86±5.76	**64.70±22.23***	**67.19±15.75***	**90.25±4.96***
CCP	82.41± 2.50	48.06±7.90	66.97±4.70	46.66±29.89	61.35±18.56	82.52±10.85
CCF	**83.42±2.7***	48.12±7.80	67.27±4.45	**73.04±16.97***	56.94±25.73	78.57±12.83
ML [28]	76.95± 5.50	50.00±6.50	39.15±4.04	27.85±30.76	50.74±18.64	74.30±11.15
PR [27]	79.66± 4.72	27.80±14.87	9.22 ±11.66	61.10±21.92	54.53±22.62	81.16±10.07
BUIR	**85.98±5.92***	**55.22± 6.10***	**70.42±4.43***	**76.13±12.29***	**74.19±15.70***	**88.19±4.68***



(a) YouTube (b) Flickr (c) Last.fm

Figure 7: Comment and view statistics (mean) for items returned by the ten queries.



(a) YouTube (b) Flickr (c) Last.fm

Figure 9: Comment statistics (mean and standard deviation) for items in the ten popularity tiers.

We recap our example query about "The Voice of China" from Figure 1. As shown in the figure, the top results of Google's search are all past popular videos that do not touch on the second season until the 16th search result. Looking into the comments, we find the top three videos receive very few comments close to the crawl date, while videos concerning the second season in contrast, received many recent comments. We import the top 20 results (along with their comment streams) of Google's search for this query, providing to our BUIR algorithm for reranking. BUIR ranks the video about the second season in the top position (which we view as the correct ranking). This example also shows the relevance signals resided in user generated comments, which complement the search ranking when other signals are insufficient to rank well.

From our two case studies, we can see that if an item is experiencing a burst and the burst is reflected in comments, BUIR successfully ranks it high. However, in the case of items receiving a disproportionally high number of comments to views, disobeying *H1*, BUIR is misled into making incorrect rankings.

5.2.2 *Query-Specific Evaluation*

We also evaluated prediction quality on a per-query basis to test BUIR's variability for specific queries and its feasibility for use with Web search ranking.

On our datasets, per query, BUIR needs to rank between 500 to 3,500 items. Figure 7 shows the average number of comments and views of items for each query, which highlights the variability in comment and view count between queries; it is not necessary that items with many views have a corresponding number of comments, or vice versa (seen the case of query "pop" and "rock" of Last.fm).

Table 3 shows the average performance over all queries. We perform one-sample paired t-test (p-value = 0.05) to assess statistical significance. Supporting our previous results in Table 2, BUIR performs the best in all datasets. Specifically, as judged by the Spearman coefficient, BUIR outperforms all baselines except the case of CCF in YouTube, where they are statistically comparable. Surprisingly, for nDCG@10, VC achieves comparable performance (the same significance level) with BUIR in all datasets. As nDCG@10 only evaluates the ranking of the top 10 positions, of which are all popular items, we hypothesize that the current view count is a good indicator of popular items. This motivates the need to analyze prediction at other popularity levels (detailed later in Section 5.2.3).

We further examine the performance for each query. Figure 6 shows the percentage of improvement in Spearman coefficient between BUIR and the best baselines (CCF, ML, and VC for YouTube, Flickr, and Last.fm, respectively). As can be seen, BUIR bests the baselines compared in all cases with the exception of "olympic" in YouTube and "classical" in Last.fm. We investigate the cause for these performance exceptions.

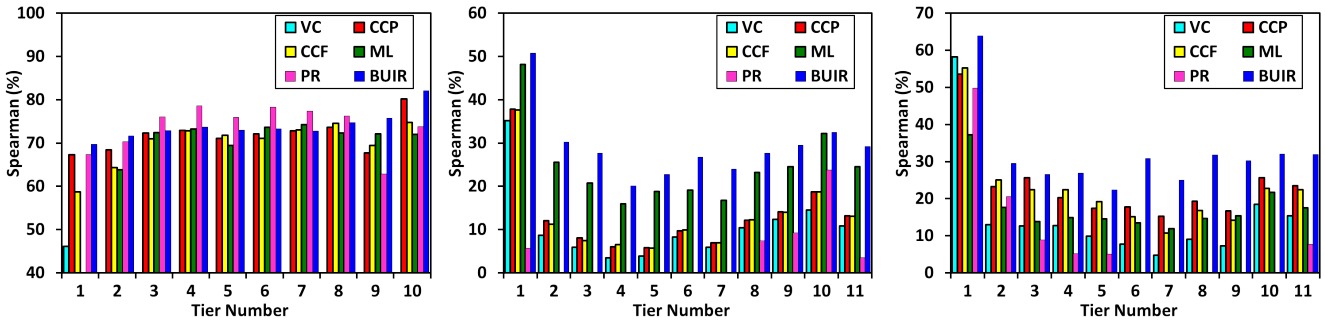
For "olympic", CCF and CCP show a significant improvement over other methods (0.80 for CCF and CCP; 0.72 and 0.34 for BUIR and VC, respectively). The YouTube dataset is crawled on 9th August 2012, during the London Olympic Games. Many collected videos of the query "olympic" from YouTube are indeed about the London Olympic Games. These videos are rather new, such that they have not accumulated enough view count to reflect their popularity (*cf.* Figure 7's view statistics). However, the recent comments are more reflective, as users are actively commenting on the events. From the user comments, we observe that users watch videos largely according to their interests or perhaps their country's medaling in an event. In this case, *H2* (Social Influence Factor) does not strictly hold. Hence, our method does not give the better result. For such new items, we postulate that performance may be improved with a more fine-grained time unit for BUIR. Changing the time granularity to an hourly basis, BUIR's performance improves (from 0.72 to 0.76), although still underperforming CCP and CCF. This lends tentative support to our idea, but which needs further investigation in future work.

