Recommender Systems

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

Machine Learning

Recommender Systems

- Goal: to find most relevant ***items*** (products) for ***users***
  - Or, to find most relevant users for items
- Used in many areas such as Amazon, Facebook/Instagram, Spotify, Netflix/Youtube/TikTok, UberEats, Google Ads, etc.

Content-based vs. Collaborative Filtering

- Basic idea: to find users/items similar to engaged users/items
- Content-based algorithms
  - Content similarity
  - Assume that ***user/item features*** are available
    - E.g., user profiles, product descriptions, etc.
- Collaborative filtering (CF)
  - Interaction similarity
  - Assumes that ***user-item interactions*** are available
    - E.g, user1 clicks/likes/rates item100

Content-based vs. Collaborative Filtering

- Basic idea: to find users/items similar to engaged users/items
- Content-based algorithms
  - Content similarity
  - Assume that ***user/item features*** are available
    - E.g., user profiles, product descriptions, etc.
- Collaborative filtering (CF)
  - Interaction similarity
  - Assumes that ***user-item interactions*** are available
    - E.g, user1 clicks/likes/rates item100
  - Popular since
    - There's no need to analyze users/items
    - Performs well empirically

Outline

1. **Collaborative Filtering**
   - Matrix Factorization
   - AutoRec
2. **Implicit Feedback & Personalization**
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - Self-Attentive Sequential Recommendation (SASRec)
4. **Using Side Features**
   - Factorization Machines (FM)
   - Deep Models
5. **Performance Evaluation & System Design**

Outline

1. **Collaborative Filtering**
   - Matrix Factorization
   - AutoRec
2. **Implicit Feedback & Personalization**
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - Self-Attentive Sequential Recommendation (SASRec)
4. **Using Side Features**
   - Factorization Machines (FM)
   - Deep Models
5. **Performance Evaluation & System Design**

Rating Matrix

$$\boldsymbol{R}^{M\times N} \in \mathbb{R}^{M\times N}$$

	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

- M: number of users
- N: number of items
- $R_{m,n}$: interaction between user m and item n

Rating Matrix

$$\boldsymbol{R}^{M \times N} \in \mathbb{R}^{M \times N}$$

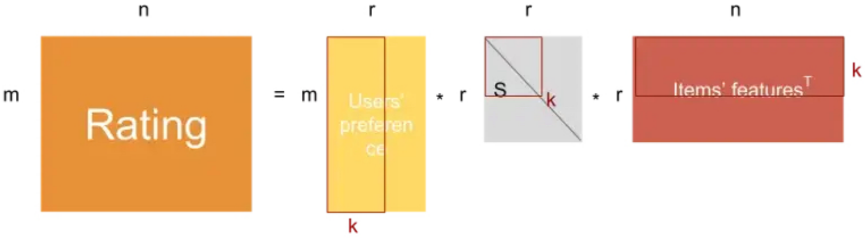
	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

- M: number of users
- N: number of items
- $R_{m,n}$: interaction between user m and item n
- Goal: to estimate ***missing values***
  - Example recommendations: top items in $\boldsymbol{R}_{m,:} \in \mathbb{R}^N$ for user m

Outline

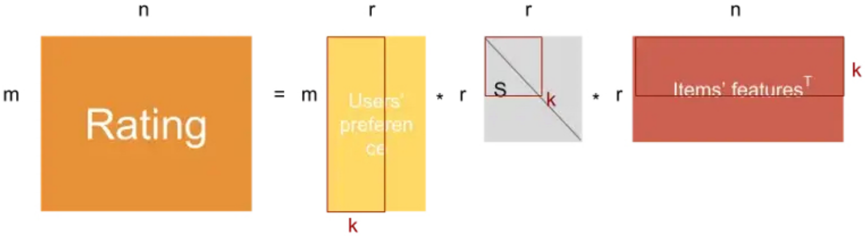
1. **Collaborative Filtering**
   - Matrix Factorization
   - AutoRec
2. **Implicit Feedback & Personalization**
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - Self-Attentive Sequential Recommendation (SASRec)
4. **Using Side Features**
   - Factorization Machines (FM)
   - Deep Models
5. **Performance Evaluation & System Design**

Singular Value Decomposition (SVD)



- Approximates $\boldsymbol{R}$ with a ***low-rank***, dense matrix
  - Denote K the target rank, $K \ll M,N$

Singular Value Decomposition (SVD)



- Approximates $\boldsymbol{R}$ with a ***low-rank***, dense matrix
  - Denote K the target rank, $K \ll M, N$
- Problems:
  - Slow to compute
  - No incremental update
    - Newly observed interactions?
    - New users/items?

