Adversarial Machine Learning for Recommendation Systems

by

Aksheshkumar Ajaykumar Shah

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master Of Science

Approved February 2022 by the
Graduate Supervisory Committee:

Hemanth Venkateswara, Co-Chair
Spring Berman, Co-Chair
Leila J. Ladani

ARIZONA STATE UNIVERSITY

May 2022

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. This thesis introduces 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. This system models user purchase behavior using weighted user-product preferences (explicit feedback) rather than binary user-product interactions (implicit feedback). Using this a novel sparse adversarial model, **S**parse **R**egu**L**arized **G**enerative **A**dversarial **N**etwork (SRLGAN), is developed for Cold-Start Recommendation. SRLGAN leverages the sparse user-purchase behavior which ensures training stability and avoids over-fitting on warm users. The performance of SRLGAN is evaluated on two popular datasets and demonstrate state-of-the-art results.

DEDICATION

To my parents, Ajaykumar Shah and Parul Shah for never ending love and support by always having my back and allowing me to follow my dreams.

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to Dr. Hemanth Venkateshwara for his constant support, guidance and motivation throughout my Masters' journey at Arizona State University. I would not have been able to reach this level without his encouragement and trust in my research ideas. I would also like to thank my committee members Dr. Spring Berman and Dr. Leila Ladani, for their advice and feedback on my research work.

I would also like to thank my friend Maunil for insightful discussion that helped me to improve my research work and provided the right boost. I cannot forget the support my roommates from reading my paper drafts to encouraging me to push myself to create something new.

Last but not the least, I would like to thank my parents, my brother for their constant love and support that helped me to stay focused on my work. No words or actions will ever be enough to do justice to everything they have done for me.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

From the near infinite inventory the task finding the right products for the customer and recommending them to the customers is performed by recommendation systems. When we open an online portal such as Amazon, Netflix, Spotify, YouTube, etc. which consists of products and customers, there are a set of recommendations the application makes for its customers. These recommendations reduces the time customer needs to spend on finding the right product and are something that will help the portal increase its business. The recommendations can be done in two ways.

1. Recommending the top products from the platform.

2. Making a personalized set of recommendations for the user.

For the first method it is convenient as the platform has to just make a list of top selling products and recommend it to the user. This method sounds convenient and tempting but the there are chances that the customer might or might not like these recommendations [Li *et al.* (2021)]. The second method consists of making personalized recommendations for the customers. This can be pursued by using the customers previous purchase behavior [Bobadilla *et al.* (2013); Li *et al.* (2017)]. This will help in understanding what user might like and recommend it accordingly after learning about the user. This method gives our recommendations a higher probability that they will be liked by the user. If the user likes it this helps the platform to increase its revenue and also increase the loyalty of the customers. This is how recommendation systems helps the platform to increase its revenue as when the recommended products are of customers choice it increases his interest in the platform. The main motive

behind the recommendations is that to increase the revenue and customer loyalty and so it is seen that recommendation systems play a major role in the performance of the platform [Li *et al.* (2021)]. Another Advantage of recommendation systems is that it helps save the customers time by giving the recommendation based on the product of his choice so the customer does not have to spend time searching for it from millions of product and also the customer has a pleasant experience [Feng *et al.* (2019)].

1.1 Recommendation Systems

According to Malcolm Gladwell,

> The success of any kind of social epidemic is heavily dependent on the involvement of people with a particular and rare set of social gifts.

These people include Connectors, mavens and salesmen. The role of connectors is to know the people, the mavens play the role of collecting information of the commodity and the salesmen carry the task of persuading the people to purchase the items [Gladwell (2000)]. In an online marketplace all the three roles are played by the recommendation system at the same time simultaneously. A Recommendation system is an engine which finds out products of interest from a near infinite inventory and recommends them to their potential users [Li *et al.* (2021)]. From above it can be seen that recommendation engine plays the role of connector by knowing the people in order to find a potential user for an item, it also contains the information of products like mavens to suggest interesting items to users and like salesmen it persuades the users to purchase the recommended items. So, in simple terms we can say that the success of online market place it heavily dependent on the recommendation system [Li *et al.* (2021)].

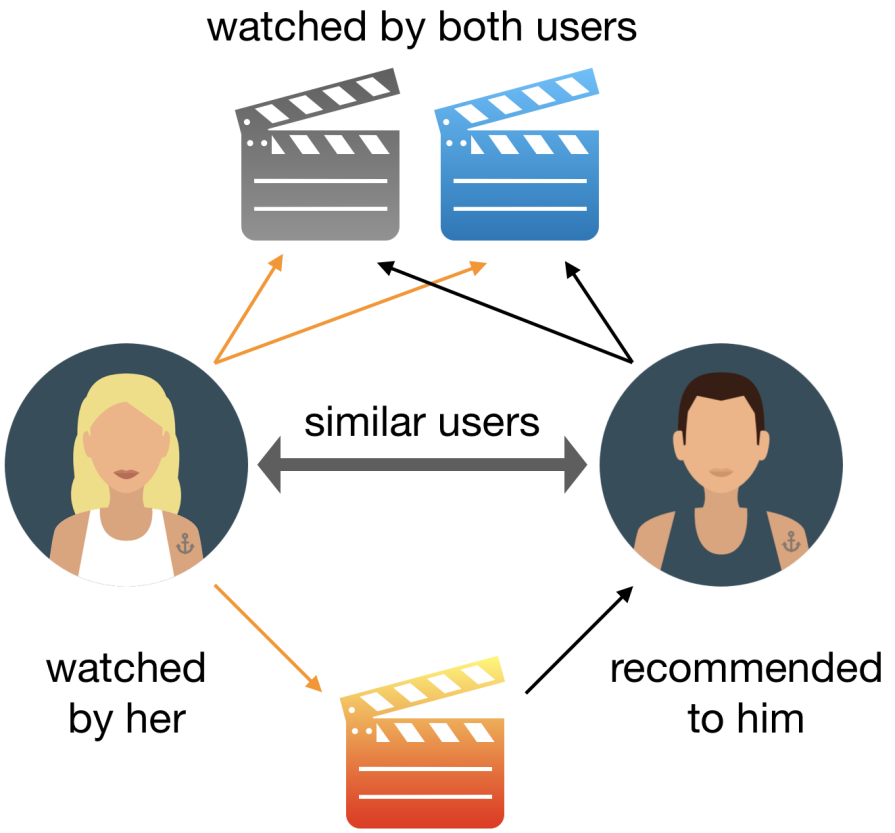


Figure 1.1: Shows a general example of the Recommendation System and the concept. Image credits [Le (2019)].

One of the most successful methods for developing recommendation systems is Collaborative Filtering [Adomavicius and Tuzhilin (2005)]. The recommendation systems that are based on Collaborative Filtering require a large amount of information to make recommendations for the users. Now, this data is available for old users or in terms of recommendations systems *Warm Users*. These are the users whose previous purchase behavior is known. Now, when a new user or *Cold Users* a user whose purchase behavior is unknown joins the platform these collaborative filtering based recommendation systems fail[Li *et al.* (2017)]. This usually faced problem in the field of recommendation systems is called as the Cold-Start recommendation problem [Lin *et al.* (2013)]. The problem of Cold-Start Recommendation can be further classified

in two types

1. User Based Cold Start Recommendation.

2. Item Based Cold Start Recommendation.

The first type of systems can be described as a system in which the user is new and no interactions of the users are available in the target space which means we have no purchase information about the user. The second type of system is the one where the item is new and we have no interactions available in the target space which means we have no user interactions for the item. This is further explained in detail in sec. 2.1.

1.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a generative modelling approach that uses deep learning methods. Generative modelling is an unsupervised machine learning task in which the model can be used to generate new output which looks similar to the examples from the data set by discovering and learning the patterns and regularities that are existing in the data set [Brownlee (2019)]. This technique of GANs helps to convert the unsupervised learning task to a supervised learning problem. This can be achieved by by using two sub-models which include the Generator Model and the Discriminator Model. The task of Generator is to capture the distributions of the data in the ground truth so it can generate synthetic data similar in characteristic to the ground truth data. The generator can be trained either by using the conditioned data or by using random noise. The discriminator is trained to detect whether the sample is from the ground truth or it is generated by the generator.

The training process is a minmax game between Generator and Discriminator: Generator tries to increase the error rate of the Discriminator while Discriminator tries to classify real data from the fake data. As a result of this adversarial training,

Generator learns how to generate realistic data. The objective function of the training is,

$$\max_{\theta} \min_{\phi} \mathsf{V(G,D)} = \mathsf{E}_{x \sim P_{data}}[\log D_\theta(x)] + \mathsf{E}_{\hat{x} \sim G_\phi(x)}[\log(1 - D_\theta(\hat{x}))] \tag{1.1}$$

ϕ and θ are the parameters of Generator and Discriminator. Here, Generator and Discriminator minimize and maximize the same objective function to optimize the model parameters. This process is continued until the generator model is trained enough to generate the samples to fool the discriminator more than half of the time. The discriminator is trained on the type of groundtruth samples from the data set similar to which the generator is expected to produce the output samples. Generative Adversarial Networks are highly used in various machine learning tasks such as image to image translation for generating photo realistic images of objects, people which can be good enough to fool even human beings. This architectures was proposed by Goodfellow *et al.* (2014). The technique of generative modelling has been used extensively since then ranging in different tasking including Computer Vision, Music Generation, recommendation systems.

