Uncertainty-adjusted Inductive Matrix Completion with Graph Neural Networks

Petr Kasalicky
kasalpe1@fit.cvut.cz
Singapore Management University
Singapore, Republic of Singapore
Czech Technical University in Prague
Prague, Czech Republic

Antoine Ledent
aledent@smu.edu.sg
Singapore Management University
Singapore, Republic of Singapore

Rodrigo Alves
rodrigo.alves@fit.cvut.cz
Czech Technical University in Prague
Prague, Czech Republic

ABSTRACT

We propose a robust recommender systems model which performs matrix completion and a ratings-wise uncertainty estimation jointly. Whilst the prediction module is purely based on an implicit low-rank assumption imposed via nuclear norm regularization, our loss function is augmented by an uncertainty estimation module which learns an anomaly score for each individual rating via a Graph Neural Network: data points deemed more anomalous by the GNN are downregulated in the loss function used to train the low-rank module. The whole model is trained in an end-to-end fashion, allowing the anomaly detection module to tap on the supervised information available in the form of ratings. Thus, our model's predictors enjoy the favourable generalization properties that come with being chosen from small function space (i.e., low-rank matrices), whilst exhibiting the robustness to outliers and flexibility that comes with deep learning methods. Furthermore, the anomaly scores themselves contain valuable qualitative information. Experiments on various real-life datasets demonstrate that our model outperforms standard matrix completion and other baselines, confirming the usefulness of the anomaly detection module.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Anomaly detection**; *Factorization methods*; Neural networks.

KEYWORDS

matrix completion, graph neural network, anomaly detection, uncertainty

ACM Reference Format:
Petr Kasalicky, Antoine Ledent, and Rodrigo Alves. 2023. Uncertainty-adjusted Inductive Matrix Completion with Graph Neural Networks. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3604915.3610654

1 INTRODUCTION

Recommender Systems (RS) exploit historical user-item interactions (e.g., clicks, ratings, and likes) to construct a mathematical model of human preferences, with the final aim of anticipating each individual consumer's needs and growing business value. By accurately predicting user preferences and delivering personalized recommendations, RS play a crucial role in a wide range of domains, from movie recommendations [27, 47] to personalized food recipes [30].

A variety of factors make such an endeavour especially challenging and ambitious. For instance, the interaction data is inherently fickle due to human's propensity to impulsivity [4]: a user may give a poorer or better rating to a movie depending on their mood at the time of watching it, or purchase an item they do not really want in moment of excitement. Additionally, a satisfactory model of human preferences ought to remain reasonably simple and interpretable [15, 37]. Since the model is used to make financially impactful decisions, a certain amount of human-interpretable trustworthiness is expected. Furthermore, only by understanding which specific *characteristics* of each item appeal to each user, one can in fact understand the market's needs and eventually create new items with lucrative properties.

Recommender Systems approaches are typically divided into two main classes of methods. On the one hand, many recent methods exploit the larger expressivity of *artificial neural networks* [3, 46] to provide refined predictions based on combined modelling of all aspects of the data, including all interaction types and any user and item features. This approach inherits all the properties of deep learning methods, including both state-of-the-art performance and, unfortunately, black-box behaviour: it is still challenging to make reasonable interpretable sense of the trained weights of deep machine learning methods [8]. On the other hand, many classical approaches rely on *matrix factorization (MF)* [24, 31, 44] to express each predicted rating $R_{i,j}$ as an inner product between a user feature $u_i \in \mathbb{R}^d$ and an item feature $v_j \in \mathbb{R}^d$: $R_{i,j} = \langle u_i, u_j \rangle$. Here, $\mathbb{R}^d$ is the much smaller dimensional *feature space*, i.e. $d \ll m, n$. Thus, the underlying assumption is that the ground truth matrix is *low-rank*, i.e., can be decomposed into the form $UV^\top$ for some $U \in \mathbb{R}^{m\times d}$ and $V \in \mathbb{R}^{n\times d}$. Such approaches are fundamentally interpretable since they provide low-dimensional features which capture the geometric properties of user and item space. However, they are typically far more sensitive to any inaccuracy in the data.

