Qu W, Song KS, Zhang YF *et al.* A novel approach based on multi-view content analysis and semi-supervised enrichment for movie recommendation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(5): 776–787 Sept. 2013. DOI 10.1007/s11390-013-1376-7

A Novel Approach Based on Multi-View Content Analysis and Semi-Supervised Enrichment for Movie Recommendation

Wen Qu (屈 雯), *Student Member*, *CCF*, Kai-Song Song (宋凯嵩), *Student Member*, *CCF*
Yi-Fei Zhang (张一飞), *Member*, *CCF*, Shi Feng (冯 时), *Member*, *CCF*, Da-Ling Wang (王大玲), *Member*, *CCF* and Ge Yu* (于 戈), *Senior Member*, *CCF*, *Member*, *ACM*, *IEEE*

School of Information Science and Engineering, Northeastern University, Shenyang 110819, China

E-mail: quwen@research.neu.edu.cn; songkaisongabc@126.com; {zhangyifei, fengshi, wangdaling, yuge}@mail.neu.edu.cn

Received May 5, 2013; revised August 14, 2013.

Abstract Although many existing movie recommender systems have investigated recommendation based on information such as clicks and tags, much less efforts have been made to explore the multimedia content of movies, which has potential information for the elicitation of the user's visual and musical preferences. In this paper, we explore the content from three media types (image, text, audio) and propose a novel multi-view semi-supervised movie recommendation method, which represents each media type as a view space for movies. The three views of movies are integrated to predict the rating values under the multi-view framework. Furthermore, our method considers the casual users who rate limited movies. The algorithm enriches the user profile with a semi-supervised way when there are only few rating histories. Experiments indicate that the multimedia content analysis reveals the user's profile in a more comprehensive way. Different media types can be a complement to each other for movie recommendation. And the experimental results validate that our semi-supervised method can effectively enrich the user profile for recommendation with limited rating history.

Keywords movie recommendation, feature extraction, multi-view, multimedia content analysis, personalization

1 Introduction

With the rapid progress of Internet technology, it is convenient to find, review and share large-scale visual data on the Web. But the explosive growth of data leads to information overload when users want to find interesting items, which brings in significant opportunities for personalized recommendations. The recommender system on visual data has become an essential tool for information retrieval and content discovery in today's information-rich environment. Movies as a kind of visual data are popular on many social media networks, such as YouTube and YouKu. The recommendation on movies intends to find the preferences of users in an accurate manner, to improve the service of the websites and benefit the users.

Given some movies rated by a user, the content-based recommender system aims to rate a new movie based on the rating histories and content relevance. Current movie recommender systems generally employ the metadata (e.g., tags[1-2], genre[3], cast[4]) to capture the content of movies for recommendation. However, the metadata fails to describe movies' multimedia features, such as visual experience or music rhythm of movie videos. On the other hand, the recommender systems based on metadata content tend to produce recommendations with a limited degree of novelty. For example, the system based on genre will recommend just the same genre movies when a user has high rating on "comedy". Hence, we aim to explore the multimedia content of movies to reveal the user's underlying interest and leverage it to find movies targeted to the user.

Most content-based methods avoid video content analysis for the following reasons: 1) Massive storage space is required for movie videos; 2) The cost of collecting all the movie videos is expensive; 3) Video content analysis is time consuming, which includes shot boundary detection, key frame selection, and object

Regular Paper
Supported by the National Basic Research 973 Program of China under Grant No. 2011CB302200-G, the Key Program of National Natural Science Foundation of China under Grant No. 61033007, the National Natural Science Foundation of China under Grant Nos. 61100026, 60973019, and the Fundamental Research Funds for the Central Universities of China under Grant Nos. N110604003, N100704001, N100304004, N120404007.
*Corresponding Author

recognition. Thus, it is usually infeasible to directly analyze movie video content for recommendation. However, most of today's online movie sites (e.g., imdb.com, douban.com) offer abundant multimodal data about movies such as posters, photos, storylines, and sound tracks, which can be the cues for providing visual content, plot and music content of the movies. Fig.1 is a sample webpage for the movie "Golden Eye" on imdb.com, which has many images and text data about the movie. We believe that the relevance between two movies should be described not only based on relevance of metadata but also based on visual, textual and aural relevance.

Fig.1. Sample webpage for the movie "Golden Eye (1995)" on imdb.com.

Moreover, most previous work has the assumption that a sufficient collection of user ratings or profiles is available. However, in real scenarios, there are many casual users providing limited rating items (only one to ten movies are viewed), which will hinder the recommender system to generate effective recommendations. In this paper, we propose a multi-view semi-supervised recommendation algorithm, which describes movies in multiple views and enriches the profiles of casual users who provide few rated histories.

In order to recommend items in a multimodal setting, our proposed method relies on the framework of multi-view learning. In a parallel corpus of multimodal data for a movie, we consider each type of media as a separate view of a movie. Existing work considers multi-view learning as an efficient way to learn classifiers, and in this paper, our work extends multi-view learning to recommendation and applies it for recommending movies. Fig.2 is the overall framework of our approach. First, the multimedia information about movies is crawled from websites. Then these data are analyzed and represented with multi-view features. The single view score assignment is executed to correct outliers in each single view. Based on the user's profile, a semi-supervised enrichment process is carried out to get an enriched profile. Finally, for a new movie, the multi-view recommendation is adopted to fuse the results from each view.

In this paper we aim to focus on three critical issues of a movie recommender system: 1) extract proper and effective features from different views (textual, visual and aural views) to describe movie data; 2) enrich the profiles of users who only have few user histories; 3) fuse the multi-view information and provide personalized recommendation.

The rest of this paper is organized as follows. Section 2 depicts related work on movie and video recommendation. Section 3 presents the feature extraction and representation of each view. Section 4 details our proposed method. The experimental results are given in Section 5. Finally we conclude our work in Section 6.

