Exploit Side Information for Recommender System

Qing Guo, Zhu Sun, Yin-Leng Theng

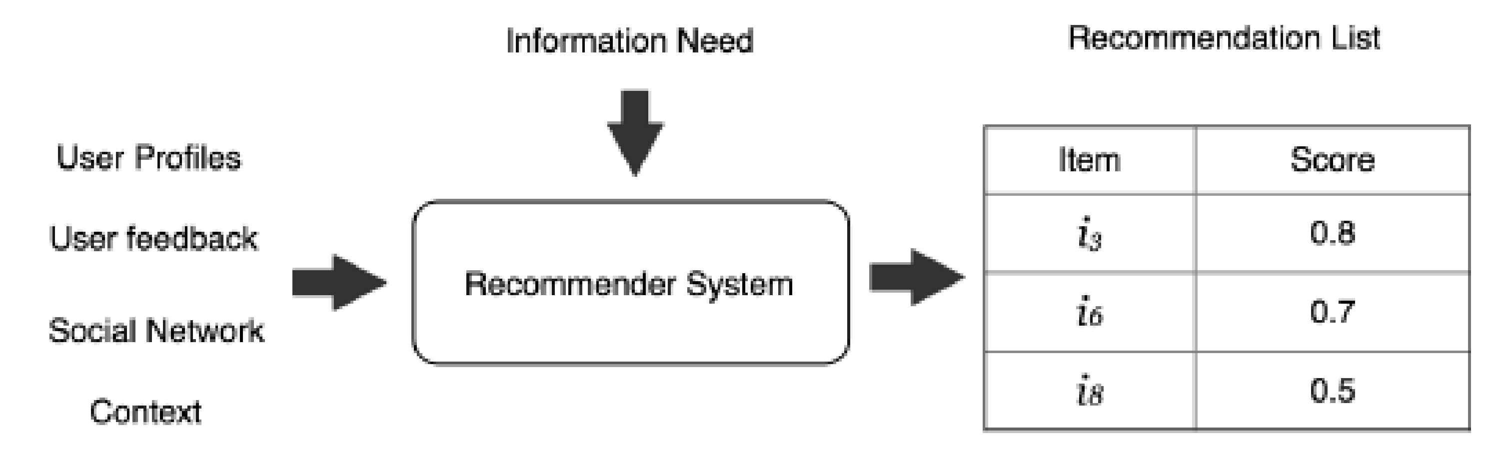
Nanyang Technological University, Singapore

Agenda

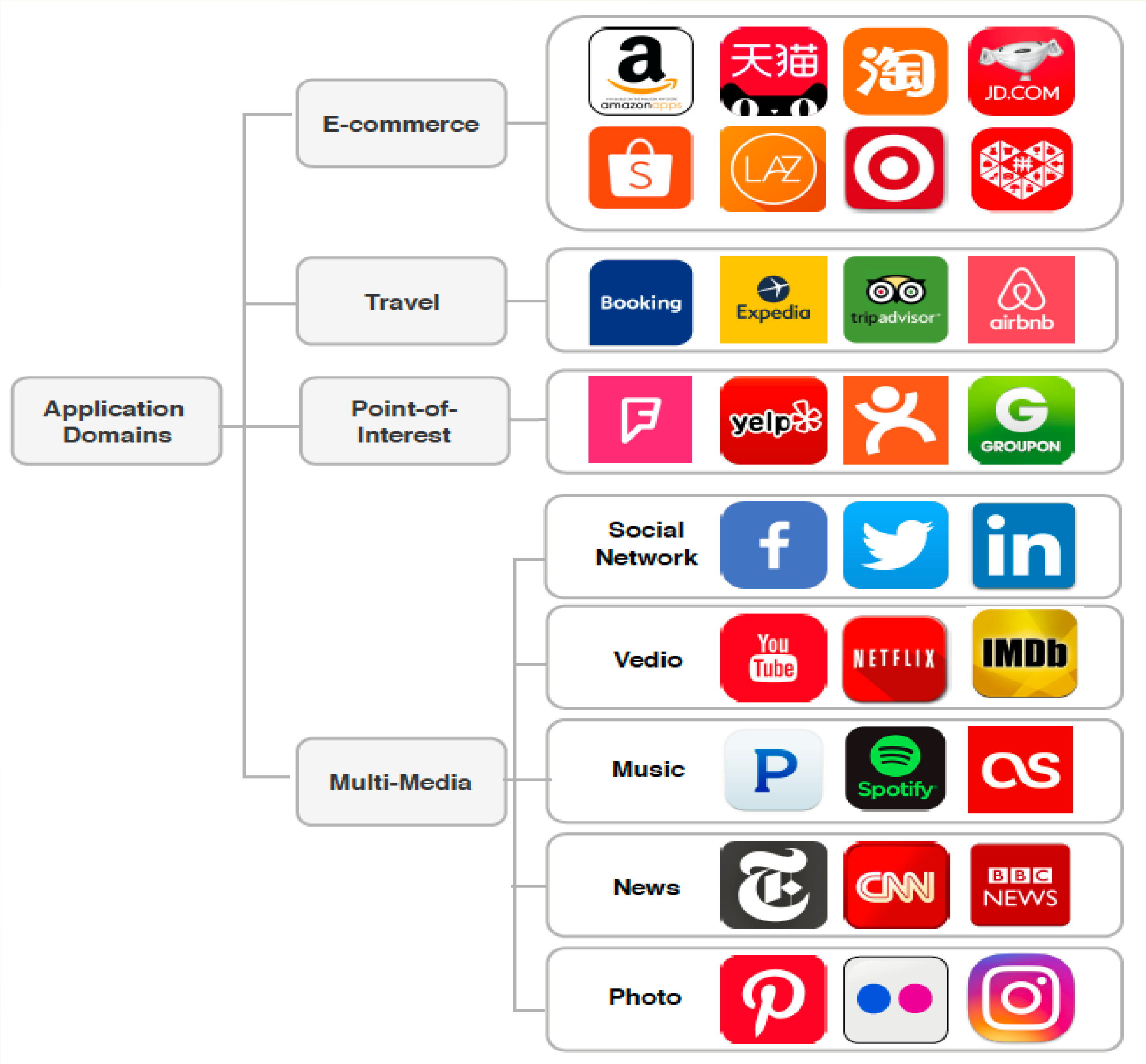
- **Introduction**
  - Background
  - Challenges of existing recommender systems
  - Significance of this tutorial
- **Overview of recommender systems with side information**
  - Evolution of fundamental methodologies
  - Evolution of side information
- **Recommender system with side information**
  - Memory-based methods with side information
  - Latent factor models with side information
  - Representation learning models with side information
  - Deep learning models with side information
- **Future directions**

Background

- Recommender system has become a vital tool to tackle *information overload issue*.

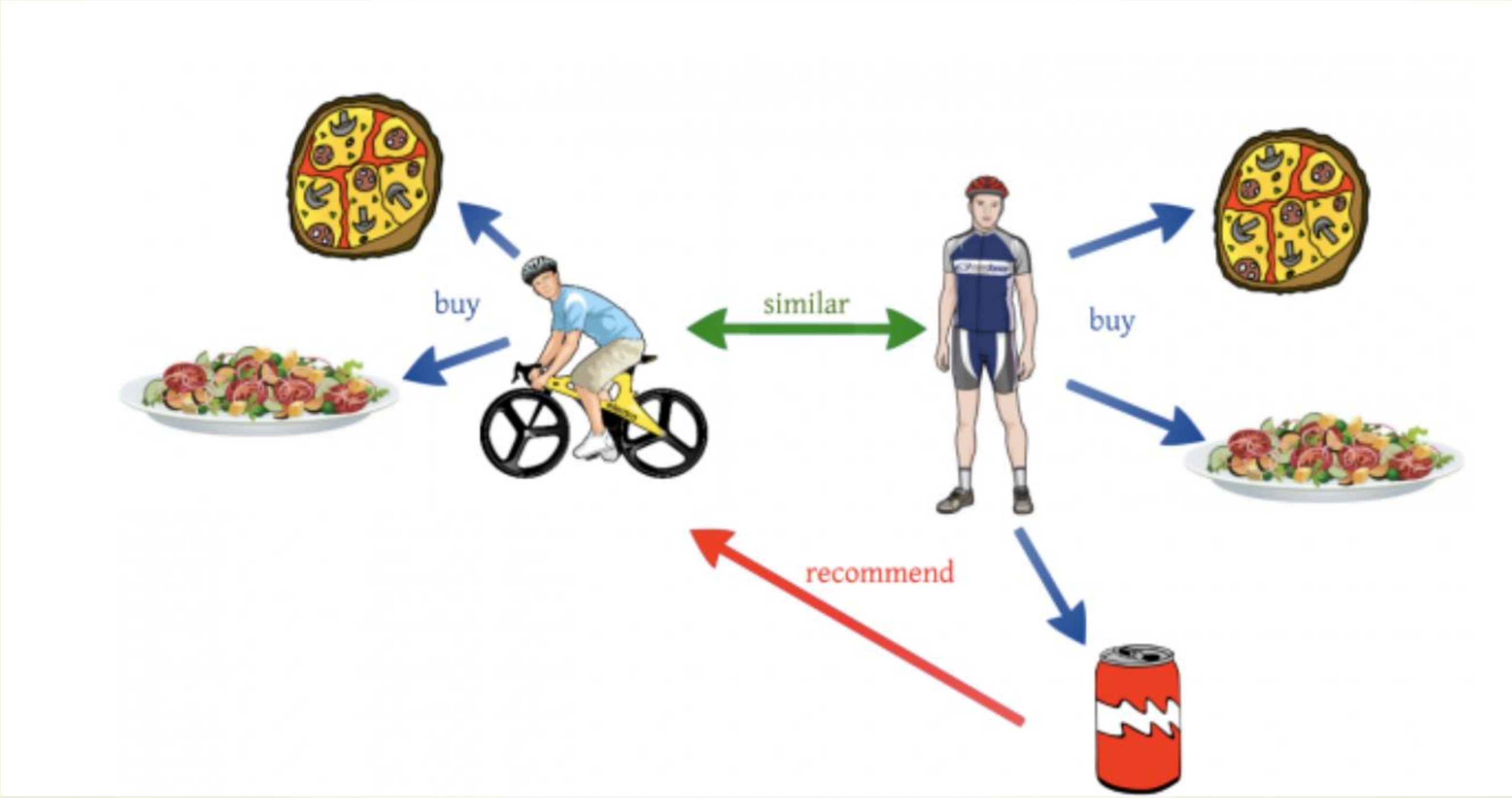


- Recommender system has been adopted in various domains.
- Recommender system has become an important topic in research.



Challenges of Recommender Systems

- Most recommender systems are based on collaborative filtering.

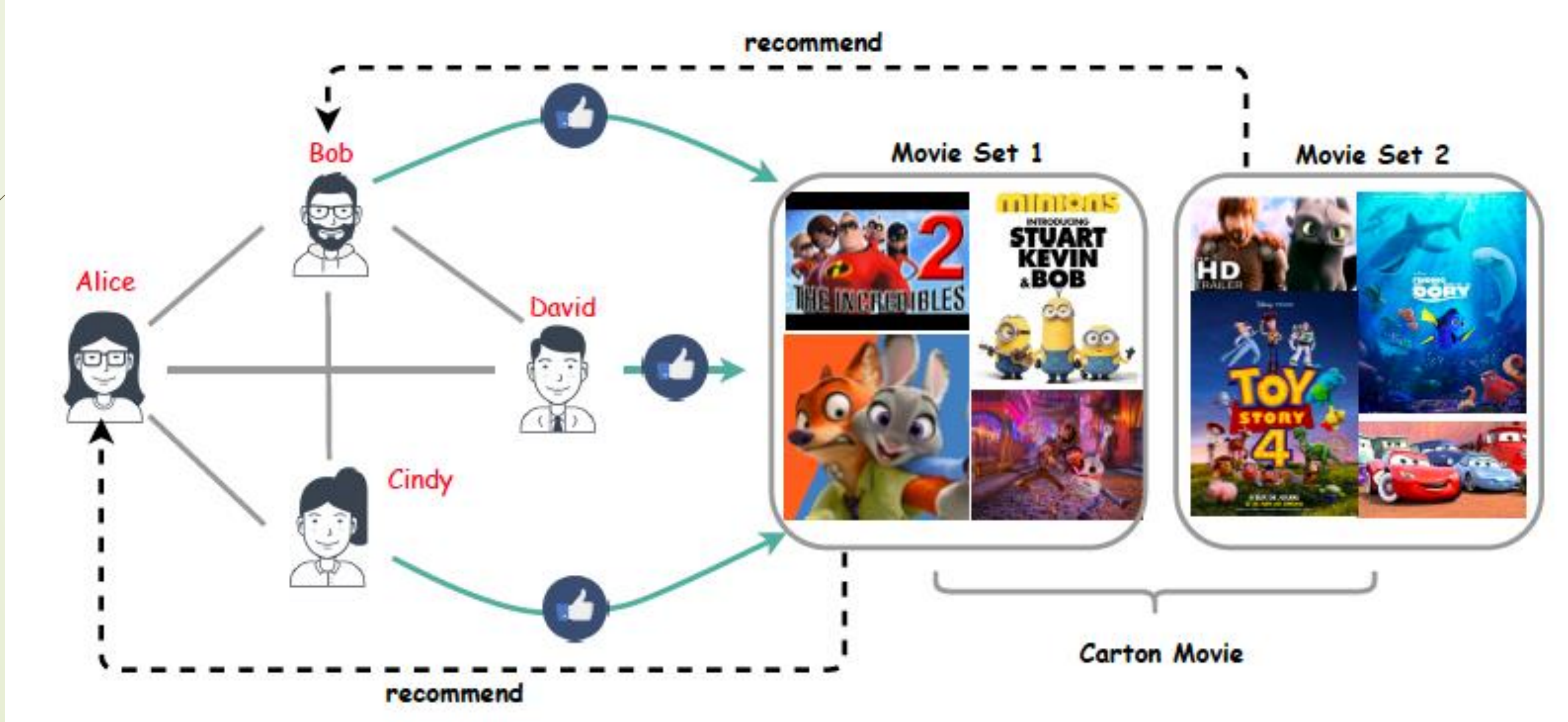


Overview of Recommender Systems

- Traditional CF-based methods purely rely on user-item interaction matrix, assuming that a user's preference can be inferred by aggregating the tastes of other similar users.
- Two outstanding issues of collaborative filtering.
  - Data sparsity. Users face an extremely large amount of items, even the most active users only rate a small set of items. It would be difficult to learn users' preferences.
  - Cold start.
    - There's little chance for a cold-start items getting exposed to users.
    - It's hard to learn a cold-start user's preference.

