Collaborative Deep Learning for Recommender Systems

Sreeniketh.P (spa44), Avinash.R (ar2041), Naga Vamsi Krishna.B (nb847),
Sai Revanth.T (st1109), Sai Nikhil.E (se449)

28th April 2023

1 ABSTRACT:

Recommender systems have gained immense popularity in recent years due to their ability to provide personalized recommendations to users. Matrix Factorization (MF) is a widely used technique in traditional recommender systems, which aims to predict user-item preferences based on the past interactions between users and items. However, MF has limitations such as the inability to handle cold start problems and sparsity in data. This paper explores the motivation for Collaborative Deep Learning (CDL) as an alternative to traditional recommender systems. CDL utilizes Collaborative Filtering (CF) along with deep neural networks to overcome the limitations of MF. The paper presents an architecture and objective function for CDL, and evaluation metrics are used to measure the performance of CDL.

2 KEYWORDS:

Recommender Systems, Autoencoder, Content-based model, Latent Factor model, Collaborative Filtering Model, Matrix Factorisation.

3 INTRODUCTION:

Recommender systems (RS) are increasingly important due to the abundance of choices in online services. They enable more effective use of information and help companies such as Amazon and Netflix to target customers by recommending products or services. There are three main categories of RS: content-based methods, collaborative filtering (CF) based methods, and hybrid methods that combine the two. Due to privacy concerns, it is more difficult to collect user profiles than past activities, so CF-based methods have gained popularity in recent years. However, CF-based methods have their limitations, such as significant drops in prediction accuracy when ratings are very sparse, and they cannot be used to recommend new products that have not yet received rating information from users.

In this project, we evaluate a hierarchical Bayesian model called collaborative deep learning (CDL) proposed in the paper [1]. CDL is a tightly coupled method that combines deep learning models with CF and allows two-way interaction between deep representation learning for content information and collaborative filtering for the ratings matrix. The experiments in this project shows that CDL is more effective at learning from sparse data than previous models. By addressing the representation learning problem CDL significantly improves the recommendation performance.

4 PROBLEM FORMALIZATION:

- Similar to the work in [2], the recommendation task considered in this project takes implicit feedback [3] as the training and test data.
- The entire collection of J items (movies) is represented by a $J \times S$ matrix X_c, where row j is the bag-of-words vector $X_{c,j}$ for item j based on a vocabulary of size S.
- With I users, we define an $I \times J$ rating matrix $R = [R_{ij}]I \times J$. For example, in the dataset Movie Lens $Rij = 5$ if user i has rated movie j.
- Given part of the movie ratings in R and the content information X_c, the problem is to predict the other ratings in R. Note that although we focus on movie recommendation (where plots of movies are considered as content information) our model is general enough to handle other recommendation tasks (e.g., tag recommendation).
- The matrix X_c plays the role of clean input to the SDAE while the noise-corrupted matrix, also a $J \times S$ matrix, is denoted by X_0.
- The output of layer l of the SDAE is denoted by X_l which is a $J \times K_l$ matrix. Similar to X_c, row j of X_l is denoted by $X_{l,j}$.
- W_l and b_l are the weight matrix and bias vector, respectively, of layer l, $W_{l,*n}$ denotes column n of W_l, and L is the number of layers.
- For convenience, we use W^+ to denote the collection of all layers of weight matrices and biases.
- Note that an $L/2$-layer SDAE corresponds to an L-layer network.

5 COLLABORATIVE DEEP LEARNING

5.1 Stacked Denoising Autoencoders

SDAE [4] is a feedforward neural network for learning representations (encoding) of the input data by learning to predict the clean input itself in the output, as shown in Figure 2. Usually the hidden layer in the middle, i.e., X_2 in the figure, is constrained to be a bottleneck and the input layer X_0 is a corrupted version of the clean input data.

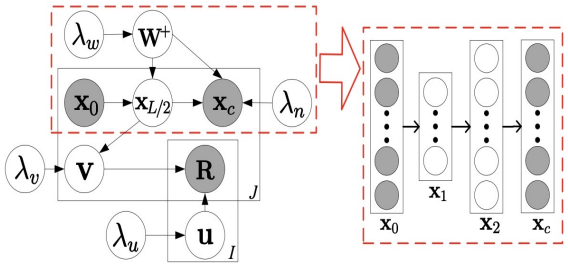


Figure 1: Graphical Model of CDL. The part inside the dashed rectangle represents an SDAE.

