Joint Modeling in Recommendations: Fundamentals and Advances

Xiangyu Zhao¹, Yichao Wang², Bo Chen², Pengyue Jia¹, Yuhao Wang¹, Jingtong Gao¹, Huifeng Guo², Ruiming Tang²

¹City University of Hong Kong, ²Huawei Noah’s Ark Lab
Recommender Systems

Age of Information Explosion

Recommend item X to user

Information overload

Items can be Products, News, Movies, Videos, Friends, etc.
Recommender Systems

➢ Recommendation has been widely applied in online services
  • E-commerce, Content Sharing, Social Networking, etc.

Product Recommendation

Frequently bought together

A + B + C

Total price: $208.9

Add all three to Cart
Add all three to List
Recommendation has been widely applied in online services

- E-commerce, Content Sharing, Social Networking, etc.

News/Video/Image Recommendation

For you
Recommended based on your interests

This Research Paper From Google Research Proposes A ‘Message Passing Graph Neural Network’ That Explicitly Models Spatio-Temporal Relations
MarkTechPost - 2 days ago

Tested: Brydge MacBook Vertical Dock, completing my MacBook Pro desktop
YouTube - 23 hours ago
Recommender Systems

➢ Recommendation has been widely applied in online services
  • E-commerce, Content Sharing, Social Networking, etc.
Deep Recommender Architecture

Advantages

- Feature representations of users and items
- Non-linear relationships between users and items
Why Joint Modeling?

Feature Embedding Layer
High/low-frequency features
embedding sizes

Input Layer
Feature selection
Why Joint Modeling?

**Output Layer**
BCE, BPR, MSE

**Feature Interaction Layer**
Pooling, convolution, and the number of layers, inner product, outer product, convolution, etc.

**System Design**
Hardware infrastructure, data pipeline, information transfer, implementation, deployment, optimization, evaluation, etc.
Why Joint Modeling?

V.S.

Recommendation

Engine
Joint Modeling in Recommendations

- Handling the inter-dependency between users and items under more complex circumstances

Advantages
- One model for several situations
- Performance improvement caused by information sharing in different situations

Two typical representatives:
- Multi-task recommendation (MTR)
- Multi-scenario recommendation (MSR)
Joint Modeling in Recommendations

➢ More joint modeling methods:
  - Multi-modal recommendation
  - Multi-interest recommendation

- Multi-behavior recommendation
- Large language model-based recommendation
Agenda

Introduction

Multi-task Recommendation

More Joint-learning Methods

Preliminary

Multi-scenario recommendation

MTR+MSR

Conclusion

Future Work

Xiangyu Zhao

Yuhao Wang

Jingtong Gao

Pengyue Jia

Yichao Wang

Xiangyu Zhao
Why Joint Modeling?

Joint Modeling

- Multi-Task
- Multi-Scenario

Joint with Heterogeneous Data

- Multi-Behavior
- Multi-Interest
- Multi-Modal
- Large Language Model Based
Why Joint Modeling?

➢ Multi-Task Recommendation:
  • Independent tasks: Comments, repost, likes, bookmarks
  • Multi-stage conversion tasks: click, application, approval, activation ...

10:20 · 2023/7/31 · 15.3K Views
8 Retweets 1 Quote 13 Likes 3 Bookmarks

How to extract useful information from other tasks?
How to capture task dependences and resolve the sparsity issue?

Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. KDD 2021.
Why Joint Modeling?

➢ Multi-Scenario Recommendation: construct multiple scenarios for user diverse requirements.

How to extract more comprehensive user portrait from interactions in different scenarios, and make recommendations based on the characteristics of the current scenario?
Why Joint Modeling?

- Multi-Modal Modeling: user interactions, images, text ...

How to extract and align data from different modalities?
Why Joint Modeling?

➢ Multi-Behavior Modeling: click, download, like, buy

How to learn the relationship between different type of behaviors?
Why Joint Modeling?

Multi-Interest Modeling: behaviors → interests

How to accurately and efficiently extract users’ diverse interests from user behaviors?
**Why Joint Modeling?**

- **Large Language Model-based Recommendation**

  - Trained on labeled data with supervised learning
  - Collaborative signals
  - ID-based in-domain collaborative knowledge

  - Pre-trained on large-scale corpora with self-supervised learning
  - Semantic signals
  - Generalization, reasoning and open-world knowledge
Relations and Formulations of Joint Modeling

Multi-Scenario

Multi-Task

Task/scenario adaption

Representation extraction

Multi-Interest

Multi-Behavior

Multi-Modality

Joint Modeling

\[ \text{Joint Modeling} \]

\[ E^{\text{Merge}} = \text{U}(E, E^{\text{Ext}}, E^{\text{m}}) \]

\[ E^{\text{Ext}} = F(H^{\text{UB}}) \]

\[ H^{\text{UB}} = G(H_1, H_2, ..., H_N) \]

\[ E^{\text{m}} = M(E^{\text{txt}}, E^{v}, ..., E^{p}) \]

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]
Agenda

Introduction

Xiangyu Zhao

Multi-task Recommendation

Yuhao Wang

Multi-scenario recommendation

MTR+MSR

Pengyue Jia

More Joint-learning Methods

Jingtong Gao

Conclusion

Future Work

Xiangyu Zhao
Multi-Task Recommendation (MTR)

Multi-Task Deep Recommender Systems (MTDRS)

➢ How
  • Multi-Task Learning (MTL) + Deep Neural Networks

➢ Why
  • Learning high-order feature interactions and
  • Modeling complex user-item interaction behaviors
Benefits & Challenges

➢ Benefits
   • Mutual enhancement among tasks
   • Higher efficiency of computation and storage

➢ Challenges
   • Effectively and efficiently capture useful information & relevance among tasks
   • Data sparsity
   • Unique sequential dependency
Multi-task Recommendation

Multi-Task

Joint Modeling

$E_{\text{Merge}} = U(E, E^{\text{Ext}}, E^{m})$

Multi-Interest

$E^{\text{Ext}} = F(H^{UB})$

Multi-Behavior

$H^{UB} = G(H_1, H_2, ..., H_N)$

Multi-Modality

$E^{m} = M(E^{\text{txt}}, E^{v}, ..., E^{p})$

Multi-Scenario

$w L(E_{\text{Merge}}, \Theta, \Theta^{t}, \Theta^{s})$

Multi-Task

$w L(E_{\text{Merge}}, \Theta, \Theta^{t}, \Theta^{s})$
Formulation

➢ Problem:
   • Learning MTL model with task-specific parameters \( (\theta^1, ..., \theta^K) \) and shared parameter \( \theta^s \), which outputs the \( K \) task-wise predictions

➢ Optimization problem:

\[
\text{arg min } \mathcal{L} (\theta^s, \theta^1, \cdots, \theta^K) = \text{arg min } \sum_{k=1}^{K} \omega^k L^k (\theta^s, \theta^k)
\]

• \( \mathcal{L}(\theta^s, \theta^k) \): loss function for \( k \)-th task with parameter \( \theta^s, \theta^k \)
• \( \omega^k \): loss weight for \( k \)-th task

