Multimedia Systems (2016) 22:535–542
DOI 10.1007/s00530-015-0451-z

SPECIAL ISSUE PAPER

A location-aware TV show recommendation with localized sementaic analysis

Fanglin Wang · Daguang Li · Mingliang Xu

Published online: 30 January 2015

Abstract Nowadays, microblogging platforms such as Twitter and Weibo can also be seen as a good media to present reviews about topics. More and more people tend to share their thoughts through various microblogging sites. For example, when a TV show is being shown, the users would like to share and discuss their opinions about it on these platforms. However, one phenomenon is the popularity of the TV usually varies for different regions due to the cultural differences, custom and some other factors. Predicting whether a TV show will be popular at certain locations is then desirable. In this paper, a location-aware TV show recommendation scheme is proposed. By incorporating the social network information of users from different locations, a location-based user profile is obtained. Then, the scheme conducts prediction of TV show popularity for different regions based on the profile and similar shows. For a new TV show, the popularity of the similar shows is utilized to get the initial location-show matrix. Then, the location profiles and physical distance are used as regularizer into the collaborative filtering framework to further refine the prediction. A TV show dataset with location-aware social network information has been collected. Experiments have been conducted on real data and encouraging results have been achieved.

F. Wang (✉)
School of Computing, National University of Singapore, Singapore, Singapore
e-mail: hardegg@gmail.com

D. Li
Publicity Department, Northeast Forestry University, Harbin, China

M. Xu
School of Information Engineering, Zhengzhou University, Zhengzhou, China

1 Introduction

There are many factors impacting the audience rating of a TV show, e.g., broadcasting time, the influence of the TV channel, the marketing strategy, etc. But one phenomenon is the popularity of a TV show usually differs at different regions. In China, regional culture is an important factor determining audience rating, especially the regional appreciation preference division has become more and more significant in these years [16, 21]. One example is the TV show named “Country Love” achieved 21.41 % of high audience rating in Shenyang, but in contrast less than 5 % in Guangzhou [21] in 2010. Some TV shows can resonate with audiences from some regions but fail in other regions. Hence it is important for a TV show to predict the audience rating prior to the broadcasting and make the right strategy for different regions. However, it is not straightforward to know the preference of a region’s audiences which is hard to describe or quantize, and it is not easy to establish the relationship between a TV program with a region. In this paper, we explore the problem of automatic popularity prediction for TV shows at different local regions from the angle of multimedia analysis. Basically it is a recommendation problem and the purpose is to offer new TV shows to a region based on multiple sources of data.

Recent years have witnessed the high-speed development of social networks. People would like to spend more time on social network and feel more free to express their views there. Many applications such as social image search [6, 14] have been proposed. For TV shows, the social network platform like Sina Weibo[1] is a good platform to disseminate the TV shows and gather the comments from

[1] http://weibo.com.

audience. A typical scenario is if an audience watched a TV show and got good or bad experience, he or she would post the feeling on social network and sometimes '@' (the action of mentioning) his friend. Moreover, location of the users is also an important factor which has been shown in [20]. To analyze the microblogs from a region on platform like Sina Weibo, we can look for these microblogs according to the registration place of the users posting these microblogs. Hence it is more effective and convenient to analyze TV show popularity on social network.

There have been huge effort devoted to recommendation system in these years such as [9, 18, 24]. In [18], location semantic analysis is proposed to model every location for news article recommendation. In this method, the location has its own geographical topics just like the articles have topic vectors. Kim and Pyo et al. [9] proposed a TV program scheme where the users' interests on watched TV program contents is implicitly inferred and the recommendation is conducted by collaborative filtering. Kaminskas et al. [8] proposed a location-aware music recommendation method, which is mainly based on representing both POIs and music with tags. These location-aware recommendation methods generally recommend items based on the matching between location profile and semantic content of items. They are not suitable for our location-aware TV show recommendation because the new TV show generally do not have any semantic tags and the video auto-tagging is not reliable. To the best of our knowledge, the application of location-aware video recommendation has not been studied and no suitable methods can be directly applied.

How to represent a geographical region has been researched a lot recently. Yin et al. [25] proposed a latent geographical topic analysis method to combine location and text information. This method can be used to find regions of interests and compare topics across different locations. Son et al. [19] proposed a probabilistic explicit semantic analysis model in order for a probabilistic of topics and construction of relevance of locations. However, these methods work mainly on general topics instead of TV show related topics. For these works, the corpus is mainly from Wikipedia.