In this paper, we construct a model which keeps the advantages of both approaches above: our final predictions come from a low-rank matrix, providing simple feature representations for each user

and item. However, our training procedure leverages the power of deep learning methods to improve the robustness and accuracy of the achieved low-rank representation by incorporating an estimate of uncertainty $W_{i,j}$ for each rating. This estimate is then used to weigh the loss function through a method analogous to existing work on mitigating aleatoric uncertainty in heteroscedastic regression [22, 25, 57]. Thus, our model consists of two parts: (1) our prediction module is a matrix in the factorized form $UV^\top$ with rank sparsity being induced by a nuclear norm regularization term; (2) In parallel, our model also involves a Graph Convolutional Neural Network f_Θ with parameters Θ on the 1-hop subgraph $G_{i,j}$ extracted from each rating as in [53]. Unlike existing works where the GCNN is trained directly to provide predictions, the output of our GCNN module, $f_\Theta(G_{i,j})$, is an *anomaly score* $W_{i,j}$. Although the first layers of the GCNN are pre-trained with the rating prediction task, our final model is trained in a fully joint manner, allowing the representations learned by the GCNN to match the needs of the anomaly detection task closely. Our main contributions are as follows:

- We propose a novel Recommender Systems architecture incorporating deep graph anomaly detection into the nuclear norm regularization objective. Ratings that are considered more anomalous are then downregulated in their contribution to the loss function used to train the matrix factorization method. This allows the MF module to be more robust to outliers.
- Our model is trained jointly, allowing the GCNN to learn feature representations which are relevant to the anomaly detection task whilst still utilizing the supervised information available in the ratings.
- We perform a wide range of experiments on five datasets and demonstrate that our model outperforms the baselines in almost all scenarios, confirming the utility of our anomaly detection module.

2 RELATED WORK

Moderating and estimating Uncertainty is an important topic in Recommendation Systems and Machine Learning in general. One of the first works to deal with natural noise in a recommender system setting was [35]. They observe that given that all human activity is vulnerable to error and that users usually perceive the rating process as a tedious task, the presence of some aberrant ratings is unavoidable. Unusually, this work focuses on purely detecting the noise: they provided a method to determine the consistency of an actual rating for a given user–item by comparing it to the corresponding predicted rating. Unlike us, the authors assume the existence of a training set with no-noise users, which was demonstrated to be highly unrealistic [10, 21]. Furthermore, they did not attempt to use the information obtained to improve the performance of the recommender, which is a key aspect of our work. In the Machine Learning and statistics literature, there has been a lot of interest in the estimation of uncertainty and its use to improve predictions. *Heteroskedastic regression*, the classic statistical framework where the variance of the noise varies with the input [34], has undergone a recent resurgence in the machine learning community [1, 22, 25, 36, 57]. In particular, our model takes inspiration from the formulation in [22], where the key idea is to introduce, for each predicted label $\hat{y}_i$, an uncertainty estimate σ_i, which is trained in a model-free manner (treated as a parameter). The loss function $\sum_{i=1}^N \ell(\hat{y}_i, y_i)$ is then replaced by $\sum_{i=1}^N \exp(-\sigma_i)\ell(\hat{y}_i, y_i) + \sum_{i=1}^N \sigma_i$. Thus, the model is free to ignore a small number of training entries in its training procedure. The benefit of this approach is that an uncertainty estimate for each training datapoint is output as a byproduct of the training procedure. More recently, [2] performed a nuclear-norm-based matrix factorization method, which relies on side information in the form of an estimate of the unreliability of each user and item. There are several key differences between this branch of the literature and our work: firstly, to the best of our knowledge, our method is the first application of this joint uncertainty estimation technique in a Recommendation Systems Context. Secondly, whilst earlier works treat the uncertainty estimates as free parameters to learn, we learn them using a Graph Neural Networks. This has several consequences: (1) our model is able to tap into the additional information from the one-hop subgraph to detect anomalies, and therefore, indirectly, to make predictions; and (2) our model is able to predict the uncertainty rating not just of datapoints in the training set, but also for the case of *unseen entries*.

