# TEM: Tree-enhanced Embedding Model for Explainable Recommendation

XiangWang1, XiangnanHe1, FuliFeng1, LiqiangNie2, Tat-SengChua1

1 National University of Singapore, 2 Shandong University

In WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.

https://doi.org/10.1145/3178876.3186066

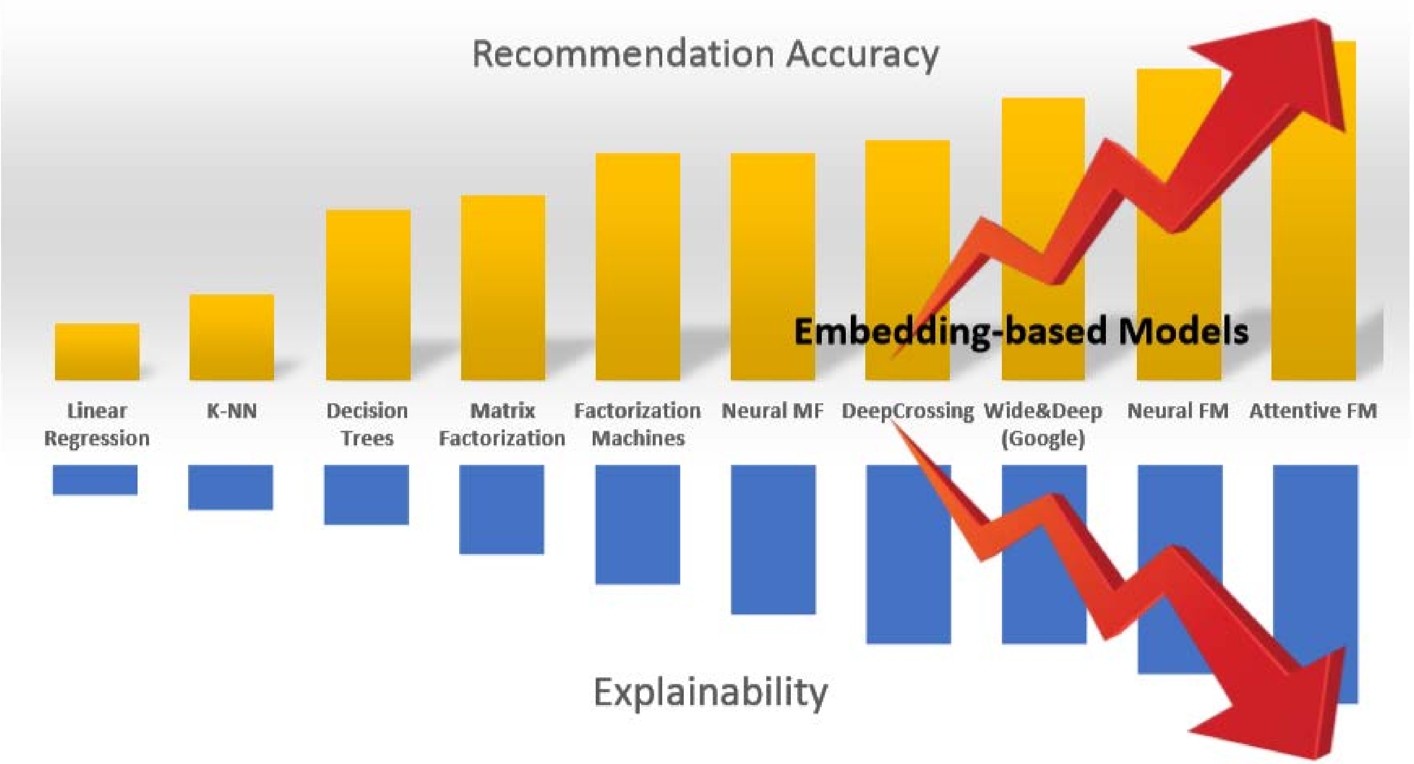
# OUTLINE

- Introduction
- Tree-enhanced Embedding Model
- Experimental Results
- Conclusion

