Recommender Systems with Deep Learning

Paula Gómez Duran
paulagomezduran@gmail.com

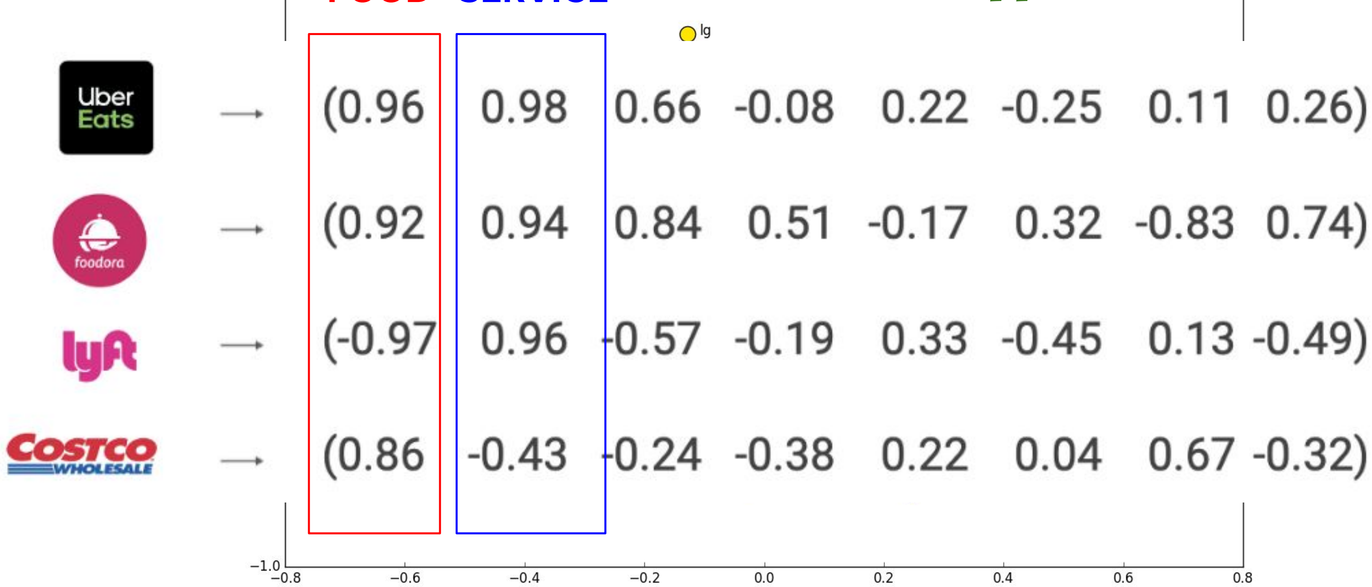
PhD student

UNIVERSITAT DE BARCELONA

WHAT is an EMBEDDING?

- **Embeddings** are a way to represent entities using learned vectors.

Use case: Application's recommender



How do we use embeddings for RS?