1.3 Zero-Shot Learning

The goal of Zero Shot Learning is to recognize objects from the unseen classes that the model has not seen during the training process. The two main spaces of zero shot learning is seen and unseen and the data from both the spaces are disjoint. The semantic information or the attributes of the seen classes are used to transfer the knowledge from seen classes to unseen classes. The main aim of Zero shot learning is to develop a model which is able to leverage the knowledge of the seen classes and is able generalize for the unseen classes. The scope of Zero Shot Learning evaluates it generalizability using its performance on the unseen classes.

1.4 Auto Encoders

Auto encoders are a type of neural networks which are used in unsupervised learning for the task of representation learning. It is a network in which a bottle neck is intentionally imposed in order to represent the data in a compressed manner. The input of the network is the unlabelled data $\mathbf{X}$ which is processed through the network to output $\hat{\mathbf{X}}$, which is a reconstruction of the input data $\mathbf{X}$. This network is minimizing the error $\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}})$ during the training. This error measures the difference between the original input and produced reconstruction by the network. Auto encoder can be of different types such as Sparse Auto Encoders, Denoising Auto Encoders, Variational Auto Encoders.

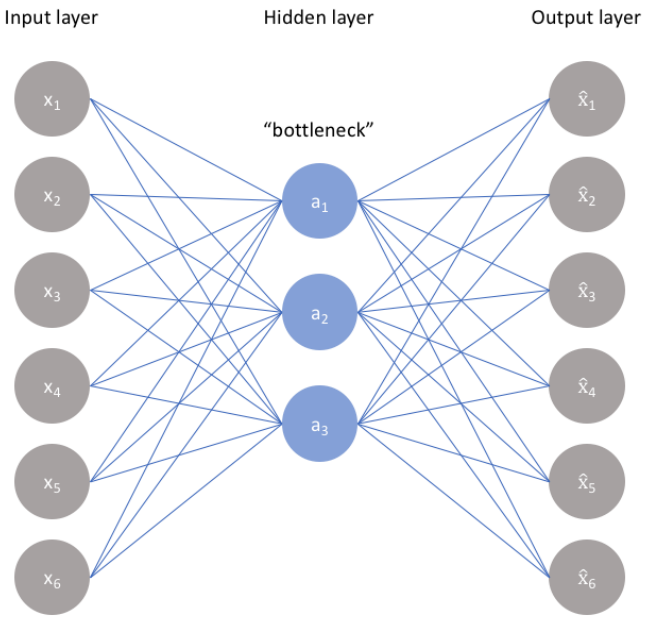


Figure 1.2: Shows the Autoencoder model in which the input x is reconstructed by using the bottleneck representation a and the final output is $\hat{x}$ which is forced to be similar to x. This helps the model to learn a constrained representation of the unlabelled data x with the help of correlations. Image Credits [Jordan (2018)].

1.4.1 Sparse Auto Encoder

Sparse auto encoders focus on penalizing the activation in layers as opposed to reducing the number of neurons. The network learns an encoding and decoding which relies only on a limited number of neurons. This is a different apporoach of regularization where we do not regularize the weights. It is necessary to note that which nodes are activated depends on the data as different set of nodes are activated for different data samples. As compared to the under complete auto encoder which uses the entire network, this uses only a limited numbers of nodes. The sparsity constrain can be applied in two ways L1 Regularization and KL-Divergence. These terms are added to the loss function to penalize the excessive activations.

1.4.2 Denoising Auto Encoder

In denoising auto encoders the the input data is slightly corrupted but the output still expected out of it is the uncorrupted data. Here the input and target output are no longer same so model does not simply develop a mapping which memorizes the training data. Contractive auto encoders are very similar to denoising auto encoders in a sense that the small changes in our input are regarded as noise and our model is robust against noise. Which means that even if there is a small change in the input then too the encoded state remains the same.

1.4.3 Variational Auto Encoder

When the latent space in an autoencoder is regularized enough the decoder can be used for the purpose of generating new samples. This can be achieved by explicit regularization during the training process. We can say in simple terms that a Variational autoencoder is an autoencoder whose decoder can be used for generative

purpose and the latet space is regularized to avoid over fitting. In variational autoencoder the input is encoded as a distribution over the latent space instead of a single point. This is followed by sampling a point from the distribution in latent space. Now the reconstruction error is calculate for the decoded output of the sampled point and lastly the reconstruction error is backpropagated through the network. This is the training procedure of a Variational Autoencoder.

1.5 KL - Divergence

KL-Divergence can be termed as a measure of difference between two probability distributions. A sparsity parameter ρ is defined which acts as the average activation of neuron over a collection of samples.The expected value $\hat{\rho}_m$ can be calculated by using,

$$\hat{\rho}_m = \frac{1}{j}\sum_i [a_i^{(h)}(x)] \tag{1.2}$$

in Eq. 1.2 the specific neuron in layer h is denoted by m, and sums the observations of j training observations which are denoted individually as x and a_i is the activation function. This constraining encourages the neurons to fire only for the subset of observations. The value of ρ can be described as a Bernoulli random variable distribution. The KL-Divergence can be used to compare the ideak distribution ρ with the observed distribution $\hat{\rho}$ over all the nodes of hidden layer. The KL-Divergence between two Bernoulli distributions is written as,

$$\sum_m \mathrm{KL}(\rho||\hat{\rho}_m) = \sum_{m=1}^{l^{(h)}} \rho \log\frac{\rho}{\hat{\rho}_m} + (1-\rho)\log\frac{(1-\rho)}{(1-\hat{\rho}_m)}. \tag{1.3}$$

The value for $\mathrm{KL}(\rho||\hat{\rho}_m) = 0$ when $\hat{\rho}_m = \rho$. On the other hand the value increases monotonically as the values of $\hat{\rho}_m$ diverges from the value of ρ as shown in fig. 1.3.

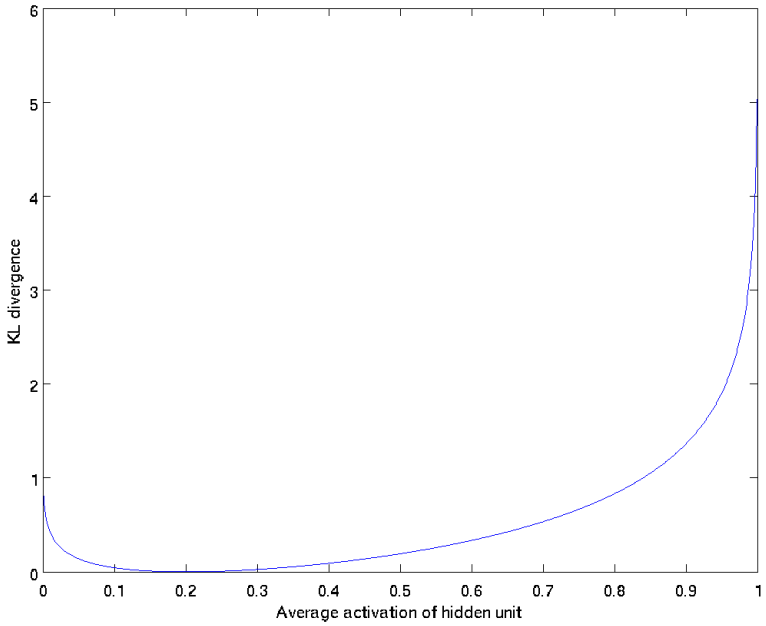


Figure 1.3: Depicts how the value of the penalty $\sum_m \mathrm{KL}(\rho||\hat{\rho}_m)$ behaves for multiple values of $\hat{\rho}_m$. Here the value of $\rho = 0.2$ and it can be seen that $\sum_m \mathrm{KL}(\rho||\hat{\rho}_m) = 0$, when the value of $\hat{\rho}_m = 0$ else it increases monotonically. Image Credits [Ng *et al.* (2011)].

1.6 Goals and Motivation

The goal of this dissertation is to propose a novel Sparse Regularization which can be used to leverage sparsity in User and Item Cold-Start Recommendation and implement a sparse adversarial model using the proposed Sparse regularization to perform the task of User Cold-Start Recommendation. It highlights the role a Recommendation systems plays in online markets and also highlights the importance of recommendation systems in machine learning and deep learning domain. It summarizes the different approaches which are currently used to develop recommendations systems. The future research direction has also been highlighted.

This dissertation has derived its motivation from the existing problems that are currently faced in the domain of Cold-Start Recommendation. There are some lim-

itations that have been observed even after the recent progress in the field. First, we can take into consideration that two person with same demographic information can have different preference over their genre choices but in collaborative filtering only the demographic and previous purchases are used.This leads to recommending same products to users with same demographics even if their preferences are different. The second major concern is the sparsity in the user purchase behavior which is not taken into proper consideration. Another concern is that the user purchase behavior is represented by using implicit feedback only which just states interaction but not actually measure the users reaction over the product. The other major concern is that systems that are based on only collaborative filtering fail to perform for the Cold-Start scenario so a hybrid system that is a combination of Collaborative Filtering and Content-Based Filtering needs to be applied to deal with the Cold-Start problem.

1.7 Contribution

The contributions of this dissertation are as follows.

