Review of Collaborative Filtering in Recommender System

Final Project Report of CS 6241

Wentao Guo

Abstract

Keyword

SVD, SVD++, Recommender System, Collaborative Filtering

1 Background

Developed in 1990s, the recommender system (RS) has been applied in e-commence, music apps, job portal, social networking and more [2]. The recommender system collects information on users' behaviors explicitly (users' ratings on items) and implicitly (users' mouse movements, attention on one page, etc.) and makes prediction on users' preferences. For example, Netflix applies a five-star ratings to help users find their favorite movies and maintain their subscriptions [5], and Amazon selects products based on the predicted users' favors to gain profits.

In October 2006, Netflix launched a competition for which they rewarded the team that can beat Netflix's Cinematch system by at least 10% Root Mean Squared Error [5]. This competition arouse great attention in collaborative filtering field as the dataset covered 100 million ratings for 500, 000 anonymous customers on 17, 000 movies, which was greater than previous public dataset in the orders of magnitude [17]. The final grand prize was given to "Bellkor's Pragmatic Chaos" team in 2009 [5] [15].

In this competition, matrix factorization approaches raised people's attention as the top teams in this competition frequently applying it to beat Bayes or probabilistic approaches [14]. Singular Value Decomposition as the classical latent semantic indexing approach in information retrieval was found as a great fit for spanning customers data with low dimensions and often adapted to various variants as regularized SVD, SVD++, iterative SVD, and more [5] [11] [9] [14].

2 SVD-Based CF Methods

2.1 Intro to SVD

SVD, as a common dimension reduction tool, originally used in information retrieval to identify latent semantic factor (as Latent Semantic Indexing), can represent the user-item interactions in a low dimensional space [17]

[25]. SVD approximates a matrix $A \in \mathbb{R}^{m \times n}$ in a form as

$$A = U \Sigma V^T \tag{1}$$

Suppose the rank of A is r, there are three main properties in SVD[25] [5]:

- The initial r singular values of diagonal matrix Σ holds that $\forall i \in [1, r]\, \sigma_i > 0$ and $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$.
- The first r columns of orthogonal matrix U are eigenvectors of $A\,A^T$ and span the column space of A.
- The first r columns of orthogonal matrix V are eigenvectors of $A^T\,A$ and span the row space of A.

People usually takes the best rank-k approximation of matrix A as

$$A_k = U_k \Sigma_k V_k^T \tag{2}$$

with $U_k \in \mathbb{R}^{m \times k},\ \Sigma_k \in \mathbb{R}^{k \times k},\ V_k \in \mathbb{R}^{k \times n}$.

2.2 Standard SVD-Based Neighborhood Approach of Collaborative Filtering

We will take a rating matrix R with users $u_1, u_2, ...u_i$ in the row and items $i_1, i_2, ...i_j$ in the column, and the entry R_{ij} is the estimated interest of user u_i on item i_j. [17] [13] [25].

Notice that matrix R is often sparse and require some prepossessing. A common prepossessing approach is to take the row averages to replace all of the missing values in the matrix R and normalize R by subtracting row averages [25] [13].

We then compute $U_k \Sigma_k V_k^T$ as equation 2, and form two matrix products: $U_k \sqrt{\Sigma_k^T}$ and $\sqrt{\Sigma_k} V_k^T$.

There are some variants regarding the computation of predicted rating [25] [24]. For example, Manolis et al. [24] provided a formula for SVD-CF prediction rating for user u on item i as:

$$\hat{r}_{ui} = \bar{r_u} + U_k \sqrt{\Sigma_k^T [u]} \sqrt{\Sigma_k} V_k^T (i) \tag{3}$$

Another variant is based on PCA, with the previous steps identical except that we need to replace and subtract column average $\bar{c_j}$ for prepossessing R. In addition, prediction score for user u on item i will be

$$\hat{r}_{ui} = \bar{r_i} + U_k \sqrt{\Sigma_k^T [u]} \sqrt{\Sigma_k} V_k^T (i) \tag{4}$$

Some researchers incorporate the reduced matrix $R_{\text{red}} = U_k \Sigma_k V_k^T$ into the calculation of similarity measure to predict the new user or item's score based on the existing clusters. These approaches are usually categorized to user-based collaborative filtering or item-based collaborative filtering depending on the subject of similarity measure.