BCE loss

\[
L^k (\theta^s, \theta^k) = - \sum_{n=1}^{N} [y^k_n \log (\hat{y}^k_n) + (1 - y^k_n) \log (1 - \hat{y}^k_n)]
\]
MTR, MOR, MSR, MBR

(a) MTR

Data → Scenario → Task-1 → Task-K

(b) MOR

Data → Scenario → Task → Objective-1 → Objective-M

(c) MSR

Data → Scenario-1 → Scenario-2 → Scenario-S

(d) MBR

Task Behavior:
- Click → Like → Buy
### Comparison with CV & NLP

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>Multi-target segmentation and further classification for each object</td>
<td>Utilizing <strong>feature transformation</strong> to represent common features based on a multi-layer feed-forward network</td>
</tr>
<tr>
<td>NLP</td>
<td>Mostly focus on the design of MTL architectures</td>
<td>Based on RNN because of the sequence pattern Can be divided into word-, sentence-, and document-level by granularity</td>
</tr>
</tbody>
</table>
Taxonomy

Parallel  Cascaded  Auxiliary + Main

Task Relation

Multi-task DRS

Methodology

Parameter Sharing
- Hard Sharing
- Sparse Sharing
- Soft Sharing
- Expert Sharing

Optimization
- Negative Transfer
- Multi-objective Trade-off
- Gradient Dominating
- Parameter Conflict

Training Mechanism
- Joint Training
- Reinforcement Learning
- Auxiliary Task Learning
Task Relation

- Parallel
- Cascaded
- Auxiliary + Main

Task Relation
Parallel

- Tasks independently calculated without sequential dependency
- Objective function: Weighted sum with constant loss weights
Cascaded task relationship: **sequential dependency**

- Computation of current task depends on **previous** ones
  - E.g. CTCVR = CTR × CVR

- General formulation:
  \[
  \hat{y}_{n}^{k} (\theta^s, \theta^k) - \hat{y}_{n}^{k-1} (\theta^s, \theta^k) = P (\epsilon_k = 0, \epsilon_{k-1} = 1)
  \]

  - \(\epsilon_k\): Indicator variable for task \(k\)
  - Difference is the probability of the task \(k\) not happening while the task \(k-1\) is observed
<table>
<thead>
<tr>
<th>Model</th>
<th>Problem</th>
<th>Behavior Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESMM [Ma et al., 2018b]</td>
<td>SSB &amp; DS</td>
<td>impression → click → conversion</td>
</tr>
<tr>
<td>ESM² [Wen et al., 2020]</td>
<td>SSB &amp; DS</td>
<td>impression → click → D(O)Action → purchase</td>
</tr>
<tr>
<td>Multi-IPW &amp; DR [Zhang et al., 2020]</td>
<td>SSB &amp; DS</td>
<td>exposure → click → conversion</td>
</tr>
<tr>
<td>ESDF [Wang et al., 2020b]</td>
<td>SSB &amp; DS &amp; time delay</td>
<td>impression → click → pay</td>
</tr>
<tr>
<td>HM³ [Wen et al., 2021]</td>
<td>SSB &amp; DS &amp; micro and macro behavior modeling</td>
<td>impression → click → micro → macro → purchase</td>
</tr>
<tr>
<td>AITM [Xi et al., 2021]</td>
<td>sequential dependence in multi-step conversions</td>
<td>impression → click → application → approval → activation</td>
</tr>
<tr>
<td>MLPR [Wu et al., 2022]</td>
<td>sequential engagement &amp; vocabulary mismatch in product ranking</td>
<td>impression → click → add-to-cart → purchase</td>
</tr>
<tr>
<td>ESCM² [Wang et al., 2022a]</td>
<td>inherent estimation bias &amp; potential independence priority</td>
<td>impression → click → conversion</td>
</tr>
<tr>
<td>HEROES [Jin et al., 2022]</td>
<td>multi-scale behavior &amp; unbiased learning-to-rank</td>
<td>observation → click → conversion</td>
</tr>
<tr>
<td>APEM [Tao et al., 2023]</td>
<td>sample-wise representation learning in SDMTL</td>
<td>impression → click → authorize → conversion</td>
</tr>
<tr>
<td>DCMT [Zhu et al., 2023]</td>
<td>SSB &amp; DS &amp; potential independence priority (PIP)</td>
<td>exposure → click → conversion</td>
</tr>
</tbody>
</table>

SSB: Sample Selection Bias   DS: Data Sparsity
Auxiliary with Main Task

➢ A task specified as the main task
   while associated auxiliary tasks help to improve performance

➢ Probability estimation for main task the probability of auxiliary tasks

➢ Provide richer information across entire space
## Auxiliary with Main Task

<table>
<thead>
<tr>
<th>Model</th>
<th>References</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESDF Multi-IPW and Multi-DR</td>
<td>[Wang et al., 2020b] [Zhang et al., 2020] [Zhao et al., 2021] [He et al., 2022]</td>
<td>Adopt the original recommendation tasks as auxiliaries</td>
</tr>
<tr>
<td>DMTL Metabalance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTRRec</td>
<td>[Li et al., 2020a] [Lin et al., 2022] [Yang et al., 2021] [Yang et al., 2022a]</td>
<td>Manually design various auxiliary tasks</td>
</tr>
<tr>
<td>PICO MTAE Cross-Distill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSRec</td>
<td>[Bai et al., 2022]</td>
<td>Contrastive learning as the auxiliary</td>
</tr>
<tr>
<td>Self-auxiliary*</td>
<td>[Wang et al., 2022b]</td>
<td>Under-parameterized self-auxiliaries</td>
</tr>
</tbody>
</table>
Methodology

- Parameter Sharing
  - Hard Sharing
  - Sparse Sharing
  - Soft Sharing
  - Expert Sharing
- Optimization
  - Negative Transfer
  - Gradient Dominating
  - Parameter Conflict
  - Multi-objective Trade-off
- Training Mechanism
  - Joint Training
  - Reinforcement Learning
  - Auxiliary Task Learning
Parameter Sharing

- Parameter Sharing
  - Hard Sharing
  - Soft Sharing
  - Sparse Sharing
  - Expert Sharing

- Optimization
  - Negative Transfer
  - Multi-objective Trade-off

- Methodology
  - Joint Training
  - Reinforcement Learning
  - Auxiliary Task Learning

- Training Mechanism

- Gradient Dominating
- Parameter Conflict
Parameter Sharing

(a) Hard Parameter Sharing

(b) Sparse Sharing

(c) Soft Parameter Sharing

(d) Expert Sharing
Hard Sharing

➢ Shared bottom layers extract the same information for different tasks,
➢ Task-specific top layers are trained individually

✓ Improving computation efficiency and alleviating over-fitting

✗ Limited capacity of the shared parameter space → Weakly related tasks and noise
Sparse Sharing

➢ Extracting sub-networks for each task by parameter masks from a base network
  ○ Special case of Hard Sharing

✓ Coping with the weakly related tasks flexibly

✗ Negative transfer when updating shared parameters
Soft Sharing

Building separate models for tasks but the information among tasks is **fused by weights** of task relevance.

