Journal of Engineering Science and Technology Review 17 (6) (2024) 23 - 29

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

VAEMRS: Variational Autoencoder Based Movie Recommender System

Gopal Behera[1,*], Basanta Kumar Swain[1], Ravindra Kumar Soni[2] and Jitendra Parmar[3]

[1]Department of CSE, Government College of Engineering Kalahandi, Odisha, India
[2]Department of AI and ML, Manipal University, Jaipur, India
[2]School of Computing Science and Engineering, VIT Bhopal, India

Received 15 July 2024; Accepted 13 October 2024

Abstract

Nowadays, deep learning is an emerging technique used in many research domains. The recommender system, specifically collaborative filtering, has significantly improved its performance by deploying this technique. Neural collaborative networks and their related neural network models are the bench- mark models in this domain. However, these models do not exhibit to create a continuous, robust, and structured latent space like autoencoder. On the other hand, autoencoder does not perform well in sparse data like Movielens. This article proposes a variational autoencoder based movie recommendation system (VAEMRS) to handle the above issues. In our proposed model, we consider implicit data like click vectors, normalize the interaction matrix, and pass them to the dropout layer to learn the VAE. Further, our approach applies variational concepts in neural networks. Also, use multinomial likelihood and Bayesian inference for parameter estimation. The proposed model has been tested using different quality measures on open-source datasets such as Movielens and compared with baselines. The performance results of the proposed work show the superiority over the baselines.

Keywords: Collaborative Filtering, Deep learning, Auto-encoder, Variational inference, Bayesian

1. Introduction

Recommender Systems (RecSyss) is a tool that predicts and suggests products or things to clients/users (i.e., movies to users, online out- comes to customers, services to industry, etc.) [1]. Netflix, Amazon, Flip-kart, Spotify, etc., are the popular RecSyss. RecSyss are categorized according to their filtering mechanism, which is content- based [2] and Collaborative Filtering (CF)[1], [3-4]. CF is the most widely used and accurate filtering technique in the recommender domain. There have been several types of CF models developed over time, including the Probabilistic Matrix Factorization (PMF) [5], the K-Nearest Neighbors (KNN), the Non-Negative Matrix Factorization (NMF) [5], and the Bayesian NMF (BNMF). Deep learning (DL) is an emerging research approach that per- forms better than machine learning-based matrix factorization (MF) techniques [6]. Also, DL frame- works offer more flexibility than MF techniques, allowing for the integration of shallow and deep learning [7], as well as integrating auxiliary con- tents [8], and generative techniques [9-10], among others. Deep Matrix Factorization (DeepMF) [11] implements the popular MF concept through a neural network model. Current DeepMF implementations use two embedding layers with user and item inputs based on a matrix with explicit ratings and non-preference implicit feedback. The results of the experiments confirm the superiority of DeepMF over matrix factorization-based RS [11-12] models such as NMF, BNMF, and PMF. The DeepMF framework has also been employed in the recommender domain to com- bine social behaviors (ratings and clicks) with images. It has also been used to retrieve features from the interaction matrix to enhance the initialization accuracy of social trust in RS. NCF, though not widely spread, can be viewed as a deepened DeepMF model, in which deeper layers are replaced with the 'Dot' layer, and the 'Dot' layer is replaced with a 'Concatenate' layer. How- ever, NCF produces better results than DeepMF. At the same time, it increases the running time needed to train the model. Variational AutoEncoders (VAEs) work like regular autoencoders. The encoder networks use latent space to compress raw data into latent representation, while decoder networks decompress from latent representation to output data. A key difference between VAEs and classical autoencoders is how the latent space is designed, explained, and operated. VAEs learn continuous and structured latent spaces through statistical processes, unlike traditional autoencoders. VAEs transform samples into statistical distribution parameters, typically the variance and mean of a Gaussian distribution [13]. According to Bobadilla et al. [13], the AE and VAE differ in the way to represent the latent space due to stochastic. This improves the robustness of the latent space representation and forces it to be continuous and meaningful. VAEs are considered generative models because of these properties in image processing. With a VAE [14], the latent space of Gaussian distribution parameters has been parameterized, allowing for the reconstruction of a multi-spectral image. According to Liu et al. [15], super-resolution images can be created with the help of VAEs. Zhang et al. [16] propose a flexible AE model that can adapt varying data patterns with time. Multiple papers have used VAEs as a model to improve RS results by importing the concept from image processing. For example, Liang et al. [17] tested denoising and VAEs and reported that the VAE performs better among other models. Similarly, Mohan and Nisha [18] combined social information with VAEs to enhance the quality of the RS. In this article, we assume that ratings or the interaction values can be better estimated when an autoencoder learnt latent space variationally, as the latent

*E-mail address: gbehera@gcekbpatna.ac.in
ISSN: 1791-2377
doi:10.25103/jestr.176.04

space is more robust and covers a wider latent area. In contrast, the traditional AE encodes each example as a value. Whereas VAE encodes each sample of a multivariate distribution. Random sampling is carried out from the distribution to extract the stochastic latent space values. Each epoch generates a new set of latent values in the training process. When a (user, item) tuple is provided to the proposed approach after it has been trained, the generated latent space value can be predicted more accurately in the VAE than in the regular autoencoder.

2. Related work

Variational autoencoders (VAES) [19,20] have been widely used in images. Doersch [21] studied various applications of VAE. In contrast, Miao et al. [22] presents a review of VAES on text data. The outcomes from Krishnan et al. [23] show that variational autoencoders suffer from under-fitting when modeling sparse, large, and high-dimensional data. Similar drawbacks were found while fitting VAE without annealing or annealing to $\beta = 1$.