In this paper, we propose an automatic location-aware TV show recommendation method. The method predicts the popularity of the TV show at a specific region before it is shown. Multiple data sources are employed including the social media and geographic information of a region, and audio feature and visual features of the TV show. There are mainly three contributions of this work. First, to our best knowledge, it is the first location-based TV show recommendation work by incorporating microblog information, geographic information, visual information and audio information. Second, we propose a location profile representation based on geographic information and microblog text information. Third, we develop a TV show similarity metric-based learning by using multiple video and audio features. A real world dataset has been collected and experiments have been conducted to demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces the whole work flow of this work. In Sects. 3 and 4, we detail the location profile model and the TV show representation model, respectively. Section 5 shows the experimental results. Finally, the paper is concluded in Sect. 6.

2 Whole framework

We denote the popularity of different TV shows across different locations as a location-show matrix M. Here, M_{ls} is the popularity of show s at location l. For a new TV show v, we first initialize the popularity using the popularity of other similar shows at the corresponding location:

$$M_{lv} = \sum_{u=1}^{n} M_{lu} \times \mathrm{SIM}_{uv}^{V}, \tag{1}$$

where $u = 1, \ldots, n$ is the TV show with known popularity and SIM_{uv}^{V} denotes the similarity between TV show u and TV show v. A relevance graph is constructed based on the relevance among different shows. A random walk-based updating procedure is employed as:

$$M_{lv}^{t+1} = \beta M_{lv}^{0} + (1-\beta) \sum_{u \neq v} \mathrm{SIM}_{uv}^{V} M_{lu}^{t} \tag{2}$$

The initialized location-show popularity matrix contains many noises. The latent representation of location and shows are desirable to obtain the accurate popularity prediction, similar to traditional user-item matrix. Given the location-show matrix M, we aim to find the latent representation of location L and show V such that $M \approx LV^{T}$. The non-negative matrix factorization (NMF) [10] is a popular method to achieve this goal. Moreover, in order to introduce the location similarity, a graph regularized non-negative matrix factorization (GNMF) [2] method is employed. In our proposed method, we aim to minimize

$$||M - LV^{T}||^{2} + \lambda \sum_{ij} ||L_{i} - L_{j}||^{2} \mathrm{SIM}_{ij}^{L}, \tag{3}$$

where SIM_{ij}^{L} denotes the similarity between two locations and similar locations tend to have similar latent representation.

The overall process is illustrated by Fig. 1. We extract geographical and topic features for location representation, and fuse the two features for the computation of SIM_{ij}^{L} used in (3). The TV show feature consisting of visual part and

