One-Class Recommendation through Unsupervised Graph Neural Networks for Link Prediction

Marcos P. S. Gôlo[1] and Leonardo G. Moraes[1] and Rudinei Goularte[1] and Ricardo M. Marcacini[1]

Institute of Mathematics and Computer Sciences - University of São Paulo (USP)
{marcosgolo,leonardo.g.moraes}@usp.br, {rudinei,ricardo.marcacini}@icmc.usp.br

Abstract. Recommender systems play a key role in every online platform to provide users a better experience. Many classic recommendation approaches might find issues, mainly modeling user relations. Graphs can naturally model these relations since we can connect users interacting with items. On the other hand, when we model user-item relations through graphs, we do not have interactions between all users and items. Furthermore, it is difficult to cover the scope of non-recommendations, i.e., every item the user does not want to be recommended, and there are few interactions of this type. An alternative is One-Class Learning (OCL) which is able to recommend or not an item for a user only with the recommendations, mitigating the needing to cover the scope of non-recommendations and the imbalance scenario. However, before using OCL to recommend, we must obtain a representation through the graph for the users and items. In this sense, we propose the one-class recommendation via representations learned by unsupervised graph neural networks (GNNs) for link prediction to generate a more robust and meaningful representation of users and items in the one-class recommendation. In the results, comparing the representations in the one-class recommendation, our GNNs for link prediction outperform other methods to represent the users and items in the one-class recommendation. Furthermore, our proposal, a GNN for link prediction and an OCL algorithm, outperforms an end-to-end GNN for link prediction. Thus, our proposal recommended better and learned more robust representations.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Machine learning algorithms**; *Neural networks*; • **Information systems** → **Recommender systems**.

Keywords: One-Class Learning, Recommender Systems, Graph Neural Networks, Link Prediction

1. INTRODUCTION

Finding items that match users' interests is an important feature for any online platform, and recommender systems are indispensable for helping in this task. Still, these systems must find a way to deal with some issues, mainly the modeling of users' preferences and relations [Wu et al. 2020; Khoali et al. 2022]. User preferences and relations essentially have a graph structure since nodes can be user and item, and the edges can be user-user, item-item, and user-item relations. Furthermore, graphs benefit from incorporating structured external information [Wu et al. 2020; Ru et al. 2021].

Common modeling of recommender systems through graphs is the original modeling. The original modeling considers only user and item relationships to generate the graph [Wu et al. 2020]. However, those graph representations are incomplete, i.e., there are no interactions between all users and items. Even so, the graph has interactions the user would like to be recommended (positive) and interactions that the user would not like (negative). However, positive interactions are more present in the real world than negative interactions, which are more scarce [Khoali et al. 2022].

Traditional recommender systems algorithms need positive and negative interactions to recommend items to users. Therefore, we need to cover the scope of positive and negative interactions in this scenario. Still, covering the scope of positive recommendations is easier, given the number of posi-

This work was partially funded by CAPES grant number [88887.671481/2022-00].
Copyright©2022 Permission to copy without fee all or part of the material printed in KDMiLe is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

tive recommendations. However, covering the scope of negative interactions is challenging as these interactions have a more extensive scope and few interactions [Khoali et al. 2022].

In this scenario, One-Class Learning (OCL) arises as an alternative. OCL algorithms use only one class to train and predict whether or not a new example belongs to the training class. In this sense, OCL is a learning that only needs interest class data to train and can recommend or not an item for a user. Thus, OCL has the advantage of not having to cover the scope of non-recommendations [Pan et al. 2008; Gôlo et al. 2021; Khoali et al. 2022]. Furthermore, OCL is adequate for imbalance scenarios, such as recommender systems [Fernández et al. 2018]. However, OCL is more challenging, as it only has interesting recommendations for training. Part of this challenge relates to the representation directly influencing the OCL [Gôlo et al. 2021; Khoali et al. 2022; Gôlo et al. 2022]. Therefore, one-class recommendations have the challenge of representing the user, the item, and its iterations, a challenge already known in recommender systems [Wu et al. 2020].

