Attentional Neural Mechanisms for Social Recommendations in Educational Platforms

Italo Zoppis[1][a], Sara Manzoni[1][b], Giancarlo Mauri[1][c], Ricardo Anibal Matamoros Aragon[1], Luca Marconi[2] and Francesco Epifania[2]

[1]*Department of Computer Science, University of Milano Bicocca, Milano, Italy*

[2]*Social Things srl., Milano, Italy*

Keywords: Social Networks, WhoTeach, Social Recommendations, Graph Attention Networks.

Abstract: Recent studies in the context of machine learning have shown the effectiveness of deep attentional mechanisms for identifying important communities and relationships within a given input network. These studies can be effectively applied in those contexts where capturing specific dependencies, while downloading useless content, is essential to take decisions and provide accurate inference. This is the case, for example, of current recommender systems that exploit social information as a clever source of recommendations and / or explanations. In this paper we extend the social engine of our educational platform "WhoTeach" to leverage social information for educational services. In particular, we report our work in progress for providing "WhoTeach" with an attentional-based recommander system oriented to the design of programmes and courses for new teachers.

1 INTRODUCTION

Modern Recommender Systems (RS) use social information to identify the interest of a target user and provide reliable suggestions (Bonhard and Sasse, 2006). These "social" recommendations extend traditional approaches (e.g., collaborative filter) by obtaining compelling information from complex networks of users and items (Zhou et al., 2012) to identify personal interests and implicitly forecast preferences on available items (Gupta et al., 2013; Schafer et al., 2007).

In this context, explainable AI is aimed at providing intuitive explanations for the suggestions and recommendations given by the algorithm (Zhang and Chen, 2018). Basically they try to address the problem of why certain recommendations are suggested by the applied models. The results of this research are progressively shortening the distance between social networks and recommender systems, changing the way people interact and the content they can share (Zhou et al., 2012). At the same time, different attempts in current deep learning literature try to extend deep techniques to deal with social data, recommendations and explanations. Initial work in this context used recursive networks to process structured data such as direct acyclic graphs (Frasconi et al., 1998; Sperduti and Starita, 1997). More recently, Graph Neural Networks (GNNs) and others machine learning techniques (Dondi et al., 2019; Dondi et al., 2016; Zoppis et al., 2019b) have been introduced as a generalization of recursive networks capable of handling more general classes of graphs (Gori et al., 2005; Scarselli et al., 2008). In this regard, emerging researches on deep architectures focus on how to bring out relevant parts of a network to perform a given task (Veličković et al., 2017). Technically, this approach is known as "attentional mechanism" for graphs, or "Graph attention networks" (GATs). Introduced for the first time in the deep learning community in order to access important parts of the data (Bahdanau et al., 2014), the attention mechanism has recently been successful for the resolution of a series of objectives (Lee et al., 2018). For example, it is worth to cite (Chen et al., 2018) where explainable sequential recommendations are extracted due to memory networks. Another interesting approach comes from capsule networks (Li et al., 2019), namely neural networks empowered with capsule structures to manage hierarchies.

[a] https://orcid.org/0000-0001-7312-7123
[b] https://orcid.org/0000-0002-6406-536X
[c] https://orcid.org/0000-0003-3520-4022

Zoppis, I., Manzoni, S., Mauri, G., Aragon, R., Marconi, L. and Epifania, F.
Attentional Neural Mechanisms for Social Recommendations in Educational Platforms.
DOI: 10.5220/0009568901110117
In *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020) - Volume 1*, pages 111-117
ISBN: 978-989-758-417-6

In this article, focusing on these researches, we extend the social engine of our educational platform (Zoppis et al., 2019a) called "WhoTeach" with a graph attentional mechanism aiming to provide social recommendations for the design of new didactical programmes and courses. More in details, we describe the "WhoTeach" platform and its social engine in Sec. 2. In Sec. 3 we consider the main theoretical aspects used in our framework. We introduce the attentional based architecture applied in this work in Sec. 4. In Sec. 5 we report the numerical experiments on a public dataset. Finally, we conclude the paper in Sec.6 by describing future directions of this research.

