SAR: A Semantic Analysis Approach for Recommendation

Han Xiao*, **Minlie Huang***, **Xiaoyan Zhu**
State Key Lab. of Intelligent Technology and Systems,
National Lab. for Information Science and Technology,
Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
bookman@vip.163.com; {aihuang,zxy-dcs}@tsinghua.edu.cn
*Corresponding Authors: http://www.ibookman.net, http://www.aihuang.org

Abstract

Recommendation system is a common demand in daily life and matrix completion is a widely adopted technique for this task. However, most matrix completion methods lack semantic interpretation and usually result in weak-semantic recommendations. To this end, this paper proposes a **S**emantic **A**nalysis approach for **R**ecommendation systems **(SAR)**, which applies a two-level hierarchical generative process that assigns semantic properties and categories for user and item. SAR learns semantic representations of users/items merely from user ratings on items, which offers a new path to recommendation by semantic matching with the learned representations. Extensive experiments demonstrate SAR outperforms other state-of-the-art baselines substantially.

1 Introduction

Recommendation systems are a common need in daily life and can be seen in many applications such as information retrieval, question answering, sentiment analysis, and more. Rating matrix is a simple, common mathematical formulation for recommendation systems, whose entry $M_{u,i}$ indicates the rating of item i made by user u. However, there are many unknown entries in this matrix and the goal of recommendation is to predict the missing values with various techniques. To this end, much efforts have been attempted, such as low-rank based methods [Lee *et al.*, 2016], latent space models [Song and Zhu, 2015], collaborative filtering [Rao *et al.*, 2015], norm constrained methods [Cai and Zhou, 2013], deep learning [Zhang and Chang, 2006], social integration [Liu *et al.*, 2016], etc.

Among these models, matrix completion methods, namely matrix factorization, is one of the most popular branches. Specifically, this branch treats the rating matrix with various assumptions such as low-rank. Under some specific assumption, an optimization problem is formulated to complete the rating matrix by supplementing missing values in it. However, due to the lack of underlying semantics, the accuracy of this branch is unsatisfactory in real-world applications. To incorporate more external resources, social information such as relationship between users can be integrated into matrix completion [Liu *et al.*, 2016]. However, social information is also noisy and only contributes limited improvement on the performance of recommendation.

In this paper, we address a novel problem in recommendation: *will recommendation be improved by representing users and items with richer latent semantics?* Latent semantics is referred to ***the properties or attributes for a user or an item, which are not directly observable***. For instance, a user can be semantically described as *(Age:Middle, Gender:Male, Occupation:Engineer, Action Film Fan:No...)*, while an item can also be represented in the same way, namely *(Age Rage:Adult, Favored by Male:Yes, Welcomed by Engineer:Yes, Action Film:Yes, ...)*. Under this assumption, semantic matching between a user and an item would be more effective than traditional matrix completion algorithms.

In comparison to purely real-valued vectorial representations, this semantic representation indicates the property of a user/item in a semantic form, where recommendation can be more easily approached as a ***semantic matching*** issue. In other words, if we can simultaneously represent user and item in the same semantic space, the rating inference is simply to compute similarity between the corresponding properties of a user and those of an item. As shown in Figure 1, the semantic properties of the user *Gentleman* are consistent with those of the movie *King Kong*, which makes a "to recommend" rating.

Therefore, we propose a two-level hierarchical generative model ***SAR*** to discover and leverage the property semantics merely from the rating matrix. At the first level of our model, we generate some properties such as *Action Film, Gender, etc*. At the second level of our model, we assign a corresponding category to each property for users and items. Taking the example of movie *"Quick and the Dead"*, we assign *Yes* in the *Action Film* property, *Youth* in the *Loved by Youth or Elder* property and so on. In this manner, users/items are semantically organized in a multi-view clustering form as shown in Figure 1, which is a novel unsupervised paradigm. It is worth noting that, the semantics are learned in a latent form, the observable words of which are addressed as future work.