1. A detailed survey of the existing state of the art methods for Cold-Start Recommendation.

2. TF-IDF based representation of user-vectors containing Demographic Information and preferences.

3. A novel hybrid Cold-Start Recommendation system Combining Collaborative filtering and Content-Based filtering approaches.

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

5. Competitive performance on the popular MovieLens100K and MovieLens1M

datasets.

1.8 Dissertation outline

The dissertation is structured in the following manner.

Chapter 2 provides an overview of Recommendation systems. The first section focuses on the User-based Cold-Start Recommendation system and in detailed description of the system. It includes the different state of art approaches used for the purpose such as Matrix Factorization based approached, Deep Neural Network based approached and Generative Adversarial model based approaches. The third section concentrates on the problem settings for the purpose of this dissertation. The next section discusses about the state of art data sets used for the purpose of this experiment and how they are processed to bring the best representation and results out of them. The next section consists the terminology used for this dissertation to provide an ease of reading and understanding.

Chapter 3 This chapter consists of the proposed SRLGAN model. The second section explains in detail about the proposed model and its architecture as well as the mathematics of the proposed model. It also explains in detail about the advantage of explicit user behavior representation, vector representation of user attributes and the proposed sparse penalty. It also showcases the objective function of the model. The following section explains in details the training algorithm that has been implemented to train the model.The following chapter is broadcasts the various experiments and the results obtained for the proposed model.The next section analyses the model and its effectiveness.

Chapter 4 This chapter concludes the dissertation with a summary of the contributions and also provides for a future line of research work in the domain.

Chapter 2

BACKGROUND AND RELATED WORK

This chapter introduces the problem background, the available data sets and the existing approaches to perform the task of Recommendation. In this chapter the section 2.1 concentrates on the problem of Cold-Start Recommendation. The section 2.2 discusses about the Matrix Factorization based methods for Collaborative Filtering. The following section 2.3 and section 2.4 focuses on the Deep Neural Networks based methods and Generative Adversarial Networks based methods for Collaborative Filtering respectively. The problem settings for the purpose of this dissertation is explained in section 2.5. The section 2.6 describes how the data is transformed for the purpose of this dissertation and the datasets used for experimentation. The last section 2.7 contains a table which describes the terminology used in this dissertation for the ease of reading.

2.1 Recommendation Systems

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 filtering 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

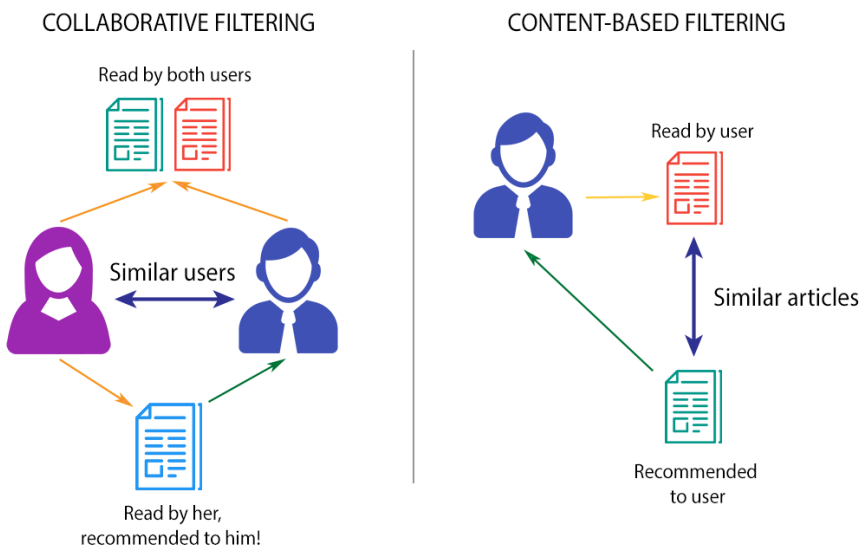


Figure 2.1: Explains the general example of Collaborative Filtering and Content-Based Filtering and how the two approaches are unique. Image credits [Liao (2018)].

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 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. The matrix Factorization based methods are covered in detail in sec. 2.2. 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).

2.2 Matrix Factorization Based Collaborative Filtering

According to Ning and Karypis (2011), Matrix Factorization methods that build models to learn the pattern of the past preferences of the users on items are known to be most successful. These model based methods learn linear interactions between the latent features of the users and the items [Chae *et al.* (2018)]. The popular examples of model based Matrix Factorization methods for the task of ratings predictions are SVD++ [Koren (2008)], BiasedMF [Koren *et al.* (2009)] and PMF [Mnih and Salakhutdinov (2008)]. The prominent Matrix Factorization based methods for performing top-n recommendations are FISM [Kabbur *et al.* (2013)], BPR [Rendle *et al.* (2012)], WRMF [Hu *et al.* (2008)], CLiMF [Shi *et al.* (2012)], PureSVD [Cremonesi *et al.* (2010)], SLIM [Ning and Karypis (2011)].

In BPR Rendle *et al.* (2012), have focused on the optimization of ranking in the task of personalized recommendation. From the Bayesian analysis of this problem they have proposed BPR-OPT which is an optimization criterion that is maximum posterior estimator. Their learning method is based on stochastic gradient decent with bootstrap sampling. In FISM Kabbur *et al.* (2013), have stated that as the sparsity in data increases the model performance decreases for top-n recommendation, to deal with this problem they have proposed an item based method for top-n recommendation. In this method the item-item similarity matrix is learned as a product of two low dimensional latent factor matrix. In this the matrix are learned using the structural equation modelling approach. In this approach the value that is being estimated is not used for its own estimation. Although, these methods are successful there is limitation that is faced in matrix factorization based methods that they learn only the linear interactions between the latent features of items and users interactions and lack ability to learn non-linear features [Hinton and Salakhutdinov (2006)].

2.3 Deep Neural Network Based Collaborative Filtering

In order to overcome the limitation faced in Matrix Factorization based approaches, Deep Neural Network based Collaborative filtering models have been gaining popularity owing to their capabilities of exploiting the non-linear interactions between the latent features of users and items interactions and developing an arbitrary continuous function [LeCun *et al.* (2015); Hinton and Salakhutdinov (2006)]. Some of the popular Deep Neural Networks based Collaborative Filtering models for top-n recommendation are CDAE [Wu *et al.* (2016)] and NCF [HE *et al.* (2017)]. Deep Neural models have also exhibited impressive performance on the rating prediction task and some of the methods include AutoRec [Sedhain *et al.* (2015)] and CDL [Wang *et al.* (2015)].

In NCF HE *et al.* (2017), state that neural architectures can replace the inner products in Matrix Factorization to learn an arbitrary function from the data. They have proposed a generic framework under the name of Neural Collaborative Filtering which can be used to generalize Matrix Factorization. They have proposed a multilayer neural network to learn the non-linear interactions between user and item. In CDAE Wu *et al.* (2016), state that advances in the top-n recommendation have far more consequences in the practical applications as the platforms in real world measure the performance of their recommender engine based on the top-n recommendations shown to the end user. They have proposed a method that uses the concept of Denoising Auto-Encoders for the task of top-n recommendation. They state that their proposed model is a generalization of several renowned collaborative filtering methods but has more flexible components. From the above stated methods it can be seen that neural network based methods have been able to showcase impressive results as compared to the traditional Matrix Factorization based approaches.

2.4 Generative Adversarial Networks Based Collaborative Filtering

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. Some of the popular Methods that are based on Generative Adversarial Networks include IRGAN [Wang *et al.* (2017)], GraphGAN Wang *et al.* (2018), CFGAN [Chae *et al.* (2018)], VAEGAN [Yu *et al.* (2019)].

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.

2.5 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 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 $x \in \mathsf{R}^d$. 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 $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$, and thereby predict the purchase behavior $\{\hat{y}_i^u\}_{i=1}^{n_u}$ of cold users in $\mathcal{X}^u$.

2.6 Data Sets

The performance of the model is evaluated on 2 State of the art data sets namely MovieLens 100k, MovieLens 1M [Harper and Konstan (2015)]. For the purpose of this experiment the data set is being divided into two files the user vector which consists of the user attributes consisting of the side information and the preferences of the user. The other file is the user behavior representation.