2 WhoTeach

WhoTeach is an innovative e-learning system, aimed at promoting the development of customized learning and training paths by aggregating and disseminating knowledge created and updated by experts. WhoTeach is conceived as a Social Intelligent Learning Management System (SILMS) and it is structured around three main components:

1. The Recommender System, which helps experts and teachers to quickly assemble high-quality contents into courses: thanks to an intelligent analysis of available material, it suggests teachers the best resources, in any format, to include. Moreover, it helps students, employees and workers to acquire intelligent suggestions. This way, they can find customized courses in real time, taking into consideration their personal profile, objectives and ambitions, along with rules and criteria defined by their belonging organization.
2. The "Knowledge co-creation Social Platform", which is a technological infrastructure based on an integrated and highly interactive social network. This component allows learners to interact and cooperate in the learning path, exchanging information and mutual advice about the contents. That amplifies their learning effort and motivates them to get the best from it. Moreover, it provides experts and teachers with feedback from learners about clarity and comprehension of content and with high-quality comments from domain experts and other teachers (like a peer review). Consequently, it enables the creation and the exchange of knowledge inside organizations at no additional cost. All is done in an informal, unstructured, interactive and pervasive way.
3. The content's repository where to upload contents from any course or training material, either proprietary or open, like the ones promoted by the European Union (for example EMMA or the Open Education portal). Thanks to the use of metadata, the system can build a new, original and high-quality course because WT can identify the right information needed and obtain it from any course (internal e/o external) already available. This mechanism enriches the contents' offer without additional cost, increasing the efficacy of the recommender system that works on a much wider range of data coming from different information sources.

In the current complex and dynamic labour market, every organization needs to deploy a continuous and never-ending learning and formation process in order to keep its working force and stakeholders always updated and ready to leverage innovative technologies, new working paradigms and be proactive versus market changes. For the same reasons, training institutions need to provide always updated classes and materials so that students can acquire skills and competences required by the market.

The challenge for all involved parties in the learning and formation process is huge. Employees, workers and students (learners) feel the need to be able to cope with these changes and want to quickly access to the right, up-to-date and high-quality learning material. They would like to be guided through a specific learning path customized for their personal background, ambitions and objectives. They also require mutual interaction with their peers, teachers and subject matter experts, in order to exchange knowledge and experience as well as to feedback regarding the different contents available.

Subject matter experts and teachers, on the other end, always need to know which learning material and contents are mostly requested by users and where to invest their time. They also seek for high-level comparison and advice from their peers like other teachers, domain experts and even learners, in order to create learning material with the required quality standards that match the learners' needs. To save time and be quick in updating or producing new courses, they would like to easily reuse still valid contents and to be supported in selecting the right course structure.

All organizations and training institutions need to provide training to their learners in a continuous, pervasive way with constantly updated contents. Cost of training is already relevant and will grow steadily not leveraging the accumulated knowledge of all learners or reusing still valid contents for new training material or courses. In addition, it becomes mandatory to easily integrate proprietary contents with free and publicly available training contents.

In this scenario, WhoTeach is a solution then conceived for demanding users, aiming to teach or to learn in highly-dynamic and complex disciplinary environments. Specifically, WhoTeach is the result of the exploitation of the European project NETT, which was focused on the conception and design of a social learning solution for promoting and stimulating the diffusion of the entrepreneurship knowledge and mindset in the European countries. In order to continuously help teachers to conceive, organize and create effective and right courses, WhoTeach provides them with suggestions thanks to the recommender system. Due to the learning algorithms, the system identifies relations between the features of the didactic resources that may be relevant, according to the inquirer's needs. In this way, teachers and experts are supported getting intelligent and consistent suggestions, by reducing the high number of available resources. The system intelligence progressively guides the user in the composition of the course, picking resources in any available format and quickly assembling them to avoid frustration and waste of time.
Moreover, the solution recommends learners the best training that fits their background and their goals in terms of skill improvement and career/curriculum development in the belonging organization, providing them with the most updated and customized content and training material. By the means of the several interaction possibilities available in the social platform, learners can exchange their experience and feedback with experts and other users. Teachers and experts will then receive suggestions on how to improve their contents based on user comments and which contents require update.