Generally, the gaps (i) cover the non-recommendations scope and (ii) represent the user, item, and iterations on the OCL scenario are mitigated individually or by studies that apply one-class recommendation without graphs, or by studies that use graph representations considering non-recommendations. In this way, we propose a Graph Neural Network (GNN) for link prediction to the one-class recommendation. GNNs are recently used for representation learning and have obtained state-of-the-art results, even more in the recommender systems literature [Wu et al. 2020], but not in the one-class recommendation. Another novelty of our proposal is an unsupervised GNN for link prediction, which is little explored in the recommendation systems field. This GNN type is directed to the recommendation problem since the link prediction task predicts interactions between nodes, which in recommender systems are interactions between users and items, i.e., recommendations and non-recommendations [Li and Chen 2013; Yang et al. 2019]. In short, our proposed approach has the following contributions:

(1) We model an enriched heterogeneous graph considering only interest class relations (recommendations) to improve the recommendation;
(2) We propose an unsupervised GNN via link prediction task for recommender systems;
(3) We recommend items considering only recommendations to train without having to cover the wide scope of non-recommendations from users.

We carried out an experimental evaluation involving a real-world dataset for movie recommendations. We compared our proposal with four other strategies. Three are user and movie representations considering the movie's review and the graph structure. We use a state-of-the-art algorithm for text representation to represent the reviews. The last strategy is an end-to-end GNN. The results demonstrated that our proposal outperforms other methods to represent users and items in the OCL scenario and outperforms an end-to-end GNN.

We organized the remainder of this paper as follows. First, section 2 discusses related work under OCL and GNN via Link prediction for recommender systems. Second, section 3 presents the proposal to deal with a one-class recommendation. Third, section 4 presents the experimental evaluation, discussing the results obtained when applying the proposed representations to real-world data for recommender systems. Finally, Section 5 presents our concluding remarks and future work.

2. RELATED WORK

One-class approaches have been used as a tool in recommender systems. [Pan et al. 2008] propose two frameworks to deal with problems with one-class collaborative filtering (OCCF). The authors explore the user-item matrix as a feature for the OCCF. On the other hand, [Zhao et al. 2015] explore the one-class recommendation through item and user representations generated by a personalized matrix factorization (MF). Also, using OCCF, [He and McAuley 2016] uses MF and visual features represented by a Deep Convolutional Neural Network.

Addressing some common issues of one-class recommendation and commonly used strategies to tackle such issues, [Khoali et al. 2022] propose a bayesian personalized ranking based on a neural network using as input the user-item matrix as features. With a different one-class classification approach, [Li and Chen 2013] converts a recommendation problem into a link prediction one by using a generic kernel-based ML approach to map the transactions into a bipartite user-item graph, with the prediction model built from the one-class SVM algorithm.

Graph representation approaches have been used in recommender systems [Li and Chen 2013; Wu et al. 2020]. The study [Wu et al. 2020] is a survey of Graph Neural Networks (GNNs) in recommender systems. This study shows the extensive use of GNNs in this field. The authors highlighted the advantages of using heterogeneous graphs to represent relations, as well as user-item iterations, to obtain a robust representation of the user and the items. Furthermore, GNNs allow link prediction approaches that are adequate for the recommender systems problem [Wu et al. 2021].

Considering GNN for link prediction in the recommender systems, we can cite studies such as [Zhang and Chen 2018] that propose a novel method that uses GNN to predict edges in graphs, indicating the GNN for link prediction as promising in the recommender systems. Also, [Islam et al. 2020] suggests that GNN for link prediction is an alternative to the recommender systems field. Furthermore, [Wu et al. 2021] also cites recommender systems as promising applications for these GNNs.

As the above discussion shows, one-class recommendation studies use traditional and non-enriched representations and generally do not explore graphs for recommendations modeling. Following a different path, [Li and Chen 2013] explores one-class recommendations with a graph for link prediction. However, this study does not explore the GNNs. GNNs for link prediction studies indicate recommender systems as a promising application. However, using GNNs for link prediction in recommender systems is scarce. Thus, there exists a need for more studies on GNNs via link prediction for the one-class recommendation. Therefore, with the use and advantages of OCL, heterogeneous graph modeling for the recommendation, and GNN for link prediction to represent users and items, we propose the one-class recommendation through unsupervised graph neural networks for link prediction.