2.6.1 MovieLens 100K

The data set consists of a total of 943 unique users. The total number of unique items that a user can purchase and rate are 1682. The total number of attributes in which a user can be represented is 103. The user vector consists of the Demographic Information of the user such as Age, Gender, Occupation and the 19 genres that are defined by Harper and Konstan (2015) in the dataset. This indicates that the dimension of the user vector is 103. The user vector is a weighted representation of the attributes and its calculation is defined in section 2.6.3. The total number of unique movies that any user can see and rate is 1682 which the dimension of user purchase behavior vector. More details about the user purchase behavior vector are mentioned in section 2.6.4

2.6.2 MovieLens 1M

The data set consists of a total of 6040 unique users. The total number of unique items that a user can purchase and rate are 3952. The total number of attributes in which a user can be represented is 48. The user vector consists of the Demographic Information of the user such as Age, Gender, Occupation and the 19 genres that are defined by Harper and Konstan (2015) in the dataset. This indicates that the

dimension of the user vector is 48. The user vector is a weighted representation of the attributes and its calculation is defined in section 2.6.3. The total number of unique movies that any user can see and rate is 3952 which is the dimension user purchase behavior vector. More details about the user purchase behavior vector are mentioned in section 2.6.4

2.6.3 User Vector Representation Calculations

The User Vector consists of the data associated with the user which gives us some details about the user. Now, not all details provided about the user are important and play a major role in the user's behavior. In order to determine the factors that play a major role in the user's behavior and are dominant I proposed to represent the user's attributes in the form of Term Frequency - Inverse Document Frequency (TF-IDF) vectors. This representation will help in giving a higher weighted representation to the factors that are dominant, helping our model to learn more about the user. This TF-IDF vector consists of all the possible attributes that the users can have. In this method the term frequency defines the number of times a particular term is repeated in a document. In order to reduce the importance of common words and highlight the dominant terms inverse document frequency is applied. The final Vector is the product of the Term Frequency vector for that user and the Inverse Document Frequency vector for that user. In this algorithm the dominant factors are given higher scores as compared to the common factors. The formula for calculation of the `TF-IDF scores` is given below.

$$\texttt{TF} = \frac{\texttt{Number of repetitions for a term for each user}}{\texttt{Total number of terms for each user}}$$

$$\texttt{IDF} = \log\left(\frac{\texttt{Total Number of users}}{\texttt{Number of users Containing the Term}}\right)$$

$$\texttt{TF-IDF Score} = \texttt{TF} \times \texttt{IDF}$$

2.6.4 Explicit User Behavior Representation

The user behavior can be represented in two ways ***Implicit*** which represents the user's interaction with the product such as purchase or click and ***Explicit*** which mostly is in terms of ratings given by the user's for the product. It is observed from previous research works that more attention is given to the implicit representation as compared to the explicit form. I believe as the explicit form consists of the ratings given by the user for the products they have interacted with it is the indication of the appreciation for the product by the user. If a user has given 5 rating to the product on a scale of 0 - 5 it indicates that the user has highly liked the product and a rating of 0 indicates that the user did not like the product or the user has not purchased the product. This plays a major role as we are not only considering if the user has interacted with the product or not but we are also taking into consideration that after purchasing the product the user actually likes it or not. This will help us in making recommendations to the new users as it is not always necessary that if the user has purchased it the user might like it. It is also possible that the user has purchased or viewed the product but he might not actually like it after using it so might have given a lower rating to the product. So it is better to recommend a product to the customer which has been rated highly after being purchased by the user with similar attributes as if we recommend a product which has higher appreciation then the new user might also like it and purchase it. This will help us in increasing the customer loyalty as the recommendations will also include niche products which are purchased by users with similar attributes and are rated highly which the new user might actually like and purchase it increasing our sales and loyalty of the customers.

2.7 Terminology

The table 2.1 consists a list of all the notations and their meanings for the ease of reading this dissertation.

Table 2.1: Terminology Used in This Dissertation

Notation	Description
$\mathcal{X}^s$	Warm User Space
x^s	Warm User Vector
n_s	Number of Warm Users
m	Total number of items available to users
C	Highest rating a user can give to a product.
$\mathcal{Y}^s$	Ground Truth User Purchase behavior for warm users
y^s	User Purchase Behavior Vector
$\mathcal{X}^u$	Cold User Space
x^u	Cold User Vector
n_u	Number of Cold Users
$\hat{y}^u$	Generated User Purchase Behavior
$\mathcal{X}$	User attributes space
$\mathcal{Y}$	User Purchase behavior space
G_{θ_g}	Conditional Generator Function
θ_g	Parameters of Generator
$\mathcal{L}_G$	Objective Function of User Purchase Behavior Generator
D_{θ_d}	Discriminator Function
θ_d	Discriminator parameters
$\mathcal{L}_D$	Objective Function of User Purchase Behavior Discriminator
ρ	Average Ground truth Purchase behavior
$\hat{\rho}$	Average Predicted Purchase Behavior
b	Batch Size
$\mathcal{L}_{SR}$	Objective Function of Sparse Regularization
d	Dimension of User Attribute Vector

Chapter 3

PROPOSED SRLGAN MODEL FOR COLD-START RECOMMENDATION

Recently, Generative Adversarial Networks was proposed by Goodfellow *et al.* (2014) in which two models are trained simultaneously by playing a minmax game. These two models are a Generator and a Discriminator. The task of Generator is to capture the distributions of the data in the ground truth so it can generate synthetic data similar in characteristic to the ground truth data. The task of the Discriminator is to Distinguish the Generated data from the Ground truth data. Here, Generator and Discriminator minimize and maximize the same objective function to optimize the model parameters. The Generative Adversarial Networks have achieved great results in domains such as Image Generation, Music Generation and Sentence Generation.

Owing to the Success of GANs in other domains the concepts of GANs have also been applied to the domain of Recommendation Systems for predicting the rating given by the user and also for the task of top-n recommendations. These models are based on Collaborative filtering and use the user's past behavior and context [Bobadilla *et al.* (2013); Li *et al.* (2017)] to make personalized recommendations. These systems have shown impressive results for warm users whose purchase behavior is available. When the purchase behavior is not available for Cold users these systems get stuck [Li *et al.* (2017)]. There are various solutions proposed for this problem of recommendation which is called as Cold-Start Recommendation [Lin *et al.* (2013)].

Some of the proposed methods for Cold-Start Recommendation suggest using cross-domain information, personal information and social-network information of the user [Fernández-Tobías *et al.* (2012); Fernández-Tobías *et al.* (2016); Sedhain *et al.* (2017)]. From this we can observe that the basic idea behind the cold star methods

is to *Leverage the preferences to generate recommendations for new users* [Lin *et al.* (2013); Li *et al.* (2017)]. This logic is quite reasonable as we can barely recommend something to a person we barely know and the chances of the person liking it are also low. So, by the above idea we have two space a user attribute space and a user behavior space. This attribute space can be defined as a space containing side information about the users such as age, Gender, Occupation, Zip, etc which can be used to relate the new user to the previous users as mentioned in previous chapters. According to Li *et al.* (2019), the attributes of warm users and the cold user and the behaviour of the warm user can be utilized to generate a behavior for the cold user by using the hypothesis *People with similar attributes tend to have a similar purchase behavior*. Utilizing this hypothesis the cold start recommendation can be done in two steps using which they have coined it as a special case of Zero-Shot Learning.

1. Mapping the behavior space to attribute space so that the new users can be linked with the old users.

2. Reconstructing user behavior by user attributes, so that we can generate recommendations for new users.

The limitation here is that 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)]. This was the first major concern for cold-start recommendation. 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 Second major concern is that implicit feedback is given more preference as compared to explicit feedback of the users[Chae *et al.* (2018)]. The problem with this is the implicit feedback only represents the interaction with the item but not the actual experience of the user. On the other hand if explicit feedback is used we can

gauge the actual experience of the users over the item from the ratings given by the users which can be treated as a weighted score of the users interaction and experience. Another major concern that exists is the above models do not take into account the sparsity in the user-purchase behavior they are prone to over-fitting on warm users and instability during training.

The following chapter is organized as following: The section 3.1 gives a short overview of our proposed SRLGAN model. The following section 3.2 contains detailed information of the Adversarial user purchase behavior generation process and the components such as the User Purchase Behavior Generator, User Purchase Behavior Discriminator, the proposed Sparse Regularization and the overall objective function of the SRLGAN. The following section 3.3 explains the algorithm used for training of SRLGAN. The section 3.4 focuses on the Datasets used, the details on the implementation of SRLGAN, the different evaluation matrices used for testing the performance of SRLGAN, the details of the model architeecture of SRLGAN, the experiments conducted with SRLGAN and the comparision with baselines and also the sensitivity to hyperameter and the ablation study based on the different components of SRLGAN and to check how they contribute to the success of SRLGAN.

3.1 SRLGAN: A Novel Sparse Adversarial Model for Cold-Start Recommendation

In SRLGAN 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 and develop a hybrid recommendation system. We model

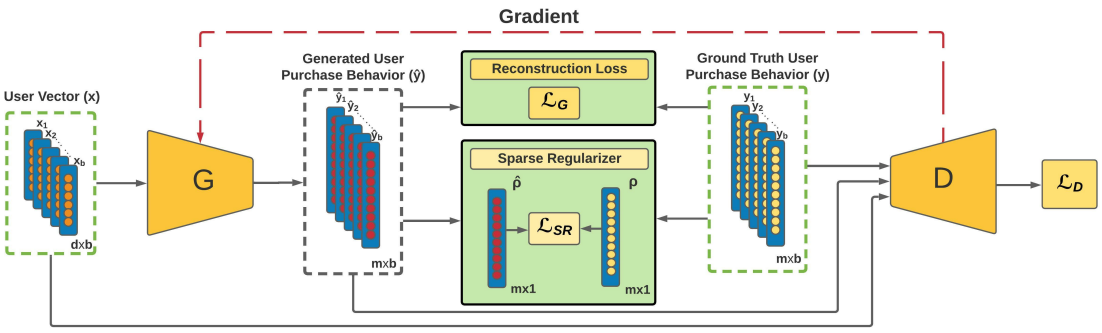


Figure 3.1: System diagram of the SRLGAN. The input to the Generator module are user attributes of dimensions $(d \times b)$ and the output of the Generator are the predicted user purchase behaviors of dimensions $(m \times b)$. The objective term $\mathcal{L}_G$ (Eq. 3.1), is a reconstruction loss to 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. 3.2. 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. 3.4), by minimizing the KL-divergence between the average generated (predicted) user purchase prediction $\hat{\rho}$ and the average ground truth user purchase prediction ρ.