- Relatively high **flexibility** in parameter sharing v.s. hard sharing.
- Can not reconcile the flexibility.
- Computation cost of the model.
Multi-task Based Sales Predictions for Online Promotions. CIKM 2019.
Expert Sharing

- Employing multiple **expert networks** to extract knowledge from shared bottom
  - Fed into **task-specific** modules like gates
  - Passed into the task-specific tower

- Mainly non-sequential input features
- **Special case** of Soft Sharing
MMoE modeling task relationships in multi-task learning with multi-gate mixture-of-experts. KDD 2018.
Processing **non-sequential** input features, while the remaining models is ameliorated based on MMOE.

Processing **sequential** input features utilizing LSTM & sequential experts.
Optimization

Parameter Sharing
- Hard Sharing
- Sparse Sharing

Soft Sharing
- Expert Sharing

Optimization
- Negative Transfer
- Multi-objective Trade-off
  - Gradient Dominating
  - Parameter Conflict

Training Mechanism
- Joint Training
- Reinforcement Learning
- Auxiliary Task Learning
## Negative Transfer

Gradient dominating \( \| \nabla_{\theta} L^k(\theta) \| \)

<table>
<thead>
<tr>
<th>Works</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaTask [Yang et al., 2022b]</td>
<td>Quantifying task dominance of shared parameters, calculate task-specific accumulative gradients</td>
</tr>
<tr>
<td>MetaBalance [He et al., 2022]</td>
<td>Flexibly balancing the gradient magnitude proximity between auxiliary and target tasks by a relax factor</td>
</tr>
</tbody>
</table>

Opposite directions of gradient \( + - \nabla_{\theta} L^k(\theta) \)

<table>
<thead>
<tr>
<th>Works</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLE     [Tang et al., 2020]</td>
<td>Proposing customized gate control (CGC) separating shared and task-specific experts</td>
</tr>
<tr>
<td>CSRec   [Bai et al., 2022]</td>
<td>Alternating training procedure and contrastive learning on parameter masks to reduce the conflict probability</td>
</tr>
</tbody>
</table>
\[ \theta^{t+1} = \theta^t - \alpha \cdot G_{total}^t \]

\[ G_{total}^t = \nabla_\theta \mathcal{L}_{total}^t = \nabla_\theta \mathcal{L}_{tar}^t + \sum_{i=1}^{K} \nabla_\theta \mathcal{L}_{aux,i}^t \]
\[ \theta^{t+1} = \theta^t - \alpha \cdot G^t_{total} \]

\[ G^t_{total} = \nabla_\theta \mathcal{L}^t_{total} = \nabla_\theta \mathcal{L}^t_{tar} + \sum_{i=1}^{K} \nabla_\theta \mathcal{L}^t_{aux,i} \]
MetaBalance

\[ \theta^{t+1} = \theta^t - \alpha \times G^{t}_{\text{total}} \]

\[ G^{t}_{\text{total}} = \nabla_{\theta} L^{t}_{\text{total}} = \nabla_{\theta} L^{t}_{\text{tar}} + \sum_{i=1}^{K} \nabla_{\theta} L^{t}_{\text{aux},i} \]

\[ G^{t}_{aux,i} \leftarrow (G^{t}_{aux,i} \times \frac{\| G^{t}_{\text{tar}} \|}{\| G^{t}_{aux,i} \|}) \times r + G^{t}_{aux,i} \times (1 - r) \]
Multi-objective Trade-off

Objectives optimized regardless of the **potential conflict**

<table>
<thead>
<tr>
<th>Works</th>
<th>Trade-off</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Wang <em>et al.</em>, 2022b]</td>
<td>Minimizing task conflicts and improving multi-task generalization</td>
</tr>
</tbody>
</table>
Training Mechanism

Training process & Learning strategy

- Methodology
  - Parameter Sharing
    - Hard Sharing
    - Sparse Sharing
  - Soft Sharing
  - Expert Sharing
  - Optimization
    - Negative Transfer
    - Gradient Dominating
    - Parameter Conflict
    - Multi-objective Trade-off
  - Joint Training
  - Reinforcement Learning
  - Auxiliary Task Learning
**Parallel manner**

<table>
<thead>
<tr>
<th>Category</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session-based RS</td>
<td>[Shalaby et al., 2022]</td>
</tr>
<tr>
<td></td>
<td>[Qiu et al., 2021]</td>
</tr>
<tr>
<td></td>
<td>[Meng et al., 2020]</td>
</tr>
<tr>
<td>Route RS</td>
<td>[Das, 2022]</td>
</tr>
<tr>
<td>Knowledge graph enhanced RS</td>
<td>[Wang et al., 2019]</td>
</tr>
<tr>
<td>Explainability</td>
<td>[Lu et al., 2018]</td>
</tr>
<tr>
<td></td>
<td>[Wang et al., 2018]</td>
</tr>
<tr>
<td>Graph-based RS</td>
<td>[Wang et al., 2020a]</td>
</tr>
</tbody>
</table>
Sequential user behaviors as MDP

<table>
<thead>
<tr>
<th>Summary</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulating MTF as MDP and use batch RL to optimize long-term user satisfaction</td>
<td>[Zhang et al., 2022b]</td>
</tr>
<tr>
<td>Using an actor-critic model to learn the optimal fusion weight of tasks rather than greedy ranking strategies</td>
<td>[Han et al., 2019]</td>
</tr>
<tr>
<td>Using dynamic critic networks to adaptively adjust the fusion weight considering the session-wise property</td>
<td>[Liu et al., 2023]</td>
</tr>
</tbody>
</table>
### Auxiliary Task Learning

#### Joint training & Others

<table>
<thead>
<tr>
<th>Summary</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employing Expectation-Maximization (EM) algorithm for optimization</td>
<td>ESDF [Wang et al., 2020b]</td>
</tr>
<tr>
<td>Trained with task-specific sub-networks</td>
<td>Self-auxiliaries [Wang et al., 2022b]</td>
</tr>
</tbody>
</table>
Application Fields

- **E-commerce**: Main focus

- **Advertising**
  - **Utility & Cost**
    i. MM-DFM [Hou et al., 2021]: Performing multiple conversion prediction tasks in different observation duration
    ii. MetaHeac [Zhu et al., 2021]: Handling audience expansion tasks on content-based mobile marketing
    iii. MVKE [Xu et al., 2022]: Performing user tagging for online advertising

- **Social media**
  i. MMoE [Zhao et al., 2019b]: YouTube - engagement and satisfaction
  ii. LT4REC [Xiao et al., 2020]: Tencent Video
  iii. BatchRL-MTF [Zhang et al., 2022b]: Tencent short video platform
## Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Stage</th>
<th>Tasks</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>Recall &amp; Ranking</td>
<td>Rating, Explanation</td>
<td><a href="https://www.yelp.com/dataset/">https://www.yelp.com/dataset/</a></td>
</tr>
<tr>
<td>Kuairand [18]</td>
<td>Recall &amp; Ranking</td>
<td>Click, Like, Follow, Comment, ...</td>
<td><a href="https://kuairand.com/">https://kuairand.com/</a></td>
</tr>
<tr>
<td>Tenrec [77]</td>
<td>Recall &amp; Ranking</td>
<td>Click, Like, Share, Follow, ...</td>
<td><a href="https://github.com/yuangh-x/2022-NIPS-Tenrec/">https://github.com/yuangh-x/2022-NIPS-Tenrec/</a></td>
</tr>
</tbody>
</table>
## Challenges & Future Directions