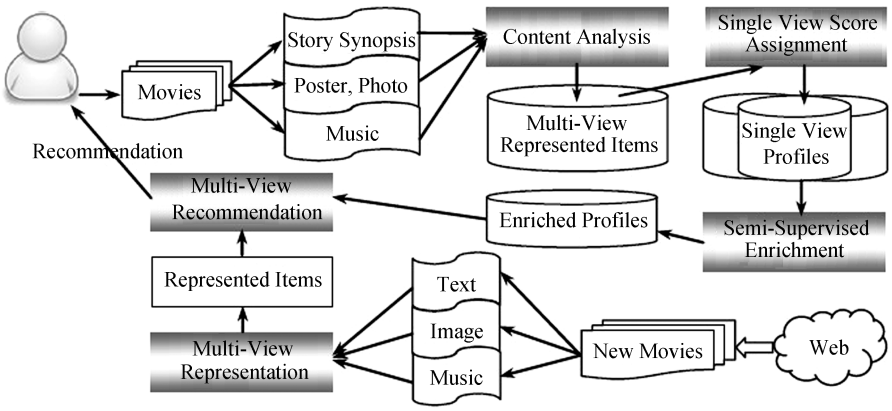


Fig.2. Framework of our approach. The multimedia data are crawled from websites and represented with multi-view items. Then the system recommends new movies to the user based on multi-view content.

2 Related Work

In this section, we review the state-of-the-art literatures on personalized video, movie, and multimedia item recommendation, video visualization, multi-view learning, and multi-view ranking.

The recommendation system has proceeded along three major dimensions, namely, content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering that combines the above two approaches.

The personalized video recommendation system focuses on recommending a list of videos on websites. The system provides recommendation based on the user's operation such as clicking, viewing, favoriting/liking or subscribing a video. Davidson *et al.*[5] made use of association rule mining on click history to find related videos for recommendation. Park *et al.*[1] constructed the user profile as an aggregate of tag clouds and recommended online videos according to users' viewing patterns. Chen *et al.*[6] explored the queries issued by users to suggest videos on an integrated tripartite graph of the query, user, and video. Besides click-through, tag, and query information, there is some work based on social relationship and real-time web (RTW). Zhao *et al.*[7] proposed a multi-task rank aggregation method to integrate ranks based on profile, history, social network and collaborative filtering. Esparza *et al.*[8] described a number of experiments of using RTW data source like Twitter to recommend movies. VideoReach presented by Mei *et al.*[9] is similar to our work in using multimodal relevance between videos, but their work aims to present an online video recommender system.

Compared with video recommendation, recommendation on movies focuses on predicting the rating values of new movies based on previous ratings given by users. The information of movie genres, reviews and actors are mostly considered in the recent work. Manzato[3] recommended a movie based on a user-genre model which could handle the sparseness of traditional factorization method. Qumsiyeh and Ng[4] proposed a multimedia recommender system MudRecS according to users' ratings, genres, reviews and other information. As the development of Web 3.0 and social network, the context-aware movie recommendation has attracted substantial attention over the past few years. Contextual factors, such as social networks[10-11], the company of the user being in [12], mood[13], and temporal information[14-15] are integrated into the recommendation process. Since this kind of information may be sensitive, it is unsuitable in realistic settings when most users want to keep their privacy.

Video visualization addresses the challenges of representing and browsing video data efficiently. Chen *et al.*[16] considered vision and audio features to get semantic visual storylines of a movie sequence. Their method presents the semantic content of movie in a static image. A hierarchical event representation method was proposed in [17] to describe the content of videos in an image, which selects a set of events according to the importance at each level. To compare video visualization techniques based on fast-forward, Hoferlin *et al.*[18] evaluated four different visualizations. The controlled user study shows that the object trail visualization supports the object identification, whereas the trajectory visualization is more useful for motion perception. It is surprising that frame-skipping presents reasonable performance for both tasks. Thus, we adopt the posters and image frames to represent movie videos in our paper, which focus on content understanding rather than motion perception.

More recently, there has been a great deal of interest in multi-view learning. It deals with the observations that can be described in several representation spaces, and each representation space may be used to build a predictor. One successful application of co-training algorithm is for word sense disambiguation[19]. The overall goal of multi-view learning is to combine predictors over each view (called view-specific predictors) in order to improve the overall performance beyond that of predictors trained on each view separately, or on trivial combinations of views. Although both multi-view and semi-supervised learning of classifiers have been studied extensively in recent years, their application to the problem of recommendation is novel.

The work most similar to ours is [20]. It extends multi-view learning to rank documents in a multilingual context. The main difference is that our approach uses three different media types as three views of the movie data while that is in the same modality (text in different languages). Besides, our work focuses on rating predicting on movie. The characteristics of recommendation make it more difficult to do multi-view recommendation than to do multi-view classification and ranking. Multi-view learning has an assumption of the agreement of different views. But the rating values of a movie in each view may be not consistent. For example, if the movie is rated as 4, it is the comprehensive score from the visual, audio, and text views. It does not mean that each view gets the score of 4. To solve the problem, we assign a new score to the items whose ratings are not consistent among different views. And the co-training process is executed only when there are a few instances in the training set.

3 Multi-View Representation of Movie

This section presents the multi-view representation for movie recommendation. We first present the notations and the formulation of the recommendation pro-

blem. Then feature extraction, representation, and similarity measure for each view will be introduced in the following subsections.

3.1 Notations and Problem Formulation

In a classical movie recommender system, there are two types of entities, users and movies. Throughout this paper, we use $U = \{1, 2, \ldots, |U|\}$ and $M = \{1, 2, \ldots, |M|\}$ to denote the sets of all user IDs and all movie IDs, respectively.

Given a movie item $m \in M$, we describe it from three views, which are in different media types, i.e., image, text, and audio. It is represented by the triplet of visual, textual, and audio views as $m = (m_v, m_t, m_a)$, where v, t, and a denote the visual, textual, and audio view respectively, and m_v, m_t, and m_a denote the corresponding visual, textural and audio data respectively. Here, we collect images (posters and photos), text (storylines), and audio (sound tracks) data of a movie as the three views. The set of movies is then represented as $M = (V, T, A)$, where V, T and A are the multimedia sets of three views of all the movies in M.

For each individual view $i \in \{v, t, a\}$ corresponding to movie m, a set of features F_i^m are extracted to describe the movie. It is denoted as $F_i^m = \{f_{i1}^m, f_{i2}^m, \ldots, f_{in}^m\}$, where f_{ij} denotes the j-th feature from view i.

