Social Neural Hybrid Recommendation with Deep Representation Learning

Presented by: Dr Lamia Berkani
Dyhia Laga
Abdelhak Aissat

Laboratory for Research in Artificial Intelligence (LRIA)
Dep. of Computer Science, USTHB University

MEDI 2021 – Tallinn, Estonia, June 21-23, 2021

Outline

- Introduction
- Background & related work
- Our approach : Social Neural Hybrid Recommendation (SNHF)
  - Description of the different layers of the SNHF model
  - Training
- Experiments
- Conclusion & perspectives

RECOMMENDER SYSTEMS

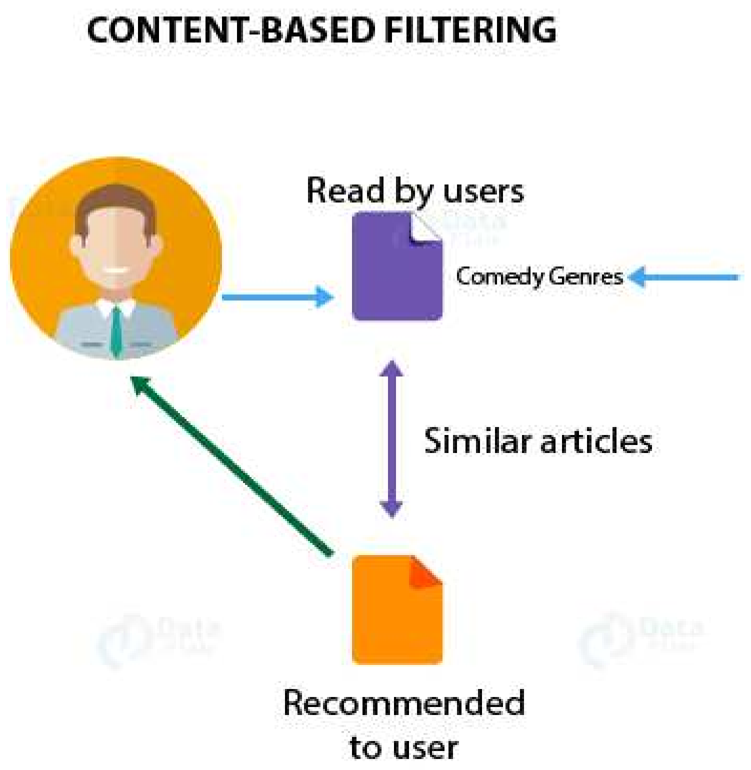
- Recommender systems (RSs) are the most successful and most popular applications of data science.

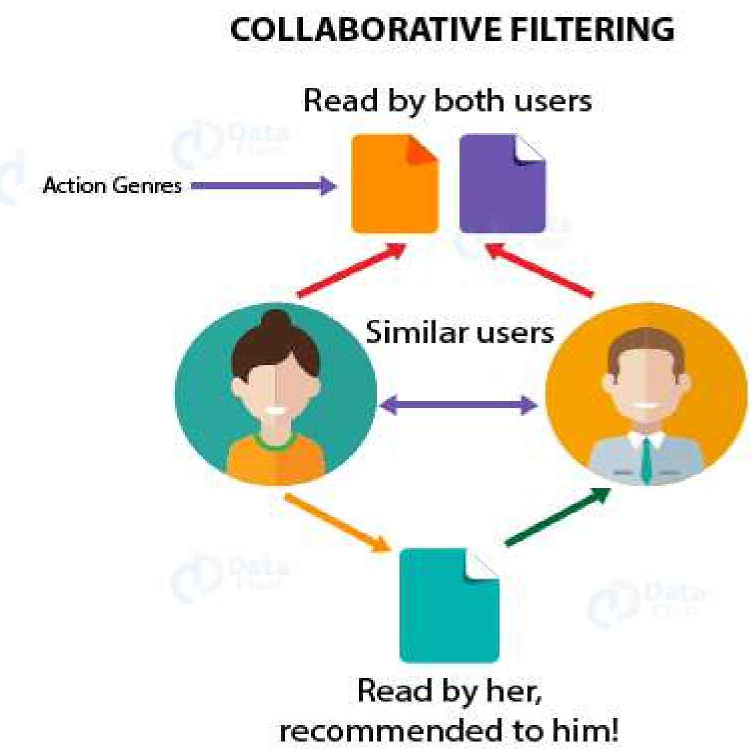
- RSs are self-explanatory algorithms that leverage historical data to recommend / suggest a particular product, service or person, by inferring correlation strength between them.

Recommender systems (RSs) - Approaches

Content-based filtering (CBF), based on a comparison of user profiles with item profiles to recommend items that match their preferences and tastes.

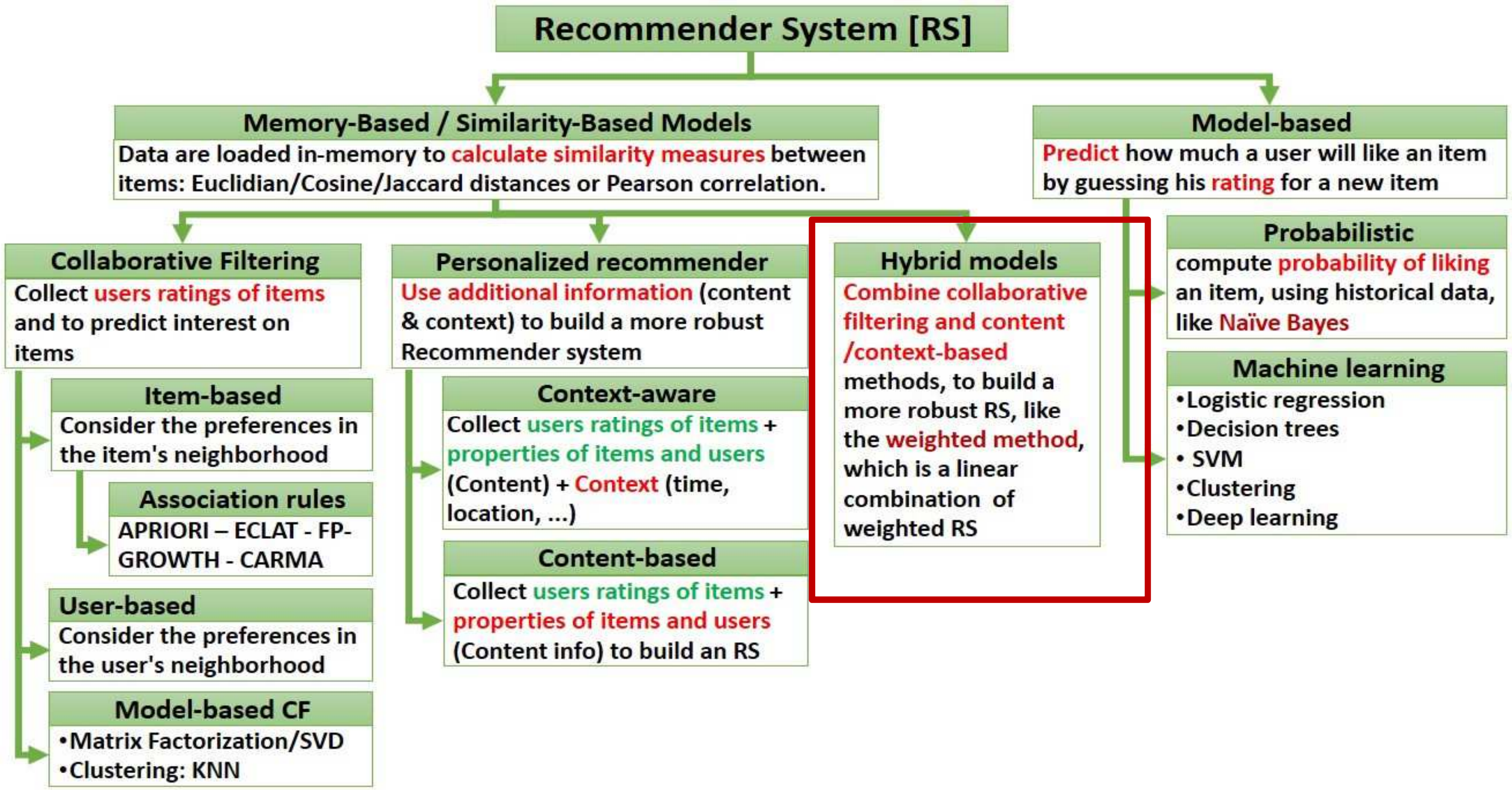
Collaborative filtering (CF), based on the principle that a prediction of a given item can be generated by aggregating the ratings of like-minded users;