Overview of Recommender Systems

- A toy example to show how side information can help.



Significance of this tutorial

- Existing survey papers mainly focus on a single perspective instead of conducting a thorough investigation. That is, they either discuss the methodologies or side information.
- Recent research puts lots of efforts on exploring more sophisticated structures to represent various kind of side information including flat features, network features, hierarchical feature and knowledge graph.
- The more complex representation of side information often need to be coupled with more advanced methodologies for fully realizing the value of side information.
- This tutorial provides a comprehensive and systematic review from both representation of side information and methodology perspectives.

How this tutorial is prepared

- To cover recent studies, hundreds of papers published in prestigious venues related recommender system have been collected, such as NIPS, ICML, KDD, WWW, WSDM, IJCAI, AAAI, SIGIR, RECSYS, CIKM, TKDE…
- Google scholar is primarily used for searching papers. Other academic search engines are also used such as ACM DIGITAL LIBRARY, IEEE Xplore, Web of Science and Springer.
- A number of keywords are used to search related papers, such as recommender system, recommendation, side information, knowledge graph …

Overview of Recommender Systems with Side Information

- Recommender systems predict users' preferences on items to assist users for making easier decisions. They can be classified based on recommendation strategies. Tasks and outputs.

Perspective	Strategies	Tasks	Outputs
Category	• Content-based filtering • Collaborative filtering • Hybrid methods	• General • Temporal • Sequential	• Rating Prediction • Item Ranking

Overview of Recommender Systems with Side Information

- Classification by strategies.
  - Content-based filtering
    - It mainly uses user profiles and item descriptions to infer users' preferences towards items.
    - It may suffer from over-specialization, that is, users are constantly get recommendations which are similar to what they have bought before.
  - Collaborative filtering
    - It aims to predict users' preferences by learning from user-item historical interactions, either in the form of explicit feedback (rating or reviews) or implicit feedback (click or view). It can be further categorized into memory- and model-based methods.
    - However, it often suffers from data sparsity and cold-start issues.
  - Hybrid methods.
    - They take advantage of both CF- and content-based approaches to overcome the shortcomings of each methods.
    - Two types of fusing strategies: early fusion and late fusion.

Overview of Recommender Systems

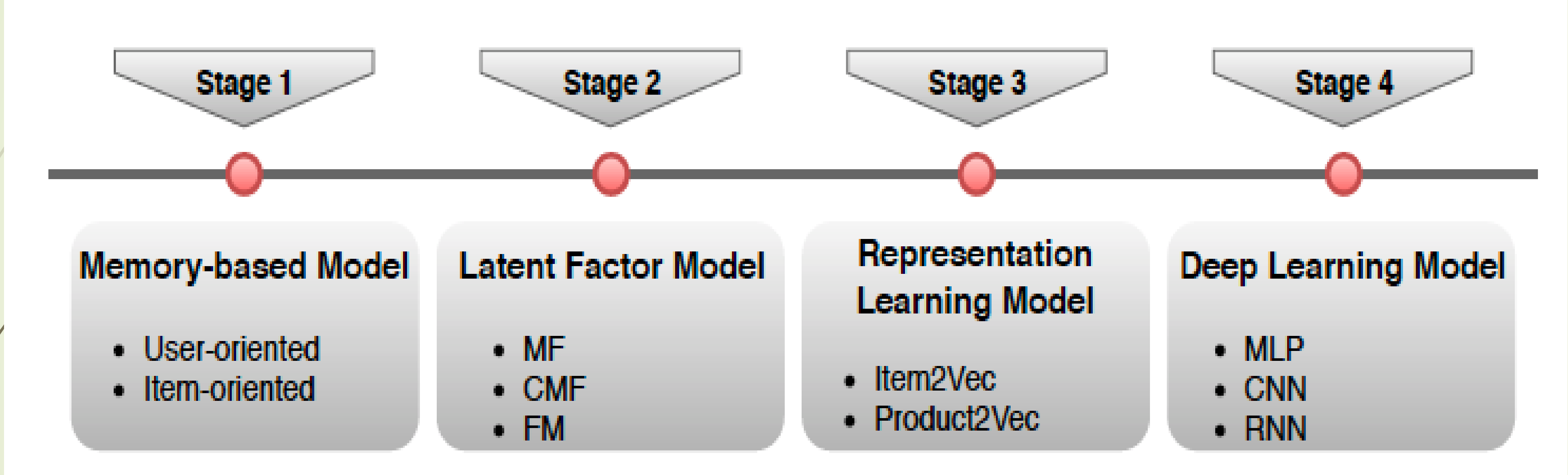
- Classification by tasks
  - General recommendation. It normally leverage historical user-item interactions to recommend top-N items for users.
  - Temporal recommendation. It usually captures users' preferences given a timestamp or a time period. To be an effective temporal recommendation model, the key is to model the dynamics of user preferences.
  - Sequential recommendation (or next-item recommendation). It predicts users' next preferences based on their most recent activities. That is, it models the sequential patterns among successive items.

- Classification by outputs
  - Rating-based recommendation. It predicts users' explicit preference scores towards items.
  - Ranking-based recommendation. It focuses on the relative ranking positions of items and usually generates a top-N item recommendation list for each user.

Overview of Recommender Systems

- Discussion
  - Existing classification taxonomies cannot deliver a complete picture of the research studies in recommendation with side information.
  - We create a new taxonomy to classify related literature based on two aspects: the representation of side information and methodologies.

Evolution of Fundamental Methodologies for Recommendation



Evolution of Fundamental Methodologies for Recommendation

- Memory-based approaches
  - Referred as neighborhood-based collaborative filtering
  - User-based CF and item-based CF
  - User-based CF: identify similar users/neighbors and aggregate the interests of neighbors for recommendations.
  - Item-based CF: estimate a user's preference for an item based on the ratings of similar items rated by the same user.
  - Discussion
    - It's time-consuming to search similar users and items in large scale user or item space.
    - It's hard to estimate the preferences of cold-start users or items.

Evolution of Fundamental Methodologies for Recommendation

- Model-based approaches
  - Model-based approaches build predictive models by adopting machine learning techniques on user-item rating matrix to uncover user behavior patterns.
  - They can categorized latent factor models, representation learning models and deep learning models.

- Latent factor models
  - Assume both users and items can be characterized by a few latent features.
  - Decompose the high-dimensional user-item rating matrix into low-dimensional user and item matrices.

$$\hat{r}_{ui} = q_i^T p_u$$

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

Rating Matrix (Item × User)

User	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

=

User Matrix

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

X

Item Matrix

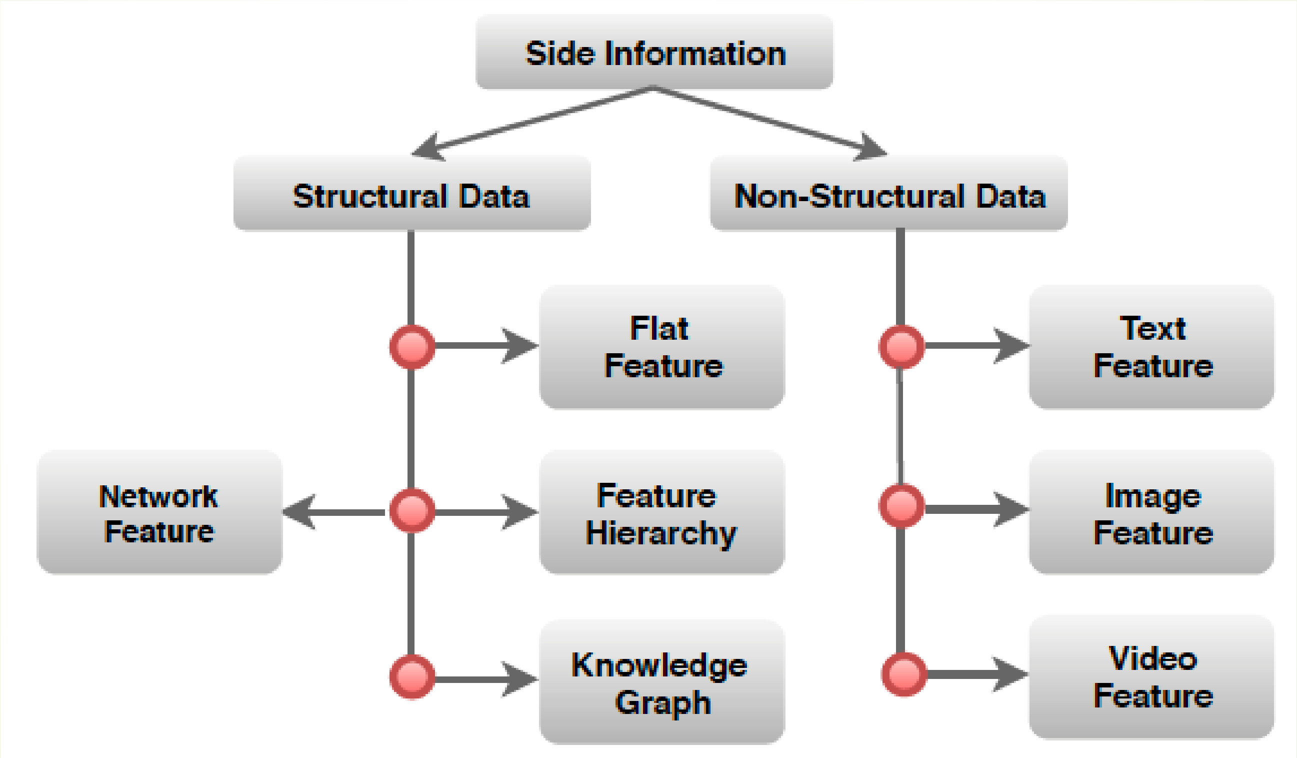
W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

- Representation learning models (RLM)
  - RLM is originally inspired by word embedding.
  - Capture the local item relationships by model item co-occurrence in each user's interactions.
  - Many Item2Vec based recommendation approaches are based on word2Vec idea (Barkan et al, 2016; Liang et al., 2016; Grbovic et al., 2015, Feng et al., 2018).
- Deep learning models (DLMs)
  - Deep learning models can learn high-order and non-linear latent representations via various types of activation functions (e.g., sigmoid, ReLu).
  - Recurrent neural network (RNN) based approaches have shown strong capabilities for sequential recommendation.
  - Convolutional neural network (CNN) are able to extract hierarchical features of various contextual information.

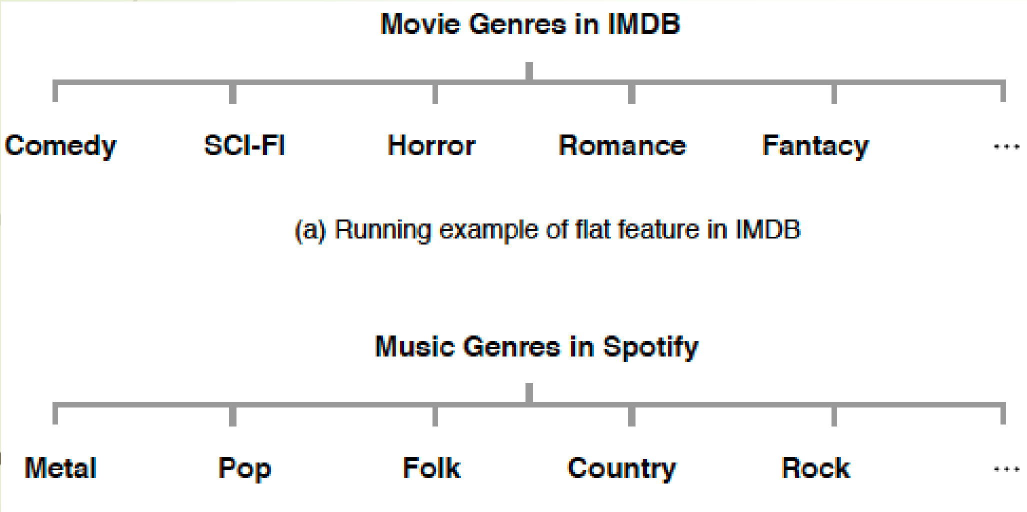
- Discussion
  - LFM and RLM can be considered as special cases DLM, i.e., the shallow network (He et al., 2016).
  - Matrix factorization can be regarded as a one-layer neural network that transform one-hot user and item vectors into dense representations.
  - But DLMs cause more computational cost.
  - How to incorporate more side information in DLMs in an efficient way remains a promising research direction.

