Knowl Inf Syst (2014) 41:251–276
DOI 10.1007/s10115-013-0717-8

REGULAR PAPER

Learning to annotate via social interaction analytics

Tong Xu · Hengshu Zhu · Enhong Chen · Baoxing Huai · Hui Xiong · Jilei Tian

Received: 5 May 2013 / Revised: 23 September 2013 / Accepted: 29 November 2013 / Published online: 14 December 2013

Abstract Recent years have witnessed increased interests in exploiting automatic annotating techniques for managing and retrieving media contents. Previous studies on automatic annotating usually rely on the metadata which are often unavailable for use. Instead, multimedia contents usually arouse frequent *preference-sensitive interactions* in the online social networks of public social media platforms, which can be organized in the form of *interaction graph* for intensive study. Inspired by this observation, we propose a novel media annotating method based on the analytics of *streaming social interactions* of media content instead of the metadata. The basic assumption of our approach is that different types of social media content may attract latent social group with different preferences, thus generate different preference-sensitive interactions, which could be reflected as localized dense subgraph with clear preferences. To this end, we first iteratively select nodes from streaming records to build the *preference-sensitive subgraphs*, then uniformly extract several static and topologic features to describe these subgraphs, and finally integrate these features into a learning-to-rank

T. Xu · H. Zhu · E. Chen (✉) · B. Huai
School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, Anhui, China
e-mail: cheneh@ustc.edu.cn

T. Xu
e-mail: tongxu@mail.ustc.edu.cn

H. Zhu
e-mail: zhs@mail.ustc.edu.cn

B. Huai
e-mail: bxhuai@mail.ustc.edu.cn

H. Xiong
Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 07102, USA
e-mail: hxiong@rutgers.edu

J. Tian
Nokia, Beijing 100176, China
e-mail: jilei.tian@nokia.com

framework for automatic annotating. Extensive experiments on several real-world date sets clearly show that the proposed approach outperforms the baseline methods with a significant margin.

Keywords Automatic annotating · Learning to rank · Social media

1 Introduction

With the prosperity of Web 2.0, a large amount of "*user-generated content*" (UGC) [11] have been produced and diffused everyday within online social media platforms like YouTube and Flickr. For instance, a report reveals that on average 24 h long of video are uploaded and more than one million views occurred per minute at YouTube.[1] Such a huge amount of media contents raises significant challenges for content management and retrieval. Since artificial labeling is clearly a mission impossible, automatic media annotating techniques become urgently required. Indeed, the candidate annotations are usually predefined as the *system-level labels* to describe the general properties of media content. From the machine learning perspective, the system-level annotating task can be casted as the multi-classification or ranking problems.

Specially, the traditional annotating techniques always analyze the *metadata* for labels, such as genre or album information for music and directors or leading roles for movies. However, for most of the UGC in social networks, the metadata are usually insufficient or even unavailable, especially for those media content uploaded by so-called *grassroot* users. For example, at Youku,[2] one of the largest video sharing Web site in China, most of the original videos contain only titles or at most a few sentences as description. Such situations raise significant challenges for label generation, since the contextual information is quite limited with only few semantic words contained in the short description, which are obviously too sparse and ambiguous. Therefore, further efforts must be made to support automatic annotating even there is no metadata.

In the literature, there are several directions for automatic annotating research. First, the textual context of media content, such as the title, descriptions, comments or external Web information [4,28], can be analyzed to annotate media content. However, the context may not be directly related to media content. For example, the comments of a photograph may be about merely emotional expression rather than the photograph topic. Also, in some other application scenarios, such context may not be always available. Second, there is a lot of work studying how to annotate with mining the characteristics of media content themselves, e.g., low-level image/video features [17]. However, feature extraction from multimedia contents is computationally expensive and the learning effectiveness is limited due to the gap between low-level features and high-level semantic annotations. Finally, the most popular way is to analyze the personalized tags given by end users, also known as folksonomy, but the ambiguity and irrelevance of personalized tags may influence the results [27]. Also, it is a nontrivial task to process the tags due to cross-language translation [16] as well as some other issues.

Indeed, a unique property of online social media platform is that users could share their ideas and comments on certain media contents with other viewers, in other words, the "social" factor. With frequent interaction, users may choose to build the social connection or even

[1] http://www.website-monitoring.com/.

[2] http://www.youku.com/.

form the social community, especially for those who hold *common preferences*. Then, due to the message pushing mechanism of the online social network services (SNS), if users create, comment or share some fresh media contents, all the friends will be notified and usually those friends with similar tastes will join the discussion and diffusion, which leads to the *preference-sensitive interactions* in social network. The following motivating example intuitively illustrates the description above.

Example 1 (**A Motivating Example**) Alice and Bob are anonymous without any common connection in real life. However, since they are both big fans of disaster movies, they usually focused on related contents in online social media platform and then discussed with some other viewers. Owing to the similar preference and frequent interactions, they decided to build the online social connection. Then, if Alice viewed some new disaster movies, e.g., the 3-D "Titanic," Bob will be notified and probably he will join the discussion, which results in the interaction behavior between Alice and Bob. In this case, the interaction indeed reveals that the movie "Titanic" is relevant to the **common preferences** of Alice and Bob, i.e., the **"disaster"** genre.

The above example is not occasional. Actually, social media represent the developing "content-based" social network, in which *preference factors* play an increasingly significant role in the formation of social structure. A report [13] has announced that, on one hand, users tend to be friend those who have similar preferences rather than try to influence their friends after being connected. On the other hand, users tend to rely on their friends as "collector" and "filter" of fresh information, which motivates the formation of interactions. Inspired by the observations above, the analytics of preference-sensitive interaction within online social network might be a promising way to discover adequate annotations for media contents.

However, there are still many challenges along this line. First of all, it is difficult to describe the frequent preference-sensitive interactions within social network. If only pairwise interactions are considered, the performance might be severely limited with ignoring the global social links, while the complete social network graph will definitely lead to expensive computation. Besides, the motivation of interactions might be various but not only common preferences, for instance, viewers may be influenced by opinion leaders or fashion trend especially in the asymmetric social network like Twitter, which results in noisy records to distinguish their preferences. This problem might be relieved with considering the graph structure to highlight the real preferences during the information flow. Thus, the social network should be introduced in an effective and efficient way for the preference-sensitive interaction analysis.

To deal with the above challenges, in this paper, we propose a novel media annotating approach based on the analysis of the *streaming interactions* of media contents instead of metadata. The basic assumption of our approach is that different types of social media content may attract viewers with different preferences, which generates *preference-sensitive interactions* in the online social network. For reducing the computational complexity with comprehensive graph structure, we first select nodes iteratively from the streaming records to build the *preference-sensitive subgraphs* based on social links and time series. Then, we describe each subgraph with two types of features simultaneously when loading the streaming records, namely *static* and *topologic* features. Finally, the features will be integrated into a "learning-to-rank" framework to score and rank each candidate annotation. Particularly, a unique perspective of our approach is that it can integrate social interactions with multiple information sources for annotating media contents without any semantic gap. To be specific, the contributions of this paper can be summarized as follows:

– First, to the best of our knowledge, we are probably the first to iteratively leverage the streaming interaction records for revealing the adequate annotations of social media contents. The proposed approach provides an effective and efficient way for social media annotating, even if there is no metadata.

– Second, we study and extract several effective features through the deep analytics of social interaction graphs and then integrate them into the learning-to-rank framework for annotating. The proposed approach can generate uniformed representation for annotations without semantic factors and this helps to reduce incremental learning and model retraining processes; therefore, the scalability of our approach with new defined labels can be guaranteed.
– Third, we introduce the social network and social interaction factors to learn the users' interests, which can effectively relieve the interference from various viewing motivations for better understanding users' real preferences and then the adequate annotations. The experimental results clearly show that our approach outperforms the baselines with a significant margin, especially for unpopular preferences.

Overview The rest of paper is organized as follows. Firstly, Sect. 2 summarizes the related work about social media annotating and ranking techniques. Then, Sect. 3 illustrates some preliminaries of the social media annotating task and then introduces the overview of our framework. In Sect. 4, we explain some technical details, including how to iteratively build the preference-sensitive interaction subgraphs from streaming data and how to extract effective features to describe the candidate annotations. Then, in Sect. 5, we discussed some related problems, e.g., the ranking models, user preference estimation and the multi-source integration. The experimental results and further discussion will be demonstrated in Sect. 6. Finally, in Sect. 7, we conclude our work and propose the future plan.

2 Related work

In recent years, automatic annotating techniques have been researched intensively due to the urgent requirement of social media content management and retrieval. Generally, the related works could be grouped into two categories, namely **personalized tagging** and **generalized labeling**. The former tends to describe content in detailed and individual way, while the latter tends to provide a categorized description and thus hold a relatively fixed vocabulary and higher reuse frequency [12]. In this paper, we focus on the generalized labeling task. To deal with the deficiency of metadata in "user-generated contents," for the documents or media contents with rich text [19], text or expanded information, like commentary, anchor text [10] or search engine results [4] could be analyzed. For multimedia contents, prior arts focus on analyzing the characteristics of multimedia contents, such as graphic [17] or audiovisual [20] features to learn the classifiers. Besides, personalized tags are also analyzed, usually following Random Walk or similar ways like in [14].

