AutoLoss: Automated Loss Function Search in Recommendations

Xiangyu Zhao, Haochen Liu, Wenqi Fan
Hui Liu, Jiliang Tang, Chong Wang

1: City University of Hong Kong, 2: Michigan State University
3: The Hong Kong Polytechnic University, 4: Bytedance
Deep Recommender Systems

- Advantages
  - Feature representations of users and items
  - Non-linear relationships between users and items

- Typical architecture
  - Embedding layer
  - Inference layer
Deep Recommender Systems

- Advantages
  - Feature representations of users and items
  - Non-linear relationships between users and items

- Typical architecture
  - Embedding layer
  - Inference layer

- Well-designed loss functions
  - Item rating prediction (regression)
  - CTR prediction (binary classification)
Existing Optimization Methods

- Predefined and fixed loss functions
  - E.g., MAE or MSE loss for regression tasks
Existing Optimization Methods

- Predefined and fixed loss functions
  - E.g., MAE or MSE loss for regression tasks

- The gradients generated from a given loss function are optimal?

![MAE vs Predictions](chart1.png)

![MSE vs Predictions](chart2.png)
Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time
  - Unified and static loss weights → Overlooking the different convergence behaviors

Existing Optimization Methods

- Fusing multiple loss functions in a weighted sum manner
  - E.g., Panoptic FPN leverages a grid search to find better loss weights [1]
  - E.g., UPSNet manually investigates the weights of loss functions [2]

- Disadvantages
  - Exhaustively or manually searching for loss weights → Costly in computation and time
  - Unified and static loss weights → Overlooking the different convergence behaviors
  - Retraining loss weights is always desired → Bad generalizability and transferability

AutoLoss Framework
AutoLoss Framework

- Forward-propagation step
  - Generating predictions
AutoLoss Framework

- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses
AutoLoss Framework

- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses
  - Calculating probabilities
AutoLoss Framework

- Forward-propagation step
  - Generating predictions
  - Calculating candidate losses
  - Calculating probabilities
  - Calculating the overall loss
**AutoLoss Framework**

- Backward-propagation step
  - Updating the main DRS network parameters upon the *training* data examples
## AutoLoss Framework

- **Backward-propagation step**
  - Updating the main DRS network parameters upon the **training** data examples
  - Optimizing the controller network parameters based on **validation** data examples
Experimental Settings

- Two recommendation datasets/tasks
  - Criteo: binary classification
  - ML-20m: multiclass classification

- Two deep recommendation models
  - DeepFM and IPNN

<table>
<thead>
<tr>
<th>Data</th>
<th>Criteo</th>
<th>ML-20m</th>
</tr>
</thead>
<tbody>
<tr>
<td># Interactions</td>
<td>45,840,617</td>
<td>20,000,263</td>
</tr>
<tr>
<td># Feature Fields</td>
<td>39</td>
<td>2</td>
</tr>
<tr>
<td># Feature Values</td>
<td>1,086,810</td>
<td>165,771</td>
</tr>
<tr>
<td># Behavior</td>
<td>click or not</td>
<td>rating 1~5</td>
</tr>
</tbody>
</table>
### Overall Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Metric</th>
<th>Focal</th>
<th>KL</th>
<th>Hinge</th>
<th>CE</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteo</td>
<td>DeepFM</td>
<td>AUC↑, Logloss↓</td>
<td>0.8046</td>
<td>0.8042</td>
<td>0.8049</td>
<td>0.8056</td>
<td>0.8063</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.4466</td>
<td>0.4469</td>
<td>0.4463</td>
<td>0.4457</td>
<td>0.4436</td>
</tr>
<tr>
<td>Criteo</td>
<td>IPNN</td>
<td>AUC↑, Logloss↓</td>
<td>0.8077</td>
<td>0.8072</td>
<td>0.8079</td>
<td>0.8085</td>
<td>0.8090</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.4435</td>
<td>0.4437</td>
<td>0.4432</td>
<td>0.4428</td>
<td>0.4423</td>
</tr>
<tr>
<td>ML-20m</td>
<td>DeepFM</td>
<td>AUC↑, Logloss↓</td>
<td>0.7681</td>
<td>0.7682</td>
<td>0.7685</td>
<td>0.7692</td>
<td>0.7695</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.2320</td>
<td>1.2317</td>
<td>1.2316</td>
<td>1.2310</td>
<td>1.2307</td>
</tr>
<tr>
<td>ML-20m</td>
<td>IPNN</td>
<td>AUC↑, Logloss↓</td>
<td>0.7721</td>
<td>0.7722</td>
<td>0.7725</td>
<td>0.7733</td>
<td>0.7735</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.2270</td>
<td>1.2269</td>
<td>1.2266</td>
<td>1.2260</td>
<td>1.2256</td>
</tr>
</tbody>
</table>