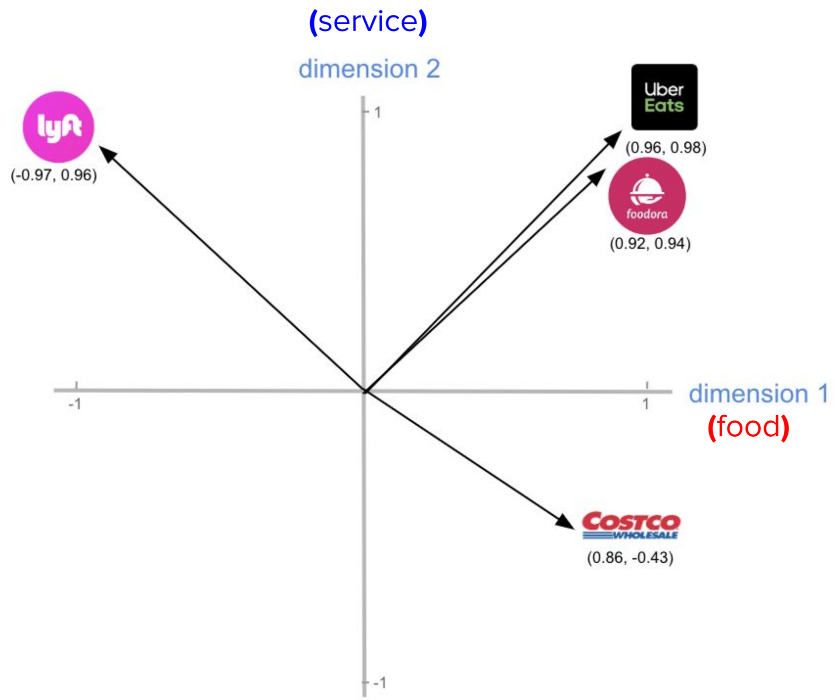


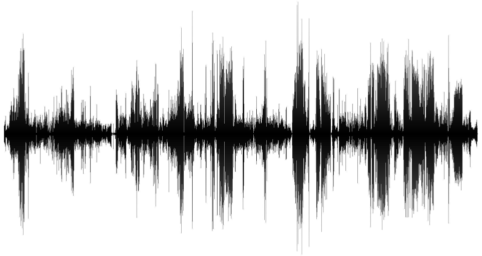
Figure from this blog.

EMBEDDINGS can be **learned** and **reused** across models:

-1	0.4	0.2	...	-0.45	0.87	0.23	-0.67

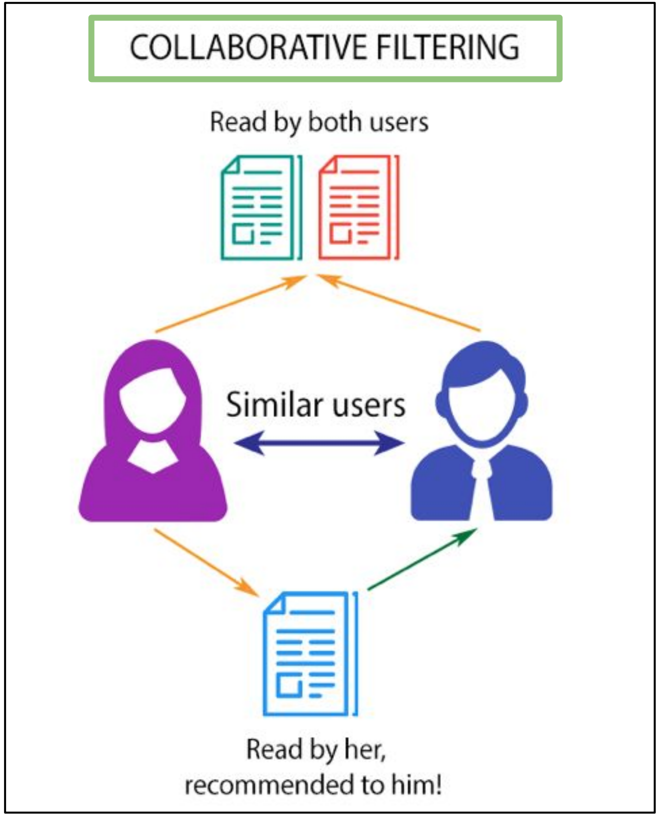
Corporate Social Responsibility (CSR) can generate a positive reputation for a company leading to possibly more sales and growth. According to Jones et al (2019), a corporation that invests in the environmental and ethical approaches of CSR will demonstrate to the public and the media that they are a responsible company. Watson (2018) provides evidence that this improves consumer sales as customers tend to support ethical green business practice thus improving profitability and encouraging growth. For example, a yoghurt company called Yeo Valley has been investing in making its products organic, creating fully recyclable packaging and reducing its CO2 output. As a result, profits have doubled within the last two years providing the company with a range of opportunities to expand (Peterson, 2019). Overall, the evidence seems to suggest that investing in CSR can improve brand image and productivity.