Funk-SVD [5]

- Model parameters: $\boldsymbol{P}^{M \times K}$ and $\boldsymbol{Q}^{N \times K}$ such that

$$\boldsymbol{R}^{M \times N} \approx \hat{\boldsymbol{R}}^{M \times N} = \boldsymbol{P}^{M \times K}(\boldsymbol{Q}^{N \times K})^{\top}, \text{ where}$$

$$\hat{R}_{m,n} = \sum_{k=1}^{K} P_{m,k} Q_{k,n}$$

Funk-SVD [5]

- Model parameters: $\boldsymbol{P}^{M\times K}$ and $\boldsymbol{Q}^{N\times K}$ such that

$$\boldsymbol{R}^{M\times N} \approx \hat{\boldsymbol{R}}^{M\times N} = \boldsymbol{P}^{M\times K}(\boldsymbol{Q}^{N\times K})^{\top}, \text{ where}$$

$$\hat{R}_{m,n} = \sum_{k=1}^{K} P_{m,k} Q_{k,n}$$

- RMSE (root mean square error) loss:

$$L = \sum_{\{m,n:R_{m,n}>0\}} (R_{m,n} - \hat{R}_{m,n})^2 = \sum_{\{m,n:R_{m,n}>0\}} (R_{m,n} - \sum_{k=1}^{K} P_{m,k} Q_{k,n})^2$$

  - Regularization terms (minimizing $\|\boldsymbol{P}_{m,:}\|^2$ and $\|\boldsymbol{Q}_{n,:}\|^2$) omitted

Funk-SVD [5]

- Model parameters: $\boldsymbol{P}^{M\times K}$ and $\boldsymbol{Q}^{N\times K}$ such that

$$\boldsymbol{R}^{M\times N} \approx \hat{\boldsymbol{R}}^{M\times N} = \boldsymbol{P}^{M\times K}(\boldsymbol{Q}^{N\times K})^\top, \text{ where}$$

$$\hat{R}_{m,n} = \sum_{k=1}^{K} P_{m,k} Q_{k,n}$$

- RMSE (root mean square error) loss:

$$L = \sum_{\{m,n:R_{m,n}>0\}} (R_{m,n} - \hat{R}_{m,n})^2 = \sum_{\{m,n:R_{m,n}>0\}} (R_{m,n} - \sum_{k=1}^{K} P_{m,k} Q_{k,n})^2$$

  - Regularization terms (minimizing $\|\boldsymbol{P}_{m,:}\|^2$ and $\|\boldsymbol{Q}_{n,:}\|^2$) omitted
- ***SGD training***: for each randomly sampled batch of observed $R_{m,n}$, update $\boldsymbol{P}_{m,:}$ and $\boldsymbol{Q}_{n,:}$:
  - $\frac{\partial L}{\partial P_{m,k}} = -2(R_{m,n} - \hat{R}_{m,n})Q_{k,n}$ and $\frac{\partial L}{\partial Q_{n,k}} = -2(R_{m,n} - \hat{R}_{m,n})P_{m,k}$

Bias-SVD

- Introduces ***bias terms***

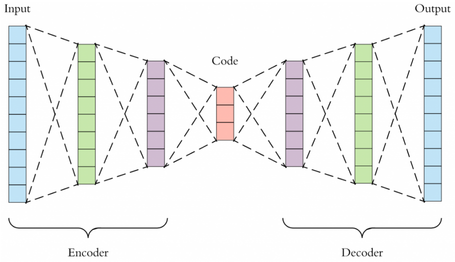
$$\hat{R}_{m,n} = \sum_{k=1}^{K} P_{m,k} Q_{k,n} + b_m^{\text{User}} + b_n^{\text{Item}} + b^{\text{Global}}$$