Matrix Completion and Inductive Matrix Completion: The most classic approach to RS is based on low-rank matrix factorization, which aims to represent all observed entries as the inner product between two low-dimensional feature vectors u_i and v_j, i.e., the prediction matrix takes the form $R = UV^\top$ for $U \in \mathbb{R}^{m\times d}$ and $V \in \mathbb{R}^{n\times d}$. One of the earliest attempts to utilize this technique is arguably [24]. Whilst a simple approach is to restrict the rank explicitly by setting $d \ll m, n$, this poses the problem of choosing the appropriate value for d. In particular, in the exactly observed case, finding the appropriate value of d is NP-hard [6, 9]. As a result, much of the community has shifted to favouring low-rank solutions indirectly through regularization. The most canonical approach to this would be to use the nuclear norm $\|\cdot\|_*$ of the matrix of predictions as a regularizer. The nuclear norm, which is the L^1 norm of the vector of singular values (L^1 Shatten norm), induces low-rank solutions in a similar way as L^1 regularization induces component sparsity. The problem of recovering the unknown entries of an exactly observed matrix via nuclear norm minimization has been extensively studied in the pure mathematics community [5–7, 38].

Similarly, the use of the nuclear norm as a regularizer together with the square loss is also well established in the literature [13, 31] and is also well studied from a theoretical standpoint [13, 41, 42]. Inductive Matrix Completion (IMC) usually refers to a variant of the matrix completion where additional side information is present and it is assumed there exists a low-rank matrix M such that each rating can be expressed $x_i^\top M y_i$, where x_i and y_j are the side information feature vectors associated to user i and item j respectively. There is a wide body of works establishing the theoretical properties of this model [50, 55, 56], and a lot of interest in its practical applications in various fields [26, 28, 32, 48]. However, in all the cases above (both MC and IMC), the assumption is usually that the noise is i.i.d., which is an unrealistic assumption in practice. Thus, there is very limited research on how to make matrix completion robust to the

presence of large perturbations on a small proportion of anomalous entries.

Graph Neural Networks are a generalization of convolutional neural networks, which allow information to flow from layer to layer through the edges of a graph to produce an output. They have been gaining a considerable amount of attention recently, including with their use in Recommendation Systems [53], where each rating is predicted as the output of a GNN where the input is a bipartite graph containing the nearby ratings extracted from the one-hop subgraph (here the nodes correspond to the users and the items). This method achieves similar accuracy to matrix completion methods whilst focusing only on 'local' information and therefore being more somewhat robust to anomalous entries. Since the introduction of that method, various simplifications which forego the GNN structure in favor of a more simple architecture have been provided [14, 29, 39, 51], including most famously LightGCN [19]. However, most of these methods are still black boxes and do not yield low-rank solutions. Furthermore, LightGCN is only applicable to the implicit feedback prediction task (predicting which entries will be interacted with rather than predicting a numerical rating associated with the interaction). Finally, feed-forward neural networks have also been used for the recommendation task. In this case, the input is typically a one-hot encoding of the user and the item [11, 20]. The advantages and disadvantages of such methods are similar to those of GNNs.

3 METHODOLOGY

In this section, we present the requisite notation before thoroughly explaining how we have integrated uncertainty into the matrix completion optimization problem. This leads to our proposed model, denoted as *Uncertainty-adjusted inductive matrix completion* (UAIMC).

Basic notation: we write m and n for the number of users and items respectively and represent the ratings as a ground truth matrix $X \in \mathbb{R}^{m \times n}$, where $X_{i,j}$ is the rating of user i to item j. The set of observed entries is denoted by $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$. Thus, $(i, j) \in \Omega$ if the training set contains a rating from user i to item j. The nuclear norm is denoted by $\|\cdot\|_*$ and the Frobenius norm is denoted by $\|\cdot\|_F$.

One of the most commonly employed algorithms for matrix factorization is nuclear norm regularization [18, 31, 44] which minimizes the following objective function:

$$\underset{U,V}{\text{minimize}} \ \frac{1}{2|\Omega|} \sum_{(i,j)\in\Omega} \left(X_{i,j} - (UV^\top)_{i,j}\right)^2 + \frac{1}{2}\lambda \left(\|U\|_F^2 + \|V\|_F^2\right) \tag{1}$$

where $\lambda \geq 0$ is a regularization parameter. It is well known [12, 31, 43] that the above regularization strategy is equivalent to regularizing the nuclear norm [1] (i.e. the sum of the singular values) of the predictions, which naturally induces low-rank solutions.

