Framework and Principles of Matching Technologies

Hang Li
Bytedance Technology
This talk gives a high-level review of matching technologies in search and recommendation
Outline of Talk

• Matching Problem
• Framework and Principles of Matching
• State-of-the-Art Techniques for Matching
• Summary
Matching Problem

Training Data

\((x_1, y_1, r_1)\)
\((x_2, y_2, r_2)\)
\(\ldots\)
\((x_n, y_n, r_n)\)

Learning System

Learning to Match

\[ r = f(x, y) \]

Prediction System

Test Data

\((x_{n+1}, y_{n+1})\)

Matching Model

\(r_{n+1}\)
Matching vs Classification and Regression

- Matching model: $f(x, y)$
- Classification and regression models: $f(x)$
- Matching can be viewed as special case of classification and regression
- But, there are also differences
- Features need to be carefully designed to represent the interactions between inputs $x$ and $y$
Matching and Ranking

• Matching model: $f(x, y)$
• Ranking model: $g(x, y)$
• In search and recommendation:
  • Matching models can be features of ranking model
  • Ranking model is more ‘content-agnostic’ than matching models, its features = BM25, PageRank
• Sometimes, matching model and ranking model are combined and trained together with pairwise loss
Learning to Rank

- Pointwise loss: $L(f(x, y), r)$
- Pairwise loss: $L(f(x, y_1), f(x, y_2), r_1, r_2)$
- Listwise loss:
  $L(f(x, y_1), f(x, y_2), \cdots f(x, y_m), r_1, r_2, \cdots r_m)$

- Pairwise approach and listwise approach work better than pointwise approach
- Pairwise approach is more widely used
- Sometimes listwise approach works best
Text Matching and Entity Matching

• Matching between two sets of objects
• Text matching
  – Order exists between objects in each set (i.e., words in each sentence)
  – E.g., query title matching in search
• Entity matching
  – No order exists between objects in each set
  – E.g., user item matching in recommendation
Matching in Search

• Text matching: query-title matching
• Lexical matching is more important
• Asymmetric matching: query to title (document)
• Query can consist of multiple phrases (i.e., partial order)
• Query term importance may need to be considered
• E.g., “talk geoffrey hinton deep learning” → “Prof. Hinton’s Lecture at University of Toronto on Deep Learning”
Matching in Question Answering

- Text matching: question-answer matching
- Semantic matching is more important
- Asymmetric matching: question to answer
- E.g., “how far is sun from earth” $\rightarrow$ “distance between sun and earth”
Matching in Paraphrasing

• Text matching: sentence-sentence matching
• Semantic matching is more important
• Symmetric matching: text to text
• E.g., “Harry Potter 4”, v.s. “Harry Potter and the Goblet of Fire”
• E.g., “Harry Potter 4”, v.s. “Harry Potter 5”
Matching in Recommendation

• Entity matching: user-item matching
• Interactions (similarities) between entities are useful information
• Data is sparse
• Hidden structure of interactions (obtained via matrix factorization) is powerful
Natural Language Processing Problems

• Classification: $x \rightarrow c$
• Matching: $x, y \rightarrow \mathcal{R}$
• Sequence-to-Sequence: $x \rightarrow y$
• Structured Prediction: $x \rightarrow [x]$
• Sequential Decision Process: $\pi: s \rightarrow a$

Li 2017
Natural Language Problems

- **Classification**
  - Text classification
  - Sentiment analysis

- **Matching**
  - Search
  - Question answering
  - Single-turn dialogue (retrieval)

- **Sequence to Sequence**
  - Machine translation
  - Summarization
  - Single-turn dialogue (generation)

- **Structured Prediction**
  - Sequential labeling
  - Semantic parsing

- **Sequential Decision Process**
  - Multi-turn dialogue
Outline of Talk

• Matching Problem
• *Framework and Principles of Matching*
• State-of-the-Art Techniques for Matching
• Summary
Overview of Matching

• Deep learning (neural networks) is state-of-the-art in search and recommendation
• Different network architectures are needed for different tasks
• There are general framework and principles
Deep Learning

Mimicking human behaviors using deep learning tools

\[ y = f(x) \]

\[ \max_f P_f(y|x) \]
Deep Learning Techniques

• Models and Tools
  – Feedforward Neural Network
  – Convolutional Neural Network
  – Recurrent Neural Network
  – Sequence-to-Sequence Model
  – Attention
  – …..