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. For using this regularization the user-purchase behavior distribution is calculated from the warm users purchase behavior. This user-purchase distribution is used to model the sparsity using the proposed sparse regularization which is inspired from KL-Divergence and Sparse Auto-Encoder [Ng *et al.* (2011)]. 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.

3.2 Proposed Approach

3.2.1 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.

3.2.1.1 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,

$$\mathcal{L}_G = \min_{\theta_g} \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i^s - G_{\theta_g}(x_i^s))^2. \tag{3.1}$$

3.2.1.2 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,

$$\mathcal{L}_D = \min_{\theta_g} \max_{\theta_d} \frac{1}{2}\mathsf{E}_{(x,y)\sim(\mathcal{X}^s,\mathcal{Y}^s)}\left[(D_{\theta_d}(x,y)-1)^2\right] + \frac{1}{2}\mathsf{E}_{x\sim\mathcal{X}^s}\left[(D_{\theta_d}(x,G_{\theta_g}(x))^2\right]. \quad (3.2)$$

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.

3.2.2 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 *et al.* (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,

$$\rho = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i^s. \quad (3.3)$$

$\rho = [\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(m)}]^\top$ 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 $\mathrm{KL}(\rho^{(i)}||\hat{\rho}^{(i)})$, 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,

$$\begin{aligned}\mathcal{L}_{SR} &= \sum_{i=1}^{m} \mathrm{KL}(\rho^{(i)}||\hat{\rho}^{(i)}) \\ &= \sum_{i=1}^{m} \rho^{(i)} \log \frac{\rho^{(i)}}{\hat{\rho}^{(i)}} + (1-\rho^{(i)}) \log \frac{(1-\rho^{(i)})}{(1-\hat{\rho}^{(i)})}. \end{aligned} \tag{3.4}$$

The divergence $\mathrm{KL}(\rho^{(i)}||\hat{\rho}^{(i)}) = 0$ 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.

3.2.3 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. 3.1). The Discriminator and the Generator are trained using the adversarial Least Squares objective (see Eq. 3.2). The SRLGAN leverages the sparsity in the user purchase behaviour distribution to train the Generator and avoid over fitting to warm users (see Eq. 3.4). The overall objective function of the SRLGAN is,

$$\mathcal{L}_G + \mathcal{L}_D + \beta \mathcal{L}_{SR}, \tag{3.5}$$

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. 3.1.

3.3 Training Algorithm

The training procedure for the SRLGAN model is illustrated below in Algorithm 1. Mainly the Generator (G_{θ_g}) and the Discriminator D_{θ_d} are trained alternatively using the Adam Optimizer. The learning rate was set to 10^{-6}.

Algorithm 1 Training the SRLGAN

Input: $x^s \in \mathcal{X}^s$, $y^s \in \mathcal{Y}^s$, ρ

Constants: β, b, n_e, n_G, n_D

Output: G_{θ_g}

Initialize $G_{\theta_g}$and D_{θ_d}
for n_e iterations **do**
    Update G_{θ_g} using $\{x_i^s, y_i^s\}_{i=1}^b$ with Eq. 3.1
while Not Converged **do**
    for n_D iterations **do**
        Generate $\hat{y}_i \leftarrow G_{\theta_g}(x_i^s)$ using $\{x_i^s, y_i^s\}_{i=1}^b$
        Update D_{θ_d} and G_{θ_g} using Eq. 3.2
    for n_G iterations **do**
        Generate $\hat{y}_i \leftarrow G_{\theta_g}(x_i^s)$ using $\{x_i^s, y_i^s\}_{i=1}^b$
        calculate $\hat{\rho}$ using Eq. 3.3 & Update G_{θ_g} using Eq. 3.5
Return G_{θ_g}

3.4 Experiments and Analysis

3.4.1 Data Sets

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 3.1. 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.

Table 3.1: Details of the Datasets. The % Sparsity is the percentage of 0s in the user purchase behavior space $\mathcal{Y}$.

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

3.4.2 Implementation Details and Performance Metrics

3.4.2.1 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 for n at 5 and 20.

Precision: It is the calculation of how many recommendations the model recommends accurately from the list of top n products considered. It is very Important to have a good precision score as if we are able to recommend more products the user might actually like then we will have a chance of increasing our sales and loyalty. The calculation of `Precision` is performed as,

$$\texttt{Precision} = \frac{\texttt{Number of correct products}}{\texttt{Number of top products}}$$

Here, `Number of correct products` indicates the count of correct products that are present in our generated purchase behavior and `Number of top products` indicates the top-n products in the recommendation that we are considering.

Normalized Discounted Cumulative Gain (NDCG): This is based on the assumption that a highly relevant product is more useful than a product that is mod-

erately relevant but is also useful as compared to the product that is completely irrelevant. The NDCG has a proper upper and lower bound which helps in taking mean across all the recommendations score in order to report a final score. In order to calculate the NDCG score we have to calculate two parameters.

1. Discounted Cumulative Gain(DCG) of the recommended order.

2. DCG of the ideal order(iDCG).

NDCG is the ratio of DCG of recommended order to the DCG of ideal orderiDCG. The calculations for DCG and NDCG are performed as,

$$\mathtt{DCG} = \sum_{i=1}^{n} \frac{\mathtt{relevance}}{\log_2(i+1)}$$

$$\mathtt{NDCG} = \frac{\mathtt{DCG}}{\mathtt{iDCG}}$$

Mean Reciprocal Rank (MRR): It is the reciprocal for the rank of the 1st item that our model predicts correctly. It is also called as average reciprocal rank. We take the 1st item our model recommends correctly find its rank in the recommended list and repeat the same procedure for all the test data and it is averaged which gives us the ranking accuracy of our model. The MRR is calculated as,

$$\mathtt{MRR} = \frac{1}{\mathtt{N}} \sum_{i=1}^{\mathtt{N}} \frac{1}{\mathtt{Rank\ of\ the}\ 1^{st}\ \mathtt{correct\ predicted\ product\ for}\ i^{th}\ \mathtt{user}}$$

Here, N is the total number of samples in that batch.

3.4.2.2 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.

3.4.2.3 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, $[m + d, 2048, 512, 128, 1]$. The Weights were initialized by using the Kaiming-He weight initialization technique[He *et al.* (2015)]. The mode for the initialization was set to *Fan in* as the weights were created implicitly from the previous linear layers as compared to externally generating the matrix for *Fan out* mode. 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.

3.4.3 Experiments

3.4.3.1 Experiment 1: Model with Binary Cross Entropy as Discriminator loss

In this experiment the Objective function of the Discriminator was set to be Binary cross entropy. It was used to calculate the distance between the Generated output and the original ground truth for the Old users. The reconstructions loss as defined in Eq. 3.1 was also used. The results of the performance evaluation are as shown in table 3.2

Table 3.2: Shows the performance of the model in Experiment 1 with Binary Cross Entropy loss for Discriminator and a Reconstruction loss as in 3.1

Datasets	Precision	NDCG	MRR
MovieLens 100K	0.12	0.1	0.17
MovieLens 1M	0.2	0.26	0.56

3.4.3.2 Experiment 2: Model with Least Squares loss for Discriminator

Table 3.3: Shows the performance of the model in Experiment 2 with Least Squares Loss for Discriminator and a Reconstruction loss as in 3.1

Datasets	Precision	NDCG	MRR
MovieLens 100K	0.375	0.4	0.62
MovieLens 1M	0.386	0.37	0.53

In this model the objective function of the discriminator is replaced with the Least Squares Loss EQ. 3.1. The advantage of this objective function is that it provides a strong gradient to the generator to train if the produces output is far from the actual output [Mao *et al.* (2017)]. THis was combined with the reconstruction loss to train the model. The results of this model are as shown in table 3.3

3.4.3.3 Experiment 3: Model with Sparse Regularization and Least Squares Loss

In this experiment the Model in section 3.4.3.2, was updated with a Sparse Regularization as in Eq. 3.4 and the over objective function of the model is Eq. 3.5. This is the SRLGAN model and contains all its components. The performance of this model is shown in detail in table 3.4. The detailed results of this experiment and its comparison with the baselines are as mentioned in table 3.5.

Table 3.4: Shows the performance of the model in Experiment 3 where the objective function of the model is of the SRLGAN as in Eq. 3.5

Datasets	Precision	NDCG	MRR
MovieLens 100K	0.521	0.53	0.69
MovieLens 1M	0.499	0.504	0.674

3.4.4 Results and Analysis

3.4.4.1 Comparison Against Baselines

Table 3.5 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 MovieLens 1M compared to the best model (CFGAN).

Table 3.5: 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.