Contributions. This paper proposes a semantic analysis method for recommendation systems **(SAR)**, which applies a two-level hierarchical generative process that globally allocates semantic properties and then locally assign categories in each property for users and items. Experimental results on real-world datasets show that our model consistently outper-

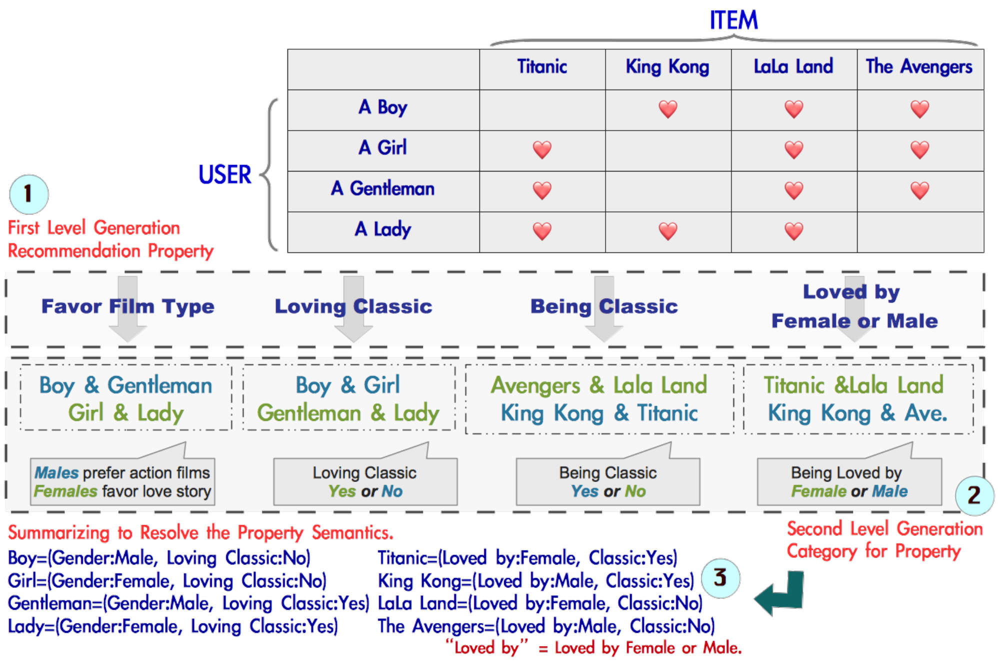

Figure 1: An example of the generative process. The users/items are semantically clustered from multiple views. Semantic properties are generated from the first-level generative process, while the category in each property, is generated from the second-level generative process. *"Being Classic"*, *"Gender"'*, etc. are semantic properties while *Male*, *Yes* etc. are categories for the corresponding properties.

forms the state-of-the-art baselines.

2 Related Work

Existing recommendation methods can be roughly classified into four categories: *matrix factorization, neighborhood based method, regression based method and social information integration method.* Notably, the first three methods are all based on matrix completion.

Matrix Factorization is a conventional paradigm for recommendation. This methodology firstly applies a factorization form on the rating matrix $M \approx UV$ to get the factorization matrices U,V and then multiplies $UV \approx \hat{M}$ to estimate the missing ratings, where $\hat{M}$ is the estimated rating matrix and U/V is the user/item-related latent factor matrix respectively. Since this branch addresses different assumptions on U and V, there methods fall into four primary subcategories according to the applied assumptions. *(1.) Basic matrix factorization* generally emphasizes latent factors as being nonnegative, such as NMF [Wang and Zhang, 2013], SVP [Meka *et al.*, 2009], MMMF [Rennie and Srebro, 2005], PMF [Mnih and Salakhutdinov, 2012]. *(2.) Combination of matrix factorization with neighborhood-based modeling*, such as CISMF [Guo *et al.*, 2015]. *(3.) Matrix factorization with complex rank assumptions* explores the rank properties and generalization ability to enhance the factorization, such as LLORMA [Lee *et al.*, 2016][Ko *et al.*, 2015], R1MP [Wang *et al.*, 2014b], SoftImpute [Rahul Mazumder, 2010], [Ganti *et al.*, 2015], [Zhang *et al.*, 2013], [Kirly *et al.*, 2012]. *(4.) Matrix factorization with discrete assumptions* treats the output of the rating matrix as discrete values to avoid noise and obtain more interpretations, such as ODMC [Huo *et al.*, 2016].

Neighborhood Based Method is one of the most classical approaches, assuming that the similar items/users hold similar rating preference. There are three main variants including item-based, user-based and global similarity, surveyed in [Guo *et al.*, 2015] and [Ricci *et al.*, 2011].

Regression Based Method is formulated as a regression problem, such as regression for graph GRALS [Cai *et al.*, 2011], blind regression [Song, 2016], Riemannian manifold based regression [Vandereycken, 2013], and others [Davenport *et al.*, 2014].

Social Information Integration leverages social information to enhance recommendation such as relationship between users, personalized profiles or movies' attributes. There list some latest researches: SR [Ma, 2013], PRMF [Liu *et al.*, 2016], geo-specific personalization [Liu *et al.*, 2014], social network based [Deng *et al.*, 2014] and other social context integration methods [Wang *et al.*, 2014a].