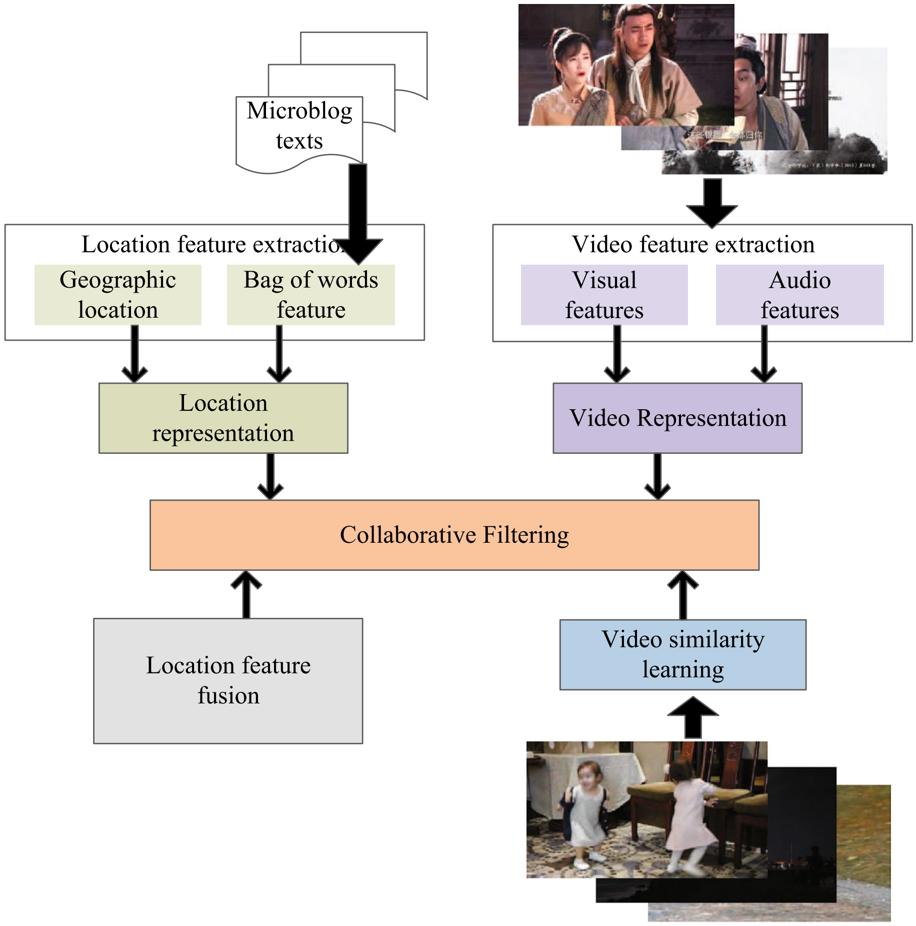

Fig. 1 The overall process of the localized TV show recommendation

audio part are computed, and the video similarity metric SIM_{uv}^{V} used in (2) is learnt. The feature representations and similarity metrics are put into the collaborative framework and achieve the TV show popularity prediction.

3 Location profile

In this paper, we think a location is characterized by two factors, i.e., geographical coordinate and textual comments on TV shows obtained from Sina Weibo.

3.1 Geographical location

Each location is a province in China in this paper. In most cases, two adjacent provinces will have similar culture and custom which will possibly make the two provinces to have similar preference towards TV programs. There are a few features to calculate location similarity including shape, orientation and position [22]. In our case, we only consider the geographical distance as the location similarity metric.

We set the location of each province as the coordinate **L** depicted by longitude and latitude of the capital. The longitude and latitude of a province capital are looked up from Baidu Map.[2] To calculate the spatial distance d_{ij} between two capitals i and j, we use Haversine formula by inputting the coordinates of two locations. We calculate the similarity of two regions according to

$$\mathrm{SIM}^{G}(i,j) = \exp\left(-\frac{d_{ij}^{2}}{\sigma_G}\right), \tag{4}$$

where we choose σ_G empirically.

3.2 Social network information

For some provinces far apart, they may also tend to have similar TV show preference which can be reflected by what the audiences posted on social networks. To explore this phenomenon, we employ the information from social networks. After watching a TV show, nowadays lots of people will use social network platform to post TV show related information including comments or recommendation to

[2] http://map.baidu.com.