# OUTLINE

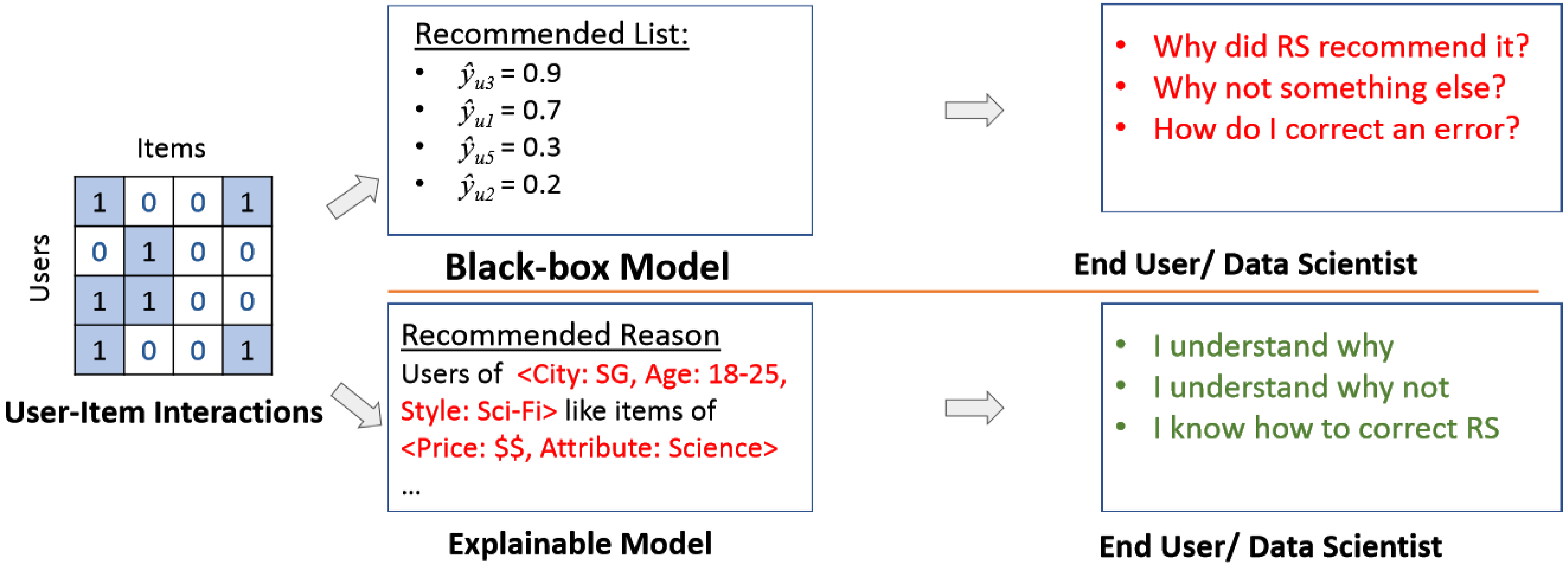
- Introduction
- Tree-enhanced Embedding Model
- Experimental Results
- Conclusion

# Trade-off of Accuracy & Explainability




- **Our Goal:**
  - **Accurate:** achieve the same level or comparable performance as embedding-based methods
  - **Explainable:** be transparent in generating a recommendation & can identify the key rules for a prediction

# Explainable Recommendation



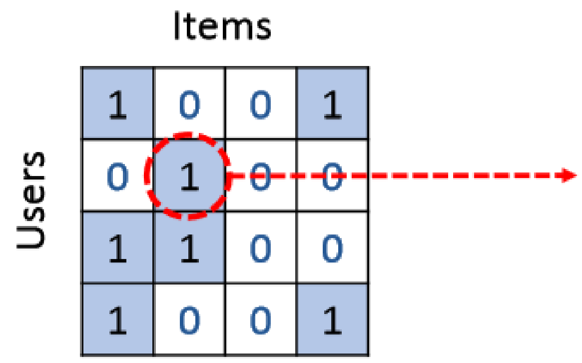

**Transparency, Trust, Explainability, Scrutability**

# OUTLINE

- Introduction
- Tree-enhanced Embedding Model
- Experimental Results
- Conclusion

# Embedding-based Models

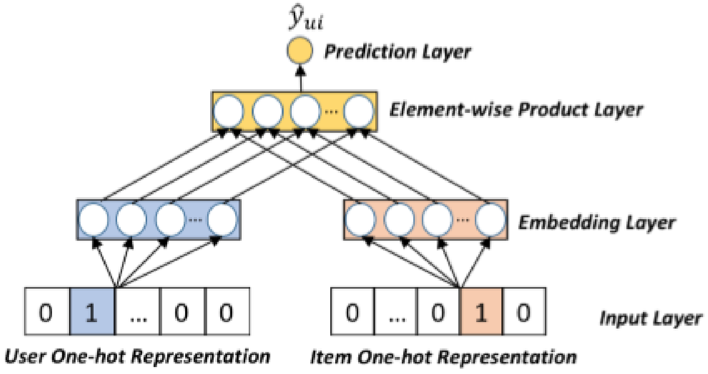
- Learn latent factors for each feature (IDs & side Info)




**User-Item Interactions**

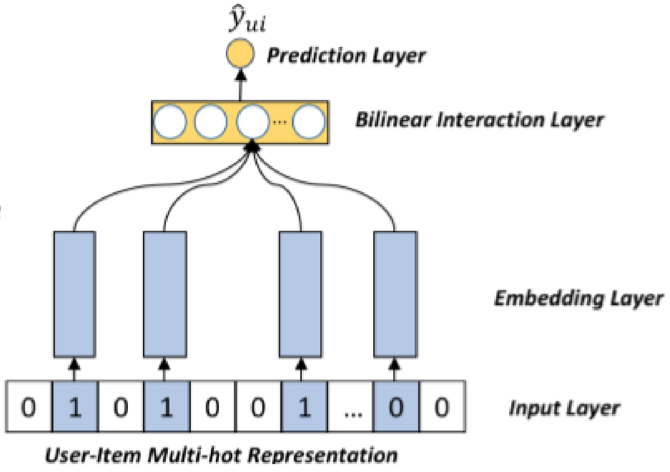
**Matrix Factorization (MF)**
**Input:** user ID, item ID
**Interaction:** Inner Product

$$\hat{y}_{MF}(u, i) = b_0 + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i$$




**Factorization Machine (FM)**
**Input:** user ID, item ID, side features ID
**Interaction:** Element-wise Product

$$\hat{y}_{FM}(\mathbf{x}) = w_0 + \sum_{t=1}^{n} w_t x_t + \sum_{t=1}^{n} \sum_{j=t+1}^{n} \mathbf{v}_t^\top \mathbf{v}_j \cdot x_t x_j$$