3 Methodology

3.1 Model Description

We apply a two-level hierarchical generative process to discover and leverage the property semantics as shown in Figure 2. All the parameters of $\mathcal{P}(z|u,f), \mathcal{P}(y|t,f), \mathcal{P}(\hat{p}|z,y,f)$ are learned by the training procedure. Notably, $\mathcal{P}(f), \mathcal{P}(u), \mathcal{P}(t)$ are uniformly distributed, indicating that they can be safely omitted with simple mathematical manipulation.

For each user-item pair $(u,t) \in \mathcal{D}_{train}$:
For each preference $p \in \{1,...,|R|\}$:

1. ***(First-Level)***
   Draw a property f_n from $\mathcal{P}(f_n|u,t)$:
   (a) ***(Second-Level)*** *Draw a user-specific category z_n from $\mathcal{P}(z_n|u,f_n)$.*
   (b) ***(Second-Level)*** *Draw an item-specific category y_n from $\mathcal{P}(y_n|t,f_n)$.*
   (c) ***(Second-Level)*** *Draw a preference strength $\hat{p}$ from $\mathcal{P}(\hat{p}|z_i,y_n,f_n,u,t)$.*
2. ***(Rating Generation)***
   Draw the rating r from a softmax distribution

$$\mathcal{P}(r|u,t) \propto e^{\hat{p}\omega_u\omega_t}$$

Figure 2: The generative process of SAR. $\mathcal{D}_{train}$ is the training set. $|R|$ is the maximum of ratings.

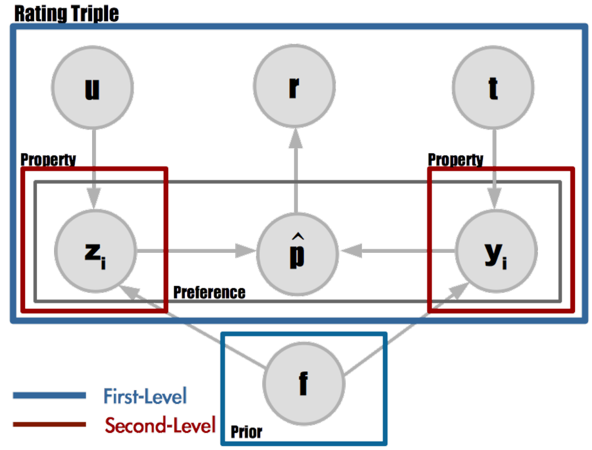

Figure 3: The probabilistic graph of SAR. All the symbols and plates are explained in Figure 2.

Globally, we generate the rating r for each user-item pair (u,t) from a softmax distribution (Gibbs distribution) where ω_u, ω_t are user/item-relevant parameters. We denote the sense of each entry in the softmax distribution as preference p whose range is the same as that of all possible ratings, that is, $\{1,...,|R|\}$.

Regarding the first-level, we generate a property randomly from the uniform distribution $\mathcal{P}(f_n|u,t)$, because we suppose each property is equally important. For example, we can hardly distinguish which one is more important between *Gender* and *Occupation*.

Regarding the second-level, we generate a user/item-specific category and a preference strength for each property. Obviously, the meaning of category depends on property, since different categories are semantically differentiated under different properties. For instance, the first category $(z=1)$ under the property of *Gender* is *Male or Female*, while that under the property of *Occupation* is *Student or Teacher*. Intuitively, the preference strength $\hat{p}$ depends on the user/item-specific category and the property, because the property semantics make an effect on the ratings.

Besides, the properties of user and item occurring in the same rating entry should be semantically close. Briefly speaking, the category distribution $\mathcal{P}(z_n|u,f_n)$ and $\mathcal{P}(y_n|t,f_n)$ are supposed to be consistent as required by the semantic matching demand. Taking Figure 1 as example, the *Lady* favors *"Titanic"*, which indicates she loves the *classic* movies while the *"Titanic"* is a classic one. Therefore, a Laplace prior is applied in $\mathcal{P}(\hat{p}|z_n,y_n,f_n)$ to formulate this observation as below:

$$\mathcal{P}(\hat{p}|z_n,y_n,f_n,u,t) = \tau_{\hat{p}|z_n,y_n,f_n} e^{-\frac{|\mathcal{P}(z_n|u,f_n)-\mathcal{P}(y_n|t,f_n)|}{\sigma}} \quad (1)$$

where $\tau_{\hat{p}|z_n,y_n,f_n}$ is tabular model parameters which can be tuned in the learning process and σ is the hyper-parameter.

