MetaRec: Meta Learning Meets Recommendation Systems

Presentation made by
James Le
Advisor: Dr. Alexander Ororbia II

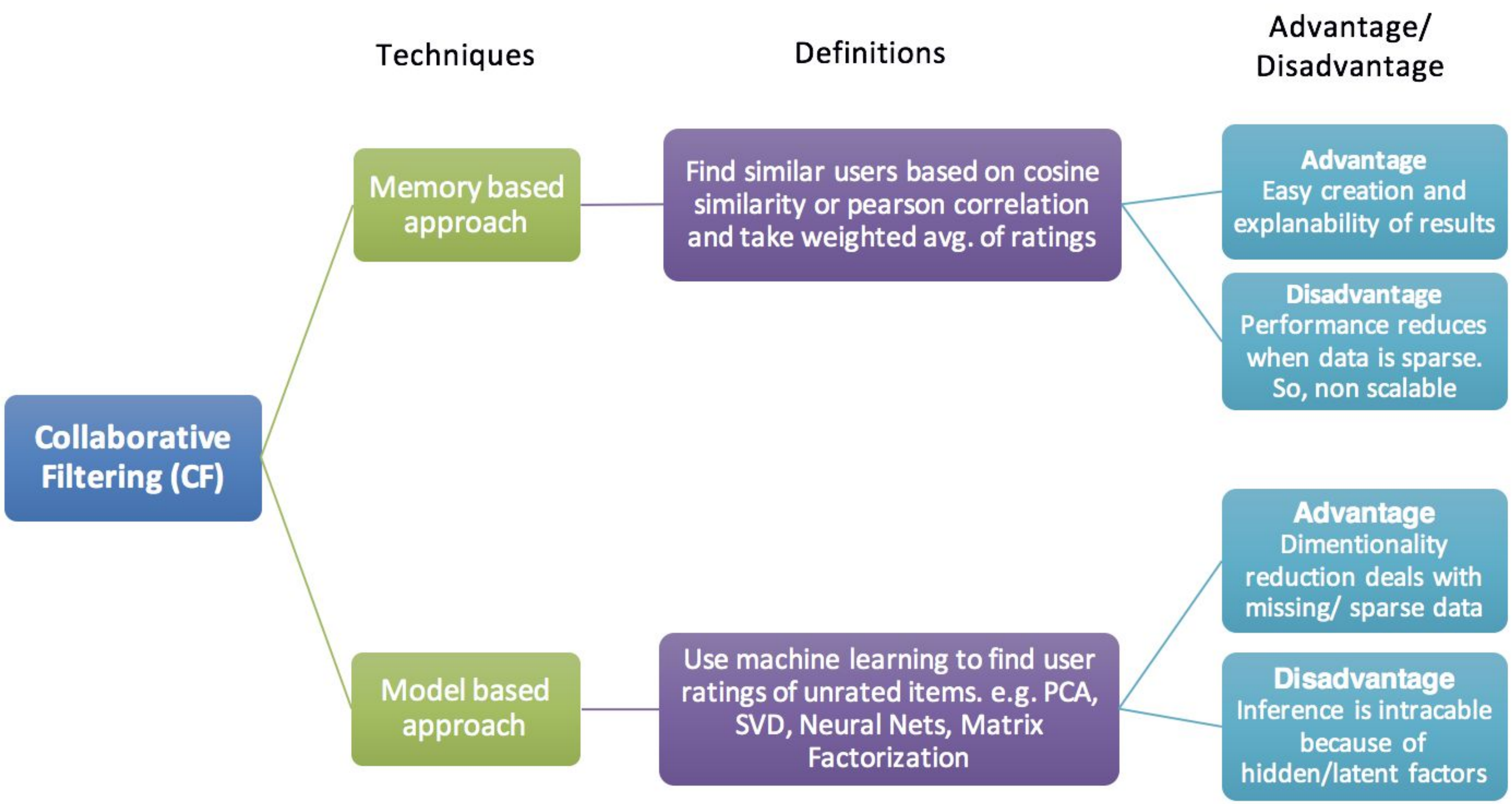
R·I·T
ROCHESTER INSTITUTE OF TECHNOLOGY

1 - Recommendation Systems

80% of all content consumed

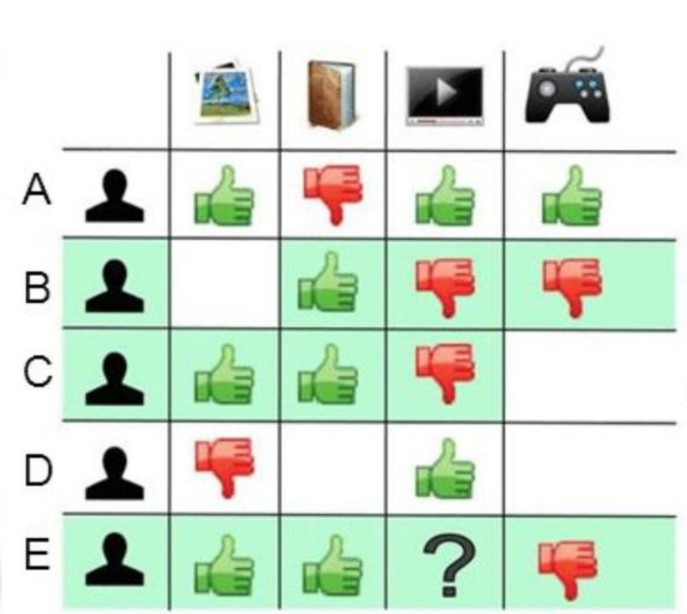
$98 billion of annual revenue

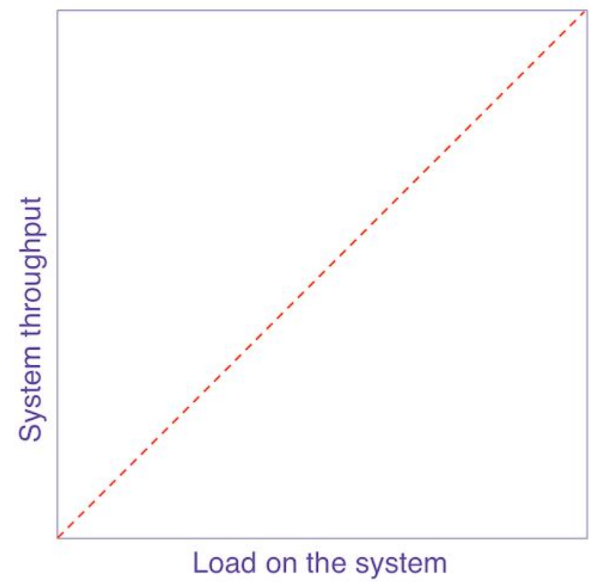
Collaborative Filtering