3. UNSUPERVISED GRAPH NEURAL NETWORKS VIA LINK PREDICTION FOR ONE-CLASS RECOMMENDATION

We propose learning a more robust and adequate representation for the one-class recommendation considering the advantages of the graph structure. Thus, we separate the proposal into three steps: (i) create an enriched graph for the users, items, and metadata; (ii) learn representations through a graph neural network via link prediction task; and (iii) use the One-Class Support Vector Machines to perform the one-class recommendation. Figure 1 illustrates each proposal step. First, we model our recommender problem with a heterogeneous enriched graph. Second, we learn representations for the graph nodes with a GNN for link prediction. Finally, we use one-class learning to recommend or not an item for a user. The following sections explain each step.

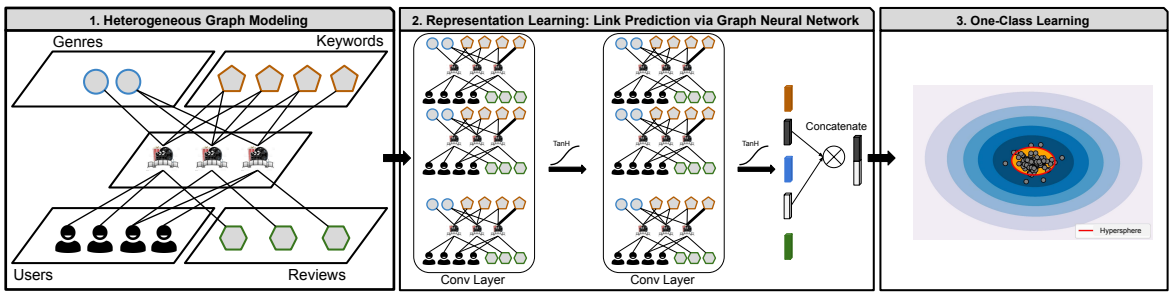

Fig. 1. Illustration prepared by the authors of the proposal steps.

3.1 Heterogeneous Enriched Graph Modeling

We model our user-item recommendation problem with graphs since graphs naturally model the recommendation problem. First, we create a heterogeneous graph. Then, we model the graph with five nodes (users, items, keywords, genre, and review) and four edge types (user-item, keyword-item, genre-item, and review-item). The keywords represent the movie plot keywords, the genre is the movie genre, and the reviews are the reviews for the movie. We add these metadata to enrich the graph and make it more connected. Finally, we model our graph as a one-class graph since we only add edge user-item of the interest, i.e., user evaluations for items with rating 5, with the intuition that learn a representation only with the set of interest labeled. After modeling the graph, we can learn a representation with graph neural networks considering the link prediction task.

3.2 Graph Neural Networks for Link Prediction

Before applying the graph neural network for link prediction, we adopted a strategy for all graph nodes aiming to reach representations to improve the representation learning of our Graph Neural Network. Thus, we use a graph regularization framework from [Rossi et al. 2014; do Carmo and Marcacini 2021]. Equation 1 defines regularization and has two terms. The first term determines that neighboring nodes in the graph have similar embedding vectors. The second term preserves the initial node representation according to a factor. Finally, our goal is to minimize the $Q(\boldsymbol{F})$:

$$Q(\boldsymbol{F}) = \frac{1}{2} \sum_{o_i, o_j \in O} w_{o_i,o_j} \Omega(\boldsymbol{f}_{o_i}, \boldsymbol{f}_{o_j}) + \mu \sum_{o_i \in O_e} \Omega(\boldsymbol{f}_{o_i}, \boldsymbol{k}_{o_j}) \tag{1}$$

in which, $\boldsymbol{F}$ represents the graph nodes representations, O is the set of all graph nodes, O_e is the set of graph nodes with initial representations, w_{o_i,o_j} indicates the weight of the connection between the nodes o_i and o_j, Ω is a distance function between embedding vectors, $\boldsymbol{f}_{o_i}$ is a generated embedding, $\boldsymbol{g}_{o_i}$ is an initial representation, and μ is a factor of preserving (with $\mu > 0$).