Figure 3 presents the probabilistic graph form of SAR, with which we can compute the joint probability as $(2) \sim (4)$, where $|F|$ and $|C|$ is the number of properties and categories, respectively. It is noteworthy that, $\mathcal{P}(f_i|u,t)$ are uniformly distributed.

Figure 3 presents two inference tasks of SAR: rating prediction and user/item-specific category distribution computation. As to the rating prediction, it is estimated as the expectation of softmax distribution, as formulated in (5).

$$r_{|u,t} \doteq \mathbb{E}_{r|u,t}(r) = \frac{\sum_{p=1}^{|R|} p \times e^{\hat{p}\omega_u\omega_t}}{\sum_{p=1}^{|R|} e^{\hat{p}\omega_u\omega_t}} \quad (5)$$

where $\hat{p}$ is the corresponding preference strength. As to the category distribution, it is calculated respectively as (6) and (7), which are the direct results of (2) by applying the sum rule. Notice that all the $\mathcal{P}(f_i)$, $\mathcal{P}(u)$ and $\mathcal{P}(t)$ are uniform distributions.

$$\mathcal{P}(z_n|u) = \sum_{t,f_n,y_n,\hat{p}} \mathcal{P}(\hat{p},z_n,y_n,f_n|u,t) \quad (6)$$

$$\mathcal{P}(y_n|t) = \sum_{u,f_n,z_n,\hat{p}} \mathcal{P}(\hat{p},z_n,y_n,f_n|u,t) \quad (7)$$

3.2 Objective & Training

We formulate the objective function as the mean square error between the predicted and golden ratings. Furthermore, we also apply the regularization term in the objective function.

$$\mathcal{L} = \sum_{(u,r,t)\in\mathcal{D}_{train}} (r_{|u,t} - r)^2 + \lambda \left\{ \sum_{n=1}^{|F|} ||\mathcal{P}(z_n|u,f_n)||_2^2 + \sum_{n=1}^{|F|} ||\mathcal{P}(y_n|t,f_n)||_2^2 \right\} \quad (8)$$

where $\mathcal{D}_{train}$ is the training dataset and $r_{|u,t}$ is the predicted rating using (4). $||\ ||_2^2$ is the l_2 norm.

The model parameters are as follows: $\mathcal{P}(z_n|u,f_n)$,$\mathcal{P}(y_n|t,f_n)$, $\tau_{\hat{p}|z_n,y_n,f_n}$, ω_u, ω_t, all of which are learned by minimizing the objective function $\mathcal{L}$. For a more efficient and facilitating solution, a moment-based gradient method AdaDelta [Zeiler, 2012] is adopted with hyper-parameters: moment factor η and RMSE factor ϵ.

$$\mathcal{P}(\hat{p}, z_n, y_n, f_n|u,t) \quad = \mathcal{P}(z_n|u,f_n)\mathcal{P}(y_n|t,f_n)\mathcal{P}(\hat{p}|z_n,y_n,f_n,u,t) \tag{2}$$

$$\mathcal{P}(\hat{p}|u,t) \quad = \overbrace{\sum_{n=1}^{|F|} \underline{\mathcal{P}(f_n|u,t)} \overbrace{\sum_{i,j=1}^{|C|} \underline{\mathcal{P}(z_n=i|u,f_n)\mathcal{P}(y_n=j|t,f_n)}\mathcal{P}(\hat{p}|z_n=i,y_n=j,f_n,u,t)}^{Second-LevelMixture}}^{First-LevelMixture} \tag{3}$$

$$= \sum_{n=1}^{|F|} \mathcal{P}(f_n|u,t) \sum_{i,j=1}^{|C|} \mathcal{P}(z_n=i|u,f_n)\mathcal{P}(y_n=j|t,f_n)\tau_{\hat{p}|z_n,y_n,f_n} e^{-\frac{|\mathcal{P}(z_n=i|u,f_n)-\mathcal{P}(y_n=j|t,f_n)|}{\sigma}} \tag{4}$$

Regarding the efficiency, theoretically, the time complex of SAR is $\mathcal{O}(|F| \cdot |C|^2)$, while practically the running time is present in Table 3, which illustrates our method is indeed efficient.

3.3 Analysis from the Clustering Perspective

Regarding the mixture form of Equation (3) where the *underline* terms are the mixture factors, in the both first- and second-level generation, SAR adopts the idea of hierarchical mixture models, which can be further analyzed from the clustering perspective. On one hand, the second-level generative process clusters the users/items according to the corresponding property semantics. These semantic properties originate from the first-level process, according to all the probabilistic terms involved with f_n. On the other hand, the first-level generative process adaptively adjusts different property semantics with the information from the second-level. Mathematically, the feed-back information indicates $\mathcal{P}(z_{1..n}, y_{1..n}, f|u,r,t)$.