<table>
<thead>
<tr>
<th>Topic</th>
<th>Challenge &amp; future direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Transfer</td>
<td>• Extra complex inter-task correlation</td>
</tr>
<tr>
<td></td>
<td>• What, where, and when to transfer to alleviate negative transfer</td>
</tr>
<tr>
<td>AutoML</td>
<td>• Existing models only focus on the <strong>parameter sharing routing</strong>, while other components and hyper-parameters still under-explored</td>
</tr>
<tr>
<td>Explainability</td>
<td>• Complex task relevance</td>
</tr>
<tr>
<td>Task-specific Biases</td>
<td>• Most existing models only focus on <strong>one</strong> specific bias</td>
</tr>
<tr>
<td></td>
<td>• <strong>Multiple</strong> bias should be tackled in future</td>
</tr>
</tbody>
</table>
## Challenges & Future Directions

<table>
<thead>
<tr>
<th>Topic</th>
<th>Challenge &amp; future direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Transfer</td>
<td>• Extra complex inter-task correlation</td>
</tr>
<tr>
<td></td>
<td>• What, where, and when to transfer to alleviate negative transfer</td>
</tr>
<tr>
<td>AutoML</td>
<td>• Existing models only focus on the parameter sharing routing, while other components and hyper-parameters still under-explored</td>
</tr>
<tr>
<td>Explainability</td>
<td>• Complex task relevance</td>
</tr>
<tr>
<td>Task-specific Biases</td>
<td>• Most existing models only focus on one specific bias</td>
</tr>
<tr>
<td></td>
<td>• <strong>Multiple</strong> bias should be tackled in future</td>
</tr>
<tr>
<td>Topic</td>
<td>Challenge &amp; future direction</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Negative Transfer            | • Extra complex inter-task correlation  
                                • What, where, and when to transfer to alleviate negative transfer |
| AutoML                       | • Existing models only focus on the parameter sharing routing, while other components and hyper-parameters still under-explored |
| Explainability               | • Complex task relevance                                                                   |
| Task-specific Biases         | • Most existing models only focus on one specific bias                                      
                                • **Multiple** bias should be tackled in future
## Challenges & Future Directions

<table>
<thead>
<tr>
<th>Topic</th>
<th>Challenge &amp; future direction</th>
</tr>
</thead>
</table>
| Negative Transfer    | • Extra complex inter-task correlation  
                        • What, where, and when to transfer to alleviate negative transfer                                                                                     |
| AutoML               | • Existing models only focus on the **parameter sharing routing**, while other components and hyper-parameters still under-explored                         |
| Explainability       | • Complex task relevance                                                                                                                                     |
| Task-specific Biases | • Most existing models only focus on **one** specific bias  
                        • **Multiple** bias should be tackled in future                                                                                            |
Task relation:
- Parallel, Cascaded, Auxiliary with Main

Methodology:
- Parameter Sharing, Optimization, Training Mechanism
https://arxiv.org/abs/2302.03525

Multi-Task Deep Recommender Systems: A Survey

YUHAO WANG*, HA TSZ LAM*, and YI WONG*, City University of Hong Kong
ZIRU LIU, City University of Hong Kong
XIANGYU ZHAO†, City University of Hong Kong
YICHAO WANG, BO CHEN, HUIFENG GUO, and RUIMING TANG†, Huawei Noah’s Ark Lab
Multi-Scenario Recommender Systems

\[ wL(E_{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

\[ E_{\text{Merge}} = U(E, E^{\text{Ext}}, E^m) \]

\[ E^{\text{Ext}} = F(H^{UB}) \]

\[ H^{UB} = G(H_1, H_2, \ldots, H_N) \]

\[ E^m = M(E^{\text{Ext}}, E^v, \ldots, E^p) \]

\[ wL(E_{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

Joint Modeling

Multi-Interest

Multi-Behavior

Multi-Modality

Multi-Scenario

Multi-Task

Multi-Interest

Multi-Behavior

Multi-Modality
Multi-Scenario Recommender Systems:
• By using a unified model to simultaneously model multiple scenarios, the goal of improving the effects of different scenarios at the same time is achieved through information transfer between scenarios.

Importance:
• Time/Memory efficiency; Maintenance cost
• Accuracy

Classification on Methods:
• Shared-Specific network paradigm
• Dynamic weight
• Multi-scenario & Multi-task recommendation

Formulation:
\[
\omega \mathcal{L}(E_{\text{Merge}}, \theta, \theta^t, \theta^s)
\]
• \(\theta\): parameters of the backbone network
• \(\theta^s\): parameters of modeling scenarios
Recommendation Scenarios

➢ What is Scenario?
  • Homepage, Searching page, Detailed page ...
  • Food, Leisure and entertainment, ...
  • Usually refers to different business scenarios

➢ Scenario and Domain?
  • Generally do not make a distinction
  • The same in this tutorial
Commonalities and Diversities

➢ Commonalities
  • User Overlap

➢ Commonalities
  • Item Overlap
Commonalities and Diversities

➢ Diversities
  • The specific user group may be different
  • User’s interest changes with the scenarios
Table of Contents

Shared-specific network paradigm

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, (\Theta^{\text{shared}}, \Theta^{\text{specific}})) \]

Dynamic weight

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

Multi-Scenario & Multi-Task

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s, \Theta^T) \]
Shared-specific Network Paradigm

Fusion

Shared Network

Specific Network for Scenario 1

Specific Network for Scenario 2

Specific Network for Scenario 3

One data sample from scenario 1
Shared-specific Network Paradigm

Shared Network

Specific Network for Scenario 1

Specific Network for Scenario 2

Specific Network for Scenario 3

Fusion

One data sample from scenario 2
Shared-specific Network Paradigm

- Shared Network
- Specific Network for Scenario 1
- Specific Network for Scenario 2
- Specific Network for Scenario 3

Fusion

One data sample from scenario 3
Motivation:
• Training individual models for each domain → does not fully use the data from all domains
• Data across domains owns commonalities and characteristics

Target:
• Use a single model to serve multiple domains simultaneously
• Shared network → commonalities
• Specific network → characteristics

Methods:
• Partitioned Normalization
• STAR Topology
• Auxiliary Network
STAR Details

➢ Partitioned Normalization (PN)

➢ Training

\[ z' = (\gamma \ast \gamma_p) \frac{z - \mu}{\sqrt{\sigma^2 + \epsilon}} + (\beta + \beta_p) \]

➢ Testing

\[ z' = (\gamma \ast \gamma_p) \frac{z - E_p}{\sqrt{\text{Var}_p + \epsilon}} + (\beta + \beta_p) \]

➢ Batch Normalization (BN)

➢ Training

\[ z' = \gamma \frac{z - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \]

➢ Testing

\[ z' = \gamma \frac{z - E}{\sqrt{\text{Var} + \epsilon}} + \beta \]
STAR Details