**Neural Network Methods**
NCF, Deep Crossing, Wide&Deep, DIN, NFM

# Cross Features

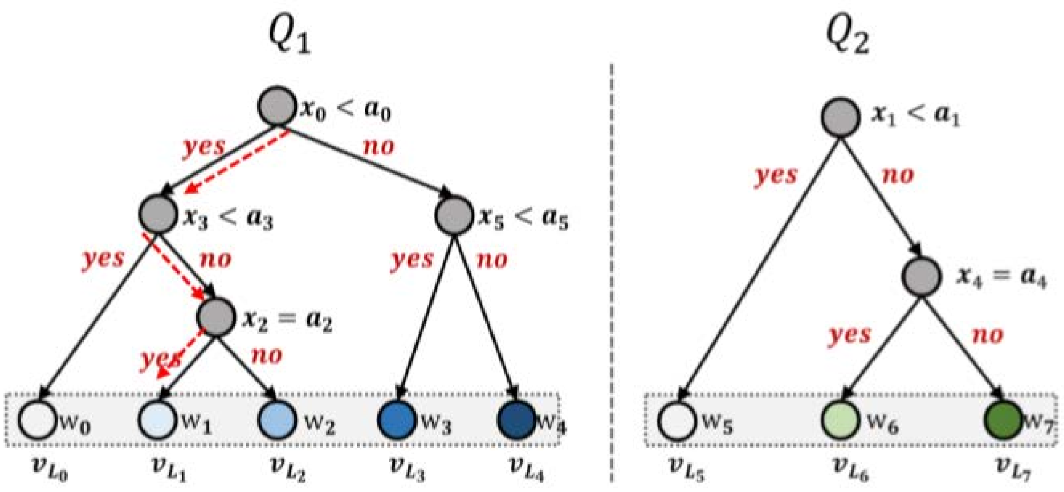
- Cross Feature: combinatorial feature that crosses (or multiplies) multiple individual input features.
- Why?
  - Higher-order feature interactions: e.g., [Age * Occupation *Gender]
  - Explicit decision rules

- For example
  - Users of <Gender=female & Age=20-25 & Income Level=$8,000> tend to adopt items of <Color=Pink & Product=Apple>

# Tree-based Methods

**Decision Tree (DT):**

- Each node splits a feature variable into two decision edges based on a value.
- A path from the root to a leaf -> a decision rule (like a cross feature).
- The leaf node -> the prediction value.




leaf node $v_{L_2}$ represents $[x_0 < a_0]\&[x_3 \geq a_3]\&[x_2 \neq a_2]$

**Forest (ensemble of trees)**

- Since a single tree may not be expressive enough, a typical way is to build a forest, i.e., an ensemble of additive trees

\# of trees

$$\hat{y}_{GBDT}(\mathbf{x}) = \sum_{s=1}^{S} \hat{y}_{DT_s}(\mathbf{x}),$$

Prediction of the *s*-th tree

# Tree-based vs. Embedding-based Model

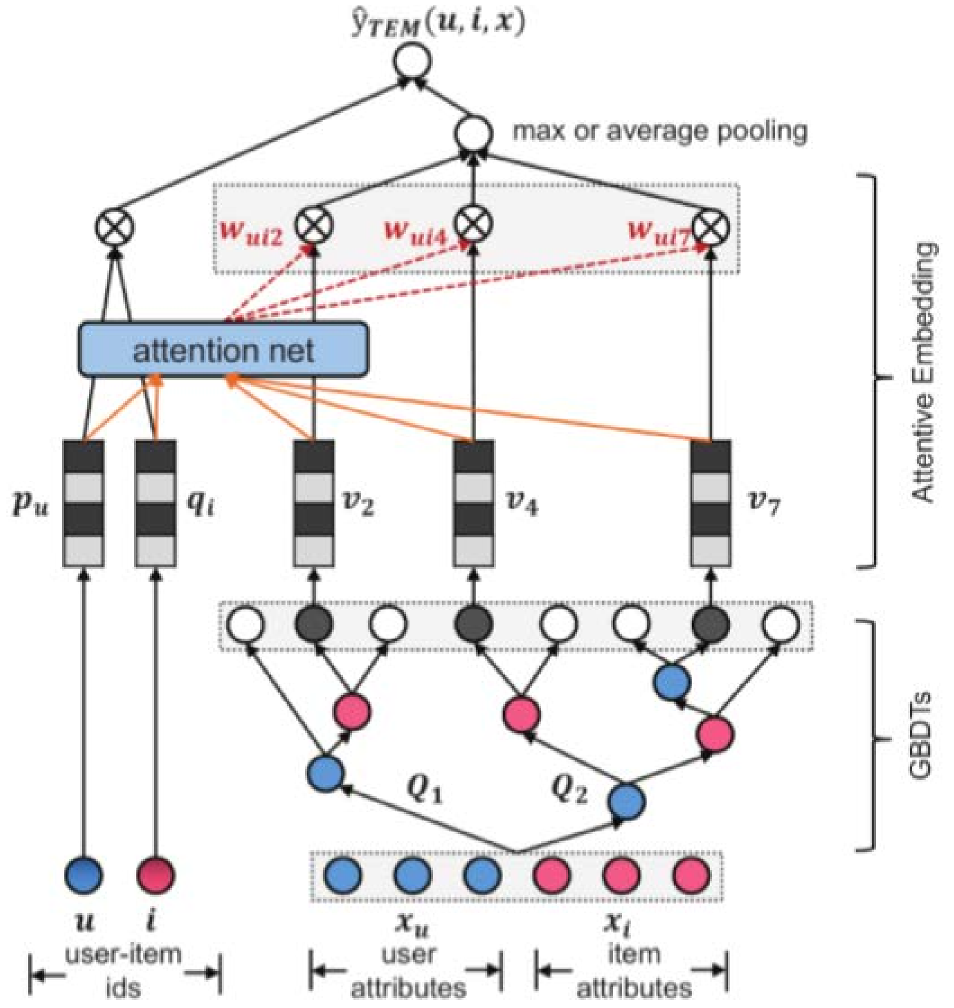
| Tree-based Model (e.g., GBDT) | Embedding-based Model (e.g., DNN, FM) |
|---|---|
| + Strong at continuous features | + Strong at categorical features |
| + Explainable | - Blackbox |
| + Low serving cost | - High serving cost |
| - Weak generalization ability to unseen feature combinations. | + Strong generalization ability to unseen feature combinations. |