  - $\boldsymbol{b}^{\text{User}} \in \mathbb{R}^N$ compensates user bias
    - E.g., mean or kind
  - $\boldsymbol{b}^{\text{Item}} \in \mathbb{R}^M$ compensates item bias
    - E.g., good or bad
  - $b^{\text{Item}} \in \mathbb{R}$ compensates global bias
    - E.g., data preprocessing
- Greatly improves performance empirically

Outline

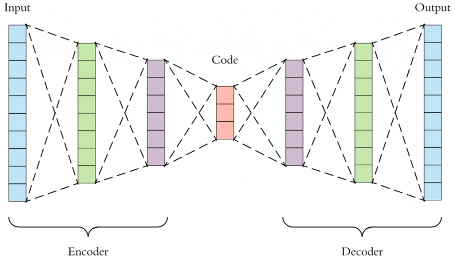
1. **Collaborative Filtering**
   - Matrix Factorization
   - AutoRec
2. **Implicit Feedback & Personalization**
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - Self-Attentive Sequential Recommendation (SASRec)
4. **Using Side Features**
   - Factorization Machines (FM)
   - Deep Models
5. **Performance Evaluation & System Design**

AutoRec [9]



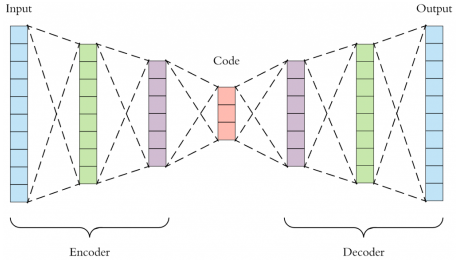
- Idea: to use an autoencoder $f(\cdot;\Theta)$ to reconstruct $\boldsymbol{R}^{M\times N}$
  - Bottleneck layer learns underlying manifold

AutoRec [9]



- Idea: to use an autoencoder $f(\cdot;\Theta)$ to reconstruct $\boldsymbol{R}^{M\times N}$
  - Bottleneck layer learns underlying manifold
- User-based: f takes $\boldsymbol{R}_{m,:}$ as input
  - Objective: $\arg\min_{\Theta}\sum_{m=1}^{M}\|\boldsymbol{R}_{m,:}-f(\boldsymbol{R}_{m,:};\Theta)\|^2$
- Item-based: f takes $\boldsymbol{R}_{:,n}$ as input
  - Objective: $\arg\min_{\Theta}\sum_{m=1}^{M}\|\boldsymbol{R}_{:,n}-f(\boldsymbol{R}_{:,n};\Theta)\|^2$

AutoRec [9]



- Idea: to use an autoencoder $f(\cdot;\Theta)$ to reconstruct $\boldsymbol{R}^{M\times N}$
  - Bottleneck layer learns underlying manifold
- User-based: f takes $\boldsymbol{R}_{m,:}$ as input
  - Objective: $\arg\min_{\Theta}\sum_{m=1}^{M}\|\boldsymbol{R}_{m,:}-f(\boldsymbol{R}_{m,:};\Theta)\|^2$
- Item-based: f takes $\boldsymbol{R}_{:,n}$ as input
  - Objective: $\arg\min_{\Theta}\sum_{m=1}^{M}\|\boldsymbol{R}_{:,n}-f(\boldsymbol{R}_{:,n};\Theta)\|^2$
- Learns to reconstruct data only, ***no representations for users/items***

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Explicit vs. Implicit Feedback

- So far, we use only the observed interactions (***explicit feedback***)
  - The missing values in $\boldsymbol{R}^{M\times N}$ are ignored
- Problems:
  - Very sparse
  - Only enough to train a global model; not personalized ones

Explicit vs. Implicit Feedback

- So far, we use only the observed interactions (***explicit feedback***)
  - The missing values in $\boldsymbol{R}^{M\times N}$ are ignored
- Problems:
  - Very sparse
  - Only enough to train a global model; not personalized ones
- Can we utilize the missing values (***implicit feedback***) in $\boldsymbol{R}^{M\times N}$?
- Semantics behind implicit feedback:
  - Negative (users are not interested in the items)
  - Neutral (the users have not interacted with the items yet)

Implicit Feedback & Personalization

- More available data, which enable ***personalized models***
  - $f(\cdot;\Theta)$ learns from $\boldsymbol{R}_{m,:} \in \mathbb{R}^N$ of a particular user

