Sparsity Regularization for Cold-Start Recommendation

Aksheshkumar Ajaykumar Shah[1*] , **Hemanth Venkateshwara**[2]

Center for Cognitive Ubiquitous Computing (CUbiC), Arizona State University

{aashah16, hemanthv}@asu.com

Abstract

Recently, Generative Adversarial Networks (GANs) have been applied to the problem of Cold-Start Recommendation, but the training performance of these models is hampered by the extreme sparsity in warm user purchase behavior. In this paper we introduce a novel representation for user-vectors by combining user demographics and user preferences, making the model a hybrid system which uses Collaborative Filtering and Content Based Recommendation. Our system models user purchase behavior using weighted user-product preferences (explicit feedback) rather than binary user-product interactions (implicit feedback). Using this we develop a novel sparse adversarial model, SRLGAN, for Cold-Start Recommendation leveraging the sparse user-purchase behavior which ensures training stability and avoids over-fitting on warm users. We evaluate the SRLGAN on two popular datasets and demonstrate state-of-the-art results.

1 Introduction

Recommendation systems identify a fraction of items from a very large inventory of items and recommend them to a user for purchase. This task is achieved with the help of the user's past purchase behavior [Bobadilla *et al.*, 2013; Li *et al.*, 2017]. Recommendation systems help in enhancing the overall purchase experience for the users. Recommendation systems can be broadly classified into three categories, viz., Collaborative Filtering-based, Content-based and Hybrid Systems [Feng *et al.*, 2019]. Among these, collaborative filtering is the most popular approach [Adomavicius and Tuzhilin, 2005]. In collaborative filtering, the past purchase behavior of the user is used for making recommendations [Smith and Linden, 2017]. The user purchase behavior consists of user's feedback which can be either implicit or explicit. Implicit feedback is binary and represents the user's interaction with the items (purchased or not-purchased) [Sidana *et al.*, 2021]. Explicit feedback is generally the discrete ratings assigned by the user ranging

---

[*]Contact Author

between 1-to-C for every item purchased.

Collaborative filtering can be performed using the matrix factorization [Li *et al.*, 2020], where the user and item information is projected into a K-dimensional latent space and their interaction is modelled by the inner product of the latent vectors. Matrix factorization attempts to model the linear relationship between users and items and thereby predicts the items a new user would purchase. Deep learning based collaborative filtering can model highly nonlinear relationships between users [LeCun *et al.*, 2015] and items and therefore exhibits superior performance to traditional matrix factorization approaches [HE *et al.*, 2017; Wu *et al.*, 2016].

Collaborative filtering is successful in recommending items for *warm users* - users whose previous purchase behavior is known. However, the approach fails in recommending items for *cold users* - users whose purchase behavior is unknown [Li *et al.*, 2017]. This problem is popularly known as Cold-Start Recommendation [Lin *et al.*, 2013]. It is of two types, (i) User Cold-Start, and (ii) Item Cold-Start. In the former, the user is new and no purchase history is available and in later, the item is new and no user interactions are available. When both are compared, the User Cold-Start problem is more complex and also more popular.

The Cold-Start Recommendation problem can be tackled by using cross-domain information [Fernández-Tobías *et al.*, 2012], personal information [Fernández-Tobías *et al.*, 2016] and social-network information [Sedhain *et al.*, 2017] of the user. These kinds of models are called as Content-based systems. These systems have a limitation where they tend to suggest the same items to all users who are similar even if the users have rated the items differently [Lee *et al.*, 2019]. To overcome this problem hybrid systems based on Collaborative Filtering which use content information are widely used [Cheng *et al.*, 2016; Kouki *et al.*, 2015].

The popular Generative Adversarial Networks (GANs) models from computer vision and natural language processing have been successfully applied towards collaborative filtering [Goodfellow *et al.*, 2014]. The potential of GANs

for Collaborative Filtering based Recommendation systems has been demonstrated by methods such as IRGAN [Wang *et al.*, 2017], GraphGan [Wang *et al.*, 2018] and CFGAN [Chae *et al.*, 2018]. In these models the Generator is used to generate the user purchase behavior and the Discriminator is trained to distinguish between the ground truth purchase behavior and the generated purchase behavior (output of the Generator).