According to Meila et al. [24], and Liang et al. [17], the regularization of the ELBO achieves maximum entropy discrimination. Therefore, ELBO attempts to incorporate Bayesian inference with discriminative estimation in a generative model. Alemi et al. [25] describe achieving maximum entropy discrimination with Bayesian inference as an information-theoretic connection. The researchers explore the deep variational information as a bottleneck and is a variational approximation of information bottleneck principle [26]. Also, they found that this is a special case while recovering the learning objective of VAEs. Further, the researchers found a more robust classification technique with $\beta < 1$. Higgins et al. [27] explained β-VAE; they use β-VAE for obtaining learning disentangled representations of images. However, they effectively set $\beta >> 1$ on the latent space. Early investigation on CF recommender system using neural networks focus on explicit data and evaluates rating prediction [28 - 31]. However, the importance of implicit data has gradually been recognized, and became recent research, therefore our work has focused on it. Behera and Nain [12] have discussed the DeepNNMF a deep non-negative MF model to address the sparsity issue in collaborative filtering-based RS. Further, the authors embedded metadata into the deep structure of neural networks to enhance the prediction task of the CF-based RS [32-33]. Jena et al. [34] proposed the neural model to recommend movies. Whereas the researchers did not consider any ranking measure while evaluating their model and did not consider coverage metrics and genome information.

Collaborative filtering models based on neural networks concentrate on explicit feedback information and assess their performance by predicting ratings [11, 39, 41, 54]. There has been a growing acknowledgment of the significance of implicit feedback, leading to a shift in focus to it in most recent research, including this study. Neural collaborative filtering (NCF) and collaborative denoising autoencoder (CDAE) [35] articles are closely related to this article.

The standard denoising autoencoder is enhanced in the collaborative denoising autoencoder (CDAE) [51] by incorporating a per-user latent factor into the input. The growth of the CDAE model's parameter count is directly proportional to both the number of users and the number of items, which increases its susceptibility to overfitting. In contrast, the parameter count in the VAE expands linearly with the number of items. Additionally, the CDAE requires extra optimization to acquire the latent factor for unseen users in order to make predictions. Rather than using the widely used dot product, NCF [14] investigate a model with non-linear interactions between the user and item latent components. Using two small datasets, the authors show how NCF performs better than traditional baselines. The number of parameters in NCF increases linearly with the number of items and users, just like in CDAE. CDAE incorporates the classical denoising autoencoder (DAE) by including a latent factor per user. In their article, the researchers investigate the logistic like-likelihood loss and Gaussian function. On the other hand, NCF explores non-linear interactions between the user and item rather than traditional dot product. Vaishaniv and Kalpan [43] use machine learning technique for sentiment-based product recommendation. Dong et al. [46] developed a hybrid model with deep structure to tackle the data sparsity issue of CF. The hybrid model jointly learns the latent factors from auxiliary information and interaction matrix. Li and James [45] proposed a Bayesian generative model (CVAE) to handle a multimedia data. The CVAE model captures deep latent structures from content data and also learn implicit relationship between user-items. Nahta et al. [47] discussed a two-step hybrid variational model to address uncertainty of ratings. They computed latent representation dynamically through encoder. Further, they use generative process to handle sparsity problem of CF. Dervishaj and Cremnesi [48] proposed a generative adversial network (GAN) based MF to learn user-item latent vectors from rating matrix. Also, they train their model vector wise using autoencoder and generator and found that GAN based model outperform the traditional CF model.

3. Dataset

The MovieLens-20 million dataset used in this model was taken from the Movie-lens library [36]. The dataset primarily contains numerous sub- datasets, such as movie genome, genre, user-rated, movie links, and tag information. The movie rating dataset is the subset of data that we are using in our model. Since the user provided the ratings directly through any feedback channel, it is regarded as an explicit dataset.

Data Preprocessing

We use MovieLens 20*M* dataset, which contains 20 million ratings and 465, 000 tag applications applied to 27, 000 movies by 138, 000 users. In the original data, there are 20000263 rating events from 138493 users and 26744 movies (sparsity: 0.540%). Tab.1, shows the snippet of a dataset. The following steps are used in preprocessing.

Table 1. Snippet of dataset

	MovieId	UserId	Ratings	Timestamp
0	2	1	3.5	1112486024
1	29	1	3.5	1113484676
2	32	1	3.5	1113484819
3	47	1	3.5	1112484727
4	50	1	3.5	1113484580

- For the VAEMRS algorithm, we only keep items that are rated by at least 50 users.
- We binarize the data by setting ratings ≥ 4 as 1, and others equal to 0. After filtering, 9868061 rating events are from 138287 users and 7345 movies (sparsity: 0.972%).

4. Proposed Work

This Section elaborates in detail about the proposed model (VAEMRS) as follows:
First, we discussed the notations that are used for model building, such as $i \in \{1, \cdots, I\}$ and $u \in \{1, \cdots, U\}$ to denote the number of items and users respectively. Also, consider learning with implicit data. The interaction between the user and the item is the click (such as watch, listen, or purchase) matrix $R \in N^{U \times I}$. The lowercase $r_u = [r_{u1}, \cdots, r_{uI}]^T \in N^I$ is a bag-of-words vector that includes number of clicks given by user u to each item i. Further, we normalize matrix $R \in [0, 1]$ and pass it to the dropout layer before training the VAE. This means that the model will have to reconstruct the click vector as some elements from the input will be missing; hence, it will learn to predict the recommendation for a given click vector. Fig.1, illustrates the variational autoencoder for the movie recommender system.