After generating initial representations for all nodes, we will apply a Graph Neural Network (GNN). The GNN considers the structured representation of each node in the graph and the adjacency matrix $\boldsymbol{A}$ as input for learning the representations of the nodes. Thus, the initial representation of the nodes will be called $\boldsymbol{n}_i \in \boldsymbol{N}$. We denote $g(\boldsymbol{N}, \boldsymbol{A}; \boldsymbol{\mathcal{W}})$ to represent a GNN with trainable weights $\boldsymbol{\mathcal{W}} = \{\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(l)}\}$ in l hidden layers. For the i-th layer, the GNN propagation rule can be summarized as [Wu et al. 2020]:

$$\boldsymbol{H}^{i+1} = g(\boldsymbol{H}^{(i)}, \boldsymbol{A}; \boldsymbol{W}^{(i)}) \tag{2}$$

in which, $\boldsymbol{H}^{(i)}$ is the input to the i-th layer of the GNN, and $\boldsymbol{H}^{i+1}$ is the output of this layer. It is worth noting that the $\boldsymbol{N}$ representations are the input to the first layer, which is equivalent to $\boldsymbol{H}^{(0)}$. The embeddings learned for each object in the graph are represented by $\boldsymbol{H}^{(l)}$. Therefore, the information, i.e., characteristics present in the representations of adjacent nodes (neighbors), are aggregated through a neural network.

For unsupervised representation learning via GNN for link prediction, we pass the last layer $\boldsymbol{H}^{(l)}$ for a LinkEmbedding layer that predicts a link between nodes. Equation 3 defines this layer:

$$\boldsymbol{z} = \sigma \left(\sum_{o_i, o_j \in O} \boldsymbol{h}_{o_i}^{(l)} \cdot \boldsymbol{h}_{o_j}^{(l)} \right) \tag{3}$$

in which, σ is a activation function and $\boldsymbol{z}$ are the link predictions.

For this GNN for link prediction, learn embeddings in an unsupervised way. First, we generate fake edges and use original edges to train through Equations 2 and 3. Finally, we use binary accuracy as a loss function. To obtain the embedding nodes, we use the $\boldsymbol{H}^{(l)}$ layer. After generating the user and item representations with the GNN, we use OCL to recommend or not an item for a user.

3.3 One-Class Learning

In One-Class Learning (OCL) [Tax 2001; Alam et al. 2020], the training of the algorithms is only with examples of the interest class (rating 5), i.e., in the absence of counterexamples (other ratings). Therefore, in OCL for recommendations, the algorithm's objective is to inform if the item should be recommended for the user or not, learning only with data of items recommended for users. We use the One-Class Support Vector Machines (OCSVM) to perform the classification [Tax and Duin 2004]. The OCSVM of [Tax and Duin 2004] classifies a new example of belonging to the interest class if this example is inside a hypersphere. Formally, the center of the hypersphere [Tax and Duin 2004] is:

$$\mu_{(\boldsymbol{c})} = \arg\min_{\boldsymbol{\mu} \in U} \max_{1 \leq i \leq m} \|\varphi(\boldsymbol{ui}_i) - \boldsymbol{\mu}\|^2, \tag{4}$$

in which $\boldsymbol{ui}_i$ is the representation for a user and item for a rating 5, m is the number of examples, U is the feature space associated with the function kernel φ, $\boldsymbol{\mu_{(c)}}$ is the center of the hypersphere in which the greater distance between $\varphi(\boldsymbol{ui}_i)$ to $\boldsymbol{\mu_{(c)}}$ is minimal and $\varphi(\boldsymbol{ui}_i)$ map $\boldsymbol{ui}_i$ into another feature space defined according to the kernel chosen. [Tax and Duin 2004] define the hypersphere:

$$\min_{\mu, \varphi, r} \quad r^2 + \frac{1}{m} \sum_{i=1}^{m} \frac{\varepsilon_{\boldsymbol{ui}_i}}{\nu}, \tag{5}$$

$$\|\varphi(\boldsymbol{ui}_i) - \boldsymbol{\mu_{(c)}}\|^2 \leq r^2 + \varepsilon_{\boldsymbol{ui}_i}, \quad \forall i = 1, ..., m. \tag{6}$$

in which r is the radius of the hypersphere, $\varepsilon_{\boldsymbol{ui_i}}$ is the external distance between $\varphi(\boldsymbol{ui_i})$ and the surface of the hypersphere, and $\nu \in (0, 1]$ defines the smoothness level of the hypersphere volume.