In this paper we use GANs to implement user-based Cold Start Recommendation. We first introduce a novel technique to represent user information by combining the users' demographic information and their preferences. This is implemented by creating a Term Frequency - Inverse Document Frequency (TF-IDF) vector for each user. The TF-IDF vectors give a weighted score to user attributes including genre preferences which allows us to leverage the advantages of collaborative filtering as well as Content-based methods. We model user-purchase behavior using explicit feedback from the users where user-ratings for items are taken into consideration instead of binary implicit feedback which merely indicates if a user has purchased the item or not. The Generated user-purchase behavior is in the form of vectors instead of discrete item ID's to take full advantage of adversarial training as also implemented in [Chae *et al.*, 2018]. In addition we propose a novel sparsity regularization that models the high sparsity in the user-purchase behavior, which in turn ensures stability when training the GAN and avoids the problem of over-fitting on warm users. The proposed model is tested on two popular datasets namely MovieLens 100K and MovieLens 1M [Harper and Konstan, 2015] and demonstrates state-of-the-art results.

In summary, the main contributions of this paper are listed as follows:

- TF-IDF based representation of user-vectors containing demographic information and preferences.

- A novel hybrid Cold-Start Recommendation system combining Collaborative Filtering and Content-based approaches.

- A novel sparsity regularization to model the sparsity in the data and ensure training stability.

- Competitive performance on the popular MovieLens 100K and MovieLens 1M datasets.

2 Related Work

In this section we discuss Collaborative filtering methods based on deep neural networks and specifically GANs from literature.

2.1 Deep Neural Network based Collaborative Filtering Methods

According to [Li *et al.*, 2019], Cold-Start Recommendation models can be classified into three categories, viz., (i) Similarity-based models, (ii) Matrix Factorization models, and (iii) Feature Mapping models. The most popular approach is Matrix factorization, where the user-item

relationship matrix is typically factorized into two latent representations which are used to make predictions. Matrix factorization models the linear interactions between the latent features of the users and the items. Popular methods include, BPR [Rendle *et al.*, 2012] and FISM [Kabbur *et al.*, 2013].

Deep Neural Network based Collaborative filtering models have been gaining popularity owing to their capabilities of exploiting the non-linear interactions between the features and developing an arbitrary continuous function [LeCun *et al.*, 2015]. Some of the popular top-n recommendation models are CDAE [Wu *et al.*, 2016] and NCF [HE *et al.*, 2017].

2.2 Generative Adversarial Networks based Collaborative Filtering Methods

In domain of recommendation systems Generative Adversarial Networks have also been implemented by some methods for Collaborative filtering. They have adopted the adversarial training approach rather than the traditional approach of optimizing the pointwise or pairwise objective function for Collaborative filtering. In IRGAN [Wang *et al.*, 2017] and GraphGAN [Wang *et al.*, 2018], the Generator attempts to generate discrete item indices and the Discriminator attempts to distinguish between the synthetically generated item from the ground truth item. When discrete item indices are generated, the Discriminator has difficulty identifying if the item is relevant or not [Chae *et al.*, 2018]. This degrades the quality of the feedback provided by Discriminator to Generator and fails to avail the advantages of GANs. To overcome this problem CFGAN [Chae *et al.*, 2018], proposed a framework where vectors are generated instead of discrete item ID's to prevent Discriminator's confusion. This allows the Discriminator to guide the Generator consistently and produce results closer to the ground truth. The VAEGAN [Yu *et al.*, 2019] proposes a model which uses Variational Bayes along with a Generative Adversarial Networks to generate recommendations on strong generalization and weak generalization. As the above models do not take into account the sparsity in the user-purchase behavior they are prone to over-fitting and instability during training. Our proposed model leverages the sparse user-purchase behavior which ensures training stability and avoids over-fitting on warm users.

