Modurec: Recommender Systems with Feature and Time Modulation

Javier Maroto, Clément Vignac and Pascal Frossard

EPFL

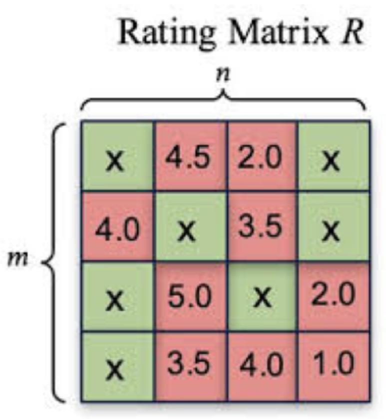
Motivation

- Combine different sources of information to improve over state-of-the-art collaborative filtering approaches in recommendation
- Address cold start and concept drift, which affect collaborative filtering

Recommender systems

Goal: **Matrix reconstruction**

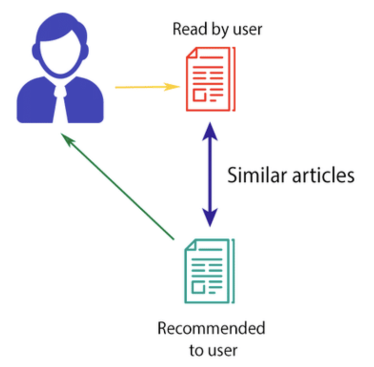
- Rows, columns $\longrightarrow$ Users, items
- Very few known values (less than 1% of entries)
- Many items with few users, few items with many users



Two main solving strategies

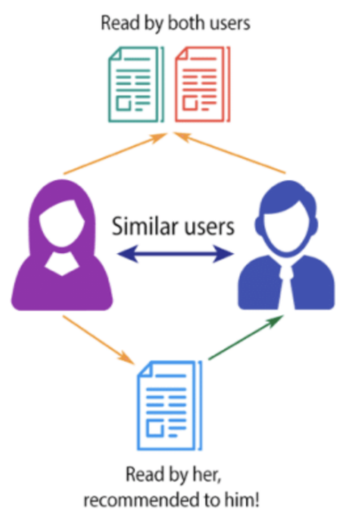
Content filtering

- Recommend similar items to the ones the user liked
- Requires prior info on items (e.g. movie genre, lead actor...)
- This info provides limited information
- Performance **does not scale with data**



Collaborative filtering

- Recommendation based on user with similar rating history
- Performance scales with data
- Rich information based on user behavior
- Susceptible to **cold start and concept drift**

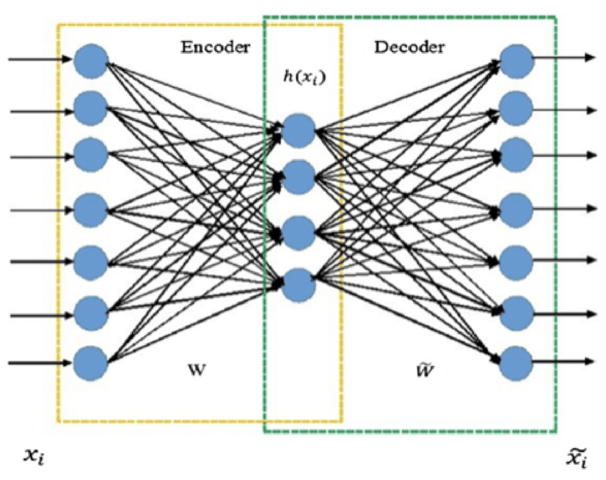


Autorec

- First approach based on autoencoders (latent factor model) [1]
- Input: the rating matrix, with zeros for the unknown entries
- Output: reconstructed rating matrix, with all entries filled with the model predictions

$$\boldsymbol{H} = \sigma(\boldsymbol{R}\mathbf{W}_{enc} + \boldsymbol{b}_{enc})$$
$$\hat{\boldsymbol{R}} = \boldsymbol{H}\mathbf{W}_{dec} + \boldsymbol{b}_{dec}$$

- The loss is calculated only for known entries



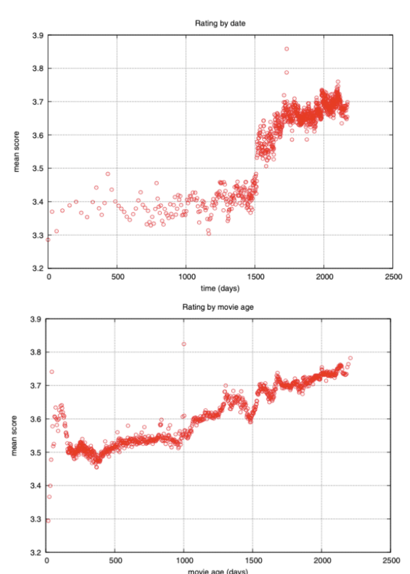
Cold start and concept drift

Cold start

- Bad performance on new users or items due to lack of ratings
- Could stop new users from joining the platform