Let us denote a given user by $u \in U$, a movie item by $m \in M$, and a rating value by $r \in \{1, 2, 3, 4, 5\} \equiv R$. A set of items rated by user u is represented by S_u, and based on this set, we predict the rating value of a new movie q as: $q \times S_u \rightarrow \arg\max \boldsymbol{P}(q)$. $\boldsymbol{P}(q)$ is a vector whose length is the cardinality of R and i-th element responds to the prediction score of q rated as the i-th value in R.

3.2 Visual View Representation

Visual experience is an important part of a movie, which is a unique element of video. The basic assumption of visual view is that the user may be prone to like movies that have similar visual experience. The visual view is constructed aiming to describe the visual content of a movie.

The direct way to explore visual content of movies is to analyze movie videos. But it is computing expensive and time-consuming. The film posters as the advertisement of a film usually have visual experience similar to that of the movie itself. Fig.3(a) includes the sample posters of some movies from MovieLens dataset①. For comedy and romantic movies, the posters have lighter color. Conversely, the posters of Sci-Fi, horror movies are prone to make the audience feel unpleasantness. So we use posters and photos of movies as the visual view instead of video data, which are easier to collect and represent than videos. Fig.3(b) shows some examples of posters and photos of movie "Clueless(1995)".

For each movie $m \in M$, the posters and photos of it are collected from imdb.com, denoted as $m_v = \{p_1, p_2, \ldots, p_n\}$. We use one poster and five photos, which are provided in the website (imdb.com), for each movie in our experiments. Visual features like color and texture are proved to be useful in visual content analysis[21]. So RGB color histogram and texton histogram are considered here. Besides these two features, another two features of color emotions are used. For each image $p_i \in m_v$, four kinds of features are extracted: color histogram, mean score of emotion factors, color emotion histogram (ehist)[22] and histogram of textons. As a result, the visual view can be represented as $F_v^m = \{f_{v1}^m, f_{v2}^m, f_{v3}^m, f_{v4}^m\}$.

Fig.3. (a) Example of poster images for 10 genres. It shows that as the kind of genre changes, the visual experience of the posters also changes. (b) Example of the poster and five photos of comedy movie "Clueless (1995)".

①http://movielens.umn.edu, Aug. 2013.

RGB Color Histogram. Color features can describe the visual experience directly. Fig.3(a) shows some exemplar poster images for ten movie genres. It shows that posters of the same genre have similar color distribution. The color feature may be helpful in capturing users' preference in visual experience. We compute color histogram of RGB, which has 8 quantization levels per color channel. It constructs a 512-dimension vector for each image. We compute RGB histograms for all images in the set m_v, denoted as feature f_{v1}^m in F_v^m .

Color Emotion Feature. Color histogram only represents the color distribution of images, while the same genre of movies may have diverse color distributions. Color emotion is the emotion feelings evoked by either single color or color combination. It belongs to the cognitive aspect of color and can be a good complement to color histogram. Compared with the traditional color histogram, it defines similarity in a semantic way. The semantic similarities between color and affective words is validated by affective images colorization in [23]. Moreover, it can be quickly extracted and compacted, which characterizes images with three emotion values (activity, weight and heat). For each image, we compute the mean score for each emotion factor as a three-dimension vector, denoted as f_{v2}^m in F_v^m .

The relationship between color emotion and color preference was investigated by Ou *et al.*[24] We use the transformation equations between the color space and the color emotion space following [22]. The mean score of each emotion fails to distinguish the images that have multiple emotional appearances. Thus a color emotion histogram with 64 bins[22] is further extracted as a complement to f_{v2}^m. It is denoted as f_{v3}^m in F_v^m.

Texture Feature. Histogram of textons is a distribution-based method to describe the texture feature. The basic idea is to compute a texton histogram based on a universal texton dictionary. First, 1 000 images are randomly selected from image set V. These images are convolved with a filter bank[25] to generate filter responses. Then the filter responses over these images are aggregated together, and n texton cluster centers ($n = 300$ in our experiments following [25]) are computed using the standard k-means algorithm. Given the n texton cluster centers as the texton vocabulary, each image is represented as a histogram of texton labels. We compute texton histograms over the posters and photos of m, denoted as f_{v4}^m in F_v^m.

3.3 Textual View Representation

Movies with similar visual experience may tell different stories. Besides visual experiences, another factor influencing the rating of users is the storyline of movies. People are used to look toward the movie synopsis before seeing a new movie, from which to judge if a movie is attractive. So it is not enough to predict ratings of a movie for users with just visual views.

Synopsis provides valuable content information of a movie including background (e.g., location, organization, and time) of the story, occupation of the characters. Fig.4 is an exemplar of storylines for movie "Golden Eye" from two websites (imdb.com, rotterntomator.com). The text includes background information ("China", "UK", etc.), activity content ("stop", "capture", "war", "plant", etc.), and so on. It shows that the text words could capture the content of the movie effectively. Thus text information could be used to mine users' preferences on country (foreign or domestic), era (modern or ancient), and so on, which can be used to enhance the accuracy of rating prediction of a recommender system.

James Bond **heads** to **stop** a **media** mogul's **plan** to **induce war** between **China** and the **UK** in order to **obtain exclusive global media** coverage.

Agent 007 and his **partner**, **Agent** 006 (Sean Bean), **pull** a daring raid on a **chemical weapons plant** in the **Soviet Union**; however, they are **captured** by **Russian troops**, and while Bond is able to **escape**, 006 is not so **lucky**. Several years later, the **Soviet Union** and the **Cold War** are a thing of the past, but Bond is still at **work ferreting** out **evildoers** everywhere.

Fig.4. Example of synopsis for movie "Golden Eye (1995)" from imdb.com and rottentomato.com and its features (bold).

We collected the synopses of movies from the two websites as sources of textual view. To utilize the valuable information included in the set of synopses on items, we 1) use a named entity recognizer to identify story background features of items from synopses, 2) mine the most frequent nouns to describe the content of movie story, 3) select adjectives and verbs in storyline as feature items.

For each movie $m \in M$, its synopses are collected from imdb.com and rottentomatoes.com, which are denoted as $m_t = \{t_{\text{imdb}}, t_{\text{rt}}\}$. For each text $t_i \in m_t$, $i \in \{\text{imdb}, \text{rt}\}$, four kinds of features are extracted: location and organization, frequent noun, adjective, and verb.