Evolution of Side Information for Recommendation

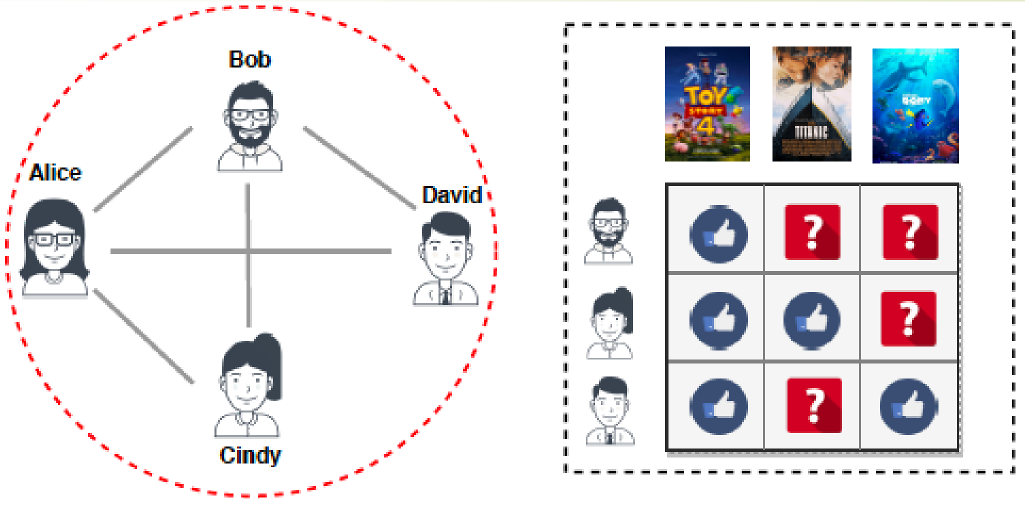
- To resolve data sparsity and cold-start issues, side information are widely used in recommender systems.



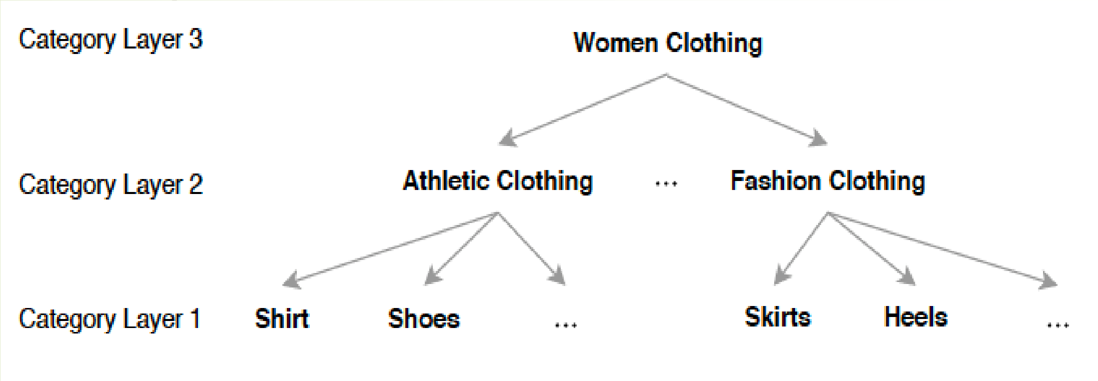
- Flat features (FF)
- Network features (NF)

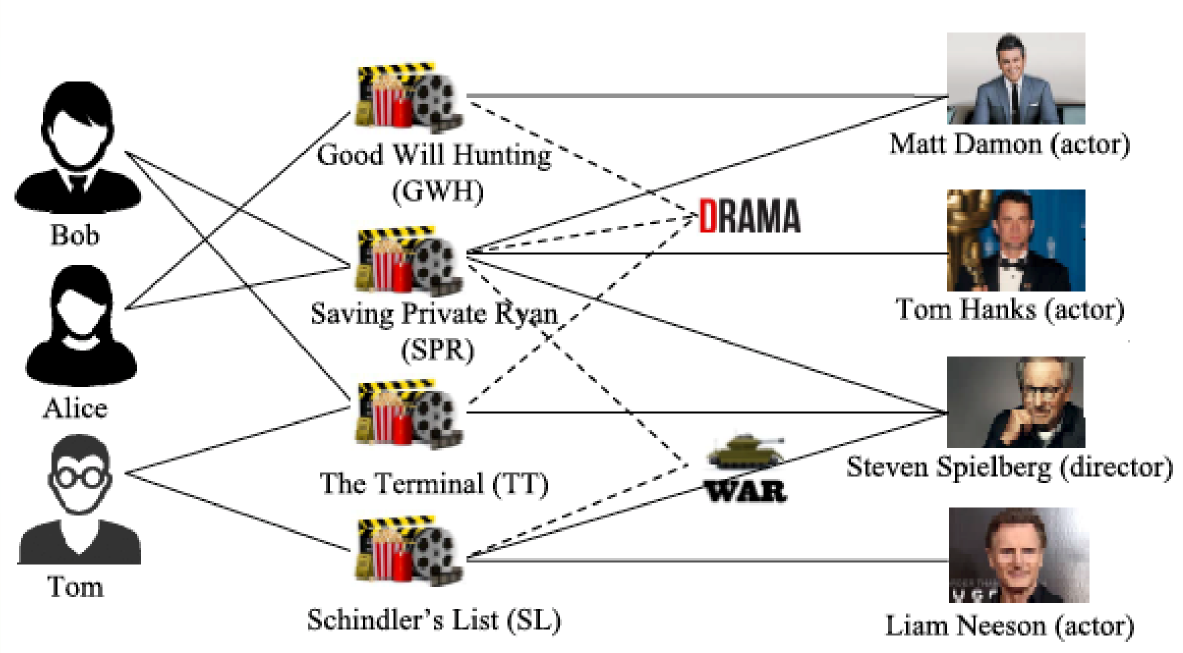


(a) Running example of flat feature in IMDB



- Feature hierarchy (FH)
- Knowledge graph (KG)
- Non-structural data





Memory-based Methods with Side Information

- Many memory-based methods consider flat features (FF) for recommendation in pre- or post-filtering manner based on the assumption that users sharing similar feature would also share similar interests for items. For example, Hwang et al. (2012) introduce *category experts*. Davidson et al. (2010) propose a YouTube recommender system where flat categories are used to promote the recommendation diversity.
- Memory-based models also exploit social network in trust-aware recommender systems. They take an assumption that friend share similar interests (Guo et al., 2015; Catherine et al., 2016).
- Several studies also attempt to fuse feature hierarchy (FH) into memory-based methods by exploiting user- and item-taxonomy distribution. For example, Ziegler et al. (2004) represent each item and user's preference with a taxonomy distribution vector. Then apply user-based CF to generate recommendation.

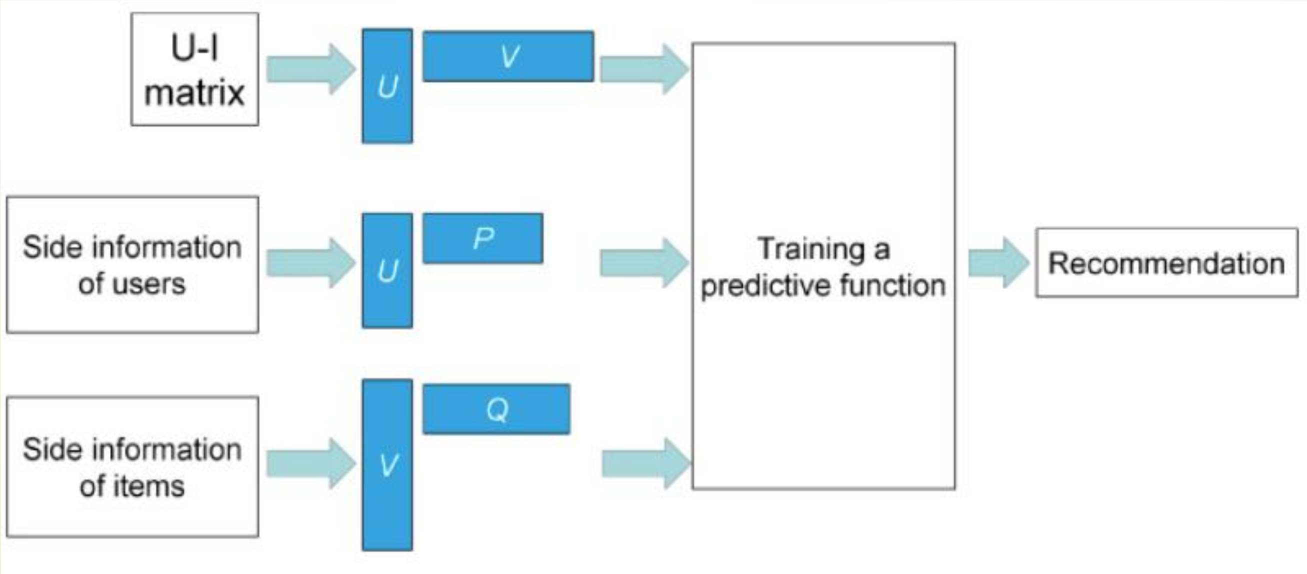
- Some studies adopt text features (e.g., reviews) via word-level text similarity or extracted sentiment. For instance, Teriz et al. (2014) measure user similarity based on their reviews. Pappas et al. (2013) develop a sentiment-aware nearest neighbor model (SANN) for recommending TED talks.

- Discussion
  - Memory-based methods are less effective due to the time-consuming search in large scale user and item space.
  - The weak scalability of memory-based methods limits the strength of side information as well as more complex relationships between users and items.

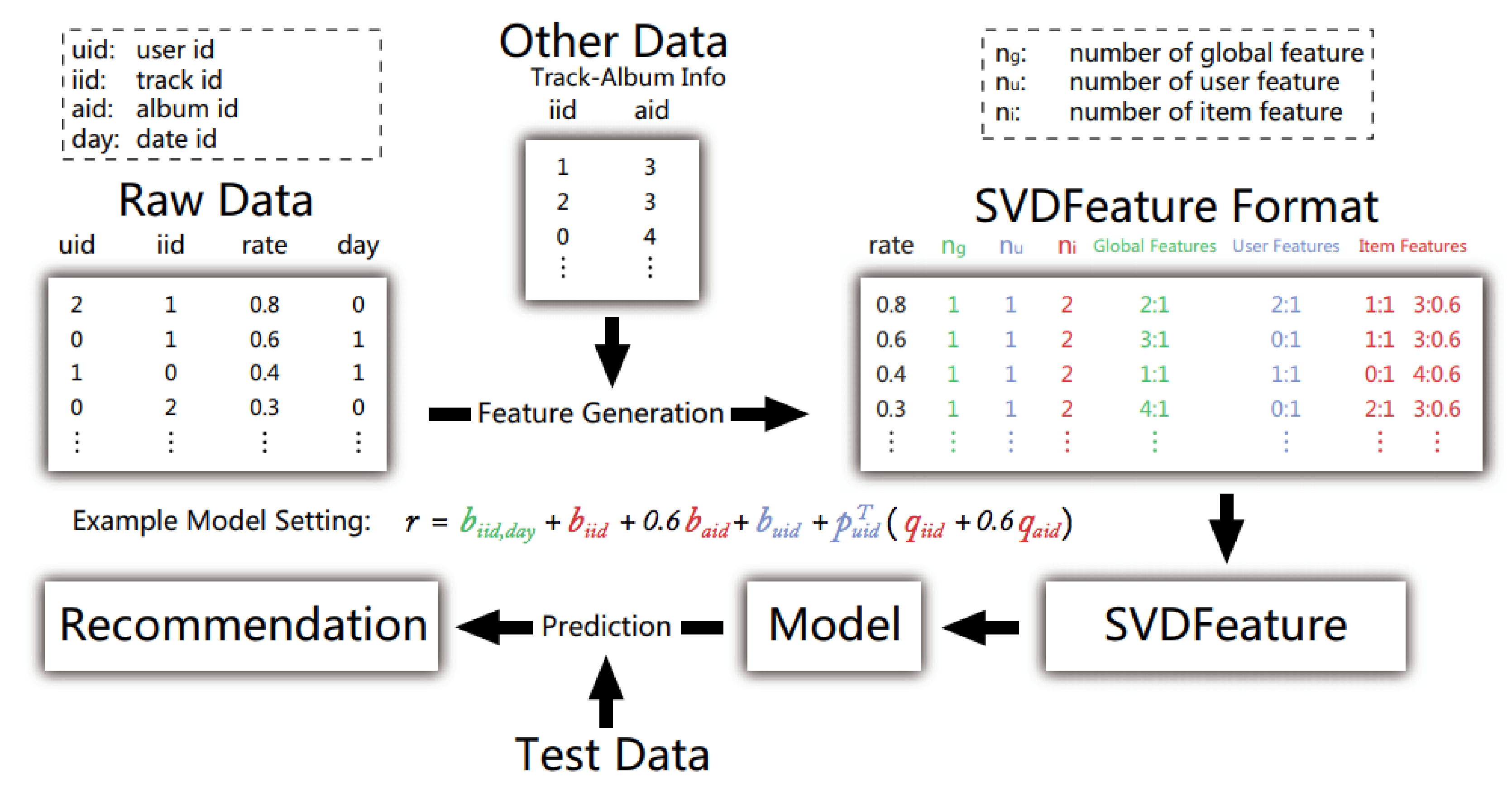
Latent Factor Models with Side Information

- LFMs include matrix factorization (MF), weighted non-negative factorization (WNMF), Bayesian personalized ranking (BPR), tensor factorization (TensorF), factorization machine (FM), SVD++ and timeSVD++.

