Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Matrix Factorization+ for Movie Recommendation

Lili Zhao,[†] Zhongqi Lu,[†] Sinno Jialin Pan,[*] Qiang Yang[†]
[†]Hong Kong University of Science and Technology, Hong Kong
[*]Nanyang Technological University, Singapore
[†]{lzhaoae, zluab, qyang}@cse.ust.hk, [*]sinnopan@ntu.edu.sg

Abstract

We present a novel model for movie recommendations using additional visual features extracted from pictural data like posters and still frames, to better understand movies. In particular, several context-based methods for recommendation are shown to be special cases of our proposed framework. Unlike existing context-based approaches, our method can be used to incorporate visual features – features that are lacking in existing context-based approaches for movie recommendations. In reality, movie posters and still frames provide us with rich knowledge for understanding movies as well as users' preferences. For instance, user may want to watch a movie at the minute when she/he finds some released posters or still frames attractive. Unfortunately, such unique features cannot be revealed from rating data or other forms of context being used in most of existing methods. In this paper, we take a step forward in this direction and investigate both low-level and high-level visual features from the movie posters and still frames for further improvement of recommendation methods. Extensive experiments on real world datasets show that our approach leads to significant improvement over several state-of-the-art methods.

1 Introduction

The problem of movie recommendation can be defined as follows: given a set of users and a set of movies, the goal is to find the potential movies that a user may be interested in based on the user's historical behaviors or preferences on movies. One promising approach in this respect is learning latent features and relation features [Rennie and Srebro, 2005; Salakhutdinov and Mnih, 2007; Koren *et al.*, 2009]. A major discussion in most of the existing work on recommendations has been about scarce historical data. For example, in a movie recommendation system like Netflix[1], the average user rates only about 200 movies. Compared with tens of thousands of movies in the database, the rating set is too sparse to learn a well-performed model. It is desirable that considering additional information may be able to help recommendations.

Naturally, research on the problem of context-based movie recommendation has gained a lot of attention: given a set of users, a set of movies and some context, find the underlying movies that users may be interested in. The context may include movie attributes, user demographics, social networks or movies reviews, etc. These methods are expected to alleviate the sparsity issue, thus to improve the quality of recommendations because the factors behind prediction are assumed coming from two parts, rating and context. When rating is not available, the prediction can be still inferred from context.

However, we find that some existing recommendation systems based on context information only give minor improvements above the rating based methods. The prediction quality even drops when the context is sparser than rating data. A notable drawback of these methods is that they only leverage the value of context in the way of basic regularization in the model. The most common assumption is that the user/movie preferences are related with the context. For example, if two users have some common friends, they probably share particular tastes for movies. Generally, this assumption tends to narrow down the preference space, it can not bring more accurate learning on preferences. A less notable issue with current context-based methods is that it does not address plenty movie features, such as movie posters and still frames, which limits its power for recommendations. An interesting observation as illustrated in Figure 1 demonstrates the idea. Movie posters and still frames actually reveal a great amount of information to open the mystery of user behaviors, which can not be derived from other forms of context. For example, when a user is watching one movie presented in cold, blue and mysterious visual effects, the user may be interested in receiving recommendations for movies with similar styles, rather than others like casted by the same actor or actress. From the posters and still frames, we can extract such features, and then utilize they to better understand movies as well as users.

In this paper, we explore the potential of integrating visual features to improve context-based Matrix Factorization methods for movie recommendations, which we call Matrix Factorization+, abbreviated as MF+. Our method first identifies a set of useful visual features from movie posters and still frames, then embeds them into a model for movie prediction. More concretely, we extend a context-based Ma-