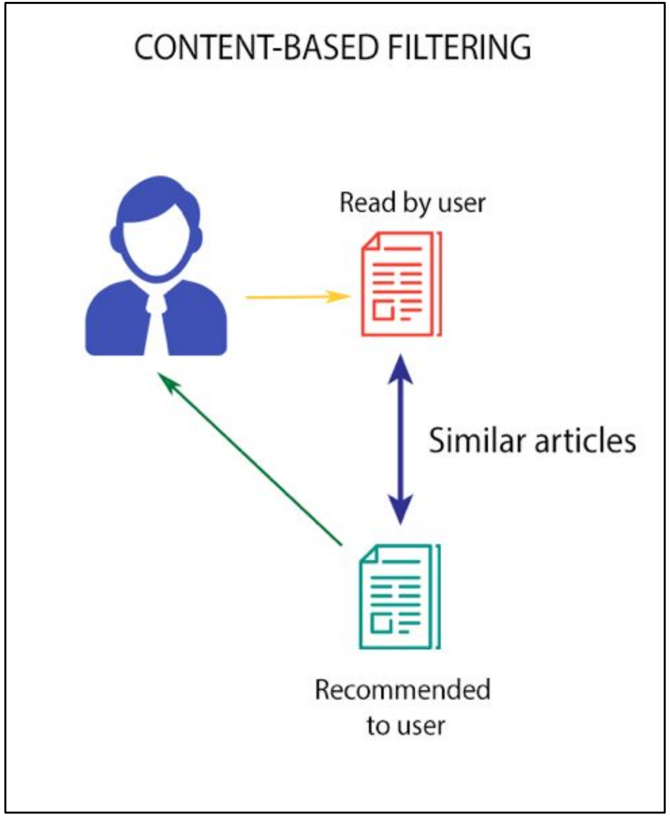
1	-0.2	...	0.5	0.9



0.8	0.2	-0.87	...	-0.52	0.7	-0.32	-0.79

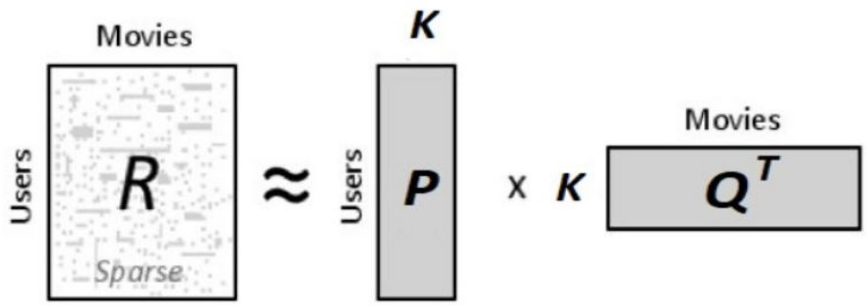
TECHNIQUES FOR RECOMMENDATION:





SOTA models

1. Matrix Factorization



$$\boxed{r_{ij} = p_i^T q_j}$$

$\boldsymbol{p}$ → *latent features for users (user embeddings)*

$\boldsymbol{q}$ → *latent features for items (item embeddings)*

2. Factorization Machines (FM)

(Paper here)

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \boxed{\langle \mathbf{v}_i, \mathbf{v}_j \rangle} \boxed{x_i x_j}$$

LINEAR REGRESSION

EMBEDDING LAYER

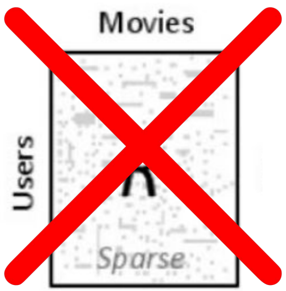
USER-ITEM FEATURES

All nodes

All nodes

	User_0 .... User_Nu	Item_0 ... Item_Ni
User_0 … User_Nu	0	Interactions user-item
Item_0 … Item_Ni	Interactions user-item	0

Adjacency matrix



JUST FOR AN EASIER IMPLEMENTATION ...

FM part

$$\boxed{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j}$$

=

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i\rangle\, x_i\, x_i$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i\right)$$

$$= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$

$$\boxed{= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)}$$

Why is FM a generalization of MF ?

MF

$$r_{ij} = p_i^T q_j$$

FM

$$r_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

- **Forced to learn features of users and items while factorizing.**

- **Allow side-info in x_i vector**
- **Allow more dimensions**

	User_0 .... User_N_u	Item_0 ... Item_N_i	Context_0 ... Context_N_C
User_0 ... User_N_u	0	Interactions user-item	Interactions user-context
Item_0 ... Item_N_i	Interactions user-item	0	Interactions item-context
Context_0 ... Context_N_C	Interactions user-context	Interactions item-context	0

FM PROBLEM!