Another category of related work is the learning-to-rank techniques, which could be generally divided into three main aspects: The **pointwise ranking** methods usually translate the ranking problems as regression [5] or multiple ordinal classification [15] problem, while the **pairwise ranking** techniques define the loss function based on the partial order relationships, then indeed build a binary classifier to compare each given pair of samples like RankSVM [9], RankBoost [8] and IRSVM [2]. And finally, the **listwise ranking** algorithms target at directly optimizing the global order which will be measured by some IR metrics like NDCG, MAP or KL-divergence, e.g., AdaRank [22], SVM-MAP [26] and ListNet [3].

Recently, some works attempted to utilize the learning-to-rank techniques to deal with the annotating task. For instance, the correlation between tags and visual characteristics is analyzed to formulate the task as extended correlation ranking [21], and another example is to rank the enhanced geospatial proximity of the photographs [18]. Since the learning-to-rank techniques have been maturely studied, those researches usually focus on the feature selection and derivation, with directly utilizing the state-of-the-art ranking models.

3 Annotating framework via social interaction analytics

In this section, we introduce the overview of our novel annotating framework. To be specific, we first introduce some preliminaries of social media annotating task. Then, we present the formulation of interaction subgraphs based on streaming media records. Finally, we demonstrate the framework of our novel media annotating approach in the perspective of learning-to-rank task. Particularly, to facilitate understanding, Table 1 illustrates the mathematical notations used in this paper.

3.1 Preliminaries

In this paper, we mainly focus on the system-level media annotating task. To be specific, the problem can be formally defined as follows:

Definition 1 (*System-level Media Annotating*) Given a media content c and a system-level annotation set A, which is predefined by media content administrators, the task of system-level media annotating is to find a subset $A_c^* \subset A$ for labeling c with respect to its characteristics.

In the above definition, each annotation $a_z \in A$ indicates a certain characteristic of media content, such as creator, genre or nationality. What should be noted is that the annotations in A_c^* have no partial order preference, since the labels may just represent different aspects of media content. For example, it is hard to say the "Titanic" is firstly a love story, then a 3-D movie.

As we mentioned above, users' viewing and interacting behaviors indicate that the content meets their preferences, especially when frequent preference-sensitive interactions occurred. Intuitively, the preferences can be captured by the labels contained in media contents. Thus, we can also leverage the annotation set A to represent the *preference factors* of users within

Table 1 Mathematical notations

Symbol	Description
$A = \{a_z\}$	The set of system-level annotation
$C = \{c_k\}$	The set of media contents
$V = \{u_i\}$	The set of users in social network
$E = \{e_{ij}\}$	The set connections in social network
$G = \langle V, E\rangle$	Social network graph
$G_c = \langle V_c, E_c\rangle$	Interaction graph of content c
$G_c^z = \langle V_c^z, E_c^z\rangle$	Preference-sensitive subgraph of content c for annotation a_z
c_k	A piece of media content
$\mathbf{f_i}$	Preference factor for u_i
r_{ik}	Rating for c_k given by u_i
w_{ij}	Common preference strength between u_i and u_j

the social network. Indeed, a media content may attract users because of multiple preference factors, e.g., Cindy appreciates the movie new 3-D "Titanic" since she really enjoys the romantic story and also the 3-D effect. Therefore, we define the preference factor as a multi-dimensional vector as follows:

Definition 2 (*Preference Factor*) The preference factors of each individual user u_i could be represented as a $|A|$-dimensional vector $\mathbf{f_i}$, where the zth element f_i^z denotes the normalized preference level of annotation a_z, which satisfies $\Sigma_z f_i^z = 1$.

Indeed, an intuitive approach for calculating $\mathbf{f_i}$ is to count the annotation frequency based on viewing history. For example, if Cindy viewed "Titanic" containing label "Love" and "3-D" and "Waterloo Bridge" containing tags "Love" to Bob before, then the preference factor of Alice can be counted as {"Love":2/3, "3-D":1/3}. However, as discussed in Sect. 1, this counting approach may be too simple to reflect users preferences due to the interference of various motivations. Considering that the preference-sensitive interactions in social network may strengthen the authentic preference reflection, such information should be merged to refine the preference discovery. Related technical details will be discussed in Sect. 5.2.

Furthermore, social network plays an important role in social interactions due to the message pushing mechanism. Similarly, we define the social network factors in online social network service (SNS) as follows:

Definition 3 (*(Asymmetric) Social Network*) The online social network could be represented as a graph $G = \langle V, E\rangle$, here V represents the set of users (nodes), in which each $u_i \in V$ contains preference factor $\mathbf{f}_i$. Also, E denotes the set of edges, i.e., links in the social network.

What should be noted is that in asymmetric SNS *(*e.g., Twitter*)*, the social links only indicate **"follow"** but not mutual connections, i.e., the edge $e_{ij} \in E$ only indicates that u_i is followed by u_j. For symmetric social networks *(*e.g., Facebook*)*, the mutual relationship could be described as two edges, i.e., (i, j) as e_{ij} and e_{ji} in the definition.

3.2 Formation of interaction graph

Based on the preliminaries above, in this part, we start to discuss about how to deal with the social interactions for media annotating. Since the clear point-to-point interaction records is usually unavailable, with considering about the message pushing mechanism in typical SNS scenario, we simply define the *social interactions* as consequential behaviors (e.g., commenting, sharing or any other operations which trigger the system notification to friends/followers) occurred within social links, i.e., if two behavior records $\langle u_i, t_i\rangle$ and $\langle u_j, t_j\rangle$ occurred which satisfies $t_i < t_j$ and $e_{ij} \in E$, we announce interactions happened between these two users. With this definition, intuitively we could analyze the social interactions as *interaction graph* structure, and clearly since each specific media content will be notified only for once, the graph will be a *directed acyclic graph* (DAG) which does not contain circle structure.

As discussed in Sect. 1, some localized interactions with certain preference exist in the interaction process, which lead to the *preference-sensitive subgraphs*. Figure 1 illustrates how to partition preference-sensitive subgraphs from the viewing records of the movie "Titanic," which contains three annotations as "Disaster," "3D" and "Love." The edges in graph are directed from the top down. Since the system-level annotations do not contain explicit semantic connections with each other, for simplifying the formulation, we could divide those subgraphs which are mutually independent to present the interactions which only reflect one

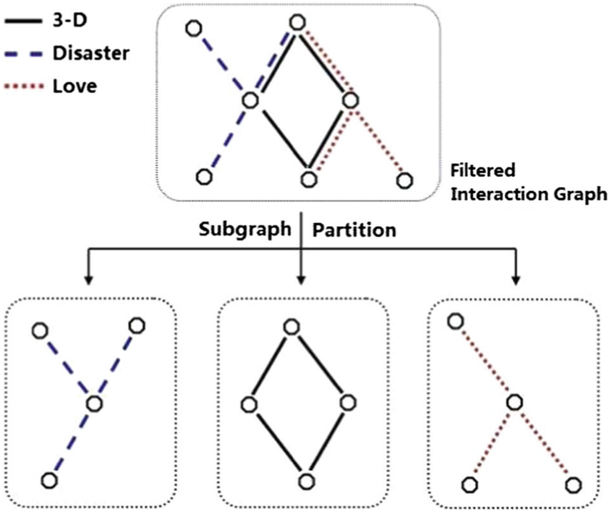


Fig. 1 The example of generating preference-sensitive subgraphs

Algorithm 1 Building interaction graph

Input: social network $G = \langle V, E \rangle$, sorted viewing records $\{u_i\}_z$ for a_z;
Output: the a_z-sensitive interaction graph $G_z = \langle V_z, E_z \rangle$
Initialization: $V_z = \emptyset$, $E_z = \emptyset$;

```
1: for each u_i in {u_i}_z
2:   for each u_s ∈ V_z
3:     if e_si ∈ E then
4:       V_z = V_z ∪ {u_i};
5:       E_z = E_z ∪ {e_si};
6: return G_z = ⟨V_z, E_z⟩;
```

specific preference. And definitely, only those users with preferences on certain topic will be adopted in the related subgraph. Formal definition is as follows:

Definition 4 (*Preference-sensitive Subgraph*) The preference-sensitive interaction subgraph of media content c can be described as $G_c^z = \left\langle V_c^z, E_c^z \right\rangle$ for each candidate annotation a_z, here $V_c^z \subseteq V_c$ denotes the users who viewed c with clear preferences on a_z and $E_c^z \subseteq E_c$ denotes the links between them.