Source: [Various Implementations of Collaborative Filtering]

Challenges





Accuracy

Sparsity

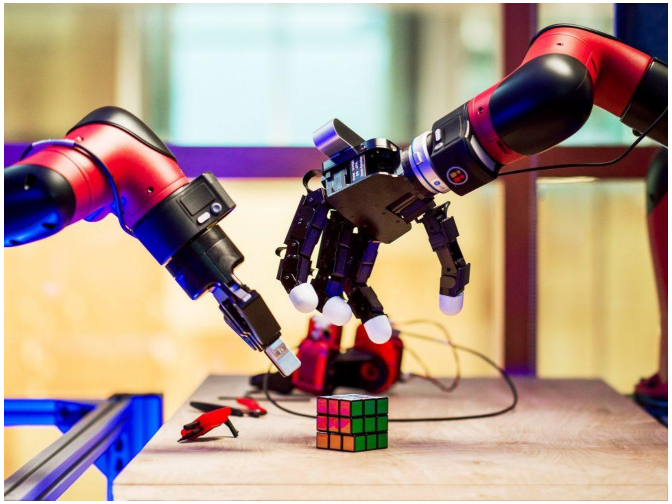
Scalability

Motivation

How can we design an effective and efficient learning paradigm that enables collaborative filtering systems to get better performance (accuracy), work well with limited data (sparsity), and take reasonable training time (scalability)?