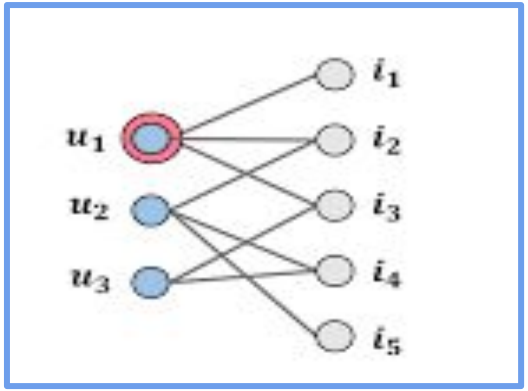
NFM

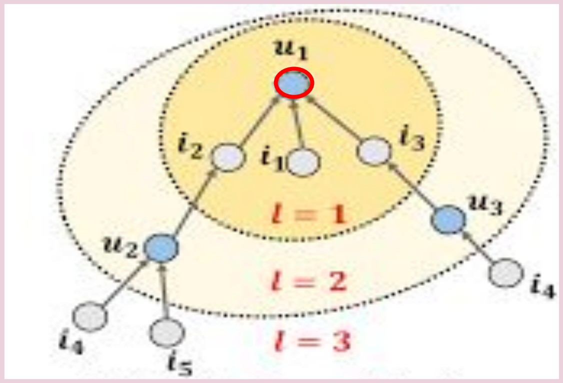
DFM

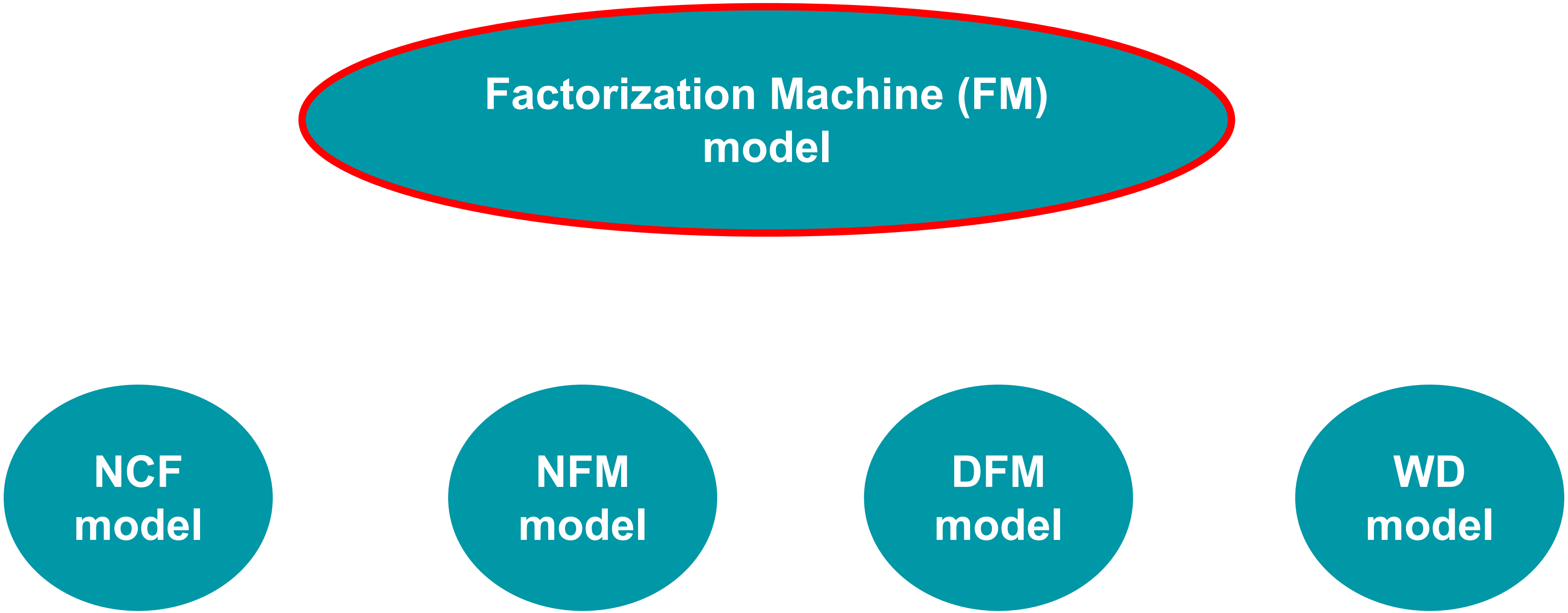
WD

FM captures interactions of only second order! $(\ell = 1)$

We might want to capture high order interactions ....