**Why not combining the strengths of the two types of models?**

# Tree-enhanced Embedding Model

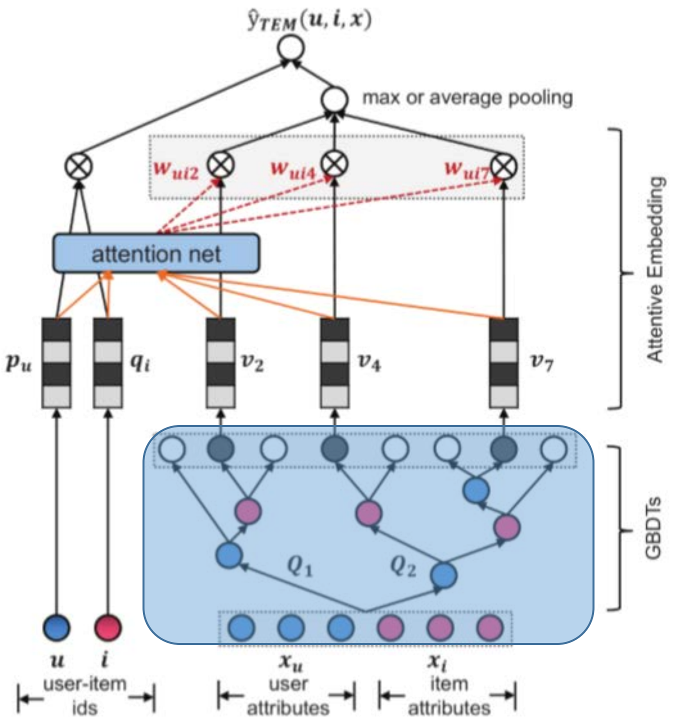
**Concrete Reasons:** Explicitly discover effective cross features from rich side information of users & items

**Explicit Decision Process:** Estimate user-item matching score in an explainable way

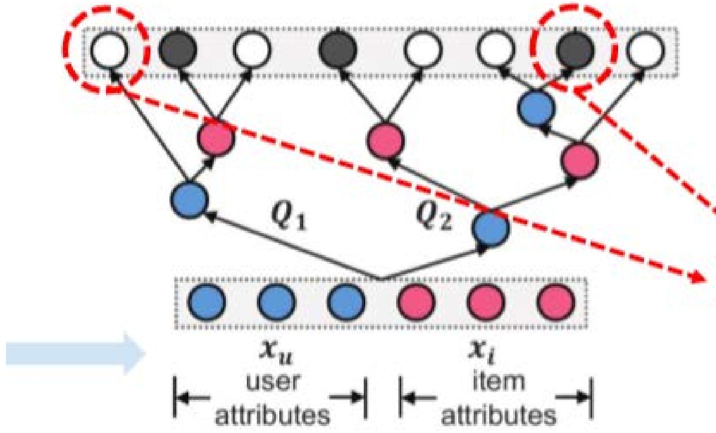


# Constructing Cross Features

- Traditional Solution: manually cross all values of feature variables
- Our Solution: GBDT -> automatically identify useful cross features
- We bulid GBDT on user attributes and item attributes.