2 - Meta Learning

Motivation

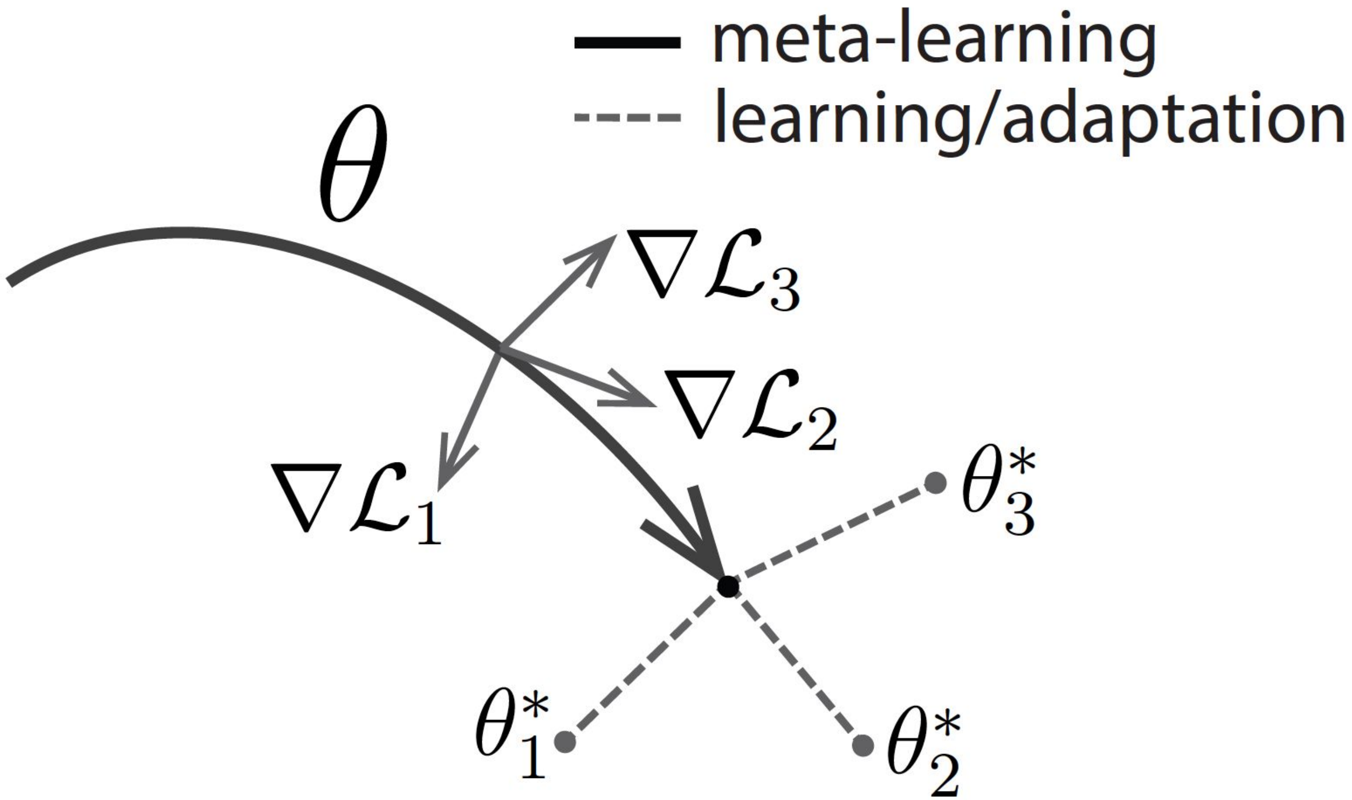


Limited Data

Long-Tail

Fast Inference

Model-Agnostic Meta-Learning



Source: [Learning to Learn]