Implicit Feedback & Personalization

- More available data, which enable ***personalized models***
  - $f(\cdot;\Theta)$ learns from $\boldsymbol{R}_{m,:} \in \mathbb{R}^N$ of a particular user
- Let $\mathbb{I}^{m+}$ (and $\mathbb{I}^{m-}$) be the set of items that user m has interacted with (or not)
  - Basically, each item $\boldsymbol{x}^{(i)}$ $(i = 1, \cdots, N)$ is an N-dimensional one-hot vector
- Point-wise approaches
  - $f(\boldsymbol{x}^{(i)};\Theta) = 1$ if $\boldsymbol{x}^{(i)} \in \mathbb{I}^{m+}$; 0 otherwise
- Pair-wise approaches
  - $f(\boldsymbol{x}^{(i)};\Theta) > f(\boldsymbol{x}^{(j)};\Theta)$ for any $\boldsymbol{x}^{(i)} \in \mathbb{I}^{m+}$ and $\boldsymbol{x}^{(j)} \in \mathbb{I}^{m-}$

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Bayesian Personalized Ranking (BPR) [8]

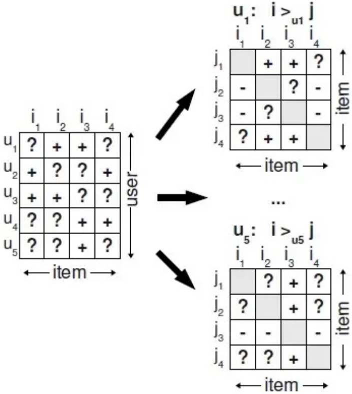
- Let $>_m$ represent the desired personalized total ranking of all items for user m
- BPR loss

$$L = -\ln \mathrm{P}(\Theta | >_m)$$

where

$$\mathrm{P}(\Theta | >_m) \propto \mathrm{P}(>_m | \Theta)\mathrm{P}(\Theta)$$

Pair-wise Data with Implicit Feedback



- Let $\mathbb{X} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) : \boldsymbol{x}^{(i)} \in \mathbb{I}^{m+}, \boldsymbol{x}^{(j)} \in \mathbb{I}^{m-}\}$ be the pair-wise dataset with implicit feedback for user m

Model & Objective

- BPR model (pair-wise): given $(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \in \mathbb{X}$, outputs

$$\sigma(f(\boldsymbol{x}^{(i)};\Theta) - f(\boldsymbol{x}^{(j)};\Theta)) \in \mathbb{R}$$

  - $\sigma(\cdot)$ is the logistic function
  - $f(\cdot;\Theta)$ is an underlying point-wise model (to be discussed later)
- BPR objective

$$\begin{aligned}\arg\min_{\Theta} L &= \arg\min_{\Theta} - \ln \mathrm{P}(>_m |\Theta) - \ln \mathrm{P}(\Theta)\\ &= \arg\min_{\Theta} - \sum_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \in \mathbb{X}} \ln \sigma(f(\boldsymbol{x}^{(i)};\Theta) - f(\boldsymbol{x}^{(j)};\Theta)) + \lambda \|\Theta\|^2\end{aligned}$$

  - Can be solved by gradient descent

Underlying Model $f(\cdot;\Theta)$

- BPR is a general framework and can work with different underlying models
- For example, let $f(\cdot;\Theta)$ be matrix factorization:

$$f(\boldsymbol{x}^{(i)}) = \boldsymbol{p}^\top \boldsymbol{q}^{(i)}$$

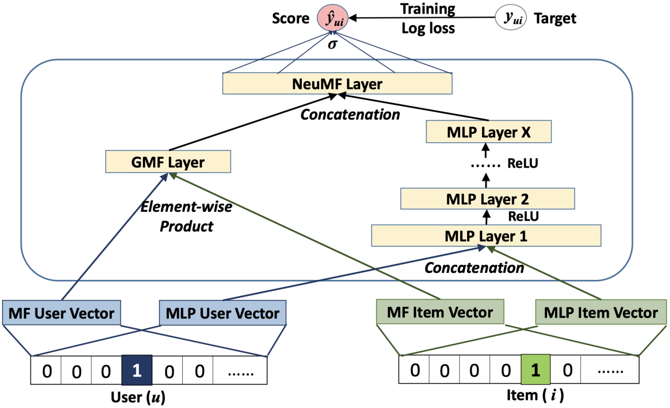
  where $\Theta = \{\boldsymbol{p}, \boldsymbol{q}^{(i)} \in \mathbb{R}^K\}_{i=1}^N$
- GD training: gradients $\frac{\partial L}{\partial \boldsymbol{p}}$, $\frac{\partial L}{\partial \boldsymbol{p}^{(i)}}$ can be easily computed