For the results of "classical" in Last.fm, VC obtains the highest Spearman coefficient (0.781), followed by BUIR (0.780) and CCF (0.765). The query "classical" reflects a wide range of classic musicians, such as Frédéric Chopin and The Beatles. Such items have existed for a long time, and have already accumulated many views and reached a steady state in attracting views. In these cases, current view count (VC) reflects their future popularity well.

5.2.3 *Tiered Popularity Evaluation*

While BUIR performs well overall, does it perform consistently on items of different popularity? To answer this question, we study the prediction quality over different popularity levels. We first sort items by descending view count at t_0 and then split into ten equal-sized subsets: Tier-1 (most popular) to Tier-10 (least popular). We report the results for ranking correlation (note that as each tier accounts for a popularity range, nDCG is already considered).

Figure 9 reports the comment statistics (mean and standard deviation) of the ten tiers. Both the YouTube and Last.fm datasets show the same trend: the average number of comments decreases when moving to higher (less popular) tiers, while all tiers in Flickr do not show much difference. This is because Flickr users largely refrain from making comments compared to YouTube and Last.fm users. As a result, popular items with high view count do not necessarily



(a) YouTube (b) Flickr (c) Last.fm

Figure 8: Results of the tiered popularity evaluation for the three datasets. Note that the Y-axis differs per chart.

mean that they will have a high number of comments. From the comment statistics, we can see that the items in high tiers are less popular items with a low number of comments in general.

Figures 8 shows the performance broken down by tier. In general, we observe the same trends over all three datasets. Firstly, BUIR consistently performs well and the improvements over the comment-based baselines (CCP and CCF) are more noticeable for higher tiers, corresponding to less popular items. Secondly, current view count (VC) performs well for low tiers while suffers significantly for high tiers, and is worse than CCF and CCP. As VC ranks well for most popular items (*cf.* nDCG@10 of query-specific evaluation, in Table 3), we conclude that the current view count is a good predictor for popular items, but not for less popular items. Furthermore, we also note CCF does not always outperform CCP, although CCF utilizes the future knowledge. This indicates the limitation of simply using the comment count for popularity prediction, and motivates the necessity of mining more signals from user comments for prediction.

For Flickr, BUIR improves over CCP and CCF significantly in all tiers; while in the Last.fm case, BUIR shows slight improvement in lower tiers (less than 5), which represent more popular items. To be precise, the average improvement over CCF in Tiers 1–5 is 5.0%, while in Tiers 6–10, the improvement is 12.1%. We note that the average number of comments in Tier 1–5 is 30.0, while in Tier 6–10 is only 4.3. This indicates that for items in the top tiers (which can be said to have already accumulated sufficient comments), taking social influence into account may not capture much additional signal. Conversely, this highlights social influence as a good signal for prediction of less popular items, earlier given as *H2*.

To conclude the above three sets of experiments, we recap the key findings to predict popularity based on user comments:

- For popular items which have already accumulated many views, the current view count predicts future popularity well.
- For items with sufficient number of comments, the recent comments are a good predictor for future popularity.
- For the bulk of less popular items, neither the current views nor recent comments is sufficient for quality prediction; it is important to incorporate more signals, such as social factors.
- Most importantly, our proposed BUIR method realizes the most effective and consistent prediction performance, by accounting for temporal, social and current popularity factors.

5.3 Hypotheses Study

In this final subsection, we wish to validate the necessity for modeling all three comment-based hypotheses in BUIR. As *H1* is intuitive and has been studied in Section 3, we concern ourselves primarily with the *H2* (social influence) and the *H3* (current popularity) factors.