- Latent factor models with flat features (LFMs + FF)
  - Collective matrix factorization (CMF) (Singh et al., 2008) simultaneously decomposing user-item, user-feature and item feature matrices

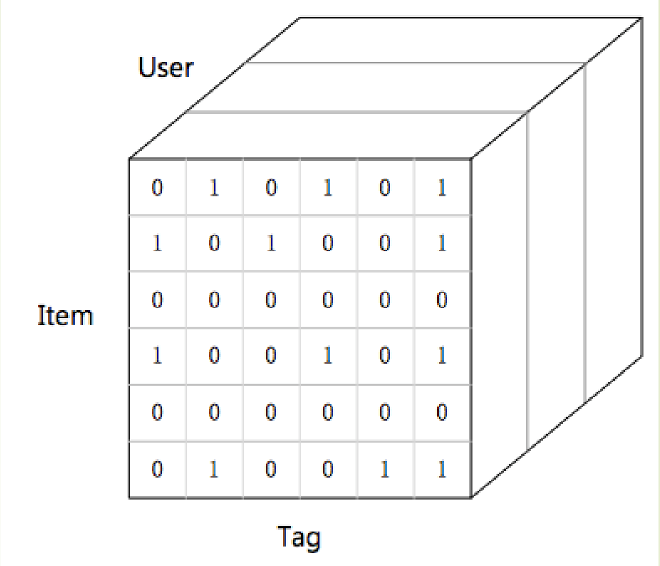


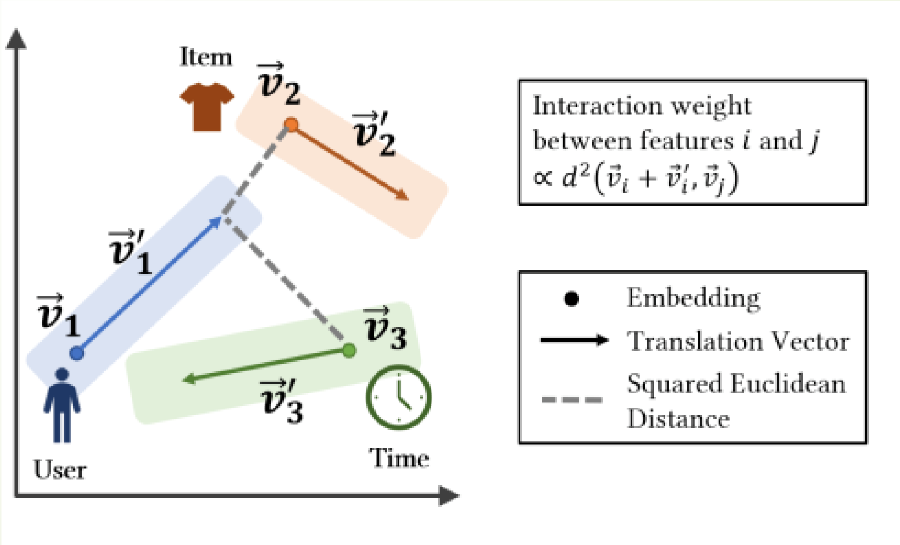
- SVDFeature (Chen et al., 2012) claims that representations of users or items could be influenced by those of their affiliated features. Veloso et al. (2019) incorporate hotel themes into SVD arguing that embedding of a hotel relates to its theme.



- Tensor factorization (Karatzoglou et al., 2010), a generalization of matrix factorization, can flexibly incorporate different types of features by modeling a user-item-feature N-dimensional tensor instead of 2D user-item matrix.

- Factorization machine (FM) (Rendle et al., 2010) model the pairwise interactions between all variables using factorized parameters. Pasricha et al. (2018) propose a sequential recommendation model using FM to fuse use and item flat features, i.e., the transition from user to item is influenced by side information.





- **Latent factor models with network feature (LFMs + NF)**
  - Many studies consider integrating social network. The underlying rationale is that users could share similar interests with their trusted friends. Basically, there are three types of methods: collective MF (CMF), SVDFeature based and regularization based.
  - CMF-based decomposes both user-item interaction matrix and user-user trust matrix. For instance, SoRec (Ma et al., 2008) learn user embedding by simultaneously factorizing user-item and user-trust matrices.

$$
\begin{aligned}
\min_{U,L} \Omega(U,L) &= \sum_{i=1}^{|\mathcal{U}|}\sum_{j=1}^{|\mathcal{L}|} I_{ij}\left(g(F_{ij}) - g(U_i^T L_j)\right)^2 \\
&+ \boxed{\beta \sum_{i=1}^{|\mathcal{U}|} \sum_{f \in \mathcal{F}(i)} Sim(i,f) \|U_i - U_f\|_F^2} \\
&+ \lambda_1 \|U\|_F^2 + \lambda_2 \|L\|_F^2,
\end{aligned}
$$

Social influence

- SVDFeature based methods suppose that the representation of a user will be influenced by his friends. Ma et al. (2011)

$$\mathcal{L}(R,S,U,V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}\left(R_{ij} - g\left(\alpha U_i^T V_j \boxed{+ (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik} U_k^T V_j\Big)\Big)^2}\right.\right. + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2,$$

Social influence

- Regularization based methods constrain the distance of embedding of a user and his friends (Jamali et al., 2010; Ma et al., 2009).

$$\mathcal{L}(R,T,U,V) = \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i} - g(U_u^T V_i))^2 + \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u^T U_u + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i^T V_i + \frac{\lambda_T}{2}\boxed{\sum_{u=1}^{N}\Big(\big(U_u - \sum_{v\in N_u} T_{u,v}U_v\big)^T\big(U_u - \sum_{v\in N_u} T_{u,v}U_v\big)\Big)}$$

Regularization terms based on social friendships

- Summary of LFMs+NF
  - Perform better than basic LFMs with time efficiency.
  - Distrust information can perform as well as trust information (Ma et al., 2009).

- **Latent factor models with feature hierarchy (LFMs + FH)**. This line of research can also be categorized into SVDFeature and regularization. They all incorporate the influence of categories in different layers.

- **Latent factor models with knowledge graph (LFMs + KG)**. Most LFMs+KG generally extract meta paths from KG, then feed them into LFM.
  - Incorporate relationships between item based on path-based similarity into LFM.

$$\min_{U,V,\Theta} \quad \|Y \odot (R - UV^T)\|_F^2 + \lambda_0(\|U\|_F^2 + \|V\|_F^2) + \frac{\lambda_1}{2} \cdot \sum_{i,j} \sum_{l=1}^{L} \boxed{\theta_l S_{ij}^{(l)} \|V_i - V_j\|_2^2 + \lambda_2 \|\Theta\|_2^2,}$$

Meta path based similarity

  - Or through diffusion following meta paths (Yu et al., 2013) or random walk (Catherine et al., 2016), learn to user and item similarity matrices, which are embedded in LFM.

- Summary of LFMs+KG
  - LFM+KG can be regarded as a generalized version of feature-aware approaches. LFMs+KG can be downgraded as other simple versions, LFMs+FF or LFMs+NF.
  - The majority of these methods make use of meta paths (Sun et al., 2011) to extract knowledge from KG. Such way allows easy modeling of user or item-based CF, e.g., user->user->movie can reach movies that similar users have watched.
  - However, these methods heavily relies on the quality and quantity of the handcrafted meta paths, which cannot uncover all sorts of relations.

Discussion of LFMs with side information

- Compared with memory-based methods, LFMs have higher scalability and flexibility to incorporate various types of side information.
- Generally, the more complex side information has been incorporated, the higher quality recommendation can be achieved. For example, LFMs+FH perform better than LFMs+FF, and LFMs+KG outperform other LFMs with side information.
- However, LFMs cannot directly use side information. That's why there are two phases: feature extraction and preference learning. The independence of two phases limits further performance enhancements.

Representation Learning Models with Side Information

- Representation learning models (RLMs) are capable of learning item embedding by item relationship.
- Item2vec is the earlier study (Barkan et al., 2016) based on Skip-gram technique.

$$\mathcal{L} = \sum_{s \in \mathcal{S}} \sum_{p_i \in s} \sum_{-c \le j \le c, j \ne 0} \log \mathbb{P}(p_{i+j} | p_i)$$

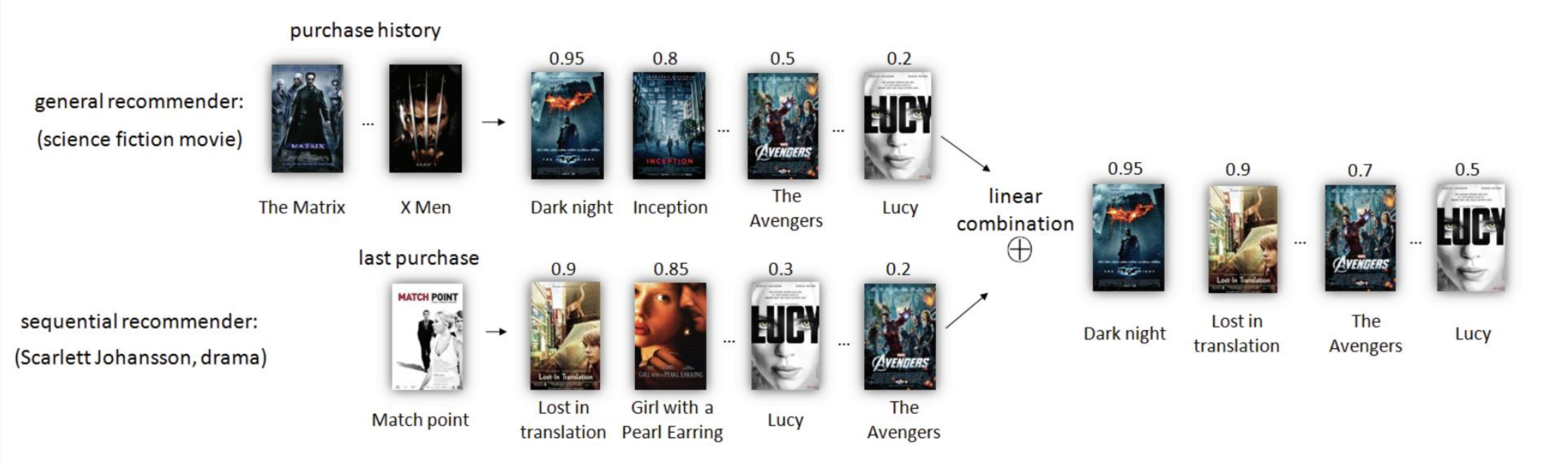
$$\mathbb{P}(p_{i+j} | p_i) = \frac{\exp(\mathbf{v}_{p_i}^\top \mathbf{v}'_{p_{i+j}})}{\sum_{p=1}^{P} \exp(\mathbf{v}_{p_i}^\top \mathbf{v}'_p)}$$

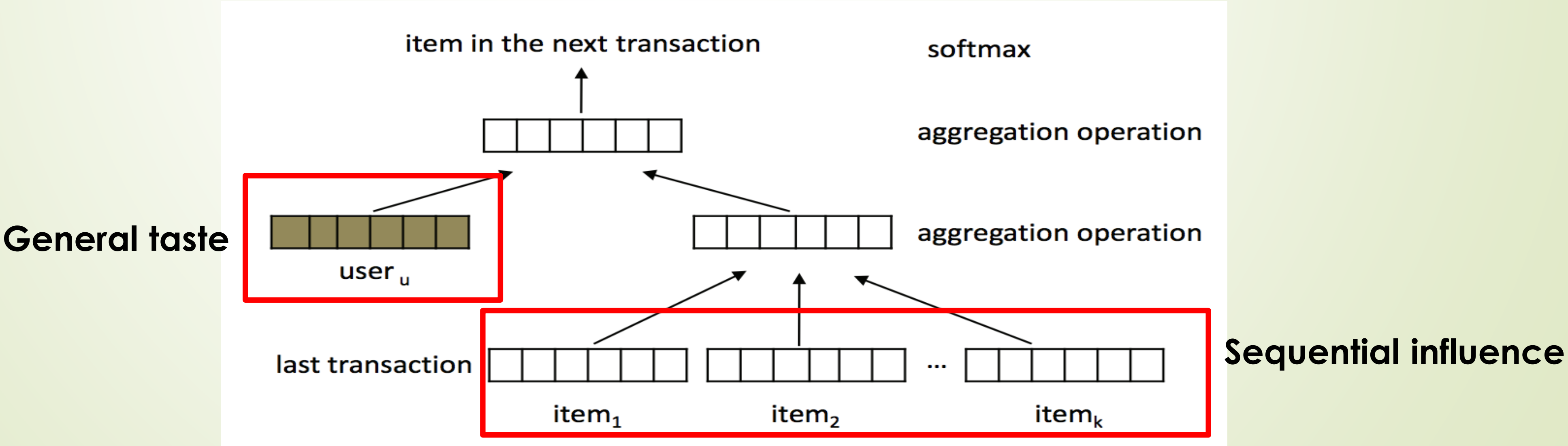
- Next, there are some studies modeling in user preference with item2vec. For example, Grbovic et al. (2015) develop a user2vec model simultaneously learns vector representations of products and users by considering the user as a "global context".