We assume that each synopsis describes the compact information about the movie video. For each text set, we adopt the following procedure for computing a fea-

ture item set. First, we run the Stanford Log-Linear Part of Speech (POS) tagger[26] on the sentences and select noun, adjective, and verb words to describe each movie. These words usually describe useful information on movie content such as career (e.g., scientist, teacher), sentiment (e.g., happy, evil) and action (e.g., save, kill). To remove named entities from these feature words, we further use Named Entity Recognizer tagger[27] to find 7 classes of entities (location, time, person, organization, money, percent, data). Verbs and nouns are stemmed using the Porter Stemmer[28] to group the same words in different inflected forms (e.g., "walks", "walking", and "walked" are derived from "walk"). Since there are many useless words in the selected noun, adjective, and verb words, we further filter out such words as person names, stop words, percentage, time and money words. Subsequently, we execute the following steps to describe the text view of a movie:

1) merge the sets of location and organization words to construct the feature set f_{t1}^m. Location and organization are related in some degree. For example, UK is labeled as an organization but it also includes location information;

2) select frequent nouns as f_{t2}^m. Since the quantity of noun words in synopsis is larger than other words, only the frequent ones are remained;

3) select all adjectives as f_{t3}^m;

4) select all verbs as f_{t4}^m.

These words give comprehensive description about the movie. Thus, we represent the text view as $F_t^m = \{f_{t1}^m, f_{t2}^m, f_{t3}^m, f_{t4}^m\}$. For each feature, we build a vector space model for it. And f_{ti}^m $(i = 1, 2, 3, 4)$ denotes a vector with each element representing the TF-IDF (term-frequency-inverse document frequency) value of the word.

3.4 Aural View Representation

Music in a movie plays an important role in evoking the emotion of audiences. Studies have shown that music has high agreement among listeners about what type of emotion is being expressed[29-30]. Many classic movies usually have their representative music, such as "My heart will go on" in "Titanic". When people listen to the music, they will be reminded of the movie. Different genres of movies are also different in music. Romantic movies usually have smooth and consistent rhythm while action movies usually have rough and irregular rhythm. Thus we adopt the audio feature of music in movies to describe audio content.

Soundtracks of movies can be collected from websites conveniently. We collect average five audios for each movie from soundtrack.net, a website with a large amount of soundtracks of movies and user reviews. These audio data usually include main titles, end credits, and other soundtracks of movies. For each movie $m \in M$, a set of audio data $m_a = \{a_1, a_2, \ldots, a_n\}$ is corresponding to the audio view of movie m. Then the audio features are extracted from a_i $(i = 1, 2, \ldots, n)$ to describe the movie, which are denoted as F_a^m.

To represent the music of movies comprehensively, we extract the most important low-level features and psychoacoustic features (timbre, rhythm, and tonality feature)[29] from m_a as audio features.

Root Mean Square Energy. Root mean square (RMS) energy is one of the dynamics descriptions of sound. It computes energy by using an RMS operator on the amplitude of audio data.

Mel-Frequency Cepstral Coefficients. Mel-frequency cepstral coefficients (MFCCs) offer a description on the spectral shape of the sound. They are derived from a type of cepstral representation of the audio chip. MFCC is widely used as the feature in speech recognition systems[31], content-based music information retrieval[32] and audio similarity measures. We build a vector of 13 coefficients as the MFCC feature for each audio clip.

Brightness. A high value of brightness indicates the high-frequency register of the music, whereas a low value indicates the low-frequency register. We compute spectral centroid to describe the spectral shape of the audio, which is correlated with the psychoacoustic features sharpness and brightness[33].

Tempo. Tempo is the speed or pace of a musical piece. The studies indicate that a fast tempo usually associates with happiness or excitement, while a slow tempo usually associates with sadness or serenity[34]. The periodicities of the onset detection curve are estimated as tempo. Besides, the beat spectrum which is a measure of acoustic self-similarity is also used. These two features are integrated as the tempo feature of audio.

Mode. We estimate the modality (major vs minor) of the music audio. Major tonality often conveys happiness or joy, while minor tonality is associated with sadness[30]. We compute the key strength difference between the best major key and the best minor key for mode description.

As a result, m_a is represented by $F_A^m = \{f_{a1}^m, f_{a2}^m, f_{a3}^m, f_{a4}^m, f_{a5}^m\}$, which represent the RMS energy, MFCC, brightness, tempo, and mode of the music respectively.

3.5 Single View Similarity Measure

After feature extraction, we define the similarity measure method for each view. Given two movies $i, j \in M$, and their three views' representations $\{F_v^i, F_t^i, F_a^i\}$

and $\{F_v^j, F_t^j, F_a^j\}$. These features are computed following the process of previous subsections. We regard each kind of features in a view as a channel c, then the similarity of instance i and j of each view $c \in \{v, t, a\}$ is defined as:

$$d(f_c^i, f_c^j) = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \frac{1}{\Omega_c} D_c(f_c^i(m), f_c^j(n)), \quad (1)$$

where D_C is the distance measure for channel c, $f_c^i(m)$ and $f_c^j(n)$ correspond to the features of an element (image, text, or audio) for movie i and j respectively. N_i, N_j are the number of elements in the c view of i and j, respectively. Ω_c is the average channel distance.

In visual view, D_C are L2 norm and χ^2 distance. The L2 norm is used for calculating the distance for color histogram, mean score of emotion factors, and color emotion histogram. Specially, the distance function used for texton histogram is χ^2 distance[35]. For two histograms $h_1 = (u_1, u_2, \ldots, u_L)$ and $h_2 = (w_1, w_2, \ldots, w_L)$, the χ^2 distance is defined as:

$$D(h_1, h_2) = \frac{1}{2} \sum_{i=1}^{L} \frac{(u_i - w_i)^2}{u_i + w_i}. \quad (2)$$

In text view, cosine similarity is used for feature comparison, which is generally used in the vector space model of text retrievals.

In audio view, the similarity distance of two features is computed depending on feature types. In our experiment, we follow the setting of MIRtoolbox[36]. If the features are not decomposed into frames, the cosine similarity is used as default. If features contain peaks, the vectors representing the peak distributions are compared using Euclidean distance. The detailed description of the audio features and distance measure can be found in the user's guide for MIRtoolbox②.

4 Multi-View Movie Recommendation

The multi-view semi-supervised recommendation consists of four components: single view score assignment, single view recommendation, multi-view profile enrichment, and multi-view recommendation.