The learning algorithms allow WhoTeach to dynamically choose, organize and update contents, given the needs, the objectives and the feedback provided by users. Due to the use of metadata, the contents are homogeneously identified through a vector of parameters to represent them. Thereafter, the feedback of other users (learners or experts) allow to associate each composition of vectors to a score. As a consequence, a dynamical decision tree procedure leads to where, satisfactory completion of the course represented by specific branches of decision trees.
The organization of the learning material is composed by different knowledge areas related to disciplinary macro-areas. Contents are divided in resources, modules and courses. In particular, the resources can vary in their structure (wiki, discussion forum, eBook, etc) and format (word, pdf,etc).

In order to highly empower the social learning processes, especially in relatively new or highly dynamic learning contexts, the second macro-component of the platform structure is the so-called "Knowledge co-creation Social Platform", based on a social network providing multiple possibilities of interactions, material exchange and communities creation, fostered around each disciplinary sector. Especially oriented to students, the platform helps them to make them evolve their personal learning experience, reaching a more collaborative and amazing one. The platform has standard social network tools (like blogs, chat, forum, messaging), plus some advanced features, like the following ones:

- Definition of Community. Their can be created around each disciplinary sector, in particular for teachers, but even for interdisciplinary areas, connecting other different communities. Thematic communities can be freely created by teachers or experts. All communities are moderated by the master of the knowledge area. In this way, disciplinary aggregations led to communities of practice and knowledge exchange and improvement.
- Sharing of Didactic Materials. Users can share all the didactic material, according to their need. In particular, there is the possibility to share non official material, without waiting experts' or masters' approvals. In this way, other feedback can be acquired, thus making the knowledge management and evolution faster.
- Informal Communication among Users. Private or public feedback progressively help teachers or experts in improving their materials. Feedback are stimulated in the knowledge areas but also among different areas or disciplines, so as to get insights for knowledge evolution and disciplinary innovation.
- Users Profile. Teachers and students have the possibility to create and edit their own profile: there is a wide range of possibilities to share personal experience, academic background, interests, competences and personal or professional ambitions. Organizations can also interact directly in order to stimulate talent selection and competences matching.
  Groups and Forums. They have several possibilities of interactive features to stimulate cooperation among users. Besides directly sharing materials, teachers and students can create or share different kind of contents in a dynamic way (e.g. tests, links, videos, exercises etc.).

Therefore, this kind of social platform is aimed at stimulating various types of collaborations among users. That gives rise to rich, efficient and fruitful communities of practice to favor course design activities and allow peer-to-peer learning methodologies.

From a technical perspective, the system is based on a PHP shell piloting and empowering the customization of the Moodle platform, serving as a base for a Content Management System, while Mahara is used to build the nested social platform. Mahara is a fully-featured web application to build an electronic portfolio. A user can create journals, upload files, embed social media resources from the web and collaborate with other users in groups. In Mahara user can control which items and what information other users see within their portfolio. The Moodle system was chosen because of its high diffusion within basically any kind of training institution and due to the presence of a wide development community. The platform has been then integrated with social network features coming from Mahara, in order to introduce meta-services as previously described.