In essence, users/items are semantically differentiated in a multi-view clustering form. Hence, by characterizing the multi-view clustering nature, SAR can promote recommendation by semantic matching.

Let's revisit this process in Figure 1. There is a pool of users/items as input. The single-view clustering process (similar to K-MEANS) is ambiguous, because there are always many clustering angles, such as clustering by *Gender*, or by *Loving Classic*, etc. However, once the first-level process generates different semantic properties such as *Gender* and *Occupation*, clustering of users/items at the second-level could be addressed according to one exact property semantics. For the example depicted in Figure 1, *Boy* within the *Gender* property clustering angle belongs to the *Male* cluster rather than *Female*, while that in the *Loving Classic or Not* property belongs to the *No* cluster rather than *Yes*. Finally, summarizing each semantic property, SAR discovers the property semantics of the users/items, namely *Boy = (Gender:Male, Loving Classic:No, ...). The property semantics is a "latent" semantic information to empower recommendation systems.*

3.4 Analysis from the Semantic Perspective

Once SAR discovered the property semantics as shown in Figure 1, *the recommendation can be strengthened in the manner of semantic matching. It means it is sufficient to compare the semantic properties between users and items to infer the ratings.* For example, when predicting whether the *boy* would like *"Titanic"*, we just compare the corresponding properties of the user with those of the item and then no matching is found, so it is reasonable to predict a "not to recommend" rating. Conversely, when the girl rates *"LaLa Land"*, semantic matching would reasonably derive a "to recommend" rating. Mathematically, semantic matching is modeled by the Laplace prior of Equation (1).

4 Experiment

4.1 Datasets.

We conduct our experiments on two public benchmark datasets: MovieLens 100K (*ML100K* for short) and MovieLens 1M (*ML1M*). ML100K consists of 100,000 ratings given by 943 users to 1,682 movies, while 93.69% of entries in the rating matrix is empty, while ML1M contains 1,000,209 ratings given by 6,040 users to 3,706 movies, and 95.53% of entries are missing.

4.2 Performance Evaluation

The rating prediction is a traditional benchmark task, which concerns the predictive ability for matrix completion. This task directly benefits many recommendation applications, such as item-based recommendation [Zhang *et al.*, 2013].

Protocol. & Metric. We evaluate how SAR works on the datasets of different sizes and sparsity . Hence, we vary the ratio of *training set to test data* ρ to evaluate our algorithm and compare with baselines. For example if $\rho = 80\%$, 80% of the observed ratings are randomly sampled for training, and the remaining 20% observed ratings are used for test. Two widely used measures are employed for prediction evaluation, namely *Root Mean Square Error (RMSE)* and *Mean Absolute Error (MAE)*.

$$MAE = \frac{\sum_{r_i \in \mathcal{D}_{test}} |r_i - \hat{r}_i|}{N_{test}} \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{r_i \in \mathcal{D}_{test}} (r_i - \hat{r}_i)^2}{N_{test}}} \tag{10}$$

Above, $\mathcal{D}_{test}$ indicates the test set, where each r_i means an original rating and $\hat{r}_i$ is the predicted rating. N_{test} is the number of ratings in the test dataset.

Table 1: Rating prediction results on ML100K, measured by RMSE and MAE.

Dataset	Method	MAE	RMSE
ML100K $\rho = 80\%$	PMF	0.7522	0.9667
	NMF	0.7724	0.9874
	MMMF	-	0.9800
	User-Based	0.7370	0.9440
	CISMF	0.7279	0.9268
	ODMC	0.7033	0.9609
	GRALS	-	0.9450
	SR	0.7298	0.9218
	PRMF	0.7215	0.9135
	SAR(Ours)	***0.6772***	***0.9069***
ML100K $\rho = 50\%$	SVP	-	0.9683
	R1MP	-	1.0168
	SoftImpute	-	1.0354
	SAR(Ours)	***0.6954***	***0.9280***

Table 2: Rating prediction results on ML1M, measured by RMSE and MAE.