Firstly, the prediction of single view is processed from each view. Then we update the prediction score using the relations between views. Finally, the scores from the three views are integrated to predict the rate for new movies.

Based on the notations defined previously, the problem of movie recommendation can be formulated as follows.

4.1 Single View Score Assignment

This process assigns a new score to a movie item that has large difference with its k nearest neighbors in the single view feature space. Given a rated movie m with rating r, k nearest neighbors are found from the rated movies (except itself). Then the final new rating nr is computed using (3), where R_k is the set of rating values from these neighbors, kr is the most frequent rating in R_k.

$$nr = \begin{cases} \left\lfloor \dfrac{kr + r}{2} \right\rfloor, & \text{if } kr < r,\ r \notin R_k, \\ r, & \text{if } r \in R_k, \\ \left\lceil \dfrac{kr + r}{2} \right\rceil, & \text{if } kr > r,\ r \notin R_k. \end{cases} \quad (3)$$

4.2 Single View Recommendation

Each view of the movie contributes to the final prediction of movie rating. Single view recommendation predicts a vector of score for each rating through the relevance in single view space.

To determine the degree of a user $u \in U$ rating a movie $m \in M$ with rating value of $r \in \{1, 2, 3, 4, 5\} \equiv R$, the system computes the prediction score of m from view k, denoted as $\boldsymbol{P}_k\ (m)$. It is a vector with the length of $|R|$, where the i-th element corresponds to the prediction score of movie m rated as the i-th value of R. The i-th element $P_k(m, u, r)$ is define as

$$P_k(m, u, r) = \frac{\sum_{m_r} d_k(m, m_r)}{|M_r|} + \frac{n_r}{N} \delta(r), \quad (4)$$

where $k \in \{v, t, a\}$, m_r is a movie item that is rated as r by user u, d_k means the distance between m and m_r according to (2), M_r is the set of movies rated as r by user u and $|M_r|$ is the cardinality of it. The first part computes the average similarity of m with all movies rated as r. The value of the second part increases if there are more movies rated as r in the N nearest neighbors of movie m. $\delta(r)$ is an indicator function judging whether there exists $m_r \in N(m)$, which is computed as:

$$\delta(r) = \begin{cases} 1, & \text{if } \exists m_r \in N(m), \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

It shows that the larger $P_k(m, u, r)$ is, the higher probability of movie m is rated as r by user u. After computing the $P_k(m, u, r)$ score for each rating value, the rating that has the highest prediction score is regarded as the prediction rating of single view recommendation. The process of the single view recommendation is denoted as $Recom\_SV(Z, Q, k)$, where Z is the

②MIRtoolbox user's guide. https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox, May 2013.

rating history of users, Q is the movies to be rated, and k is the set of views.

4.3 Semi-Supervised Enrichment

Most times in realistic settings, users provide only a few rating histories of movies. The lack of rating data leads to the cold start problem in recommendation. We enrich the user profiles with a semi-supervised way when the number of rating histories is lower than a threshold. An iterative co-training technique of multi-view learning is used here. First, the prediction scores of unrated movies from each view are computed respectively. Then the items rated with the same score by all the view predictors are added to the training set. This process repeats until there is no new item to add. The algorithm of this process is shown in Algorithm 1 and denoted as $MVE(Z, Q, k)$. The enriched training set is used to predict the remaining items in the testing set in multi-view recommendation. Note that the enrichment will be executed only when the rated history of users is less than a threshold σ.

Algorithm 1. Semi-Supervised Enrichment Procedure
Input: rated movies $Z : (u, m, r)$, movies to be rated Q: (u, m), views: $\{v, t, a\}$
Output: Z'

1 **for** each $q \in Q$, view $k \in \{v, t, a\}$
  $\boldsymbol{P}_k(q) = Recom\_SV(Z, Q, k)$
endfor
2 **for** each $q \in Q$
  if $\forall \boldsymbol{P}_k(q) = r$
    $Z' \longleftarrow Z \cup \{(q, r)\}$
  endif
endfor

4.4 Multi-View Recommendation

Algorithm 2 presents the multi-view recommendation procedure. First, the rating value in each view is assigned using the method presented in Subsection 4.1, denoted as $SVSA(Z, k)$. If the user has few rating histories, the semi-supervised process $MVE(Z)$ executes in step 2 to enrich the user profile. Then $Recom\_SV(Z', Q, k)$ is computed from each view as in step 3 with the enriched data Z'. Here $Recom\_SV(Z', Q, k)$ denotes the single view recommendation algorithm as described before, with training set $Z'(u, m, r)$ as its input. In step 4, the prediction score of each view is updated according to the scores of other views. The rating values are partitioned into three score sets: R_-, R_m, R_+, where $R_- = \{1, 2\}$, $R_m = \{3\}$, and $R_+ = \{4, 5\}$. If all the other views $w \in \{v, t, a\}/k$ have the same score set R_i ($_i \in \{-, m, +\}$), then the current view updates the prediction score in R_i to keep consistent with the other views. The added item gets large value if the difference between the current view and the other views is small. On the other hand, if the other views are not agreed with the same score set R_i, the current score will not change. This update makes sure that the prediction score converges to the rating that has high confidence in all views. Finally, the three single view recommendations are integrated to get the final result.

Algorithm 2. Multi-View Movie Recommendation
Input: rated movies $Z : (u, m, r)$, movies to be rated $Q : (u, m)$, views $\{v, t, a\}$
Output: $\boldsymbol{P}(m)$

1 **for** each view $k \in \{v, t, a\}$
  $R'_k = SVSA(Z, k)$;
endif
2 **if** $|Z| < \sigma$
  $Z' = MVE(Z(u, m), R')$;
else
  $Z' = Z$;
endif
3 **for** each view $k \in \{v, t, a\}$
  $\boldsymbol{P}_k(Q) = Recom\_SV(Z', Q, k)$
endfor
4 **for** each $q \in Q$, view $k \in \{v, t, a\}$
  if $\forall \arg\max(\boldsymbol{P}_w(q)) \in R_+$, $w \in \{v, t, a\}/k$
    $P_k(q, u, R_+) = P_k(q, u, R_+) + e^{-\frac{1}{|r_k - r_w|}}$
  elseif $\forall \arg\max(\boldsymbol{P}_w(q)) \in R_-$, $w \in \{v, t, a\}/k$
    $P_k(q, u, R_-) = P_k(q, u, R_-) + e^{-\frac{1}{|r_k - r_w|}}$
  elseif $\forall \arg\max(\boldsymbol{P}_w(q)) \in R_m$, $w \in \{v, t, a\}/k$
    $P_k(q, u, R_m) = P_k(q, u, R_m) + e^{-\frac{1}{|r_k - r_w|}}$
  endif
  $\boldsymbol{P}(q) = \sum_k \boldsymbol{P}_k(q)$
endfor

5 Experiments

This section provides the experimental results of the recommender system proposed in this paper. It consists of the comparison results of different methods. In particular, we evaluate the recommender results using the root mean squared error (RMSE) metric[37], providing the performance for each user considered in the dataset.