3 MAIN CONCEPTS AND DEFINITIONS

Graphs (annotated with $G = (V,E)$) are theoretical objects widely applied to model the complex set of relationships that typically characterize current networks. They consist of a set of "entities ", V (vertices or nodes), and relationships between them, i.e. edges, E. In this paper, we use weighted graphs (graphs whose edges $\{i, j\} \in E$ have assigned weights $label(i, j)$), with an associated graph adiacency matrix, A, to indicate whether two vertices are connected by some edge, i.e., $A_{i,j} = label(i, j)$ if $\{v_i, v_j\} \in E$. Moreover, given a vertex $v \in V$, we indicate with $\mathcal{N}(v) = \{j : \{v, j\} \in E\}$ the neighborhood of the vertex v. We will also indicate with $G[A]$ the graph induced by A. In order to summarize the relationships between vertices and capture relevant information in a graph, embedding (i.e., objects transformation to lower dimensional spaces) is typically applied (Goyal and Ferrara, 2018). This approach allows to use a rich set of analytical methods offering to deep neural networks the capability of providing different levels of representation. Embedding can be defined at different level: for example, at node level, at graph level, or even through different mathematical strategies. Typically, the embedding is realized by fitting the (deep) network's parameters using standard gradient-based optimization. The following definitions can be useful (Lee et al., 2018).

Definition 3.1. Given a graph $G = (V,E)$ with V as the set of vertices and E the set of edges, the objective of node embedding is to learn a function $f : V \to \mathcal{R}^k$ such that each vertex $i \in V$ is mapped to a k-dimensional vector, $\vec{h}$.

Definition 3.2. Given a set of graphs, $\mathcal{G}$, the objective of graph embedding is to learn a function $f : \mathcal{G} \to \mathcal{R}^k$ that maps an input graph $G \in \mathcal{G}$ to a low dimensional embedding vector, $\vec{h}$.

4 A GRAPH ATTENTION MECHANISM FOR RECCOMENDER SYSTEMS

The principle according to which the attention mechanisms play their role is to select the most relevant information among those available for the neural response computation. In other words, "attention" is essentially a way to non-uniformly weight the contributions of input - or part of it, in order to optimize the learning process for some specific task.
There are many way to get this result. In this paper we consider the case of node embedding–based attention, as proposed in (Veličković et al., 2017).
Let us consider an user/item relationship matrix A, and the corresponding weighted graph, $G[A] = (V,E)$, whose edges are labeled with scores attributed by users, $U \subseteq V$, to resources, $R \subseteq V$, and collected within A. Given a pair of vertices $(u,r), u \in U, r \in R$, the induced graph representation of A has an edge between user u and resource r in case that u applied r and evaluated such a resource with the score $label(u,r)$. In this way, we have $label(u,r) = A_{u,r}$.
With the above notation, we conveniently adapt the definition of "attention" reported in (Lee et al., 2018) as follows.

Definition 4.1. Let A be an user/item relationship matrix, and $G[A] = (U \cup R, E)$ its induced weighted graph with vertices equal to the union of users, U, and items R, respectively.
Given $(u,r), u \in U, r \in R$, an attentional mechanism for G is a function $a : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$ which computes coefficients $e^{(l)}_{u,r} = a\big(\vec{h}^{(l)}_u, \vec{h}^{(l)}_r\big)$ across the pairs of vertices, (u,r), based on their feature representation $\vec{h}^{(l)}_u, \vec{h}^{(l)}_r$ at level l.

Coefficients $e^{(l)}_{u,r}$ are parameters which act by weighting the relevance of the vertex r's features to (user) u. Following (Veličković et al., 2017), we define a as a feed-forward neural network with a learnable vector of parameters (i.e., weights) $\vec{a}$ - updated with the others network's parameters according to standard optimization, and nonlinear *LeakyReLU* activation function. In this way, we have

$$e^{(l)}_{u,r} = \text{LeakyReLU}\Big(\vec{a}^{(l)^T}\Big[\mathbf{W}^{(l)}\vec{h}^{(l)}_u || \mathbf{W}^{(l)}\vec{h}^{(l)}_r\Big]\Big), \quad (1)$$

where $\mathbf{W}$ is a learnable parameter matrix and $\mathbf{W}^{(l)}\vec{h}_u^{(l)}||\mathbf{W}^{(l)}\vec{h}_r^{(l)}$ is the concatenation of the embedded representation for the vertices u, r.
The coefficients $e_{u,r}^{(l)}$ can be normalized over all elements in the neighborood of u. For instance, we can apply the *softmax* function to obtain the following expression.