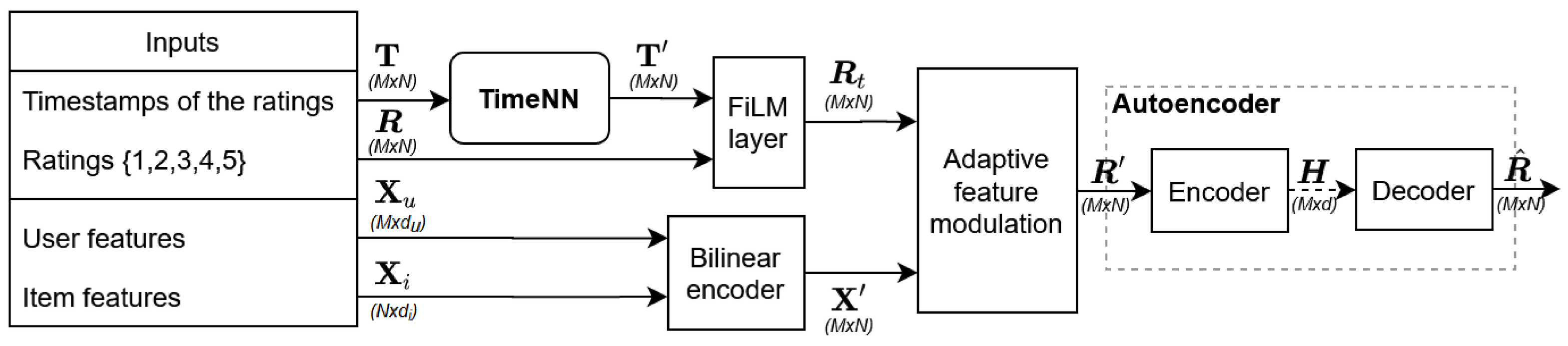
Concept drift

- Bad performance on older users or items due to distribution shifts over time [2]



Modurec

- **Our proposal**: Address cold start and concept drift within a collaborative setting



TimeNN

- Create 3 feature maps by normalizing the timestamps wrt user/item/platform first rating
- Use fully-connected layers (32 and 1 hidden units) that are applied rating-wise

Feature-wise Linear Modulation (FiLM) [3]

- Used to combine time and rating information
- Only 3 free parameters
- Much more expressive than concatenation

$$\boldsymbol{R}_t = \alpha\boldsymbol{R} + \beta\boldsymbol{T}' + \gamma\boldsymbol{R}\cdot\boldsymbol{T}'$$

Bilinear encoder

- Combines the user and item features into a feature matrix of the same shape as $\boldsymbol{R}_t$

$$\boldsymbol{X}' = \boldsymbol{X}_i\boldsymbol{\Theta}\boldsymbol{X}_u^T$$

Adaptive feature modulation

- Adds the user/item feature information
- Uses a importance matrix to leverage when the feature information is most valuable (i.e. cold start)

$$\mathbf{A}_{ij} = \begin{cases} \sigma(w_1|\mathcal{O}_{i,i}| + w_2|\mathcal{O}_{u,j}| + b) & |\mathcal{O}_{i,i}|, |\mathcal{O}_{u,j}| > 0 \\ 0 & |\mathcal{O}_{i,i}|, |\mathcal{O}_{u,j}| = 0 \end{cases}$$

$$\boldsymbol{R}' = \mathbf{A}\cdot\boldsymbol{R}_t + (1-\mathbf{A})\cdot\boldsymbol{X}'$$

Autoencoder input dropout

- Denoise the sparse signal received by the user/item.
- It allows both explicit minimization of the prediction error on unobserved ratings, and the reconstruction error on observed ratings

Results

Ablation and comparison with state of the art

- Average RMSE recommendation results on several MovieLens datasets
- We use Modurec_[DFT] as the nomenclature for our model
  - D = with autoencoder dropout; F = with user and item features module; T = with time module

Dataset	GRALS	sRGCNN	GC-MC	STAR -GCN	CF-NADE	Sparse FC	TimeSVD++ flipped*	I-Autorec*	Modurec _D	Modurec _DT	Modurec _DFT
ML-100K	0.945	0.929	0.905	0.895	—	—	0.890	0.908	0.905	**0.887**	**0.887**
ML-1M	—	—	0.832	0.832	0.829	0.824	0.842	0.831	0.826	**0.821**	**0.821**
ML-10M	—	—	0.777	0.770	0.771	0.769	**0.749**	0.782	0.789	0.777	0.779

Cold start evaluation

- Evaluate on specific scenarios:
  - *Few ratings*: both $|\mathcal{O}_{i,i}|$ and $|\mathcal{O}_{u,j}|$ are in the bottom quantile
  - *Many ratings*: both $|\mathcal{O}_{i,i}|$ and $|\mathcal{O}_{u,j}|$ are in the top quantile
- Different architectures:
  - *Nothing*: Remove the bilinear encoder and the adaptive combiner (no user or item features are used).
  - *Static*: Use a much simpler combiner instead of our adaptive combiner. It is characterized by the following relation: $\boldsymbol{R}' = \alpha\boldsymbol{R}_t + (1-\alpha)\boldsymbol{X}'$, where α is a scalar trainable parameter.
  - *Adaptive*: Use the adaptive feature modulation.

Dataset	Algorithm	Few ratings	Many ratings
ML-100K	Nothing	1.6093	**0.8371**
	Static	1.6000	0.8417
	Adaptive	**1.3412**	0.8380
ML-1M	Nothing	1.1481	**0.7895**
	Static	1.1457	0.7900
	Adaptive	**1.1360**	0.7897

References

[1] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. AutoRec: Autoencoders Meet Collaborative Filtering. In *WWW*, pages 111–112. ACM Press, 2015.
[2] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
[3] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI Conference on Artificial Intelligence*, April 2018.