Pearson correlation and adjusted cosine similarity are common similarity measures in literature[22][25] [18] [1] [3]:

- Pearson correlation

  $\mathcal{I}_{uv}$ means common item rated by both user u and user v. $\bar{r_u}$ means average rating of user u on the shared item set.

$$\text{sim} = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r_u})^2 (r_{vi} - \bar{r_v})^2}} \tag{5}$$

- Adjusted cosine similarity

  $\bar{r_u}$ means average rating of user u on the entire item set.

$$\text{sim} = \frac{\sum_i (r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_i (r_{ui} - \bar{r_u})^2 (r_{vi} - \bar{r_v})^2}} \tag{6}$$

Pearson-correlation looks really similar to adjusted cosine similarity except that adjusted cosine similarity is calculated over the entire set of rated vectors while person correlation is calculated over co-rated vector [7]. For the missing value case, a typical approach is to set it as zero [18].

In item-based collaborative filtering, after we calculate similarity measure, we normally select nearest l neighbors and then make the prediction (or compute prediction rating). The first algorithm proposed by Vozalis and Margaritis [25] provides a formula as

$$\hat{r_{ui}} = \frac{\sum_{j=1}^{l} \text{sim}_{ij} * (R_{\text{red}_{uj}} + \bar{r_u})}{\sum_{j=1}^{l} |\text{sim}_{ij}|} \tag{7}$$

Variants on prediction rating formula exist, but this is one classical approach of item-based collaborative filtering. The process of user-based collaborative filtering is similar to item-based collaborative filtering except the implementation of SVD on R and prediction rating is different.

If we formulate a framework neighborhood-based collaborative filtering (including user-based and item-based) based on previous steps, we will obtain what Bokde et al. purpose [4][1]:

1. Prepossessing: build a user-item matrix and perform dimensional reduction
2. Similarity Evaluation: formulate the neighborhood with similarity metrics
3. Prediction Rating Process & Recommendation: calculate the correlation and finally provide a recommendation

2.3 Standard SVD-Based Latent Factor Approach of Collaborative Filtering

The previous framework more applies to neighborhood approach. Some researchers are also interested in the framework of "latent semantic model".

2.3.1 User-item Interaction in Latent Factor Space

Matrix factorization models represents the user-item interaction in a latent joint space [17]. Each item i is associated with a vector $q_i \in \mathbb{R}^f$ and user u is associated with a vector $p_u \in \mathbb{R}^f$. The rationale behind such representation is: item i will possess some unknown factors in a measure, positive or negative, stored in entries of q_i, and user will have different favors toward these factors, and such interest is represented as entries in p_u. Therefore, we can take inner product to get an approximate of the user u's rating on item i as [17]

$$\hat{r}_{ui} = q_i^T p_u \tag{8}$$

Although latent factor model in matrix factorization method is not directly involved with SVD (as it does not explicitly computes $U\Sigma V$), researchers often dub the name "SVD" since 1) SVD generates the minimum of

[1] The prediction rating process can be combined with or separated from recommendation in real practices.

$R - \hat{R}$ in Frobenius norm with $\hat{R}$ rank k and the objective of latent factor model is often $\min R - PQ$ and 2) the idea of matrix factorization is to represent R in a product of low rank matrix of latent factors as P (user matrix) and Q (item matrix), which is similar to the idea of SVD on decomposing a large matrix into low rank matrices.