Our strategy is to augment this optimization problem with a joint uncertainty estimation module that learns an anomaly score $W_{i,j}$ for each individual rating from the local graph structure of the rating:

$$\begin{aligned} \underset{U,V,\Theta}{\text{minimize}} \ & \frac{1}{2|\Omega|} \sum_{(i,j)\in\Omega} \left(\exp(-W_{i,j}) \left(X_{i,j} - (UV^\top)_{i,j}\right)^2 + \alpha W_{i,j}\right) \\ & + \frac{1}{2}\lambda \left(\|U\|_F^2 + \|V\|_F^2\right) + \beta \mathcal{L}_{ARR}, \\ \text{s.t.} \quad & W_{i,j} = f_\Theta(G_{i,j}) \end{aligned} \tag{2}$$

where $G_{i,j}$ is the enclosing one-hop subgraph around the rating (i, j) and f_Θ is a jointly trained Graph Neural Network with parameters Θ, whose precise architecture we describe below. Here, the term $\alpha \sum_{(i,j)\in\Omega} W_{i,j}$ is a regularization term inspired from the heteroscedastic regression literature [22, 25, 36] which induces sparsity in the uncertainty scores $W_{i,j}$ and prevents the vanishing of the first component of the loss from predicting infinite uncertainty for all data points. $\mathcal{L}_{ARR}$ is a regularization term associated with the GCNN. The whole model is trained jointly, as illustrated in Figure 1.

The precise procedure for extracting the anomaly score $W_{i,j}$ from the trainable Graph Convolutional Neural Network (GCNN) in the forward pass is described as follows. First, the enclosing subgraph extraction procedure described in **Algorithm 1** of [53] is used to convert a rating $X_{i,j}$ into a one-hop subgraph $G_{i,j}$, which is processed by the graph-level GCNN to predict the final weight $W_{i,j}$. The architecture of GCNN is composed of a few message-passing (or 'graph convolution') layers to extract a graph-level feature representation, followed by a multilayer perceptron: the relational graph convolutional operator (RGCO) [40] is used as GCNN's message passing layers with the following form:

$$x_i^{l+1} = P_0^l x_i^l + \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|} P_r^l x_j^l \tag{3}$$

where x_i^l denotes node i's feature vector at layer l, P_0^l and $\{P_r^l | r \in \mathcal{R}\}$ are learnable parameter matrices, $\mathcal{R}$ is a set for all possible ratings, and $\mathcal{N}_r(i)$ represents a set of all items rated by a user i with rating r. The output of those L stacked message-passing layers produces a representation of the nodes in the enclosing subgraph corresponding to the target user h_i and target item h_j [49, 54]:

$$h_i = \text{concat}(x_i^1, x_i^2, \dots, x_i^L). \tag{4}$$

Those representations are then aggregated to the graph-level feature vector. There are several sophisticated methods for this operation [52], but [53] empirically verified that a simple concatenation of h_i and h_j brings a better performance for some matrix completion tasks. Accordingly, our final weight $W_{i,j}$ is predicted as

$$W_{i,j} = g_\theta(\text{concat}(h_i, h_j)) \tag{5}$$

where g_θ is a multi-layer perceptron with trainable parameters θ representing M hidden layers followed by an output layer with sigmoid activation producing a one-dimensional integer. To simplify the notation in the further description, we group Equations (3), (4) and (5), as one GCNN operation $W_{i,j} = f_\Theta(G_{i,j})$ where

$$\Theta = \{\theta\} \cup \bigcup_{l\in\{1,2,\dots,L\}} \left(\{P_0^l\} \cup \{P_r^l | r \in \mathcal{R}\}\right)$$

is the set of all trainable parameters in our GCNN. Finally, Adjacent Rating Regularization (ARR) is included in our loss (2) to encourage

[1] Indeed, $\|Z\|_* = \min_{U,V:Z=UV^\top} \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2)$, cf. [31]