• Learning algorithm: back propagation
• Regularization, e.g., dropout, early-stopping
Framework of Matching

- Output: MLP
- Aggregation: Pooling, Concatenation
- Interaction: Matrix, Tensor
- Representation: MLP, CNN, LSTM
- Input: ID Vectors, Feature Vectors
Typical Architecture for Search and Question Answering

- Input: two sequences of word embeddings
- First, create *semantic* representations of two inputs
- Next, make interaction between the two representations
- Finally, make aggregation
• Input: two sequences of word embeddings
• First, make *lexical* interaction between two inputs
• Next, make aggregation of interaction
Typical Architecture for Recommendation

- **Factorization Machine**
- **Embeddings**
- **Representation**
- **Interaction (2nd Order)**
- **Interaction (1st Order)**
- **Aggregation**

Input: \( X \) and \( Y \)
Output: Match
Typical Architecture for Recommendation

• Input: two vectors are combined
• First, create embeddings of combined inputs
• Next, make interactions using factorization machine (1\(^{st}\) order feature interaction and 2\(^{nd}\) order feature interaction)
• Finally, make aggregation of interactions
Two Principles

• Modular Principle: System consists of different modules (functions) implemented with different techniques
  – Representation: CNN, RNN, MLP
  – Interaction: matrix, tensor
  – Aggregation: pooling, concatenation

• Hybrid Principle: Combination of dichotomic techniques may be necessary
  – Deep model and wide model
  – Nonlinear model and linear model
  – Factorization and non-factorization (2nd order interaction and 1st order interaction)
Outline of Talk

• Matching Problem
• Framework and Principles of Matching
• *State-of-the-Art Techniques for Matching*
• Summary
Search: DSSM

Posterior probability computed by softmax
Relevance measured by cosine similarity

Semantic feature
Multi-layer non-linear projection
Word Hashing
Term Vector

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_i$</td>
<td>30k</td>
</tr>
<tr>
<td>$l_2$</td>
<td>300</td>
</tr>
<tr>
<td>$l_3$</td>
<td>300</td>
</tr>
<tr>
<td>$y$</td>
<td>128</td>
</tr>
</tbody>
</table>

$W_i$, $W_2$, $W_3$, $W_4$, $b_2$, $b_3$, $b_4$

$x$, $Q$, $D_1$, $D_2$, $D_n$

$P(D_i|Q)$, $R(Q, D_i)$

Huang et al. CIKM 2013
Search: DSSM

- Input: two vectors of letter n-grams
- Representations: two vectors created by MLP
- Interaction: cos between two vectors
- Alternatives: representations created by using CNN, RNN
Question Answering: Arc II

Hu at al. NIPS 2014
Question Answering: Arc II

- Input: two sequences of word embeddings
- Interaction: matrix created by 1-D CNN
- Aggregation: vector created by 2-D CNN
- Output: value generated by MLP
Search: DRMM

Score Aggregation

Feed Forward Matching Network

Matching Histogram Mapping

Local Interaction

Matching Score

Term Gating Network

Guo at al. CIKM 2016
Search: DRMM

- **Input**: two sequences of word embeddings
- **Interaction**: lexical interaction matrix, asymmetric
- **Aggregation**: weighted sum created by MLP
- **Attention**: query term weighting
- **Alternative**: aggregation by kernel pooling or max pooling
Recommendation: NeuMF
Recommendation: NeuMF

- **Representation**
  - Embedding
  - Element-wise Product

- **Interaction (Factorization)**
  - Value
  - Sigmoid

- **Interaction**
  - Concatenation
  - MLP

- **Aggregation**
  - Match

- **Vector**
Recommendation: NeuMF

• Input
  – Combined user ID vector and item ID vector

• Representation
  – Two vectors (embeddings) for factorization and for neural network respectively

• Interaction
  – Two vectors obtained by factorization and neural network

• Aggregation
  – Value generated by concatenation and sigmoid function
Recommendation: DeepFM

![Diagram of DeepFM](image-url)

Guo at al. IJCAI 2017
Recommendation: DeepFM

- Interaction (Factorization)
- Interaction
- Aggregation
- Match
- Vector
- Concatenation Sigmoid
- MLP
- Vector
- Factorization Machine
- Vector
- Sigmoid
- MLP
- Embedding

X
Y
Recommendation: DeepFM

• Input
  – Combined user feature vector and item feature vector

• Representation
  – Two shared vectors (embeddings) for factorization machine and neural network

• Interaction
  – Two vectors by factorization machine and neural network

• Aggregation
  – Value generated by concatenation and sigmoid function
Recommendation: NFM

He at al. SIGIR 2017
Recommendation: NFM

- Aggregation
  - Match
  - Value
  - Value
  - Linear
  - Values
- Interaction (Factorization)
  - Factorization Machine + MLP
  - Vector
  - Linear model
- Representation
  - Embedding
- Interaction
  - Linear model

X
Y
Recommendation: NFM

• **Input**
  – Combined user feature vector and item feature vector

• **Representation**
  – Vector (embedding) from combined vectors

• **Interaction**
  – Vector by factorization machine plus neural network, as well as values by linear model

• **Aggregation**
  – Value generated by linear combination
Outline of Talk

• Matching Problem
• Framework and Principles of Matching
• State-of-the-Art Techniques for Matching
• Summary
Summary

• Matching is key technology for search and recommendation
• Text matching and entity matching
• Deep learning is state-of-the-art
• Framework: input, representation, interaction, aggregation, output
• Principles: modular and hybrid
Acknowledgement

I thank Jun Xu, Xiangnan He, Chao Qiao, Shengxian Wan for valuable discussions with them on matching technologies.
References

• Jun Xu, Xiangnan He, Hang Li, Deep Learning for Matching in Search and Recommendation, WSDM 2019 Tutorial

Thank you!

lihang.lh@bytedance.com