STAR Topology

The final weight and bias for $p$-th domain is obtained by:

$$W_p^* = W_p \otimes W, b_p^* = b_p + b$$

The output for $p$-th domain is derived by:

$$out_p = \phi((W_p^*)^\top in_p + b_p^*)$$
Motivation

- Traffic characteristics of different scenarios are significantly different (individual data scale or representative topic)

Target

- Train a unified model to serve all scenarios
Cross-Scenario Behavior Extract Layer

How to aggregate the sequence?

\[ p^{B_i} \] is item behavior sequence

\[ p^i_k = [e_{itemId}||e_{destination}||e_{category}||\cdots] \]

\[ p^{B^s} \] is scenario context sequence

\[ p^s_k = [e_{scenarioId}||e_{scenarioType}||e_{behaviorTime}||\cdots] \]

\[ \alpha^i_k = \frac{\exp(\psi(p^i_k, p^i_j))}{\sum_{l=1}^{n} \exp(\psi(p^i_k, p^i_l))} \]

\[ \alpha^s_k = \frac{\exp(\psi(p^s_k, p^s_j))}{\sum_{l=1}^{n} \exp(\psi(p^s_k, p^s_l))} \]

\[ \alpha^i_k \] and \[ \alpha^s_k \] indicate the relevance between user’s kth behavior item and the target item or target scenario
Cross-Scenario Behavior Extract Layer

How to aggregate the sequence?

\[
\alpha^i_k = \frac{\exp(\psi(p^i_k, p^i_j))}{\sum_{l=1}^{L} \exp(\psi(p^i_j, p^i_l))},
\]

\[
\alpha^s_k = \frac{\exp(\psi(p^s_k, p^s_l))}{\sum_{l=1}^{L} \exp(\psi(p^s_l, p^s_k))},
\]

\[
p^i_k = [e_{itemId} || e_{destination} || e_{category}] \cdots
\]

\[
v_{cb} = \sum_{k=1}^{t} \alpha^i_k \odot \alpha^s_k \odot p^i_k
\]
Scenario-Specific Transform Layer

\[ v' = v \otimes \beta_i + \gamma_i \]

Mixture of Debias Experts

Multi-expert network. Each scenario has some scenario-specific experts and all the scenarios share several common experts.
The output of experts:

\[ S^k(x) = [o_{k,1}, o_{k,2}, \ldots, o_{k,m_k}, o_{s,1}, o_{s,2}, \ldots, o_{s,m_s}]^T \]

Final predicted score of scenario \( k \)

\[ y^k(x) = w^k(x)S^k(x) \]

\( w^k(x) \) is derived by a single-layer feed-forward network with a SoftMax activation function.
Motivation

- Separate model for each scenario, ignoring the cross-domain overlapping of user groups and items
- One shared model trained on mix data, model performance may decrease when different domains conflict

Target

- Modeling commonalities and diversities $\rightarrow$ common networks and domain-specific networks
- Tackle the feature-level domain adaptation $\rightarrow$ domain-specific batch normalization, domain interest adaptation layer
ADI Details

Backbone Network

Shared Network & Domain-Specific Network

\[ a_{zk} = \frac{W_{zk}^{shared}(f_{domain}) + b_{zk}^{shared}}{\sum_{n=1}^{K} (W_{zn}^{shared}(f_{domain}) + b_{zn}^{shared})} \]

\[ E_{shared} = \sum_{k=1}^{K} \alpha_k MLP_{shared}^k(F) \]

\[ E_{spec} = MLP_{spec}^{(d)}(F^{(d)}) \]

\( f_{domain} \) Domain indicator embedding

\( F^{(d)} \) Data from domain \( d \)

\( K \) hyperparameter, number of Shared Network

\( D \) domains, \( D \) Domain-Specific Network
ADI Details

Backbone Network

Fusion Layer

\[\beta_1^{(d)} = \sigma(W_{fusion\_spec}\cdot f_{domain})\]
\[\beta_2^{(d)} = \sigma(W_{fusion\_shared}\cdot f_{domain})\]
\[E_{fusion}^{(d)} = \text{concat}(\beta_1^{(d)}E_{spec}^{(d)}\mid \beta_2^{(d)}E_{shared}^{(d)})\]

Domain-Specific Forward Network

\[E = FC_{forward}^{(d)}(E_{fusion}^{(d)})\]
Domain Adaptation

Domain-Specific Batch Normalization (DSBN)

\[ \tilde{X}^{(d)} = \alpha^{(d)} \frac{X^{(d)} - \mu^{(d)}}{\sqrt{(\sigma^{(d)})^2 + \epsilon}} + \beta^{(d)} \]
Domain Adaptation

Domain Interest Adaptation Layer

\[ \alpha^{(d)} = F_{se}(\text{concat}(F_{avg}(F_1^{(d)}) | \cdots | F_{avg}(F_N^{(d)}))) \]

\[ \hat{F}^{(d)} = \alpha^{(d)} \otimes \text{concat}(F_1^{(d)} | \cdots | F_N^{(d)}) \]

- \( F_i^{(d)} \) denotes the \( i \)th feature of embedded input collected from domain \( d \)
- \( F_{se} \) denotes a \((\text{FC, ReLU, FC})\) block and \( F_{avg} \) denotes the average pooling operator.
Table of Contents

Shared-specific network paradigm
\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, (\Theta^{\text{shared}}, \Theta^{\text{specific}})) \]

Dynamic weight
\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

Multi-Scenario & Multi-Task
\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s, \Theta^T) \]
Dynamic Weight

Why Dynamic?

Backbone Model

Dynamic Weight Generator

Input Features

Scenario Sensitive Features
Target
• To mine and model implicit scenarios

Methods
• Scenario Learning Module to project data samples, and assign scenarios to these data samples
MUSENET Details

Soft Assignment

\[ \Lambda = \{c_1, c_2, \ldots, c_K\} \]

\[ Q_s(c|x) = P_\omega(c|x) = \frac{\exp(-d(f_\omega(x), c))}{\sum_{c'} \exp(-d(f_\omega(x), c'))} \]

\[ \{a_{1,i}, a_{2,i}, \ldots, a_{k,i}\} \]

Hard Assignment

Gumbel-Softmax trick

\[ a_{k,i} = \frac{\exp((\log \pi_{k,i} + g_{k,i})/\tau)}{\sum_{k'=1}^{K} \exp((\log \pi_{k',i} + g_{k',i})/\tau)} \]

\[ \pi_{k,i} = \frac{\exp(-d(f_\omega(x_i), c_k))}{\sum_{k'=1}^{K} \exp(-d(f_\omega(x_i), c_{k'}))} \]
Given the $\omega$, the objective is to minimize the distance expectation from each data sample to the corresponding scenario prototypes

$$
\mathcal{L}_C(\Lambda, \Theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} a_{k,i} d(f_\omega(x_i), c_k)
$$

Final Prediction

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} a_{k,i} z_{k,i}\right)$$
Motivation
• Lacking of fine-grained and decoupled information transfer controls among multiple scenarios
• Insufficient exploitation of entire space samples
• Item’s multi-scenario representation disentanglement problem