Motivated by the partition process, we realize that the viewing records are indeed extracted *iteratively* from the streaming data, which is presented as a list $\{u_i\}$. Here, *"streaming"* means all the records are sorted by their time stamp t_i and then read one after another. By combining with the social network and users interests, we can now build the preference-sensitive interaction subgraphs for each annotation. To be specific, when loading a piece of record, we firstly pick up those related subgraphs according to the users preferences, then check whether the mentioned user follows anyone who exists in these current subgraphs, and finally, connect them with directional edges. The details are shown in the Algorithm 1.

Indeed, some popular media contents may attract millions of viewers, which result in the huge interaction graph and incredible computation cost. Therefore, when reading the streaming viewing records to mine the preference-sensitive subgraphs, we could control the

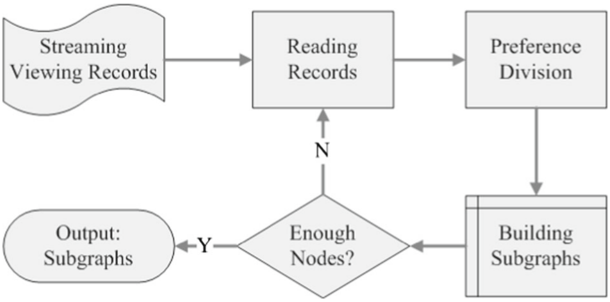


Fig. 2 The process of building preference-sensitive subgraphs

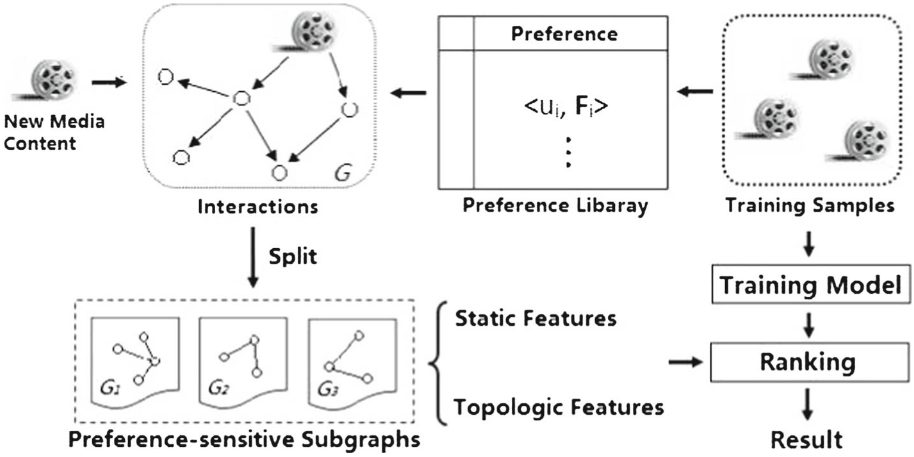


Fig. 3 The flowchart of our novel annotating system

amount of selected nodes to reduce the computational complexity. In this paper, we simply define the stop condition as total amount of selected records, i.e., only the first N viewers will be considered to build the interaction graph. The detailed process is shown in Fig. 2. In the future, some more complex stop conditions, such as the convergence of feature vectors or target function, will be considered.

3.3 Framework of annotating approach

Based on the definitions above, here we formally present the framework of our novel annotating approach with preference-sensitive subgraph analytics. To be specific, our approach is based on the well-known learning-to-rank framework, which contains two different stages, namely *training stage* and *test stage*. Figure 3 illustrates the complete flowchart of our annotating system.

Training stage Given a set of historical media contents with ground-truth labels, we first iteratively build interaction graphs and partition the preference-sensitive subgraphs simultaneously from the streaming records. Then, some discriminative features are extracted for each preference-sensitive subgraph to learn the ranking models.

Test stage Given a new media content c, we first derive preference-sensitive subgraphs $\{G_c^z\}$ for each annotation. Then, we extract features to rank the subgraphs with the well-trained ranking models, which results in the ranked list Λ_c of annotations. Finally, we choose top K annotations in Λ_c as candidates for labeling media content c.

Indeed, the main challenge of our approach is how to extract discriminative features, especially if we hope to achieve the results at the same time when we iteratively read the streaming data. Intuitively, if a subgraph contains more edges and higher weights, the corresponding annotation would contribute more to encourage the interaction process. To that end, in Sect. 4.1.1, we introduce some static features for ranking subgraphs. Also, to deal with the problem that some popular labels, which usually represent hot topics like Hollywood movies, may probably attract more viewers and thus definitely influence the preference estimation, we borrow some goodness metrics [24] for community structure to measure the density of interaction graph, which will be useful to simulate the frequently localized interactions with clear preference, also the real attributes in most cases. Correspondingly, a set of topological features are mentioned in Sect. 4.1.2 for ranking subgraphs.

Another critical problem here is the new media contents may lack sufficient viewing records for building interaction graphs; thus, the performance will be impacted. As mentioned above that we could control the amount of selected nodes to keep the balance between effectiveness and efficiency, in this paper, we will discuss how the limited samples influence the annotating performance. The detailed discussion with experimental results will be shown in Sect. 6.2.2.

4 Learning to annotate with iterative feature extraction

In this section, we introduce some technical details of our ranking scheme, also the feature extraction part. Firstly, two categories of selected features will be summarized, i.e., the static and topologic ones. Furthermore, we will explain how to calculate the features iteratively when we load the streaming data for leveraging state-of-the-art learning-to-rank models.

4.1 Feature extraction for ranking subgraphs

Here, we introduce two kinds of effective features extracted to describe the preference-sensitive subgraphs, namely *static features* and *topologic features*, which reflect different characteristics of corresponding subgraphs.

4.1.1 Static features

The static features denote the basic statistical results for preference-sensitive viewing without graph structure, which can be easily updated with accumulating records. Intuitively, the percentage of preference-sensitive viewing might be the first choice. However, as discussed above, this metric could be easily disturbed by popularity, and thus some other features are extracted as follows:

Average preference This feature intuitively reflects the basic assumption that more users with clear preference leads to higher probability to annotate. Specifically, this feature can be calculated as follows,

$$AVP(G_c^z) = \frac{\sum_{u_i \in V_c^z} f_i^z}{|V_c^z|}. \tag{1}$$

Density of subgraph This feature is built on the intuition that good communities are well connected [7]; in other words, viewers frequently discuss about this content. Generally, if the graph contains as many edges as possible with limited nodes, then the graph has a higher value of density. Specifically, this feature can be calculated as follows,

$$DEN(G_c^z) = \frac{2|E_c^z|}{|V_c^z| \cdot (|V_c^z| - 1)}. \tag{2}$$

Separability In the perspective of community detection in ordinary graphs, this feature presents the idea that good communities are well separated from the rest of the network [6]. Here, we borrow the idea to measure the purity of preference-sensitive connections, i.e., whether majority of interactions occurred within users holding the same certain preference. Specifically, this feature can be calculated by

$$SEP(G_c^z) = \frac{|I(G_c^z)|}{|S(G_c^z)|}, \quad \begin{array}{l} \forall e_{ij} \in |I(G_c^z)|, u_i, u_j \in V_c^z, \\ \forall e_{ij} \in |S(G_c^z)|, u_i \notin V_c^z, u_j \in V_c^z. \end{array} \tag{3}$$

Indeed, as discussed in Sect. 3.3, these features can smooth the inference of annotation popularity to a certain degree. However, since the mass may not have clear preference and thus impact the statistics, thus the graph structure is also needed for more comprehensive features.

4.1.2 Topologic features

As mentioned above, static features may highlight the popular annotations but fail to capture the unpopular ones; thus, the graph structure should be considered to describe the preference-sensitive interactions comprehensively. To this end, we extract several topologic features, which imply the scale and density to measure the level of mutual communications.

Longest path length This feature might be the most appropriate feature to measure the size of latent community. Generally, a longer route indicates a larger latent community, i.e., broader interactions. Here, we define the "path length" as the longest distance between pairwise nodes. Since the interaction graph is DAG as mentioned in Sect. 3.2, distance will be easily achieved via iterative counting. Related details will be introduced in next section.

In/out degree This feature also describes the density of graph, as the node which owns high in/out degrees usually plays the important role as the "hub" or "authority" to encourage the information flow in online social network. For the overall estimation, we both consider the *maximal in/out degree* and the *average in/out degree*. To be specific, only those nonzero degrees will be counted. Related details will be introduced in next section.

Clustering coefficient It is the classic metric to measure the connectivity of community, while indeed based on the counting of triangle structure. Here, *triangle* is defined as if $u_i, u_j, u_k \in V_c^z, e_{ij}, e_{ik}, e_{jk} \in E_c^z$ happens (since u_i, u_j, u_k are read in order from the streaming records, the reverse edge indeed could not exist). Actually, it reflects the locally homogeneous distributions of edges, since usually friends of your friends might be prob-

Algorithm 2 Extracting Features.