Model-Agnostic Meta-Learning

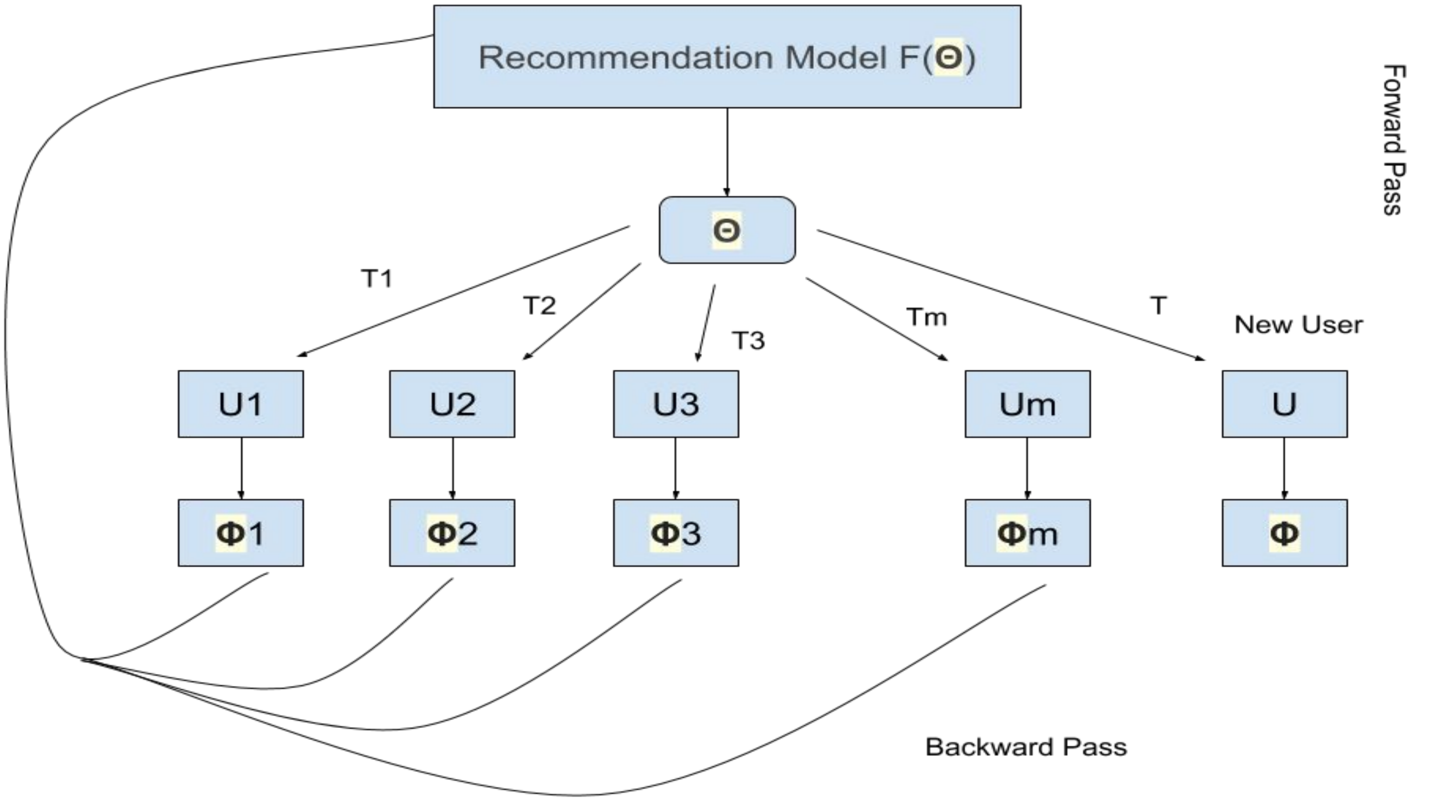
1. Sample a task T_i (or mini batch of tasks)
2. Sample disjoint sets D_i^{tr} and D_i^{t} from D_i
3. Optimize $\theta^*_i \leftarrow \theta - \alpha \nabla\_\theta L(\theta, D_i^{tr})$
4. Update θ using $\nabla\_\{\theta\} L(\theta^*_i, D_i^{t})$