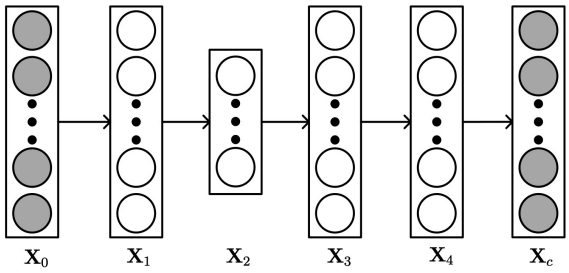


Figure 2: A 2-layer SDAE with L = 4.

We follow the same procedure as that in [2] to preprocess the text information (item content) extracted from the plots of the movies. After removing stop words, the top S words are chosen to form the vocabulary (S is 10347).

6 GENERALIZED BAYESIAN SDAE

6.1 CDL GENERATIVE PROCESS:

Similar to [5],[6] the following generative process is proposed in [1]

- For each layer l of the SDAE network,
  1. For each column n of the weight matrix W_l, draw $W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
  2. Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
  3. For each row j of X_l, draw $X_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l})$.
- For each item j,
  1. Draw a clean input $X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_J)$.
  2. Draw a latent item offset vector $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$ and then set the latent item vector to be: $v_j = \epsilon_j + X_{L,2,j*}^\top$.
- Draw a latent user vector for each user i: $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$.
- Draw a rating R_{ij} for each user-item pair (i, j): $R_{ij} \sim \mathcal{N}(u_i^\top v_j, C_{ij}^{-1})$.

6.2 Maximum A Posteriori Estimates

Maximizing the posterior probability is equivalent to maximizing the joint log-likelihood.

$$
\begin{aligned}
L = &-\frac{\lambda_u}{2}\sum_{i=1}\|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l \left(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2\right) \\
&-\frac{\lambda_v}{2}\sum_{j=1}\|\mathbf{v}_j - f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T\|_2^2 \\
&-\frac{\lambda_n}{2}\sum_{j=1}\|(f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*})\|_2^2 \\
&-\sum_{i,j}\frac{C_{ij}}{2}\left(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j\right)^2
\end{aligned}
$$

Similar to [3], [2] for a given $\mathbf{W}^+$, we compute the gradients of $\mathbf{L}$ with respect to $\mathbf{u}_i$ and $\mathbf{v}_j$ and set them to zero, leading to the following update rules:

$$
\begin{aligned}
\mathbf{u}_i &\leftarrow (\mathbf{V}\mathbf{C_i}\mathbf{V}^\mathbf{T} + \lambda_u \mathbf{I_K})^{-1}\mathbf{V}\mathbf{C_i}\mathbf{R_i} \\
\mathbf{v}_j &\leftarrow (\mathbf{U}\mathbf{C_i}\mathbf{U}^\mathbf{T} + \lambda_u \mathbf{I_K})^{-1}(\mathbf{U}\mathbf{C_j}\mathbf{R_j} + \lambda_v f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T)
\end{aligned}
$$

where $\mathbf{U} = (\mathbf{u}_i)_{i=1}^I, \mathbf{V} = (\mathbf{v}_j)_{j=1}^J, \mathbf{C}_i = diag(\mathbf{C}_{i1}, \ldots, \mathbf{C}_{iJ})$ is a diagonal matrix, $\mathbf{R}_i = diag(\mathbf{R}_{i1}, \ldots, \mathbf{R}_{iJ})^T$ is a column vector containing all the ratings of user i, and $\mathbf{C}_{ij}$ reflects the confidence controlled by a and b as discussed in [3].

$$
\begin{aligned}
\nabla_{W_l} L = &-\lambda_w \mathbf{W}_l \\
&-\lambda_v \sum_j \nabla_{W_l} f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T \mathbf{v}_j) \\
&-\lambda_n \sum_j \nabla_{W_l} f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T \mathbf{X}_{c,j*}) \\
\nabla_{b_l} L = &-\lambda_w \mathbf{b}_l \\
&-\lambda_v \sum_j \nabla_{b_l} f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T \mathbf{v}_j) \\
&-\lambda_n \sum_j \nabla_{b_l} f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T \mathbf{X}_{c,j*})
\end{aligned}
$$