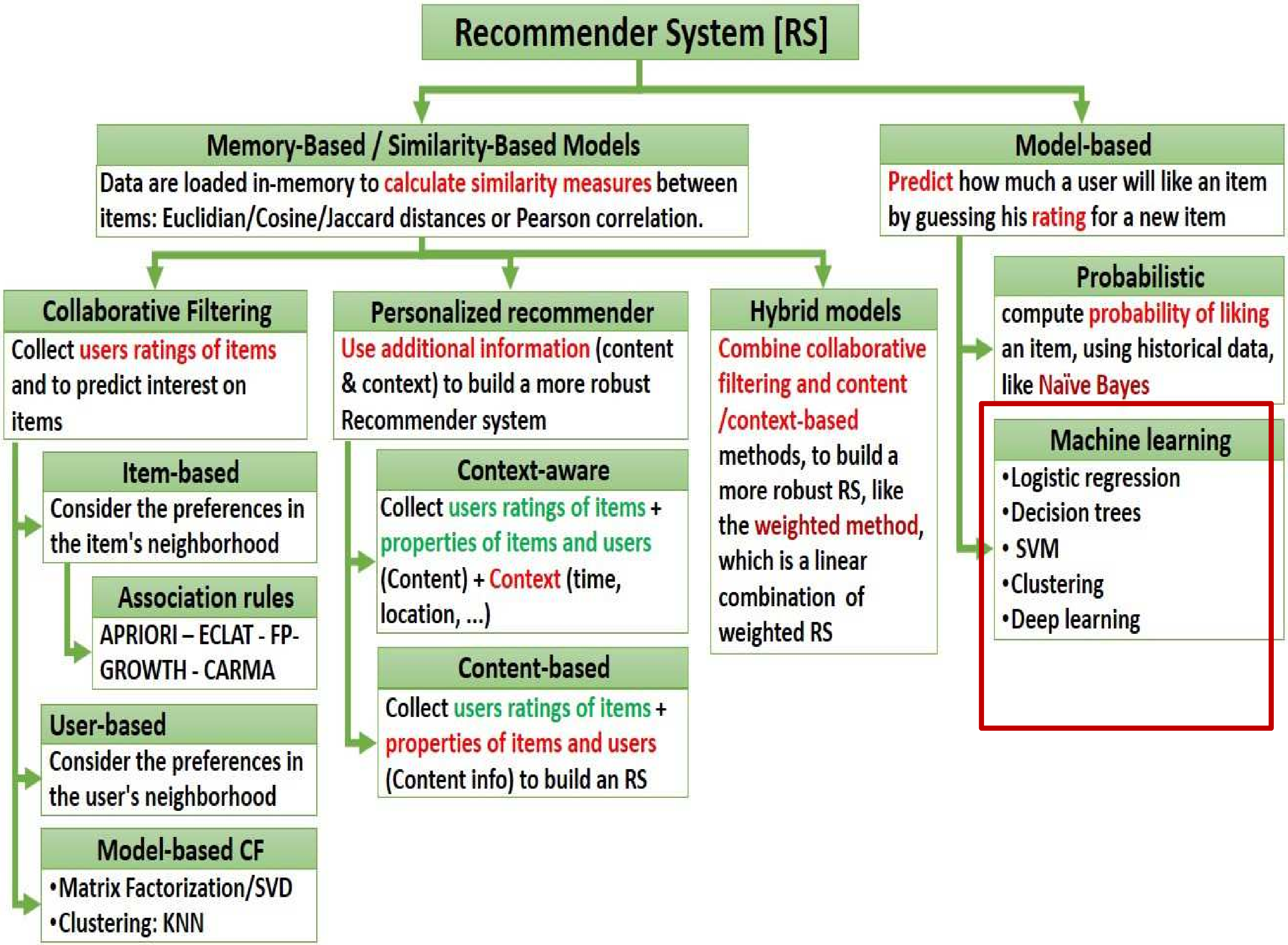




RSs A – Approaches

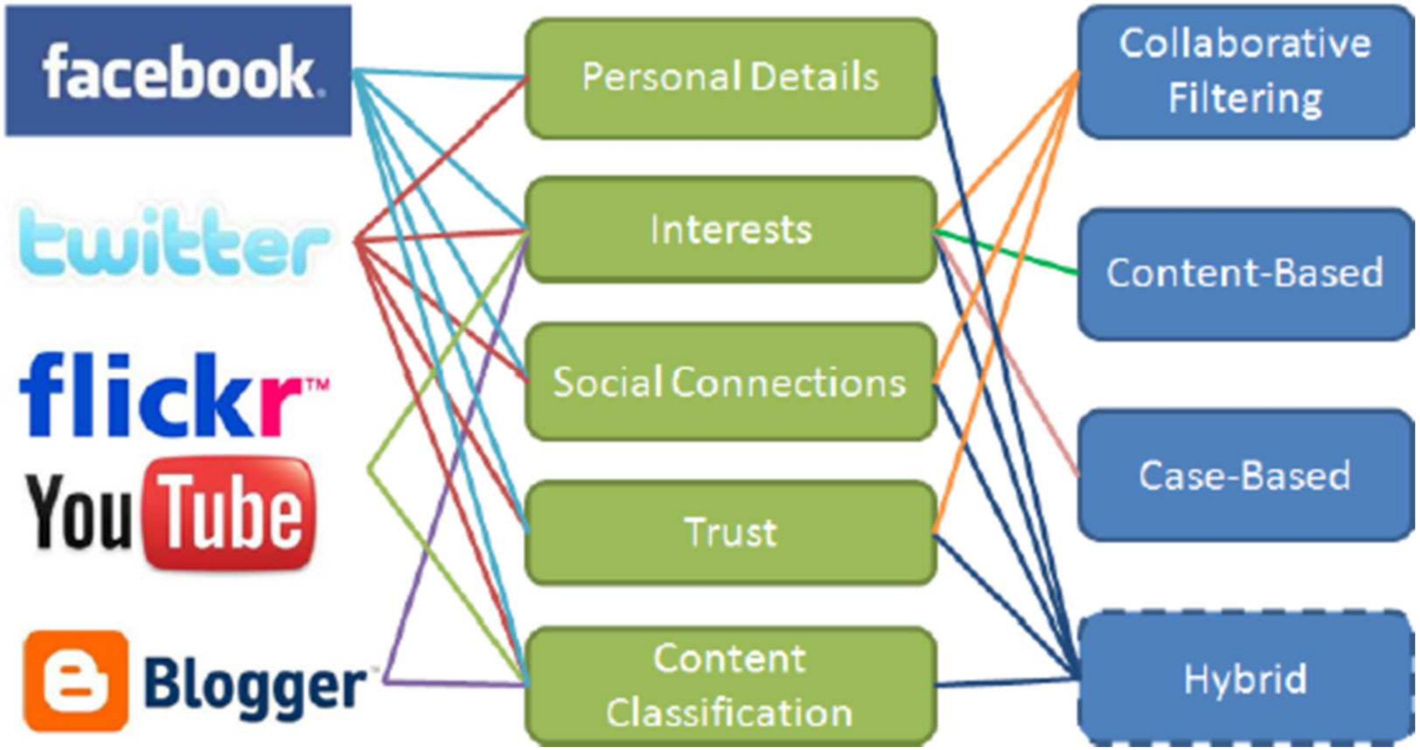
- Hybrid recommender systems combine the two recommendation techniques in different ways [1]
  - To overcome their drawbacks (cold start & sparsity problems)