[1]www.netflix.com

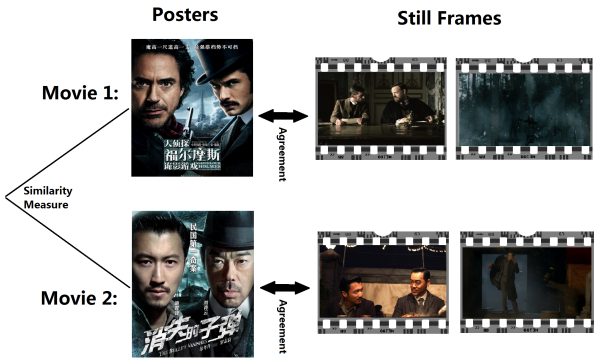

Figure 1: In this figure, we present one poster and two representative still frames for each movie. These two movies have different genres and totally different casts. Based on case study, the audiences who favor Movie 1 would also be interested in Movie 2. We also notice that Movie 2 is actually inspired by Movie 1. This indicates that visual features are strong signs for movie recommendations, and we should capture this relationships in making good recommendations.

trix Factorization model as applied to movie recommendation. Motivated by a recently proposed *Learning Using Privileged Information paradigm* [Vapnik and Vashist, 2009; Pechyony and Vapnik, 2010], which uses additional information of different kind, we model both bias and regularizations by considering visual features, training a latent factor model to make predictions for users. Another novelty of our method is that, through investigating visual features, we can understand user/movie preferences in a new aspect, e.g., we consider that recommending a movie with similar visual effects is better than one with same casts. This way of modeling improves the overall performance.

In the followings, we start by discussing the related works and introducing some preliminary notations. Then we present the unique movie recommendation settings, which involve movie posters and still frames. Under such settings, we propose our model, focusing on the above two key points. Finally, we report the experimental results and discuss the insights in this direction.

2 Related Works

This paper proposes a moving forward step of recommending movies for users. Crucially, we would like to consider adding features extracted from movie posters and still frames to predict users' movie watching interests, so that the proper movies could be recommended. This work is mostly related to the following fields.

Matrix Factorization for Recommendation Matrix factorization (MF) [Rennie and Srebro, 2005; Paterek, 2007; Marlin, 2003; Koren *et al.*, 2009; Singh and Gordon, 2008] is one family of state-of-the-art algorithms in the application of recommendation. Our work is an extension of current matrix factorization methods, by combining knowledge from visual data that contains rich features. In traditional matrix factorization, the problem can be formulated as inferring missing values of a partially observed User-Item matrix $\mathbf{X}$: each row represents a user u, each column an item v. Then, one can model user/item preference within each matrix entry x_{uv} by low-rank factor matrices $\mathbf{U} \in \mathbb{R}^{k\times m}$ and $\mathbf{V} \in \mathbb{R}^{k\times n}$, respectively, where the u-th user and the v-th item are represented by $\mathbf{U}_{*u}$ and $\mathbf{V}_{*v}$, corresponding to the u-th and v-th column of preference matrices $\mathbf{U}$ and $\mathbf{V}$. A popular matrix factorization model is the Probabilistic Matrix Factorization model (PMF) [Salakhutdinov and Mnih, 2007], where the objective function is equivalent to minimizing the sum of squared errors with quadratic regularization terms as follows,

$$\mathcal{L} = \sum_{u=1}^{m}\sum_{v=1}^{n}\left(\mathbf{U}_{*u}^{T}\mathbf{V}_{*v} - x_{uv}\right)^2 + \lambda\mathcal{R}(\mathbf{U},\mathbf{V}) \qquad (1)$$

where λ is a trade-off parameter. Inevitably, MF approaches may still suffer from the sparsity problem in recommender systems, where the learned models may be overfitting to the small set of observed ratings.