Table 4: Spearman coefficient of overall prediction and performance decrease of different parameter settings.

Setting	YouTube	Flickr	Last.fm
$\alpha = 0$	81.01 (-7.7%)	52.99 (-18.0%)	56.45 (-19.9%)
$\beta = 0$	64.05 (-27.0%)	62.68 (-3.0%)	68.36 (-2.9%)
$\alpha, \beta = 0$	51.24 (-41.6%)	53.77 (-16.8%)	47.22 (-33.0%)

In BUIR, there are two regularization parameters, α and β, which determine the weight of *H3* and part of *H2* (social influence factor captured by users' initial score) in prediction. Table 4 shows the prediction performance when regularization parameters are set to 0 (to be clear, a "0" setting nullifies the corresponding factor). As can be seen, when either α or β is set to 0, BUIR suffers and does not predict well; when both α and β are zeroed, the performance further decreases. These results provide additional support to validate our hypotheses *H3* and *H2*. As such, we conclude that every factor captured in BUIR — *H1*, *H2* and *H3* — is necessary for high-quality popularity prediction based on user comments.

6. CONCLUSION

In this work, we systematically investigate how to best leverage user comments for predicting the popularity of Web 2.0 items. We show that simply applying time series methods on the comment series does not predict well, and that it is important to mine additional signals from comments. To remedy this, we propose three hypotheses, that separately accounting for temporal, social influence and current popularity factors. We introduce a new ranking algorithm, Bipartite User-Item Ranking (BUIR), that realizes these hypotheses under a regularization framework. Extensive experiments on three different Web 2.0 media — YouTube, Flickr and Last.fm — show the effectiveness of our proposed method. Detailed analysis reveals that the factors individually only predict well for some subset of items, while combining all under the proposed BUIR methodology yields the highest quality predictions. Importantly, our proposed solution is general: it is easily extended to incorporate additional factors, and is applicable to ranking items when user comments are available.

The current work on BUIR ignores the content of the comments. In the future, we will study how to optimally incorporate the content analysis of user comments. We believe the proper modeling the relevance and sentiment of comments towards an item will aid prediction. As evidenced in our dataset, some items with unusually high comment-to-view ratio have shown the need for relevance analysis. Finally, we plan to operationalize our comment-based prediction in real-world Web search applications, such as ranking, contextual advertising and recommender systems.

7. APPENDIX — PROOF OF CONVEXITY

We prove the convexity of regularization function $Q(f)$ in Eq. (9) by showing its Hessian is positive semi-definite.

The second order derivative of $Q(f)$ is:

$$\frac{\partial^2 Q}{\partial p_j \partial p_j} = 1 + 2\alpha; \frac{\partial^2 Q}{\partial u_i \partial u_i} = 1 + 2\beta; \frac{\partial^2 Q}{\partial p_j \partial u_i} = \frac{-w_{ij}}{\sqrt{d_j^p}\sqrt{d_i^u}}. \quad (15)$$

Let the matrix $\mathbf{A}$ be the $(m+n) \times (m+n)$ weighted adjacency matrix of the user-item bipartite graph. Then, the Hessian of $Q(f)$ $\mathbf{H}$ can be written as:

$$\mathbf{H} = 2\mathbf{M} + (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}), \quad (16)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{D}$ is a diagonal matrix where each entry $\mathbf{D}_{ii}$ is the weighted degree of i-th vertex (can be an item or a user). $\mathbf{M}$ is a diagonal matrix that each entry $\mathbf{M}_{ii}$ is α or β, depending on the i-th vertex denotes an item or a user. Note that the matrix $(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})$ is the normalized Laplacian matrix of the graph. By spectral graph theory [7], the normalized Laplacian matrix of a graph is positive semi-definite. Meanwhile, $\mathbf{M}$ is also positive semi-definite because its eigenvalues are all non-negative (eigenvalues of a diagonal matrix are its diagonal values). Finally, the addition of these two positive semi-definite matrices is also positive semi-definite, concluding that the Hessian matrix $\mathbf{H}$ positive semi-definite.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments, and wish to acknowledge the additional discussions with Jun-Ping Ng, Aobo Wang, Tao Chen, and Jinyang Gao.

8. REFERENCES

[1] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proc. of WSDM '13*, pages 607–616, 2013.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, volume 463. ACM press New York, 1999.

[3] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: Quantifying influence on Twitter. In *Proc. of WSDM '11*, pages 65–74, 2011.