4. EXPERIMENTAL EVALUATION

In the experimental evaluation, we compare the representations generated by our proposal with three baselines in the one-class learning scenario for recommendations considering the OCSVM algorithm and an end-to-end GNN for link prediction baseline. Our goal is to demonstrate that our proposal outperforms other methods used in the user and item representation and the recommendation. The following sections present the dataset, experimental settings, results, and discussion. All source codes and datasets are available[1].

4.1 Dataset

We use the recommendation dataset for movies from [Rana et al. 2022]. The dataset contains 289853 ratings from users of movies. We use the ratings 5 (19668) and 1 (6624) to represent the recommendation and non-recommendation classes. Each movie has reviews, and the dataset with only ratings 1 and 5 contains 1915 users and 1612 movies. Furthermore, the dataset contains the IMDB movie id. Thus, we enriched the dataset with movie metadata to enrich our graph.[2]. We add the movie genre, keywords, and overview.

4.2 Experimental Settings

We use three baselines based on the Bidirectional Encoder From Transformers (BERT) [Devlin et al. 2019]. BERT is a pre-trained neural network that we use for text representation that generates embeddings to represent the text. The BERT model was trained in a very large textual corpus that represents sentences based on their context and outperforms other natural language pre-processing

[1] `https://github.com/GoloMarcos/One-Class-Recommendation-GNN-LinkPrediciton.git`.

[2] We collect the metadata from `https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=keywords.csv&page=2`.

models. In this way, BERT applies correlation techniques, compares the embedding, and extracts semantic and syntactic characteristics from the text [Otter et al. 2020]. We represent the movie reviews with the BERT to generate the baselines. The first baseline is the BERT representation for the movie (BERT-m). The second and third are the BERT representation for the movie concatenated with the user representation generated by the regularization for two graphs: the original graph with only movie-user relations (BERT-m-u-o) and our enriched graph (BERT-m-u-e).

For our graph modeling, the movie node content is the movie overview, and the review node content is the reviews for the movie. Both contents are in a text format. We represent these texts with the BERT embeddings. After representing these nodes, we apply the regularization for all nodes to have representations. Another method that we used to compare with our proposal was the prediction of the GNN, i.e., in an End-To-End (ETE) way. We use our enriched graph to train the GNN-ETE.

For our proposals, GNN-enriched and GNN-original, we use the regularization for all nodes to have an initial representation in our enriched and original graph (only user-items relations). Then, finally, we obtain the embedding with the unsupervised GNN with the link prediction task. In the final step, we classify a recommendation or not through the OCSVM that has as input the concatenation for the user and item representations, except for BERT-m, which uses only the movie representations, and GNN-ETE, which is end-to-end. The parameters for the representation methods and OCSVM were:

—**BERTs**: parameter free;
—**GNNs**: layer sizes = $\{64, 32\}$ and $\{32\}$, epochs = $\{200\}$, patience = $\{20, 50\}$, activation functions = $\{sigmoid, relu, tanh\}$, and learning rates = $\{1^{-3}, 1^{-4}\}$;
—**OCSVM**: kernel = $\{rbf, poly, sigmoid, linear\}$, $\nu = \{0.001, 0.005, 0.01, 0.05\}$ and $0.1 * \nu, \nu \in [1..7]$, and $\gamma = \frac{1}{n}\}$, in which n is the input dimension.