Recommender System [RS]
Memory-Based / Similarity-Based Models
Data are loaded in-memory to calculate similarity measures between items: Euclidian/Cosine/Jaccard distances or Pearson correlation.
Model-based
Predict how much a user will like an item by guessing his rating for a new item
Collaborative Filtering
Collect users ratings of items and to predict interest on items
Personalized recommender
Use additional information (content & context) to build a more robust Recommender system
Hybrid models
Combine collaborative filtering and content /context-based methods, to build a more robust RS, like the weighted method, which is a linear combination of weighted RS
Probabilistic
compute probability of liking an item, using historical data, like Naïve Bayes
Machine learning
•Logistic regression
•Decision trees
• SVM
•Clustering
•Deep learning
Item-based
Consider the preferences in the item's neighborhood
Association rules
APRIORI – ECLAT - FP-GROWTH - CARMA
User-based
Consider the preferences in the user's neighborhood
Model-based CF
•Matrix Factorization/SVD
•Clustering: KNN
Context-aware
Collect users ratings of items + properties of items and users (Content) + Context (time, location, ...)
Content-based
Collect users ratings of items + properties of items and users (Content info) to build an RS

RSs and the Social web

- With the expansion of social media platforms, the performance of traditional recommender systems can be improved with the integration of social information



INTRODUCTION

- Deep learning techniques are proved to be very effective in various domains, such as computer vision, pattern recognition, and natural language processing
- More and more deep models have been used in the recommendation system
- Only few attempts have been made in social-based recommender systems

We focus on this issue and explore the use of deep neural networks for learning the interaction function from data.

Related work

- The state of the art shows that some neural network-based methods are proposed to learn user and item representations from text notation and comments data [2; 3].
- He et al. [4] modeled the user-item assessment matrix using a multilayer feedback neural network.
- He et al. [5] proposed an approach based on CF and CNN.
- Zheng et al [6] combined all user comments and items and applied the CNN to jointly model user preferences and item characteristics.

RELATED WORK

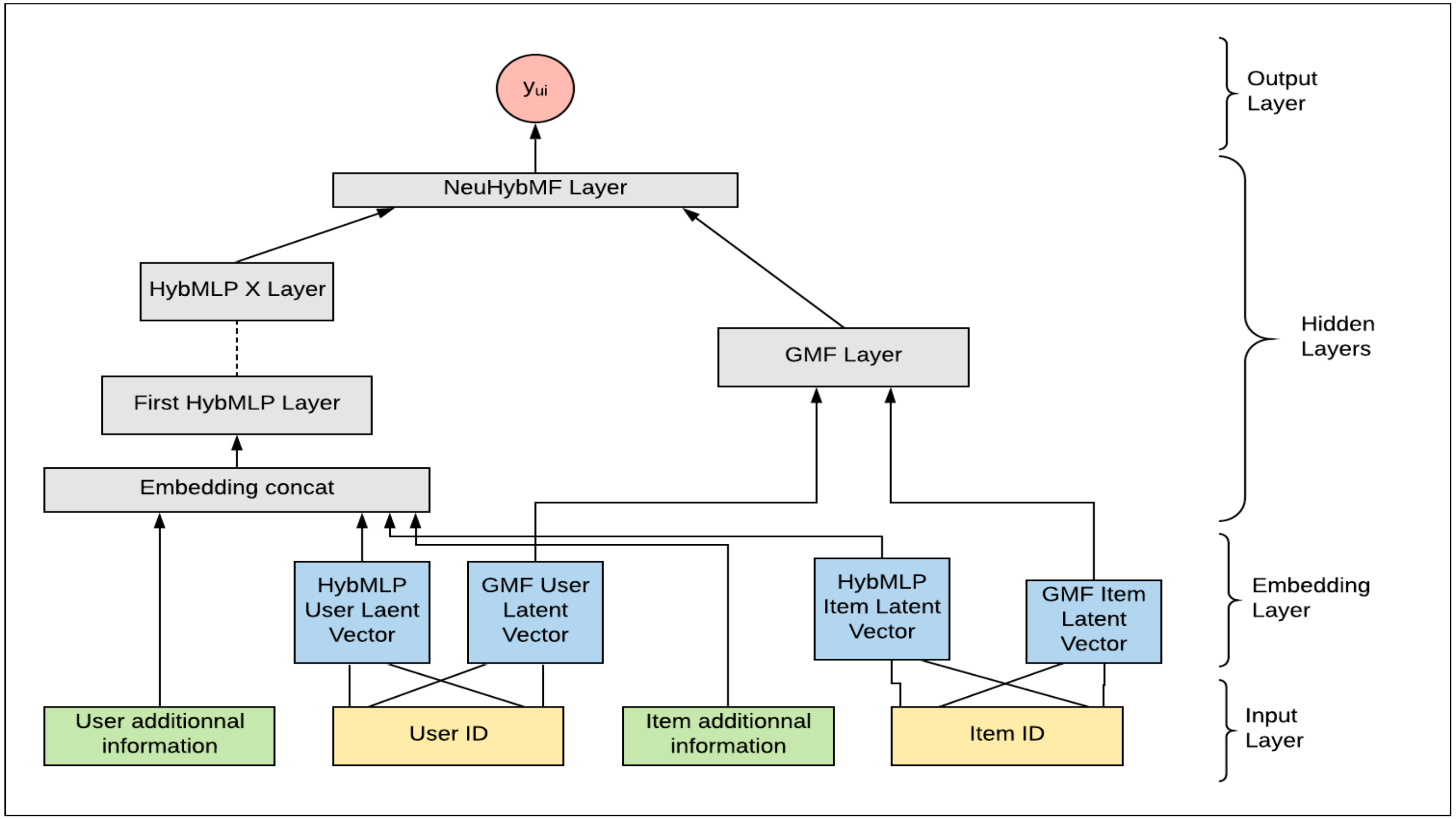
- Lu et al. [7] proposed a model of mutual learning between notes and comments, given that the method of modeling notes is based on PMF, which can only learn linear characteristics.

- Liu et al [8] used a MLP network in a note-based coder to learn deeper and higher-level features from note models.

- Berkani et al [9] proposed an extension of the NCF model [4], considering a hybridization of CF and CBF based on: (1) Generalized Matrix Factorization (GMF); and (2) Hybrid Multilayer Perceptron (HybMLP).

Related work

- Neural Hybrid Recommendation Based on GMF and Hybrid MLP [9]