$$\alpha_{u,r}^{(l)} = \frac{\exp(e_{u,r}^{(l)})}{\sum_{k\in\mathcal{N}(u)}\exp(e_{u,k}^{(l)})}.$$

At this point, the role of (coefficients) $\alpha_{u,r}^{(l)}$ becomes fundamental for our extension. Let us consider the induced graph $G[A] = (U \cup R, E)$, and focus on some user (vertex) $u \in U$. When only resources (items) around u are considered, we can use the normalized (attention) coefficients $\alpha_{u,r}^{(l)}$ to compute a combination of (the embedded resources) $\vec{h}_r^{(l)}$ in $\mathcal{N}(u)$ as follows.

$$\vec{h}_u^{(l+1)} = \sigma\big(\sum_{r\in\mathcal{N}(u), r\in R} \alpha_{u,r}^{(l)}\mathbf{W}^{(l)}\vec{h}_r^{(l)}\big) \quad (2)$$

where σ is non linear vector-valued function (sigmoid). With this formulation, Eq. 2 provides the next level embedding for user u scaled by the attention coefficients which, in turn, can be interpreted as the relevance of the resources used by (user) u (i.e., resources in the neighborood of u). Similarly to Eq. 2, the following quantity can be interpreted as the embedded representation for the resource r scaled by the attention coefficients which weight users (representation) who applied r (i.e., users in the neighborood of r).

$$\vec{h}_r^{(l+1)} = \sigma\big(\sum_{u\in\mathcal{N}(r), u\in U} \alpha_{u,r}^{(l)}\mathbf{W}^{(l)}\vec{h}_u^{(l)}\big) \quad (3)$$

In this way, the "GAT layer" returns for each pair $(u, r) \in U \times R$ the embedded representation $(\vec{h}_u^{(l+1)}, \vec{h}_r^{(l+1)})$. In our experiments we will consider only one level of embedding, i.e., $l = 1$.

4.1 A Stacked Architecture for Social-based Recommendations

The attentional mechanism described in this paper was applied as a "base" layer (Module A) for the stacked architecture reported in Fig. 1. Two outputs are provided: $\vec{h}_i^{(l+1)}$ i.e., the embedded representation for user-u's score, and $\vec{h}_j^{(l+1)}$, the embedded representation for resource-r's scores, respectively. The following details summarize this layer.

- **Architecture:** described in the previous paragraph.

  Input: Given the user/item matrix, $A = (s_{i,j})$, which contains the score, $s_{i,j}$, for each user i and resource (item) j, a training set of examples $\mathcal{T} = \{((\vec{u}_i, \vec{r}_j), s_{i,j}) : 1 \le s_{i,j} \le 5\}$ was obtained by composing the vector of scores, $\vec{u}_i$ (provided by user i for each available resource), and the vector, $\vec{r}_j$ (scores attributed by all users to the resource j).
  Output: For each i and j, the embedded user-i's scores $\vec{h}_i^{(l+1)}$ and item-j' s scores $\vec{h}_j^{(l+1)}$.

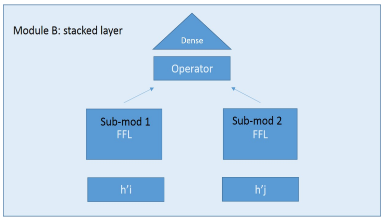
Two embedding, $\vec{h}_i^{(l+1)}$, $\vec{h}_j^{(l+1)}$ are then passed and combined through feed forward levels (FFL) in order to obtain, using a final sigmod-based activation, the score predicted for the user/item (input) pair (i, j). The whole model is trained with MSE loss and SGD (stochastic gradient descent) optimizer. In particular the following general architecture (Fig. 1a) was stacked on the top of the attention layer.