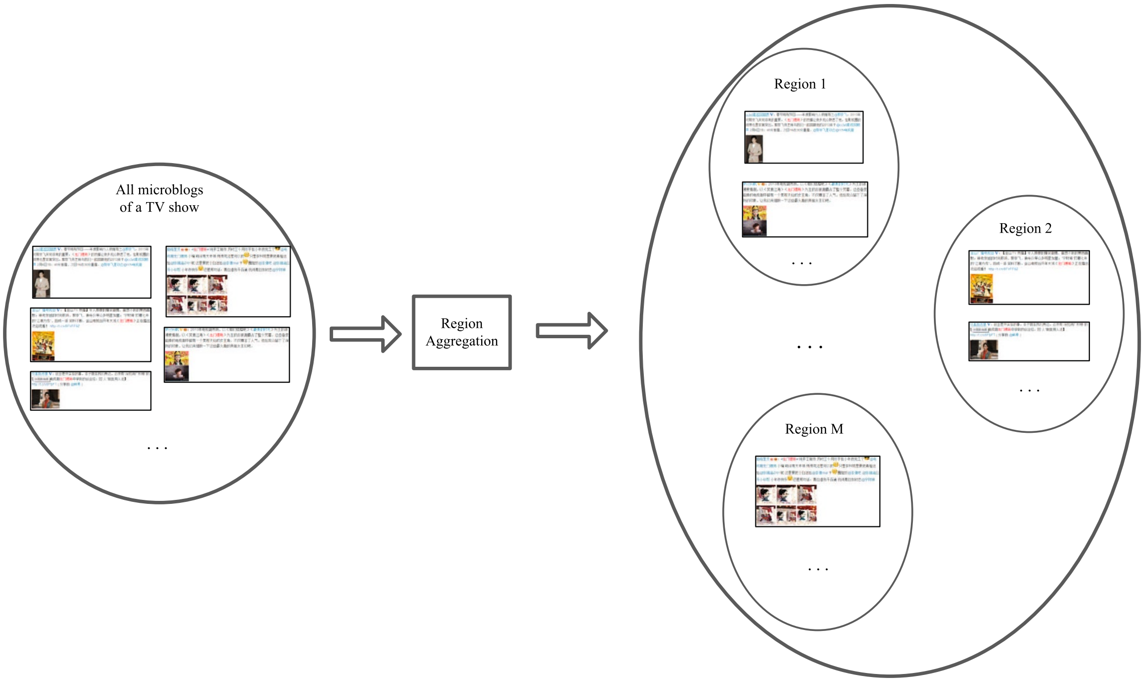

Fig. 2 Illustration of microblogs are assigned to each region

friends, etc. Hence the texts about a TV show in a region can reflect the characteristic of this region. To extract this characteristic

To acquire the location information of microblogs, the frequently used identification is GPS information. However, there are two drawbacks of making use of GPS information. On some social network platforms like Sina Weibo, only small fraction of microblogs contain GPS information which leads to massive TV show related microblogs ineffective and wasted. The other drawback is GPS does not mean the users are really living in this region, instead they might be tourists or pass by. In this work, different from [18, 19, 25], we determine the location of a microblog as the registration place of its poster which is almost always available. The registration place for user means the place he has or will stay for rather long term, hence the view from this user can stands for the place he is living in.

Suppose we have M regions and N TV shows all together. For TV show i, we collect all the microblog texts T_i related to this TV show. Then as depicted by Fig. 2 we divide the T_i into local regions and get $\left\{T_i^j\right\}$, the texts related TV show i in region j. Then for all the TV shows of one region j, we get the text set $T_j = \left\{T_i^j\right\}_{i=1}^M$. In the next step, we extract the topics at this location. For each T^j, we extract the topics in this region, where the topic vector $l_j = \langle \phi_1(T_j), \phi_2(T_j), \ldots, \phi_{N_{TO}}(T_j)\rangle$, where N_{TO} is the number of topics. Here we employ Latent Dirichlet Allocation model [1] to extract the topics and the corpus is set as all the microblog texts related to all the TV shows.

With this location representation, we calculate the similarity between location as a cosine similarity

$$\mathrm{SIM}(l, l') = \frac{l \cdot l'}{\|l\| \cdot \|l'\|}. \tag{5}$$

3.3 Similarity by feature fusion

With the geographical and textual information from Sina Weibo, to calculate the similarity between two locations, we fuse them linearly according to

$$\mathrm{SIM}^L = \alpha \mathrm{SIM}^G + (1 - \alpha)\mathrm{SIM}^T, \tag{6}$$

where α is a coefficient to determine the weight of geographical location feature which we set as 0.1 throughout this paper.

4 TV program features

If two videos are similar, we can expect it will achieve similar popularity for the same region. A video consists of consequent frames and audio. To model a video, we extract visual and audio features. The extracted features can be used to compute the similarity between videos. The feature can be represented as $f = (\phi_1, \ldots, \phi_N)$.

4.1 Visual features

Visual features are important when measuring the similarity between two videos. When two videos show visually similar shots they tend to attract similar audiences. To extract visual features, we do sampling and get a part of frames from a video. To reduce the computation overhead, we sample one frame from every 5-s frames. Two kinds of visual features are extracted on each sampled frame and concatenated to form one visual feature vector. Visual feature vector is first extracted on each sampled frame and then the average vector is computed and set as the visual feature vector of the video. In [7], a few visual features have been tested in interestingness recognition. Here, we employ the two features which achieve best performance when fusing two features in their work. Our intuition is interestingness recognition depends on video content and is similar to a similarity comparison process. Hence the effective visual features in [7] should also yield good performance in terms of video similarity computation.

Sparse SIFT (Scale Invariant Feature Transform) feature. SIFT feature is a very popular and effective feature in applications like image retrieval and object recognition. In this work, for each sampled frame, we employ the same features used in [17]. In detail, interest points are firstly detected, respectively, by using SIFT detector [11] and MSER (Maximally Stable Extremal Regions) [12] detector on each sampled frame, and then the SIFT descriptors are computed on each individual interest point region. In the following, the SIFT descriptors are quantized by following the bag-of-words (BOW) representation using a spatial pyramid with a codebook of 500 words. The BOW feature vectors extracted from two types of interest points are concatenated and form a higher dimensional feature vector.