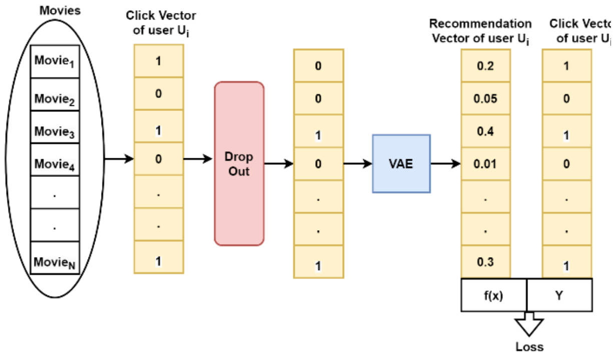

Fig. 1. Illustration of Variational Autoencoder for Movie Recommender System (VAEMRS).

Generative process: The model first samples a K-dimensional latent representation x_u from a typical Gaussian prior for each user. A non-linear function is used to transform the latent space x_u into a probability distribution over I items $\pi(x_u)$, from which it is assumed that the click history r_u was extracted and is displayed in Eq. 1.

$$x_u \sim N(0, I_k),\ \pi(x_u) \propto exp\{f_0(x_u)\},\ r_u \sim Mult(N_u, \pi(x_u)) \quad (1)$$

The multilayer perceptron with parameters θ is represented by the non-linear function $f_\theta(\cdot)$. A SoftMax function is used to normalize the transformation's output, resulting in a probability vector $\pi(x_u) \in S^{I-1}$ (an (I − 1) simplex) for the full item set. The observed bag-of-words vector x_u is assumed to be sampled from a multinomial distribution with probability $\pi(x_u)$ given the total number of clicks $N_u = \sum_i r_{ui}$ from user u. By fixing $f_\theta(\cdot)$ to be linear and utilizing a Gaussian likelihood, we may recover classical matrix factorization [49], which is a generative model that generalizes the latent factor model.

Variational Inference: Estimating θ (the parameters of $f_\theta(\cdot)$) is what we are interested in learning about the generative model in Eq. 1. In order to achieve this, we must approximate the intractable posterior distribution $p(z_u|x_u)$ for each data point. Variational inference is what we use [22]. Variational inference uses a less complicated variational distribution, $q(z_u)$, to approximate the genuine intractable posterior. We define a completely factorized (diagonal) Gaussian distribution $q(z_u)$ as follows:

$$q(z_u) = N(\mu_u, diag\{\sigma_u^2\})$$

The objective of variational inference is to optimize the free variational parameters $\{\mu_u, \sigma_u^2\}$ so that the Kullback-Leiber divergence $KL(q(z_u)||p(z_u|x_u))$ is minimized.

The objective for VAEMRS for a single user is defined in Eq. 2.

$$L_u(\theta, \phi) = \log p_\theta(r_u \mid g_\phi(r_u)) \quad (2)$$

Learning VAE: We can use variational inference to develop latent-variable models, and as usual, we can lower-bound the log marginal likelihood of the data. We aim to maximize the following for user u (the dataset's objective function is derived by averaging the objective function across all users):

$$\log p(r_u; \theta) \geq E_{q\phi}(x_u|r_u)[\log p_\theta(r_u|x_u)] - KL(q_\phi(x_u|r_u)||p(x_u)) \equiv L_u(r_u; \theta, \phi) \quad (3)$$

The evidence lower bound (ELBO) is the term used to describe this. Keep in mind that the ELBO depends on both θ and ϕ. By sampling $z_u \sim q_\phi$, we can get an unbiased estimate of ELBO and optimize it via stochastic gradient ascent. The problem, though, is that this sampling procedure does not allow us to trivially take gradients with respect to ϕ. This problem is avoided by using the reparameterization approach [24, 37]: we sample $\epsilon \sim N(0, I_K)$ and reparametrize $z_u = \mu\phi(r_u) + \epsilon \odot \sigma_\phi(r_u)$. In this way, the sampling process's stochasticity is separated, and the sampled z_u can be used to back-propagate the gradient with regard to ϕ. Algorithm1 summarizes the learning procedures of VAEMRS.

Algorithm 1. Training Procedure of VAEMRS <u>with SGD</u>
Require: Interaction Matrix $R \in N^{U \times I}$ **Ensure:**
Predicted score: $\hat{y}_{vj}$
Initialize θ, ϕ randomly
while not converged **do** Train a batch of U **for all** $u \in U$ **do**
sample $\epsilon \sim N(0, I_K)$ and compute x_u
Compute gradient $\nabla_\theta L$ and $\nabla_\phi L$ with x_u
end for
Update ϕ and θ by taking SGD steps
end while
return θ, ϕ

Further, the ELBO specified in Eq. 3 can be seen as regularization in the second KL term, whereas (negative) reconstruction error can be seen in the first term. We employ this viewpoint because it enables us to make the trade-off that is the foundation of our approach. From this angle, it makes sense to expand the ELBO by adding a parameter β to regulate the regularization's strength.