The latent factor model tries to minimize the prediction error in some metrics. Conventional evaluation metrics includes root mean squared error (RMSE) and mean absolute error (MAE). Researchers usually minimize an objective function $\mathcal{L}$ to obtain prediction. (Stochastic) gradient descent will be applied if the problem is convex, or if in the case of biconvex alternating least square (ALS) will be applied such as what Zheng et al. show [26].

Conventional SVD suffers from an incomplete (or even sparse) rating matrix. Researchers often resort to data imputation to fill in the matrix, or through regularized least squared model as [17].

$$\min_{q,p} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{9}$$

in which κ is the set of known ratings. The regularizing term $\|q_i\|^2 + \|p_u\|^2$ controls the magnitude of q_i and p_u to avoid overfitting.

2.3.2 Adding Bias to Latent Factor Model

Early recommender system exhibit a systematic bias for users giving generally higher ratings than others or items receive higher ratings than others [17]. Therefore, as Koren, et al. suggested, the entire user-item interaction cannot be simply explained by $q_i^T p_u$, and a bias term exists. A first-order approximation of bias is as follows [17]

$$b_{ui} = \mu + b_i + b_u \tag{10}$$

in which b_{ui} is the bias and μ is the overall average rating of training data, and b_u, b_i are user u and item i's average ratings compared to the overall average rating [17]. We can transform our estimated rating as

$$\hat{r}_{ui} = b_{ui} + q_i^T p_u \tag{11}$$

and the objective becomes

$$\min_{q,p} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u - b_{ui})^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2) \tag{12}$$

Jallouli et al. develop a framework for latent factor model [8] as follows

1. define decomposition model for rating matrix R

   - $R \leftarrow P * Q$

     P represents preference of users in some latent factors and Q shows items attributes scores in the same factor space.

   - $R \leftarrow P * Q * C$

     P Q remains the same and C is the matrix for context.

2. define decision function

   $\hat{r}_{ui}$ is represented by a function of q_i^T, p_u and other factors such as social contexts, bias, regularization terms, temporal dynamics and more.

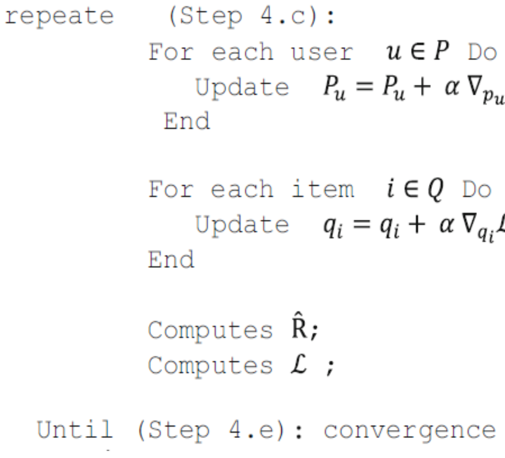
3. define objective function

   The objective $\mathcal{L}$ measures the error difference between $\hat{R}$ and R, and generally convex on variables p and q. Upon defining $\mathcal{L}$, a numerical optimization process shall also be determined.

4. define model learning

   (Stochastic) Gradient descent generally applies here as follow.

Figure 1: Generic Algorithm for Latent Factor Approach[8]



```
repeate   (Step 4.c):
          For each user  $u \in P$ Do
             Update  $P_u = P_u + \alpha \nabla_{p_u}\mathcal{L}$
           End

          For each item  $i \in Q$ Do
             Update  $q_i = q_i + \alpha \nabla_{q_i}\mathcal{L}$
          End

          Computes $\hat{R}$;
          Computes $\mathcal{L}$ ;

  Until (Step 4.e): convergence
```

Koren et al. [17] suggest that, though common in practices, stochastic gradient descent can be unfavorable compare to alternate least squares in two cases. The first case is when the system use parallelization for computing q_i and p_u independently with other item/user factors. The second case is when the system is centered on dense implicit data, which will turn looping as gradient descent does impractical.