We use the procedure k-Fold Cross-Validation for One-Class Learning. In this procedure, we apply a k-Fold Cross-Validation considering only the interest class (rating 5) since, in the OCL, we have only interest examples labeled. The procedure consists of dividing the class of interest into folds and using $k-1$ folds to train and the remaining fold to test iteratively. We also add the set of not interest (rating 1) in the test set. We chose the $k = 5$ to maintain the test set with interest and non-interest sets containing similar sizes. In addition, we use as test only user-item pairs that are in the train set. Finally, we use the Accuracy (Acc), f_1-score, Precision (P), and Recall (R) as evaluation measures:

$$Acc = \frac{tp + tn}{tp + tn + fp + tn}, \quad (7) \qquad f_1 = \frac{2 \cdot P \cdot R}{P + R}, \quad (8) \qquad P = \frac{tp}{tp + fp}, \quad (9) \qquad R = \frac{tp}{tp + fn}, \quad (10)$$

in which tp (True Positives) is the number of ratings 5 that the OCSVM has correctly classified; tn (True Negatives) is the number of ratings 1 that the OCSVM has correctly classified; fp (False Positives) is the number of ratings 1 incorrectly classified; and fn (False Negatives) is the number of ratings 5 incorrectly classified;

4.3 Results and Discussion

Table I presents the best results considering all parameters used considering the precision, recall, f_1-Score, and accuracy for the OCSVM with the representations generated from BERT and GNN variations and for the GNN-ETE. The best results are in bold font. Each value in the table is the average of all executions for each fold in the k-fold cross-validation.

BERT-m presented the lowest precision, recall, accuracy, and f_1-score values, which indicates that only the movie representation is not enough to recommend a movie to a user. We reinforced this indicative when the BERT results with a representation of users improved the recommendations (BERT-m-u-o and BERT-m-u-e). Another interesting point is the recommendation improvement

Table I. Precision, recall, F_1-Score and Accuracy for the OCSVM considering all representations methods and the end-to-end GNN. Bold fonts indicate best values of the metric.

Methods/Metrics	**precision**	**recall**	**f_1-score**	**accuracy**
BERT-m	0.541±0.007	0.536±0.013	0.527±0.014	0.537±0.013
BERT-m-u-o	0.525±0.006	0.532±0.007	0.525±0.006	0.532±0.006
BERT-m-u-e	0.581±0.009	0.574±0.015	0.553±0.021	0.574±0.015
GNN-ETE	0.703±0.004	0.694±0.013	0.693±0.014	0.694±0.013
GNN-original	**0.729±0.001**	**0.730±0.001**	**0.729±0.001**	**0.730±0.001**
GNN-enriched	0.709±0.002	0.707±0.002	0.708±0.002	0.708±0.002

when the regularization is performed on the enriched graph since BERT-m-u-e got the best results in relation to BERT-m-u-o. On the other hand, the GNN-ETE presented the highest values of precision, recall, accuracy, and f_1-score in relation to the BERT methods with the OCSVM. However, our proposals with the OCSVM outperform BERT methods and the GNN-ETE, considering precision, recall, accuracy, and f_1-score.

One advantage of our proposals besides the highest metrics values is the dimensionality reduction of the representation of the user and item based on the user's review of an item. Our representations have 32 dimensions and BERT 384. The results and advantages reinforce the representation learning relevance for the one-class recommendation. Considering our proposals, GNN-original performed better than GNN-enriched. However, the GNN-enriched performed better than all baselines.

In our intuition, the GNN-original is better since this GNN is based on link prediction trains with real and fake links only between users and items, i.e., with recommendations and not recommendations. On the other hand, the GNN-enriched trains with different types of edges and fake edges. Even with worst metric values than the GNN-original, the GNN-enriched has some advantages. First is the recommendation's explanatory power through the interactions of the different nodes. Second, the greater connectivity of the graph generates paths between users and items, which can be used in other recommender system tasks, such as user clustering based on graph relations.

Another important point that our GNN-enriched can contribute is the exploration of other aspects of the recommendation, such as coverage, novelty, and diversity, since the graph can be enriched with appropriate information to cover these aspects. In addition, these aspects may interest the user, which is advantageous in carrying out the recommendation.

5. CONCLUSIONS AND FUTURE WORK

This study presented the one-class movie recommendation to users, modeling the data through original and enriched graphs and representing the data via Unsupervised GNN for link prediction. Our proposal has the advantages: (i) heterogeneous and enriched modeling that is useful to recommender systems; (ii) easy extension to different types of heterogeneous graphs such as the original or enriched proposed in this study; and (iii) representation learning to obtain more robust representations for one class recommendation. Finally, our method is applicable in other graph modeling contexts where one of the edges or one of the node types is of interest.