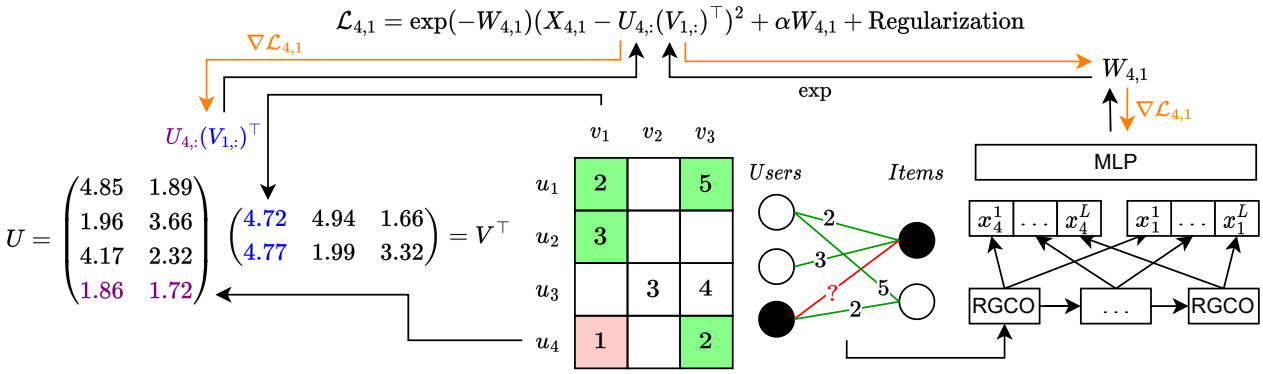

Figure 1: UAIMC: Loss function augmented by an anomaly score predicted using GCNN from an extracted enclosing subgraph

ratings adjacent to each other to have similar parameter matrices:

$$\mathcal{L}_{ARR} = \sum_{i \in \{1,2,\ldots,|\mathcal{R}|-1\}} \left\| P_{r_{i+1}} - P_{r_i} \right\|_F^2 . \tag{6}$$

Our model then consists in training the full loss function (2), jointly optimizing over the global latent representations in U, V as well as the parameters of the GCNN architecture described above to extract local graph structure.

4 EXPERIMENTS

In this section, we first describe the datasets used in our experiments, then present the baselines we compared to our model. Finally, we provide implementation details of our model and present the results.

Datasets: we selected five stable benchmarking datasets, including traditional ones like Movielens [17] and Douban [58]. Key characteristics of each dataset, including the number of users, items, and ratings, are summarized in Table 1.

Preprocessing: all datasets contain explicit ratings with values from the set $\{1, 2, 3, 4, 5\}$. The Movielens 25M dataset is the only exception with non-integer ratings. However, to maintain consistency throughout our experimental setup, we rounded all ratings up. Moreover, for all datasets, we ensured that each user-item pair had at most one rating. If a pair had more than one rating, such as in the Douban [58] or Amazon Video Games [33] dataset, we retained only the most recent rating based on its timestamp. Finally, to ensure reasonable sparsity for the Amazon Video Games dataset, we pruned the dataset by discarding the users and items with fewer than five ratings. We randomly split all datasets into training ratings, validation ratings, and test ratings in a 90:5:5 ratio.

4.1 Implementation Details

The optimization of the loss function (2) was optimized with automatic differentiation in Pytorch with the ADAM [23] optimizer using cosine decay of learning rate. Since representing every observed entry using a graph is a highly memory-intensive operation, the optimization had to be done in batches. The batch size was different for each dataset because memory requirements are affected by the size of the enclosing subgraph generated from the ratings. For dense datasets like Douban and MovieLens 100K, the batch size was 200. For Amazon Video Games, the batch size was 1000.

Parameters P in the GCNN part of UAIMC predicting h_i and h_u were initialized by a GCNN trained to predict ratings, as described in [53]. We adopted the suggested hyperparameters from the same paper and set the number of message-passing layers at $L = 4$ and $\beta = 0.001$. The architecture of MLP contains one hidden layer with 64 dimensions, ReLU activations and dropout regularization. The MLP parameters θ were initialized with Xavier Initialization [16]. P and θ were trained, during the end-to-end training of UAIMC, to predict anomaly score $W_{i,j}$ from a feature representation extracted from a graph. The regularization hyperparameter α was cross-validated, resulting in a value $\alpha = 45$.