$$\mathcal{L} = \sum_{s \in \mathcal{S}} \Bigg(\boxed{\sum_{u_n \in s} \log \mathbb{P}(u_n | p_{n1} : p_{nU_n})} + \sum_{p_{ni} \in u_n} \log \mathbb{P}(p_{ni} | p_{n,i-c} : p_{n,i+c}, u_n) \Bigg)$$

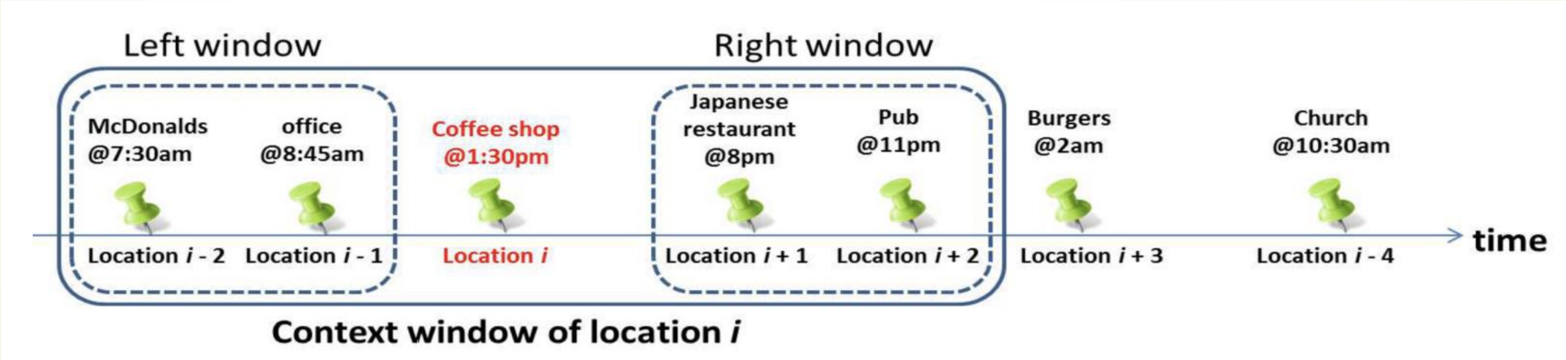
Incorporate user embedding

- Then, Wang et al. (2015) propose a novel hierarchical representation model (HRM) for next basket recommendation. HRM performs better than traditional markov chain based models.

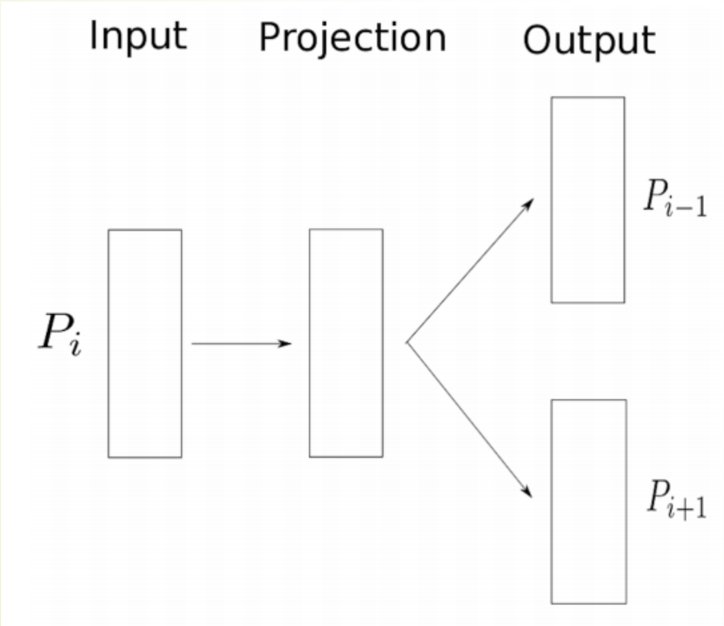


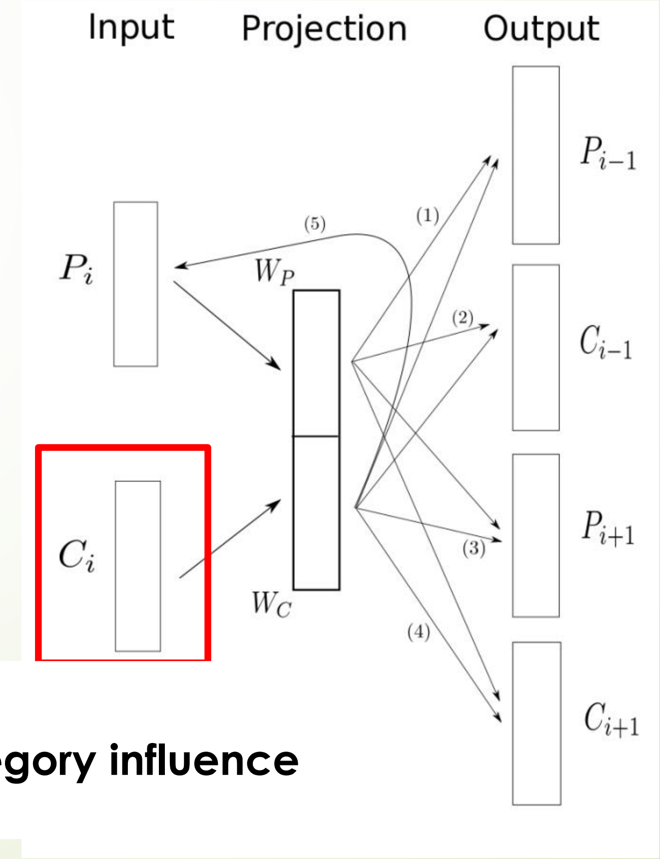


- By taking advantage of RLMs, some researchers integrate side information into RL models.
  - Liu et al. (2016) propose a temporal-aware model (CWARP-T) by leveraging skip-gram model. It can jointly learn the latent representation for a user and location to capture users' preference and location context.



- **Meta2Prod** utilizes item categories to assist in regularizing the learning of item embedding (Vasile et al., 2016). It assumes that categories have significant effect on what user would buy, e.g., it is more likely that the next visited product will belong to the same category, or it is more likely that the next category is or one of the related categories.





Category influence

- **Discussion of RLMs with side information**
  - Though, there are fewer works on RLMs, they provide us a different point of view, that is, learn the item and user embedding by considering item local relations, while LFMs aim to learn them at a global level.
  - The objective function of RLMs is actually a softmax layer. Thus, RLMs can be considered as the transition from shallow to deep neural network.

$$P(v_k|u_p,\Theta) = \frac{exp(\mathbf{u}_p^T \mathbf{v}'_k)}{\sum_{v_g \in \mathcal{I}} exp(\mathbf{u}_p^T \mathbf{v}'_g)}$$

  - The fundamental RLMs (item2vec) do not well consider personalization. Recent studies extend it by averaging embeddings of items a user has interacted with or be treated as a global "context".
  - Most existing RLMs with side information focus on flat features. Other types pf data (e.g., feature hierarchy) can further be incorporated into RLMs.

Deep Learning with Side Information

- Basic deep learning models
- Deep learning models with flat features (DLMs + FF)
- Deep learning models with network features (DLMs + NF)
- Deep learning models with feature hierarchy (DLMs + FH)
- Deep learning models with knowledge graph (DLMs + KG)

Basic Deep Learning Models

- **Auto-encoder based models**
- Auto-encoder is the simplest neural network with 3-layers which projects (encodes) high-dimensional input layer into a low-dimensional hidden layer and finally decodes the hidden layer to output layer. The early one is AutoRec (Sedhain et al., 2015).

$$\min_{\theta} \sum_{\mathbf{r} \in \mathbf{S}} ||\mathbf{r} - h(\mathbf{r}; \theta)||_2^2,$$

$$h(\mathbf{r}; \theta) = f\left(\mathbf{W} \cdot g(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b}\right)$$

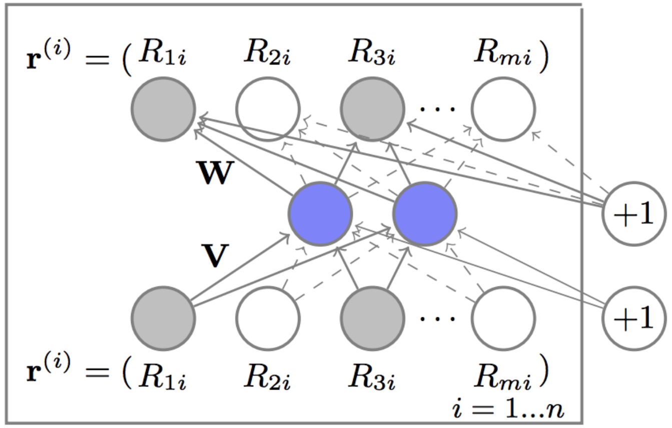
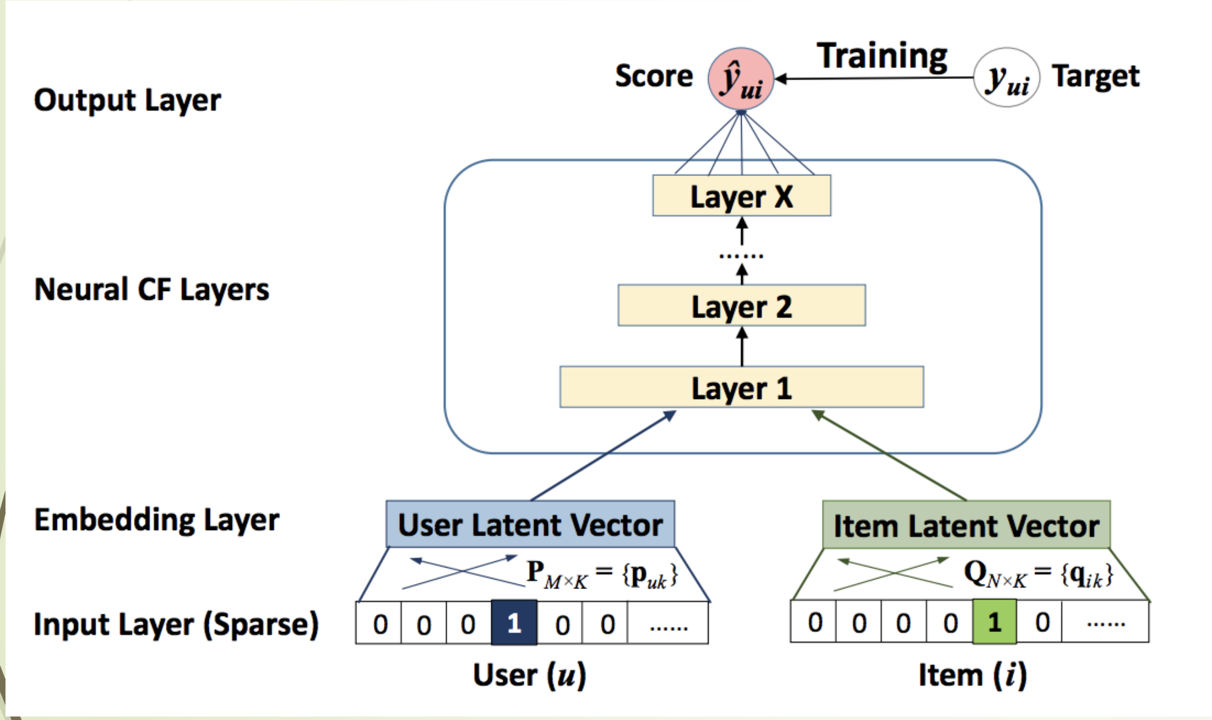
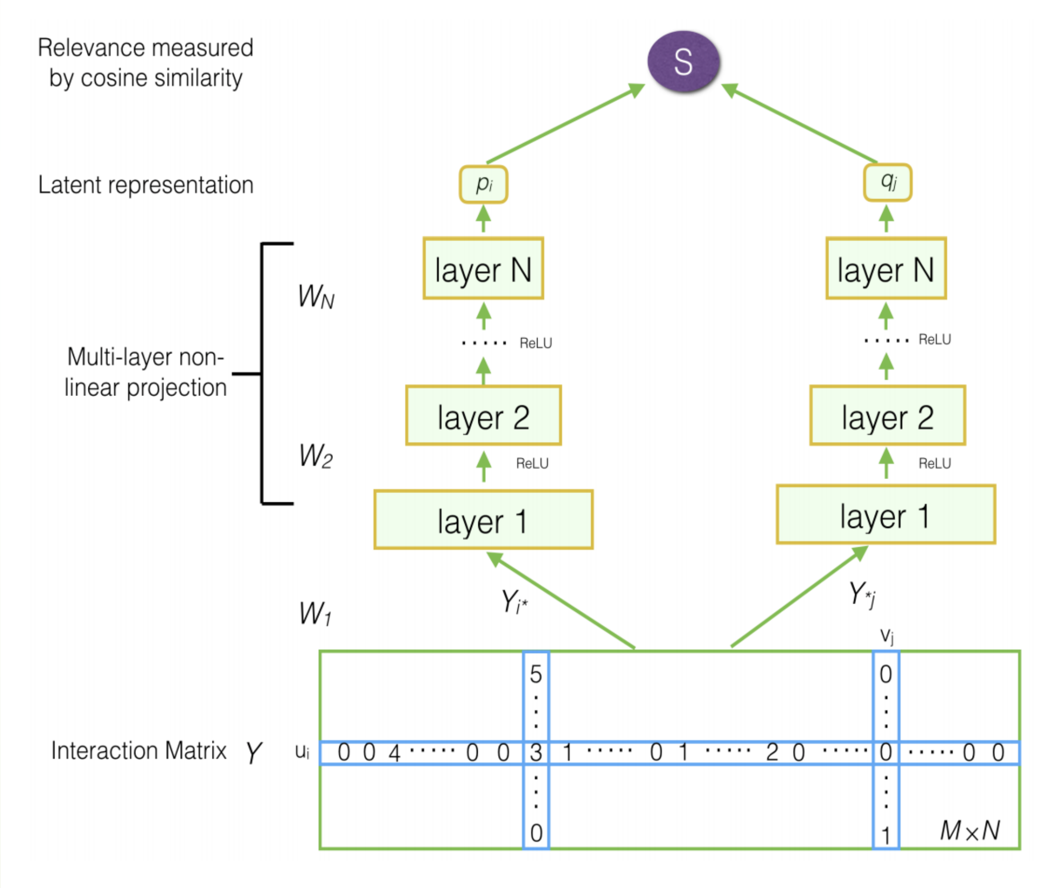


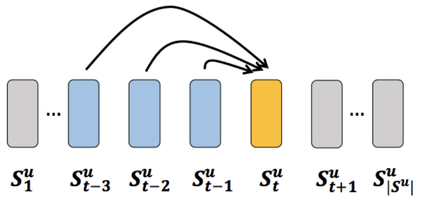
Figure 1: Item-based AutoRec model. We use plate notation to indicate that there are n copies of the neural network (one for each item), where $\mathbf{W}$ and $\mathbf{V}$ are tied across all copies.