	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**

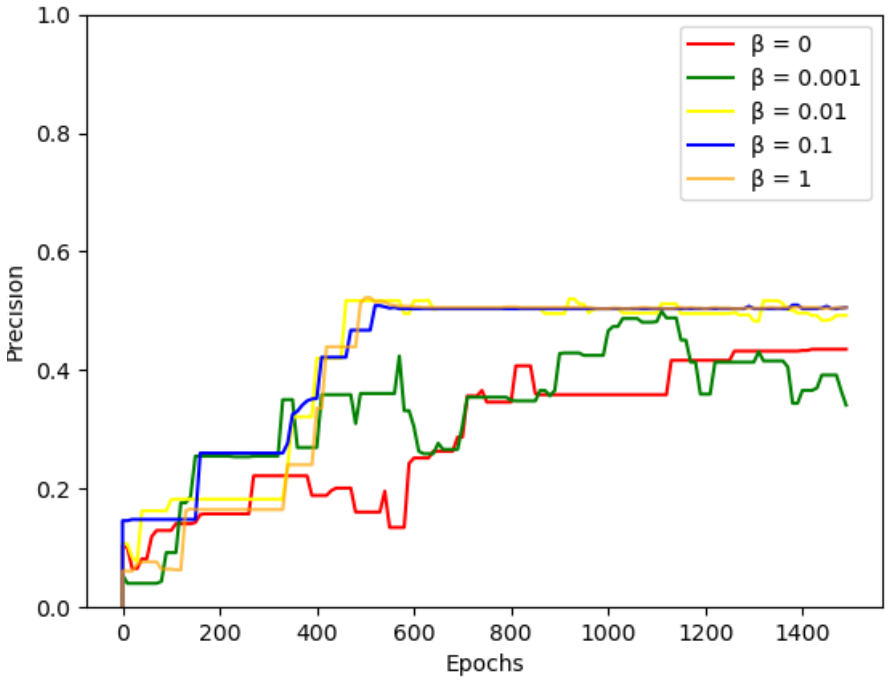
Similarly, the SRLGAN gains 52% and 41.1% for MovieLens 100K and MovieLens 1M, respectively, when predicting the top-20 recommendations (P@20). The values in bold in Table 3.5 demonstrate that the SRLGAN outperforms the other approaches by significant margins across all three metrics for Cold Start recommendation.

3.4.4.2 Hyper-parameter Sensitivity

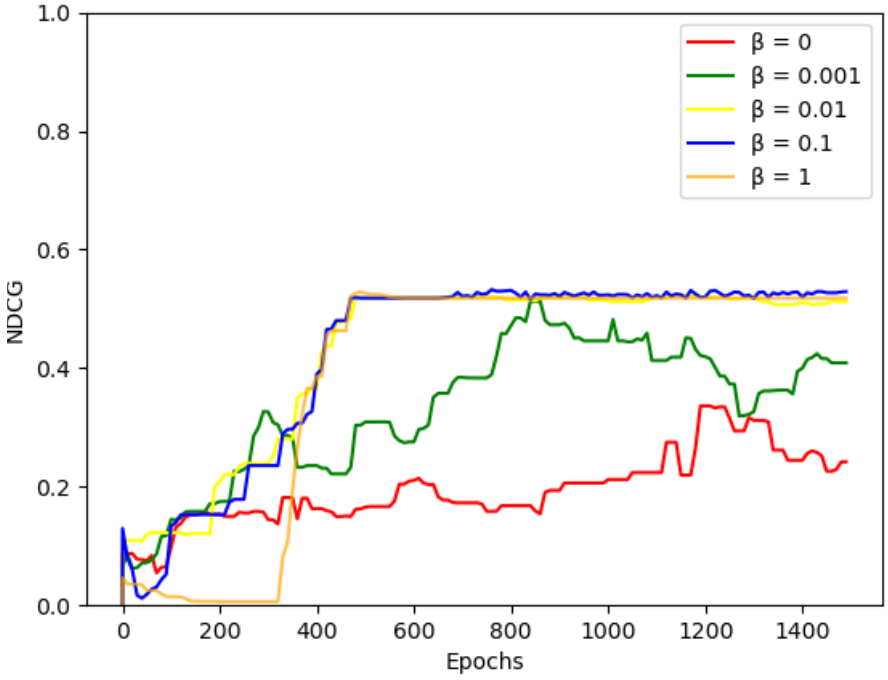
The β hyper-parameter controls the importance of sparsity regularization. We estimate the optimal value of $\beta = 0.1$ using cross-validation by predicting the purchase behavior of 10% of warm users in the training set after training with the remaining users. Fig. 3.2 plots the evolution of Precision and the NDCG scores for MovieLens 100K and Fig. 3.3 for MovieLens 1M for different values of β.

3.4.4.3 Ablation Study

We also report an ablation study of the SRLGAN in Fig. 3.4, Fig. 3.5 and Fig. 3.6 to estimate the importance of different components in the SRLGAN. We treat the S1 model as the baseline whose results are as mentioned in table3.2. Here the loss function driving the training is the standard binary cross entropy (BCE) loss for the GAN and a reconstruction loss (Eq. 3.1) to train the Generator. In the S2 model we replace the BCE loss with the least squares loss (Eq. 3.2) and the results are mentioned in table 3.3. Model S3 is model S2 along with the sparse regularization (Eq. 3.4) and the results are mentioned in table 3.4 and table 3.5. The plots in Fig. 3.4, Fig. 3.5 and Fig. 3.6 depict a steady improvement in Precision, NDCG and MRR respectively with the introduction of each of the components across both the datasets.

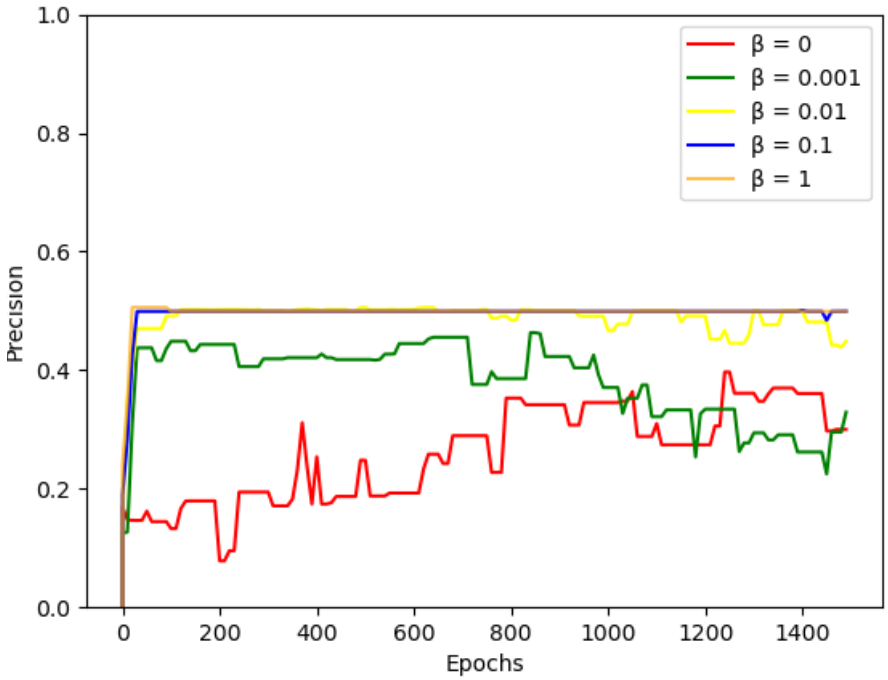


(a)

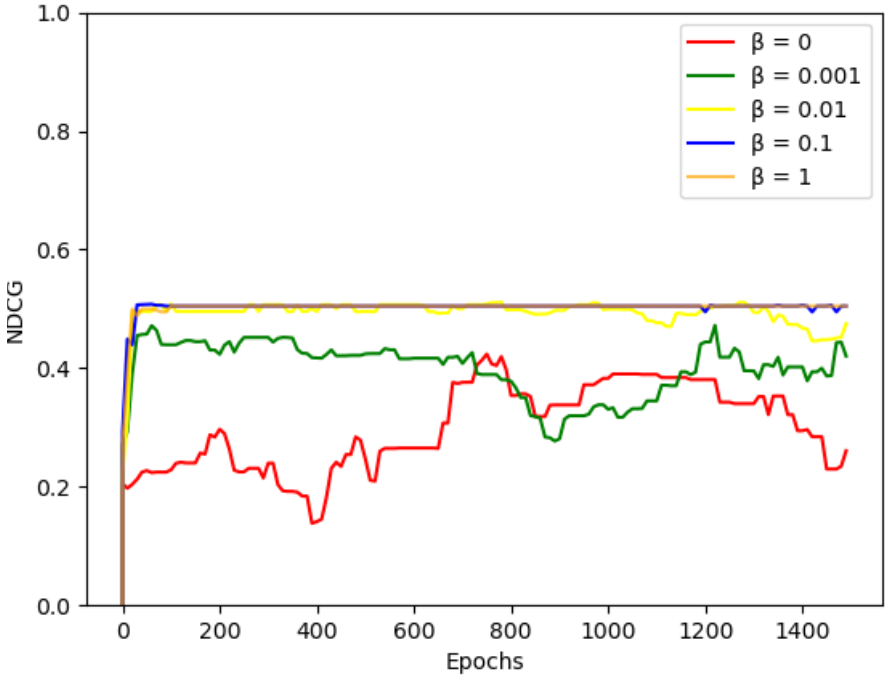


(b)

Figure 3.2: Depicts the training curves for Precision and NDCG as the value of β is varied for the MovieLens 100K dataset. Fig. 3.2a shows the precision curves and ,Fig. 3.2b shows the NDCG curves. $\beta = 0.1$ produces the best results.