- Stacked layer (Module B in Fig. 1a).
  Input: $\vec{h}_i^{(l+1)}, \vec{h}_j^{(l+1)}$.
  Output: For each i and j, the predicted score, $\hat{s}_{i,j}$, for user i when choosing (resource) j.
  FFL, Sub-mod. n.1: $\vec{h}_i^{(l+1)}$ as input + Dense layer with ReLU Activation function + h_i as embedded output representation for user-i's scores.
  FFL, Sub-mod. n.2: $\vec{h}_j^{(l+1)}$ as input + Dense layer with ReLU Activation function + h_j as embedded output representation for resource-j's scores.
  Operator Layer: $\vec{h}_i^{(l+1)}$ and $\vec{h}_j^{(l+1)}$ are combined (through a specific operator) to obtain the vector $(\vec{h}_i^{(l+1)}, \vec{h}_j^{(l+1)})$.
  Dense Layer: This final layer uses an output sigmoid activation. The value $\hat{s}_{i,j}$ assumed by the sigmoid function is then interpreted as output score for user i and resource j.

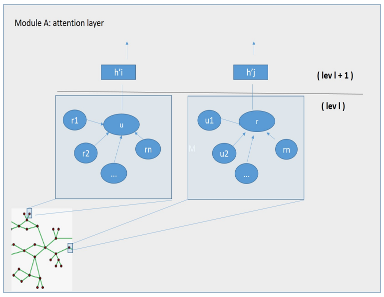
Note that the architecture described above shows a general structure, and can provide different models according to the type of operation applied by the considered "operator layer".

5 NUMERICAL EXPERIMENTS

Numerical experiments use an homogeneous set of data whose characteristics combine well with the requirements of the WhoTeach platform. These data come from the "Goodbooks" data-set (https://www.kaggle.com/zygmunt/goodbooks-10k), a large collection reporting up to 10000 books and 1000000 ratings assigned by 53400 readers. In particular, numerical ratings, ranging from "1" to "5", are given by



(a)



(b)

Figure 1: System Architecture. A feed-forward based network (Module B) is stacked on the top of the GAT layer (Module A). The embedding of user-u's scores $\vec{h}_i^{(l+1)}$, and item-r's scores $\vec{h}_j^{(l+1)}$, are passed and combined ("operator layer" in Mod. B) with further embedding (FFL Submodules) in order to provide through a sigmod-based activation (output of "Dense layer") the predicted final suggestion.

users (readers) for each resource type, in this case, different sort of book. Numerical experiments are planed to evaluate the capability of the attentional-based models to reduce error (loss function) between the reported and predicted preference scores. To provide robust estimation, we sub-sampled the data using cross-validation. The models described in this paper was implemented using the Pytorch library (https://pytorch.org/), and then executed using different parameters for early stopping and learning rate, on COLAB (https://colab.research.google.com/). In this work in progress the attention-based model with concatenation operator in the stacked layer (see Fig. 1) was compared with the following alternative models. Performances were averaged on the number of folds (10 cross-validation).

1. Dot product model.
   Input: Training set $\mathcal{T} = \{((u_i, r_j), s_{i,j}) : 1 \leq s_{i,j} \leq 5\}$ as described previously for the attentional based architectures.
   Output: for each i, j, the score $\hat{s}_{i,j}$ recommended for user i and resource j.
   Loss Function: MSE; **Optimizer**: SGD.
   Architecture: similar to the stacked architecture (with no attention). A dot product operation is applied to "aggregate" the embedded representations of $\vec{u}_i$ and $\vec{r}_j$. No learnable parameters are considered.