Although the original VAE (trained with ELBO in Eq. 3) is a potent generative model, one may wonder if problems in recommender systems require all of a generative model's statistical features. A trade-off is introduced between the degree to which we can fit the data and the degree to which the approximate posterior remains near to the prior during learning in the regularization view of the ELBO (Eq. 4).

$$L_u(\theta, \phi) = E_q\phi(xu, ru)[\log p_\theta(r_u \mid x_u)] - \beta KL(q_\phi(x_u \mid r_u)|| p(x_u)) \quad (4)$$

Selecting β: We inspired from KL annealing [51], a heuristic method for training VAEMRS. According to the liang et al

[17], when the β value is gradually increase from 0 to 1, the annealing KL terms slowly over a large number of gradient updates to θ, ϕ and record the best value of β, when the performance reaches at the optimal.

Further, we describe how the proposed model makes predictions of the form Eq. 1, We take into account every item based on the unnormalized estimated multinomial probability $f_\theta(y)$, given a user's click vector r. The construction of the latent space representation y for r is as follows: We just take the μ of the distribution $y = \mu_\phi(r)$ for $Mult{-}VAE^{pr}$. We can make user predictions by evaluating two methods: the encoder $g_\phi(\cdot)$ and the generative model, i.e., decoder $f_\theta(\cdot)$. For most of the latent space, CF-based RS models used matrix factorization [37], [38]; in order to obtain the latent factor for a user whose click history is not included in the training set, we often need to carry out some sort of optimization. Because of this, autoencoders are very appealing for use in industrial settings where low latency and low-cost prediction are critical.

5. Baselines

We compare the result of the VAEMRS with the following linear and non-linear state-of-the-art models:

- Collaborative denoising autoencoder (CDAE) [35]: It is a non-linear model incorporating the typical denoising autoencoder by including a latent dimension in the input.
- KNN [39]: is a non-linear model that recommends the *K* nearest items to the user according to her preference.
- SVD [40]: This technique uses a matrix factorization approach to reduce the latent dimension and takes explicit data to recommend the items.
- SVD++ [41]: SVD++ is an enhanced version of SVD, which works on implicit data.
- Weighted matrix factorization [38]: A low-rank approximation technique will train with ALS optimization to enhance the performance.
- Slim [42]: A linear technique that learns a sparse matrix (specifically item-item similarity) and solves a constrained $l1$-regularized problem.

6. Experimental Setup

This Section elaborates on the experimental setup in terms of hardware, software requirements, and performance metrics. To experiment with the proposed model, we configured the system with the CPU with Intel Core 8th generation and NVIDIA graphics. The experiment is carried out on Windows 11 and uses Python 3.8 with Keras. Further, we split the data into train-test sets with a proportion of 80: 10, respectively, and 10% of data for validation.

Performance Metrics

To evaluate the performance of VAEMRS, we employed ranking-based measures such as Recall, Normalized Discounted Cumulative Gain (NDCG), and coverage Eqs. 5 -8 represent *Recall@K*, *NDCG@K*, and *Coverage@K*, respectively. *Recall@K* includes all products ranked within the top *K* to be equally important. The choice of choosing the performance metrics such as recall, NDCG, and coverage is to prioritize the most relevant product at the top of the list.

$$Recall\ @K\ (u,v) = \frac{\sum I[w(n) \in I_u]}{\min(M|I_u|)} \quad (5)$$

$$DCG@K(u,w) = \frac{\sum 2^{I[w(n) \in I_u]} - 1}{\log(n+1)} \quad (6)$$

$$NDCG@K = \frac{DCG@K}{Ideal\ DCG} \quad (7)$$

Further, Coverage@K shows the percentage of recommended user-item pairs over the total number of potential pairs. The length of the recommended lists L can represent the number of recommended user-item pairs.

$$Coverage\ @K = \frac{Length\ (L)}{N \times U} \times 100 \quad (8)$$

Where w(n) denotes the items at rank n. Iu: denotes set of held out items that the user u clicked, and I [·]: is an indicator function. N and U represent a number of items and users, respectively.

7. Result Discussion

In this Section, we compare VAEMRS with base- line recommendation methods. Tab. 2, shows the experimental outcomes of the proposed model against the baseline approaches.

Table 2. Performance comparison of VAEMRS with Baselines.

Models	Recall@20	Recall@50	NDCG@100	Coverage@20
KNN [39]	0.38	0.40	0.26	0.20
SVD[40]	0.34	0.36	0.25	0.19
SVD++ [41]	0.35	0.37	0.24	0.19
WMF [38]	0.36	0.498	0.386	-
SLIM [42]	0.37	0.495	0.401	-
CDAE[35]	0.391	0.523	0.418	-
NCF [7]	0.526	0.541	0.447	-
CDL [44]	-	-	0.354	-
CVAE [45]	-	-	0.386	-
NeuMF [7]	-	-	0.402	-
aSDAE[46]	-	-	0.412	-
VAEMRS	**0.61**	**0.63**	**0.299**	**0.68**

Since Recall, NDCG and coverage are widely deployed metrics for comparing top-K recommendations. Therefore, we consider all three metrics on the Movielens data set in our work. According to the results of Recall@20 and Coverage@20, VAEMRS consistently outperforms other compared methods. On the Movielens data, VAEMRS outperforms the second-best techniques with a margin of at least 11% and 21% on the evaluation metrics of recall@50 and recall@20, respectively. However, other models such as CDAE, SLIM, and WMF beats VAEMRS on metrics NDCG@100. For the Movielens data set, VAEMRS achieves better results than other techniques, such as KNN, SVD, and SVD++, particularly on the metrics NDCG@100. It is surprising to see that VAEMRS achieves better results than all baseline methods.

Figs. 2, 3, and 4 show the Recall@K, NDCG@K, and Coverage@K scores of all Models on MovieLens, respectively. It is clearly visible from Fig.2, that recall@K is gradually increased and achieves the optimal value when K=100.