Input: a list of viewing records $\{u_i\}_z$;
Store: amount of edges e, a set of nodes $\mathbf{V}_c^z = \{u_i\}, u_i = \langle d_i, o_i, \mathbf{P}_i \rangle, \forall u_i \in \mathbf{V}_c^z$;
Output: longest path length LPL^*, clustering coefficient TCC^*
Initialization: $e = 0, \mathbf{V}_c^z = \emptyset$;

for each u_r in VR_z
  $d_r = 0, o_r = 0, \mathbf{P}_r = \emptyset$;
  for each $u_s \in \mathbf{V}_c^z$
  if $(u_s, u_r) \in E_c$ **then**
    $e++, o_s++, \mathbf{P}_r = \mathbf{P}_r \cup \{u_s\}$;
    if $d_r < d_s + 1$ **then** $d_r = d_s + 1$;
    $TCC^* = TCC^* + |\mathbf{P}_s \cap \mathbf{P}_r|$; // Counting Triangles
  $\mathbf{V}_c^z = \mathbf{V}_c^z \cup \{u_r\}$;

$LPL^* = \max d_i, \forall u_i \in \mathbf{V}_c^z$;
$TCC^* = 3 \times TCC^*/e$;
return LPL^*, TCC^*;

ably your friends especially when you all hold similar preferences. Here, we calculate the coefficient as the triplicate amount of triangles divided by the number of edges as, i.e.,

$$TCC(G_c^z) = \frac{3 \times |\{(i, j, k)|u_i, u_j, u_k \in V_c^z, e_{ij}, e_{ik}, e_{jk} \in E_c^z\}|}{|E_c^t|}. \tag{4}$$

4.2 Iterative features extraction

Now we explain in detail how to extract static and topologic features from streaming viewing records. To be specific, we propose a graph traversal algorithm to refine the features when iteratively loading the records. The details for the preference-sensitive subgraph of a_z is showed in Algorithm 2. In the algorithm, the $\{u_i\}_z$ represents the nodes in the preference-sensitive subgraph, in which each u_i contains three attributes, i.e., 1) the longest distance (to the starter) d_i, 2) the out degree o_i and 3) the set of parent nodes P_i. For simplifying the presentation, here, we only show the output of "Longest Path Length" (LPL) and "Clustering Coefficient" (TCC), while the others could be counted from the variables, e.g., *In/Out Degree* based on o_i and $|P_i|$.

Based on the proposed algorithm, we could calculate the feature vectors at the same time when we read the viewing records step by step, which results in an iterative refinement process. Since no global structure information is needed during this process, we could easily achieve the amount control of reading records to reduce the computational complexity.

5 Further discussion

In this section, we discuss some further issues about the technical framework, including the ranking models selected, the preference learning with multi-source information and finally, the discussion on how to integrate our approach with other annotating approaches.

5.1 Ranking models

After the feature extraction, now we could utilize the learning-to-rank models to rank the preference-sensitive subgraphs and then pick out those significant annotations. In this paper, we take two state-of-the-art models as examples to verify the effectiveness, namely the pairwise based RankBoost [8], and listwise-based AdaRank [22] for annotating, which are both boosting algorithm to iteratively combine weighted weak ranker.

RankBoost is indeed an adaptive method of classical binary classification method AdaBoost. In RankBoost framework, weights are embedded on pairs of samples. To minimize the ranking error, RankBoost iterates as these steps. First, a weak ranker is chosen, and the score will be calculated according to the effectiveness. Then, weights of the entire sample pairs will be updated, i.e., those pairs with incorrect partial order will have a greater weight, which will be focused in next round.

AdaRank is another boosting algorithm whose optimization target is directly defined on the performance measures. In AdaRank, the weight is embedded with a list of query-related samples but not isolated samples. In each step of iteration, firstly, a weak ranker is chosen, and then, the weight for both ranker and queries is updated based on the general performance function, those queries which fail to get a high performance will have a higher weight. Then in next round, those queries will be focused.

With the definitions, we realize that those iteration-based ranking models utilize the weights to regulate the new learners to focus on hard queries. Indeed, some other models, like RankNet [1] or ListNet [3] could also be utilized in our annotating framework to replace the models mentioned above, which leads to the ranking model as a replaceable part of our approach.

5.2 Learning preference factors

As introduced in Sect. 3.1, we could directly derive the preference factors of users from the intuitive statistics, i.e., counting the annotations of media contents viewed before, which is indeed learning the preference factors via the bipartite graph between users and media contents. However, this approach is too straightforward to take consideration of some important issues. First, viewing records might be generated due to various motivations but not only preferences, which may disturb the judgment of preference. Second, the bipartite graph ignores the interaction between users which may influence the preferences. Moreover, some other reasons may also influence the performance, e.g., the additional prior knowledge like user profiles. Based on above points, we develop an optimization framework to refine the preference factors with the extended graph between users and media contents.

In this framework, we denote the set of media contents is presented as C and each $c_k \in C$ represents a piece of media content. Then, based on the social connections as well as the viewing records, we define the edges in extended graph as $E^* = E \cup E'$. Here, E means the connections between users, while E' presents the connection between users and media contents, i.e., viewing behaviors, where $e'_{ik} \in E'$ represents user u_i viewed the media content c_k. Figure 4 shows the structure of the extended graph, where all the edges are directional, and the blue lines means social connection, while the red ones means the viewing behaviors.

For each node $u_i \in V$, we have a normalized $|A|$-dimensional vector $\mathbf{f}_i$ to represent the preference factors, while correspondingly a $|A|$-dimensional vector $\mathbf{l}_k$ for each media content $c_k \in C$ to present the annotations, where $l_k^z = 1$ if c_k is labeled with the zth annotation, otherwise $l_k^z = 0$. For each edge $e'_{ik} \in E'$, we have the weight r_{ik} to present the rating of c_k given by u_i. Also, for each edge $e_{ij} \in E$, we assume that their interactions may reflect their common preferences, and we denote w_{ij} as the weight to present connection strength. There are several ways to define w_{ij}, such as the follow one which indicate the similarity of ratings of the pairwise users:

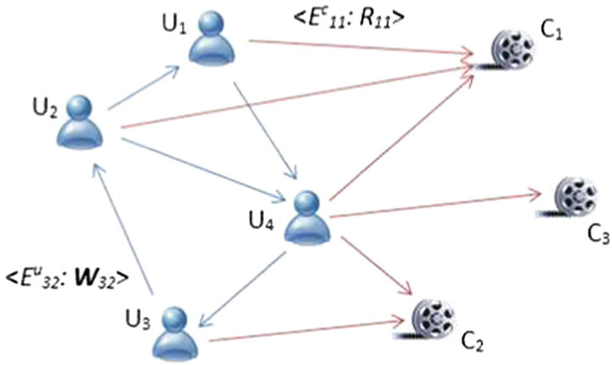


Fig. 4 An illustration of extended user-content graph

$$w_{ij} = \frac{\Sigma_{k \in \mathbf{K}}(r_{ik} \cdot r_{jk})}{|\mathbf{K}|}, \mathbf{K} = \left\{k | e'_{ik}, e'_{jk} \in E'\right\}. \tag{5}$$

With the definitions above, now we define an objective function for preference-factor refinement as follows:

$$O(\mathbf{f}) = \lambda_u \Sigma_{u_i \in V} \|\mathbf{f}_i - \mathbf{f}_i^{(0)}\|_2 + \mu_c \Sigma_{e'_{ik} \in E'} \|r_{ik} \cdot (\mathbf{f}_i - \mathbf{l}_k)\|_2 \\ + \mu_u \Sigma_{e_{ji} \in E} \|w_{ji} \cdot (\mathbf{f}_i - \mathbf{f}_j)\|_2. \tag{6}$$

Specifically, the objective function is formulated as three parts, corresponding to the three constraints to achieve. The first part presents the constraint that the estimation of $\mathbf{f}_i$ and should keep close to the prior knowledge. The other two parts separately present the constraints concerning about the different types of edges in the extended graph. Particularly, the second constraint is actually the variant of intuitive statistics that users' preferences should be reflected by what they have viewed, while the third constraint follows the assumption that since the interactions indicate the common preferences between users, thus the corresponding preference factors should be similar.

To achieve our target to minimize the function $O(\mathbf{f})$, we process the optimization task with two steps in each round. In the first step, we solve the objective function to derive the iterative formulas to update preference distribution. As discussed in Sect. 3.2, we treat all the annotations as mutual independent for easing the modeling; thus, each element of the preference factor (vector) could be calculated as follows:

$$f_i^z = \frac{\lambda_u \cdot (f_i^{(0)})^z + \mu_c \Sigma_{e'_{ik} \in E'} (r_{ik})^2 \cdot l_k^z + \mu_u \Sigma_{e_{ji} \in E} (w_{ji})^2 \cdot f_j^z}{\lambda_u + \mu_c \Sigma_{e'_{ik} \in E'} (r_{ik})^2 + \mu_u \Sigma_{e_{ji} \in E} (w_{ji})^2}. \tag{7}$$

After each round of iteration, all the $\mathbf{f}_i$ will be normalized again as $\|\mathbf{f}_i\| = 1$.