5.1 Dataset

The MovieLens dataset③ is chosen for evaluating the rating prediction accuracy. It was created during a 7-

③http://www.grouplens.org/node/12/, Aug. 2013.

month period from September 19, 1997 to April 22, 1998 by the developers of MovieLens④, a web-based recommender system on movies.

This dataset consists of 100 000 ratings (1∼5) from 943 users on 1 682 movies. Each user has rated at least 20 movies. The raw data is partitioned into training and testing sets. The training set is used for generating recommendation rating using our method and other recommendation algorithms. The testing set is used for measuring the effectiveness of these algorithms.

The image data of MovieLens dataset are collected from imdb.com. For each movie, one poster and five photos are downloaded. Because of the limitation of web sources, a small portion of movies have photo images less than 5. The synopsis data are retrieved from imdb.com and rottentomatoes.com for each movie. If there is no text information in the website, we use movie name as the text content. Sound tracks of movies are downloaded from soundtrack.net. The audio data are rarer compared with the other media types. We collect all the available audio in soundtrack.net for movies in MovieLens. Table 1 is the statistical information of these data of three views, including the sum of items for each view and average number of items per movie. There are 9 298 images, 3 364 texts, and 3 047 audio data in all. It shows that there are 5.5 image, 2 text, and 1.8 audio data items per movie in the collected data.

Table 1. Statistical Information of Collected Data from Three Views

Media Type	Number of Items	Avg. Number of Items per Movie
Image	9 298	5.5
Text	3 364	2.0
Audio	3 047	1.8

Table 2 is the statistical information of feature words in two data sources imdb.com (imdb) and rotterntomato.com (rt). Since the storyline in imdb.com is generally shorter than that in rotterntomato.com, the number of feature words in the former source is smaller.

Table 2. Statistical Information of Extracted Feature Words in Textual View

Feature Type	Number of Word		Avg. Number of Words per Movie	
	imdb	rt	imdb	rt
Location & organization	1 619	8 147	0.96	4.8
Noun	14 258	68 970	8.50	41.0
Adjective	4 411	22 885	2.60	13.6
Verb	5 985	29 656	3.60	17.6

In average, there are one word about location and organization, eight nouns, two adjectives and three verbs to describe each movie. The average words number per movie shows that the imdb dataset is very sparse. So we build vector space model on the union feature set of two data sources.

5.2 Results

The first experiment is to validate whether the multi-view descriptors benefit the elicitation of user interests. Table 3 is an example of user preferences from the textual view feature space. The feature of movies rated

Table 3. Example of User Preference in Textual View

Feature Type	Word	TF-IDF Value
Location & organization	**Manhattan**	**4.840 3**
	Dante	3.164 6
	Academy	3.100 6
	Shanghai	3.020 7
	Mexico	2.839 2
	Christmas	2.517 2
	Gwynet	2.514 9
	Paltrow	2.514 9
	Italy	2.432 3
	Alabama	2.415 4
Noun	**whale**	**1.257 5**
	boy	0.903 4
	film	0.721 8
	aquarium	0.704 5
	killer	0.692 7
	batman	0.661 2
	movie	0.631 0
	risk	0.628 7
	owner	0.614 3
	war	0.581 2
Adjective	**beloved**	**4.175 3**
	sexual	**1.697 1**
	casual	1.556 7
	local	1.460 6
	everyday	1.454 6
	lunar	1.443 1
	real	1.435 8
	good	1.425 8
	young	1.358 3
	sticky	1.346 9
Verb	**kill**	**1.694 6**
	free	**1.573 2**
	put	1.514 9
	begin	1.225 3
	accompany	1.208 6
	plan	1.203 7
	find	1.173 5
	forget	1.120 8
	determine	1.0537
	arrive	0.986 2

④http://movielens.umn.edu, Aug. 2013.

with 5 by a user is illustrated. The text words correspond to the top-10 feature words according to their sum of TF-IDF values. The same user profile represented with 19 kinds of movie genre is shown in Table 4. It shows that the feature words in texture view are consistent with the preference in genres. The drama movies are usually related with "beloved", "sexual". And the words "kill", "free", and "killer" highly occur in thriller, action and crime movies.

Table 4. User Preference in Movie Genre

Genre	Frequency
Drama	25
Comedy	15
Thriller	14
Action	11
Crime	7

Besides, the location & organization feature words indicate users' interest in America (Manhattan, Alabama) and Italy (Dante, Italy). But there are also some unexpected words like "Gwynet Paltrow", which is the name of an American actress. We plan to add cast information in our further research.

The visualization of user preference shows that the multi-view content analysis reveals the user profile in the movies semantically.

Since the number of images and audio data is not very large, the feature extraction is efficient for realistic application. Table 5 lists the time cost for the feature extraction in each view. It shows that the time cost of visual feature extraction without texton quantization is similar to that of word processing. Since the k-means clustering for texton quantization is run once before computing histogram of texton, the time cost for feature extraction from a new image is low. Audio feature is a litter more time-consuming than both textual and visual features, but the cost is endurable.

Table 5. Cost for Feature Extraction in Each View

View	Time (s)
Visual view (per image)	3.072
Textual view (per text)	2.889
Audio view (per audio)	10.355

We then compare the performance of using the combination of features in each view respectively on 639 movies from MovieLens, which have complete description of the three views without lack of audio or photos. Fig.5 represents the RMSE scores of single view recommendation. We use the content of visual view, textual view, and aural view to recommend movies with the same method *Recom_SV* (single view recommendation in Section 4).