Methods
• Multi-Layer Scenario Adaptive Transfer (ML-SAT) module
• Two-stage training process including pre-training and fine-tune
Pre-training Stage and Fine-Tune Stage

(a) Pre-Training Stage of SASS

(b) Fine-Tune Stage of SASS

\[ L_{ij} = -\log \frac{\exp(\frac{\text{sim}(e^i_s, e^j_s)}{\tau})}{\sum_{k=1, k \neq i}^{2N} \exp(\frac{\text{sim}(e^i_s, e^k_s)}{\tau})} \]
Multi-Layer Scenario Adaptive Transfer Module

Scenario Modeling

\[ a = f(W_a x_a + b_a) \]

Scenario-adaptive gate unit

\[ r_l = \sigma(W_r^l [g_l, s_{l-1}] + W_{br} a) \]
\[ h_l = \tanh(W_h^l [r_l \cdot g_l, s_{l-1}]) \]
\[ z_l = \sigma(W_z^l [g_l, s_{l-1}] + W_{bz} a) \]
\[ s_l = (1 - z_l) \cdot s_{l-1} + z_l \cdot h_l \]
Multi-Layer Scenario Adaptive Transfer Module

Scenario Bias Fusion

\[ e_s = \alpha \cdot s_T + (1 - \alpha) \cdot a \]
\[ \alpha = \sigma(W_0[s_T, a]) \]
Table of Contents

Shared-specific network paradigm
\[ wL(E^{Merge}, \Theta, \Theta^t, (\Theta^{shared}, \Theta^{specific})) \]

Dynamic weight
\[ wL(E^{Merge}, \Theta, \Theta^t, \Theta^s) \]

Multi-Scenario & Multi-Task
\[ wL(E^{Merge}, \Theta, \Theta^t, \Theta^s, \Theta^T) \]
Multi-Scenario & Multi-Task Studies

(a) MTR

(b) MSR
Target
- Develop a unified ranking model for multi-task and multi-scenario problem

Methods
- Independent/non-shared embeddings for each task and scene, new tasks or scenes could be added easily
- A simplified network is chosen beyond the embedding layer, which largely improves the ranking efficiency for online service.
MTMS Details

Independent embeddings for every “task+scenario”
Aggregation of different components -> shared modeling
Loss function: sum of different tasks, -> performance not be hurt by auxiliary tasks (E.g. CTR)
MTMS Details

➢ First step: embedding update, no shared information modeling
Second step: network fine tune. Embedding is fixed. DNN has more fields for inputs.
Target
• Develop a unified framework that could realize both MSL and MTL requirements

Methods
• Propose AESM\textsuperscript{2}, a flexible hierarchical structure where the multi-task layers are stacked over the multi-scenario layers
• General expert selection algorithm
AESM²

Task-related

Scenario-related
Multi-Scenario Layer

- Input $x$, scenario embedding $s$, Gaussian noise $n_j$, learnable parameter $s_j$, $m$ scenarios/gates. For every expert:

\[
G = [g_1, \ldots, g_m] \\
g_j = S_j[x, s] + \eta_j \\
\tilde{G} = \text{softmax}(G)
\]

- Expert selection

\[
\mathcal{E}_{sp} = \text{TopK}(h_1^p, \ldots, h_n^p) \\
h_k^p = -KL(p_j, \tilde{G}[k,:]) \\
\mathcal{E}_{sh} = \text{TopK}(h_1^q, \ldots, h_n^q) \\
h_k^q = -KL(q_j, \tilde{G}[k,:]) \\
p_j (e.g., [1, \ldots, 0]) \\
q_j = [1/m, \ldots, 1/m]
\]

- Expert aggregation:

\[
\hat{g}_j[k] = \begin{cases} 
  g_j[k], & \text{if } k \in \mathcal{E}_{sh} \cup \mathcal{E}_{sp} \\
  -\infty, & \text{else}
\end{cases}
\]

Specific

Shared

\[
z_j = \text{ScenarioLayer}(x, s_j) = \text{MMoE}(x, \hat{g}_j)
\]
AESM²

Multi-Task Layer

➢ Input $x$, scenario embedding $s$, task embedding $t_k$, Gaussian noise $n_j$, learnable parameter $T_k$, the gating scalar $g_k$ for $k$-th task:

$$g_k = T_k [x, s, t_k] + \eta_k$$

$$z_k = \text{TaskLayer}(z_j, t_k) = \text{MMoE}(z_j, \hat{g}_k)$$

➢ Output layer

$$\hat{y}_k = \sigma(\text{MLP}(z_k))$$
PEPNet

➢ Motivation
  • The imperfectly double seesaw phenomenon
  • More accurate personalization estimates can alleviate the imperfectly double seesaw problem

➢ Target
  • Jointly model multi-domain and multi-task
  • an efficient, low-cost deployment and plug-and-play method that can be injected in any network.
Gate Neural Unit (Gate NU)

\[ x_1 = \text{Relu} \left( x^{(0)} W^{(0)} + b^{(0)} \right) \]

\[ x_2 = \gamma \times \text{Sigmoid} \left( x^{(1)} W^{(1)} + b^{(1)} \right), \quad x_2 \in [0, \gamma] \]
EPNet
\[ E = E(F_S) \oplus E(F_D) \]

Embeddings of sparse features and dense features
\[ \delta_{\text{domain}} = U_{ep}(E(F_d) \oplus (\otimes(E))) \]

\[ O_{ep} = \delta_{\text{domain}} \otimes E \]
PEPNet Details

PPNet

\[ O_{\text{prior}} = E(u_f) \oplus E(i_f) \oplus E(a_f) \]
\[ \delta_{\text{task}} = U_{pp}(O_{\text{prior}} \oplus (\otimes(O_{cp}))) \]
\[ O_{pp}^{(l)} = \delta_{\text{task}}^{(l)} \otimes H^{(l)} \]
\[ H^{(l+1)} = f(O_{pp}^{(l)} W^{(l)} + b^{(l)}), l \in \{1, ..., L\} \]
➢ Motivation
  • Less attention has been drawn to advertisers
  • Major e-commerce platforms provide multiple marketing scenarios.

➢ Methods
  • Meta unit
  • Meta attention module
  • Meta tower module
M2M Overview

Poisson Based Multi-Task Learning

Meta Learning Mechanism

Scenario Attributes
- Scenario Knowledge
- Advertiser Profile

Meta Tower
- Meta Attention
  - load w,b

Task 1 Output

Task 2 Output

Meta Tower
- Meta Attention
  - load w,b

Meta Unit

Backbone Network

Expert View1
- Expert View2
- ... Expert Viewk

Task1 View

Task2 View

Dense Features
- Transformer Layer
- Shared Bottom Embedding
- Multi-type Behavior & Performance Sequences

Task Encoding
M2M Details

**Backbone Network**

**Expert View Representation**

\[ E_i = f_{MLP}(F), \forall i \in 1, 2, \ldots, k \]

\( F \) is the output of transformer layer

**Task View Representation**

\[ T_t = f_{MLP}(Embedding(t)), \forall t \in 1, 2, \ldots, m \]