3 Proposed Approach

3.1 Problem Settings

The Cold-Start Recommendation problem is of two types, User-based and Item-based. In this work we discuss user-based models, but the same principles can be extended to the item-based setting as well. The User-based Cold-Start Recommendation consists of warm users (known users i.e., purchase behavior is known) and cold users (unknown users i.e., purchase behavior is unknown). The user attributes are vectors consisting of the user's demographic information and preferences represented in the form of Term-Frequency-Inverse-Document-Frequency (TF-IDF) vectors which gives
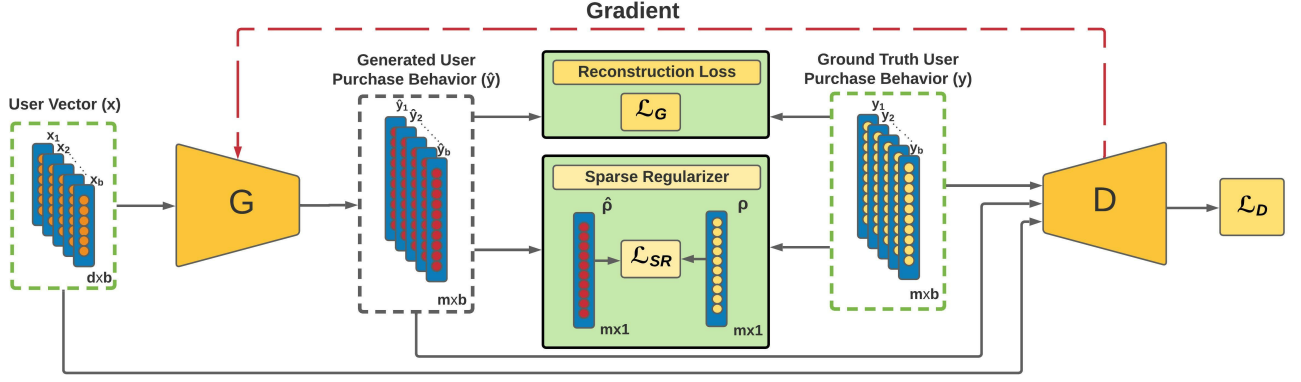
Figure 1: System diagram of the SRLGAN. The input to the Generator module are user attributes of dimensions $(d \times b)$ and the output of the Generator are the predicted user purchase behaviors of dimensions $(m \times b)$. The objective term $\mathcal{L}_G$ (Eq. 1), is a reconstruction loss to train the Generator to predict user purchase behavior similar to the ground truth. The objective term $\mathcal{L}_D$ is the least squares GAN objective in Eq. 2. The novel Sparse Regularizer models the sparsity in the user purchase behavior. It aligns the distributions of the predicted and ground truth user purchase behavior with $\mathcal{L}_{SR}$ (Eq. 4), by minimizing the KL-divergence between the average generated (predicted) user purchase prediction $\hat{\rho}$ and the average ground truth user purchase prediction ρ.

weighted representation of the users interests in genres. The set of warm users is represented as $\mathcal{X}^s = \{x_i^s\}_{i=1}^{n_s}$, where n_s is number of warm users and $x \in \mathbb{R}^d$. The users purchase from m items and also provide a rating from $1, \ldots, C$ for each of the purchased items. A user's purchase behavior is the vector of m ratings where a 0 indicates the item was not purchased and 1 is the least rating and C is the best rating. The ground truth purchase behaviors corresponding to the warm users are $\mathcal{Y}^s = \{y_i^s\}_{i=1}^{n_s}$, where $y \in [0,1]^m$ is a discrete vector of m dimensions (items) where the ratings have been normalized by dividing with C. The cold users are $\mathcal{X}^u = \{x_i^u\}_{i=1}^{n_u}$, where n_u is the number of cold users. Let the space of user attributes be $\mathcal{X}$ and the space of user purchase behaviors be $\mathcal{Y}$. The goal of Cold-Start recommendation is to learn a function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$, and thereby predict the purchase behavior $\{\hat{y}_i^u\}_{i=1}^{n_u}$ of cold users in $\mathcal{X}^u$.

3.2 Adversarial User Purchase Behavior Generation

We propose a generative model to hallucinate the purchase behavior of the users. We apply a conditional Generative Adversarial Network (GAN) which takes user attributes as input to the Generator and outputs user purchase behavior corresponding to the user. The Discriminator is trained to distinguish ground-truth user purchase behavior vs. generated user behavior. The components of the GAN model are described below.

User Purchase Behavior Generator
The conditional generator is a mapping $G_{\theta_g} : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters θ_g where, $\mathcal{X}$ and $\mathcal{Y}$ are the space of user attributes and user purchase behavior, respectively. The Generator is trained to predict user purchase behavior for warm users $(\hat{y}^s \leftarrow G_{\theta_g}(x^s))$ as well as for cold users $(\hat{y}^u \leftarrow G_{\theta_g}(x^u))$. The ground truth warm user purchase behavior y^s can be used

to supervise the Generator with a least squares objective,

$$\mathcal{L}_G = \min_{\theta_g} \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i^s - G_{\theta_g}(x_i^s))^2. \quad (1)$$