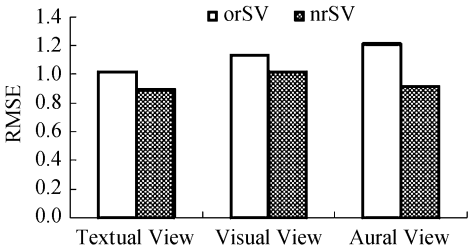


Fig.5. Comparison of RMSE for different views for recommendation.

and aural views are more effective than visual view. The nrSV and orSV represent two kinds of ratings values in training set. The orSV uses original movie rating values in the dataset as the ratings of each view; while nrSV uses new assigned rating of each view using the method described in Section 4 (Single View Score Assign). The RMSE decreases average 0.17 in each view after assigning new ratings, which shows the effectiveness of the score assignment strategy of our method. The result proves that the ratings of movies are not equivalent to the rating of each view. It is necessary to assign the ratings before recommending movies in a single view space. The multi-view recommendation achieves an RMSE value of 0.8017, which is lower than all single views.

To compare the performance before enrichment and after enrichment, we compute the RMSE for different amounts of rating history in the dataset. In Fig.6, MVR is the multi-view recommendation without enrichment. MSR is the semi-supervised recommendation (with $\sigma = 500$) based on multi-view content analysis, which iteratively adds items in unrated set into the training set. We can observe that as the number of rated movies is reduced, the performance of MSR keeps smooth, while the RMSE of MVR increases largely.

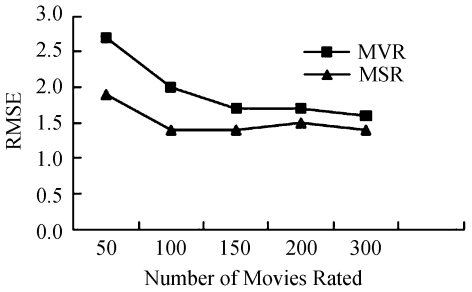


Fig.6. Comparison of performance of multi-view recommendation with enrichment strategy (MSR) and without enrichment (MVR).

Since there is no other method both using multimedia information and handling the limited user profile problem, we compare our method with two methods from two aspects. LF[3] enhances recommendation

using latent factors from movie genres. This method could reduce the effect of limited rating data in collaborating filtering. The method achieved an RMSE of 0.961 7 in MovieLens dataset with enriched profiles. The MudRecS[4] predicts the ratings of multimedia items, which considers reviews, popularity of actors, and genre of the movies. The system considers multiple information to recommend movie, music, book, and painting. But the method does not analyze the visual and audio content.

The RMSE scores indicate that our method is comparable with other methods. Though the RMSE of ours is a little higher than MudRecS (0.72), the later does not consider the recommendation for casual users who lack user history.

6 Conclusions

In this paper, we proposed a novel approach based on multi-view content analysis and semi-supervised enrichment for movie recommendation, which leverages multi-views of movie data to enrich and represent user profile. We adopt multimedia information including image, audio, and text data to provide a comprehensive content description of movie data. To solve the disagreement between different views in using the multi-view learning framework, we assign new scores in each view. And a semi-supervised strategy is adopted to enrich the user profile to handle the cold-start problem. Compared with the metadata content-based method, our method: 1) provides a semantic and comprehensive representation of user interests, 2) avoids over-specialization problem for content-based recommender systems, 3) uses multi-view content to enrich the user profile based on co-training technique, which improves the performance for casual users. Experimental results demonstrate that the video content analysis is helpful for the elicitation of user interest and the recommendation. The features extraction in each view space is efficient and feasible in realistic applications.

For future work, we will focus on the visualization of user profile and improvement of the recommendation quality. In particular, as the proposed method is based on low-level features, we will also make an effort to introduce more semantic features into content analysis.

References

[1] Park J, Lee S J, Lee S J *et al.* Online video recommendation through tag-cloud aggregation. *IEEE MultiMedia*, 2011, 18(1): 78-87.

[2] Hu J, Wang B, Liu Y, Li D Y. Personalized tag recommendation using social influence. *Journal of Computer Science and Technology*, 2012, 27(3): 527-540.

[3] Manzato M G. Discovering latent factors from movies genres for enhanced recommendation. In *Proc. the 6th ACM Conference on Recommender Systems*, Sept. 2012, pp.249-252.

[4] Qumsiyeh R, Ng Y K. Predicting the ratings of multimedia items for making personalized recommendations. In *Proc. the 35th ACM SIGIR Int. Conf. Research and Development in Information Retrieval*, Aug. 2012, pp.475-484.

[5] Davidson J, Liebald B, Liu J *et al.* The YouTube video recommendation system. In *Proc. the 4th ACM Conference on Recommender Systems*, Sept. 2010, pp.293-296.

[6] Chen B, Wang J, Huang Q, Mei T. Personalized video recommendation through tripartite graph propagation. In *Proc. the 20th ACM Int. Conf. Multimedia*, Oct. 2012, pp.1133-1136.

[7] Zhao X, Li G, Wang M *et al.* Integrating rich Information for video recommendation with multi-task rank aggregation. In *Proc. the 19th ACM Int. Conf. Multimedia*, Nov. 2011, pp.1521-1534.

[8] Esparza S G, O'Mahony M P, Smyth B. On the real-time web as a source of recommendation knowledge. In *Proc. the 4th ACM Conf. Recommender Systems*, Sept. 2010, pp.305-308.

[9] Mei T, Yang B, Hua X S, Li S. Contextual video recommendation by multimodal relevance and user feedback. *ACM Transactions on Information Systems*, 2011, 29(2): Article No. 10.

[10] Toutanova K, Klein D, Manning C D, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. the 2003 Conf. the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, May 2003, pp.173-180.

[11] Carrer-Neto W, Hernandez-Alcaraz M L, Valencia-Garcia R, Garcia-Sanchez F. Social knowledge-based recommender system. Application to the movies domain. *Journal of Expert Systems with Applications*, 2012, 39(12): 10990-11000.

[12] Said A. Identifying and utilizing contextual data in hybrid recommender systems. In *Proc. the 4th ACM Conference on Recommender Systems*, Sept. 2010, pp.365-368.