where

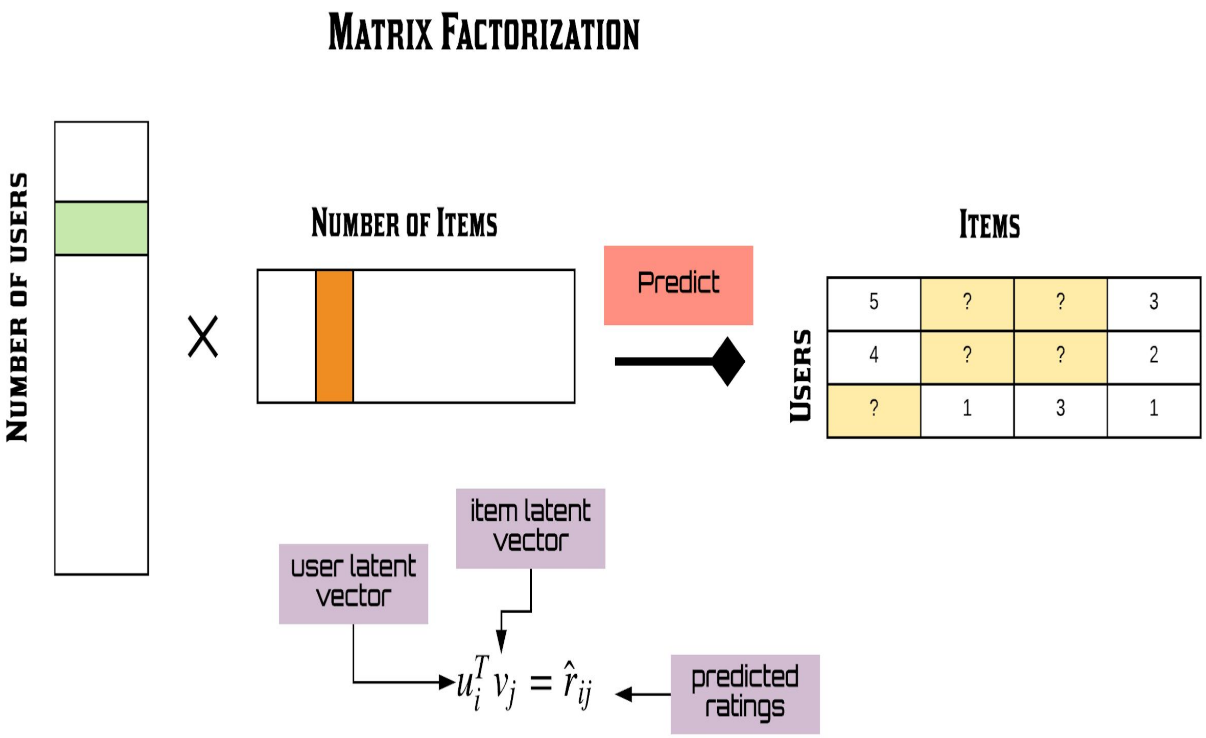
- Model Parameters: $\boldsymbol{\theta}$
- Task-Specific Parameters: $\boldsymbol{\theta^*_i}$
- Task-Specific Dataset: $\mathbf{D_i}$
- Training Task-Specific Set: $\mathbf{D_i^{tr}}$
- Test Task-Specific Set: $\mathbf{D_i^{t}}$