[4] Y. Cha, B. Bi, C.-C. Hsieh, and J. Cho. Incorporating popularity in topic models for social network analysis. In *Proc. of SIGIR '13*, pages 223–232, 2013.

[5] C. Chatfield. *The Analysis of Time Series: An Introduction, Sixth Edition*. Taylor & Francis, 2003.

[6] S. V. Chelaru, C. Orellana-Rodriguez, and I. S. Altingovde. Can social features help learning to rank Youtube videos? In *Proc. of WISE '12*, pages 552–566, 2012.

[7] F. R. Chung. Spectral graph theory. 92, 1997.

[8] E. Cohen and M. J. Strauss. Maintaining time-decaying stream aggregates. *Journal of Algorithms*, 59(1):19–36, 2006.

[9] H. Deng, M. R. Lyu, and I. King. A generalized Co-HITS algorithm and its application to bipartite graphs. In *Proc. of KDD '09*, pages 239–248, 2009.

[10] Y. Ding and X. Li. Time weight collaborative filtering. In *Proc. of CIKM '05*, pages 485–492, 2005.

[11] F. Figueiredo, F. Benevenuto, and J. M. Almeida. The tube over time: characterizing popularity growth of Youtube videos. In *Proc. of WSDM '11*, pages 745–754, 2011.

[12] K. Filippova and K. B. Hall. Improved video categorization from text metadata and user comments. In *Proc. of SIGIR '11*, pages 835–842, 2011.

[13] M. A. Gonçalves, J. M. Almeida, L. G. dos Santos, A. H. Laender, and V. Almeida. On popularity in the blogosphere. *Internet Computing, IEEE*, 14(3):42–49, 2010.

[14] T. H. Haveliwala. Topic-sensitive PageRank. In *Proc. of WWW '02*, pages 517–526, 2002.

[15] X. He, M.-Y. Kan, P. Xie, and X. Chen. Comment-based multi-view clustering of web 2.0 items. In *Proc. of WWW '14*, pages 771–782, 2014.

[16] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proc. of SIGIR '08*, pages 291–298, 2008.

[17] S. Jamali and H. Rangwala. Digging Digg: Comment mining, popularity prediction, and social network analysis. In *Proc. of WISM '09*, pages 32–38, 2009.

[18] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proc. of SIGIR '00*, pages 41–48, 2000.

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

[20] H. Lakkaraju and J. Ajmera. Attention prediction on social media brand pages. In *Proc. of CIKM '11*, pages 2157–2160, 2011.

[21] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks*, 33(1):387–401, 2000.

[22] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *Proc. of WWW '10*, pages 621–630, 2010.

[23] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proc. of WSDM '11*, pages 287–296, 2011.

[24] F. McSherry. A uniform approach to accelerated PageRank computation. In *Proc. of WWW '05*, pages 575–582, 2005.

[25] G. Mishne and N. Glance. Leave a reply: An analysis of Weblog comments. In *Third annual workshop on the Weblogging ecosystem*, 2006.

[26] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford InfoLab, 1999.

[28] H. Pinto, J. M. Almeida, and M. A. Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *Proc. of WSDM '13*, pages 365–374, 2013.

[29] K. Radinsky, K. Svore, S. Dumais, J. Teevan, A. Bocharov, and E. Horvitz. Modeling and predicting behavioral dynamics on the web. In *Proc. of WWW '12*, pages 599–608, 2012.

[30] J. San Pedro, T. Yeh, and N. Oliver. Leveraging user comments for aesthetic aware image search reranking. In *Proc. of WWW '12*, pages 439–448, 2012.

[31] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to comment?: Recommendations for commenting on news stories. In *Proc. of WWW '12*, pages 429–438, 2012.

[32] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[33] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the popularity of online articles based on user comments. In *Proc. of WIMS '11*, pages 67–75, 2011.

[34] A. Wang, T. Chen, and M.-Y. Kan. Re-tweeting from a linguistic perspective. In *Proc. of NAACL-HLT '12*, pages 46–55, 2012.

[35] D. T. Wijaya and S. Bressan. A random walk on the red carpet: Rating movies with user reviews and PageRank. In *Proc. of CIKM '08*, pages 951–960, 2008.

[36] R. Yan, M. Lapata, and X. Li. Tweet recommendation with graph co-ranking. In *Proc. of ACL '12*, pages 516–525, 2012.

[37] P. Yin, P. Luo, M. Wang, and W.-C. Lee. A straw shows which way the wind blows: Ranking potentially popular items from early votes. In *Proc. of WSDM '12*, pages 623–632, 2012.

[38] Y. Zhang, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proc. of SIGIR '14*, 2014.

[39] D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Pattern Recognition*, pages 361–368. Springer, 2005.