Given $\mathbf{U}$ and $\mathbf{V}$, we can learn the weights $\mathbf{W}_l$ and biases $\mathbf{b}_l$ for each layer using the back-propagation learning algorithm. The gradients of the likelihood with respect to $\mathbf{W}_l$ and $\mathbf{b}_l$ are as above:

6.3 PREDICTION

Let D be the observed test data. Similar to [2], we use the point estimates of u_i, W^+, and ϵ_j to calculate the predicted rating.

$$E[R_{ij}|D] \approx E[u_i|D]^T \left(E[fe(X_{0,j}^{\cdot} W^+)^T|D] + E[\epsilon_j|D]\right)$$

where $E[\cdot]$ denotes the expectation operation. The predicted rating is approximated as: $\hat{R}_{ij} \approx (\mathbf{u}_i)^T \mathbf{v}_j^*$.

7 EXPERIMENTS:

7.1 Dataset:

In our study, we leveraged the MovieLens dataset to compare the performance of a Collaborative Deep Learning model with that of traditional Collaborative Filtering techniques. Through our experiments, we aimed to demonstrate the advantages of using the former over the latter in recommendation systems.

The Movielens dataset is a collection of over 1 million movie ratings provided by more than 6,000 users for 3,000 movies. It also includes metadata for each movie, such as title, genre, and plot.

Each user has rated at least 20 movies and each rating ranges from 0 to 5.

7.2 Data Preprocessing:

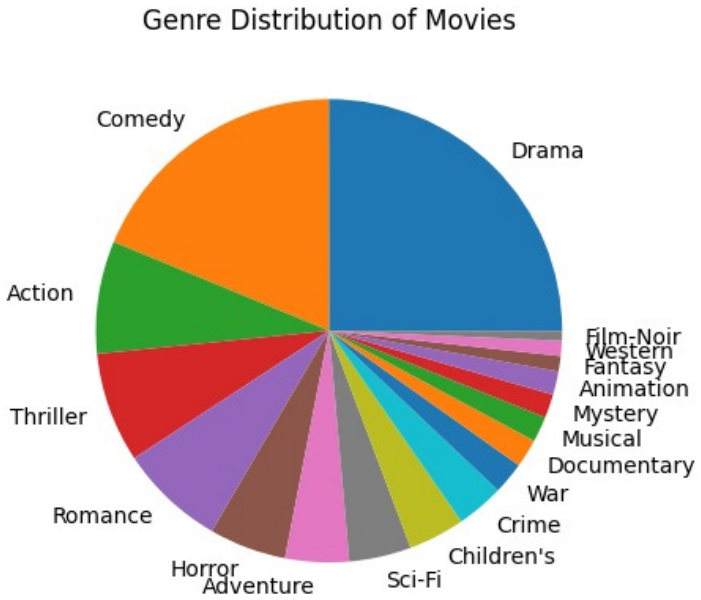
We worked with two types of data: textual data, including movie title, genre, and plot, which we transformed into content-based feature vectors using an autoencoder, and numeric data, such as userId, movieId, and rating, used to train a matrix factorization model.

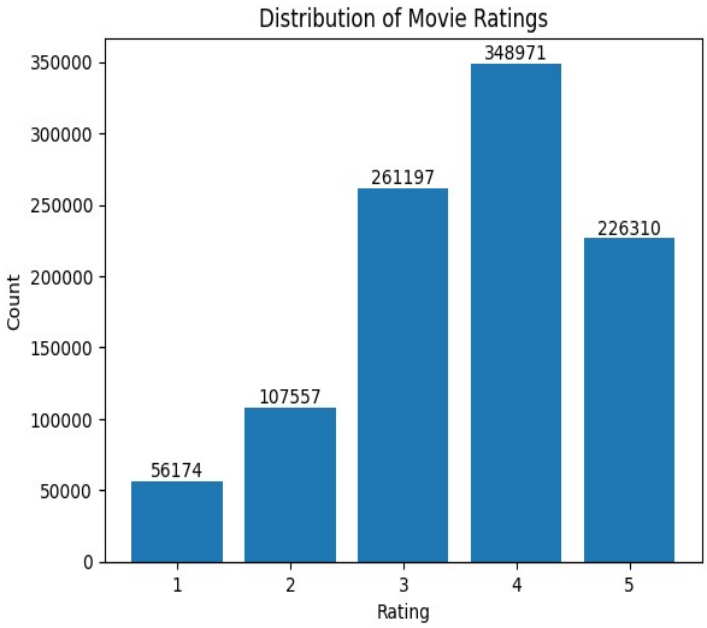
For textual data we have implemented a bag of words approach to convert the raw data into numeric vectors of fixed vocabulary size which is fed to the SDAE to capture latent item features.