MF with Context For solving the sparse issue in recommender systems, many works consider including context that is proven to be useful for improving the recommendations: movie attributes, user demographics, social networks or reviews about movies, etc. Here, we only list a few representative works that contributed to this direction. At early stage, attribute information, such as user's demographics, item's category or content were commonly used [Koenigstein *et al.*, 2011; Moshfeghi *et al.*, 2011; Lu *et al.*, 2013; Zhao *et al.*, 2013]. Due to the privacy issue, works on using attribute information are limited adopted. The emergence of social networks has led to another trend on considering context [Ma *et al.*, 2008; 2011; Li and Yeung, 2009]. This relationship can be encoded that users who are within a positive relationship also share the similar preferences, vice versa. Although social networks have brought leverage about recommender systems, still the sparsity issue cannot be solved perfectly due to the sparsity natural of social network its own. Subsequently, with the introduction of Web 2.0 technology, user-generated content, such as tags, reviews, were widely used as new context. Sen *et al.* [2006] used short textual labels that users assigned to items as user's profiles. Like tags, reviews are another type of context that is generated by users. Research works in this area, have not only considered the semantic meaning of reviews, but also explored the sentiment/emotional dimensions [Levi *et al.*, 2012; Moshfeghi *et al.*, 2011]. This form of context is valuable, but needs sophisticated tools to analyse. Last but not least is the user behaviors that differ from ratings, we indicate them as implicit feedback. Implicit feedback is originated from the area of information retrieval and the related techniques have been successfully applied in the domain of recommender systems [Kelly and Teevan, 2003; Rendle *et al.*, 2009; Koren, 2008; Oard *et al.*, 1998; Singh and Gordon, 2008; Lu *et al.*, 2015]. Usually, the implicit feedbacks are inferred from user behaviors, such as browsing items, marking items as like/dislike, etc. Intuitively, the implicit feedback approaches are based on an assumption that implicit feedbacks could be used to regularize or supplement the explicit rating behaviors.

Compared to above forms of context, there are relatively few works on considering visual data to recommendation task. We speculate that this is partly due to the lack of large-scale visual data associated with rating data. Commonly used datasets (MovieLens[2], Netflix, EachMovie[3]) only contain rating data and some provide with meta data about movie and user attributes. Despite of the data limitations, some social media sties, like Youtube [Davidson *et al.*, 2010], have made some extensions of multimedia data to facilitate recommendations. For instance, the type of video clips that user has posted may reflect user's tastes. We consider this extension as only one of the possible generalizations in this work.

3 Preliminary

3.1 Notations

In a standard recommendation setting, we have an extremely sparse preference matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m is the number of users and n is the number of items. Each entry x_{uv} of $\mathbf{X}$ corresponds to user u's preference on item v. If $x_{uv} \neq 0$, it means for user u, the preference on item v is observed, otherwise unobserved. Let $\mathcal{I}$ be the set of all observed (u, v) pairs in $\mathbf{X}$. The goal is to predict users' unobserved preferences based on observed ones. For rating-based recommender systems, preferences are represented by numerical values (e.g., $[1, 2, ..., 5]$, one star through five stars),where higher values indicate stronger preferences. We use χ_v to represent poster feature vectors of move v, and ψ_v to represent still frame feature vectors. The predicted value is represented by $\hat{x}_{uv}$. We use the superscript $^\top$ to denote the transpose of a matrix.

3.2 Matrix Factorization Models

Matrix Factorization models comprise an important approach to recommendation. A major advantage of the models is to tackle the aforementioned sparsity issue. We will focus on the models that are induced by the Singular Value Decomposition (SVD) on the user-movie preference matrix. A typical model associates each user u with a user-factor vector $\mathbf{U}_{*u}$, and each movie v with a movie-factor vector $\mathbf{V}_{*v}$. The prediction is then given by

$$\hat{x}_{uv} = b_{uv} + \mathbf{U}_{*u}^T \mathbf{V}_{*v},$$

where b_{uv} denotes a baseline estimate for an unknown rating x_{uv}:

$$b_{uv} = \mu + b_u + b_v,$$

and μ is the overall average rating, b_u and b_v indicate the biases of user u and movie v, respectively.