**Scenario Knowledge Representation**

\[ \hat{S} = f_{MLP}(S, \Lambda) \]
Meta Unit

\[ h_{output} = h^K = Meta(h_{input}) \]
M2M Details

**Meta Attention Module**

\[ a_{t_i} = v^T \text{Meta}_i([E_i \parallel T_i]) \]

\[ \alpha_{t_i} = \frac{\exp(a_{t_i})}{\sum_{j=1}^{M} \exp(a_{t_j})}, \quad R_t = \sum_{i=1}^{k} \alpha_{t_i} E_i \]

**Meta Tower Module**

\[ L_t^{(0)} = R_t \]

\[ L_t^{(j)} = \sigma(Meta^{(j-1)}(L_t^{(j-1)}) + L_t^{(j-1)}), \forall j \in 1, 2, .., L \]
### Multi-Scenario Recommendation

<table>
<thead>
<tr>
<th>Model</th>
<th>Setting</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAR</td>
<td>Multi-Scenario</td>
<td>Shared-Specific</td>
</tr>
<tr>
<td>SAR-Net</td>
<td>Multi-Scenario</td>
<td>Shared-Specific; Experts</td>
</tr>
<tr>
<td>ADI</td>
<td>Multi-Scenario</td>
<td>Shared-Specific</td>
</tr>
<tr>
<td>MUSENET</td>
<td>Multi-Scenario</td>
<td>Dynamic Weight</td>
</tr>
<tr>
<td>SASS</td>
<td>Multi-Scenario</td>
<td>Dynamic Weight</td>
</tr>
<tr>
<td>MTMS</td>
<td>Multi-Scenario &amp; Multi-Task</td>
<td>Two-stage fine-tune</td>
</tr>
<tr>
<td>PEPNet</td>
<td>Multi-Scenario &amp; Multi-Task</td>
<td>Dynamic Weight</td>
</tr>
<tr>
<td>M2M</td>
<td>Multi-Scenario &amp; Multi-Task</td>
<td>Dynamic Weight; Experts</td>
</tr>
</tbody>
</table>
Multi-Scenario Recommendation

<table>
<thead>
<tr>
<th>Topic</th>
<th>Challenge &amp; future direction</th>
</tr>
</thead>
</table>
| LLM-based multi-scenario & multi-task modeling | • Design specific prompts for each scenario or tasks  
• Take the texts to bridge different scenarios or tasks |
| Robustness                                      | • Scenarios with different available information (multimodal ... )                          |
| Privacy                                         | • Data need to be shared between different scenarios to build a unified model. Methods to protect user privacy should be proposed. |
| Fairness and Bias                               | • The issue of fairness in recommendation scenarios.                                         |
Joint Modeling in Recommendations

Coffee Break

Huawei Noah’s Ark Lab

IJCAI23 Huawei Noah’s Ark Lab Chat Group

Xiangyu Zhao
City University of Hong Kong
Agenda

Introduction
- Xiangyu Zhao

Multi-task Recommendation
- Yuhaao Wang

More Joint-learning Methods
- Jingtong Gao

Preliminary
- Yichao Wang

Multi-scenario recommendation
- MTR+MSR
- Pengyue Jia

Conclusion
- Xiangyu Zhao

Future Work
More Joint-Learning Methods

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]

\[ E^{\text{Merge}} = U(E, E^{\text{Ext}}, E^m) \]

\[ E^{\text{Ext}} = F(H^{UB}) \]

\[ H^{UB} = G(H_1, H_2, \ldots, H_N) \]

\[ E^m = M(E^{\text{txt}}, E^v, \ldots, E^p) \]

\[ wL(E^{\text{Merge}}, \Theta, \Theta^t, \Theta^s) \]
More Joint-Learning Methods

- Multi-modal recommendation
- Multi-interest recommendation
- Multi-behavior recommendation
- Large language model-based recommendation

Multi-modal recommendation

Multi-interest recommendation

Large language model-based recommendation
Multimodal Recommender Systems (MRS)

- **Multi-modal recommendation**
  - Interaction
  - Image
  - Text

- **Multi-interest recommendation**
  - Interest 1
  - Interest 2
  - Interest 3

- **Multi-behavior recommendation**
  - Browse
  - Click
  - Purchase

- **Large language model-based recommendation**
  - LLM
  - RS model
Multimodal Recommender Systems (MRS)

- Using various types of information generated by multimedia applications and services to enhance recommender systems’ performance
- Making use of multimodal features simultaneously, such as image, audio, and text
- Challenge:
  - Acquisition of different representations -> Modality Encoder
  - Fusion of different modality features -> Feature Interaction
  - Acquisition of representations under the data-sparse condition -> Feature Enhancement
  - Effectiveness and efficiency improvement -> Model Optimization

Multi-modal recommendation

Modality Encoder

- Encoding different multimodal features
- Commonly used:
  - Visual: CNN-based, ViT / Transformer-based
  - Textual: Word2Vec, CNN-based, RNN-based, Transformer-based
  - Others: E.g., converting acoustic and video data into text or visual information

<table>
<thead>
<tr>
<th>Modality</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Encoder</td>
<td>CNN, ResNet, Transformer</td>
</tr>
<tr>
<td>Textual Encoder</td>
<td>Word2Vec, RNN, CNN, Sentence-transformer, Bert</td>
</tr>
<tr>
<td>Other Modality Encoder</td>
<td>Published Feature</td>
</tr>
</tbody>
</table>

Example: Multimodal encoder in VLSNR: Clip+ViT

Feature Interaction

➢ Connecting different modalities to enhance the model performance
➢ Three mainly used types: Bridge, Fusion, and Filtration
➢ These methods are combined and used together in some research
Feature Interaction: Bridge

- The construction of a multimodal information transfer channel
- Capturing the inter-relationship between users and items
- Form: User-item Graph, Item-item Graph, Knowledge Graph

Example: item-item graph in MICRO

Latent structure mining with contrastive modality fusion for multimedia recommendation. TKDE 2022.
Feature Interaction: Fusion

- Aiming at combining various preferences in modalities
- Concerning more about the multimodal intrarelationships of items
- The attention mechanism is the most widely used feature fusion method

Example: Noninvasive feature fusion in NOVA

Noninvasive self-attention for side information fusion in sequential recommendation. AAAI 2021.
Aiming at filtering out noisy data (data that is unrelated to user preferences)

This step could be done for modality features, or the feature interactions

Example: interaction denoising with an active attention mechanism in PMGCRN

Feature Enhancement

- Different modalities of the same object have unique and common semantic information.
- The recommendation performance and generalization of MRS can be significantly improved if the unique and common characteristics can be distinguished.
- Methods: Disentangled Representation Learning, Contrastive Learning.
MDR: multimodal disentangled recommendation

-> fuse representations that have the similar meaning
Contrastive Learning

GHMFC: contrastive learning modules with two loss functions (text2image and image2text)

- Learning similar semantic knowledge
Model Optimization

- The computational requirements are greatly increased with multimodal information
- Training strategies: End-to-end training (with pre-trained encoder), Two-step training
Multi-Behavior Modeling

Multi-modal recommendation

Multi-interest recommendation

Large language model-based recommendation
Multi-Behavior Modeling

➢ Understanding behavior patterns and behavior correlations at a fine-grained granularity
➢ Explicitly considering the different behavior types as they convey subtle differences in user interest modeling
Behavior Type Definition