Dataset	Method	MAE	RMSE
ML1M $\rho = 80\%$	PMF	0.7306	0.9234
	NMF	0.7286	0.9203
	MMMF	-	0.8810
	User-Based	0.7030	0.9050
	LLORMA	0.6941	0.8911
	ODMC	0.6583	0.9371
	SAR(Ours)	***0.6436***	***0.8715***
ML1M $\rho = 50\%$	SVP	-	0.9085
	R1MP	-	0.9595
	SoftImpute	-	0.8989
	Blind Regression	-	0.9220
	SAR(Ours)	***0.6543***	***0.8827***

Baselines & Implementation. We compare SAR with 14 state-of-the-art baselines, almost including all the major approaches introduced in Related Work. Specially, SR is a method encodes the social information into matrix completion, and NMF constrained all the factored matrix positive. *For a fair comparison with previously proposed methods, we directly reprint the results under the same setting from the literature.* On both datasets, the optimal configuration of SAR are as follows: property number $|F| = 10$, category number $|C| = 10$, Laplace prior hyper-parameter $\sigma = 1.0$, regularization factor $\lambda = 0.05$, moment factor $\eta = 0.6$ and $\epsilon = 1 \times 10^{-6}$.

Prediction Accuracy. Evaluation results are shown in Table 1 and Table 2. We can make the following statements.

1. SAR outperforms all the baselines , justifying the effectiveness of our model. Theoretically, the effectiveness stems from the semantics-specific modeling of SAR.
2. The results on ML100K ($\rho = 80\%$) show that, SAR outperforms SR by 0.081 vs. NMF by 0.066 in the metric of RMSE, justifying that SAR can discover and utilize more semantics than social information integration methods do.

Table 3: Average computational time for one round (measured in seconds). The experiments are conducted on i7-6900K (3.87 GHz) with 32GB Memory.

Dataset	NMF	PMF	MMMF	SAR(Ours)
ML100K	0.9	4.3	121.2	5.5
ML1M	5.5	16.8	395.5	38.1

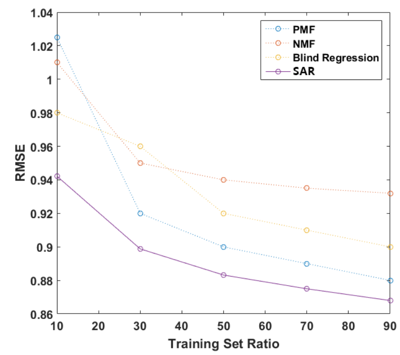

Figure 5: RMSE of PMF, NMF, Blind Regression on the ML1M dataset when varying the training set ratio ρ.

Time Complexity. Computation efficiency is evaluated as shown in Table 3, from which we can observe that SAR is as efficient as NMF. As to the trade-off between efficiency and effectiveness, we propose the metric of RMSE improvement over NMF per second: $-\frac{RMSE(algorithm)-RMSE(NMF)}{Time(algorithm)-Time(NMF)}$ *(the bigger, the better)*. In terms of this measurement, the result of PMF is 6.00×10^{-3} and that of SAR is 1.60×10^{-2} on ML100K, while 2.74×10^{-4}, 1.50×10^{-3} on ML1M for PMF and SAR, respectively. The comparison illustrates that SAR is a better time-performance trade-off method, which is more practical for real-world applications.

Sparsity. We change the ρ to verify the effect of sparsity and the results are shown in Figure 5. We find that all the methods improve as the training set size increases while SAR consistently outperforms the other baselines. For the case of blind regression, where ρ varies from 90% to 30%, the RMSE degrades 0.050, while for SAR, the result is 0.032. This comparison demonstrates that SAR is less sensitive to data sparsity than blind regression which is a strong baseline from this aspect.

Hyper-Parameters. As illustrated in Figure 6, under the normal settings, SAR is insensitive to hyper-parameters. Concretely, when the category number $|C|$ varies between 6 and 12, the performance (RMSE) changes within a magnitude of 0.015, and when the property number $|F|$ changes between 6 and 12, the performance (RMSE) also varies within a small magnitude of 0.1. This comparison justifies the robustness of SAR.