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Neural Matrix Factorization (NeuMF) [3]

- Can work with either point-wise or pair-wise loss (under BPR)
- ***Wide-and-deep*** architecture
  - Wide network: extended ***matrix factorization*** (no summation)
  - Deep network: ***representation learning***



Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Long- vs. Short-term Interests

- So far, we analyze user interests using long-term data $\boldsymbol{R}^{M\times N}$
  - For example, the $\boldsymbol{P}_{m,:}$ from matrix factorization represents the ***long-term interests*** of user m
- In practice, users also have ***short-term interests***
  - E.g., news/trending topics, social events, seasonal events, time-limited sales, etc.
- Interactions given by a user are not i.i.d.

Long- vs. Short-term Interests

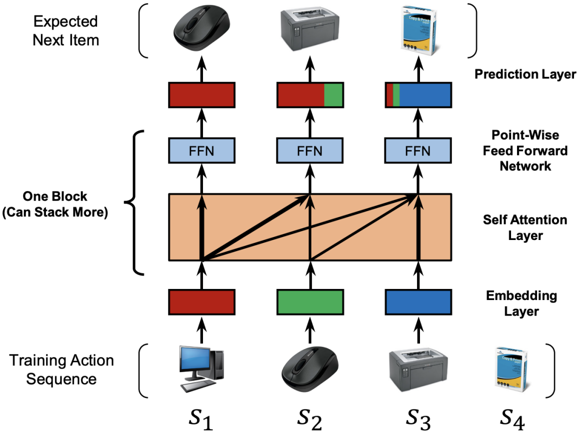
- So far, we analyze user interests using long-term data $\boldsymbol{R}^{M\times N}$
  - For example, the $\boldsymbol{P}_{m,:}$ from matrix factorization represents the ***long-term interests*** of user m
- In practice, users also have ***short-term interests***
  - E.g., news/trending topics, social events, seasonal events, time-limited sales, etc.
- Interactions given by a user are not i.i.d.
- Why not use ***sequential learning*** models?
  - CNNs, RNNs, or transformers?

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. **Sequence Awareness**
   - **Self-Attentive Sequential Recommendation (SASRec)**
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Self-Attentive Sequential Recommendation (SASRec) [4]

- Sequential, collaborative rating tensor $\mathbf{R}^{M\times N\times T}$
  - $R_{m,n,t}$ the interaction between user m and item n at time t
- SASRec model (based on the masked self attention):
  - Input: $S^{T\times N}$ whose rows are one-hot vectors



Model Architecture & Loss

- Architecture similar to the decoder of "Attention is all you need" paper
  - Residual connects, layer normalization, dropout, etc.
- Point-wise loss:

$$\sum_{m,t} \|R_{m,:,t} - \boldsymbol{A}^{(L)}_{t,:}\boldsymbol{Q}^{\top})\|^2$$

  - $\boldsymbol{Q} \in \mathbb{R}^{N \times K}$ is embeddings of all items
  - $\boldsymbol{A}^{(L)} \in \mathbb{R}^{T \times K}$ is output of the last (L-th) attention block

Model Architecture & Loss

- Architecture similar to the decoder of "Attention is all you need" paper
  - Residual connects, layer normalization, dropout, etc.
- Point-wise loss:

$$\sum_{m,t} \|R_{m,:,t} - \boldsymbol{A}^{(L)}_{t,:}\boldsymbol{Q}^\top)\|^2$$

  - $\boldsymbol{Q} \in \mathbb{R}^{N\times K}$ is embeddings of all items
  - $\boldsymbol{A}^{(L)} \in \mathbb{R}^{T\times K}$ is output of the last (L-th) attention block
- As compared with MF (where $R_{m,:} \approx \boldsymbol{P}_{m,:}\boldsymbol{Q}^\top$), $\boldsymbol{A}^{(L)}_{t,:}$ is similar to "user embedding"
  - Based on ***sequence behavior*** only; not tied to a specific user

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Content-based vs. Collaborative Filtering

- Content-based algorithms
  - Assume that ***user/item features*** are available
    - E.g., user profiles, product descriptions, etc.