2.4 Regularized SVD Model

2.4.1 Standard L1/L2 Regularization

Regularized SVD model begins with Simon Funk in the Netflix challenge [17]. This is essentially a ridge regularization and the main body of looping update part is shown in Algorithm 1

Algorithm 1 Simon Funk's Update Body with Ridge (L2) Regularization[17]

```
$e_{ui} = r_{ui} - q_i^T p_u$
$q_i \mathrel{+}= \gamma(e_{ui} * p_u - \lambda * q_i)$
$p_u \mathrel{+}= \gamma(e_{ui} * q_i - \lambda * p_u)$
```

The original purpose of regularization terms is to avoid overfitting and common approach includes lasso and ridge (or L1 L2) regularization [10], as shown in Algorithm 3.

An ALS implementation of SVD with ridge (L2) regularization is provided by Zheng et al. [26].

Algorithm 2 Ji et al. Looping Update Body with Lasso (L1) Regularization[10]

```
if $p_u \geq 0$ then
    $p_u \mathrel{+}= \gamma(e_{ui}q_j - \lambda\alpha - \lambda p_u(1-\alpha))$
else
    $p_u \mathrel{+}= \gamma(e_{ui}q_j + \lambda\alpha - \lambda p_u(1-\alpha))$
end if
if $q_i \geq 0$ then
    $q_i \mathrel{+}= \gamma(e_{ui}p_i - \lambda\alpha - \lambda q_i(1-\alpha))$
else
    $q_i \mathrel{+}= \gamma(e_{ui}p_i + \lambda\alpha - \lambda p_u(1-\alpha))$
end if
```

Algorithm 3 Regularized ALS Suggested by Zheng et al.[26]

```
Input: rating matrix $R$, rank $k$, regularization weight $\lambda$
  $L = \|R - PQ^T\|_F^2 + \lambda\|P\|_F^2 + \lambda\|Q\|_F^2$
  initialize $Q$ with some random values
  repeat
      $P = RQ(Q^TQ + \lambda I)^{-1}$
      $Q = R^TP(P^TP + \lambda I)^{-1}$
  until $L$ converges
```

Zheng et al. [26] have proven that RSVD is still in SVD subspace and it converges generally faster than standard SVD approach. The RSVD objective function is not convex (RSVD is actually biconvex), but it still has a global optimal solution that can be obtained as the global optimal solution as a closed form [26].

2.4.2 Weighted-λ-Regularization

Besides L1 and L2 regularization techniques, scholars also have interests on weighted-λ-regularization, as proposed by Zhou et al. [27] and the regularization term is shown in equation 13. The rationale behind such regularization is to penalize items or users that have more ratings and to give a chance for new users or items, as a potential solution to cold-start issue of collaborative filtering. Notice that I_u means the set of all items user u has rated and U_i is the set of all users that rate item i.

$$\lambda(\sum_u |I_u| \|p_u\|^2 + \sum_i |U_i| \|q_i\|^2) \tag{13}$$

Weighted-λ-regularization is then modified by Guo et al. [6] to less penalized popular items/users and more regularize cold-start users and niche items as $\|I_u\|^{-.5}$ and $\|U_i\|^{-.5}$ as shown in equation 14 and such idea is further used in extended SVD++ model by Jallouli et al. [9].

$$\lambda(\sum_u \|I_u\|^{-.5} \|p_u\|^2 + \sum_i \|U_i\|^{-.5} \|q_i\|^2) \tag{14}$$

2.5 SVD++ Model

The traditional SVD method only considers users' explicit ratings but do not take implicit data into account. Although implicit data (mouse move, reading time, etc.) might not be publicly available, there is one kind of implicit data always available: whether a user rates on an item (rate vs. not rate).