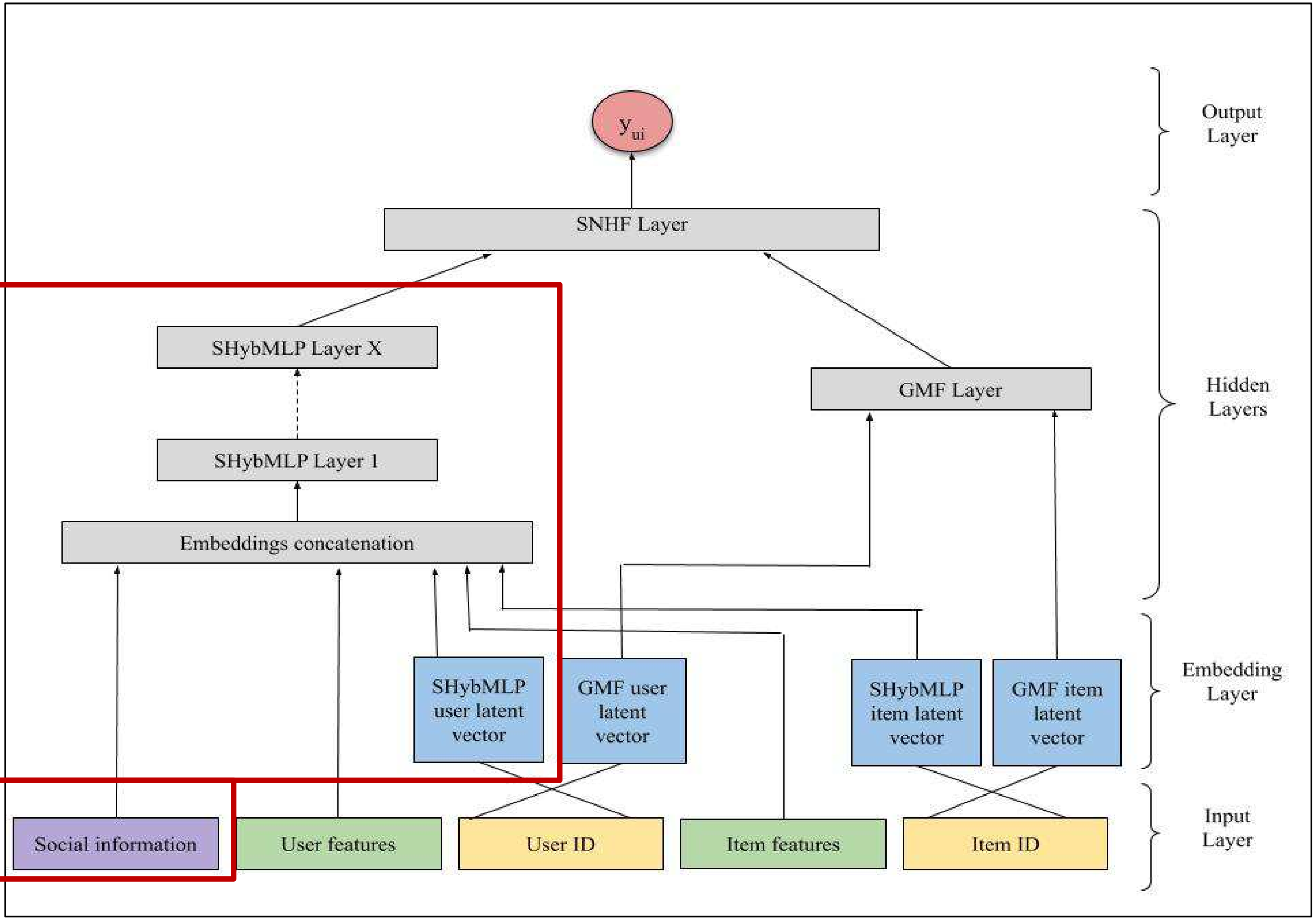


Our Approach : Social Neural Hybrid Recommendation (SNHF)

- We propose, a novel hybrid method called SNHF (Social Neural Hybrid Filtering), combining CF and CBF algorithms and using **social information** (friendship and trust between users) in the same architecture based on GMF and HybMLP (Hybrid MLP) models.

- Friendship and trust between users could significantly improve the quality of recommendation,

OVERVIEW OF THE PROPOSED SNHF METHOD



DESCRIPTION OF THE DIFFERENT LAYERS

- **Input Layer.** In addition to the NHybF model entries [9], i.e. user and item IDs with corresponding characteristics, we also consider the social information of each user (friends and trusted persons)

INPUT LAYER

- The friends list of a given user u includes all of his friends from the social network.
- The friendship degree is calculated using the Jaccard formula:

$$Friendship\,(u,v) = \frac{|F_u \cap F_v|}{|F_u \cup F_v|} \quad (1)$$

where: F_u is the set of friends of u and F_v is the set of friends of v.

- The friendship degree is calculated as follows:

$$D_{Friendship(u,v)} = 1 - Friendship\,(u,v) \quad (2)$$

INPUT LAYER

- Trust describes the degree of trust of a user u towards a user v.

- The calculation of the degree of trust is done in two steps: (1) calculation of the degree of trust directly between two users; and (2) propagation of the trust through the trust network.

INPUT LAYER

- The degree of trust between two directly related users is calculated according to the following Tanimoto formula:

$$D_{Trust(u,v)} = \frac{1}{deg(u) + deg(v) - 1} \qquad (3)$$

where:
$deg(u)$: is the number of users that user u trusts, including v; and
$deg(v)$: is the number of users that user v trusts, including u.

Input Layer

- We used the MoleTrust algorithm [10], to compute the trust degree between a source user and a target user, by browsing the trust graph and propagating it along the arcs.

- The social degree between two users is calculated according to this weighted formula:

$$D_{social} = \alpha * D_{Friendship(u,v)} + \beta * D_{Trust(u,v)} \qquad (4)$$

where: α, and β: represent the importance weights related, respectively, to friendship and trust, with: $\alpha + \beta = 1$.

EMBEDDING LAYER

- The embedding layer allows converting and representing discrete or categorical variables into a vector of continuous numbers.
- A common use in NLP is to take a word and apply Word Embedding to make it denser.
- The words are encoded in a binarized sparse vector with one-hot encoding.
- The goal : forming less sparse vectors with a logical relationship between them.

GMF - Linear model

- This part represents the operation made by the GMF to calculate the predictions.

- The textual description of the model layers is given as follows:
  - *GMF User Embedding*: User latent factor vectors.
  - *Embedding of GMF items*: Vectors of latent factors of the items.
  - *Multiplication layer*: ensures the element by element multiplication of the user and item embedding (factors)

Social Hybrid Multilayer Perceptron (SHybMLP) - Nonlinear Model

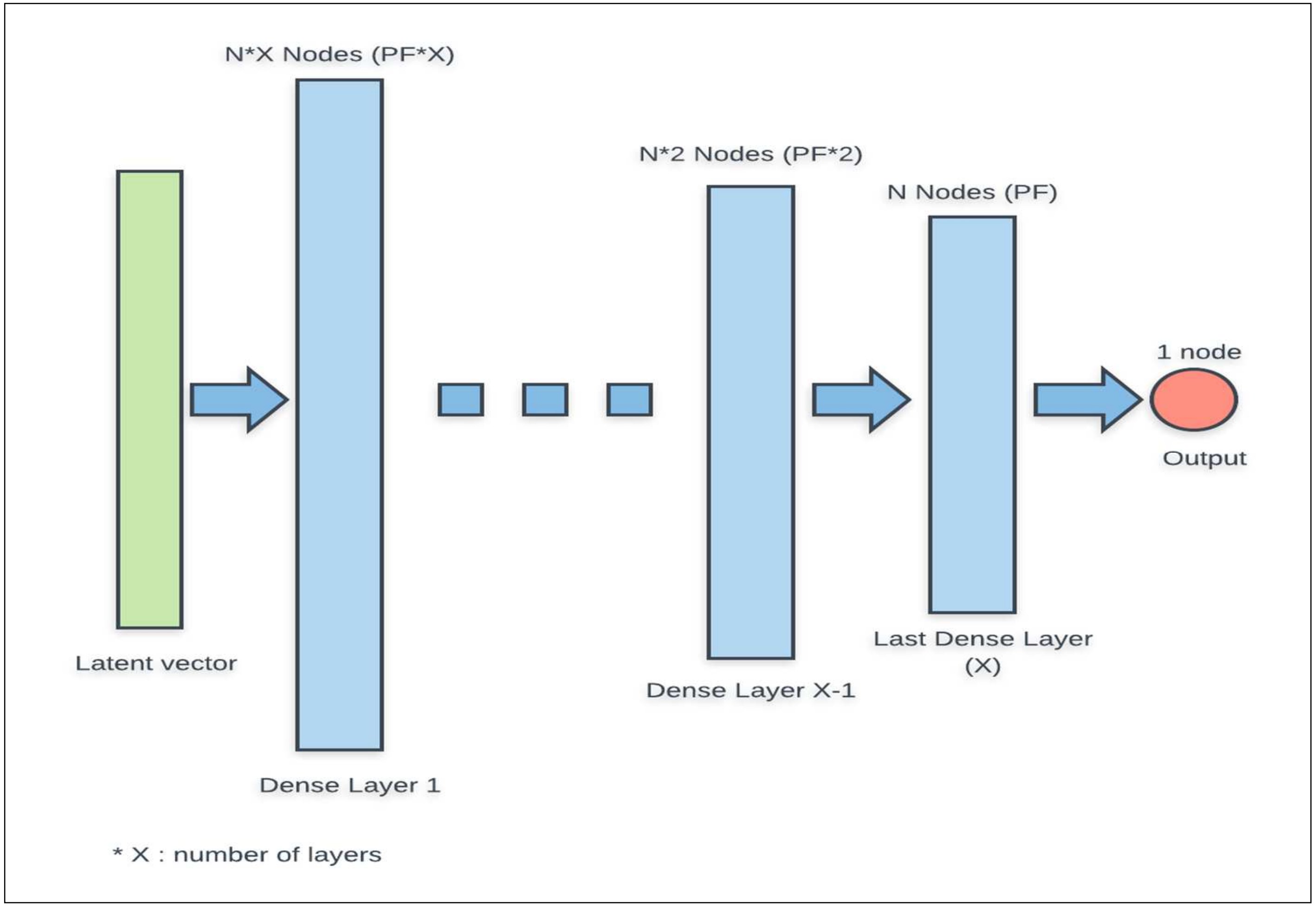
- This part deals with the learning of the interaction function which returns "1" if there is an interaction between a user u and an item i and "0" otherwise.