- Collaborative filtering (CF)
  - Assumes that ***user-item interactions*** are available
    - E.g, user1 clicks/likes/rates item100
  - Popular since
    - There's no need to analyze users/items
    - Performs well empirically

Content-based vs. Collaborative Filtering

- Content-based algorithms
  - Assume that ***user/item features*** are available
    - E.g., user profiles, product descriptions, etc.
  - Today, with DL models like BERT [1] or CLIP [6], embedding users/items becomes easy
- Collaborative filtering (CF)
  - Assumes that ***user-item interactions*** are available
    - E.g, user1 clicks/likes/rates item100
  - Popular since
    - There's no need to analyze users/items
    - Performs well empirically

Content-based vs. Collaborative Filtering

- Content-based algorithms
  - Assume that ***user/item features*** are available
    - E.g., user profiles, product descriptions, etc.
  - Today, with DL models like BERT [1] or CLIP [6], embedding users/items becomes easy
- Collaborative filtering (CF)
  - Assumes that ***user-item interactions*** are available
    - E.g, user1 clicks/likes/rates item100
  - Popular since
    - There's no need to analyze users/items
    - Performs well empirically
- Can we consider content features in CF?

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

Factorization Machines (FM) [7]

- Feature-rich dataset: $\mathbb{X} = \{(\boldsymbol{x}^{(i)} \in \mathbb{R}^D, y^{(i)})\}_{i \in \text{Interactions}}$
  - $\boldsymbol{x}^{(i)}$ contains one-hot encodings features for users and items

Feature vector $\mathbf{x}$ | 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	$y^{(6)}$

- Model:

$$\hat{y}(\boldsymbol{x}; b, \boldsymbol{w}, \boldsymbol{W}) = b + \sum_{s=1}^{D} w_s^\top x_s + \sum_{s=1}^{D} \sum_{t=s+1}^{D} W_{s,t} x_s x_t$$

  - Linear regressor with ***cross-feature weights***

Connection with Matrix Factorization

- Model:

$$\hat{y}(\boldsymbol{x};b,\boldsymbol{w},\boldsymbol{W}) = b + \sum_{s=1}^{D} w_s^{\top} x_s + \sum_{s=1}^{D} \sum_{t=s+1}^{D} W_{s,t} x_s x_t$$

- The weights ***corresponding to the one-hot encoding dimensions of user and item*** perform matrix factorization

Feature vector **x**

	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	$y^{(6)}$

Reducing Cross-Feature Costs

- $\boldsymbol{W} \in \mathbb{R}^{D\times D}$ may be too large to learn!

Reducing Cross-Feature Costs

- $\boldsymbol{W} \in \mathbb{R}^{D \times D}$ may be too large to learn!
- Simplification:

$$\boldsymbol{W}^{D \times D} = \boldsymbol{V}^{D \times K} (\boldsymbol{V}^{D \times K})^\top, \text{ where } K \ll D$$

- The cross-feature term becomes

$$\sum_{s=1}^{D} \sum_{t=s+1}^{D} W_{s,t} x_s x_t = \sum_{s=1}^{D} \sum_{t=s+1}^{D} \sum_{k=1}^{K} V_{s,k} V_{t,k} x_s x_t$$

  - Evaluation complexity $O(D^2 K)$

Reducing Cross-Feature Costs

- $\boldsymbol{W} \in \mathbb{R}^{D \times D}$ may be too large to learn!
- Simplification:

$$\boldsymbol{W}^{D \times D} = \boldsymbol{V}^{D \times K}(\boldsymbol{V}^{D \times K})^\top, \text{ where } K \ll D$$

- The cross-feature term becomes

$$\sum_{s=1}^{D} \sum_{t=s+1}^{D} W_{s,t} x_s x_t = \sum_{s=1}^{D} \sum_{t=s+1}^{D} \sum_{k=1}^{K} V_{s,k} V_{t,k} x_s x_t$$