User Purchase Behavior Discriminator
We train the conditional GAN with an adversarial Discriminator which is inspired by LSGAN to ensure training stability [Mao *et al.*, 2017]. The loss function for this GAN is a least squares error in place of the standard binary cross entropy loss as it gives a better gradient for the model to learn. The input to the Discriminator is a user attribute vector (x) concatenated with the user purchase vector (y). The Discriminator learns a mapping $D_{\theta_d} : \mathcal{X} \times \mathcal{Y} \rightarrow [0,1]$, where θ_d are the parameters of the Discriminator. The Discriminator is trained to distinguish between the ground truth user purchase behavior (y) and the generated user purchase behavior $(\hat{y})$. The objective function for training Discriminator and Generator is,

$$\mathcal{L}_D = \min_{\theta_g} \max_{\theta_d} \frac{1}{2} \mathbb{E}_{(x,y)\sim(\mathcal{X}^s,\mathcal{Y}^s)} \big[(D_{\theta_d}(x,y) - 1)^2\big]$$
$$+ \frac{1}{2} \mathbb{E}_{x\sim\mathcal{X}^s} \big[(D_{\theta_d}(x, G_{\theta_g}(x))^2\big]. \quad (2)$$

The objective function is maximized w.r.t. the Discriminator's parameters, θ_d and minimized w.r.t. the Generator's parameters, θ_g.

3.3 Sparsity Regularization

When there are a large number of items, the purchase behavior of a random user is usually highly sparse with lots of 0s in the purchase behavior vector, i.e., the user has purchased only a few items. The user purchase behavior is a discrete vector $y \in [0,1]^m$ with high sparsity (multiple zero values or multiple not purchased items). We observed instability in training with poor convergence when we attempted to train the Cold-Start recommendation GAN without taking into account sparsity. In addition, GAN based models are biased

towards warm users leading to poor generalization and can often end up in a mode collapse where the Generator outputs the same user-purchase behavior for all users. With inspiration from the sparse autoencoder [Ng and others, 2011], we introduce a novel regularization to model the sparsity in user purchase behavior by aligning the distributions of the predicted purchase behavior and the ground truth purchase behavior. We treat the purchase behavior for each item as a Bernoulli random variable with an unknown mean. We estimate the mean purchase behavior for all the items from the training set for n_s samples $\{y_i^s\}_{i=1}^{n_s}$. The average purchase behavior for the dataset is ρ, where,

$$\rho = \frac{1}{n_s} \sum_{i=1}^{n_s} y_i^s. \qquad (3)$$

$\rho = [\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(m)}]^\top$ is the average ground truth purchase behavior where $\rho^{(i)} \in [0, 1]$ is the average purchase behavior for item i. Likewise, the average predicted purchase behavior for a batch size b is given by $\hat{\rho} = \frac{1}{b} \sum_{i=1}^{b} \hat{y}_i$. This is estimated from the output of the Generator. The Kullback-Leibler divergence $\text{KL}(\rho^{(i)}||\hat{\rho}^{(i)})$, is a measure of divergence between a Bernoulli random variable with mean $\rho^{(i)}$ and a Bernoulli random variable with mean $\hat{\rho}^{(i)}$. We propose a sparsity regularization to align the distributions of the predicted user purchase behaviors with the ground truth user purchase behaviors, which we model as Bernoulli random variables. This regularize the user purchase prediction by minimizing the divergence between the distributions of the predicted and ground truth user purchase behaviors. The sparsity regularization for a batch of m user purchase predictions is defined as,

$$\mathcal{L}_{SR} = \sum_{i=1}^{m} \text{KL}(\rho^{(i)}||\hat{\rho}^{(i)})$$
$$= \sum_{i=1}^{m} \rho^{(i)} \log \frac{\rho^{(i)}}{\hat{\rho}^{(i)}} + (1 - \rho^{(i)}) \log \frac{(1 - \rho^{(i)})}{(1 - \hat{\rho}^{(i)})}. \quad (4)$$

The divergence $\text{KL}(\rho^{(i)}||\hat{\rho}^{(i)}) = 0$ if $\rho^{(i)} = \hat{\rho}^{(i)}$, otherwise it increases monotonically to infinity as the value of $\hat{\rho}^{(i)}$ diverges from $\rho^{(i)}$. The regularization prevents the Generator from getting over-fitted on the warm users and also helps in avoiding the problem of mode collapse which are two of the prominent problems in the training of GAN based recommendation systems.