For numeric data such as user-movie rating, We have divided the ratings dataset into test and train categories based on UserID. So that 80% of user reviews are utilized as training set and the remaining 20% are used to validate the model's correctness.

The dataset was cleaned in part by eliminating duplicated and non-contributing characteristics. Genres and other categorization traits are converted to a single vector.

7.3 Data Analysis:





S.No	Statistics	Value
0	Number of Movies	3883
1	Number of Users	6040
2	Number of Ratings	1000209
3	Average Ratings per User	165.60
4	Average Rating per Movie	269.89

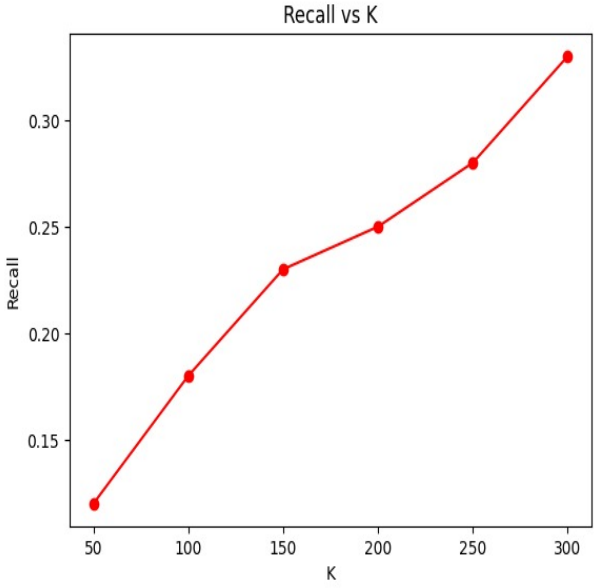
Table 1: Statistic Value

8 RESULTS:

Here is a sample output for top 10 movie recommendations.

	Movie Title	Genre
0	Duck Soup (1933)	Comedy|War
1	Arachnophobia (1990)	Action|Comedy|Sci-Fi|Thriller
2	Tic Code, The (1998)	Drama
3	Bonheur, Le (1965)	Drama
4	Close Shave, A (1995)	Animation|Comedy|Thriller
5	Love & Human Remains (1993)	Comedy
6	Cinema Paradiso (1988)	Comedy|Drama|Romance
7	All Dogs Go to Heaven (1989)	Animation|Children's
8	Heaven Can Wait (1978)	Comedy
9	Small Soldiers (1998)	Animation|Children's|Fantasy|War

We achieved a 0.33 recall for top 300 recommendations.



9 CONCLUSIONS AND FUTURE WORK:

In conclusion, collaborative deep learning is presented as a solution that overcomes the limitations of traditional recommendation systems by combining the strengths of content-based and collaborative filtering approaches. The significance of personalized recommendations is highlighted, and successful applications of collaborative deep learning, such as Netflix's movie recommendation system, are discussed.

The potential of this technology to improve recommendation accuracy and handle complex user preferences is recognized, and further research and exploration are encouraged. Additionally, collaborative deep learning can be applied beyond movie or product recommendations to personalize news articles or music recommendations.

To optimize and improve this technology, researchers are exploring using more diverse sources of data and more advanced deep learning architectures. Among the possible extensions that could be made to CDL, the bag-of-words representation may be replaced by more powerful alternatives, such as [7]. Collaborative deep learning can lead to more accurate and personalized recommendations, increasing user satisfaction and engagement, and providing insights into user behavior and preferences. Furthermore, it can contribute to the development of more advanced artificial intelligence systems, leading to further advances in the field of machine learning.

References

[1] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems, 2015.

[2] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456, 2011.

[3] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee, 2008.

[4] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[5] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, 26, 2013.

[6] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.