$$\mathbf{q} = GBDT(\mathbf{x}|Q) = [Q_1(\mathbf{x}), \cdots, Q_S(\mathbf{x})]$$



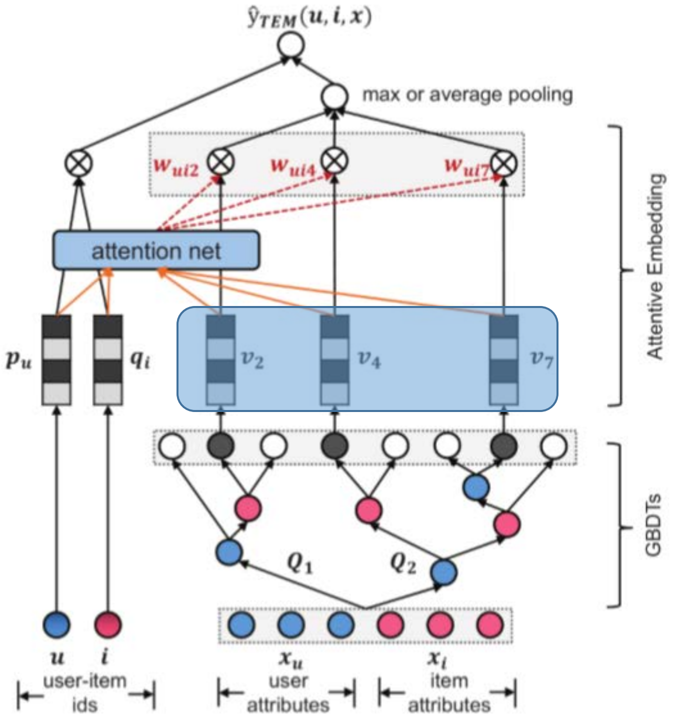

| | | |
|---|---|---|
| $x_0 \leftarrow$ *Age* <br> $a_0 \leftarrow 18$ | $x_1 \leftarrow$ *Expert Level* <br> $a_1 \leftarrow 4$ | $x_2 \leftarrow$ *Restaurant Tag* <br> $a_2 \leftarrow$ *French* |
| $x_3 \leftarrow$ *Country* <br> $a_3 \leftarrow$ *France* | $x_4 \leftarrow$ *Traveler Style* <br> $a_4 \leftarrow$ *Luxury Traveler* | $x_5 \leftarrow$ *Price* <br> $a_5 \leftarrow$ *$$$$* |

(1) $v_{L_1}$: *[Age< 18] & [Country≠France] & [Restaurant Tag= French].*
(2) $v_{L_7}$: *[Expert Level≥ 4] & [Traveler Style≠Luxury Traveler].*

**Explicit Cross Features with easy-to-comprehend semantics!**

# Cross Features Embedding

- Primary Consideration: seamlessly integrate cross features with embedding-based CF
- Our Solution: embed them into user-item latent space




$$\mathbf{q} = GBDT(\mathbf{x}|\mathcal{Q}) = [Q_1(\mathbf{x}), \cdots, Q_S(\mathbf{x})].$$

Multi-hot encoding of cross-feature ID

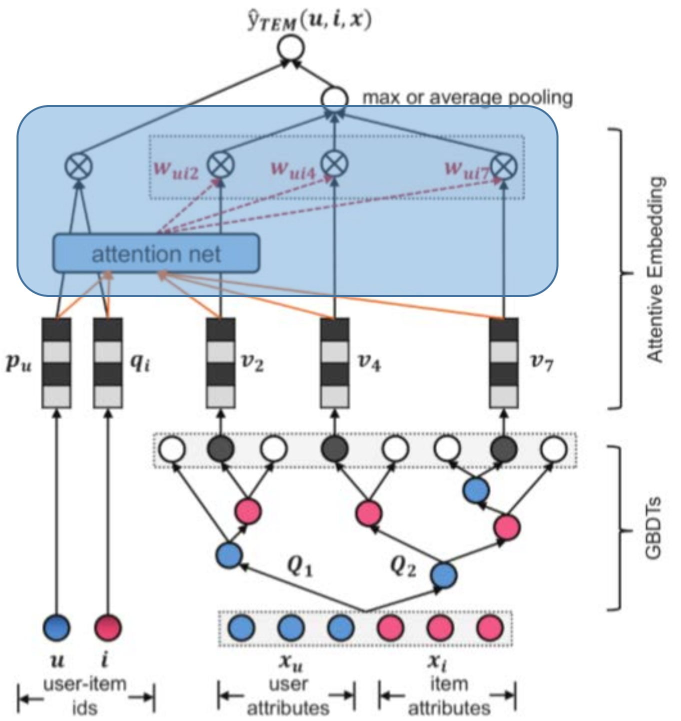
$$\mathcal{V} = \{q_1\mathbf{v}_1, \cdots, q_L\mathbf{v}_L\}$$

Embedding for each cross-feature ID

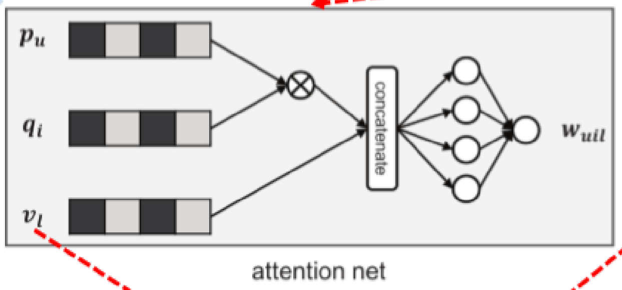
The correlations among cross features may be captured in the embedding space.

# Attention Network

- Primary Consideration: different cross features contribute differently for a prediction
- Solution: Attention Network




$$\begin{cases} \mathbf{e}_{avg}(u, i, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{v}_l \in \mathcal{V}} w_{uil} \mathbf{v}_l, \\ \mathbf{e}_{max}(u, i, \mathcal{V}) = max\_pool_{\mathbf{v}_l \in \mathcal{V}} (w_{uil} \mathbf{v}_l), \end{cases}$$