Indeed, if we treat the labels of media contents as prior knowledge but not ground truth, Eq. 6 above could be further extended as:

$$O(\mathbf{f}, \mathbf{k}) = \lambda_u \Sigma_{u_i \in V} \|\mathbf{f}_i - \mathbf{f}_i^{(0)}\|_2 + \mu_u \Sigma_{e_{ji} \in E} \|w_{ji} * (\mathbf{f}_i - \mathbf{f}_j)\|_2 \\ \times \lambda_c \Sigma_{c_k \in C} \|\mathbf{l}_k - \mathbf{l}_k^{(0)}\|_2 + \mu_c \Sigma_{c_i, c_j \in C} \|ss_{ij} \cdot (\mathbf{l}_i - \mathbf{l}_j)\|_2 \\ + \mu_{uc} \Sigma_{e'_{ik} \in E'} \|r_{ik} \cdot (\mathbf{f}_i - \mathbf{l}_k)\|_2 \tag{8}$$

Here, the second line presents the additional constraint of media contents, in which the former one presents the reliability of samples' label and the later one indeed constrains that the similarity between labels of media contents should be close to their semantic relations (presented as ss_{ij} in the formulation) which is based on the textual description. Along this line, we could refine the labels of training samples simultaneously when we learn the preference factors.

It is worth noting that we could only calculate the preference factors for those users who have viewed at least one media contents, since if there is no viewing records, all the r and w will be set as 0 and thus the calculation is unavailable. Though generally speaking, those inactivated users should be deleted when selecting training samples. However, if really needed, in this case we could simply define $f_i^z = 1/|A|, \forall a_z \in A$, or set the value as the average of their friends' preferences.

5.3 Multi-source integration

In this paper, we mainly focus on analyzing the preference-sensitive interaction behaviors within social network and then target at annotating the media contents with few metadata. Since there are many other approaches proposed for labeling the annotations, which depend on different source of prior knowledge, in this subsection, we discuss about how to integrate our approach with the multi-source data for our approach to achieve better annotating performance.

Intuitively, we could treat our approach as one weak classifier and then integrate it with other approach by boosting method, which usually improve the accuracy of annotating. At the same time, we could also directly integrate the prior knowledge in our novel framework. Here, we take the textual description as an example, i.e., the introduction, comments or personalized tags given by end users. The content-based annotating methods have been well studied, mainly concerning about the textual analysis [19] or connecting the low-level keywords with high-level categories [25]. Indeed, the textual information could be integrated seamlessly, especially in the preference learning part.

In the preference learning part discussed in Sect. 5.2, the textual information could be reflected when connecting users and media contents. For example, since users may have personalized tag records, we could utilize topic models to mine the latent topics among the open-vocabulary personalized tags, and then add them as the features for measuring the strength of connection. Similarly, the potential rating will also be estimated through the semantic correlation between user and media contents or within media contents, which could be treated as constraint for the preference-factor estimation. Besides, we could even regard the textual description of media contents as prior knowledge to assume that the label of training samples might be inaccurate or incomplete. These prior knowledge could all be merged into the Eq. 8.

Finally, the textual information could also be utilized to guide splitting the preference-sensitive subgraphs, or combined with features to describe the subgraphs. Also, some other sources of prior knowledge, such as user profile or multimedia characteristics, could also be integrated into the proposed annotating framework following the similar way.

6 Experimental results

In this section, we report the experimental results to validate the effectiveness of our annotating framework. Moreover, case study and some further discussion of experiments will be presented.

Table 2 Statistics of two real-world data sets

Term	Details	Douban Book	Douban Movie
Item	Total Num.	475,820	89,667
	Selected Num.	1,000	1,000
	Avg. Views	1,662.29	3,171.12
	Avg. Interaction	5,558.88	18,045.88
User	Related Num.	70,836	71,372
	Avg. Friends .	74.52	76.49
	Avg. Views	23.47	44.43

6.1 Experimental setup

We perform our experiments on several real-world data sets collected from **Douban.com**, one of the most famous Chinese SNS, which allows users to contribute comments on movies, books and music.[3] should be noted that Douban is indeed a content-based SNS, but not traditional social-based SNS like Facebook, and thus, the interactions here are usually due to the similar preferences but not relationship in real world.

6.1.1 Data statistics

We extracted view logs for more than 6 years via official API of Douban.com. Specifically, two different channels are covered, including 475,820 pieces of books in Douban Book and 89,667 movie documents in Douban Movie, concerning about 100,176 viewers in total.

For each data set, we randomly selected 1,000 samples for the fivefold cross-validation and then labeled them with extracted metadata as ground truth, e.g., nationality and genre. As discussed in Sect. 3.2 that we could hardly achieve the interaction records directly due to the limitation of data source, and thus, we extracted the viewing records as well as social network information to build the interaction graph. The related statistics of samples are shown in Table 2. We could clearly find that the average number of views for books is much less than movies, which is reasonable since readers may spend several days on one book but only a few hours for a movie.

6.1.2 Baseline approaches

To the best of our knowledge, few works take into account the annotating with graph-formed interaction records; thus, we mainly focus on the feature and model comparison in our approach. Furthermore, three practical benchmark methods will be leveraged to evaluate the performance of our approach.

Common-Interest-based Diffusion Maximization [23] (*CIDM*). This is also a social-based annotating approach which follows the intuitive assumption that the users' sharing behaviors are attributed to their common interests. Thus, if the proper annotation is revealed, we could "reproduce" the diffusion process within users on the basis of the common-interest-based social diffusion simulation. Indeed, this approach acts as the integration of traditional social

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

diffusion (influence) model and the preference factors of social media platform, which is similar with our proposal.

Voting of Users' Preferences. This approach is based on the assumption that the aggregated preferences of all viewers may indicate the feature of media content; here, the preference factors $\mathbf{f}_i$ for user u_i are actually learned in the Sect. 5.2. Then, given a media content c, all the viewers vote with their own preferences and ratings, and the top k tags ranked by $\sum_i (\mathbf{f}_\mathbf{i}^\mathbf{z} \cdot r_i^c)$ will be treated as annotating result.

Interaction-based Bayesian approach. It is a naive Bayesian approach based on the analysis of interaction process, which transfer the annotating task as a maximal posteriori probability estimation problem as follows,

$$\begin{aligned} P(a_z \mid c) \propto P(a_z \mid G_c) &\approx \prod_{s,r:e_{sr} \in E_c} P(a_z \mid P(\langle u_s, u_r \rangle)) \\ &= \prod_{s,r:e_{sr} \in E_c} \frac{P(\langle u_s, u_r \rangle \mid a_z) P(a_z)}{\sum_z P(\langle u_s, u_r \rangle \mid a_z) P(a_z)} \end{aligned} \tag{9}$$

Indeed, the first line of the formulation intuitively utilizes the interaction graph to describe characteristics of the given media contents, which follows the similar assumption with our approach. And then, the posteriori estimation is transferred into the form of Bayes formula, in which the likelihood $P(G_z \mid a_z)$ presents the occurrence probability of the interaction graph G_c that when the annotation a_z is given.

In our approach, the probability is calculated with the ranking score of each preference-sensitive subgraph. As a baseline, here we simply treat all the edges in E_c, i.e., the interaction behaviors are mutual independent. Thus, the computation is eased as shown in the second line of the Eq. (9), in which $P(\langle u_s, u_r \rangle \mid a_z)$ could be estimated as the frequency of a_z within the interaction between u_s and u_r, or directly borrow the value w_{sr}^z in Eq. (5) in Sect. 5.2.

To some certain degree, the Bayesian approach could be treated as an adaptive voting for isolated interaction. We design this baseline method to compare the performance based on individual or interaction, and further, to measure the effectiveness of our novel annotating approach based on the analytic of interaction with graph structure.

6.1.3 Evaluation metrics

To evaluate the performance of each approach, several common-used metrics in information retrieval are leveraged. **Overall Precision@**K ($P@K$) presents how many annotations in the top K lists are proper to label the given media content in average. Also, **Overall Recall@**K ($R@K$) presents how many annotations in ground truth are discovered in the top K list in average. The *overall* result here means the average measures for all the test samples. **F-measure Score** ($F@K$) presents the trade-off between $P@K$ and $R@K$, which is formulated as $F@K = \frac{2 \times P@K \times R@K}{P@K + R@K}$.