3.4 SRLGAN Objective Function

We implement a user-based Cold-Start Recommendation model called the **S**parse **R**egu**L**arized **G**enerative **A**dversarial **N**etwork (SRLGAN). The SRLGAN is trained using content-based user representation captured in the form of user preferences and the Collaborative filtering gathered from the user's purchase behavior. The training of the SRLGAN is driven by 3 objective functions. The Generator component in the SRLGAN is guided by the least squares loss to predict the ground truth user purchase behavior for warm users (see Eq. 1). The Discriminator and the Generator are trained using the

adversarial Least Squares objective (see Eq. 2). The SRLGAN leverages the sparsity in the user purchase behaviour distribution to train the Generator and avoid over fitting to warm users (see Eq. 4). The overall objective function of the SRLGAN is,

$$\mathcal{L}_G + \mathcal{L}_D + \beta \mathcal{L}_{SR}, \qquad (5)$$

where, $\beta \geq 0$ is a hyper-parameter that controls the importance of the sparse regularization. Larger the value of β, higher the sparsity in the predicted user purchase behavior. The value of β is estimated using cross validation when predicting the purchase of warm users. The SRLGAN is depicted in Fig. 1.

4 Experiments and Analysis

4.1 Datasets

The SRLGAN is evaluated using two popular datasets, MovieLens 100k and MovieLens 1M [Harper and Konstan, 2015]. In our experiment we divide the datasets into user attributes space, $\mathcal{X}$ and the user purchase behavior space, $\mathcal{Y}$. The user attributes space $\mathcal{X}$ consists of user vectors x of dimension 103 and 48 for MovieLens 100K and MovieLens 1M, respectively. The user vector consists of user attributes such as demographic information of the users including ages, gender, occupation and 19 movie genres as defined in the dataset. Since all the details provided related to the user are not equally important, the user vector is represented in the form of TF-IDF vectors. This helps in giving higher weighted score to the attributes that are dominant. Movie tags of users' previously seen movies are used to indicate the presence of each genre. Term frequency is the count of a user attribute. In order to reduce the importance of common attributes and highlight the dominant attributes, inverse document frequency is calculated using the frequency of attributes across all users. The final user attribute vector is an element-wise product of the term frequency vector and the inverse document frequency vector. The user purchase behavior space consists of the ratings given by the user to the products. We have considered explicit representation of user interaction as it gives us graded score of users' interest in the product. The details related to number of unique users and items in the dataset is defined in Table 2. To ensure a fair comparison across different approaches, 20% of the users were selected randomly as cold users (test set) and the remaining 80% were treated as warm users. In real world scenarios, the user vectors for cold users can be estimated by asking the users to note their preferences for different genres.

4.2 Implementation Details and Performance Metrics

Evaluation Metrics

We have adopted three common metrics which are used for the evaluation of Top-n recommendation systems [Chae *et al.*, 2018], [Yu *et al.*, 2019]. (P@n) is the Precision P when considering n items. (N@n) is the Normalized Discounted Cumulative Gain (NDCG) N for n items and (M@n) is the Mean Reciprocal Rank M for n items. The last two metrics are based on the rank of the correctly predicted item, where lower the rank, higher is the score. The metrics are evaluated

	MovieLens 100K						MovieLens 1M					
	P@5	P@20	N@5	N@20	M@5	M@20	P@5	P@20	N@5	N@20	M@5	M@20
ItemPop	.181	.138	.163	.195	.254	.292	.157	.121	.154	.181	.252	.297
BPR [Rendle *et al.*, 2012]	.348	.236	.370	.380	.556	.574	.341	.252	.349	.362	.537	.556
FISM [Kabbur *et al.*, 2013]	.426	.285	.462	.429	.674	.685	.420	.302	.443	.399	.637	.651
CDAE [Wu *et al.*, 2016]	.433	.287	.465	.425	.664	.674	.419	.307	.439	.401	.629	.644
GraphGAN [Wang *et al.*, 2018]	.212	.151	.183	.249	.282	.312	.178	.194	.205	.184	.281	.316
IRGAN [Wang *et al.*, 2017]	.312	.221	.342	.368	.536	.523	.263	.214	.264	.246	.301	.338
CFGAN [Chae *et al.*, 2018]	.444	.292	.476	.433	.681	.693	.432	.309	.455	.406	.647	.660
VAEGAN [Yu *et al.*, 2019]	-	-	.468	.437	.688	**.700**	-	-	.465	.416	.663	.676
SRLGAN [Ours]	**.521**	**.444**	**.530**	**.466**	**.690**	.699	**.499**	**.436**	**.504**	**.453**	**.674**	**.683**

Table 1: Comparison of the performance of SRLGAN on MovieLens 100K and MovieLens 1M datasets with respect to the baselines. Here, P@n stands for Precision (P), N@n stands for Normalized Discounted Cumulative Gain (N) and M@n stands for Mean Reciprocal Rank (M) when considering n items. The value of n = 5,20. The best results are highlighted in bold.