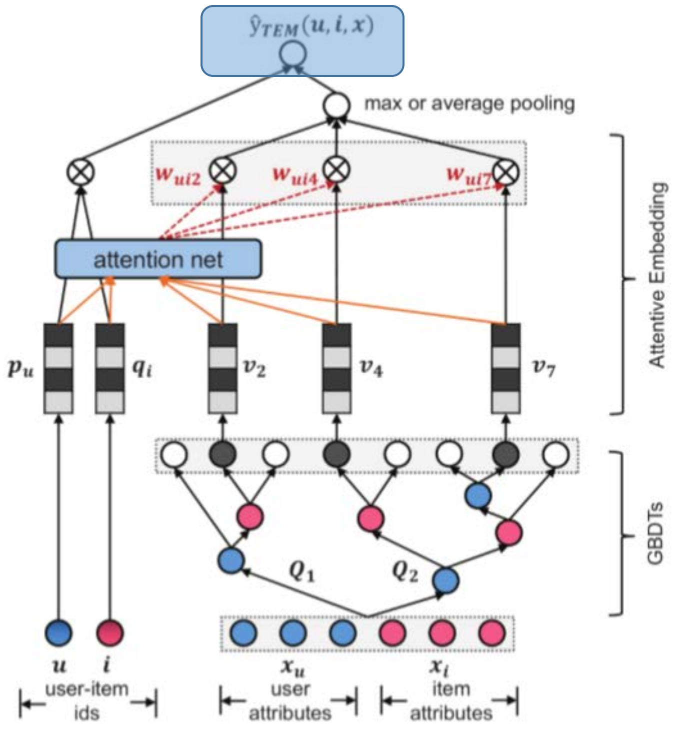



$$\begin{cases} w'_{uil} = \mathbf{h}^\top ReLU(\mathbf{W}([\mathbf{p}_u \odot \mathbf{q}_i, \mathbf{v}_l]) + \mathbf{b}) \\ w_{uil} = \frac{exp(w'_{uil})}{\sum_{(u,i,\mathbf{x}) \in \mathcal{O}} exp(w'_{uil})} \end{cases}$$

- **Easy-to-comprehend cross features**
- **Explicit contribution of each cross feature to the final prediction**

# Final Prediction

- Primary Consideration: explicit decision process & similarity-based + cross feature-based explanation mechanism
- Solution: Simple linear regression




$$\hat{y}_{TEM}(u,i,\mathbf{x}) = b_0 + \sum_{t=1}^{m} b_t x_t + \mathbf{r}_1^{T}\underbrace{(\mathbf{p}_u \odot \mathbf{q}_i)}_{\textbf{Similarity}} + \mathbf{r}_2^{T}\underbrace{\mathbf{e}(u,i,\mathcal{V})}_{\textbf{Cross Feature}}$$

$$\mathcal{L} = \sum_{(u,i,\mathbf{x})\in\mathcal{O}} -y_{ui}\log\sigma(\hat{y}_{ui}) - (1-y_{ui})\log(1-\sigma(\hat{y}_{ui})),$$

- **Pointwise logloss**
- **Pointwise regression loss**
- **Pairwise Ranking loss**

# OUTLINE

- Introduction
- Tree-enhanced Embedding Model
- Experimental Results
- Conclusion

# Experimental Settings

- Research Questions:
  - **RQ1:** Compared with the state-of-the-art recsys methods, can TEM achieve comparable accuracy?
  - **RQ2:** Can TEM make the recsys results easy-to-interpret by using cross features and the attention network?
  - **RQ3:** how do different hyper-paramaters settings affect TEM?
- Tasks: Attraction Recommendation & Restaurant Recommendation
- Dataset: TripAdvisor
  - (https://www.tripadvisor.cn/)

**Table 2: Statistics of the datasets.**

| Dataset | User# | User Feature# | Item# | Item Feature# | Interaction# |
|---|---|---|---|---|---|
| LON-A | 16,315 | 3,230 | 953 | 4,731 | 136,978 |
| NYC-R | 15,232 | 3,230 | 6,258 | 10,411 | 129,964 |

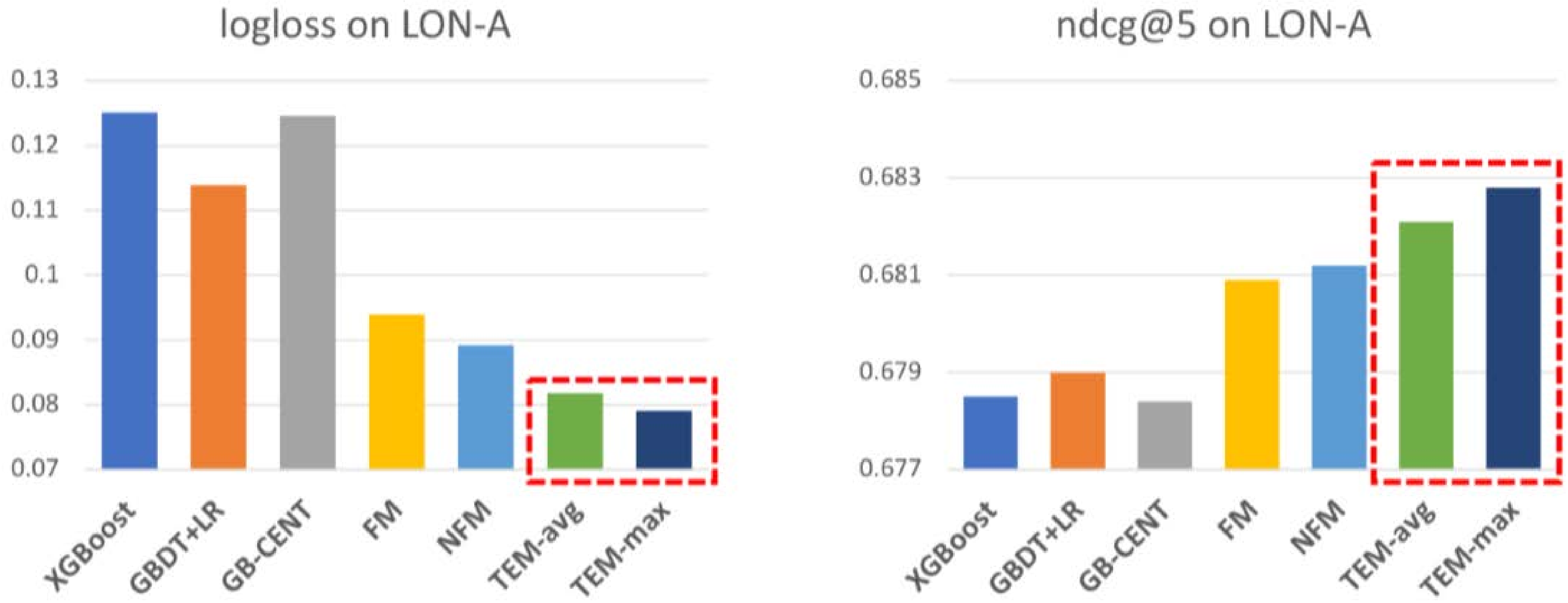
**Table 3: Statistics of the side information, where the dimension of each feature is shown in parentheses.**