3.3 Neighborhood Models

One set of popular extended models from MF are neighborhood models, which estimate unknown ratings based on either like-minded users or similar movies. While the neighbors selection could be either movie-oriented or user-oriented, in our work, we focus on the movie-oriented method. The user-oriented method could be derived in a similar way. As suggested by [Koren, 2008], one can model the latent factors of a movie v by its neighbors $N(\theta, v)$, based on some similarity measure θ. Specifically, $N(\theta, v)$ could be a neighborhood selection function, which returns the neighbors of v when the similarity measured by θ exceeds certain boundary. The prediction is given by

$$\hat{x}_{uv} = b_{uv} + \mathbf{U}_{*u}^\top \left(\mathbf{V}_{*v} + |N(\theta, v)|^{-\frac{1}{2}} \sum_{s \in N(\theta, v)} y_s \right), \quad (2)$$

where the term $\mathbf{V}_{*v} + |N(\theta, v)|^{-\frac{1}{2}} \sum_{s \in N(\theta, v)} y_s$ is defined as the latent factors of a movie v, and y_s is latent factors of implicit feedback to describe the neighbor movie s of v.

4 The Movie Recommendations

4.1 Intuition of Our Design

The setting in the domain of movie recommendations is unique to other domains. Movies, especially those not released yet, are most likely to be first exposed to users via posters. Thus, they would be an immediate representations of the users' expectations towards the movies. In most cases, if users find posters to be attractive, then they would want to watch the movie. Proper ratings would be given to the movie if the users' positive expectations are reached, i.e. the features conveyed in posters and those in movies (represented by a set of still frames) are consistent. To make recommendations under this scenario, it is desirable to design a more sophisticated model by integrating the agreement between movie posters and still frames.

4.2 Visual Features in Movies

Before we dive into the model part, we would like to elaborate several features that we consider in recommendations. Since visual features are rich, we need to choose them in a proper way, so that the recommendations can be well-performed. We assume that the triggers inside the visual effects for watching a movie include colors, abstract features and content, like the illustration in Figure 1. To be more specific, given a movie v, we are interested in the visual agreement of its posters and still frames, denoted as $\phi(v)$. We also would like to know the similarity θ_{vs} between movie v and s by measuring how they are visually correlated.

Color histogram In posters and still frames, color is the first impression. Among all the elements in filming a movie, color is also the key factor to trigger audience's emotions. For instance, yellow usually gives us feelings of brightness and liveliness. In movies, directors use yellow to express happiness, like the movie "Minions". And, blue is usually used to exhibit cold and depress, like the movie "Trois couleurs: Bleu" is fully filled with blue to express the inside feeling of the lead character. We adopt a standard color histogram feature, computed on posters and still frames, which is 576-dimensional joint histogram in RGB color space has 8,8 and 8 bins in R,G,B channels.

SIFT The classic SIFT descriptor [Lowe, 2004] is known to allow for an object to be recognised in a larger image

[2]http://grouplens.org/datasets/movielens/

[3]rouplens.org/datasets/eachmovie/

datasets. Also, image SIFT features allow for objects in multiple images of the same location, taken from different positions within the environment, to be recognised. SIFT features are also very resilient to the effects of "noise" in image. Due to the fact that many of the movie still frames are taken from the same scene but from different angles, or from same people but in different scenes, this feature can be useful. We extract 128-dimensional SIFT features after resizing posters and still frames to 256-by-256 pixels.

Convolutional Neural Networks Deep convolutional neural network can discover multiple levels of abstract representation of images, some of which may be useful for recommendations. We use the state-of-the-art architecture Caffe [Jia *et al.*, 2014] on ImageNet to extract such features. This structure is ImageNet challenge winning model, we adopt eight layers due to the reason that it not only perform well, but also computationally efficient. Follow the work in [Donahue *et al.*, 2013], we investigate using features from different layers, referred to as $Caffe_5$ and $Caffe_6$. The features are with 8000- and 4000-dimensions.

Category Features The category feature is used to indicate whether individual poster or still frame belongs to any predefined category. We adopt 1000 object categories that used in ImageNet challenge. Specifically, we present each poster and still frame to challenge winning model and use the predicted category precision result as features.

4.3 MF+ Model

Below we describe our model to incorporate with visual features into a Neighborhood model, named MF+.