Dataset	Users	Items	% Sparsity
MovieLens 100K	943	1682	93.69
MovieLens 1M	6040	3952	95.80

Table 2: Details of the Datasets. The % Sparsity is the percentage of 0s in the user purchase behavior space $\mathcal{Y}$.

for n at 5 and 20.

Baselines

We compare the performance of the SRLGAN against multiple Top-n recommendation models. **ItemPop** is a non-personalized method in which the items are ranked in descending order of the number of purchase records. In the **BPR** [Rendle *et al.*, 2012] method, the relative order of preferences of the purchased and non-purchased item pairs is optimized. In the **FISM** [Kabbur *et al.*, 2013] approach, item-item similarity matrix is learned as a product of two low-dimensional latent factor matrices. In the **CDAE** [Wu *et al.*, 2016] method, collaborative filtering is performed by using a denoising autoencoder along with user-specific latent features. **IRGAN** [Wang *et al.*, 2017] is a popular method that applies GANs for the purpose of collaborative filtering. The **GraphGAN** [Wang *et al.*, 2018] approach uses graph softmax, which is softmax applied over a graph structure. **CFGAN** [Chae *et al.*, 2018] is a vector-wise adversarial training approach that is closely related to the SRLGAN. The **VAEGAN** [Yu *et al.*, 2019] applies a variational autoencoder with a contractive loss to generate recommendations.

Model Details

The Generator and the Discriminator in the SRLGAN are implemented as fully-connected neural network layers with LeakyRelu activations except for the final layers, which have Sigmoid activations. The user data is of dimensions $(d \times b)$, where d is the dimension of the user vector and b is the batch size. The user-purchase behavior is of dimensions $(m \times b)$, where m is the number of items. The number of neurons in the Generator layers are, $[d, 512, 1024, 1024, m]$. The number of neurons in the Discriminator layers are, $[m +$

Algorithm 1 Training the SRLGAN

Input: $x^s \in \mathcal{X}^s, y^s \in \mathcal{Y}^s, \rho$
Constants: β, b, n_e, n_G, n_D
Output: G_{θ_g}

1: Initialize G_{θ_g} and D_{θ_d}
2: **for** n_e iterations **do**
3:     Update G_{θ_g} using $\{x_i^s, y_i^s\}_{i=1}^b$ with Eq. 1
4: **end for**
5: **while** Not Converged **do**
6:     **for** n_D iterations **do**
7:         Generate $\hat{y}_i \leftarrow G_{\theta_g}(x_i^s)$ using $\{x_i^s, y_i^s\}_{i=1}^b$
8:         Update D_{θ_d} and G_{θ_g} using Eq. 2
9:     **end for**
10:     **for** n_G iterations **do**
11:         Generate $\hat{y}_i \leftarrow G_{\theta_g}(x_i^s)$ using $\{x_i^s, y_i^s\}_{i=1}^b$
12:         calculate $\hat{\rho}$ using Eq. 3 & Update G_{θ_g} using Eq. 5
13:     **end for**
14: **end while**
15: **Return** G_{θ_g}

$d, 2048, 512, 128, 1]$. A dropout of 0.4 is applied to all the hidden layers of the Discriminator to prevent over-fitting, and the learning rate is 10^{-6}. The Generator is initially trained using the $\mathcal{L}_G$ loss to lead the Discriminator. This ensures stable training for the SRLGAN. Algorithm 1 outlines the training procedure for the SRLGAN. For a given user x, the output of the Generator $(\hat{y} \leftarrow G_{\theta_g}(x))$ is of dimension m, where each component $\{\hat{y}^{(i)}\}_{i=1}^m$ denotes the probability of the user x purchasing item i. The components of $\hat{y}$ are sorted in descending order to gather the user's preference and to estimate the metrics Precision, NDCG and MRR.