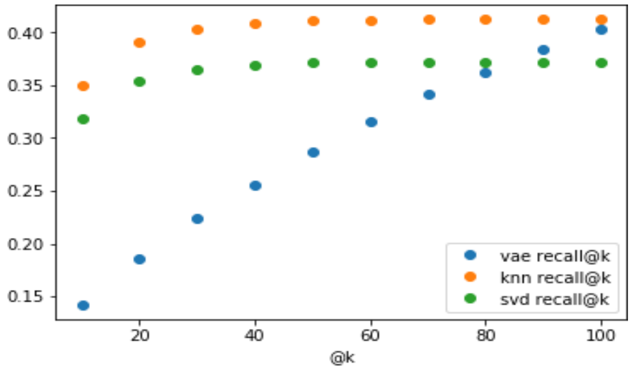

Fig. 2. The plot shows the performance of VAE across increasing K. The VAE model performs better at K is large (K=100). In the experiment, we didn't set the hyperparameter to the optimum value because the larger the parameter the slower the running time.

Similarly, we can observe from Fig. 3, that the NDCG@K of the proposed method and other baselines are almost the same at K=100.

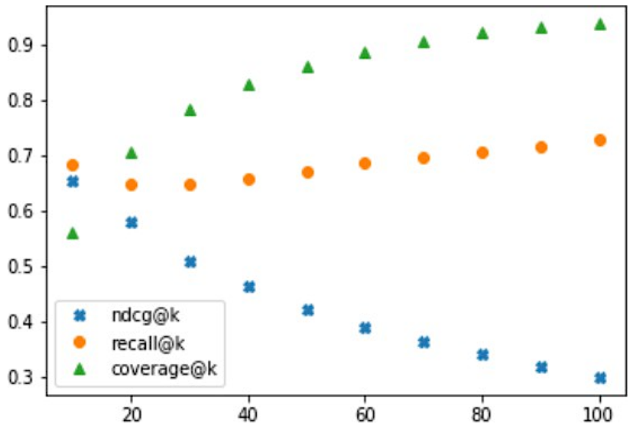

Fig. 3. The plot shows the performance of VAE across increasing K. The VAE model performs better at K=10. In the experiment, we didn't set the hyperparameter to the optimum value because the larger the parameter the slower the running time.

However, Fig. 4, signifies that the coverage of our approach gradually improved and obtained the optimal value at K=100. Meanwhile, the coverage of SVD and KNN almost overlaps each other.

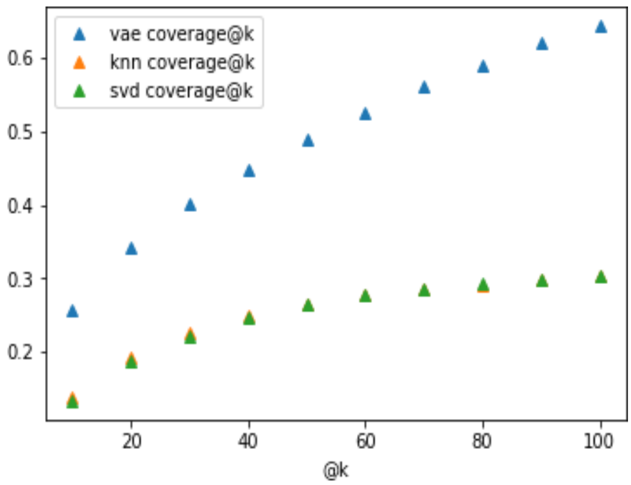

Fig. 4: The plot shows the performance of VAE across increasing K. The VAE model performs better at K is large (K=100). In the experiment, we didn't set the hyperparameter to the optimum value because the larger the parameter the slower the running time.

Further, Figs. 5 (a) and 5(b) show the training and testing time of the VAEMRS model. It can be observed that the proposed model takes less time to train and test than KNN and SVD methods

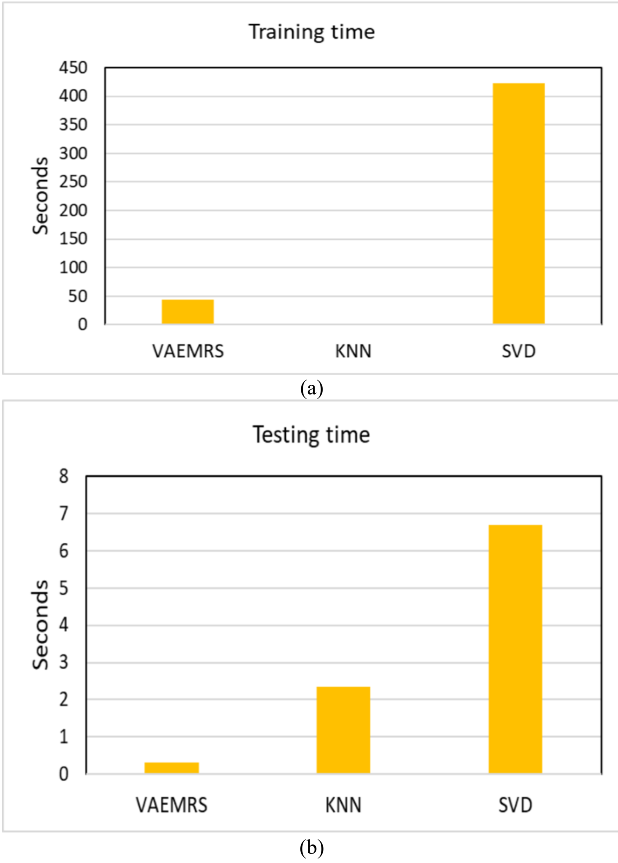

Fig. 5 (a) shows comparison of training time different models. It is found that VAEMRS takes less time during testing compared to KNN and SVD and (b) shows comparison of testing time different models. It is found that VAEMRS takes less time during testing compared to KNN and SVD.