(a)



(b)

Figure 3.3: Depicts the training curves for Precision and NDCG as the value of β is varied for the MovieLens 1M dataset. Fig. 3.3a shows the precision curves and ,Fig. 3.3b shows the NDCG curves. $\beta = 0.1$ produces the best results.

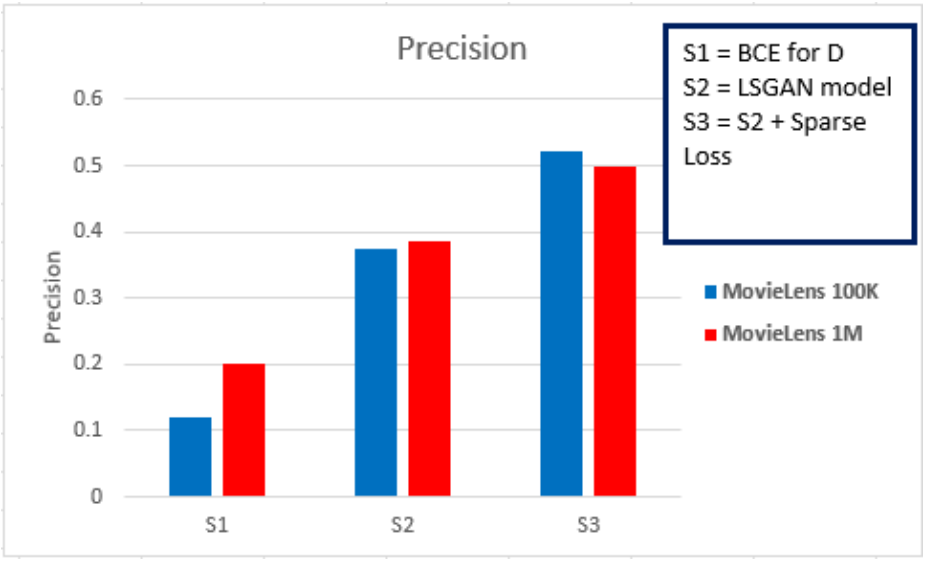


Figure 3.4: Depicts the ablation results for three different models S1, S2 and S2, each having different components of the SRLGAN. This figure depicts the variation in Precision scores across the three models for both the datasets.

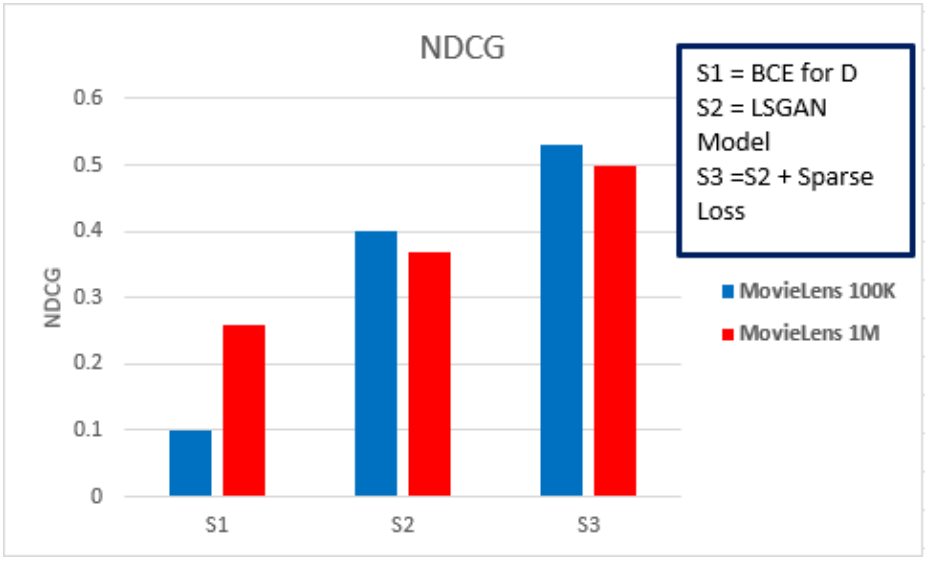


Figure 3.5: Depicts the ablation results for three different models S1, S2 and S2, each having different components of the SRLGAN. This figure depicts the variation in NDCG scores across the three models for both the datasets.

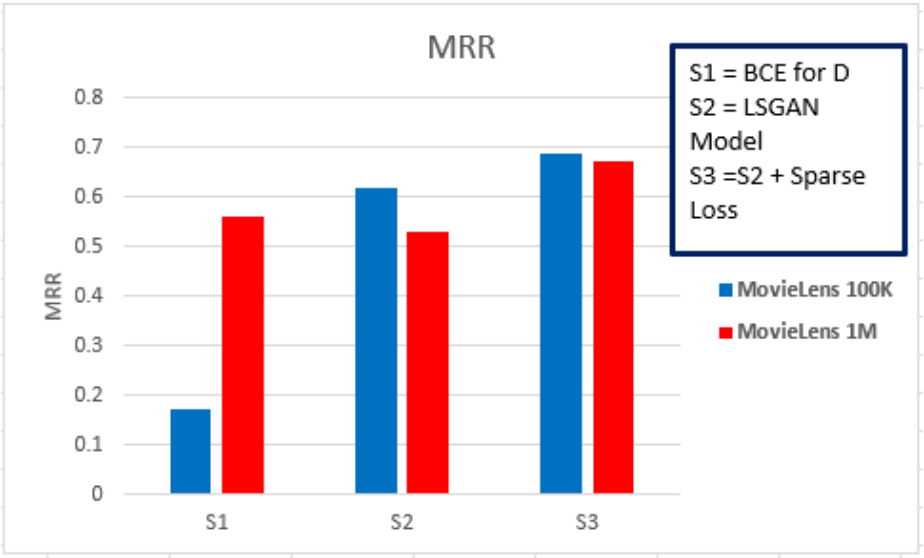


Figure 3.6: Depicts the ablation results for three different models S1, S2 and S2, each having different components of the SRLGAN. This figure depicts the variation in MRR scores across the three models for both the datasets.

Chapter 4

CONCLUSION AND FUTURE RESEARCH WORK

4.1 Conclusion

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.

4.2 Future Research Work

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. We would also like to explore further in the problem of Cross-Domain Cold-Start recommendation using the Similarity Loss.

BIBLIOGRAPHY

Adomavicius, G. and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions", IEEE Trans. KDE **17** (2005).

Bharadhwaj, H., H. Park and B. Y. Lim, "Recgan: Recurrent generative adversarial networks for recommendation systems", ACM Conference on Recommender System RecSys URL https://dl.acm.org/doi/10.1145/3240323.3240383 (2018).

Bobadilla, J., F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey", Knowledge-Based Systems **46** (2013).

Brownlee, J., "A genle introduction to generative adversarial networks (GANs)", URL https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/ (2019).

Chae, D.-K., J.-S. Kang, S.-W. Kim and J.-T. Lee, "Cfgan: A generic collaborative filtering framework based on generative adversarial networks", CIKM URL https://dl.acm.org/doi/10.1145/3269206.3271743 (2018).

Chae, D.-K., J. A. Shin and S.-W. Kim, "Collaborative adversarial autoencoders: An effective collaborative filtering model under the gan framework", IEEE Access URL https://ieeexplore.ieee.org/document/8669749 (2019).

Chen, Y., Y. Dai, X. Han, Y. Ge, H. Yin and P. Li, "Dig users' intentions via attention flow network for personalized recommendation", Information Sciences URL https://doi.org/10.1016/j.ins.2020.09.007 (2021).

Cheng, H.-T., L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu and H. 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), URL https://doi.org/10.1145/2988450.2988454.

Covington, P., J. Adams and E. Sargin, "Deep neural networks for youtube recommendations", ACM Conference on Recommender Systems RecSys URL https://dl.acm.org/doi/10.1145/2959100.2959190 (2016).

Cremonesi, P., Y. Koren and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks", in "Proceedings of the Fourth ACM Conference on Recommender Systems", RecSys '10, p. 39–46 (Association for Computing Machinery, New York, NY, USA, 2010), URL https://doi.org/10.1145/1864708.1864721.

Deldjoo, Y., T. D. Noia and F. A. Merra, "A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks", ACM Computing Survey URL https://dl.acm.org/doi/abs/10.1145/3439729 (2021).

Feng, L., Q. Zhao and C. Zhou, "Improving performance of top-n recommendations with co-clustering method", Expert Systems With Applications URL https://www.sciencedirect.com/science/article/abs/pii/S095741741930795X?via%3Dihub (2019).

Fernández-Tobías, I., M. Braunhofer, M. Elahi, F. Ricci and I. Cantador, "Alleviating the New User Problem in Collaborative Filtering by Exploiting Personality Information", User Modeling and User-Adapted Interaction **26**, URL https://doi.org/10.1007/s11257-016-9172-z (2016).

Fernández-Tobías, I., I. Cantador, M. Kaminskas and F. Ricci, "Cross-domain recommender systems: A survey of the state of the art", Proceedings of the 2nd Spanish Conf. on Information Retrieval (2012).

Gao, M., J. Zhang, J. Yu, J. Li, J. Wen and Q. Xiong, "Recommender systems based on generative adversarial networks: A problem-driven perspective", Information Sciences URL https://doi.org/10.1016/j.ins.2020.09.013 (2020).