4.3 Results and Analysis

Comparison Against Baselines

Table 1 outlines the performance of SRLGAN against multiple baselines for the two datasets. The SRLGAN outperforms the previous state-of-the-art baselines such as CFGAN and VAEGAN methods. When recommending top-5 items (P@5), the SRLGAN achieves an improvement of 17.3% for the Movielens 100K and an improvement of 15.5% for the
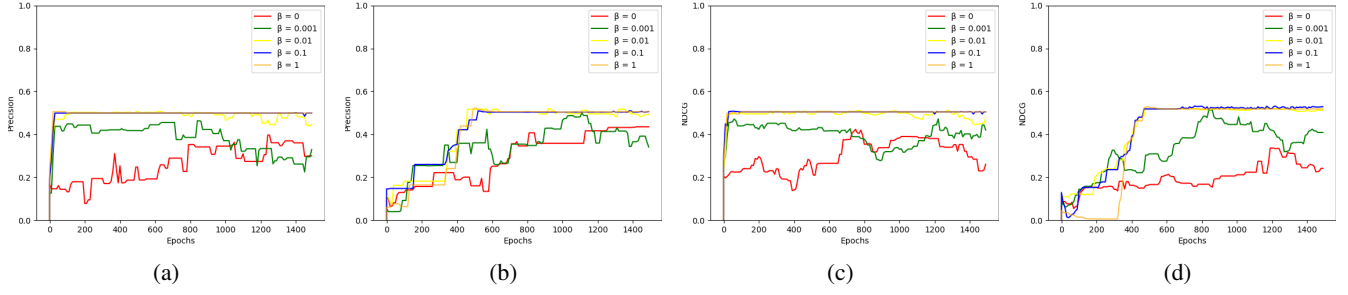
Figure 2: Depicts the training curves for Precision and NDCG as the value of β is varied. Fig. 2a, 2c are for MovieLens 1M and Fig. 2b, 2d are for MovieLens 100K. $\beta = 0.1$ produces the best results across both the datasets.
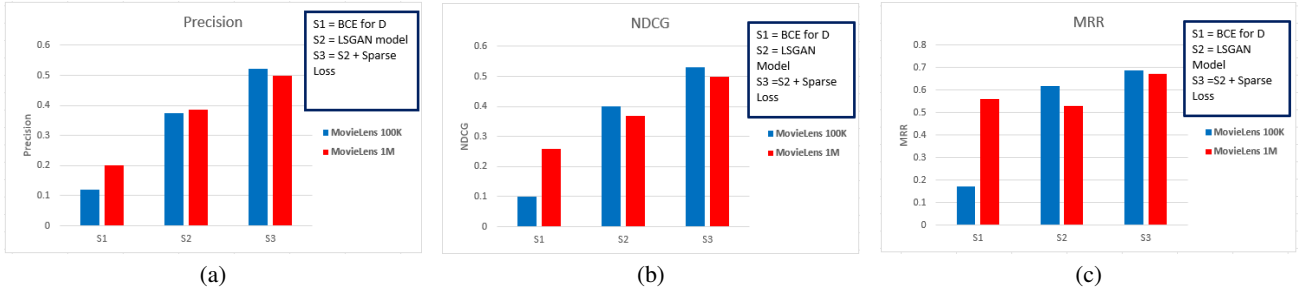

Figure 3: Depicts the ablation results for three different models S1, S2 and S2, each having different components of the SRLGAN. Fig. 3a depicts the variation in Precision, Fig. 3b showcases variations in NDCG scores, and Fig. 3c displays the variations in the MRR scores across the three models for both the datasets.

MovieLens 1M compared to the best model (CFGAN). Similarly, the SRLGAN gains 52% and 41.1% for MovieLens 100K and MovieLens 1M, respectively, when predicting the top-20 recommendations (P@20). The values in bold in Table 1 demonstrate that the SRLGAN outperforms the other approaches by significant margins across all three metrics for Cold Start recommendation.

Hyper-parameter Sensitivity

The β hyper-parameter controls the importance of sparsity regularization. We estimate the optimal value of $\beta = 0.1$ using cross-validation by predicting the purchase behavior of 10% of warm users in the training set after training with the remaining users. Fig. 2 plots the evolution of Precision and the NDCG scores for both the datasets for different values of β.

Ablation Study

We also report an ablation study of the SRLGAN in Fig. 3 to estimate the importance of different components in the SRLGAN. We treat the S1 model as the baseline. Here the loss function driving the training is the standard binary cross entropy (BCE) loss for the GAN and a reconstruction loss (Eq. 1) to train the Generator. In the S2 model we replace the BCE loss with the least squares loss (Eq. 2). Model S3 is model S2 along with the sparse regularization (Eq. 4). The plots in Fig. 3 depict a steady improvement in Precision, NDCG and MRR with the introduction of each of the components across both the datasets.