- **Fixed loss function:** Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
## Overall Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Metric</th>
<th>Focal</th>
<th>KL</th>
<th>Hinge</th>
<th>CE</th>
<th>MeLU</th>
<th>BOHB</th>
<th>DARTS</th>
<th>SLF</th>
<th>AutoLoss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteo</td>
<td>DeepFM</td>
<td>AUC ↑ Logoss ↓</td>
<td>0.8046</td>
<td>0.8042</td>
<td>0.8049</td>
<td>0.8056</td>
<td>0.8063</td>
<td>0.8065</td>
<td>0.8067</td>
<td>0.8081</td>
<td><strong>0.8092</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.4466</td>
<td>0.4469</td>
<td>0.4463</td>
<td>0.4457</td>
<td>0.4436</td>
<td>0.4435</td>
<td>0.4433</td>
<td>0.4426</td>
<td><strong>0.4416</strong></td>
</tr>
<tr>
<td>Criteo</td>
<td>IPNN</td>
<td>AUC ↑ Logoss ↓</td>
<td>0.8077</td>
<td>0.8072</td>
<td>0.8079</td>
<td>0.8085</td>
<td>0.8090</td>
<td>0.8092</td>
<td>0.8093</td>
<td>0.8098</td>
<td><strong>0.8108</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.4435</td>
<td>0.4437</td>
<td>0.4432</td>
<td>0.4428</td>
<td>0.4423</td>
<td>0.4422</td>
<td>0.4423</td>
<td>0.4418</td>
<td><strong>0.4409</strong></td>
</tr>
<tr>
<td>ML-20m</td>
<td>DeepFM</td>
<td>AUC ↑ Logoss ↓</td>
<td>0.7681</td>
<td>0.7682</td>
<td>0.7685</td>
<td>0.7692</td>
<td>0.7695</td>
<td>0.7695</td>
<td>0.7696</td>
<td>0.7705</td>
<td><strong>0.7717</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.2320</td>
<td>1.2317</td>
<td>1.2316</td>
<td>1.2310</td>
<td>1.2307</td>
<td>1.2305</td>
<td>1.2305</td>
<td>1.2299</td>
<td><strong>1.2288</strong></td>
</tr>
<tr>
<td>ML-20m</td>
<td>IPNN</td>
<td>AUC ↑ Logoss ↓</td>
<td>0.7721</td>
<td>0.7722</td>
<td>0.7725</td>
<td>0.7733</td>
<td>0.7735</td>
<td>0.7734</td>
<td>0.7736</td>
<td>0.7745</td>
<td><strong>0.7756</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.2270</td>
<td>1.2269</td>
<td>1.2266</td>
<td>1.2260</td>
<td>1.2256</td>
<td>1.2257</td>
<td>1.2255</td>
<td>1.2249</td>
<td><strong>1.2236</strong></td>
</tr>
</tbody>
</table>

- **Fixed loss function**: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- **Fixed weights over loss functions**: MeLU (meta-learning), BOHB and DARTS (AutoML)
## Overall Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Metric</th>
<th>Focal</th>
<th>KL</th>
<th>Hinge</th>
<th>CE</th>
<th>MeLU</th>
<th>BOHB</th>
<th>DARTS</th>
<th>SLF</th>
<th>AutoLoss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteo</td>
<td>DeepFM</td>
<td>AUC ↑</td>
<td>0.8046</td>
<td>0.8042</td>
<td>0.8049</td>
<td>0.8056</td>
<td>0.8063</td>
<td>0.8065</td>
<td>0.8067</td>
<td>0.8081</td>
<td>0.8092*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logloss ↓</td>
<td>0.4466</td>
<td>0.4469</td>
<td>0.4463</td>
<td>0.4457</td>
<td>0.4436</td>
<td>0.4435</td>
<td>0.4433</td>
<td>0.4426</td>
<td>0.4416*</td>
</tr>
<tr>
<td>Criteo</td>
<td>IPNN</td>
<td>AUC ↑</td>
<td>0.8077</td>
<td>0.8072</td>
<td>0.8079</td>
<td>0.8085</td>
<td>0.8090</td>
<td>0.8092</td>
<td>0.8093</td>
<td>0.8098</td>
<td>0.8108*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logloss ↓</td>
<td>0.4435</td>
<td>0.4437</td>
<td>0.4432</td>
<td>0.4428</td>
<td>0.4423</td>
<td>0.4422</td>
<td>0.4423</td>
<td>0.4418</td>
<td>0.4409*</td>
</tr>
<tr>
<td>ML-20m</td>
<td>DeepFM</td>
<td>AUC ↑</td>
<td>0.7681</td>
<td>0.7682</td>
<td>0.7685</td>
<td>0.7692</td>
<td>0.7695</td>
<td>0.7695</td>
<td>0.7696</td>
<td>0.7705</td>
<td>0.7717*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logloss ↓</td>
<td>1.2320</td>
<td>1.2317</td>
<td>1.2316</td>
<td>1.2310</td>
<td>1.2307</td>
<td>1.2305</td>
<td>1.2305</td>
<td>1.2299</td>
<td>1.2288*</td>
</tr>
<tr>
<td>ML-20m</td>
<td>IPNN</td>
<td>AUC ↑</td>
<td>0.7721</td>
<td>0.7722</td>
<td>0.7725</td>
<td>0.7733</td>
<td>0.7735</td>
<td>0.7734</td>
<td>0.7736</td>
<td>0.7745</td>
<td>0.7756*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logloss ↓</td>
<td>1.2270</td>
<td>1.2269</td>
<td>1.2266</td>
<td>1.2260</td>
<td>1.2256</td>
<td>1.2257</td>
<td>1.2255</td>
<td>1.2249</td>
<td>1.2236*</td>
</tr>
</tbody>
</table>