- **Multi-layer perceptron based methods (MLP**) contains one or more hidden layers with arbitrary activation functions providing levels of abstractions.
- He et al. (2016) propose Neural Collaborative Filtering (NCF) framework that fuses generalized matrix factorization with MLP.
- Xue et al. (2017) design a Deep Matrix Factorization (DMF) that exploits multi-layer non-linear projection to learn user and item representations. In our architecture, we have two multi-layer networks to transform the representations of user and item respectively

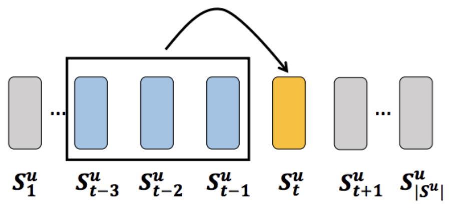




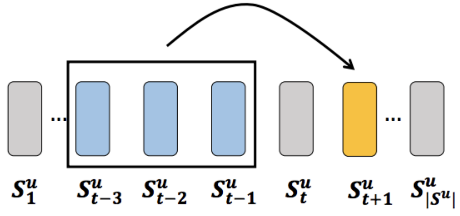
- **CNN based methods.** CNN can be treated as a variant of MLP. It takes a fixed size of input/output, and its hidden layers typically consist of convolutional layers, pooling layers, and fully connected layers. By regarding the input data as “image”, CNN can be utilized to help capture local features.
- Tang et al. (2018) propose *Caser* for next item recommendation. It embeds a sequence of recent items into latent space as an “image”.



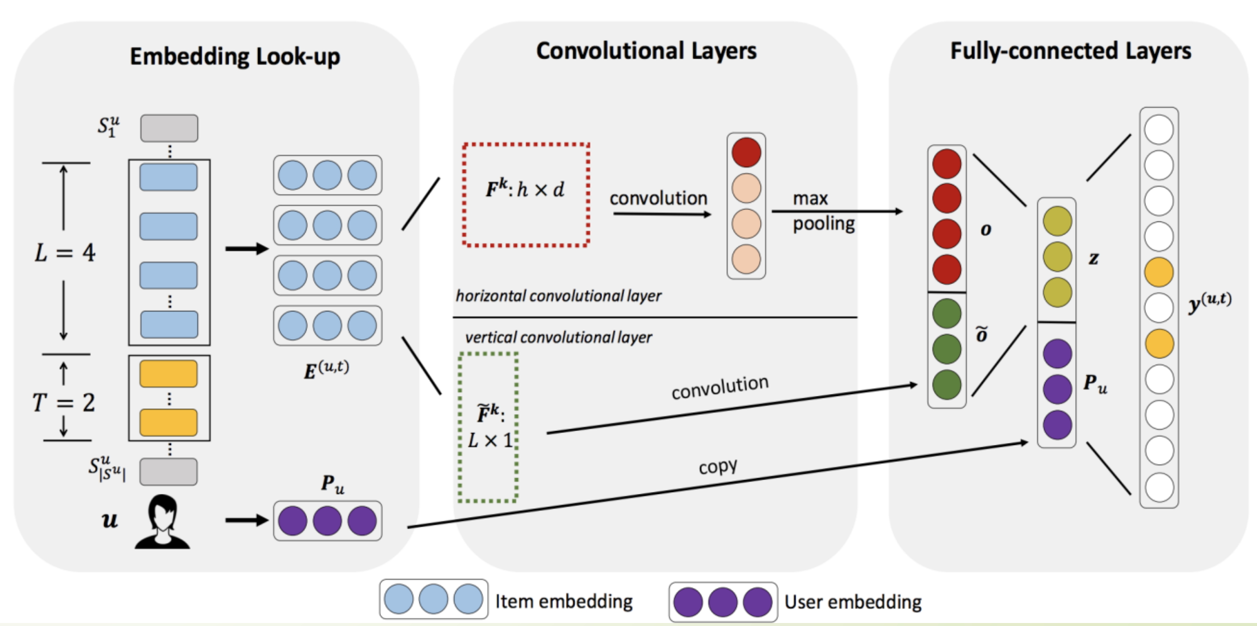
(a) point-level



(b) union-level, no skip

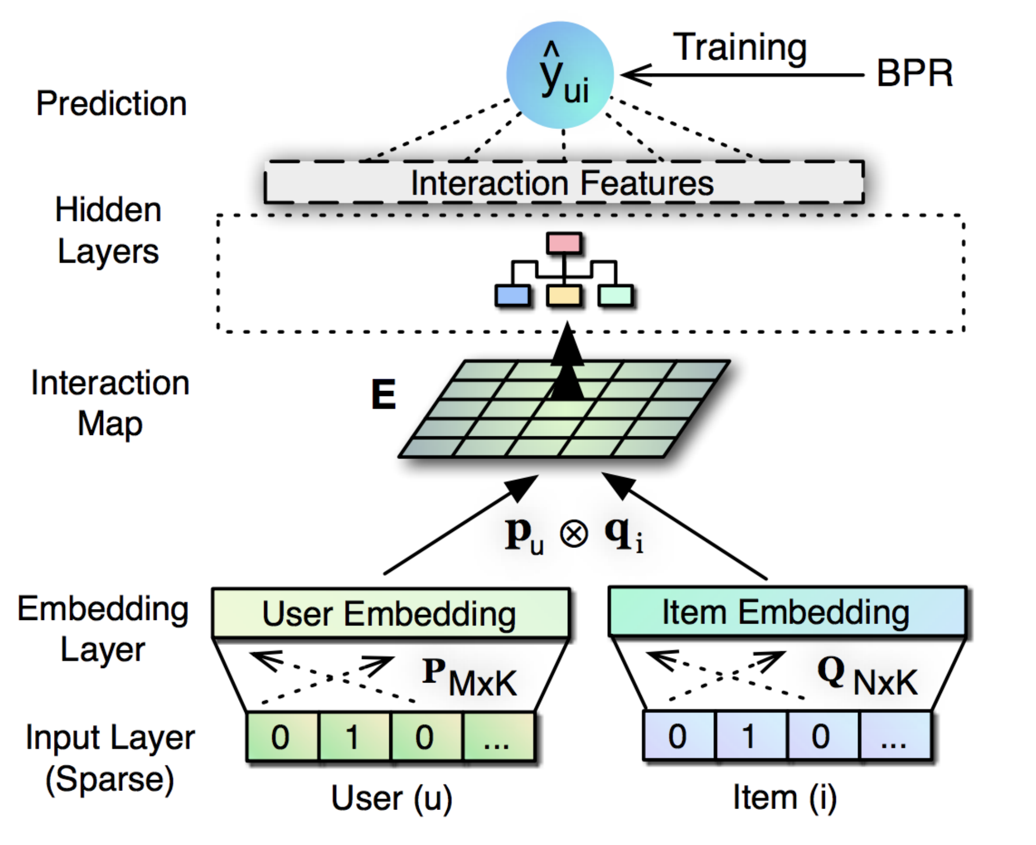


(c) union-level, skip once

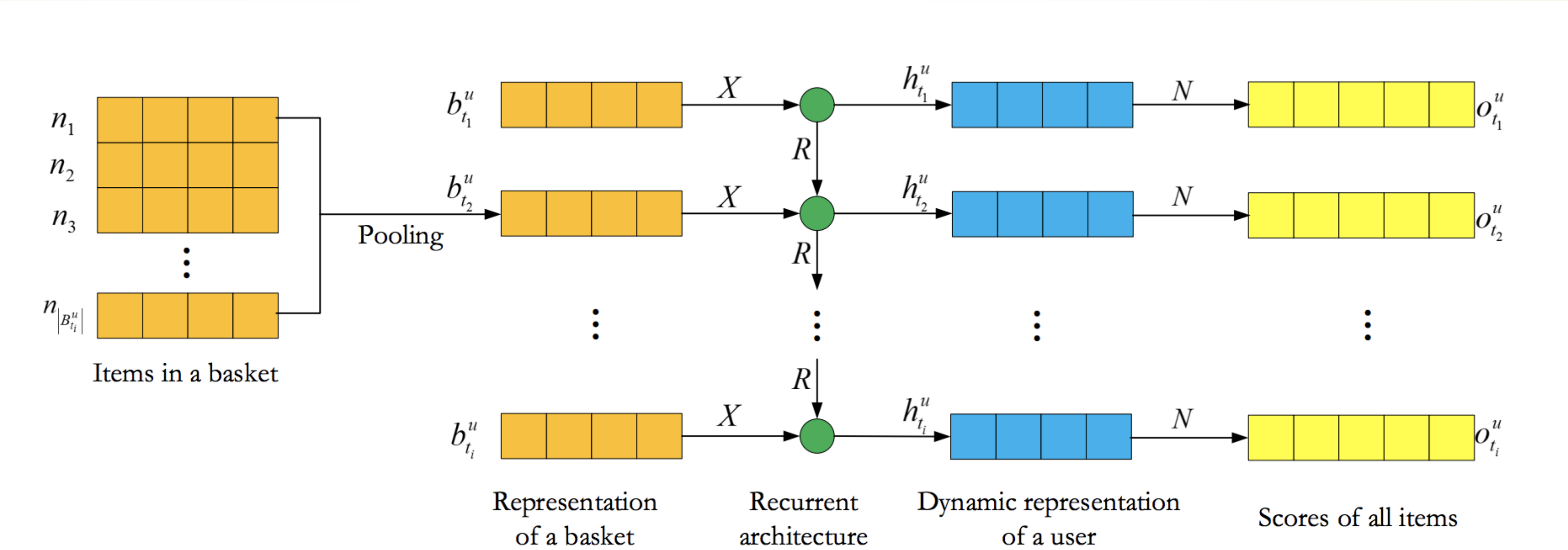


- He et al. (2018) propose an outer product (an interaction map) to explicitly model the pairwise correlations between the user and item embeddings, and then use CNN to learn high-order correlations in different levels.
  - MF essentially assumes that the embedding dimensions (i.e., dimensions of the embedding space) are independent with each other and contribute equally for the prediction of all data points.