4.2 Comparative Analysis and Results

In our study, we compared our model to five established models. Our first baseline, UserKnn, is a widely-adopted yet straightforward model that employs cosine similarity to compare users. The optimal number of neighbours was determined individually for each dataset via cross-validation.

Since our model includes a Matrix Factorization and GNN modules as sub-components in its architecture, we included both the classic Soft-Impute (SI) algorithm [31] and the original Inductive Graph-based Matrix Completion (IGMC) [53] in our baselines. The corresponding hyperparameters for IGMC were adopted from the [53]. The hyperparameters for SI were cross-validated for each dataset independently.

In addition to UserKNN, SI, and IGMC, we also included a combination of a noise correction method (NC) [45] followed by either UserKNN or SI, labeled as NC-KNN and NC-SI, respectively. NC was configured with $\kappa_u = x_u - \tau p_u$ and $\upsilon_u = x_u + \tau p_u$ settings. To enhance the algorithm's sensitivity to noise, τ was set to 0.25. Consequently, 2-4% of the ratings were identified as noise and adjusted according to **Algorithm 2** in [45].

The RMSE of our model, along with baseline models measured on the test set, is shown in Table 1. Our model outperformed SI for all datasets, and outperformed all baselines for most datasets. However, despite diligent fine-tuning of the SI hyperparameters for the Amazon Video Games dataset, its RMSE was worse than

Table 1: Datasets used in the experiment with reported sizes after the preprocessing and RMSE measured on test set

Users	Items	Ratings		KNN	SI	IGMC	NC-KNN	NC-SI	UAIMC
943	1682	100,000	Movielens 100K	1.0217	0.9363	0.9457	1.0225	0.9348	**0.9309**
59,740	33,482	526,768	Amazon Video Games	1.2898	1.3480	**1.1409**	1.3015	1.3470	1.3471
2,709	2,783	762,513	Douban	0.8135	0.7500	0.7778	0.8137	0.7495	**0.7452**
6,040	3,706	1,000,209	Movielens 1M	0.9672	0.8536	0.8895	0.9693	0.8516	**0.8479**
162,541	59,047	25,000,095	Movielens 25M	0.9571	0.7979	0.8427	0.9541	0.7974	**0.7962**

IGMC's RMSE, and even our model could not make up the difference. Preprocessing training subsets of datasets using NC also helps because NC-SI performed better than SI for all processed datasets. However, we show that UAIMC remains the preferable method overall. This superiority could likely be attributed to its trainable mechanism for detecting potentially unstable ratings, providing it with a distinct advantage in terms of adaptability and accuracy.

5 CONCLUSION

We have introduced a hybrid Recommender System model, which relies on a low-rank Matrix Factorization backbone augmented by an anomaly detection module that taps on the representation power of deep Graph Neural Networks to extract features and detect anomalous ratings based on all nearby interactions. Experiments on five datasets with five baselines confirm that our model exhibits state-of-the-art performance, outperforming all baselines in most cases. This confirms the notion that simple methods such as low-rank MF have great potential to remain competitive with more complex baselines as long as the sensitivity to outliers can be mitigated. In particular, we hope that this initial foray into the incorporation of joint uncertainty estimation modules into classic Recommendation System methods will give rise to further research efforts. Although we have experimentally verified that the inclusion of ratings-wise uncertainty estimation improves the low-rank MF backbone and outperforms it in terms of performance, it would also be worthwhile to investigate the relationship between the error of each test sample and its associated uncertainty $\exp(-W_{i,j})$. Furthermore, the uncertainty score could also be used to modulate the exploration-exploitation tradeoff in online contexts. We leave a more detailed study of those considerations for future work.

ACKNOWLEDGMENTS

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. Petr Kasalicky was also supported by the Grant Agency of Czech Technical University (SGS23/210/OHK3/3T/18) and together with Rodrigo Alves thank Recombee for their support.