Table 1: MSE comparison: Attention is applied with the concatenation operator at the stacked layer. Hadamard uses element-wise product between two vectors at the operator layer.

Model	Attention	Concatenate	Multiply	Hadamard
MSE	0.0389	0.0439	0.0437	0.0436

2. Element-wise product model (Hadamard product model).
   Similar to the previous case (dot product model) but with an element wise product operation (Hadamard product between vectors) computed by the Operator Layer. The result of the element-wise operation is is passed to the final dense layer.
3. Concatenation model.
   Similar to the dot product model but with a concatenation operation computed by the Operator Layer. In this case, the embedded representations of $\vec{u}_i$ and $\vec{r}_j$ are concatenated in a new latent vector and finally passed to a dense layer.

Preliminary results are reported in Tab. 1. A general better tendency to reduce the MSE loss is observed when attention layer with concatenation is applied as a base module for the considered stacked layer.

6 CONCLUSIONS

Online social spaces are rich sources of information for modern recommender systems. However, the success of reliable recommendations is related to both the capability of a framework to capture the social content and the availability of an effective information in the considered online social space. The work reported in this paper has focused on a recent "mechanism" formulation for learning on graphs with "attention" (Veličković et al., 2017). In particular, the proposed architecture intends to benefit from exploiting (with "attention" weights) the graph's task-relevant part in order to provide reliable social recomendations. As recently reported (Seo et al., 2017; Lu et al., 2018), these mechanisms constitute challange solutions to provide users with effective (social) justifications for the suggestions that modern recommendation systems are able to offer. Our research will follow now this target by adding to the social engine of our "WhoTeach" platform textual suggestions based on the computed "attention weights" as defined in Sec. 4. Numerical results obtained in the experiments are encouraging.

The proposed framework outperform feed forward-based networks when the attention layer is applied. Our future research in this context will also consider additional architectures and data for further evaluations and more general conclusions.

REFERENCES

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bonhard, P. and Sasse, M. A. (2006). 'knowing me, knowing you'—using profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98.

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., and Zha, H. (2018). Sequential recommendation with user memory networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 108–116, New York, NY, USA. ACM.

Dondi, R., Mauri, G., and Zoppis, I. (2016). Clique editing to support case versus control discrimination. In *Intelligent Decision Technologies*, pages 27–36. Springer.

Dondi, R., Mauri, G., and Zoppis, I. (2019). On the tractability of finding disjoint clubs in a network. *Theoretical Computer Science*.

Frasconi, P., Gori, M., and Sperduti, A. (1998). A general framework for adaptive processing of data structures. *IEEE transactions on Neural Networks*, 9(5):768–786.

Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE.

Goyal, P. and Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94.

Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., and Zadeh, R. (2013). Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 505–514. ACM.

Lee, J. B., Rossi, R. A., Kim, S., Ahmed, N. K., and Koh, E. (2018). Attention models in graphs: A survey. *arXiv preprint arXiv:1807.07984*.

Li, C., Quan, C., Peng, L., Qi, Y., Deng, Y., and Wu, L. (2019). A capsule network for recommendation and explaining what you like and dislike.

Lu, Y., Dong, R., and Smyth, B. (2018). Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference*, pages 773–782.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Seo, S., Huang, J., Yang, H., and Liu, Y. (2017). Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 297–305.

Sperduti, A. and Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Zhang, Y. and Chen, X. (2018). Explainable recommendation: A survey and new perspectives.

Zhou, X., Xu, Y., Li, Y., Josang, A., and Cox, C. (2012). The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37(2):119–132.

Zoppis, I., Dondi, R., Manzoni, S., Mauri, G., Marconi, L., and Epifania, F. (2019a). Optimized social explanation for educational platforms. In *Int. Conf. on Computer Supported Education*, volume 1, pages 85–91.

Zoppis, I., Manzoni, S., and Mauri, G. (2019b). A computational model for promoting targeted communication and supplying social explainable recommendations. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 429–434. IEEE.