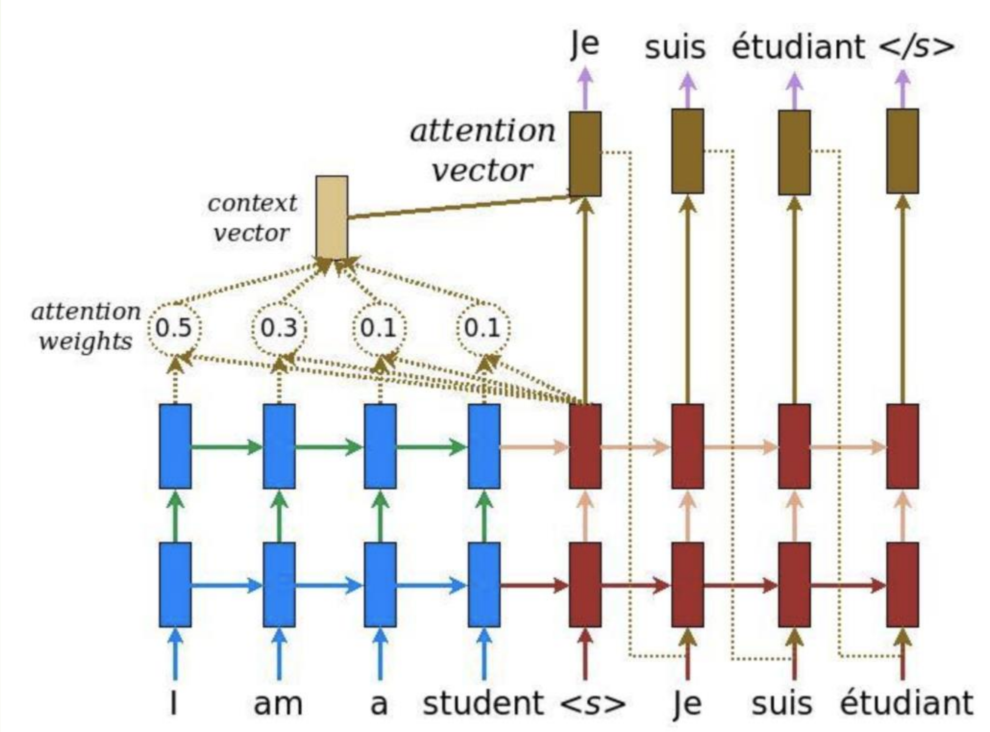
$$\mathbf{E} = \mathbf{p}_u \otimes \mathbf{q}_i = \mathbf{p}_u \mathbf{q}_i^T,$$



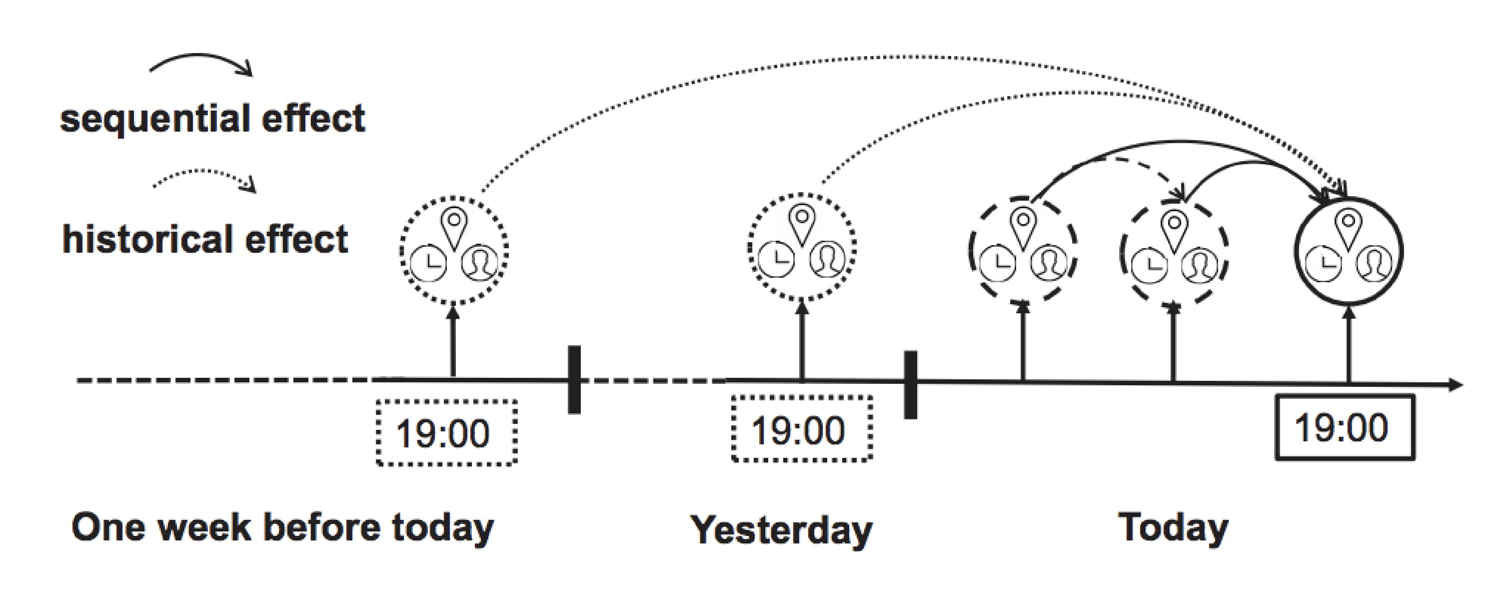
- **RNN based methods.** Recurrent neural network is able to memorize historical information and fine patterns across time.
- Yu et al. (2016) propose a dynamic recurrent basket model that learns a dynamic representation of a user but also captures global sequential features among baskets.



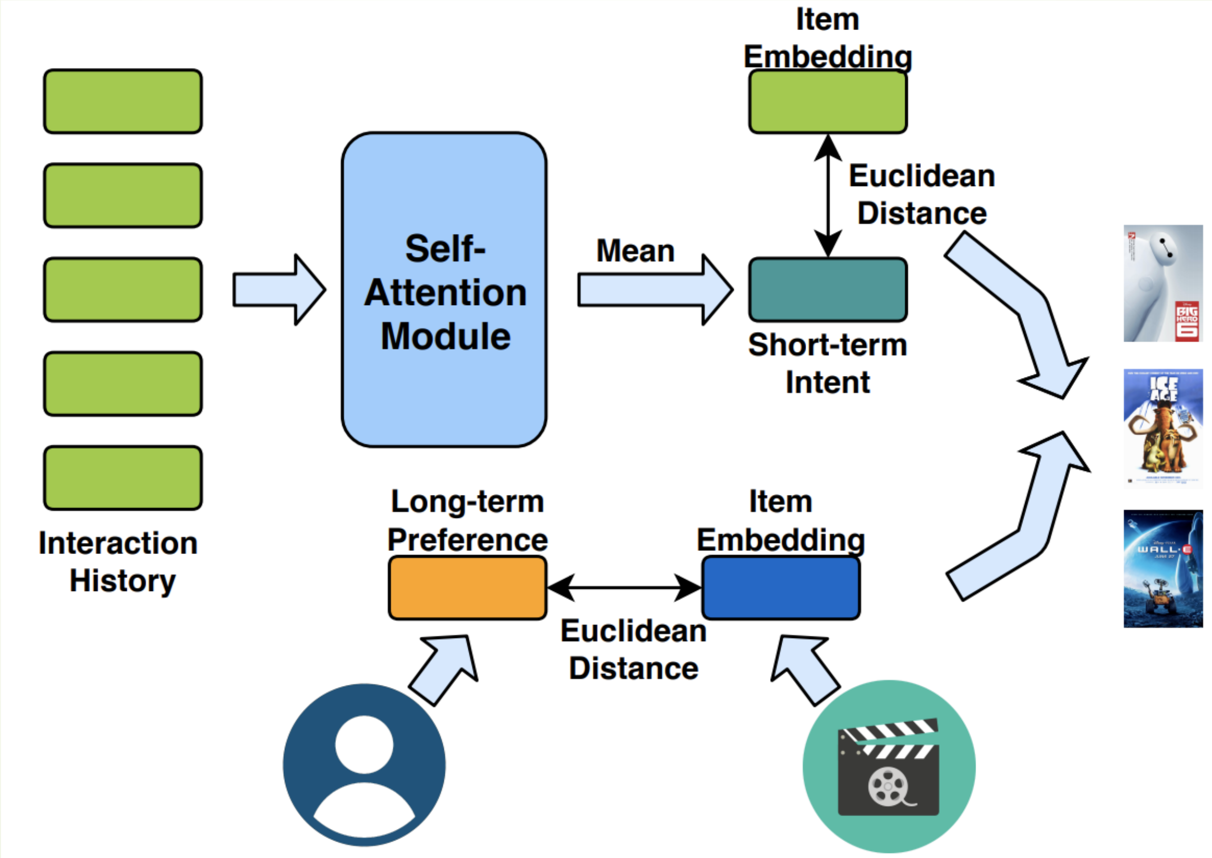
- **Attention based methods**. The idea of attention mechanism comes from computer vision, that is, human's visual attention focus on certain part of a image.
  - It can cope with noisy data to identify relevant parts of the input for modeling user-item interactions.
  - Standard vanilla attention mechanism learns the attention score for the input data by transforming the representations of input data via fully connected layers, and then use softmax layer to normalize the score.
  - It cooperates with RNN or LSTM to better memorize import long dependencies, or CNN to help concentrate on import parts of inputs.



- Feng et al. (2018) develop a DeepMove model using GRU to capture influences of both short-term trajectory and long-term historical trajectories.



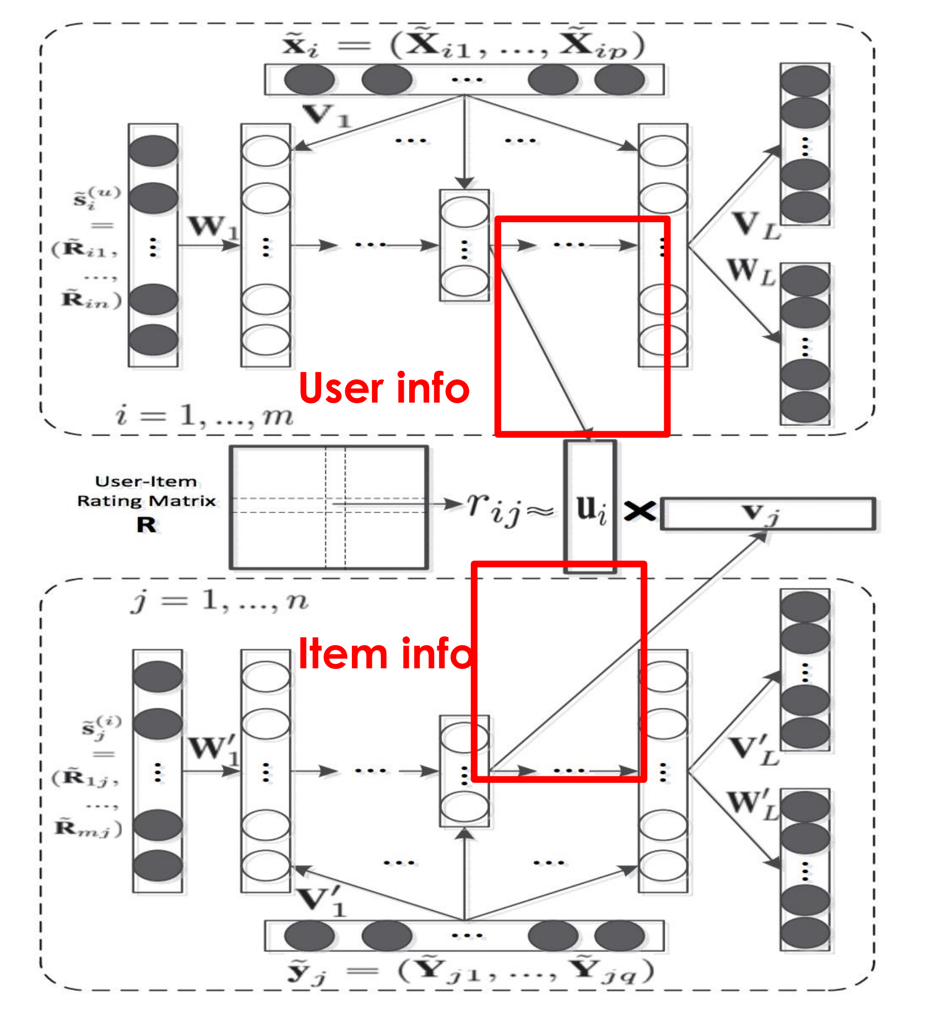
- Zhang et al. (2018) propose a sequence-aware model by considering both short-term and long-term user interests.



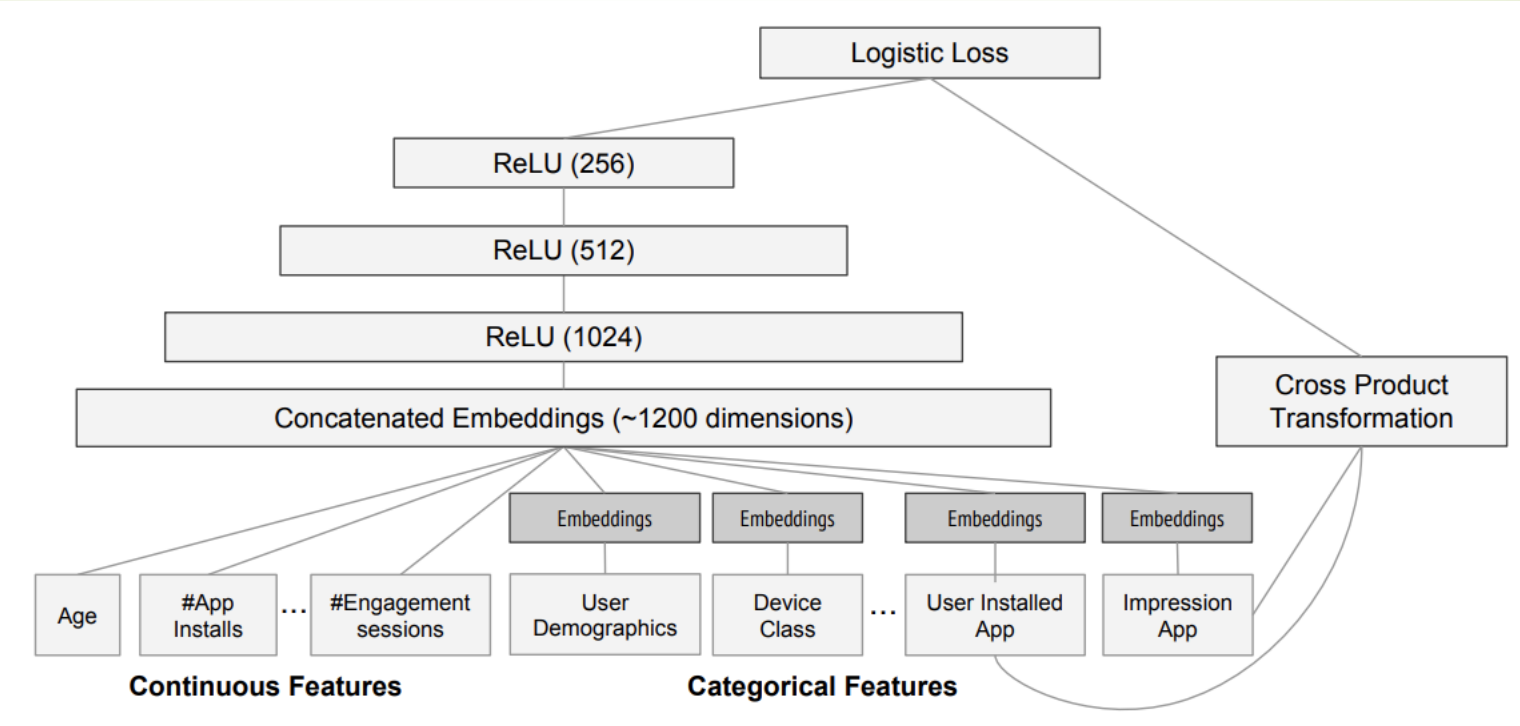
- **Summary of basic DLMs**
  - Auto-Encoder, as the simplest neural network, can be extended to fuse both structural and non-structural side information by learning the contextual representation of items from flat feature.
  - MLPs can efficiently extract high-level user and item representations. Moreover, it can easily fuse structural side information by concatenating flat features with user or items as input data.
  - CNN is always exploited to capture spatial patterns, i.e., local relations among features in the “image” with fixed input and output lengths. Hence, it’s more capable of modeling non-structural information data like text or image.
  - RNN is able to learn long-distance dependencies with arbitrary input and output lengths, thus it’s more suitable for sequential recommendation, or explainable recommendation to generate text (e.g., review or tips)
  - Attention models can distinguish the different importance of the input data.