REFERENCES

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* 76 (2021), 243–297.

[2] Rodrigo Alves, Antoine Ledent, and Marius Kloft. 2023. Uncertainty-Adjusted Recommendation via Matrix Factorization With Weighted Losses. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[3] Martin Anthony, Peter L Bartlett, Peter L Bartlett, et al. 1999. *Neural network learning: Theoretical foundations*. Vol. 9. cambridge university press Cambridge.

[4] Albert-Laszlo Barabasi. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435, 7039 (2005), 207–211.

[5] Emmanuel J Candes and Yaniv Plan. 2010. Matrix completion with noise. *Proc. IEEE* 98, 6 (2010), 925–936.

[6] Emmanuel J. Candès and Benjamin Recht. 2009. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717.

[7] Emmanuel J. Candès and Terence Tao. 2010. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory* 56, 5 (May 2010), 2053–2080.

[8] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. 2017. Interpretability of deep learning models: A survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*. 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397411

[9] Alexander L Chistov and D Yu Grigor'Ev. 1984. Complexity of quantifier elimination in the theory of algebraically closed fields. In *Mathematical Foundations of Computer Science 1984: Proceedings, 11th Symposium Praha, Czechoslovakia September 3–7, 1984 11*. Springer, 17–31.

[10] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing? How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 585–592.

[11] Gintare Karolina Dziugaite and Daniel M Roy. 2015. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443* (2015).

[12] Maryam Fazel. 2002. *Matrix rank minimization with applications*. Ph. D. Dissertation. PhD thesis, Stanford University.

[13] Rina Foygel, Ohad Shamir, Nati Srebro, and Russ R Salakhutdinov. 2011. Learning with the weighted trace-norm under arbitrary sampling distributions. In *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2133–2141. http://papers.nips.cc/paper/4303-learning-with-the-weighted-trace-norm-under-arbitrary-sampling-distributions.pdf

[14] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[15] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[16] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.

[17] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages. https://doi.org/10.1145/2827872

[18] Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. 2015. Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares. *Journal of Machine Learning Research* 16, 104 (2015), 3367–3402. http://jmlr.org/papers/v16/hastie15a.html

[19] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[21] Anthony Jameson, Martijn C Willemsen, Alexander Felfernig, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Li Chen. 2015. Human decision making and recommender systems. In *Recommender systems handbook*. Springer, 611–648.

[22] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems* 30 (2017).

[23] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

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

[25] Quoc V Le, Alex J Smola, and Stéphane Canu. 2005. Heteroscedastic Gaussian process regression. In *Proceedings of the 22nd international conference on Machine learning*. 489–496.

[26] Antoine Ledent, Rodrigo Alves, Yunwen Lei, and Marius Kloft. 2021. Fine-grained Generalization Analysis of Inductive Matrix Completion. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 25540–25552. https://proceedings.neurips.cc/paper/2021/file/d6428eecbe0f7dff83fc607c5044b2b9-Paper.pdf

[27] George Lekakos and Petros Caravelas. 2008. A hybrid approach for movie recommendation. *Multimedia tools and applications* 36 (2008), 55–70.

[28] Rong Li, Yongcheng Dong, Qifan Kuang, Yiming Wu, Yizhou Li, Min Zhu, and Menglong Li. 2015. Inductive matrix completion for predicting adverse drug reactions (ADRs) integrating drug–target interactions. *Chemometrics and Intelligent Laboratory Systems* 144 (2015), 71 – 79.

[29] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2023*. 417–428.

[30] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105* (2019).

[31] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research* 11, 80 (2010), 2287–2322. http://jmlr.org/papers/v11/mazumder10a.html

[32] Nagarajan Natarajan and Inderjit S. Dhillon. 2014. Inductive matrix completion for predicting gene and disease associations. *Bioinformatics* 30, 12 (06 2014), i60–i68. https://doi.org/10.1093/bioinformatics/btu269

[33] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Conference on Empirical Methods in Natural Language Processing*.

[34] D.A. Nix and A.S. Weigend. 1994. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Vol. 1. 55–60 vol.1. https://doi.org/10.1109/ICNN.1994.374138

[35] Michael P O'Mahony, Neil J Hurley, and Guénolé CM Silvestre. 2006. Detecting noise in recommender system databases. In *Proceedings of the 11th international conference on Intelligent user interfaces*. 109–115.