HOG2x2 (Histogram of Oriented Gradients) features. HOG feature was first proposed for pedestrian detection and achieved state-of-the-art performance [3]. Unlike sparse SIFT features which is extracted on sparsely distributed interest points, HOG feature is computed on densely sampled image patches. Following [23], HOG feature vectors from 2 × 2 neighborhood are further stacked together to form a descriptor in order for more descriptive power. The new feature vector is quantized into 300 visual words and then spatial pyramid feature is computed similar to the sparse SIFT feature computation.

4.2 Audio features

Audio effect is very important for a TV program. Without sound, audience cannot understand the dialogs or catch the tone between the roles. In particular, for talk show or music election show type program, audio plays more important part than visual. Audio feature plays an important role when comparing two videos. In this work, we choose the two features with best performance of two-feature subset in [7].

MFCC (*Mel-Frequency Cepstral Coefficiency*) MFCC is a frequently used audio feature in applications such as speech recognition [5] and music information retrieval [13]. It works well for audio similarity measure. We divide audio into 32-ms-long audio frames with 50 % overlap and compute MFCC on each audio frame. Similarly, MFCC features are also quantized into BOW representation.

Spectrogram SIFT We use the feature used in [7]. An image is synthesized based on the constant-Q spectrogram of the processed audio and the energy distribution is visualized. On the energy map, the SIFT descriptors are computed and further quantized into BOW.

4.3 Similarity metric learning for videos

With the visual feature vector and audio feature vector, we concatenate them and form a $P \times 1$ feature vector $\mathbf{v}$ to represent the video. However, with the feature vector, how to compute the distance is still a problem so far. The frequently used Euclidean distance, Cosine distance, or intersection distance might be candidates; however, they work well only in terms of single feature type. In terms of multiple features, a good distance metric is in need.

Suppose we have a training video set $\mathbf{S} = \{S_1, S_2, \ldots, S_M\}$ containing different types of videos, where the video set consists of M subsets corresponding to different categories of videos. For each subset S_i, it contains N_i videos and there are totally $N = \sum_i N_i$ videos from all subsets. Here we propose to learn the distance metric and employ a method that is similar to [15]. Our objective for distance metric learning is to minimize the distance of videos from same category and maximize the distance of videos from different categories, which can be written as

$$\min \left\{ \sum_{\mathbf{v}_i \in \mathbf{S}} \left(\sum_{\mathbf{q} \in S_i^+} d(\mathbf{v}_i, \mathbf{v}) - \sum_{\mathbf{v} \in S_i^-} d(\mathbf{v}_i, \mathbf{v}) \right) \right\}, \tag{7}$$

where S_i^+ denotes the subset video $\mathbf{v}_i$ belongs to, and S_i^- denotes the subset video $\mathbf{v}_i$ does not belong to. To simplify the computation, formula (7) can be rewritten as

$$\min \left\{ \sum_{i=1}^{N} \left(\sum_{j=1}^{n^+} d\left(\mathbf{v}_i, \mathbf{v}_{j+}\right) - \sum_{j=1}^{n^-} d\left(\mathbf{v}_i, \mathbf{v}_{j-}\right) \right) \right\}, \tag{8}$$

where $\mathbf{v}_j+$ is a video in subset S_i^+, n^+ is the number of videos in S_i^+, $\mathbf{v}_j-$ is a video from one of subsets S_j^- with shortest of distance to $\mathbf{v}_i$ in that subset.

Here we define the distance of two videos using Mahalanobis distance metric

$$d\left(\mathbf{v}_i, \mathbf{v}_j\right) = \left(\mathbf{v}_i - \mathbf{v}_j\right)^T \mathbf{W} \left(\mathbf{v}_i - \mathbf{v}_j\right), \tag{9}$$

where $\mathbf{W}$ is the weight matrix. When $\mathbf{W}$ is an identity matrix formula (9) corresponds to Euclidean distance. To make the problem well-posed we add a constraint $\det((W)) = 1$. With the Lagrange multipliers method, the optimization problem depicted in (8) can be obtained:

$$L = \sum_{i=1}^{N}\sum_{j=1}^{n_+} d\left(\mathbf{v}_i, \mathbf{v}_{j+}\right) - \sum_{i=1}^{N}\sum_{j=1}^{n_-} d\left(\mathbf{v}_i, \mathbf{v}_{j-}\right) + \lambda(\det(\mathbf{W}) - 1) \tag{10}$$