Firstly, we have to sample the data and use a much smaller dataset to train the models. Secondly, we failed to add genomes (side information) to FM, because it fits model single-threadly, taking too much time to train and making our computers crash. Besides, in VAEMRS we have to choose some small hyperparameters, specifically, the epoch and batch-size is set to a small value.

Further, we study the impacts of several latent spaces *K*. Results on Movielens data are shown in Fig. 6. The Fig. 6 shows the performance of VAE across increasing K. The VAE model performs better at K is large (K=100) in recall and coverage, whereas the NDCG achieves the best value at K=10 and then linearly decrease with increase of K. In the experiment, we didn't set the hyperparameter to the optimum value because the larger the parameter the slower the running time.

Computational Burden: Stochastic gradient descent is used in the training of previous neural network-based collaborative filtering models [14, 51], where a single (user, item) element from the click matrix is randomly picked to execute a gradient update in each step. To update the model parameters in Algorithm 1, we subsample users and use their whole click history (all rows of the click matrix). This removes the need for negative sampling, which is typically employed in the (user, item) entry subsampling method. As a result, the hyperparameter tweaking for selecting the number of negative examples is also eliminated. Our method presents a computational problem, though, in cases when there are a large number of items. This is because computing the multinomial probability $\pi(z_u)$ could be computationally costly, as it necessitates calculating all of the item predictions for normalization. When a vocabulary counts in the millions or above, this is a typical problem for language modeling [50].

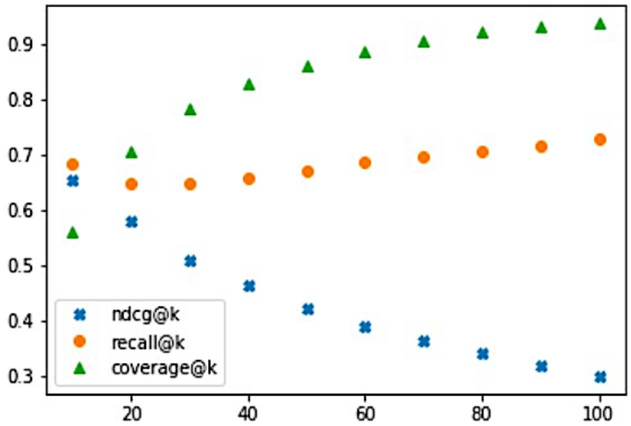

Fig. 6. The plot shows the performance of VAE across increasing K. The recall and coverage of the VAEMRS model performs better at K is large (K=100). While the NDCG of VAEMRS decreases with increase of K. In the experiment, we didn't set the hyperparameter to the optimum value because the larger the parameter the slower the running time.

Scalability and Industry Application: A Variational Autoencoder (VAE), which encodes inputs into a probability distribution across the latent space, usually a Gaussian one. By using a probabilistic method, VAEs can sample from the latent distribution, which facilitates the creation of new and varied data instances as well as improved data variability modelling.

Variational Autoencoders (VAEs) have the potential to revolutionize and change a multitude of industries, even outside of image processing. The VAEs exhibit remarkable adaptability and capacity to address intricate and diverse tasks, ranging from producing novel textual content and music to propelling drug development and evaluating financial information. This exploration of VAEs' multidomain applications demonstrates not just their technical aptitude but also the creative ways in which they can be used to address pressing issues in society.

8. Conclusion and Future Work

In this article, we develop a variational autoencoder for the movie recommendation system (VAEMRS) for the collaborative filtering of click data (implicit data). This allows us to overcome the limitations of linear factor models with respect to modeling capability. We present a generative model with a neural network parameterized multinomial likelihood function. We demonstrate that modeling user-item implicit feedback data perfectly fits multinomial likelihood. We compare the outcomes obtained against a collaborative denoising autoencoder and WMF. We empirically show that CDAE and WMF provide competitive performance in recall@50 with VAEMRS. Further, VAEMRS significantly outperforms the baseline approaches in openly available datasets, including a recently developed neural network-based model. In subsequent research, we hope to learn more about the trade-off made possible by the extra regularization parameter β and develop a deeper theoretical understanding of why it functions so effectively. Extending VAEMRS by incorporating auxiliary information about users and items might also be a way to enhance performance.

Acknowledgments. We thank the CSE Department of GCEK Bhawanipatna for providing the resources to utilize.

References

[1] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitinger, and A. Nürnberger, “Research paper recommender system evaluation: a quantitative literature survey,” in *Proc. Int. Works. Reprod. Replic. Recomm. Sys. Eval.*, Hong Kong China: ACM, Oct. 2013, pp. 15–22. doi: 10.1145/2532508.2532512.

[2] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender systems leveraging multimedia content," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–38, Sep. 2020.

[3] J. Bobadilla, F. González-Prieto, and R. Lara-Cabrera, "Deep learning feature selection to unhide demographic recommender systems factors," *Neural Comput. Appl.,* vol. 33, no. 12, pp. 7291–7308, Jun. 2021.