  - Evaluation complexity $O(D^2 K)$
- Further reduction

$$\begin{aligned}
&\textstyle\sum_{s=1}^{D} \sum_{t=s+1}^{D} \sum_{k=1}^{K} V_{s,k} V_{t,k} x_s x_t \\
&= \tfrac{1}{2} \left(\textstyle\sum_{s=1}^{D} \sum_{t=1}^{D} \sum_{k=1}^{K} V_{s,k} V_{t,k} x_s x_t - \sum_{s=1}^{D} \sum_{k=1}^{K} V_{s,k} V_{s,k} x_s x_s \right) \\
&= \tfrac{1}{2} \textstyle\sum_{k=1}^{K} \left[\left(\sum_{s=1}^{D} V_{s,k} x_s \right) \left(\sum_{t=1}^{D} V_{t,k} x_t \right) - \sum_{s=1}^{D} V_{s,k}^2 x_s^2 \right] \\
&= \tfrac{1}{2} \textstyle\sum_{k=1}^{K} \left[\left(\sum_{s=1}^{D} V_{s,k} x_s \right)^2 - \sum_{s=1}^{D} V_{s,k}^2 x_s^2 \right]
\end{aligned}$$

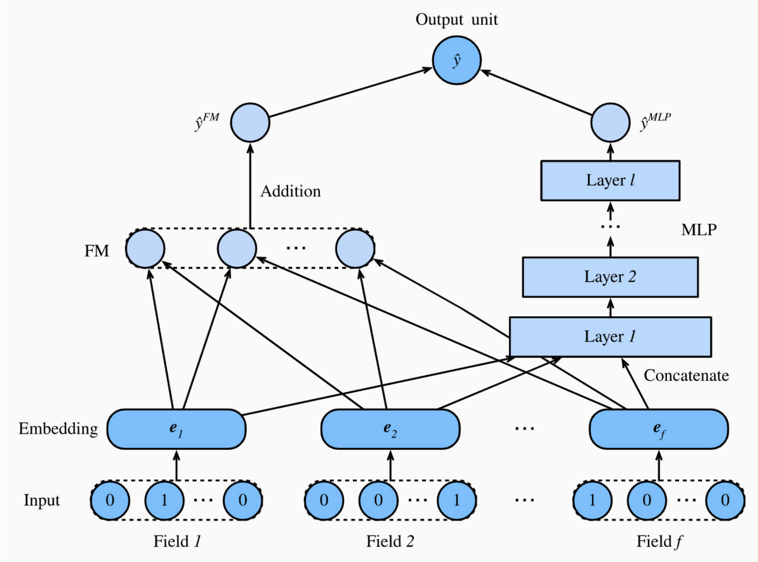
  - Evaluation complexity $O(DK)$

Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. Performance Evaluation & System Design

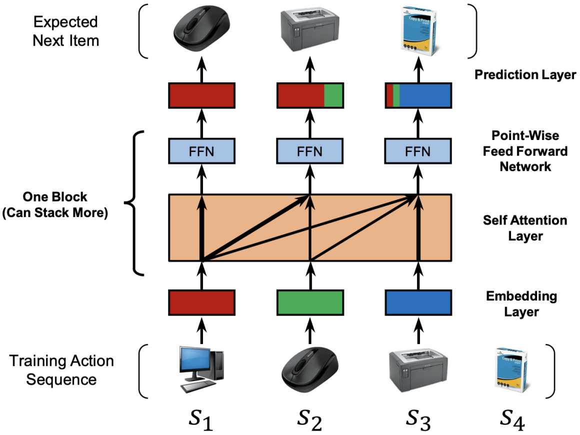
Deep Factorization Machines [2]

- Still "wide-and deep," but consider side user/item features



Explicit User Features in SASRec [4]

- Let $\boldsymbol{P} \in \mathbb{R}^{M \times K}$ be the user embedding matrix
  - Obtained via side user features
- Point-wise loss: $\sum_{m,t} \|R_{m,:,t} - (\boldsymbol{P}_{m,:} + \boldsymbol{A}^{(L)}_{t,:})\boldsymbol{Q}^\top)\|^2$