- **Fixed loss function**: Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- **Fixed weights over loss functions**: MeLU (meta-learning), BOHB and DARTS (AutoML)
- **Data example-wise loss weights**: SLF (stochastic loss function)
## Overall Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Metric</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Focal KL Hinge CE MeLU BOHB DARTS SLF AutoLoss</td>
</tr>
<tr>
<td>Criteo</td>
<td>DeepFM</td>
<td>AUC↑ Logloss↓</td>
<td>0.8046 0.8042 0.8049 0.8056 0.8063 0.8065 0.8067 0.8081 0.8092* 0.4416*</td>
</tr>
<tr>
<td>Criteo</td>
<td>IPNN</td>
<td>AUC↑ Logloss↓</td>
<td>0.8077 0.8079 0.8085 0.8090 0.8092 0.8093 0.8098 0.8108* 0.4409*</td>
</tr>
<tr>
<td>ML-20m</td>
<td>DeepFM</td>
<td>AUC↑ Logloss↓</td>
<td>0.7681 0.7682 0.7685 0.7692 0.7695 0.7695 0.7696 0.7705 0.7717* 1.2288*</td>
</tr>
<tr>
<td>ML-20m</td>
<td>IPNN</td>
<td>AUC↑ Logloss↓</td>
<td>0.7721 0.7722 0.7725 0.7733 0.7735 0.7734 0.7736 0.7745 0.7756* 1.2236*</td>
</tr>
</tbody>
</table>

- **Fixed loss function:** Focal loss, KL divergence, Hinge loss and cross-entropy (CE) loss for both classification tasks
- **Fixed weights over loss functions:** MeLU (meta-learning), BOHB and DARTS (AutoML)
- **Data example-wise loss weights:** SLF (stochastic loss function)
Transferability Study

- **Transferability among DRS models**
  - DeepFM → NFM and AutoInt
  - *CE*: cross-entropy loss
  - *SLF*: SLF controller from DeepFM
  - *AL*: AutoLoss controller from DeepFM

- **Transferability among datasets**
  - Criteo → Avazu
  - *CE*: cross-entropy loss on Avazu
  - *SLF*: SLF controller from Criteo
  - *AL*: AutoLoss controller from Criteo
Transferability Study

- **Transferability among DRS models**
  - **DeepFM** → **NFM and AutoInt**
  - **CE**: cross-entropy loss
  - **SLF**: SLF controller from DeepFM
  - **AL**: AutoLoss controller from DeepFM

- **Transferability among datasets**
  - **Criteo** → **Avazu**
  - **CE**: cross-entropy loss on Avazu
  - **SLF**: SLF controller from Criteo
  - **AL**: AutoLoss controller from Criteo
Efficiency Study

- Fastest training speed
  - AutoLoss can generate the most appropriate gradients to update DRS, which increases the optimization efficiency
  - We update the controller once after every 7 times DRS is updated, which not only reduces the training time (~ 60%) with fewer computations, but also enhances the performance
Conclusion

- We propose an end-to-end framework, AutoLoss, which can automatically select the proper loss functions for training DRS frameworks
  - Better recommendation performance and training efficiency

- A novel controller network is developed to adaptively adjust the probabilities over multiple loss functions according to different data examples’ convergence behaviors
  - Enhancing the model generalizability between different DRS frameworks and datasets