SVD++ model was developed by Koren [14]. The complete formula is shown in equation SVD++ Model in Three-Tier [14].

$$\begin{aligned}\hat{r}_{ui} = \mu &+ b_i + b_u \\ &+ q_i^T(p_u + |N_u|^{-1/2}\sum_{j \in N_u} y_j) \\ &+ |R^k(i;u)|^{-1/2}\sum_{j \in R^k(i;u)}(r_{u_j} - b_{u_j})w_{ij} + |N^k(i;u)|^{-1/2}\sum_{j \in N^k(i;u)} c_{ij}\end{aligned} \qquad \text{SVD++ Model in Three-Tier [14]}$$

Rule SVD++ Model in Three-Tier [14] is a 3-tier model. The first tier is b_{ui} as mentioned before.

The second tier $q_i^T(p_u + |N_{(u)}|^{-\frac{1}{2}}\sum_{j \in N_{(u)}} y_j)$ describes the intersection between user profile and item profile. We need to introduce a few notations to understand this tier:

- R_u is the set of all items for which user u provides a rating. N_u is the set of all items for which user u provides an implicit preference. $S^k(i)$ is the set of k items rated by u that are similar to item i.
- $N^k(i;u) = N_u \cap S^k(i)$. $R^k(i;u) = R_u \cap S^k(i)$
- w_{ij} is a weight from item j to i. c_{ij} is a significance offset (indication of predictability by item j to i)

The remaining terms form the final tier, or "neighborhood tier" called by Koren [14], as it contributes some adjustments to the profile.

The algorithm of SVD++ is shown in figure 2.

The entire SVD++ model usually begins with standard SVD decomposition and calculation of similarity measure as mentioned before. After forming $S^k(i)$, a gradient descent algorithm will be applied to get predicted rating from equation SVD++ Model in Three-Tier [14], as developed by Koren et al[16].

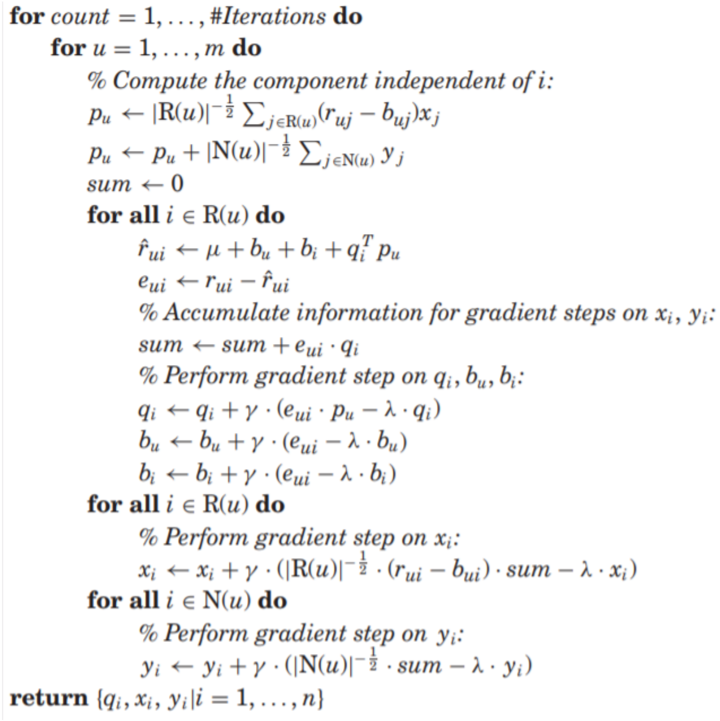
SVD++ model arouses great research interests has been applied and tested in numerous dataset [11] [14] and it generally beats traditional SVD-CF model. However, the contribution of SVD++ is far more than an improvement in benchmark but a new field for exploration: context [9], or "any information that can be used to characterize the situation of entities" by Jallouli et al. Scholars then incorporate more contexts (trust, temporal dynamics, environmental information, etc.) into SVD++ method and developed methods like TrustMF, TimeSVD++, and more [9] [21].