In addition, for measuring the order of correct answers, the **Mean Average Precision** (MAP) and the **Normalized Discounted Cumulative Gain** (NDCG) are also selected. MAP stands for the mean result of Average Precision (AP) of all the test samples, where AP is calculated as

$$AP = \sum_{i=1}^{K} \frac{P@i \cdot \delta\{c_i \textit{ is correct}\}}{\#\{\textit{correct answers in Top i results}\}},$$

where $\delta\{*\}$ is a boolean function and $\delta\{*\} = 1$ if $*$ is true, otherwise $\delta\{*\} = 0$, and c_i presents the ith result in the top K list. Also, the NDCG is defined as

$$N = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log(1 + j)},$$

where $r(j)$ means the relevant value of the returned jth annotation, while a correct annotation corresponds 1 and a wrong one leads to 0. Z_n presents the normalized factor, which is calculated by the maximal DCG results given n. Intuitively, higher value of both MAP and NDCG means better performance, and we also utilize the two metrics to evaluate the performance of the weak rankers of AdaRank approach.

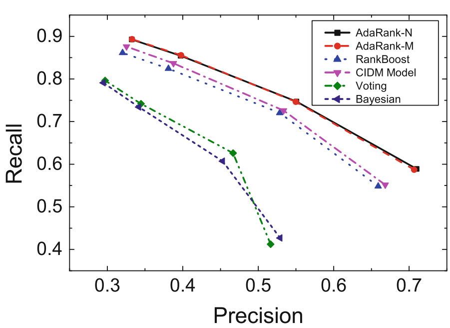
6.2 Experiments

To reduce the uncertainty of sample selection, we utilize a standard fivefold cross-validation to evaluate our annotate framework in the experiments. To be specific, we first randomly divide the data set into five equal parts and then use each part as the test data while the other four parts as the training data in five test rounds. Since the average number of labels for each media content is around 3 according to ground truth, and majority of samples contain less than 5 labels, we set four different values of K as 3, 5, 8 and 10 in the former two series of experiments and treat *NDCG@5* as one of the optimization targets for AdaRank algorithm.

6.2.1 Overall results

First of all, we verify the overall performance of our approaches with complete features and samples of complete viewing records, in which the isolated nodes are filtered. The experimental results are shown in Fig. 5, Tables 3 and 4. Particularly, AdaRank-N and AdaRank-M are the AdaRank algorithms with $NDCG$ and MAP for evaluating weak rankers, respectively. The figures describe the P–R curve of all the five approaches in two data sets, and the tables present the measures of $MAP@K$ and $NDCG@K$. Since the results derived from the AdaRank with two different optimization tasks are quite close, the result lines in figures are almost overlapping with each other.

From the experimental results, we could clearly find that our approaches can effectively discover the proper labels, and consistently outperform the traditional baselines about 20–50 %, especially when K is relatively smaller (e.g., when $K = 3$, the improvement of MAP



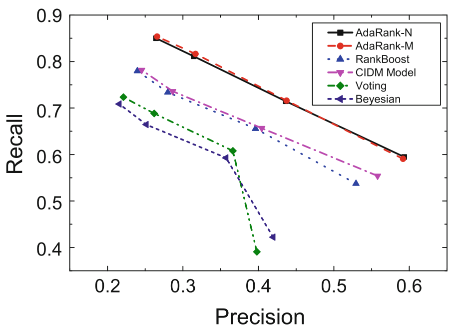


Fig. 5 P–R curve for overall result with different K: **a** Douban Book, **b** Douban Movie

Table 3 Overall result with different K for Douban Book

Model	AdaRank-N	AdaRank-M	RankBoost	CIDM	Voting	Bayesian
	MAP					
10	**0.7374**	0.7365	0.6892	0.7025	0.5684	0.5590
8	**0.7242**	0.7233	0.6754	0.6885	0.5505	0.5407
5	**0.6715**	0.6691	0.6282	0.6358	0.5013	0.4872
3	**0.5631**	0.5606	0.5175	0.5211	0.3782	0.3796
	NDCG					
10	**0.8265**	0.8256	0.7854	0.8017	0.6927	0.6850
8	**0.8093**	0.8085	0.7682	0.7839	0.6681	0.6596
5	**0.7529**	0.7510	0.7143	0.7263	0.6093	0.5949
3	**0.6524**	0.6497	0.6055	0.6165	0.4760	0.4813

The bold ones mean the best results

Table 4 Overall result with different K for Douban Movie

Model	AdaRank-N	AdaRank-M	RankBoost	CIDM	Voting	Bayesian
	MAP					
10	0.6677	**0.6702**	0.5886	0.6152	0.4590	0.4460
8	0.6557	**0.6588**	0.5754	0.6016	0.4490	0.4338
5	0.6173	**0.6189**	0.5465	0.5717	0.4176	0.4076
3	**0.5476**	0.5460	0.4801	0.5117	0.3187	0.3292
	NDCG					
10	0.7725	**0.7756**	0.7060	0.7326	0.5951	0.5814
8	0.7558	**0.7596**	0.6862	0.7126	0.5801	0.5626
5	0.7078	**0.7098**	0.6479	0.6735	0.5405	0.5282
3	**0.6363**	0.6354	0.5772	0.6116	0.4155	0.4282

The bold ones mean the best results

even achieves as 72 % for AdaRank than Voting). Since the results in the top list are the most crucial ones, the improvements of our approaches seem to be more significant. Even for the state-of-the-art CIDM approach, our method could outperform for 5–20 % with AdaRank and almost the same with RankBoost. What should be noted here is that based on the same interaction graph, the CIDM approach averagely annotate each media content using *nearly one second*, while our novel approach could do better within only **a few milliseconds**, which prove the efficiency of our approach.

For better proving the performance, we introduce the *T test for paired samples* to validate the comparison of the fivefold experiments. The results indicate that our approach **significantly** outperforms the baselines with P much less than 0.01 in 99 % confidence (except for that comparison between RankBoost and CIDM with P around 0.5 which means no obvious difference).

At the same time, we realize that the AdaRank outperforms the RankBoost, which validates that the listwise ranking models might be better than pairwise ranking models in this application scenario. It may be due to the optimization target that listwise models directly focus on mining the best results, but not correct preference pairs.

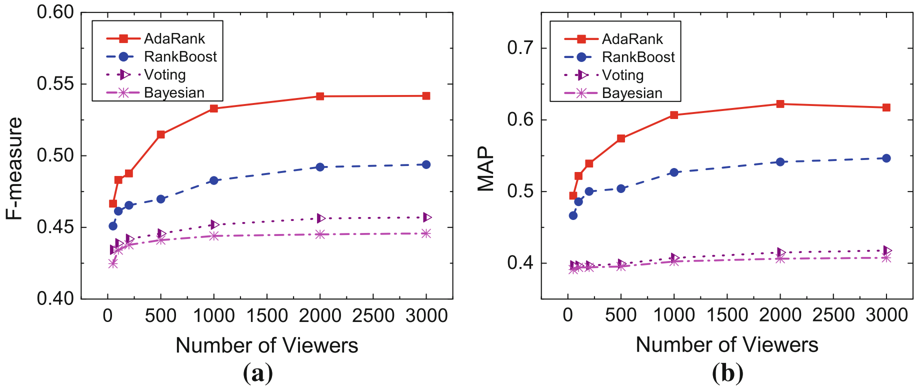


Fig. 6 Comparison between different size of records: **a** F-measure, **b** MAP

6.2.2 Different size of records

To measure if our approach is adequate for the incomplete interaction graph, and further, checking the effectiveness with the "cold-start" problem, we would like to confirm the performance with limited viewing records. To this end, we execute the experiments with different size of viewing records. We cut the records based on the time stamp and set a threshold N, i.e., only N nodes exist in the filtered interaction graph.

We verify the performance of our approach with complete features on two data sets, the results on Douban Movie data set are shown in Fig. 6, while the results for Douban Book are omitted due to the similar results trend. Here, we take the performance of F-measures and MAP with $K = 5$ as examples. We notice that our approach does not suffer a severe impact when the viewers are more than 1,000; however, the results become poor with less than 100 viewers. Obviously, the performance is reliable when the records are enough to build clear graph structure. However, when the graph structure is too much sparse, it will be difficult to distinguish the significant subgraphs. Nevertheless, we realize that our approaches still perform better than benchmark methods for more than 10 % even with tens of viewers. At the same time, we can clearly find that the results of benchmark methods keep relatively stable with decreasing the viewers' amount, it might because of the early viewers' preferences can better reflect the properties of media contents.

6.2.3 Different set of features

In this subsection, we try to analyze the effect from different features in our approaches, namely static features and topologic features. We execute the experiments on two data sets with two different ranking models and three different sets of features: (1) the complete feature set, (2) static features only and (3) topologic features only. The results for Douban Movie data set with AdaRank and RankBoost are shown in Fig. 7, respectively. Since the trend is similar in Douban Book data set, the corresponding figures are omitted.