Based on the neighborhood models as described in Section 3.3, the prediction of user u's interests on a movie v is given by

$$\hat{x}_{uv} = b_{uv} + \mathbf{U}_{*u}^T(\mathbf{V}_{*v} + \eta). \tag{3}$$

In the application of movie recommendations, with the still frames as additional information, we propose an improved version of the basic Neighborhood model. Traditionally, the term b_{uv} takes the form as introduced in (2). Recent works [Vapnik and Vashist, 2009; Pechyony and Vapnik, 2010] have shown that a learning model trained on both additional information and traditional information provides improved performance compared with the model trained solely on traditional information. Inspired by this idea, in our setting, we consider each data instance as a composite of a rating x and some still frame visual features ψ, of which still frame features are our additional information. Ideally if we have known the joint distribution of data, we have that the conditional mutual information $I(x_{uv}, \psi_v|\mathbf{U}_{*u}, \mathbf{V}_{*v}) = H(x_{uv}|\mathbf{U}_{*u}, \mathbf{V}_{*v}) - H(x_{uv}|\mathbf{U}_{*u}, \mathbf{V}_{*v}, \psi_v)$ is always non-negative, where $H(\cdot|\cdot)$ is the conditional entropy. Therefore, including feature information can lead to reduction of uncertainty about the bias b_v. Thus, in our model, we replace b_v by a linear function of ψ_v, $g(\psi_v) = W_{*v}^T\psi_v$, where W_{*v}^T is the v-th column of a weight matrix W to be learned.

Besides, we propose to model the latent features of movie v as $\mathbf{V}_{*v} + \eta$. We use the movie vector $\mathbf{V}_{*v}$ to represent the latent features from the movie v itself, and the latent feature vector is complemented by the visual features η, which is in proportion to the latent features of those similar movies and in reverse proportion to the consensus between the poster and still frames of movie $\phi(v)$, written as:

$$\eta = \frac{|N(\theta, v)|^{-\frac{1}{2}} \sum_{s \in N(\theta,v)} \theta_{sv}\tilde{\chi}_s}{\phi(v)} \tag{4}$$

where $\tilde{\chi}_\star = (\chi_\star, \psi_\star)$, composition of still frame and poster features. θ_{sv} is the interpolation weight to measure the similarity between movie v and s.

The reason behind the reverse proportion about posters is that usually, as audience, we expect the content within posters can tell a bit of clue about what story the movie is about instead of simply piling up all the characters. That is to say, the features between posters and still frames should be similar. Sometimes, posters may contain confused information, that is also the reason why we have not considered adding poster features in estimating b_v above. For computing $\phi(v)$, we adopt inner product operation.

The model parameters associated with the prediction rule in 3 are learned by solving the regularized least squares problem

$$\min_{\substack{b_\star, W_\star, \theta_\star \\ \mathbf{U}_\star, \mathbf{V}_\star}} \sum_{(u,v)} \Big(\lambda_1 b_u^2 + \lambda_2 W_{*v}^2 + \lambda_3 \|\mathbf{U}_{*u}\|^2 + \lambda_4 \|\mathbf{V}_{*v}\|^2 + \lambda_5 \theta_{sv}^2 + \big(x_{uv} - \mu - b_u - W_{*v}^T\psi_v - \mathbf{U}_{*u}^T(\mathbf{V}_{*v} + \eta)\big)^2 \Big) \tag{5}$$

where $\lambda_\star$ are regularization constants. In our work, we adopt the same calculation method presented in [Koren, 2008] for neighborhood selection.

We estimate the model parameters by minimizing the regularized squared error function through stochastic gradient descent. To ease the presentation, we define $e_{uv} = x_{uv} - \hat{x}_{uv}$. For a particular user-movie pair (u, v), we update the parameters by moving in the opposite direction of the gradient, yielding:

$$\begin{aligned}
b_u &\leftarrow b_u + \gamma_1(e_{uv} - \lambda_1 b_u) \\
W_{*v} &\leftarrow W_{*v} + \gamma_2(e_{uv}\psi_v - \lambda_2 W_{*v}) \\
\mathbf{U}_{*u} &\leftarrow \mathbf{U}_{*u} + \gamma_3(e_{uv}(\mathbf{V}_{*v} + \eta) - \lambda_3 \mathbf{U}_{*u}) \\
\mathbf{V}_{*v} &\leftarrow \mathbf{V}_{*v} + \gamma_4(e_{uv}\mathbf{U}_{*u} - \lambda_4 \mathbf{V}_{*v}) \\
\forall s &\in N(\theta, v): \\
\theta_{sv} &\leftarrow \theta_{sv} + \gamma_5(e_{uv}\mathbf{U}_{*u} \mid N(\theta, v) \mid^{-\frac{1}{2}} \tilde{\chi}_s - \lambda_5 \theta_{sv})
\end{aligned}$$

where $\gamma_\star$ are constants for the step size.

5 Experiments

5.1 Datasets

We evaluate on the Netflix and Douban datasets. Since the two original datasets do not have movie posters and still frames, we crawled these data from web. Besides, we also crawled directors, genre and leading actors for each movie for

Method	RMSE
Neighborhood Model	0.8734
Neighborhood Model+Fcolor	0.8544
Neighborhood Model+Fcolor+Fsift	0.8502
Neighborhood Model+Fcolor+Fsift+Fcaffe	0.8367
Neighborhood Model+Fcolor+Fsift+Fcaffe+Fcategory	0.8289
Neighborhood Model+Pcolor	0.8765
Neighborhood Model+Pcolor+Psift	0.8777
Neighborhood Model+Pcolor+Psift+Pcaffe	0.8643
Neighborhood Model+Pcolor+Psift+Pcaffe+Pcategory	0.8614
Neighborhood Model+full visual features	**0.8124**

Table 1: Analysis of different visual features used in $\tilde{\chi}_\star$ proposed in our method on Douban data. Neighborhood Model refers to training model introduced in Section3.3 only on rating data. The letters "P" and "F" refer to the posters and still frames, respectively. "+" means concatenate operation when computing neighbors. For example, +color means that adding color features. +caffe refers to using the aforementioned two levels of features from Caffe. "full visual features" means all features concatenated including color, sift, caffe, category from posters and still frames. Performance is measured with RMSE.

Method	CNN Feature Used	
	$Caffe_5$	$Caffe_6$
Neighborhood Model	0.8682	0.8723
Neighborhood Model+Fcolor	0.8489	0.8538
Neighborhood Model+Fcolor+Fsift	0.8456	0.8512
Neighborhood Model+Fcolor+Fcolor+Fcategory	0.8378	0.8479
Neighborhood Model+Pcolor	0.8645	0.8712
Neighborhood Model+Pcolor+Psift	0.8612	0.8709
Neighborhood Model+Pcolor+Psift+Pcategory	0.8607	0.8679
Neighborhood Model+{F,P}color+{F,P}sift+{F,P}category	**0.8112**	0.8176

Table 2: The impact of different level features from CNN. The numbers are the results from left hand combining features indicated in columns. For example, 0.8489 is the result coming from Neighborhood Model + Fcolor+ $Caffe_5$

the purpose of evaluation. We denote this part of information as Xmeta in our experiments. Before putting all the data to experiment, we filter out movies with less than 50 still frames. At the end, Netflix contains 675,236 movie still frames, 9138 posters of 6000 movies, while Douban has 415,484 movie still frames, 7523 posters of 5000 movies. For rating sparsity, we have 99.3% and 99.6% for Netflix data and Douban data, respectively.

We process each visual data in one movie by first computing four feature vectors as described in Section 4.2, and then averaging them on every feature type. Thus, for each movie, we have four feature vectors for the posters and still frames, respectively.

We split each dataset by assigning 80% to training set and the rest 20% to a test set. The parameters of our model, i.e., the number of latent factors k and the number of iterations T are tuned on Douban data, and fixed to the others. Here, $T = 30$, and $k = 20$. We adopt commonly used Root Mean Square Error (RMSE) as evaluation criterion,

$$\text{RMSE} = \sqrt{\sum_{(u,v)\in\mathcal{I}} \frac{(x_{uv} - \hat{x}_{uv})^2}{|\mathcal{I}|}},$$

where x_{uv} and $\hat{x}_{uv}$ are the ground truth and predicted ratings respectively, and $|\mathcal{I}|$ is the number of test ratings. The smaller the value, the better is the performance.