[36] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[37] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2060–2069.

[38] Benjamin Recht. 2011. A Simpler Approach to Matrix Completion. *Journal of Machine Learning Research* 12, null (Dec. 2011), 3413–3430.

[39] Seema Safar, Babita Roslind Jose, Jimson Mathew, and T Santhanakrishnan. 2022. Light-weight Recommendation System using Graph Neural Networks. In *2022 IEEE 19th India Council International Conference (INDICON)*. IEEE, 1–6.

[40] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling Relational Data with Graph Convolutional Networks. arXiv:1703.06103 [stat.ML]

[41] Ohad Shamir and Shai Shalev-Shwartz. 2011. Collaborative Filtering with the Trace Norm: Learning, Bounding, and Transducing. In *Proceedings of the 24th Annual Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 19)*. PMLR, 661–678.

[42] Ohad Shamir and Shai Shalev-Shwartz. 2014. Matrix Completion with the Trace Norm: Learning, Bounding, and Transducing. *Journal of Machine Learning Research* 15 (2014), 3401–3423. http://jmlr.org/papers/v15/shamir14a.html

[43] Nathan Srebro. 2004. Learning with matrix factorizations. (2004).

[44] Nathan Srebro and Adi Shraibman. 2005. Rank, trace-norm and max-norm. In *Learning Theory: 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005. Proceedings 18*. Springer, 545–560.

[45] Raciel Yera Toledo, Yailé Caballero Mota, and Luis Martínez. 2015. Correcting noisy ratings in collaborative recommender systems. *Knowledge-Based Systems* 76 (2015), 96–108. https://doi.org/10.1016/j.knosys.2014.12.011

[46] Vojtěch Vančura and Pavel Kordík. 2021. Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP). In *Artificial Neural Networks and Machine Learning – ICANN 2021*, Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter (Eds.). Springer International Publishing, Cham, 138–149.

[47] Vojtěch Vančura, Rodrigo Alves, Petr Kasalický, and Pavel Kordík. 2022. Scalable Linear Shallow Autoencoder for Collaborative Filtering. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 604–609. https://doi.org/10.1145/3523227.3551482

[48] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. 2009. Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices via Convex Optimization. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (Eds.). Curran Associates, Inc., 2080–2088.

[49] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks? arXiv:1810.00826 [cs.LG]

[50] Miao Xu, Rong Jin, and Zhi-Hua Zhou. 2013. Speedup Matrix Completion with Side Information: Application to Multi-Label Learning. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2301–2309.

[51] Siyong Xu, Cheng Yang, Chuan Shi, Yuan Fang, Yuxin Guo, Tianchi Yang, Luhao Zhang, and Maodi Hu. 2021. Topic-Aware Heterogeneous Graph Neural Network for Link Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 2261–2270. https://doi.org/10.1145/3459637.3482485

[52] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[53] Muhan Zhang and Yixin Chen. 2019. Inductive Graph Pattern Learning for Recommender Systems Based on a Graph Neural Network. *CoRR* abs/1904.12058 (2019). arXiv:1904.12058 http://arxiv.org/abs/1904.12058

[54] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In *AAAI Conference on Artificial Intelligence*.

[55] Kai Zhong, Prateek Jain, and Inderjit S. Dhillon. 2015. Efficient Matrix Sensing Using Rank-1 Gaussian Measurements. In *Algorithmic Learning Theory*, Kamalika Chaudhuri, CLAUDIO GENTILE, and Sandra Zilles (Eds.). Springer International Publishing, Cham, 3–18.

[56] Kai Zhong, Prateek Song, Zhao and, and Inderjit S Dhillon. 2019. Provable Nonlinear Inductive Matrix Completion. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, and R. Garnett (Eds.). 11439–11449.

[57] Lijun Zhou, Antoine Ledent, Qintao Hu, Ting Liu, Jianlin Zhang, and Marius Kloft. 2021. Model uncertainty guides visual object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3581–3589.

[58] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. 3001–3008.