To optimize (10) we can derive:

$$\frac{\partial L}{\partial w_{st}} = \sum_{i=1}^{N}\sum_{j=1}^{n_+} \frac{\partial d\left(\mathbf{v}_i, \mathbf{v}_{j+}\right)}{\partial w_{st}} - \sum_{i=1}^{N}\sum_{j=1}^{n_-} \frac{\partial d\left(\mathbf{v}_i, \mathbf{v}_{j-}\right)}{\partial w_{st}} + \lambda(-1)^{s+t}\det\left(\overline{\mathbf{W}}_{st}\right), \tag{11}$$

where w_{st} is the element with index of (s, t) of $\mathbf{W}$, and $\overline{\mathbf{W}}_{st}$ is a $(P-1)\times(P-1)$ matrix generated by removing the sth row and the tth column of $\mathbf{W}$. We set $D_{st}^{+} = \sum_{i=1}^{N}\sum_{j=1}^{n_+} \frac{\partial d\left(\mathbf{v}_i, \mathbf{v}_{j+}\right)}{\partial w_{st}}$ and $D_{st}^{-} = \sum_{i=1}^{N}\sum_{j=1}^{n_-} \frac{\partial d\left(\mathbf{v}_i, \mathbf{v}_{j-}\right)}{\partial w_{st}}$. Let $\partial L/\partial w_{st} = 0$, it can be obtained

$$\det\left(\overline{\mathbf{W}}_{st}\right) = \frac{D_{st}^{-} - D_{st}^{+}}{\lambda(-1)^{s+t}}, \tag{12}$$

where D_{st}^{+} can be calculated according to (13).

$$\begin{aligned} D_{st}^{+} &= \sum_{i=1}^{N}\sum_{j=1}^{n_+} \frac{\partial\left(\mathbf{v}_i - \mathbf{v}_{j+}\right)^T \mathbf{W}\left(\mathbf{v}_i - \mathbf{v}_{j+}\right)}{\partial w_{st}} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{n_+} \left(\mathbf{v}_{j+}(s) - \mathbf{v}_k(s)\right)\left(\mathbf{v}_{j+}(t) - \mathbf{v}_i(t)\right). \end{aligned} \tag{13}$$

D_{st}^{-} can be calculated in the same way as (13).

Let $\mathbf{A} = \lambda\mathbf{W}^{-1} = [a_{st}]$, i.e., a_{st} is the (s, t)th element of $\mathbf{A}$. Then we can calculate the element of $\mathbf{A}$ with indices of (s, t) as

$$a_{st} = \frac{\lambda(-1)^{s+t}\det\left(\overline{\mathbf{W}}_{st}\right)}{\det(\mathbf{W})} = D_{st}^{-} - D_{st}^{+}. \tag{14}$$

With $\mathbf{A}$ we can get λ. Since we have $\det(\mathbf{A}) = \lambda^P \det\left(\mathbf{W}^{-1}\right) = \lambda^P$, then

$$\lambda = (\det(\mathbf{A}))^{\frac{1}{P}}. \tag{15}$$

So $\mathbf{W}$ can be computed as

$$\mathbf{W} = \lambda\mathbf{A}^{-1}. \tag{16}$$

With the learnt distance metric $d(\mathbf{v}_i, \mathbf{v}_j)$, the similarity of two videos is defined as

$$\mathrm{SIM}_{ij}^{V} = \exp\left(-\frac{d(\mathbf{v}_i, \mathbf{v}_j)}{\sigma_v^2}\right), \tag{17}$$

where σ_v is empirically determined.

4.4 Datasets for distance metric learning

To train the distance metric, we use the dataset made public in [7]. In [7], there are two parts of datasets, one is acquired from Flickr with 15 different categories with a specific subject, and the other is from YouTube with another 14 categories. We combine the two datasets and get overall 29 different subsets for training. We make an assumption that the videos are similar if they are from the same category and different if from different categories.

During learning, we treat each individual video equally, which means all the videos will be used for training although in [7] videos are associated with levels of interestingness.