| Side Information | Features (Category#) |
|---|---|
| LON-A/NYC-R User Feature | Age (6), Gender (2), Expert Level (6), Traveler Styles (18), Country (126), City (3,072) |
| LON-A Attraction Feature | Attributes (89), Tags (4,635), Rating (7) |
| NYC-R Restaurant Feature | Attributes (100), Tags (10,301), Price (3), Rating (7) |

# Baselines

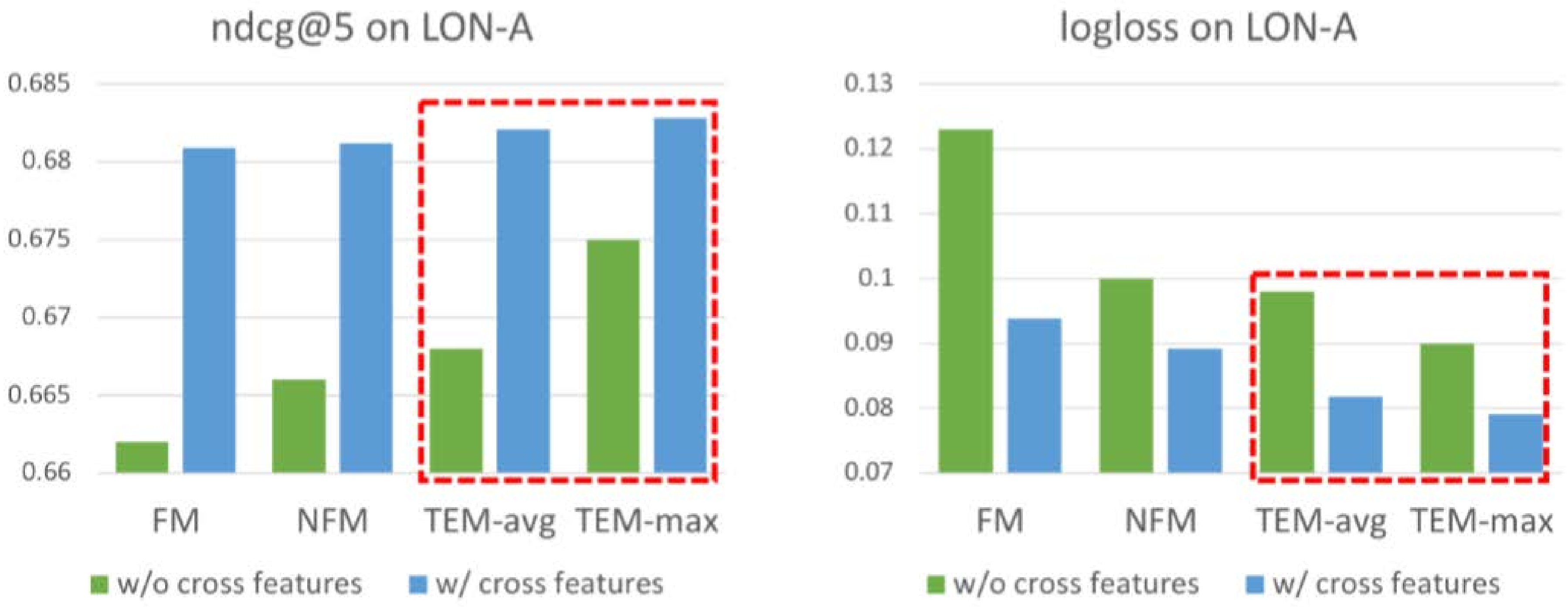
- **XGBoost:** the state-of-the-art tree-based model
- **GBDT+LR [ADKDD'14]:** feeding the cross features extracted from GBDT into the logistic regression
- **GB-CENT [WWW'17]:** modeling categorical features with embedding-based model, numerical features with decision trees.
- **FM:** a generic embedding model that implicitly models all the second-order cross features
- **NFM [SIGIR'17]:** the state-of-the-art factorization model under the neural network framework
- **Evaluation Protocols:**
  - logloss: indicate the generalization ability of each model
  - ndcg@k: reflect the top-k recommendation performance