SHYBMLP

The textual description of the different layers of the SHybMLP model is given as follows:

- *Embedding of SHybMLP users:* Vectors of user latent factors.
- *Embedding of SHybMLP items:* Vectors of items latent factors.
- *Concatenation Layer:* allows the concatenation of embeddings with user and item characteristics as well as social information related to each user.

SHybMLP


N*X Nodes (PF*X)
N*2 Nodes (PF*2)
N Nodes (PF)
1 node
Output
Latent vector
Dense Layer 1
Dense Layer X-1
Last Dense Layer (X)
* X : number of layers

SHybMLP

- The activation function used in each hidden layer is *ReLU* (Rectified Linear Unit:

  ReLU(x) = max(0,x)), to reduce the risk of over-fitting and neuron saturation.

$$\phi^{SHybMLP} = a_L(W_L^T(a_{L-1}(\ldots a_2(W_2^T[\, p_u^M \; f_u \; f_s \; q_i^M \; f_i \,] + b_2)\ldots)) + b_L) \qquad (6)$$

where:

a_L : Layer activation function L;

b_L : Bias of the layer L (has the same role as the threshold);

W_L^T : Weight of the layer L;

p_u^M : Latent factors vector of the user *u* of SHybMLP;

f_u : Information vector of the user *u*;

q_i^M : Latent factors vector of the item *i* of SHybMLP; and

f_i : Information vector of the item *i*.

Social Neural Hybrid Matrix Factorization (SNHF)

- The last layer of SHybMLP is concatenated to that of GMF so that the combination of the results of these models would lead to better predictions.

- The training of this model can be done in two ways: with or without the pretraining of the models composing it (GMF and SHybMLP).

OUTPUT LAYER

- This layer takes as input the vectors of the last SHybMLP and GMF layers previously concatenated in the SNHF layer, then passes them through the Sigmoid activation function:

$$\sigma(x) = 1/(1 + e^{x})$$

Training - Dataset management

- Two databases are used in our experiments: Yelp and FilmTrust.
- Once the transformation of the data is done, the dataset will be divided into two parts:
  - *Training data*, which will serve as learning data for the model (experiments will help to find the right balance between positive and negative instances);
  - *Testing data*, which will be used to evaluate the efficiency of our model.

TRAINING - DATASET MANAGEMENT

- To simulate the real conditions of recommender systems, we consider that only 1% of the data is relevant for the user:

- For each user, we randomly select an item with which he has interacted (representing the positive instance), and add 99 items with which he has not interacted yet (representing the negative instances).

SNHF TRAINING

- Neural networks are trained using an optimization process that requires a cost function to be minimized.
- This function calculates the difference between the prediction made by the model and the actual value it was supposed to predict.
- For the training of our models (GMF, SHybMLP, and SNHF): we used the binary cross-entropy or log loss functions because it is a *Logistic Regression problem with a binary classification*

Training - Optimization

- To update the values of the model weights, we used the Adam (Adaptive Moment Estimation) algorithm for the training of GMF and HybMLP and the SGD (Stochastic Gradient Descent) algorithm for the training of NHybSoc.

  - Adam algorithm adapts the learning rate to each parameter

  - SGD uses a single learning rate to update the parameters.

EXPERIMENTS

- **Objective :** show the contribution of DL in the prediction of the interaction + the contribution of our approach, in particular, in the cold start situation.

- **Datasets:**

Dataset	# Users	#Items	#Ratings	Density (%)
Yelp	5,436	4,733	110, 894	0.43
FilmTrust	1,508	2,071	35,497	1.14

- **Metrics** Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG)

COLLABORATIVE AND SOCIAL FILTERING

- Take into account user and item identifiers and the social information related to each user:

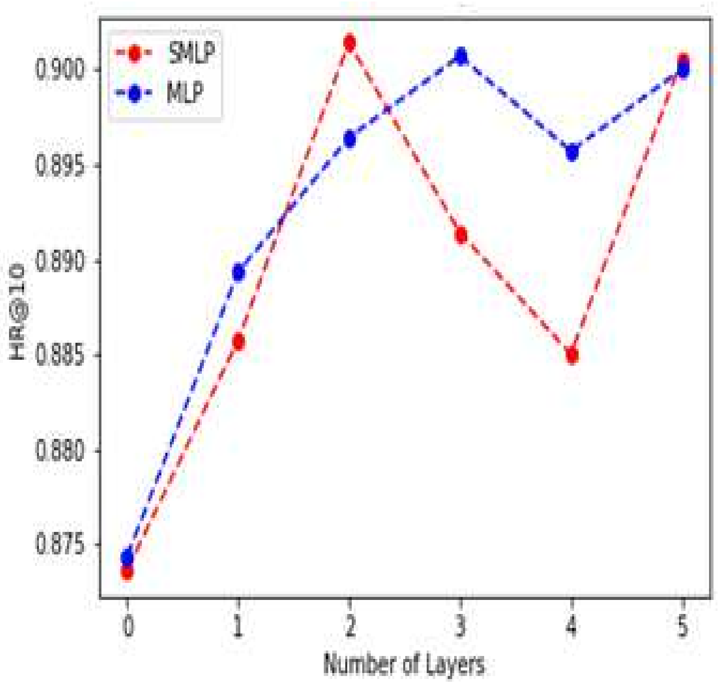
  - This version of our model is called SNCF, and its sub-models are SMLP and GMF.

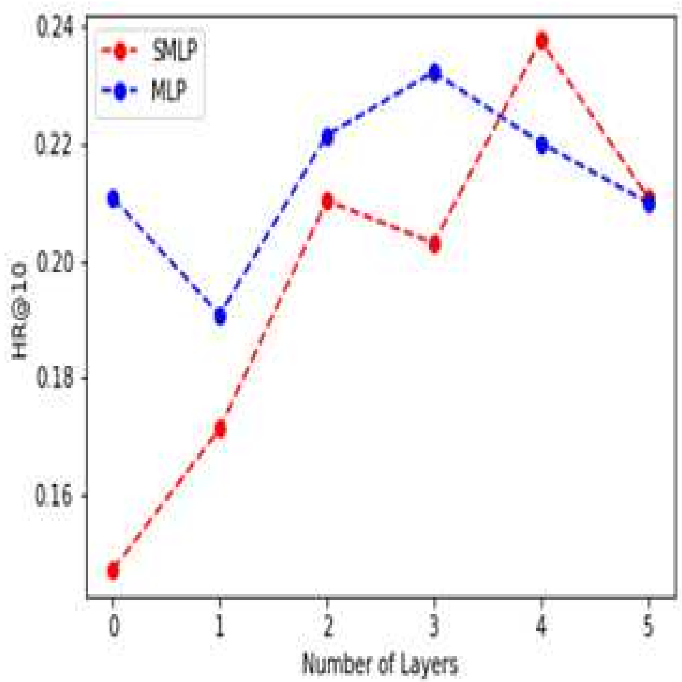
- We evaluated the SMLP model with Yelp and FilmTrust with different embedding sizes and a different number of hidden layers and a Predictive factor = 16.