Gladwell, M., "The Tipping Point: how little things can make a big difference.", (2000).

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks", Neurips URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf (2014).

Harper, F. M. and J. A. Konstan, "The MovieLens Datasets: History and Context", ACM Trans. Interact. Intell. Syst. **5**, URL https://doi.org/10.1145/2827872 (2015).

He, K., X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", (2015).

HE, X., L. Liao, H. Zhang, L. Nie, X. Hu and T.-S. Chua, "Neural collaborative filtering", WWW URL http://dx.doi.org/10.1145/3038912.3052569 (2017).

Hinton, G. E. and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", Science **313** (2006).

Hu, Y., Y. Koren and C. Volinsky, "Collaborative filtering for implicit feedback datasets", in "2008 Eighth IEEE International Conference on Data Mining", pp. 263–272 (2008).

Jordan, J., "Introduction to autoencoders", URL https://www.jeremyjordan.me/autoencoders/ (2018).

Kabbur, S., X. Ning and G. Karypis, "Fism: Factored Item Similarity Models for Top-N Recommender Systems", in "Proceedings of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining", (2013), URL https://doi.org/10.1145/2487575.2487589.

Koren, Y., "Factorization meets the neighborhood: A multifaceted collaborative filtering model", in "Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", KDD '08, p. 426–434 (Association for Computing Machinery, 2008), URL https://doi.org/10.1145/1401890.1401944.

Koren, Y., R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems", Computer **42**, 8, 30–37 (2009).

Kouki, P., S. Fakhraei, J. Foulds, M. Eirinaki and L. 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), URL https://doi.org/10.1145/2792838.2800175.

Krishnan, A., A. Sharma, A. Sankar and H. Sundaram, "An adversarial approach to improve long-tail performance neural collaborative filtering", ACM International Conference on Information and Knowledge Management CIKM URL https://dl.acm.org/doi/10.1145/3269206.3269264 (2018).

Le, J., "Recommendation system series part 1: An executive guide to building recommendation system", URL https://towardsdatascience.com/recommendation-system-series-part-1-an-executive-guide-to-\building-recommendation-system-608f83e2630a (2019).

LeCun, Y., Y. Bengio and G. Hinton, "Deep learning", (2015).

Lee, H., J. Im, S. Jang, H. Cho and S. Chung, "MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation", (2019).

Li, J., M. Jing, K. Lu, L. Zhu, Y. Yang and Z. Huang, "From zero-shot learning to cold-start recommendation", Proceedings of the AAAI Conf. on Artificial Intelligence URL https://arxiv.org/abs/1906.08511 (2019).

Li, J., K. Lu, Z. Huang and H. T. Shen, "Two Birds One Stone: On Both Cold-Start and Long-Tail Recommendation", in "Proceedings of the 25th ACM Int. Conf. on Multimedia", (2017), URL https://doi.org/10.1145/3123266.3123316.

Li, J., K. Lu, Z. Huang and H. Tao Shen, "On both cold-start and long-tail recommendation with social data", IEEE Transaction on Knowledge and Data Engineering URL https://ieeexplore.ieee.org/document/8745499 (2021).

Li, R., F. Qian, X. Du, S. Zhao and Y. Zhang, "A collaborative filtering recommendation framework based on wasserstein gan", Journal of Physics: Conference Series URL https://iopscience.iop.org/article/10.1088/1742-6596/1684/1/012057 (2020).

Liang, T., C. Xia, Y. Yin and P. S. Yu, "Joint training capsule network for cold start recommendation", SIGIR URL https://arxiv.org/abs/2005.11467 (2020).

Liao, K., "Prototyping a recommender system step by step part 1: Knn item-based collaborative filtering", (2018).

Lin, J., K. Sugiyama, M.-Y. Kan and T.-S. Chua, "Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers", in "Proceedings of the 36th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval", (2013), URL https://doi.org/10.1145/2484028.2484035.

Liu, J., W. Pan and Z. Ming, "Cofigan: Collaborative filtering by generative and discriminative training for one-class recommendation", Knowledge-based Systems URL https://doi.org/10.1016/j.knosys.2019.105255 (2020).

Mao, X., Q. Li, H. Xie, R. Y. K. Lau, Z. Wang and S. P. Smolley, "Least Squares Generative Adversarial Networks", (2017).

Mnih, A. and R. R. Salakhutdinov, "Probabilistic matrix factorization", in "Advances in Neural Information Processing Systems", edited by J. Platt, D. Koller, Y. Singer and S. Roweis, vol. 20 (Curran Associates, Inc., 2008), URL https://proceedings.neurips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

Ng, A. *et al.*, "Sparse autoencoder", CS294A Lecture notes **72**, 2011 (2011).

Ning, X. and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems", 2011 IEEE 11th International Conference on Data Mining pp. 497–506 (2011).

Perera, D. and R. Zimmermann, "Cngan: Generative adversarial networks for cross-network user preference generation for non-overlapped users", World Wide Web Conference WWW'19 URL https://dl.acm.org/doi/10.1145/3308558.3313733 (2019).

Rendle, S., C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback", (2012).

Sedhain, S., A. Menon, S. Sanner, L. Xie and D. Braziunas, "Low-Rank Linear Cold-Start Recommendation from Social Data", Proceedings of the AAAI Conf. on AI **31**, URL https://ojs.aaai.org/index.php/AAAI/article/view/10758 (2017).

Sedhain, S., A. K. Menon, S. Sanner and L. Xie, "Autorec: Autoencoders meet collaborative filtering", in "Proceedings of the 24th International Conference on World Wide Web", WWW '15 Companion, p. 111–112 (Association for Computing Machinery, New York, NY, USA, 2015), URL https://doi.org/10.1145/2740908.2742726.

Shi, Y., A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver and A. Hanjalic, "Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering", in "Proceedings of the Sixth ACM Conference on Recommender Systems", RecSys '12, p. 139–146 (Association for Computing Machinery, New York, NY, USA, 2012), URL https://doi.org/10.1145/2365952.2365981.

Sidana, S., M. Trofimov, O. Horodnytskyi, C. Laclau, Y. Maximov and M.-R. Amini, "User preference and embedding learning with implicit feedback for recommender systems", Data Mining and Knowledge Discovery **35**, 2, URL http://dx.doi.org/10.1007/s10618-020-00730-8 (2021).

Smith, B. and G. Linden, "Two Decades of Recommender Systems at Amazon.com", IEEE Internet Computing **21** (2017).

Tey, F. J., T.-Y. Wu, C.-L. Lin and J.-L. Chen, "Accuracy improvements for cold-start recommendation problem using indirect relations in social networks", Journal of Big Data URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00484-0 (2021).

Wang, C., M. Niepert and H. Li, "Recsys-dan: Discriminative adversarial netwroks for cross-domain recommender systems", IEEE Transactions on Neural Networks and Learning Systems URL https://ieeexplore.ieee.org/document/8698453 (2020).

Wang, H., J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets", Proceedings of the AAAI Conf. on Artificial Intelligence URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/download/16611/15969 (2018).

Wang, H., N. Wang and D.-Y. Yeung, "Collaborative deep learning for recommender systems", in "Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", KDD '15, p. 1235–1244 (Association for Computing Machinery, New York, NY, USA, 2015), URL https://doi.org/10.1145/2783258.2783273.

Wang, J., L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, Z. Peng and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models", SIGIR'17 URL https://dl.acm.org/doi/10.1145/3077136.3080786 (2017).

Wang, Y., H.-T. Zheng, W. Chen and Z. Rui, "Lambdagan: Generative adversarial nets for recommendation taask with lambda strategy", International Joint Conference on Neural Networks (IJCNN) URL https://ieeexplore.ieee.org/document/8851869 (2019).

Wu, Y., C. DuBois, A. X. Zheng and M. Ester, "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems", URL https://doi.org/10.1145/2835776.2835837 (2016).

Xu, Y., L. Zhu, Z. Cheng, J. Li and J. Sun, "Multi-feature discrete collaborative filtering for fast cold-start recommendation", AAAI URL https://arxiv.org/abs/2003.10719 (2020).

Yu, X., X. Zhang, Y. Cao and M. Xia, "Vaegan: A collaborative filtering framework based on adversarial variational autoencoders", IJCAI URL https://www.ijcai.org/proceedings/2019/584 (2019).

Zhang, S., L. Yao, A. Sun and Y. Tay, "Deep learning based recommender system: A survey and new perspectives", ACM Computer Survey URL https://arxiv.org/pdf/1707.07435 (2018).

Zheng, Y., S. Liu, Z. Li and S. Wu, "Cold-start sequential recommendation via meta lwarner", Proceedings of the AAAI Conf. on Artificial Intelligence URL https://arxiv.org/abs/2012.05462 (2021).

Zhu, Z., S. Sefati, P. Saadatpanah and J. Caverlee, "Recommendation for new users and new items via randomized training and mixture-of-experts transformation", SIGIR URL https://doi.org/10.1145/3397271.3401178 (2020).

APPENDIX A

DATASETS

MovieLens 100K - Original
MovieLens 1M - Original
MovieLens 100K - Transformed
MovieLens 1M - Transformed

APPENDIX B

PERMISSION STATEMENTS FROM CO-AUTHORS

Permission for including co-authored material in this dissertation was obtained from co-authors, Prof. Hemanth Venkateshwara.