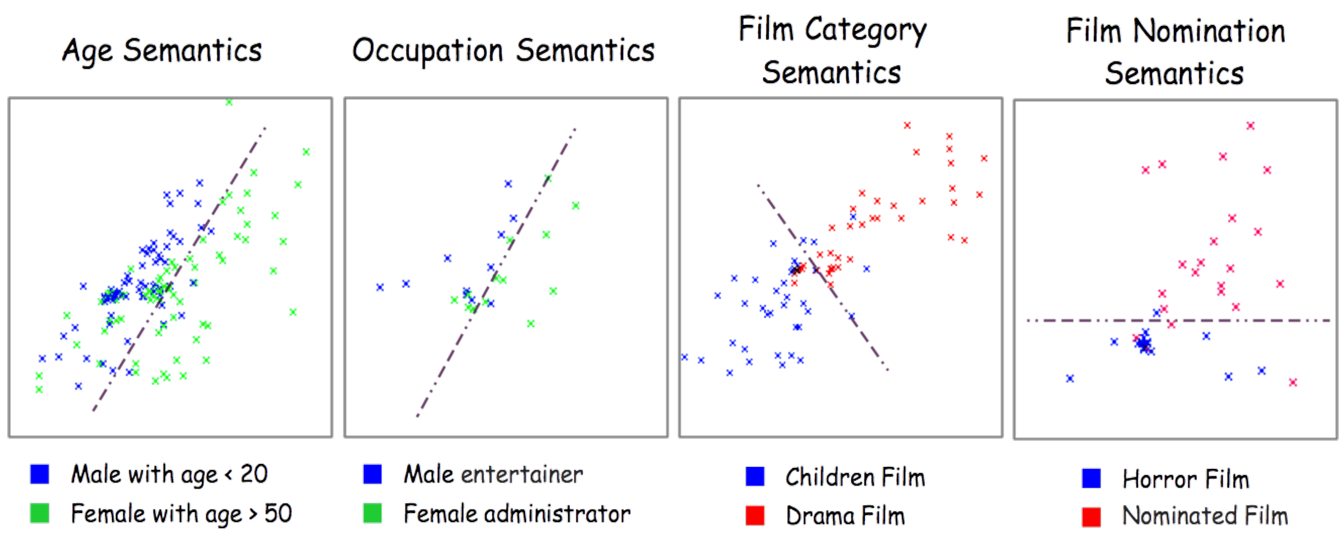

Figure 4: The demonstration of semantic evaluation. Each point indicates a distribution (namely $\mathcal{P}(z_n|u)$ or $\mathcal{P}(y_n|t)$), which is plotted after projected by PCA. The line is the boundary between two different categories. The top text indicates the property and the bottom legend represents the categories. The left two figures are user-specific and the right two are item-specific.

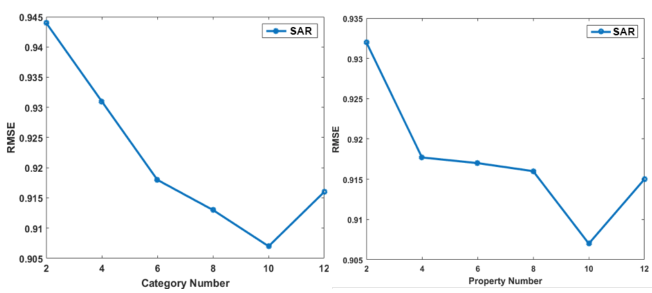

Figure 6: The effect of hyper-parameters. The y-axis is the RMSE on ML100K with $\rho = 80\%$. The x-axis means varying the corresponding hyper-parameters with the optimal configuration of other settings as introduced in **Baselines & Implementation.**

4.3 Semantic Evaluation

In this subsection, we conduct an experiment to testify our clarification about property semantics. GroupLens is chosen since it provides the user/item profiles (e.g. gender, occupation, film type) for ML100K, which can facilitate the semantic analysis of our model.

Experimental Setting. The empirical results are shown in Figure 4, which is proceeded with the following steps. First, we calculate each distribution according to (6) and (7) for each user/item and then represent a user/item with the corresponding distribution (namely $\mathcal{P}(z_i|u)$ or $\mathcal{P}(y_j|t)$) as a point in high-dimensional space $\mathbb{R}^{|C|}$. Second, PCA is conducted to project $|C|$-dimensional points to a 2D plane. At last, the corresponding attributes from the profile data are manually analyzed to color the points. For clarity, a line is painted artificially to illustrate the boundary.

Results. We can easily see that SAR discovers and leverages the property semantics appropriately, as shown in Figure 4.

Concretely, regarding the left figures, user-specific semantics are identified as a combination of gender and age attributes. In fact, only one single user attribute can hardly distinguish a preference (i.e., a rating), but a combination of user attributes is capable to semantically identify the preference. For example in the most left sub-figure, the blue points indicate the teenager boys and the green points denote the housewives or senior business women. Since the properties in terms of recommendation are different, the two point groups are discriminated to a large extent.

Additionally, we found the item-relevant semantics are visualized as a transition between attributes components in SAR. In the sub-figure of *Children - Drama Film*, the proximity of the boundary lays some *Children Drama* films, such as *"Little Prince(1995)"* and *"Secret Garden(1993)"*. But conversely in the rightest sub-figure, there are almost no films in the transition area, because in the ML100K dataset, there is not a nominated horror film at all. This phenomenon illustrates the semantic effectiveness of SAR, which can benefit the recommendation by semantic matching.