COLLABORATIVE AND SOCIAL FILTERING

- Improvement and increase in accuracy with the increasing number of layers and the size of the embedding especially when considering 3 or 5 layers

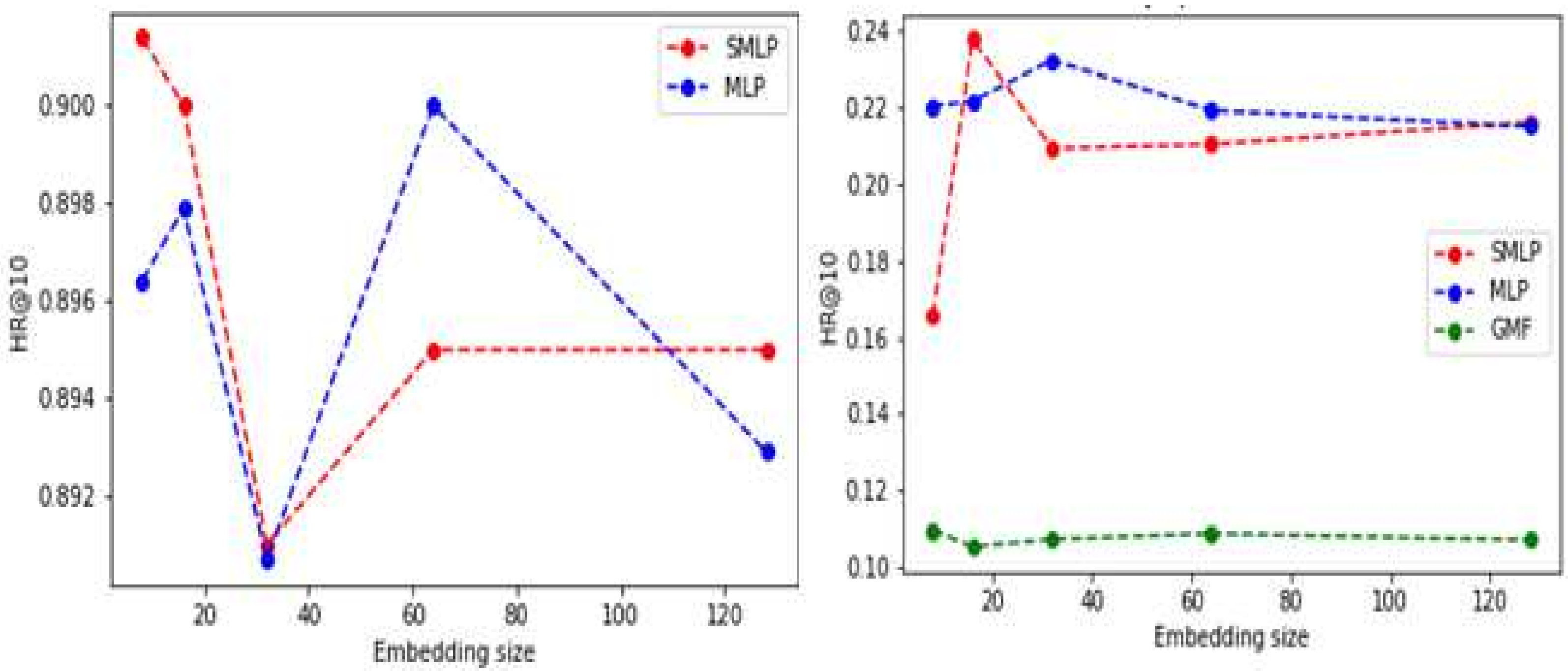
- Evolution of SMLP, by varying the number of layers:





Collaborative and social filtering

- Evolution of SMLP according to the number of layers



SNCF

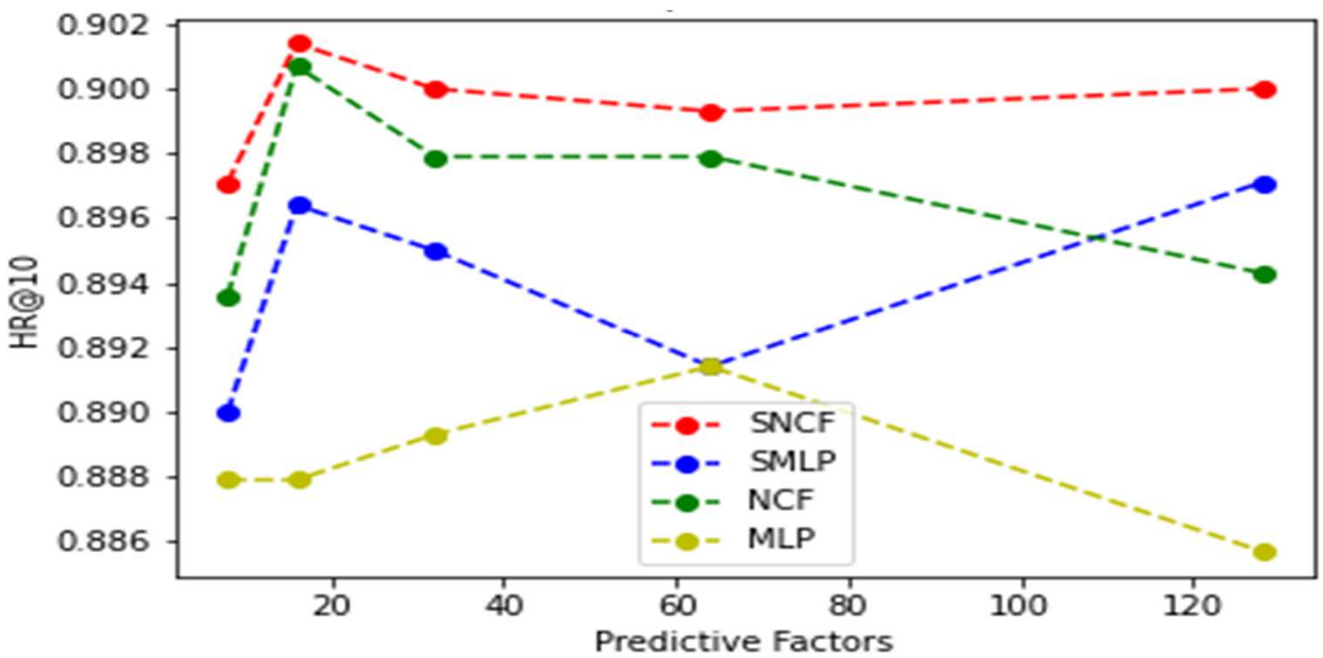
- Two ways to train the SNCF model :
  - GMF and SMLP simultaneously; or
  - SMLP and GMF separately, then train the last concatenation layer after having the weights of both models).

- The evaluation without pre-training showed that the model fails to make good predictions.
  - This is due to the combination of two models that learn to predict the same thing at the same time.

- The results with the pre-training of the sub-models are much better, demonstrating its effectiveness for the SNCF training.

SNCF

- Performance according to the number of predictive factors – FilmTrust:

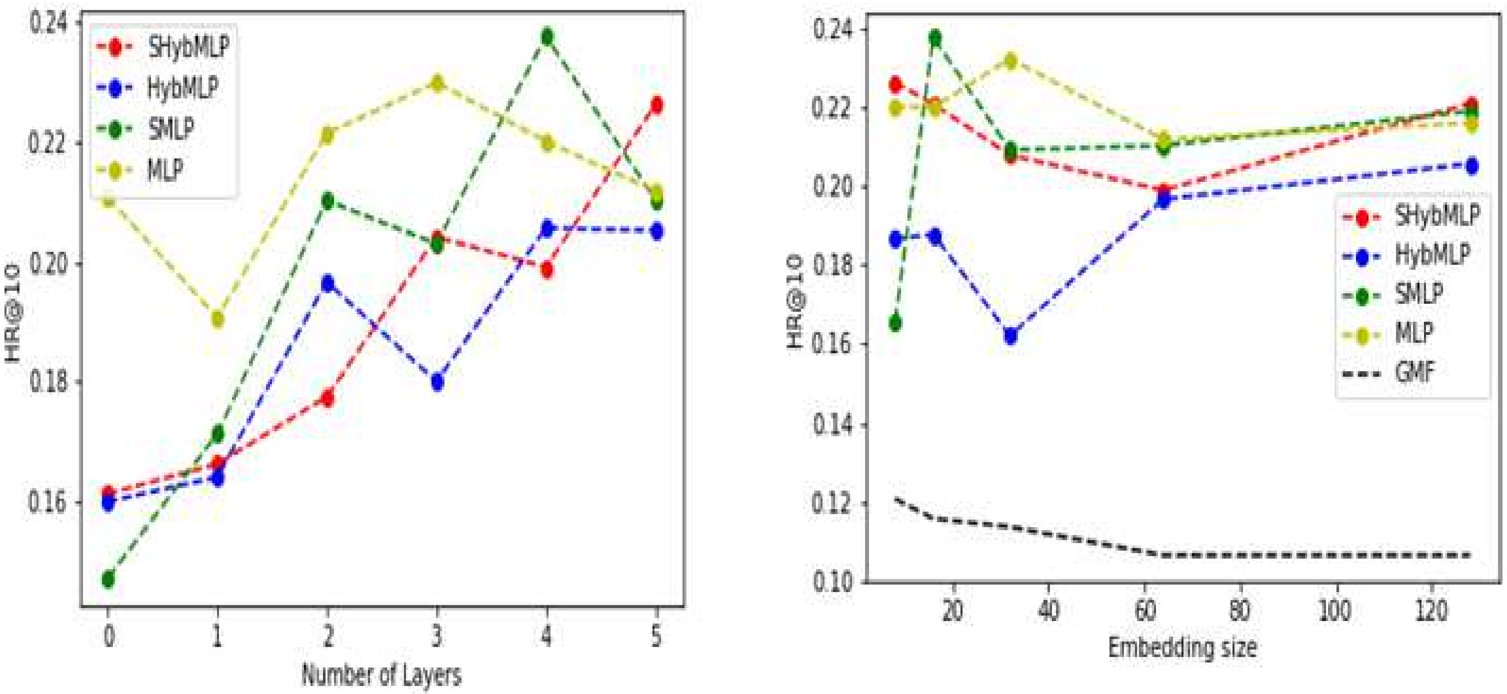


Collaborative, content-based and social filtering

- We evaluated the impact of social information on the hybridization of CF and CBF algorithms, using the Yelp dataset.

SHYBMLP

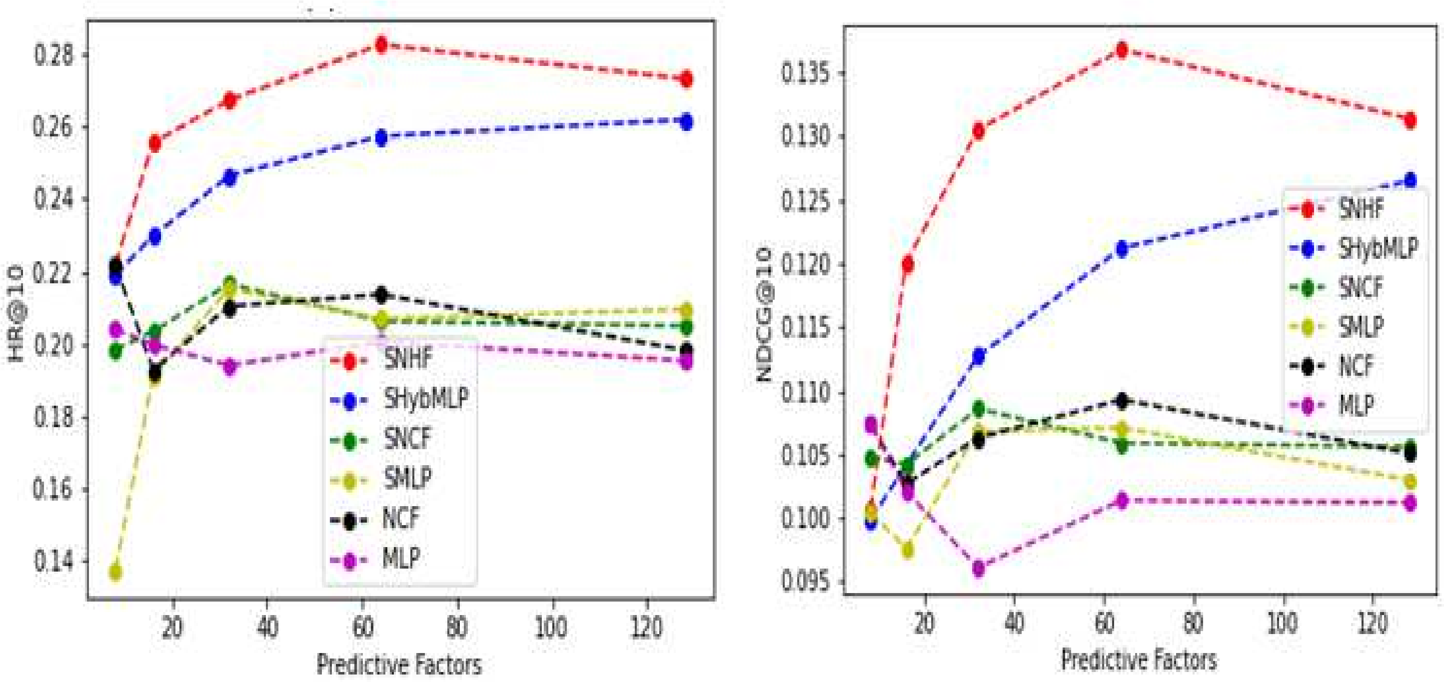
- Evolution of SHybMLP according to the number of layers and embedding size:



- SHybMLP performs better with 5 hidden layers and 8 latent factors, but SMLP achieved better results

SHYBMLP

- SHybMLP performance according to the number of predictive factors – Yelp



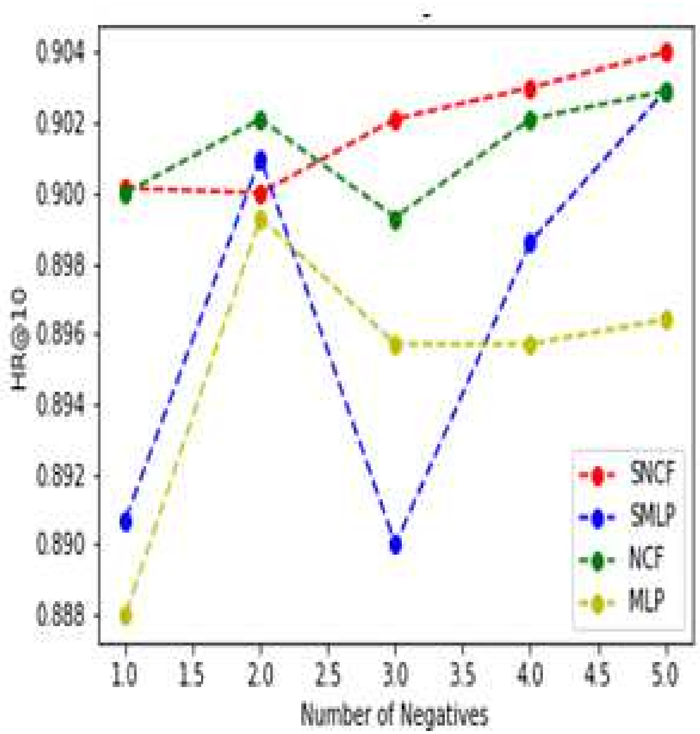
- Improvement of SNHF and SHybMLP with the increase of the number of epochs to 15 epochs for both metrics