Jallouli et al., for instance, suggests three modifications to SVD++ model to include influence of previously preferred items, user-user social information, and environmental information [9]. The first modification brings the following rating prediction function (notice the term $\sum_{i \in I_u} y_i$ as latent factors of previously preferred items):

$$\hat{r}_{ui} = \mu + b_i + b_u + (q_j^T + |I_u|^{-1/2}\sum_{i \in I_u} y_i)p_u \tag{15}$$

The minimum objective function is presented as follows:

Figure 2: SVD++ Algorithm



```
for $count = 1, \dots, \#Iterations$ do
    for $u = 1, \dots, m$ do
        % Compute the component independent of i:
        $p_u \leftarrow |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} (r_{uj} - b_{uj}) x_j$
        $p_u \leftarrow p_u + |\mathrm{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}(u)} y_j$
        $sum \leftarrow 0$
        for all $i \in \mathrm{R}(u)$ do
            $\hat{r}_{ui} \leftarrow \mu + b_u + b_i + q_i^T p_u$
            $e_{ui} \leftarrow r_{ui} - \hat{r}_{ui}$
            % Accumulate information for gradient steps on $x_i$, $y_i$:
            $sum \leftarrow sum + e_{ui} \cdot q_i$
            % Perform gradient step on $q_i, b_u, b_i$:
            $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$
            $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda \cdot b_u)$
            $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda \cdot b_i)$
        for all $i \in \mathrm{R}(u)$ do
            % Perform gradient step on $x_i$:
            $x_i \leftarrow x_i + \gamma \cdot (|\mathrm{R}(u)|^{-\frac{1}{2}} \cdot (r_{ui} - b_{ui}) \cdot sum - \lambda \cdot x_i)$
        for all $i \in \mathrm{N}(u)$ do
            % Perform gradient step on $y_i$:
            $y_i \leftarrow y_i + \gamma \cdot (|\mathrm{N}(u)|^{-\frac{1}{2}} \cdot sum - \lambda \cdot y_i)$
return $\{q_i, x_i, y_i | i = 1, \dots, n\}$
```

$$\begin{aligned}\mathcal{L} = &\frac{1}{2}\sum_{(u,j)}(r\hat{_{uj}} - r_{uj})^2 + \frac{\lambda}{2}\sum_{u \in U_j}|I_u|^{-1/2}b_u^2 + \frac{\lambda}{2}\sum_{j \in I_u}|U_j|^{-1/2}b_j^2 \\ &+ \frac{\lambda}{2}\sum_{u \in U_j}|I_u|^{-1/2}\|p_u\|_F^2 + \frac{\lambda}{2}\sum_{j \in I_u}|U_j|^{-1/2}\|q_j\|_F^2 \\ &+ \frac{\lambda}{2}\sum_{j \in I_u}|U_i|^{-1/2}\|y_i\|_F^2\end{aligned}$$

First Modification of SVD++ by Jallouli et al. [9]

Notice that the regularization term $|U_j|^{-1/2}$ and $|I_u|^{-1/2}$ are weighted-λ-regularization approach mentioned before. We can then devise a gradient descent method for this minimum objective function. The second and third modification is essentially similar to the first one in structure and the formula are omitted here for brevity.

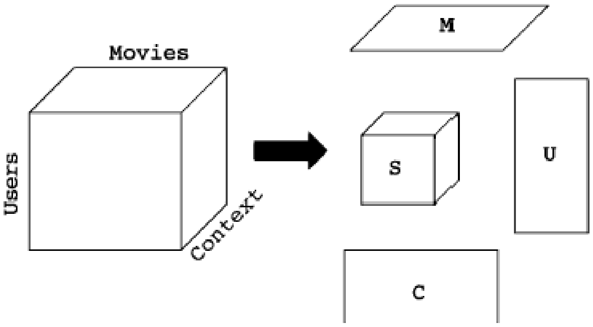
2.6 Current Research Trends

Since 2010s, researchers have proposed various models built on standard collaborative filtering framework. A current modern modification is "context", as suggested before, in which researchers add terms to p_u q_i terms to represent "trust", an abstraction of user-user social interaction that influence user ratings. Some researchers even represent the latent factors in Hilbert Feature Space and play the "kernel trick" [19] but this is beyond the scope of this paper.