[4] J. Bobadilla, S. Alonso, and A. Hernando, "Deep learning architecture for collaborative filtering recommender systems," *Appl. Sci.,* vol. 10, no. 7, Apr. 2020, Art. no. 2441.

[5] G. Behera and N. Nain, “Trade-Off Between Memory and Model-Based Collaborative Filtering Recommender System,” in *Proceed. Int. Conf. Parad. Communic., Comput. Data Sci.*, M. Dua, A. K. Jain, A. Yadav, N. Kumar, and P. Siarry, Eds., in Algorithms for Intelligent Systems. , Singapore: Springer Singapore, 2022, pp. 137–146. doi: 10.1007/978-981-16-5747-4_12.

[6] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, “Neural Collaborative Filtering vs. Matrix Factorization Revisited,” in *Fourt. ACM Conf. Recomm. Sys.*, Virtual Event Brazil: ACM, Sep. 2020, pp. 240–248. doi: 10.1145/3383313.3412488.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural Collaborative Filtering,” in *Proceed. 26th Int. Conf. World Wide Web*, Perth Australia: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 173–182. doi: 10.1145/3038912.3052569

[8] S. Narang and N. Taneja, “Deep Content-Collaborative Recommender System (DCCRS),” in *2018 Int. Conf. Adv. Comp., Communic. Control Netw. (ICACCCN)*, Greater Noida (UP), India: IEEE, Oct. 2018, pp. 110–116. doi: 10.1109/ICACCCN.2018.8748600.

[9] J. Bobadilla, R. Lara-Cabrera, Á. González-Prieto, and F. Ortega, "Deepfair: deep learning for improving fairness in recommender systems," *arXiv preprint arXiv:2006.05255*, Jun. 2020.

[10] M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, "Recommender systems based on generative adversarial networks: A problem-driven perspective," *Inf. Sci.*, vol. 546, pp. 1166–1185, Feb. 2021.

[11] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, “Deep Matrix Factorization Models for Recommender Systems,” in *Proc. Twenty-Sixth Inter. Joint Conf.Artif. Intellig.*, Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 3203–3209. doi: 10.24963/ijcai.2017/447.

[12] G. Behera and N. Nain, "DeepNNMF: deep non-linear non-negative matrix factorization to address the sparsity problem of collaborative recommender systems," *Int. J. Inf. Technol.*, vol. 14, no. 7, pp. 3637–3645, Dec. 2022.

[13] J. Bobadilla, F. Ortega, A. Gutiérrez, and Á. González-Prieto, "Deep variational models for collaborative filtering-based recommender systems," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7817–7831, Apr. 2023.

[14] X. Liu, A. Gherbi, Z. Wei, W. Li, and M. Cheriet, "Multispectral image reconstruction from color images using enhanced variational autoencoder and generative adversarial network," *IEEE Access,* vol. 9, pp. 1666–1679, Dec. 2020.

[15] Z.-S. Liu, W.-C. Siu, L.-W. Wang, C.-T. Li, M.-P. Cani, and Y.-L. Chan, "Unsupervised Real Image Super-Resolution via Generative Variational AutoEncoder," in *2020 IEEE/CVF Conf. Comp. Vision Pattern Recogn. Worksh. (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1788–1797. doi: 10.1109/CVPRW50498.2020.00229.

[16] S. Zhang, J. Liu, X. Zuo, R. Lu, and S. Lian, "Online deep learning based on auto-encoder," *Appl Intell*, vol. 51, no. 8, pp. 5420–5439, Aug. 2021, doi: 10.1007/s10489-020-02058-8.

[17] Z.-S. Liu, W.-C. Siu, L.-W. Wang, C.-T. Li, M.-P. Cani, and Y.-L. Chan, "Unsupervised Real Image Super-Resolution via Generative Variational AutoEncoder," in *2020 IEEE/CVF Conf. Comp. Vision Pattern Recogn. Worksh. (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1788–1797. doi: 10.1109/CVPRW50498.2020.00229.

[18] C. Nisha and A. Mohan, "A social recommender system using deep architecture and network embedding," *Appl. Intell.*, vol. 49, no. 5, pp. 1937–1953, May 2019.

[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, Dec. 2013.

[20] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in *Proceedings of the 31st Int. Conf. Mach. Learn.*, E. P. Xing and T. Jebara, Eds., in Proceedings of Machine Learning Research, vol. 32. Bejing, China: PMLR, Jun. 2014, pp. 1278–1286. [Online]. Available: https://proceedings.mlr.press/v32/rezende14.html

[21] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, Jun. 2016.

[22] Y. Miao, L. Yu, and P. Blunsom, "Neural Variational Inference for Text Processing," in *Proceed. of The 33rd Int. Conf. Mach. Learn.*, M. F. Balcan and K. Q. Weinberger, Eds., in Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, Jun. 2016, pp. 1727–1736. [Online]. Available: https://proceedings.mlr.press/v48/miao16.html

[23] R. Krishnan, D. Liang, and M. Hoffman, "On the challenges of learning with inference networks on sparse, high-dimensional data," in *Proceed. Twenty-First Int. Conf. Artif. Intellig. Statist.*, A. Storkey and F. Perez-Cruz, Eds., in Proceedings of Machine Learning Research, vol. 84. PMLR, Apr. 2018, pp. 143–151. [Online]. Available: https://proceedings.mlr.press/v84/krishnan18a.html

[24] T. Jaakkola, M. Meila, and T. Jebara, "Maximum Entropy Discrimination," in *Adv. Neural Inform. Process. Sys.*, S. Solla, T. Leen, and K. Müller, Eds., MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/4fa53be91b4933d536748a60458b9797-Paper.pdf

[25] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, Dec. 2016.