➢ An open question

➢ Roughly three categories:
  ▪ Macro behaviors: interaction with different items
    E.g. user 1 interact with item 1, then item 22, then item 81.
  ▪ Micro behaviors: actions taken on this item
    E.g. click, add to cart, ...
  ▪ Behaviors from different domains or scenarios
    E.g. Same behavior in two domains => different behaviors (highlight the distinctions)
Behavior Type Definition

➢ Macro behaviors:

➢ Micro behaviors:

➢ Behaviors from different domains or scenarios

E.g. Same behavior in two domains => different behaviors (highlight the distinctions)

Multi-Behavior Fusion

- Modeling the complicated cross-scenario behavior dependencies

Example: pre-training and fine-tuning of ZEUS
Multi-Behavior Fusion

➢ Modeling the complicated cross-type behavior dependencies

Example: MB-GMN

Multi-behavior sequential transformer recommender. SIGIR 2022.
Multi-Interest Recommendation

➢ Information cocoon: When a user clicks and buys an item, the platform will only recommend items that are very similar

➢ Multi-Interest Recommendation: Improving the diversity and discovery of recommendations to better meet user interests
Mining interests: Interest Capsules (clustering)

for item i and interest j:

\[ b_{ij} = \overrightarrow{u}_j^T \overrightarrow{S} \overrightarrow{e}_i \]

\[ b_{ij} = (\overrightarrow{c}_j^h)^T S_{ij} \overrightarrow{c}_i^l \]

\[ w_{ij} = \frac{\exp b_{ij}}{\sum_{k=1}^{m} \exp b_{ik}} \]

\[ \overrightarrow{z}_j^h = \sum_{i=1}^{m} w_{ij} S_{ij} \overrightarrow{c}_i^l \]

\[ \overrightarrow{c}_j^h = \text{squash}(\overrightarrow{z}_j^h) = \frac{\|\overrightarrow{z}_j^h\|^2}{1 + \|\overrightarrow{z}_j^h\|^2} \]
ComiRec

- Mining interests: Interest Capsules (clustering)
- Balancing the accuracy and diversity of the recommendation

Each interest embedding can independently retrieve top-N items based on the inner production proximity. Total N*Interest candidates.

**Algorithm 2: Greedy Inference**

| Input: Candidate item set $M$, number of output items $N$ |
| Output: Output item set $S$ |

1. $S = \emptyset$
2. for $\text{iter} = 1, \ldots, N$ do
3. \[
    j = \arg\max_{i \in M \setminus S} (f(u, i) + \lambda \sum_{k \in S} g(i, k))
\]
4. $S = S \cup \{j\}$
5. return $S$
Sparse interests: activating different concepts for different input
Making prediction based on the user intention and activated concepts
LLM-based Recommendation

Multi-modal recommendation
- Interaction
- Image
- Text

Multi-interest recommendation
- Interest 1
- Interest 2
- Interest 3

Multi-behavior recommendation
- Browse
- Click
- Purchase

Large language model-based recommendation
- LLM
- RS model
LLM-based Recommendation

➢ Large language model-based recommendation

➢ Two methods:
  • Fine-tuning
  • LLM as a submodule

Large language model-based recommendation
P5: a unified recommendation model with pre-trained LLM model T5

Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). RecSys 2022.
➢ P5: a unified recommendation model with pre-trained LLM model T5
➢ Fine-tuning with five commonly used tasks

Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). RecSys 2022.
KAR: using LLM as a submodule to obtain more general knowledge

Knowledge Encoder: NLP-based encoder. E.g. BERT

More extensive joint modeling (Multi Behavior/Interest/Modal, LLM)

- Fusing heterogeneous information from different data modalities
- Acquiring multi-aspect user preferences from different type of behaviors or interests
- Introducing open-world knowledge from large language models

<table>
<thead>
<tr>
<th>Models</th>
<th>Type (or other dimension)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB-GMN</td>
<td>Multi-Behavior</td>
</tr>
<tr>
<td>RIB</td>
<td>Multi-Behavior</td>
</tr>
<tr>
<td>ZEUS</td>
<td>Multi-Behavior</td>
</tr>
<tr>
<td>MIND</td>
<td>Multi-Interest</td>
</tr>
<tr>
<td>ComiRec</td>
<td>Multi-Interest</td>
</tr>
<tr>
<td>SINE</td>
<td>Multi-Interest</td>
</tr>
<tr>
<td>P5</td>
<td>LLM-Based</td>
</tr>
<tr>
<td>KAR</td>
<td>LLM-Based</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>Type (or other dimension)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLSNR</td>
<td>Multi-Modal</td>
</tr>
<tr>
<td>MICRO</td>
<td>Multi-Modal</td>
</tr>
<tr>
<td>NOVA</td>
<td>Multi-Modal</td>
</tr>
<tr>
<td>PMGCRN</td>
<td>Multi-Modal</td>
</tr>
<tr>
<td>MDR</td>
<td>Multi-Modal</td>
</tr>
<tr>
<td>GHMFC</td>
<td>Multi-Modal</td>
</tr>
</tbody>
</table>
Future Directions

➢ More extensive joint modeling

• Joint modeling with all the above methods
• A more comprehensive approach to realize joint modeling with LLM
Agenda

Introduction

Xiangyu Zhao

Multi-task Recommendation

Yuhaoo Wang

Multi-scenario recommendation

MTR+MSR

Pengyue Jia

More Joint-learning Methods

Jingtong Gao

Conclusion

Future Work

Xiangyu Zhao
Conclusion

➢ Utilizing diverse user feedback signals from **different tasks**

➢ Extracting commonalities and diversities of user preferences from **different scenarios**

➢ Fusing heterogeneous information from different **data modalities**

➢ Acquiring multi-aspect user preferences from different type of **behaviors or interests**

➢ Introducing open-world knowledge from **large language models**
Multi-Task Recommendation

- Task relation:
  Parallel, Cascaded, Auxiliary with Main

- Methodology:
  Parameter Sharing, Optimization, Training Mechanism
Multi-Scenario Recommendation

- From the perspective of methods, there are mainly two categories: shared-specific network paradigm, and dynamic weight paradigm.

- Overall, most the work focuses on using one unified model serving multiple scenarios and multiple tasks simultaneously based on knowledge transfer between scenarios or tasks.
More extensive joint modeling (Multi Behavior/Interest/Modal)

- Multi Behavior/Interest/Modal modeling are joint learning methods focusing on fine-grained modeling of different user/model’s aspects.

- LLM, as a new effective method for recommendation, could further be combined with recommendation models to jointly learn more universal knowledge to obtain a better performance.
Future Directions

➢ Multi-Task Recommendation
  • Negative transfer
  • Task-specific biases

➢ Multi-Scenario Recommendation
  • Robustness
  • Privacy

➢ More extensive joint modeling
  • A more comprehensive approach to realize joint modeling with LLM

➢ Ecosystem
  • Joint modeling with all the above methods
  • More convenient for other researchers to contribute to this field
https://arxiv.org/abs/2302.03525