3 - MetaRec

Architectural Diagram



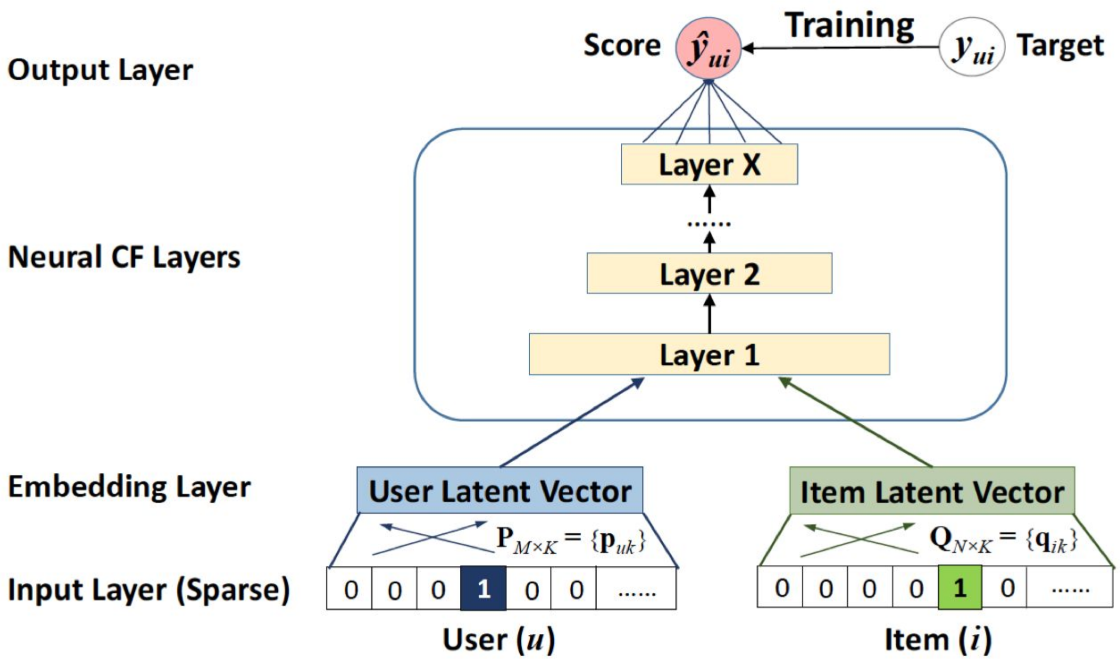
1st Base Model F(θ): Matrix Factorization



- The de facto standard model for model-based collaborative filtering
- Represent user ratings as a user-item matrix
- Find two small latent matrices that approximate the full original matrix
- Minimize the rating prediction errors
- Can be optimized via Gradient Descent
- Prediction is simple

Source: [The 7 Variants of Matrix Factorization Models for Collaborative Filtering]

2nd Base Model F(θ): Multi-Layer Perceptron

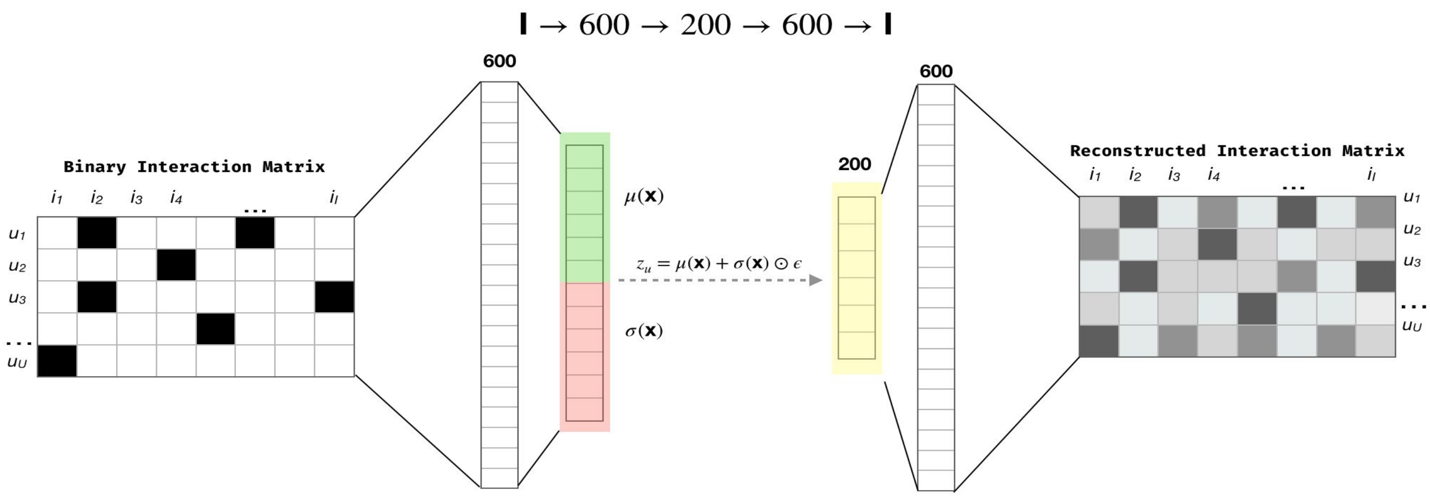


- Input Layer
- Embedding Layer
- Neural CF Layer
- Output Layer
- P - Latent Factor Matrix for Users
- Q - Latent Factor Matrix for Items
- v - Side Information for Users and Items
- θ - Model Parameters