2.6.1 Social Information

Some researchers are also interested in utilizing social network information to predict ratings [20] [21].

Figure 3: Illustration of HOSVD Model [12]



2.6.2 Tensor Decomposition - Higher Order SVD

Higher Order SVD (HOSVD) can be applied to context-based recommendation in the same way as other MF techniques [12]. In HOSVD, the prediction model for tensor $R \in \mathbb{R}^{n*m*c}$ is shown in figure 2.6.2 and equation 16 [8]. The minimum objective function $\mathcal{L}$ is shown in equation 17.

$$\hat{R}_{uik} = S \times_P P_{u*} \times_Q Q_{i*} \times_C C_{k*} \tag{16}$$

$$\mathcal{L} = \text{Loss}(R, \hat{R}) + \Omega(P, Q, C) + \Omega(S) \tag{17}$$

Here are some notations:

- $S \in \mathbb{R}^{d_P*d_Q*d_C}$ central tensor. $P \in \mathbb{R}^{n*d_P}$ user matrix. $Q \in \mathbb{R}^{m*d_Q}$ item matrix. $C \in \mathbb{R}^{c*d_C}$ context matrix. $R \in \mathbb{R}^{n*m*c}$ rating observations.
- Ω (L1/L2) regularization function.

Then, the factorization part is shown in Algorithm 4.

Tang et al. also propose a HOSVD model for neighborhood approach in collaborative filtering [23] but this is beyond the scope of this paper.

3 Comparison and Discussion of SVD-Based Framework

Technique	Explicit Rating	Implicit Rating	Data Imputation	Biases Used	Regularization
SVD-CF	✓		✓		
RSVD-CF	✓				✓
SVD++	✓	✓		✓	✓
TrustSVD	✓	✓		✓	✓
TimeSVD++	✓	✓		✓	✓

Table 1: Comparison of SVD Based Techniques[21]

Algorithm 4 Tensor Factorization by Karatzoglou et al.[12]

```
Input: $S, P, Q, C$ are filled with random values
  $t = t_0$
  while $(u, i, k)$ in $R$ do
    $\eta = 1/\sqrt{t}$; $t$ += 1; $l_t = \text{Loss}(R_{uik}, \hat{R}_{uik})$
    $\hat{R}_{uik} = S \times_P P_{u*} \times_Q Q_{i*} \times_C C_{k*}$
    $\partial_{P_{u*}}\text{Loss} = \partial_{P_{u*}}(l_t \; S \times_Q Q_{i*} \times_C C_{k*})$
    $\partial_{Q_{i*}}\text{Loss} = \partial_{P_{u*}}(l_t \; S \times_P P_{u*} \times_C C_{k*})$
    $\partial_{C_{k*}}\text{Loss} = \partial_{P_{u*}}(l_t \; S \times_P P_{u*} \times_Q Q_{i*})$
    $\partial_S\text{Loss} = \partial_S(l_t \; \bigotimes P_{u*} \bigotimes Q_{i*} \bigotimes C_{k*})$
    $P_{u*} -= \eta(\lambda_P P_{u*} + \partial_{P_{u*}}\text{Loss})$
    $Q_{i*} -= \eta(\lambda_Q Q_{i*} + \partial_{Q_{i*}}\text{Loss})$
    $C_{k*} -= \eta(\lambda_C C_{k*} + \partial_{C_{k*}}\text{Loss})$
    $S -= \eta(\lambda_S S + \partial_S\text{Loss})$
  end while
```

4 Conclusion

References

[1] Lamis Al Hassanieh, Chadi Abou Jaoudeh, Jacques Bou Abdo, and Jacques Demerjian. Similarity measures for collaborative filtering recommender systems. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–5. IEEE, 2018.