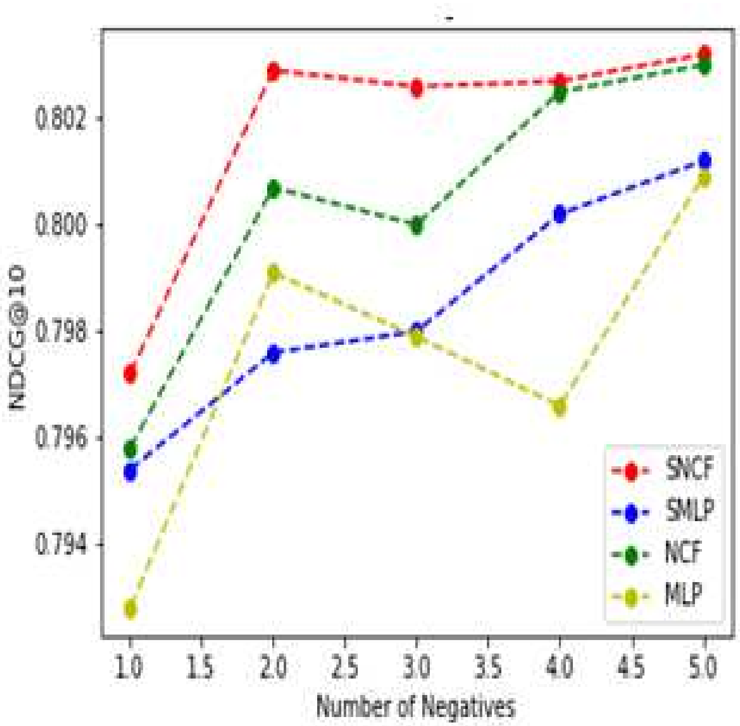
DATASET SCRAMBLING

- To demonstrate the effectiveness of the models, we proceeded to scramble the learning data
  - Increase the number of negative instances compared to positive ones.
  - Objective : find the right number of negative instances to allow the model to be more accurate.
- Tests were performed on the best SNCF, SMLP, SHybMLP and SNHF models obtained from the training on the balanced dataset:

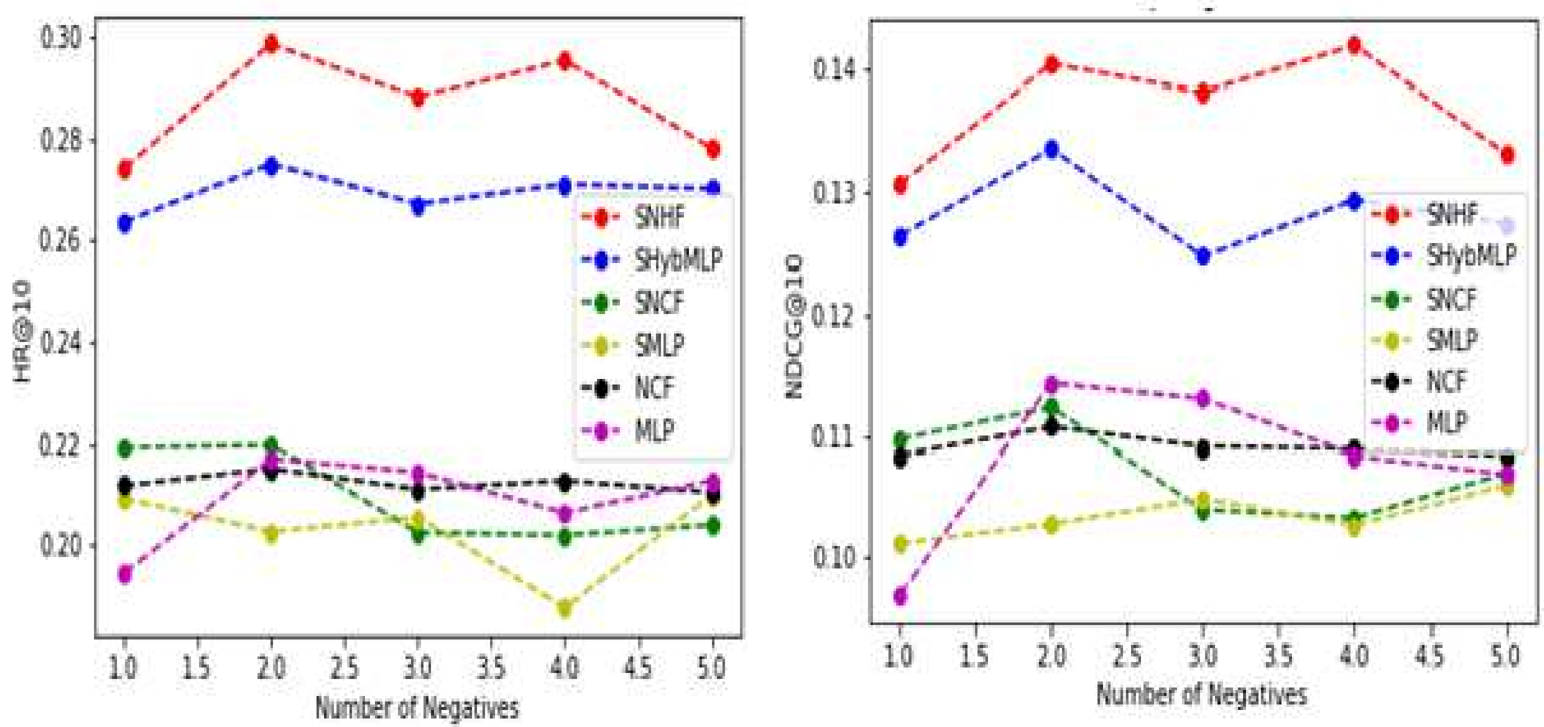
PERFORMANCE – FILM TRUST

- Performance according to the number of negative instances per positive instance:





PERFORMANCE – YELP



- SNHF model performs better when we have a dataset with two negative instances per positive instance for Yelp and five negative instances for FilmTrust.

Conclusion

- This work explored neural network architectures for item recommendation in social networks.
- A hybrid algorithm that combines CF and CBF using social information in an architecture based on GMF and HybMLP models, is proposed.
- Extensive experiments on two real-world datasets show that SNHF significantly outperforms state-of-the-art baselines and related work

PERSPECTIVES

- Enrich the social information with other features (such as the influence and credibility of users);
- Consider the use of other more complex data such as user comments on items, linked open data (LOD) or multi-criteria ratings to improve the quality of the recommendation;
- Explore other DL architectures such as convolutional neural networks to learn high-order correlations among embedding dimensions.

REFERENCES

[1] Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User Adapted Interaction, 12(4), 331–370 (2002).

[2] Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the WWW, pp. 1583–1592, (2018).

[3] Lu, Y., Dong, R. Smyth, B. Coevolutionary recommendation model: Mutual learning between ratings and reviews, in: Proceedings of the WWW, pp. 773–782, (2018).

[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T-S.: Neural Collaborative Filtering. (2017). International World Wide Web Conference Committee (IW3C2), WWW 2017, April 3-7, Perth, Australia (2017), http://dx.doi.org/10.1145/3038912.3052569.

[5] He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T. Outer Product-based Neural Collaborative Filtering. (2018).

[6] Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the WSDM, pp. 425–434, (2017).

[7] Lu, Y., Dong, R. Smyth, B. Coevolutionary recommendation model: Mutual learning between ratings and reviews, in: Proceedings of the WWW, pp. 773–782, (2018).

[8] Liu, H., Wang, Y., Peng, Q., Wu, F., Gan, L., Pan, L., Jiao, P.: Hybrid neural recommendation with joint deep representation learning of ratings and reviews. Neuro-Computing 374, 77–85, (2020).

[9] Berkani, L., Kerboua, I., Zeghoud, S. Recommandation Hybride basée sur l'Apprentissage Profond. Actes de la conférence EDA 2020, Revue des Nouvelles Technologies de l'Information, RNTI B.16 - ISBN: 979-10-96289-13-4, pp. 69-76, (2020).

[10] Avesani, P Massa, P Tiella, R.: Moleskiing it: a trust-aware recommender system for ski mountaineering. International Journal for Infonomics, (2005)

Thank you for your attention!

Emails:

lberkani@usthb.dz
l_berkani@hotmail.com

SOCIAL NEURAL HYBRID RECOMMENDATION WITH DEEP REPRESENTATION LEARNING

Presented by: Lamia Berkani
Dyhia Laga
Abdelhak Aissat

Laboratory for Research in Artificial Intelligence (LRIA)
Dep. of Computer Science, USTHB University

MEDI 2021 – Tallinn, Estonia, June 21-23, 2021