5.2 Performance Comparisons

We first perform a detailed analysis of our model on Douban before moving on to compare to other methods on both of Douban and Netflix.

Quantitative Results

Table 1 details the effect of different visual features used in $\tilde{\chi}_\star$ proposed in neighborhood model. For all the results presented, when no poster features nor still frame features are used, the agreement function $\phi(v)$ is set to be 1. As shown in the table, integrating visual features extracted from posters and still frames consistently outperform the Neighborhood Model. We observe that solely adding features from still frames works better than that from posters. This can be caused by two reasons, one is that the number of posters for each movie is small, usually less than five, resulting in variance. The other is that posters rarely convey any understanding of movies. Moreover, combining the features from CNN parts is always better than other strategies. This indicates that the CNN features are useful to express movie watching habit, and motivates our use for the rest of analysis. Furthermore, with full visual features used, we get the best performance. This proves the rational design of the agreement between posters and still frames.

The Impact of CNN Features

We conduct further experiments to analyze the effectiveness of CNN features from different levels, comparing $Caffe_5$

Method	RMSE
PMF	0.9082
MMMF	0.8909
RRMF	0.8743
CMF	0.8875
Neighborhood Model	0.8682
Neighborhood Model+Xmeta	0.8883
(MF-) + Pfully	0.8457
(MF-) + Ffully	0.8421
(MF-) + Pfully + Ffully	0.8389
(MF+) + Pfully	0.8374
(MF+) + Ffully	0.8342
(MF+) + Pfully +Ffully	**0.8289**

Table 3: Comparison of different methods on Douban data

Method	RMSE
PMF	0.8675
MMMF	0.8572
RRMF	0.8362
CMF	0.8534
Neighborhood Model	0.8421
Neighborhood Model+Xmeta	0.8578
(MF-) + Pfully	0.8237
(MF-) + Ffully	0.8210
(MF-) + Pfully +Ffully	0.8163
(MF+) + Pfully	0.8211
(MF+) + Ffully	0.8170
(MF+) + Pfully +Ffully	**0.8103**

Table 4: Comparison of different methods on Netflix data.

and $Caffe_6$. Table 2 details the results. In all cases, using a mid-level significantly improves results, so we present the remainder of the results using mid-level for CNN features.

Comparison to Several Baselines

In Table 3, we compare our methods to many prior works that have been done for movie recommendations, as listed below:

- PMF: Probabilistic Matrix Factorization model [Salakhutdinov and Mnih, 2007] is a low-rank approximation for rating prediction in recommender system,s as detailed in Section 3.2. This model only uses rating data.
- MMMF: Maximum Margin Matrix Factorization [Rennie and Srebro, 2005] is a low-norm approximation model for collaborative prediction in recommender systems. This model only uses rating data.
- RRMF: Relation Regularization Matrix Factorization [Li and Yeung, 2009] is a model using relation information to regularize the factorization procedure. In this paper, we use full visual features matrix serves as relation information.
- CMF: Collective Matrix Factorization [Singh and Gordon, 2008] is a model considering different sources of information by simultaneously factorizing multiple matrices. In this paper, the two factorized matrices are visual feature matrix and rating matrix.
- Neighborhood Model+Xmeta: For neighbor selection strategy defined by $N(\theta, v)$, we only use director, genre and leading actors for computing. This method does not include any visual features.
- MF-: Different from MF+, MF- estimates b_v without using visual features. That is $b_{uv} = \mu + b_u + b_v$, the rest of MF- model stay the same with MF+.

Our method outperforms other models. Furthermore, we performed extensive experiments across poster visual features and still frames features, also choosing different combinations. As shown in Table 3, even MF- outperforms other methods that only use poster features without consideration of agreement effects during training. Again, this indicates that the features convey in posters should be considered being consistent with the still frames. Without surprises, the performance has been improved with better CNN features due to their additional generalization abilities. Furthermore, MF+ outperform MF-, this indicates that involving linear combination of features into estimating bias is a significant advantage.