According to the result, we can find that the topologic-based ranking performs very well when $K = 10$, while becomes worse when K decreases. On the contrary, the static-based ranking is much better when K is smaller. Based on the observations, we believe that the results coincide with our purpose of extracting topologic features, i.e., to reduce the interference of popular topics and discover the cold ones. Generally, with the topologic features,

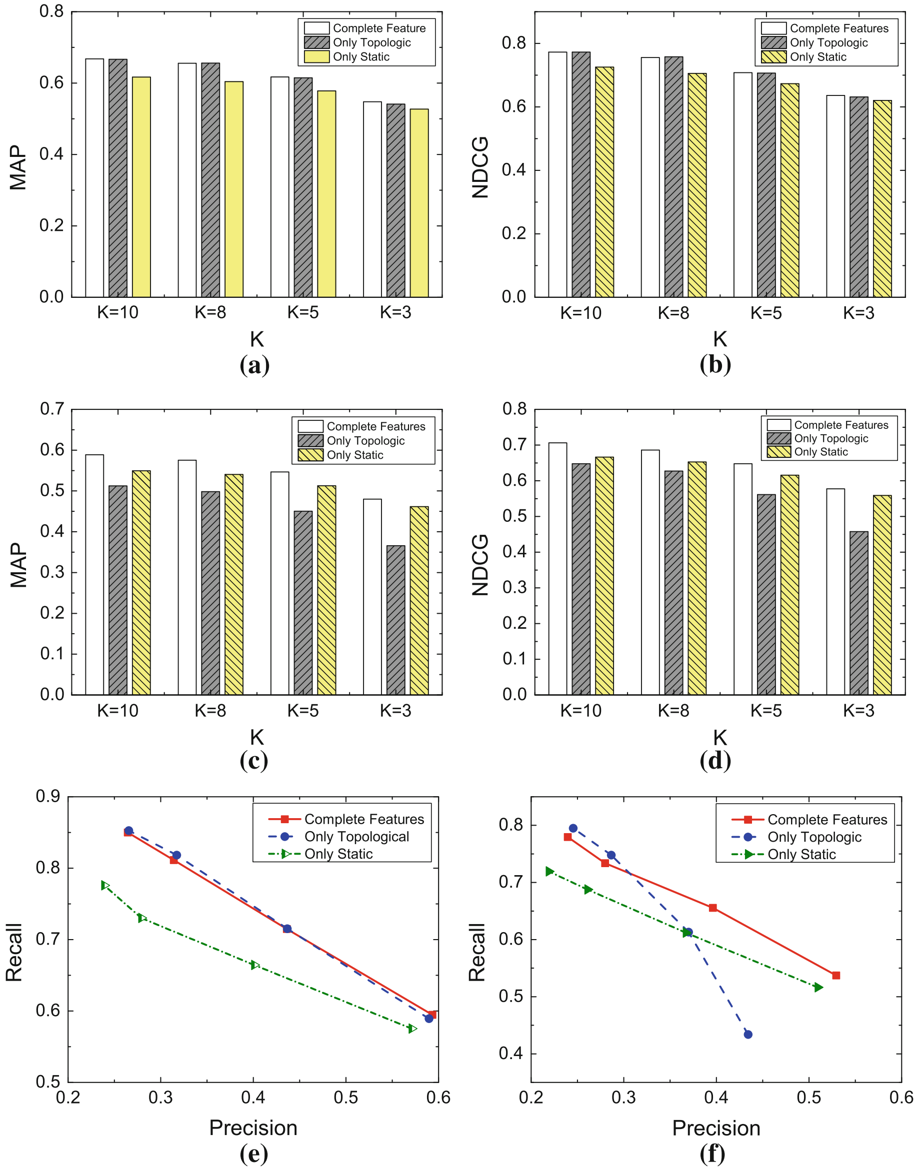


Fig. 7 Comparison between different features. **a**, **b** and **e** separately represents the MAP, NDCG and P–R Curve for AdaRank, and **c**, **d** and **f** for RankBoost

we could discover more adequate unpopular labels, but it is hard to push them into the head of list. On the contrary, the static features could easily promote the ranking of popular ones, especially for those correct ones. Combining the advantages of the both sides, the ranking models based on complete features perform well and stable when K varies.

At the same time, we realize that topologic features perform quite different with the two ranking models, which could be a nice ranker to correctly list the candidate, but fail to describe the proper pairwise relationship of annotations. Thus, for AdaRank algorithm, the topologic

Table 5 Two cases of tagging results

Name	*The Star War*	*Kagemusha*
True labels	**United States, Sci-Fi** **Action, Classic**	**Japan, War** **History, Classic**
AdaRank	**United States, Classic, Action** **Sci-Fi**, Comedy, Drama Horror, War	**Classic**, Drama, **Japan** United States, **War**, Love Comedy, **History**
RankBoost	**United States, Action, Classic** Comedy, **Sci-Fi**, Love Animation, Hongkong	United States, **Classic, Japan** Drama, Action, Comedy Love, **War**
Voting	**United States**, Comedy, **Action** **Classic**, Love, Animation **Sci-Fi**, Drama	United States, **Classic**, Comedy Love, Action, Drama **Japan**, Animation
Bayes	**United States**, Comedy, **Action** Love, **Classic**, Animation Drama, Hongkong	United States, Love, Comedy **Classic, Japan**, Drama Action, Hongkong

The bold ones mean the correct answers

features contribute the most, while the results are relatively worse when K is smaller for RankBoost.

6.3 Case study

To better understand the performance, two movies are selected from the Douban Movie data set for case study. One is the famous science fictional movie series "*The Star War*", while another is a classic Japanese movie namely "*Kagemusha*". The annotating results are shown in Table 5, in which the ground truth are first listed out, followed by the top 8 results given by different approaches. Obviously, in both cases, our approaches generate better results compared to baselines, since almost all the ground-truth tags are retrieved and ranked at prior positions.

With analysis in detail, we may have two interesting findings. First, generally the most popular ones will be much easier to be discovered, e.g., the "United States" is usually ranked in the first position. At the same time, the ordinary ones (e.g., "Classic" or "Action") can be found in lower positions and those unpopular ones, such as "War" or "History" whose reappearance might be less than 10 % of the tag "United States," could be only found by our approaches. Second, with deep looking into these two cases, we can guess that for the benchmark algorithms, the results may follow a certain order related to the popularity, for example, we observe that the tags "Comedy," "Love" and "Animation" are recommended to both movies, though they are completely irrelevant. Based on the two findings, we may conclude that the popularity of labels will severely disturb the results. At the same time, our approaches are able to retrieve those less frequent but correct ones.

6.4 Discussion

In this section, we summarize the reasons why our novel annotating framework could achieve good performance only with the interaction analytics. Naturally, the effects of learning-to-rank techniques themselves play an important role, but not the most important one. According to our basic assumption that the preference-sensitive interaction within online social network may indicate the attribute of media contents, thus the interaction records might be better to support the automatic annotating task, especially for those localized frequent interactions. However, based on the experimental results, we observe that the common preferences for isolated pairwise viewers might not be better than individual interests when describing the media content's attributes due to the interference of various interaction motivations.

By deeply analyzing the experimental results, we realize that in the comparison between different set of features, our approaches with only static features are still outperform the baselines, though the improvement is quite limited. What should be noted that the static features only consider the pairwise preference; thus, it is fairly close to the baseline of voting, which may indicate that the preference-sensitive filtering indeed makes sense to present the real favors. At the same time, though the results based on only topologic features might be even worse in some conditions, the results become dramatic and stable when combined with static features. Thus, it seems that the annotating approaches based on interaction analysis are recommendable mainly due to the graph structure. Obviously, the performance of CIDM approach could also support this announcement.

Some more findings could further support above conclusion. For one thing, we notice that the results based on only topologic features even perform better than the complete feature set when $K = 10$, which indicates that the topologic features would be beneficial to find those unpopular but proper labels. For another thing, when the size of viewing logs is limited, the results of our approach becomes worse especially when only tens of viewers concern about this media content, definitely it validates that a complete graph structure is crucial to ensure the effectiveness.

This viewpoint inspires us that the administrators of social media service should focus on the latent community mining, or at least pay more attention on those existing preference-based communities. It is because that the interactions within those online communities usually reflect clear preference on certain topics, which could be beneficial for media content analysis.

7 Conclusion

In this paper, we proposed a new automatic annotating framework based on the analysis of interaction behaviors within online social network. Specifically, for the given media content, we iteratively load the streaming viewing records to built the preference-sensitive subgraphs for each candidate annotation. Then, we extracted two types of features, namely static and topologic features to measure the scale, density and separability of preference-sensitive subgraphs. Finally, all these features would be integrated into a "learning-to-rank" framework to rank the candidates. Particularly, an unique perspective of our approach is that it is possible to integrate social interactions with multiple information sources for annotating media contents. Experimental results with two state-of-the-art learning-to-rank techniques AdaRank and RankBoost showed that our approach could effectively generate high-quality annotations and thus demonstrated the effectiveness of social interaction analysis in labeling media contents.

In the future, we will focus on several potential directions along this line. First, we will further study the technical details for the integration of multi-source prior knowledge for annotating, especially with the short textual information. Second, we will discuss the target function of ranking models to aggregate the streaming graph structure; thus, the steps of feature extracting and subgraphs ranking could be merged to simplify the modeling. Finally, we will design the self-adapted stop condition for reading the streaming records, to optimize the balance between effectiveness and efficiency.