Outline

1. Collaborative Filtering
   - Matrix Factorization
   - AutoRec
2. Implicit Feedback & Personalization
   - Bayesian Personalized Ranking (BPR)
   - Neural Matrix Factorization (NeuMF)
3. Sequence Awareness
   - Self-Attentive Sequential Recommendation (SASRec)
4. Using Side Features
   - Factorization Machines (FM)
   - Deep Models
5. **Performance Evaluation & System Design**

Performance Evaluation Metrics (1/2)

- Common metrics: average ***precision@k***, ***recall@k***, or ***NDCG@k*** across users

Performance Evaluation Metrics (1/2)

- Common metrics: average ***precision@***k, ***recall@***k, or ***NDCG@***k across users
- For each user m, let
  - $y = A$ be next interacted item in the ground truth
  - $\hat{\mathbb{Y}} = \{B, A, C, F, \cdots\}$ be top items predicted by Model 1
  - $\tilde{\mathbb{Y}} = \{G, H, A, B, \cdots\}$ be top items predicted by Model 2
- Precision@k:

$$\frac{\text{TP@}k}{\text{TP@}k + \text{FP@}k}$$

  - Model 1: precision@1 = 0/1, precision@2 = 1/2, precision@3 = 1/3
  - Model 2: precision@1 = 0/1, precision@2 = 0/2, precision@3 = 1/3
- Recall@k (or hit@k):

$$\frac{\text{TP@}k}{\text{TP@}k + \text{FN@}k}$$

  - Model 1: hit@1 = 0/1, hit@2 = 1/1, hit@3 = 1/1
  - Model 2: hit@1 = 0/1, hit@2 = 0/1, hit@3 = 1/1

Performance Evaluation Metrics (2/2)

- NDCG@k (Normalized Discounted Cumulative Gain):

$$\begin{cases} \frac{1}{\log_2(i_A+1)}, & \text{if } A \in \mathbb{Y} \\ 0, & \text{otherwise.} \end{cases}$$

  where i_A is the index of A in $\mathbb{Y}$
  - Model 1: NDCG@1 $= 0$, NDCG@2 $= 0.63$, NDCG@3 $= 0.63$
  - Model 2: NDCG@1 $= 0$, NDCG@2 $= 0$, NDCG@3 $= 0.5$

Performance Evaluation Metrics (2/2)

- NDCG@k (Normalized Discounted Cumulative Gain):

$$\begin{cases} \frac{1}{\log_2(i_A+1)}, & \text{if } A \in \mathbb{Y} \\ 0, & \text{otherwise.} \end{cases}$$

  where i_A is the index of A in $\mathbb{Y}$
  - Model 1: NDCG@1 $=$ 0, NDCG@2 $=$ 0.63, NDCG@3 $=$ 0.63
  - Model 2: NDCG@1 $=$ 0, NDCG@2 $=$ 0, NDCG@3 $=$ 0.5
  - At same $k = 3$, Model 1 still outperforms Model 2 in terms of NDCG

Remarks on Recommender Systems in Industry

- Two steps to cope with very large M, N and multiple objectives:

1. Candidate retrieval (from millions to hundreds)
   - Approximate KNN search of relevant items from simple (MF) model
   - Popular or trending items (to avoid the ***cold-start problem***)
   - Diffused items along social graph
   - Diverse items (in geographies, topics, etc.)
   - User features (for personalization)
2. Scoring & re-ranking (from hundreds to tens)
   - More expensive (deep) models are used to
   - Consolidate results to max click rate, watch time, session time, etc.
   - Personalize UI list by
     - Removing dislikes
     - Increasing freshness, diversity, fairness, etc.

- Step 2 can run at client side

Reference I

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.

[2] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He.
Deepfm: a factorization-machine based neural network for ctr prediction.
In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731, 2017.

[3] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua.
Neural collaborative filtering.
In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

Reference II

[4] Wang-Cheng Kang and Julian McAuley.
Self-attentive sequential recommendation.
In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[5] Arkadiusz Paterek.
Improving regularized singular value decomposition for collaborative filtering.
In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.
Learning transferable visual models from natural language supervision.
In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Reference III

[7] Steffen Rendle.
Factorization machines.
In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[8] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme.
Bpr: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[9] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie.
Autorec: Autoencoders meet collaborative filtering.
In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112, 2015.