5 Conclusion

In order to discover and utilize latent property semantics, we propose a semantic analysis approach for recommendation, **(SAR)**. The model applies a two-level hierarchical generative process that assigns semantic properties and categories for user and item. Experimental results on benchmark datasets demonstrate the effectiveness of our proposed methods .

References

[Cai and Zhou, 2013] Tony Cai and Wen Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *Journal of Machine Learning Research*, 14(10):3619–3647, 2013.

[Cai *et al.*, 2011] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence IEEE Transactions on*, 33(8):1548–1560, 2011.

[Davenport *et al.*, 2014] Mark A. Davenport, Yaniv Plan, Ewout Van Den Berg, and Mary Wootters. 1-bit matrix completion. *Statistics*, 3(3), 2014.

[Deng *et al.*, 2014] Shuiguang Deng, Longtao Huang, and Guandong Xu. Social network-based service recommendation with trust enhancement. *Expert Systems with Applications*, 41(18):8075–8084, 2014.

[Ganti *et al.*, 2015] Ravi Ganti, Laura Balzano, and Rebecca Willett. Matrix completion under monotonic single index models. 2015.

[Guo *et al.*, 2015] Meng Jiao Guo, Jin Guang Sun, and Xiang Fu Meng. A neighborhood-based matrix factorization technique for recommendation. *Annals of Data Science*, 2(3):1–16, 2015.

[Huo *et al.*, 2016] Zhouyuan Huo, Ji Liu, and Heng Huang. Optimal discrete matrix completion. 2016.

[Kirly *et al.*, 2012] Franz J. Kirly, Louis Theran, and Ryota Tomioka. The algebraic combinatorial approach for low-rank matrix completion. *Journal of Machine Learning Research*, 62(2):299–321, 2012.

[Ko *et al.*, 2015] Han Gyu Ko, Joo Sik Son, and In Young Ko. Multi-aspect collaborative filtering based on linked data for personalized recommendation. In *The International Conference on World Wide Web*, pages 49–50, 2015.

[Lee *et al.*, 2016] J. Lee, S. Kim, G. Lebanon, Y. Singer, and S. Bengio. Llorma: Local low-rank matrix approximation. 2016.

[Liu *et al.*, 2014] Jing Liu, Zechao Li, Jinhui Tang, and Yu Jiang. Personalized geo-specific tag recommendation for photos on social websites. *IEEE Transactions on Multimedia*, 16(3):588–600, 2014.

[Liu *et al.*, 2016] Yong Liu, Peilin Zhao, Xin Liu, Min Wu, and Xiao Li Li. Learning optimal social dependency for recommendation. 2016.

[Ma, 2013] Hao Ma. An experimental study on implicit social recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 73–82, 2013.

[Meka *et al.*, 2009] Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. Guaranteed rank minimization via singular value projection. *Nips*, pages 937–945, 2009.

[Mnih and Salakhutdinov, 2012] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *International Conference on Machine Learning*, pages 880–887, 2012.

[Rahul Mazumder, 2010] Robert Tibshirani Rahul Mazumder, Trevor Hastie. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(11):2287–2322, 2010.

[Rao *et al.*, 2015] Nikhil Rao, Hsiang Fu Yu, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. 2015.

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

[Ricci *et al.*, 2011] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor. *Recommender systems handbook /.* Springer,, 2011.

[Song and Zhu, 2015] Yang Song and Jun Zhu. Bayesian matrix completion via adaptive relaxed spectral regularization. *Computer Science*, 61(22):5689–5703, 2015.

[Song, 2016] Dogyoon Song. Blind regression : nonparametric regression for latent variable models via collaborative filtering. 2016.

[Vandereycken, 2013] Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *Siam Journal on Optimization*, 23(23):10.1137/110845768, 2013.

[Wang and Zhang, 2013] Yu Xiong Wang and Yu Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013.

[Wang *et al.*, 2014a] Fei Wang, Meng Jiang, Wenwu Zhu, Shiqiang Yang, and Peng Cui. Recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2789–2802, 2014.

[Wang *et al.*, 2014b] Z. Wang, M. J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye. Rank-one matrix pursuit for matrix completion. pages 91–99, 2014.

[Zeiler, 2012] Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *Computer Science*, 2012.

[Zhang and Chang, 2006] Feng Zhang and Huiyou Chang. Employing bp neural networks to alleviate the sparsity issue in collaborative filtering recommendation algorithms. *Journal of Computer Research and Development*, 43(4):667–672, 2006.

[Zhang *et al.*, 2013] Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma, and Shi Feng. Localized matrix factorization for recommendation based on matrix block diagonal forms. 2013.