5 Conclusions

In this paper we have proposed a novel Cold-Start Recommendation model, SRLGAN which leverages the sparsity in the user purchase behavior distribution during training. The SRLGAN model employs a KL-divergence based sparse penalty which reduces the dissimilarity between the ground truth user purchase behavior distribution and the generated user purchase behavior distribution thereby implementing stable collaborative filtering for highly sparse datasets. Extensive experiments on two popular benchmark datasets demonstrate the SRLGAN outperforms popular approaches. It also verifies that our proposed sparse penalty prevents the model from over-fitting and getting into mode collapse. In the future we intend to test the performance of the SRLGAN on the related problem of item-based Cold-Start recommendation which is similar to the user-based problem where the item-user interaction data is highly sparse.

References

[Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. *IEEE Trans. KDE*, 2005.

[Bobadilla *et al.*, 2013] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46, 2013.

[Chae *et al.*, 2018] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. *CIKM*, 2018.

[Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & Deep Learning for Recommender Systems. In *Proc. of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016. Association for Computing Machinery, 2016.

[Feng *et al.*, 2019] Liang Feng, Qianchuan Zhao, and Cangqi Zhou. Improving Performance of Top-N Recommendations with Co-Clustering method. *Expert Systems With Applications*, 2019.

[Fernández-Tobías *et al.*, 2016] Ignacio Fernández-Tobías, Matthias Braunhofer, Mehdi Elahi, Francesco Ricci, and Iván Cantador. Alleviating the New User Problem in Collaborative Filtering by Exploiting Personality Information. *User Modeling and User-Adapted Interaction*, 2016.

[Fernández-Tobías *et al.*, 2012] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. *Proc. of the 2nd Spanish Conf. on Information Retrieval*, 2012.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Neurips*, 2014.

[Harper and Konstan, 2015] F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 2015.

[HE *et al.*, 2017] Xiangnan HE, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering. *WWW*, 2017.

[Kabbur *et al.*, 2013] Santosh Kabbur, Xia Ning, and George Karypis. Fism: Factored Item Similarity Models for Top-N Recommender Systems. In *SIGKDD*, 2013.

[Kouki *et al.*, 2015] Pigi Kouki, Shobeir Fakhraei, James Foulds, Magdalini Eirinaki, and Lise Getoor. HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems. In *Proc. of the 9th ACM Conf. on Recommender Systems*, RecSys '15. Association for Computing Machinery, 2015.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning, 2015.

[Lee *et al.*, 2019] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation, 2019.

[Li *et al.*, 2017] Jingjing Li, Ke Lu, Zi Huang, and Heng Tao Shen. Two Birds One Stone: On Both Cold-Start and Long-Tail Recommendation. In *Proc. of the 25th ACM Int. Conf. on Multimedia*, 2017.

[Li *et al.*, 2019] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. From zero-shot learning to cold-start recommendation. *Proceedings of the AAAI Conf. on AI*, 2019.

[Li *et al.*, 2020] Rui Li, Fulan Qian, Xiuquan Du, Shu Zhao, and Yanping Zhang. A Collaborative Filtering Recommendation framework based on Wasserstein GAN. *Journal of Physics: Conference Series*, 2020.

[Lin *et al.*, 2013] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers. In *SIGIR*, 2013.

[Mao *et al.*, 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks, 2017.

[Ng and others, 2011] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011), 2011.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback, 2012.

[Sedhain *et al.*, 2017] Suvash Sedhain, Aditya Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. Low-Rank Linear Cold-Start Recommendation from Social Data. *AAAI*, 2017.

[Sidana *et al.*, 2021] Sumit Sidana, Mikhail Trofimov, Oleh Horodnytskyi, Charlotte Laclau, Yury Maximov, and Massih-Reza Amini. User preference and embedding learning with implicit feedback for recommender systems. *Data Mining and Knowledge Discovery*, 2021.

[Smith and Linden, 2017] Brent Smith and Greg Linden. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, 2017.

[Wang *et al.*, 2017] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Zhang Peng, and Dell Zhang. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. *SIGIR*, 2017.

[Wang *et al.*, 2018] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. GraphGAN: Graph Representation Learning with Generative Adversarial Nets. *AAAI*, 2018.

[Wu *et al.*, 2016] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proc. of the 9th ACM Int. Conf. on Web Search and Data Mining*, WSDM '16, 2016.

[Yu *et al.*, 2019] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders. *IJCAI*, 2019.