$$r_{ui} = F(P^T \cdot v_u^U, Q^T \cdot v_i^I | P, Q, \theta_F)$$

Source: He, Liao, Zhang, Nie, Hu, and Chua, 2017

3rd Base Model F(θ): Variational Autoencoder

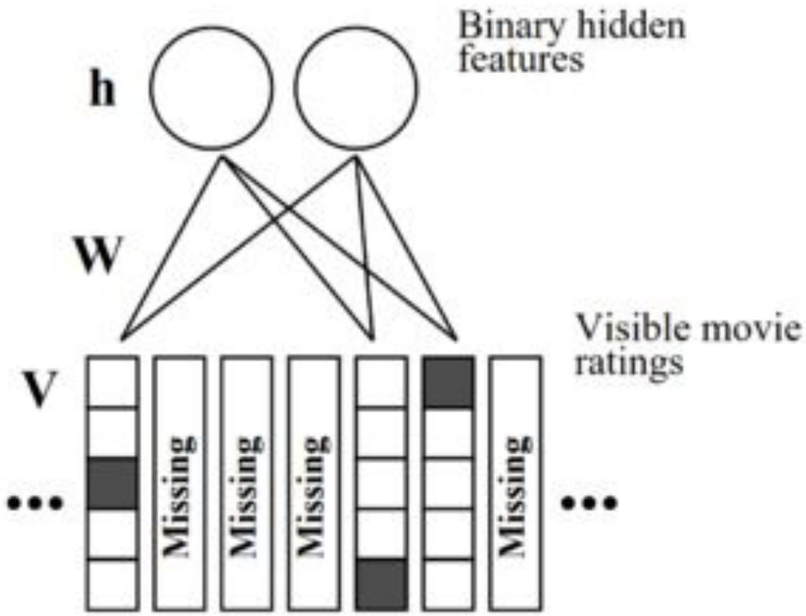


- I - Input Preference Matrix
- K - Number of Latent Dimensions
- z_u - K-dimensional latent representation for user u
- r_u - Preferences for each item from user u
- θ - Model Parameters

$$
\begin{aligned}
z_u &\sim \mathbb{N}(0, I_K), \\
\pi(z_u) &\sim softmax\{F_\theta(z_u)\}, \\
r_u &\sim Mult(\mathbb{N}_u, \pi(z_u)).
\end{aligned}
$$

Source: Liang, Krishnan, Hoffman, and Jebara, 2018

4th Base Model F(θ): Restricted Boltzmann Machine



- R - Latent Factor Matrix for Users
- h - binary ratings
- W - adjacency between ratings and hidden features
- v - visible ratings

$$p(v_i^k = 1|h) = \frac{exp(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k)}{\sum_{l=1}^{K} exp(b_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)}$$

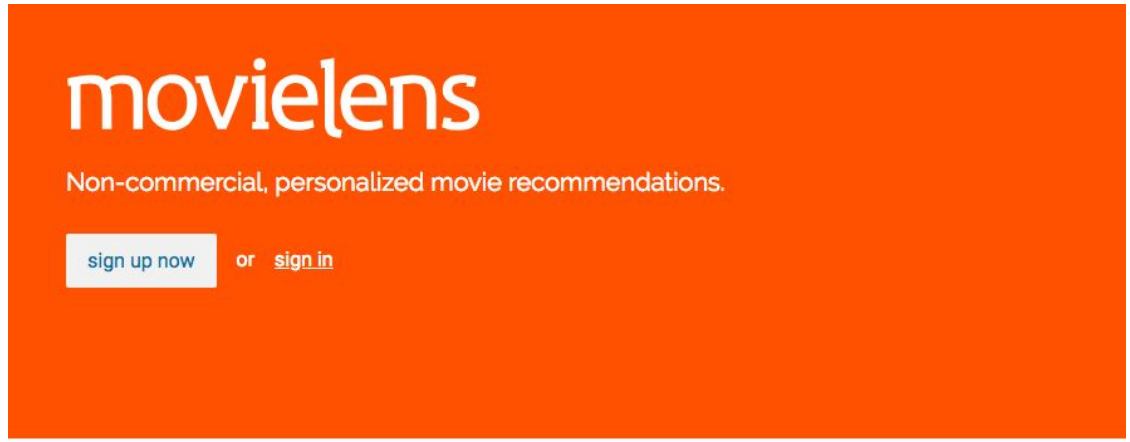
$$p(h_j = 1|R) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} r_i^k W_{ij}^k$$