5 Experiments

5.1 Settings

To demonstrate the proposed method, we collect data from multiple sources. We first crawl social network data from Sina Weibo. To cover more tweets, for each TV program we collect some keywords which accurately relate to the TV program and introduce no ambiguity in meantime. The keywords usually include the name of the TV programs and the role names in the TV programs.

To extract the location information, we extract the registration province of the microblog owner after getting the microblog data. Hence each microblog is now associated with a location. Most of the time, the registration place is the place this user is currently living in or be active in, so this property can reflect the comments of this province. After we get all the tweets related to the TV program, we divide the microblogs into different provinces according to the registration province. There are 15 TV programs in our dataset.

The videos for each TV program are downloaded from Letv.[3] We assume a TV program is usually consistent and can be represented by one of its episodes. Thus, for each program we only pick the first episode.

5.2 Experimental results

In order to evaluate the performance of the proposed method, a dataset of TV shows and their corresponding discussions on social network is collected.

In the TV show dataset, we selected 14 popular shows on social networks, including “Dad, Where Are We Going”, “Longmen Express”, “I Am Singer”, “The Voice of China”,

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

Table 1 The hit-rate of the proposed method for different TV shows

TV show name	HR@1	HR@5	HR@10
Beautiful time	0	0.2	0.3
The voice of China	0	0.2	0.2
Happy camp	0	0.2	0.4
I am singer	0	0.2	0.3
Longmen express	0	0.2	0.3

etc. The videos are collected from internet. Then, by using the title of the show as keywords, we queried Sina Weibo and relevant tweets are retrieved. By further identifying the registered locations of the authors of the tweets, the location-aware tweets for different TV shows are obtained.

The experiments are divided into two parts. In the first part, we evaluate the location-aware popularity prediction performance for new TV shows. In the second part, given the popularity of a show in certain regions, we predict its popularity for other regions. The details are given in the following sections.

In the first setup, we selected 9 shows and their corresponding location-aware social network discussions as training data. The remaining 5 shows are considered as testing data. The aim is to predict the popularity of these shows across different locations. We randomly selected “Beautiful Time”, “The Voice of China”, “Happy Camp”, “I am Singer”, and “Longmen Express” as test set, and the remaining shows as training set.

The quality was measured by looking at the number of hits and their position within the top $-N$ locations that were recommended by a particular scheme [4]. The number of hits is the number of top $-N$ locations in the test set that were also present in the top $-N$ recommended locations returned for each TV show. We will refer to the quality measure as the hit-rate @ p (HR@p) that is defined as follows. If p is the top $-p$ locations in the test set, the hit-rate @ p of the recommendation algorithm was computed as:

$$\text{HR@}p = \frac{\text{Number of hits}}{p} \tag{18}$$

An HR value of 1.0 indicates that the algorithm was able to always recommend the top $-p$ popular locations for a TV show, whereas an HR value of 0.0 indicates that the algorithm was not able to recommend any of the popular locations.

The results are shown in Table 1 and Fig. 3.

As can be seen in the results, the top -1 hit-rate for all the test shows are 0. It means that the proposed method can hardly predict the most popular locations for the particular shows. It indicates the popularity of the shows is affected by not only the content but also many other factors. However, the hit-rate increases as p increases. When $p = 10$, the proposed method can predict around 30 % popular locations. It indicates that the content and culture do have a strong effect on the popularity of the shows across different locations.

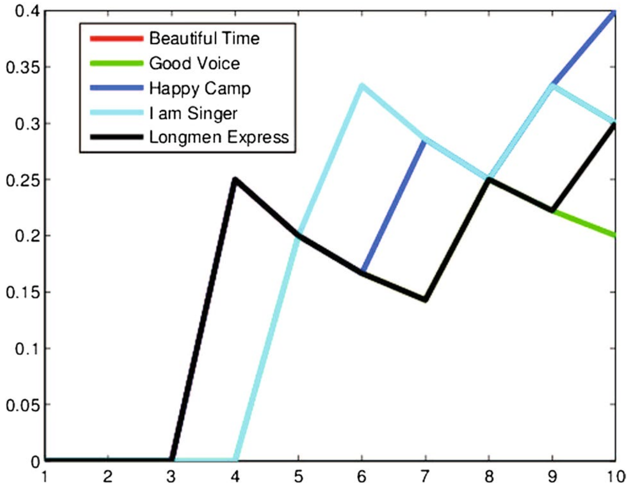

Fig. 3 The hit-rate of the proposed method for different TV shows

In the second setup, we randomly remove 20 % of the popularity information, and utilize our proposed method to recover the information. The widely used mean square error of prediction is adopted as the performance evaluation measure. If $\hat{X}$ is the predictions of n missing values, and X is the corresponding true values, then the MSE of the proposed method is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_i - X_i\right)^2 \tag{19}$$

The prediction error in the experiment is 0.0011. It shows that the proposed method can achieve reasonable error. That is, given a show played at certain locations, we may roughly estimate the popularity for locations where the show has not been broadcast yet with the proposed method.