Acknowledgments The work described in this paper was supported by grants from The National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), The Natural Science Foundation of China (Grant No. 61073110), The Natural Science Foundation of China (Grant No. 71329201), The Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20113402110024), The National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant No. 2012BAH17B03) and The National Science Foundation (Grant No. IIS-1256016). Finally, thanks for Dong Liu's help during the technical discussion.

References

1. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on machine learning (ICML), pp 89–96
2. Cao Y, Xu J, Liu T, Li H, Huang Y, Hon H (2006) Adapting ranking svm to document retrieval. In: Proceedings of the 29th international ACM conference on Research and development in information retrieval (SIGIR), pp 186–193
3. Cao Z, Qin T, Liu T, Tsai M, Li H (2007) Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th international conference on machine learning (ICML), pp 129–136
4. Chen Z, Cao J, Song YC, Guo J, Zhang Y, Li J (2010) Context-oriented web video tag recommendation. In: Proceedings of the 19th international conference on world wide web (WWW), pp 1079–1080
5. Crammer K, Singer Y (2001) Pranking with ranking. In: Advances in neural information processing systems (NIPS), pp 641–647
6. Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci (PNAS) 104(1):36–41
7. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
8. Freund Y, Iyer R, Schapire R, Singer Y (2003) An efficient boosting algorithm for combining preferences. J Mach Learn Res 4:933–969
9. Herbrich R, Graepel R, Obermayer K (1999) Large margin rank boundaries for ordinal regression. In: Advances in Neural Information Processing Systems (NIPS), pp 115–132
10. Heymann P, Ramage D, Garcia-Molina H (2008) Social tag prediction. In: Proceedings of the 31st annual international ACM conference on research and development in information retrieval (SIGIR), pp 531–538
11. Kaplan AM, Haenlein M (2010) Users of the world, unite! the challenges and opportunities of social media. Bus Horiz 53(1):59–68
12. Körner C, Kern R, Grahsl HP, Strohmaier M (2010) Of categorizers and describers: an evaluation of quantitative measures for tagging motivation. In: Proceedings of the 21st ACM conference on Hypertext and hypermedia (ACM HT), pp 157–166
13. Lewis K, Gonzalez M, Kaufman J (2012) Social selection and peer influence in an online social network. Proc Natl Acad Sci (PNAS) 109(1):68–72
14. Liu D, Hua X, Yang L, Wang M, Zhang H (2009) Tag ranking. In: Proceedings of the 18th international conference on world wide web (WWW), pp 351–360
15. Nallapati R (2004) Discriminative models for information retrieval. In: Proceedings of the 27th international ACM conference on research and development in information retrieval (SIGIR), pp 64–71
16. Noh T, Park S, Yoon H, Lee S, Park S (2009) An automatic translation of tags for multimedia contents using folksonomy networks. In: Proceedings of the 32nd international ACM conference on research and development in information retrieval (SIGIR), pp 492–499
17. Qi G, Hua X, Rui Y, Tang J, Mei T, Zhang H (2007) Correlative multi-label video annotation. In: Proceedings of the 15th international conference on Multimedia (ACM MM), pp 17–26
18. Silva A, Martins B (2011) Tag recommendation for georeferenced photos. In: Proceedings of the 3rd ACM SIGSPATIAL international workshop on location-based social network (LBSN), pp 57–64
19. Song Y, Zhuang Z, Li H, Zhao Q, Li J, Lee W, Giles CL (2008) Real-time automatic tag recommendation. In: Proceedings of the 31st international ACM conference on research and development in information retrieval (SIGIR), pp 515–522
20. Toderici G, Aradhye H, Pasca M, Sbaiz L, Yagnik J (2010) Finding meaning on YouTube: tag recommendation and category discovery. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3447–3454
21. Wu L, Yang L, Yu N, Hua X (2009) Learning to tag. In: Proceedings of the 18th international conference on world wide web (WWW), pp 361–370

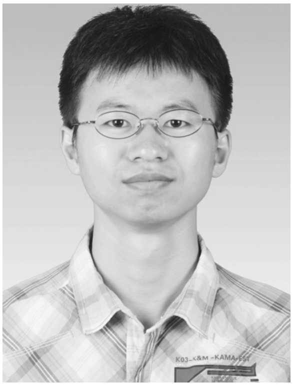
22. Xu J, Li H (2007) Adarank: a boosting algorithm for information retrieval. In: Proceedings of the 30th international ACM conference on research and development in information retrieval (SIGIR), pp 391–398
23. Xu T, Liu D, Chen E, Cao H, Tian J (2012) Adarank: a boosting algorithm for information retrieval. In: Proceedings of proceedings of the 12th IEEE international conference on data mining (ICDM), pp 1158–1163
24. Yang J, Leskovec J (2012) Defining and evaluating network communities based on ground-truth. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), pp 745–754
25. Yin Z, Li R, Mei Q, Han J (2009) Exploring social tagging graph for web object classification. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 957–966
26. Yue Y, Finley T, Radlinski F, Joachims T (2007) A support vector method for optimizing average precision. In: Proceedings of the 30th international ACM conference on research and development in information retrieval (SIGIR), pp 271–278
27. Zhu H, Chen E, Cao H (2011) Finding experts in tag based knowledge sharing communities. In: Proceedings of the 2011 international conference on knowledge science, engineering and management (KSEM), pp 183–195
28. Zhu H, Cao H, Chen E, Xiong H (2013) Mobile App Classification with Enriched Contextual Information. IEEE Trans Mobile Comput (TMC) (pre-print). doi:10.1109/TMC.2013.113

Author Biographies

Tong Xu is currently a PhD student at the School of Computer Science, University of Science and Technology of China. He received a BS degree from Zhejiang University, China in 2010. His research interests include social network and social media analysis, recommender system and mobile data mining. He has published several papers in refereed conference proceedings such as ICDM and PKDD.

Hengshu Zhu is currently a PhD student in the School of Computer Science, University of Science and Technology of China (USTC), China. He received a BE degree in Computer Science from USTC, China in 2009. His research interests include mobile data mining, recommender systems and social networks. During his PhD study, he received the KSEM-2011 and WAIM-2013 Best Student Paper Award, the National Scholarship for Graduate Students of China in 2013, and the Scholarship Award for Excellent Doctoral Students of China in 2012. He has published a number of papers in refereed journals and conference proceedings, such as IEEE TMC, ACM TIST, WWW Journal, ACM CIKM and IEEE ICDM.

Enhong Chen is a full Professor of School of Computer Science and Technology at University of Science and Technology of China (USTC), China. He received the PhD degree from USTC, China in 1996. He has been active in areas of data mining, social network analysis and personalized recommendation systems. He served on over 30 program committees of international conferences including KDD, AAAI, ICDM, SDM and PAKDD. His research has been supported by the Natural Science Foundation of China, Nokia, Huawei Inc, Alibaba Inc. He has published over 100 research papers in refereed international journals and conference proceedings including IEEE TKDE, IEEE TMC, IEEE TSMCB, ACM TIST, KDD, SIGIR, IJCAI, WWW, NIPS, ICDM, CIKM and PKDD. He received the KDD-2008 Best Application Paper Award and the ICDM-2011 Best Research Paper Award.



Baoxing Huai is currently a PhD student at the School of Computer Science, University of Science and Technology of China. He reveived a BE degree from Hefei University of Technology, China in 2009. His research interests include mobile context modeling and named entity mining.

Hui Xiong is currently an Associate Professor and Vice Chair of the Management Science and Information Systems Department at the Rutgers, the State University of New Jersey, where he received the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), and the ICDM-2011 Best Research Paper Award (2011). He received the BE degree from the University of Science and Technology of China (USTC), China, the MS degree from the National University of Singapore (NUS), Singapore, and the PhD degree from the University of Minnesota (UMN), USA. His research interests include developing effective and efficient data analysis techniques for emerging data-intensive applications. He has published prolifically in refereed journals and conference proceedings (3 books, 40+ journal papers, and 60+ conference papers). He is a co-Editor-in-Chief of Encyclopedia of GIS, an Associate Editor of IEEE TKDE and KAIS journal. He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the KDD-2012 and a Program Co-Chair for the ICDM-2013.

Jilei Tian received his BS, MS degrees in Biomedical Engineering from Xi'an Jiaotong University, China and PhD degree in Computer Science from University of Eastern Finland, in 1985, 1988 and 1997, respectively. He joined Beijing Jiaotong University faculty during 1988–1994. He joined Nokia Research Center as senior researcher since 1997, then as principal scientist and research leader, primarily in the area of spoken language processing and recently on rich context data modeling, big data processing and personalized services. He has authored more than 100 publications including book chapter, journal and conference papers. He has also about 100 patents including pending. He has served as member of technical committee and the editorial board of international conferences and journals.