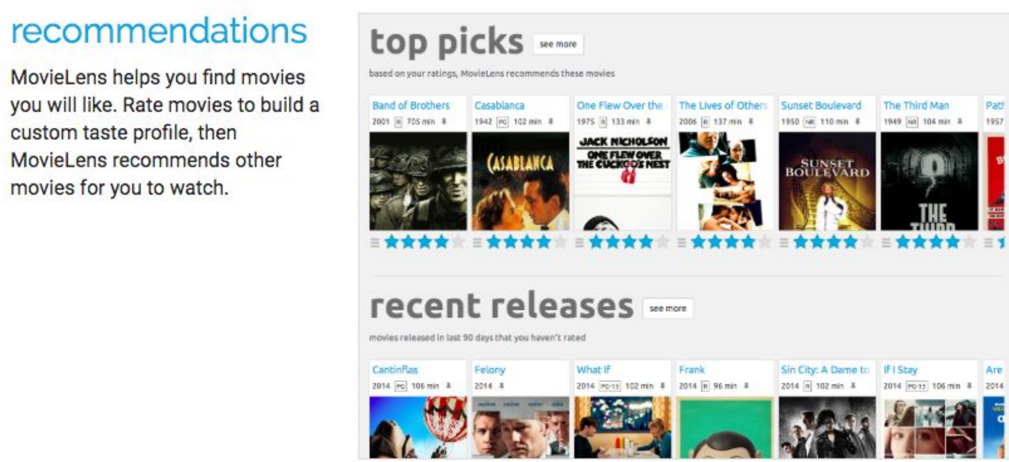
Source: [Salakhutdinov, Mnih, and Hinton, 2007]

4 - Evaluation

MovieLens1M Data

- 1 Million Ratings
  - UserID
  - MovieID
  - Rating
  - Timestamp
- 4,000 Movies
  - MovieID
  - Title
  - Genres
- 6,000 Users
  - UserID
  - Gender
  - Age
  - Occupation
  - Zipcodes





Comparative Results

Matrix Factorization Models

Model	Training Loss	Test Loss	Training Time
Vanilla MF	0.6947	0.8174	6m5s
MF Biases	0.6789	0.7895	11m38s
MF Side Features	0.6602	0.7843	13m34s
MF Temporal Features	0.7088	0.7939	18m51s
Factorization Machine	0.6542	0.8225	3m40s
MF Mixture of Tastes	0.6366	0.7878	13m44s
Variational MF	0.6206	0.8385	16m51s

Multi-Layer Perceptron Models

Model	Test AUC	Valid AUC	Runtime
wide-and-deep	0.7991	0.7995	1h12m15s
deep-fm	0.7915	0.7918	1h10m50s
xDeep-fm	0.7429	0.7408	2h15m17s
neural-fm	0.7589	0.7560	1h36m0s
neural-cf	0.7668	0.7673	54m15s

Autoencoders Models

Model	Epochs	RMSE	Precision@100	Recall@100	NDCG@100	Runtime
AutoRec	500	0.910				35m16s
DeepRec	500	0.9310				54m24s
CDAE	141		0.0894	0.4137	0.2528	17m29s
MultVAE	55		0.0886	0.4115	0.2508	6m31s
SVAE	50		8.18	58.4987	38.0714	6h37m19s
ESAE	50		0.0757	0.4181	0.2561	10m12s

Boltzmann Machines Models

Model	RMSE	Runtime
RBM	0.590	10m56s
Explainable RBM	0.3116	1m43s
NADE	0.920	90m45s

Thank you!

Questions?