6 Conclusion

In this paper, a preliminary location-aware TV show recommendation scheme is proposed. By incorporating the social network information of users from different locations, a location profile is obtained. Location similarity is then calculated by combining location profile and geographical information. Video similarity is obtained by considering both visual and audio information. By taking the location similarity and video similarity as the regularizer, the scheme makes prediction on TV show popularity for different regions via graph regularized non-negative matrix factorization. A TV show dataset with location-aware social network information is collected and the proposed method achieves promising results on it. In the future, more information will be investigated for better recommendation performance.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1548–1560 (2011)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
4. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. (TOIS) **22**(1), 143–177 (2004)
5. Ganchev, T., Fakotakis, N., Kokkinakis, G.: Comparative evaluation of various mfcc implementations on the speaker verification task. Proc. SPECOM **1**, 191–194 (2005)
6. Gao, Y., Wang, M., Zha, Z., Shen, J., Li, X., Wu, X.: Visual-textual joint relevance learning for tag-based social image search. IEEE Trans. Image Process. **22**(1), 363–376 (2013)
7. Jiang, Y.G., Wang, Y., Feng, R., Xue, X., Zheng, Y., Yang, H.: Understanding and predicting interestingness of videos. In: Proceedings of The 27th AAAI Conference on Artificial Intelligence (AAAI) (2013)
8. Kaminskas, M., Ricci, F., Schedl, M.: Location-aware music recommendation using auto-tagging and hybrid matching. In: Proceedings of the 7th ACM conference on Recommender systems, pp. 17–24 (2013)
9. Kim, E., Pyo, S., Park, E., Kim, M.: An automatic recommendation scheme of tv program contents for (ip) tv personalization. IEEE Trans. Broadcast. **57**(3), 674–684 (2011)
10. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
12. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
13. Müller, M.: Information retrieval for music and motion, vol. 2. Springer, Berlin (2007)
14. Ji, R., Duan, L.-Y., Chen, J., Huang, T., Gao, W.: Mining compact 3D patterns for low bit rate mobile visual search. IEEE Trans. Image Process. **23**(7), 113–3099 (2014)
15. Rui, Y., Huang, T.: Optimizing learning in image retrieval. In: IEEE conference on Computer Vision and Pattern Recognition, vol. 1, pp. 236–243 (2000)
16. Shen, Y.: Analyze the regional cultural difference of tv shows in china based on audience rating. Todays Mass Media (in Chinese) **21**(1), 77–78 (2012)
17. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision, pp. 1470–1477 (2003)
18. Son, J.W., Kim, A., Park, S.B., et al.: A location-based news article recommendation with explicit localized semantic analysis. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 293–302 (2013)
19. Son, J.W., Noh, Y.S., Song, H.J., Park, S.B.: Location comparison through geographical topics. In: IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 311–318 (2012)
20. Ji, R., Duan, L.-Y., Chen, J., Yao, H., Yuan, J., Rui, Y., Gao, W.: Location discriminative vocabulary coding for mobile landmark search. Int. J. Comput. Vis. **96**(3), 290–314 (2012)
21. Wang, S.: The regional analysis on high audience rating tv shows. Todays Mass Media (in Chinese) **21**(12), 88–89 (2012)
22. Winter, S.: Location-based similarity measures of regions. Int. Arch. Photogramm. Remote Sens. **32**, 669–676 (1998)
23. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: IEEE conference on Computer vision and pattern recognition, pp. 3485–3492 (2010)
24. Gao, Y., Tang, J., Hong, R., Dai, Q., Chua, T.-S., Jain, R.: W2Go: a travel guidance system by automatic landmark ranking. In: Proceedings of the international conference on Multimedia, pp. 123–132 (2010)
25. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.: Geographical topic discovery and comparison. In: Proceedings of the 20th international conference on World wide web ACM, pp. 247–256 (2011)