# RQ1: Overall Performance Comparison



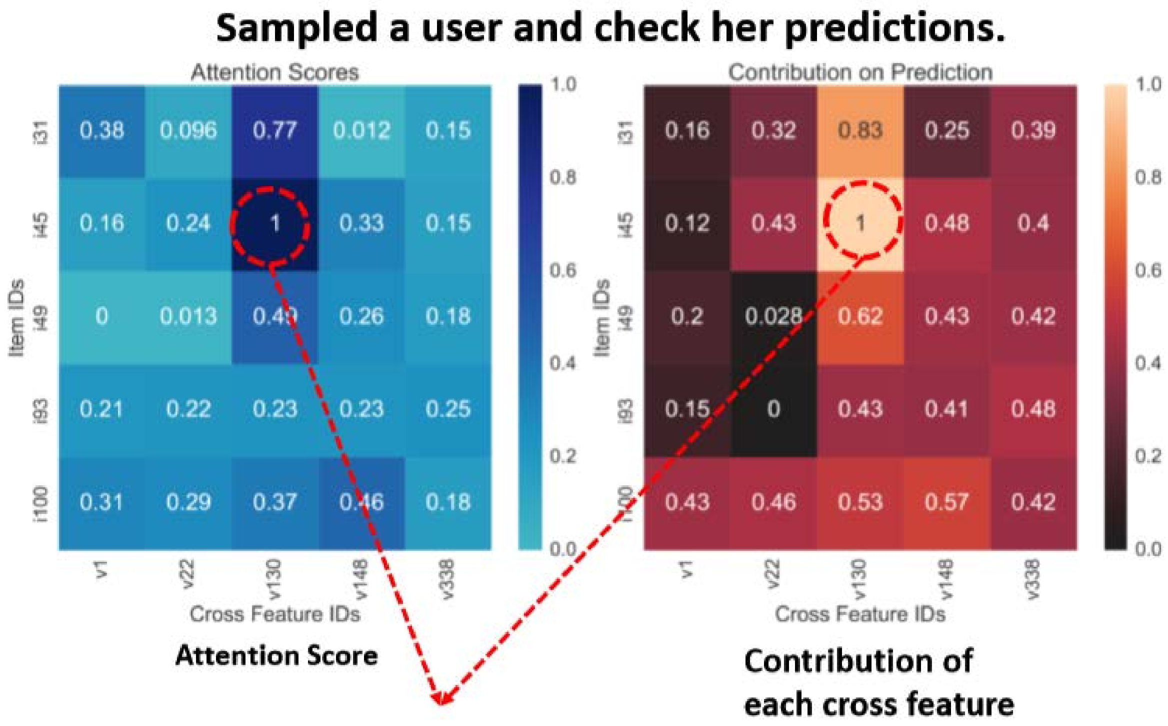

- **Observations:** Comparable Expressiveness & Accuracy
  - TEM achieves the best performance w.r.t. logloss.
  - TEM achieves comparable ndcg@5 to NFM

# RQ1: Overall Performance Comparison




- **Without cross feature modeling:**
  - All methods have worse performance
  - TEM is still better than others, due to the utility of attention network (can learn which features are more important for a user-item prediction).

# RQ2: Case Study of Explainability

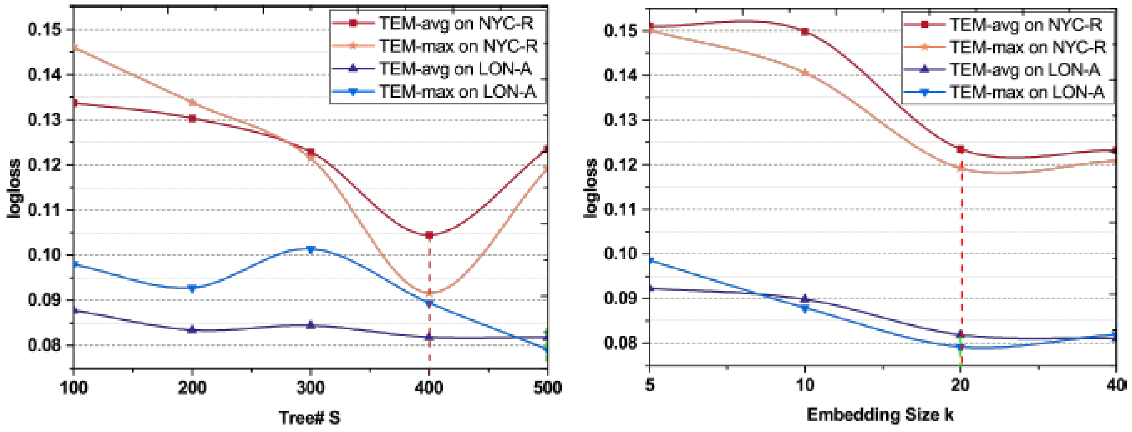



- V130: User Gender=Female] & [User Style=Peace and Quiet Seeker] ⇒ [Item Attribute=Sights & Landmarks] & [Item Tag=Walk Around]
- V148: [User Age=30-40] & [User Country=USA] ⇒ [Item Tag=Top Deck & Canary Wharf]

We attribute the user's preferences on The View from the Shard to her special interests in the item aspects of Walk Around, Top Deck & Canary Wharf.

**TEM can provide more informative explanations based on a user's preferred cross features.**

# RQ3: Hyper-parameter Studies




(a) logloss vs. tree number $S$ (b) logloss vs. embedding size $k$

**Figure 6: Performance comparison of logloss *w.r.t.* the tree number $S$ and the embedding size $k$.**

# OUTLINE

- Introduction
- Tree-enhanced Embedding Model
- Experimental Results
- Conclusion

# Conclusions

- We proposed a tree-enhanced embedding method (TEM), which seamlessly combines the generalization ability of embedding-based models with the explainability of tree-based models.
- Owing to the explicit cross features from tree-based part & the easy-to-interpret attention network, the whole prediction process of our solution is transparent & self-explainable.

# Future Work:

- Jointly learn the tree-based and embedding-based
- Relational reasoning over KG (symbolic logics) + Deep Learning
- How to evaluate the quality of explanations?

Thanks!