Comparison on Netflix

The main advantage of our method is that it allows us to do exploration in the visual data, such as the posters and the still frames. To this end, we compare performance on the Netflix dataset, which consists of 675,236 movie still frames, 9138 posters of 6000 movies. Table 4 details the results. All results reported in the baselines use the same visual features as side information whenever needed, except for Neighborhood Model+Xmeta. As we can see, our method MF+ is able to produce better predictions compared all previously-reported results. We qualitatively observe that recommendation performance clearly benefits from the visual features extracted from movie posters and still frames. On the other hand, balancing the improvements from CNN features to other features, we observe that mid-level CNN features are more likely to outperform. It appears that CNN features from mid-level are more generic, and those from final level are more of task-specific.

6 Conclusions

In this work, we propose a novel movie recommendation framework, which allows to include visual features in helping the recommendation tasks. We studied the visual information in both the posters and still frames of the movies. Naturally, the visual information in the still frames is a good measure of the similarities between movies. Meanwhile, the agreement between features expressed in posters and still frames can be used to further development of performance. Also, we find that using a linear combination of visual features is capable of learning bias more accurately.

For the future work, we hope to incorporate broader features for visual data to obtain more powerful and robust performance, like trailers and plot descriptions. Besides, we also plan to investigate more advanced model that is flexible to unify additional features of various kind and rating data together.

Acknowledgments

We thank the support of China National 973 project 2014CB340304 and Hong Kong CERG projects 16211214 and 16209715. Sinno Jialin Pan is supported by the NTU Singapore Nanyang Assistant Professorship (NAP) grant M4081532.020.

References

[Davidson *et al.*, 2010] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.

[Donahue *et al.*, 2013] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[Kelly and Teevan, 2003] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, pages 18–28. ACM, 2003.

[Koenigstein *et al.*, 2011] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 165–172, New York, NY, USA, 2011. ACM.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434. ACM, 2008.

[Levi *et al.*, 2012] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 115–122, New York, NY, USA, 2012. ACM.

[Li and Yeung, 2009] Wu-Jun Li and Dit-Yan Yeung. Relation regularized matrix factorization. In *IJCAI*, pages 1126–1131, 2009.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[Lu *et al.*, 2013] Zhongqi Lu, Weike Pan, Evan Wei Xiang, Qiang Yang, Lili Zhao, and ErHeng Zhong. Selective transfer learning for cross domain recommendation. In *SDM*, pages 641–649. SIAM, 2013.

[Lu *et al.*, 2015] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. Content-based collaborative filtering for news topic recommendation. In *AAAI*, pages 217–223, 2015.

[Ma *et al.*, 2008] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 931–940, New York, NY, USA, 2008. ACM.

[Ma *et al.*, 2011] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.*, 29(2):9:1–9:23, April 2011.

[Marlin, 2003] Benjamin M Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*, 2003.

[Moshfeghi *et al.*, 2011] Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M. Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 625–634, New York, NY, USA, 2011. ACM.

[Oard *et al.*, 1998] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, pages 81–83. Wollongong, 1998.

[Paterek, 2007] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*, 2007.

[Pechyony and Vapnik, 2010] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. In *Advances in neural information processing systems*, pages 1894–1902, 2010.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[Rennie and Srebro, 2005] Jason D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, pages 713–719, 2005.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.

[Sen *et al.*, 2006] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. Tagging, communities, vocabulary, evolution. CSCW '06, pages 181–190, New York, NY, USA, 2006. ACM.

[Singh and Gordon, 2008] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *KDD*, pages 650–658. ACM, 2008.

[Vapnik and Vashist, 2009] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

[Zhao *et al.*, 2013] Lili Zhao, Sinno Jialin Pan, Evan Wei Xiang, Erheng Zhong, Zhongqi Lu, and Qiang Yang. Active transfer learning for cross-system recommendation. In *AAAI*. Citeseer, 2013.