[13] Shi Y, Larson M, Hanjalic A. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Proc. the Workshop on Context-Aware Movie Recommendation*, Sept. 2010, pp.34-40.

[14] Ganter Z, Rendle S, Schimidt-Thieme L. Factorization models for context-/time-aware movie recommendations. In *Proc. the Workshop on Context-Aware Movie Recommendation*, Sept. 2010, pp.14-19.

[15] Biancalana C, Gasparetti F, Micarelli A, Miola A, Sansonetti G. Context-aware movie recommendation based on signal processing and machine learning. In *Proc. the 2nd Challenge on Context-Aware Movie Recommendation*, Oct. 2011, pp.5-10.

[16] Chen T, Lu A, Hu S M. Visual storylines: Semantic visualization of movie sequence. *Computer & Graphics*, 2012, 36(4): 241-249.

[17] Parry M L, Legg P A, Chung D H S *et al.* Hierarchical event selection for video storyboards with a case study on snooker video visualization. *IEEE Trans. Visualization and Computer Graphics*, 2011, 17(12): 1747-1756.

[18] Hoeferlin M, Kurzhals K, Hoeferlin B *et al.* Evaluation of fast-forward video visualization. *IEEE Trans. Visualization and Computer Graphics*, 2012, 18(12): 2095-2103.

[19] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. the 33rd Meeting on Association for Computational Linguistics*, June 1995, pp.189-196.

[20] Usunier N, Amini M R, Goutte C. Multiview semi-supervised learning for ranking multilingual documents. In *Proc. the 2011 European Conf. Machine Learning and Knowledge Discovery in Databases*, Sept. 2011, Vol.3, pp.443-458.

[21] Smeulders A W M, Worring M, Santini S *et al.* Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000, 22(12): 1349-1380.

[22] Solli M, Lenz R. Color emotions for image classification and retrieval. In *Proc. the 4th European Conference on Colour*

in Graphics, Imaging, and Vision, June 2008, pp.367-371.

[23] Wang X H, Jia J, Liao H Y, Cai L H. Affective image colorization. *Journal of Computer Science and Technology*, 2012, 27(6): 1119-1128.

[24] Ou L, Luo M, Woodcock A *et al.* A study of colour emotion and colour preference — Part 1: Colour emotion for single colours. *Color Research & Application*, 2005, 29(3): 232-240.

[25] Varam M, Zisserman A. A statistical approach to texture classification from single images. *Journal of Computer Vision*, 2005, 62(1/2): 61-81.

[26] Toutanova K, Klein D, Manning C D, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. the 2003 Conf. the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, May 2003, pp.173-180.

[27] Finkel J R, Grenager T, Manning C. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. the 43rd Annual Meeting on Association for Computational Linguistics*, June 2005, pp.363-370.

[28] Porter M F. An algorithm for suffix stripping. In *Readings in Information Retrieval*, Jones K S, Willett P (eds.), Morgan Kaufmann Publishers Inc., 1997, pp.313-317.

[29] Scherer K R, Zentner M R. Emotional effects of music: Production rules. In *Music and Emotion: Theory and Research*, Juslin P, Sloboda J (eds.), New York: Oxford University Press, 2001, pp.361-387.

[30] Hunter P G, Schellenburg E G, Schimmack U. Feelings and perceptions of happiness and sadness induced by music: Similarities, differences, and mixed emotions. *Psychology of Aesthetics, Creativity and the Arts*, 2010, 4(1): 47-56.

[31] Logan B. Mel frequency cepstral coefficients for music modeling. In *Proc. Int. Symp. Music Information Retrieval*, Oct. 2000.

[32] Casey M A, Veltkamp R, Goto M, Leman M, Rhodes C, Slaney M. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 2008, 96(4):668-696.

[33] McKinney M F, Breeebat J. Features for audio and music classification. In *Proc. the International Symposium on Music Information Retrieval*, Oct. 2003.

[34] Gabrielsson A, Lindstrom E. The influence of musical structure on emotional expression. In *Music and Emotion: Theory and Research*, Juslin P N, Sloboda J A (eds.), New York: Oxford University Press, 2001, pp.23-248.

[35] Zhang J, Marszalek M, Lazebnik S *et al.* Local features and kernels for classification of texture and object categories: A comprehensive study. *J. Computer Vision*, 2007, 73(2): 213-238.

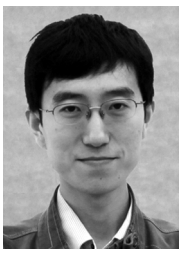
[36] Lartillot O, Toiviainen P. A Matlab toolbox for musical feature extraction from audio. In *Proc. the 10th International Conference on Digital Audio Effects*, Sept. 2007.

[37] Ricci F, Rokach L, Shapira B, Knator P B. Recommender Systems Handbook. Springer, 2011.

Wen Qu is a Ph.D. candidate in the School of Information Science and Engineering at Northeastern University, Shenyang. She received her B.S. degree in software engineering and M.E. degree in computer software and theory from the Northeastern University. Her research interests include video content analysis and multimedia data mining.

Kai-Song Song is a Ph.D. candidate in Northeastern University, Shenyang. He got his B.S. degree from Shandong University of Science and Technology, Qingdao, in 2010, and his M.S. degree in computer software and theory from Northeastern University, in 2012. His recent research interests include recommendation systems, network public opinion analysis, multi-modal learning, as well as social network analysis.

Yi-Fei Zhang is an assistant professor in the School of Information Science and Engineering at Northeastern University. She received her Ph.D. degree in computer software and theory from Northeastern University, China. Her research interests include image processing and machine learning.



Shi Feng is an assistant professor in the School of Information Science and Engineering at Northeastern University. He received his Ph.D. degree in computer software and theory from Northeastern University, China. His research interests include opinion mining, sentiment analysis and emotion detection.

Da-Ling Wang is a professor at School of Information Science and Engineering, Northeastern University, Shenyang. She received her Ph.D. degree in computer software and theory from Northeastern University in 2003. Her main research interests include data mining, machine learning and information retrieval.

Ge Yu is a professor at School of Information Science and Engineering, Northeastern University. He received his Ph.D. degree in computer science from Kyushu University of Japan in 1996. He is a member of IEEE, ACM, and a senior member of CCF. His research interests include database theory and technology, distributed and parallel systems, embedded software, and network information security.