Our proposal was limited to a concatenation for the user and item representation, a two-step proposal (representation plus OCSVM), and a link prediction proposal that learns representations by predicting all links in the graph. Therefore, in future work, we will create an OCL end-to-end Heterogeneous GNN for link prediction that uses only user-item relations to predict links while using all edges to learn embeddings. Also, we intend to propose an explainability method for this GNN.

REFERENCES

Alam, S., Sonbhadra, S. K., Agarwal, S., and Nagabhushan, P. One-class support vector classifiers: A survey. *Knowledge-Based Systems* vol. 196, pp. 1–19, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minnesota, pp. 4171–4186, 2019.

do Carmo, P. and Marcacini, R. Embedding propagation over heterogeneous event networks for link prediction. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, online, pp. 4812–4821, 2021.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. *Learning from imbalanced data sets*. Vol. 11. Springer, Switzerland, 2018.

Gôlo, M., Caravanti, M., Rossi, R., Rezende, S., Nogueira, B., and Marcacini, R. Learning textual representations from multiple modalities to detect fake news through one-class learning. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*. ACM, Belo Horizonte, MG, Brazil, pp. 197–204, 2021.

Gôlo, M. P., Araújo, A. F., Rossi, R. G., and Marcacini, R. M. Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning. *Information and Software Technology* vol. 151, pp. 106998, 2022.

Gôlo, M. P. S., Rossi, R. G., and Marcacini, R. M. Learning to sense from events via semantic variational autoencoder. *Plos one* 16 (12): e0260701, 2021.

He, R. and McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. ACM, Republic and Canton of Geneva, Switzerland, pp. 507–517, 2016.

Islam, K., Aridhi, S., and Smaïl-Tabbone, M. A comparative study of similarity-based and gnn-based link prediction approaches. In *GEM (Graph Embedding and Mining) workshop, ECML-PKDD 2020*. HAL, Ghent, Belgium., 2020.

Khoali, M., Laaziz, Y., Tali, A., and Salaudeen, H. A survey of one class e-commerce recommendation system techniques. *Electronics* 11 (6): 878, 2022.

Li, X. and Chen, H. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems* 54 (2): 880–890, 2013.

Otter, D., Medina, J., and Kalita, J. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32 (2): 604–624, 2020.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, Pisa, Italy, pp. 502–511, 2008.

Rana, A., D'Addio, R. M., Manzato, M. G., and Bridge, D. Extended recommendation-by-explanation. *User Modeling and User-Adapted Interaction* 32 (1): 91–131, 2022.

Rossi, R. G., Lopes, A. A., and Rezende, S. O. A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In *Proceedings of the 29th annual acm symposium on applied computing*. ACM, Gyeongju Republic of Korea, pp. 79–84, 2014.

Ru, S., Zhang, B., Jie, Y., Zhang, C., Wei, L., and Gu, C. Graph neural networks for privacy-preserving recommendation with secure hardware. In *2021 International Conference on Networking and Network Applications (NaNA)*. IEEE, Lijiang City, China, pp. 395–400, 2021.

Tax, D. and Duin, R. Support vector data description. *Machine Learning* 54 (1): 45–66, 2004.

Tax, D. M. J. *One-class classification: concept-learning in the absence of counter-examples*. Ph.D. thesis, Delft University of Technology, 2001.

Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys (CSUR)* just accepted (just accepted): 1–37, 2020.

Wu, W., Li, B., Luo, C., and Nejdl, W. Hashing-accelerated graph neural networks for link prediction. In *Proceedings of the Web Conference 2021*. ACM, Ljubljana Slovenia, pp. 2910–2920, 2021.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32 (1): 4–24, 2020.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems* vol. 32, pp. 5753–5763, 2019.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems* vol. 31, pp. 11, 2018.

Zhao, T., McAuley, J., and King, I. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM, Melbourne, Australia, pp. 821–830, 2015.