[26] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, Apr. 2000.

[27] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. ICLR*, Apr. 2017.

[28] K. Georgiev and P. Nakov, "A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines," in *Proceedings of the 30th Int. Conf. Mach. Learn.*, S. Dasgupta and D. McAllester, Eds., in Proceedings of Machine Learning Research, vol. 28. Atlanta, Georgia, USA: PMLR, Jun. 2013, pp. 1148–1156. [Online]. Available: https://proceedings.mlr.press/v28/georgiev13.html

[29] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceed. 24th Int. Conf. Mach. Learn.*, Corvalis Oregon USA: ACM, Jun. 2007, pp. 791–798. doi: 10.1145/1273496.1273596.

[30] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders Meet Collaborative Filtering," in *Proceed. 24th Int. Conf. World Wide Web*, Florence Italy: ACM, May 2015, pp. 111–112. doi: 10.1145/2740908.2742726.

[31] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A Neural Autoregressive Approach to Collaborative Filtering," 2016, *arXiv*. doi: 10.48550/ARXIV.1605.09477.

[32] G. Behera and N. Nain, "Handling data sparsity via item metadata embedding into deep collaborative recommender system," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9953–9963, Nov. 2022.

[33] G. Behara, V. R. Yannam, A. Nayyar, and D. K. Bagal, "Integrating metadata into deep autoencoder for handling prediction task of collaborative recommender system," *Multimedia Tools Appl.*, vol. 83, pp. 42425–42147, Apr. 2023.

[34] K. K. Jena, S. K. Bhoi, C. Mallick, S. R. Jena, R. Kumar, H. V. Long, and N. T. K. Son, "Neural model based collaborative filtering for movie recommendation system," *Int. J. Inf. Technol.*, vol. 14, no. 4, pp. 2067–2077, Jun. 2022.

[35] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems," in *Proceed. Ninth ACM Int. Conf. Web Search Data Min.*, San Francisco California USA: ACM, Feb. 2016, pp. 153–162. doi: 10.1145/2835776.2835837.

[36] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Dec. 2015.

[37] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable Recommendation with Hierarchical Poisson Factorization," in *Proceed. 31st Conf. Uncert. Artif. Intellig. (UAI)*, 2015. doi: 10.1145/1935826.1935845.

[38] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," in *2008 Eighth IEEE Int. Conf. Data Min.*, Pisa, Italy: IEEE, Dec. 2008, pp. 263–272. doi: 10.1109/ICDM.2008.22.

[39] R. Ahuja, A. Solanki, and A. Nayyar, "Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor," in *2019 9th Int. Conf. Cloud Comp., Data Sci. & Eng. (Confluence)*, Noida, India: IEEE, Jan. 2019, pp. 263–268. doi: 10.1109/CONFLUENCE.2019.8776969.

[40] G. Behera, R. K. Mohapatra, and A. K. Bhoi, "A Mixed Collaborative Recommender System Using Singular Value Decomposition and Item Similarity," in *Intellig. Sys.*, vol. 728, S. K. Udgata, S. Sethi, and X.-Z. Gao, Eds., in Lecture Notes in Networks and Systems, vol. 728. , Singapore: Springer Nature Singapore, 2024, pp. 259–267. doi: 10.1007/978-981-99-3932-9_23.

[41] W. Shi, L. Wang, and J. Qin, "User embedding for rating prediction in SVD++-based collaborative filtering," *Symmetry*, vol. 12, no. 1, Jan. 2020, Art. no. 121.

[42] X. Ning and G. Karypis, "SLIM: Sparse Linear Methods for Top-N Recommender Systems," in *2011 IEEE 11th Int. Conf. Data Min.*, Vancouver, BC, Canada: IEEE, Dec. 2011, pp. 497–506. doi: 10.1109/ICDM.2011.134.

[43] N. Vaishnavi and B. Kalpana, "Sentiment based product recommendation system using machine learning techniques," *J. Eng. Sci. Technol. Rev.*, vol. 17, no. 1, pp. 16-23, Jan. 2024.

[44] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," in *Proceed. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Sydney NSW Australia: ACM, Aug. 2015, pp. 1235–1244. doi: 10.1145/2783258.2783273.

[45] X. Li and J. She, "Collaborative Variational Autoencoder for Recommender Systems," in *Proceed. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Halifax NS Canada: ACM, Aug. 2017, pp. 305–314. doi: 10.1145/3097983.3098077.

[46] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems," *Proc. AAAI Conf. Art. Intell.*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.10747.

[47] R. Nahta, Y. K. Meena, D. Gopalani, and G. S. Chauhan, "Two-step hybrid collaborative filtering using deep variational Bayesian autoencoders," *Inform. Sci.,* vol. 562, pp. 136–154, Jul. 2021, doi: 10.1016/j.ins.2021.01.083.

[48] E. Dervishaj and P. Cremonesi, "GAN-based matrix factorization for recommender systems," in *Proceed. 37th ACM/SIGAPP Sympos. Appl. Comp.*, Virtual Event: ACM, Apr. 2022, pp. 1373–1381. doi: 10.1145/3477314.3507099.

[49] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Adv. Neural Inform. Process. Sys.*, vol. 20, pp. 1-8, Dec. 2007.

[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Adv. Neural Inform. Process. Sys*, vol. 26, pp. 1-9, Jun. 2013.

[51] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, Nov. 2015.