Factorization Machine (FM) model
NCF model
NFM model
DFM model
WD model

FM model

Rendle, Steffen. "Factorization machines." 2010

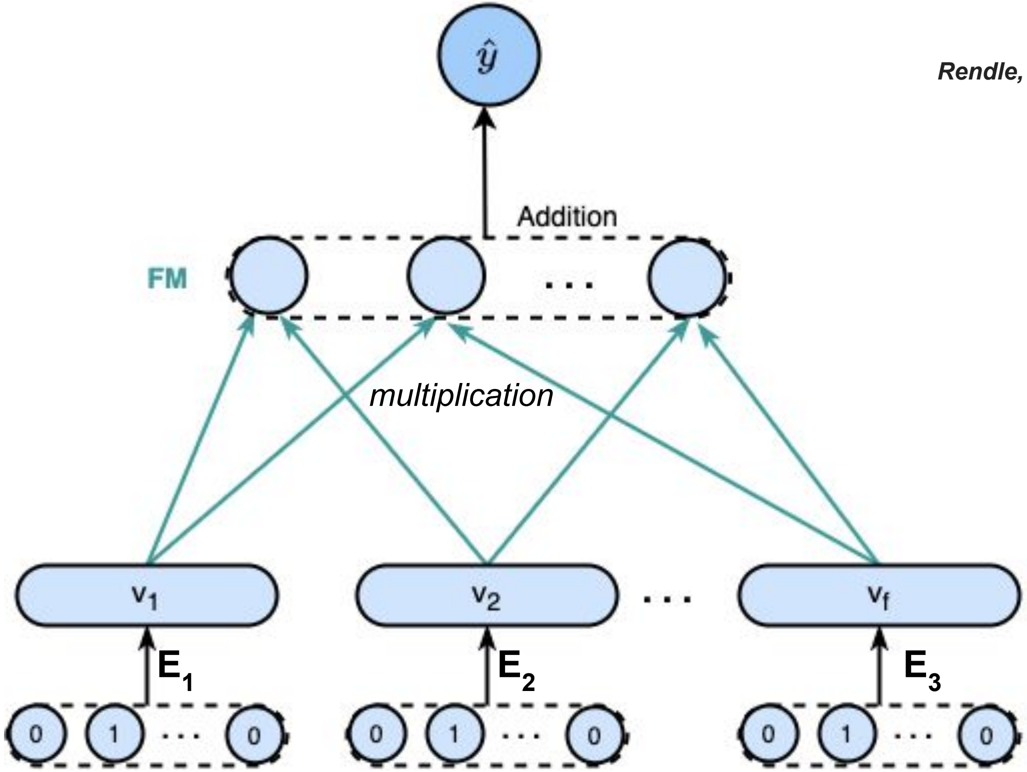
Feature vector **x** | Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