[2] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer, 2007.

[3] Jesús Bobadilla, Antonio Hernando, Fernando Ortega, and Abraham Gutiérrez. Collaborative filtering based on significances. *Information Sciences*, 185(1):1–17, 2012.

[4] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49:136–146, 2015.

[5] Stephen Gower. Netflix prize and svd, 2014.

[6] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel recommendation model regularized with user trust and item ratings. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1607–1620, 2016.

[7] Lin Guo and Qin Ke Peng. A combinative similarity computing measure for collaborative filtering. In *Applied Mechanics and Materials*, volume 347, pages 2919–2925. Trans Tech Publ, 2013.

[8] Maryam Jallouli, Sonia Lajmi, and Ikram Amous. Latent factor model applied to recommender system: realization, steps and algorithm. In *European, Mediterranean, and Middle Eastern Conference on Information Systems*, pages 606–618. Springer, 2017.

[9] Maryam Jallouli, Sonia Lajmi, and Ikram Amous. When contextual information meets recommender systems: extended svd++ models. *International Journal of Computers and Applications*, pages 1–8, 2020.

[10] Youchun Ji, Wenxing Hong, Yali Shangguan, Huan Wang, and Jing Ma. Regularized singular value decomposition in news recommendation system. In *2016 11th International Conference on Computer Science & Education (ICCSE)*, pages 621–626. IEEE, 2016.

[11] Yancheng Jia, Changhua Zhang, Qinghua Lu, and Peng Wang. Users' brands preference based on svd++ in recommender systems. In *2014 ieee workshop on advanced research and technology in industry applications (wartia)*, pages 1175–1178. IEEE, 2014.

[12] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. pages 79–86, 01 2010.

[13] Dohyun Kim and Bong-Jin Yum. Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, 28(4):823–830, 2005.

[14] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[15] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.

[16] Yehuda Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1–24, 2010.

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

[18] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56:156–166, 2014.

[19] Xinyue Liu, Chara Aggarwal, Yu-Feng Li, Xiaugnan Kong, Xinyuan Sun, and Saket Sathe. Kernelized matrix factorization for collaborative filtering. pages 378–386, 06 2016.

[20] Hao Ma, Dengyong Zhou, Chao Liu, Michael Lyu, and Irwin King. Recommender systems with social regularization. pages 287–296, 01 2011.

[21] Rachana Mehta and Keyur Rana. A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 269–274. IEEE, 2017.

[22] Ritu Sharma, Dinesh Gopalani, and Yogesh Meena. Collaborative filtering-based recommender system: Approaches and research challenges. In *2017 3rd international conference on computational intelligence & communication technology (cict)*, pages 1–6. IEEE, 2017.

[23] Xiaoyu Tang, Yue Xu, and Shlomo Geva. Learning higher-order interactions for user and item profiling based on tensor factorization. In Oliver Brdiczka, Polo Chau, Giuseppe Carenini, Shimei Pan, and Per Ola Kristensson, editors, *Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI 2015, Atlanta, GA, USA, March 29 - April 01, 2015*, pages 213–224. ACM, 2015.

[24] Manolis Vozalis, Angelos Markos, and Konstantinos Margaritis. On the performance of svd-based algorithms for collaborative filtering. In *2009 Fourth Balkan Conference in Informatics*, pages 245–250. IEEE, 2009.

[25] Manolis G Vozalis and Konstantinos G Margaritis. Applying svd on generalized item-based filtering.

[26] Shuai Zheng, Chris Ding, and Feiping Nie. Regularized singular value decomposition and application to recommender system. *arXiv preprint arXiv:1804.05090*, 2018.

[27] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In Rudolf Fleischer and Jinhui Xu, editors, *Algorithmic Aspects in Information and Management*, pages 337–348, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.