Deep Learning Models with flat features

- **Auto-Encoder+FF**. Dong et al. (2017) develop a hybrid recommender (HSD) that make use of both rating and side information. It uses two additional stacked denoising auto-encoders with side information as input, which are trained with MF.



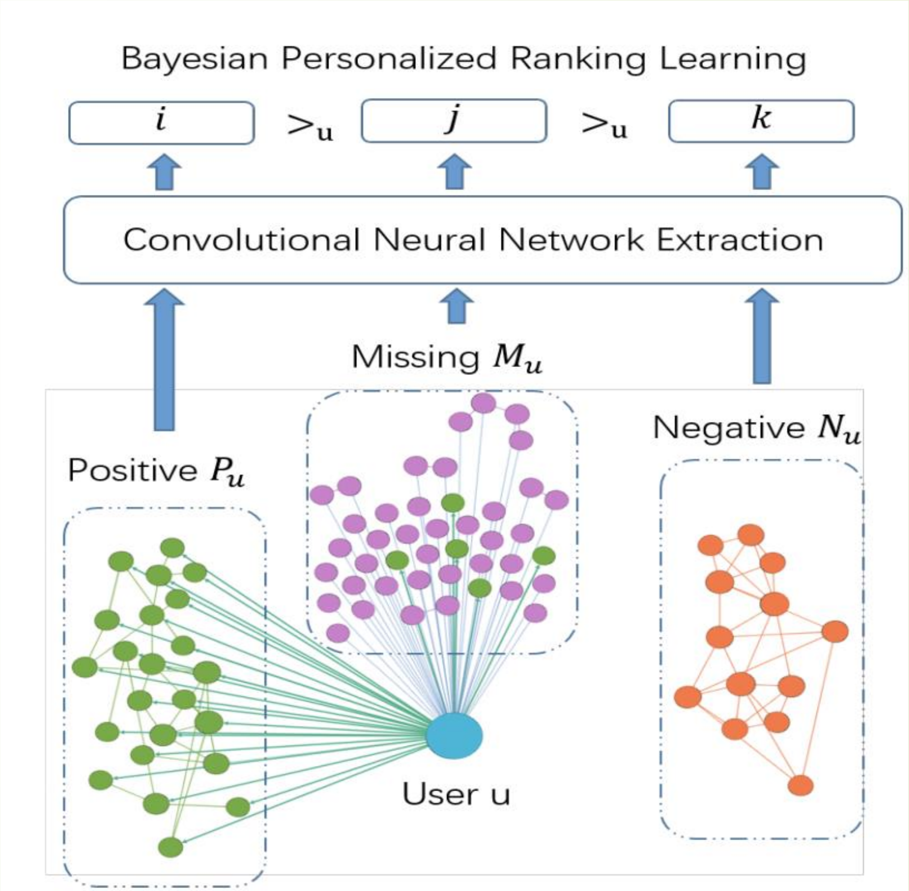
- **MLP based methods.** Cheng et al. (2016) jointly train wide linear regression models and deep neural networks. The categorical features are converted into a low-dimensional embedding, and then fed into the hidden layers of the deep neural network.



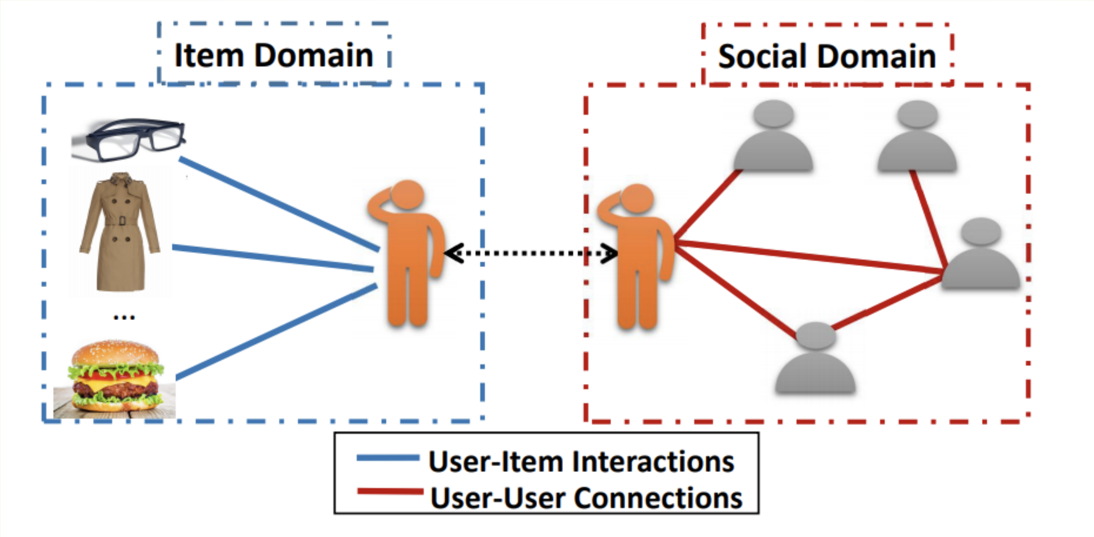
- Summary of DLMs+FF. The flat features is generally incorporated into various DLMs in three different ways:
  - Pre-filtering is the simplest way. For instance, Okura et al. (2017) propose a embedding-based new recommender uses categories to pre-select positive and negative articles.
  - Concatenation is most straightforward way. Wide&Deep (Cheng et al., 2016), DNN (Covington et al., 2017), and CDL-image (Lei et al., 2016) concatenate all features vectors together fed into neural network.
  - Projection is the most fine-grained way. HDS (Dong et al., 2017) and NPR (Niu et al., 2018) employ neural network to learn user or item representations under different context, i.e., contextual representations.

Deep Learning Models with Network Features

- **DLMs+NF**. The user embedding are influenced by their friends.
- Ding et al. (2017) design a CNN based method that extract latent deep structural feature representations by regarding the input network data as an ''image''.



- Fan et al. (2018) propose a rating prediction model, which first uses Node2Vec to learn the user embedding in social network. The user embedding are fed into matrix factorization for predicting rating.

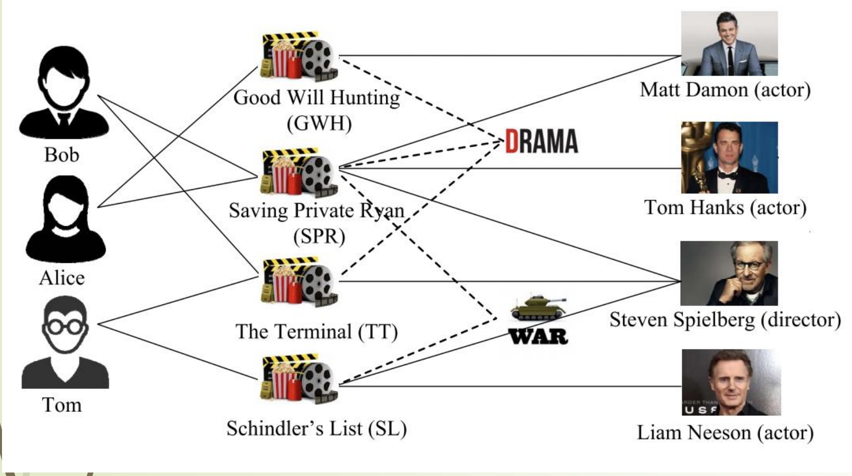


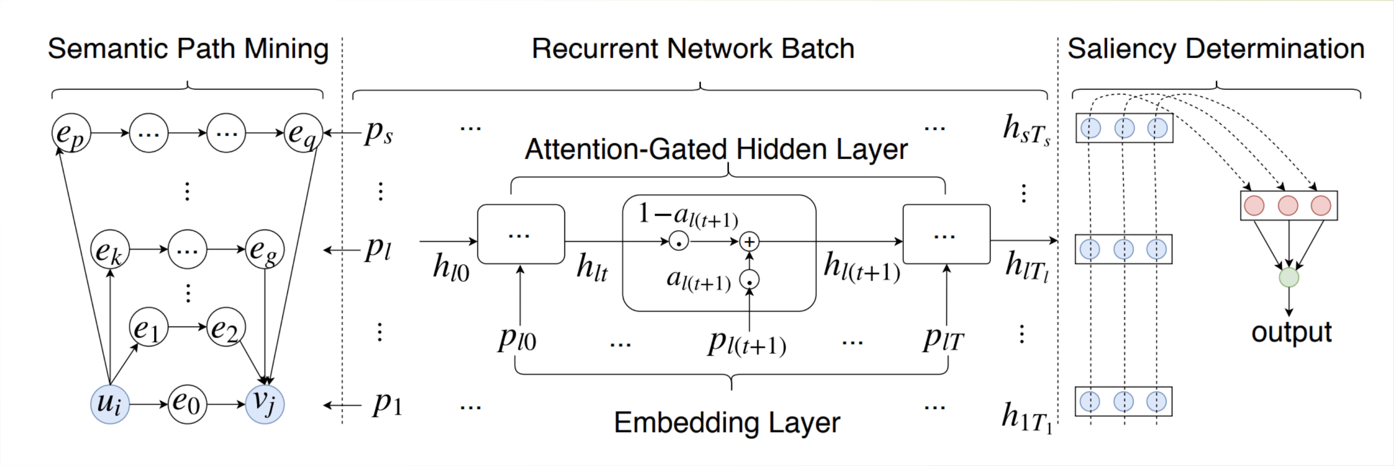
- **Summary of DLMs+NF**
  - The existing experiments show that DLMs+NF>LFMs+NF.
  - NF can be considered as an image, so graph related DLMs can be applied.
  - Distrust information is worthy of exploring in DLMs as well.

Deep Learning Models with Knowledge Graph (DLMs+KG)

- According to the way that KG is exploited, there are three types of DLMs+KG methods: graph embedding based method and path embedding based methods.
- **Graph embedding based methods**.
  - Many approaches use conventional graph embedding methods, such as TransE (Bordes et al., 2013), TransR (Lin et al., 2015), TransH (Yang et al., 2015) and TransD (Ji et al., 2015).

- **Path embedding based methods**. They extract connected paths with different semantics between user-item pairs, and then encode these paths via DLMs.
- Sun et al. (2018) propose a recurrent knowledge graph embedding method (RKGE), which first extract paths between users and items. Then RNN is used to learn the path influences on charactering user-item interactions.





- **Summary of DLMs +KG**
  - DLMs+KG approaches perform much better than LFMs+KG in terms of accuracy. However, the high computational cost limits the scalability of DLMs+KG.
  - Regardless of KG usage types, most of existing methods rely on conventional KG embedding methods like TransE/TransR/TransH/TransD. In particular, they learn KG embedding based on triple <*h, r, t*> where *h* and *t* are *head* and *tail*.
  - In order to utilize the heterogeneity of KG, distinguish entity types and relation types can deliver more accurate results, e.g., KPRN (Wang et al., 2018) outperforms RKGE (Sun et al., 2018), and also provide additional reasoning information regarding the recommendation.

Summary and Future Directions

- **Summary**. Give a comprehensive review of recommendation model with side information from new perspectives.
- **Future directions**.
  - How to further improve deep learning based recommendation with side information in complex structure?
    - Intrinsic complexity of structured side information
    - The difficulty in adapting deep learning methods for incorporating side information.
  - How to obtain high-quality side information to improve recommendation?
    - Use crowdsourcing techniques.

Our Survey Paper

- *Recommendatin with side information: a survey and new perspectives*, will be published in *Electronic Commerce Research and Applications.*

Thank you