FM model

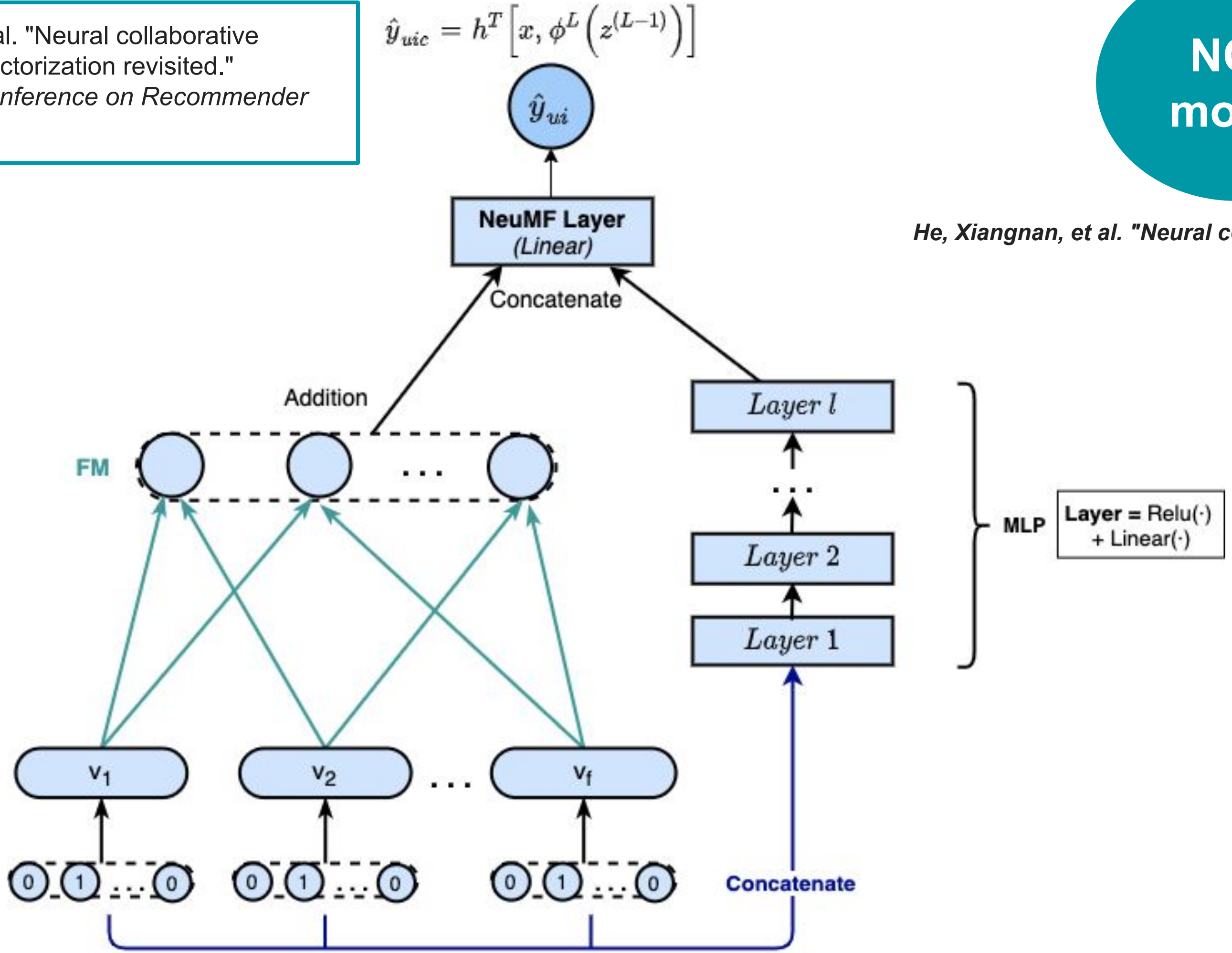
Rendle, Steffen. "Factorization machines." 2010



Rendle, Steffen, et al. "Neural collaborative filtering vs. matrix factorization revisited." *Fourteenth ACM Conference on Recommender Systems*. 2020.

NCF model

He, Xiangnan, et al. "Neural collaborative filtering." , 2017



EVALUATION ON RANKING

Explicit vs Implicit feedback

✗ **How much did the user like the film?** (explicit)

✓ **Was the user interested in the film?** (implicit)

- We won’t care about ratings (1-5).
- All provided data will be treated as positive
  → We will need to perform negative sampling.

Leave-one-out procedure:
(positive samples)

dataset

users: 1 - 5
items: 6 - 21

TEST
u1 - item_6

VAL
u1 - item_15

TRAIN
u1 - item_16
u1 - item_15
u1 - item_9
u1 - item_8
u1 - item_10
U1 - item 20

subtracting items seen in training for this user !

u1 - **item_6** **GT**

u1 - **item_7**
~~u1 - **item_8**~~
~~u1 - **item_9**~~
~~u1 - **item_10**~~
u1 - **item_11**

...

u1 - **item_18**
u1 - **item_19**
~~u1 - **item_20**~~
u1 - **item_21**

Leave-one-out procedure:
(positive samples)

dataset
users: 1 - 5
items: 6 - 21

TEST dataset

u1 - **item_6** (GT)

u1 - **Item_7**
u1 - **Item_11**
u1 - **Item_12**
u1 - **Item_13**

...

u1 - **Item_17**
u1 - **Item_18**
u1 - **Item_19**
u1 - **Item_21**

COMPUTE PREDICTIONS AND SORT THEM

Compute HR and NDCG to see whether out GT prediction is in the TOP@K

METRICS:

- **Hit Ratio (HR@K)**

  Measures whether the test item is in the top@K positions of the recommendation list (1 = yes | 0 = no).

- **Normalized Discounted Cumulative Gain (NDCG@K)**

  Measures the ranking quality which gives information about where in the raking is our test item.