Federated Recommendation

Qiang Yang
2019/12/10
Recommender Systems Have Been Widely Used

<table>
<thead>
<tr>
<th>E-commerce</th>
<th>Online Video</th>
<th>Social Network</th>
<th>News Feeds</th>
<th>Online Advertising</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>YouTube</td>
<td>Facebook</td>
<td>Touxing News</td>
<td>Google</td>
</tr>
<tr>
<td>Taobao</td>
<td>Tencent Video</td>
<td>LinkedIn</td>
<td>Baidu</td>
<td></td>
</tr>
<tr>
<td>JD.com</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Image of various platforms and logos]
Recommender Systems Improve User Engagement

YouTube Homepage: 60%+ more clicks [Davidson et al. 2010]

Netflix: 80%+ more movie watches [Gomze-Uribe et al 2016]

Amazon: 30%+ more page views [Smith and Linden, 2017]
Overview of Recommender Systems

**Input:** historical user-item interactions, and optionally additional side information (e.g., user demographic, item attributes)

**Output:** how likely a user would interact with an item (e.g., a movie, a song, a product)
More Data Used in Recommender Systems, Better Performance


Auxiliary data used
Reality in Recommender Systems: Data Silos
Facebook finally rolls out privacy tool for your browsing history

Google strengthens Chrome’s privacy controls

Top Microsoft exec says online privacy has reached 'a crisis point'
Most RecSys Uses Differential Privacy for Privacy Preservation

**Definition: Differential Privacy (DP) [Dwork 2008]**

A randomized mechanism $M$ is $\epsilon$-differentially private, if for all output $t$ of $M$, and for all databases $D_1$ and $D_2$ which differ by at most one element, we have

$$\Pr(M(D_1) = t) = e^\epsilon \Pr(M(D_2) = t).$$

**Intuition:** changes in the distribution is too small to be perceived with variations on a single element.

Differentially Private Matrix Factorization [Knijnenburg and Berkovsky, 2017]

Matrix Factorization

ALS MF Solver:
- Fix $P$: Least square optimization of $Q$
- Fix $Q$: Least square optimization of $P$

SGD MF Solver:
- for each $r_{ui}$, if $r_{ui} > 0$ compute
  - $e_{ui} = r_{ui} - (q^T p_u)$
  - and update factors:
    - $p_u = p_u + \gamma (e_{ui} q_i - \lambda p_u)$
    - $q_i = q_i + \gamma (e_{ui} p_u - \lambda q_i)$

Recommendation
- Prediction: $q_i^T p_u$

User input data
- Rating matrix $r_{ui}$

DP on data collection
- DP on gradients
- DP on model parameters
- DP on outputs
We Need New Technology for RecSys with Decentralized Data

Desired properties for new technology:

- Lossless performance in decentralized setting, compared with centralized setting.
- Increasing noise, decreasing performance
- Raw data after DP is transmitted between parties.
- Data protected in decentralized setting, with raw data staying locally.

Apply DP over data
Federated Learning to Bridge Decentralized Data

Lossless performance
- Performance of ‘A fed B’ is close to ‘A+B’

Data protected
- Raw data stays locally
- Only parameters and gradients are securely transmitted

Yang et al. 2018, Federated machine learning: concepts and applications. ACM TIST.
**Federated Recommendation**

**Assumption:** for easier understanding and system efficiency, we assume the existence of a trustworthy 3rd-party server in the following federated recommendation solution discussion.

In general, such 3rd-party servers can be removed to strengthen the data security.
Key Security Technology in Federated Recommendation

• Secure Multi-party Computation (MPC)
  • Homomorphic Encryption (HE)
  • Yao’s Garbled Circuit
• Secret sharing
• …
Secure Multi-Party Computation (MPC)

Overview of MPC:
- Provides security proof in a well-defined simulation framework
- Guarantees complete zero knowledge
- Requires participants’ data to be secretly-shared among non-colluding servers
Homomorphic Encryption

- Full Homomorphic Encryption and Partial Homomorphic Encryption.
- Paillier partially homomorphic encryption

**Addition:** \([[[u]]] + [[[v]]] = [[[u+v]]] \)

**Scalar multiplication:** \(n[[[u]]] = [[[nu]]] \)

- For public key \(pk = n\), the encoded form of \(m \in \{0, \ldots, n - 1\}\) is

\[
\text{Encode}(m) = r^n (1 + n)^m \mod n^2
\]

where \(r\) is randomly selected from \(\{0, \ldots, n - 1\}\).

- For float \(q = (s, e)\), encrypt \([[q]] = [[[s]], e] \), here \(q = s\beta^e\) is base-\(\beta\) exponential representation.

Applying Homomorphic Encryption to Machine Learning

1. Polynomial approximation for logarithm function

\[
\log \left( \frac{1}{1 + \exp(u)} \right) \approx \sum_{j=0}^{k} a_j u^j
\]

2. Encrypted computation for each term in the polynomial function

\[
\text{loss} = \log 2 - \frac{1}{2} y w^T x + \frac{1}{8} (w^T x)^2
\]

\[
[[\text{loss}]] = [[\log 2]] + \left( -\frac{1}{2} \right) [[y w^T x]] + \frac{1}{8} [[(w^T x)^2]]
\]

- Aono et al. 2016. Scalable and secure logistic regression via homomorphic encryption. CODASPY, Pages 142-144.
Is the Gradient Info Safe to Share?

Protect gradients with Homomorphic Encryption


- Algorithm ensures that no information is leaked, provided that the underlying additively homomorphic encryption scheme is secure*.

Yang et al. 2018. Federated machine learning: concepts and applications. ACM TIST.
Categorization of Federated Recommendation

Horizontal Federated Recommendation
(a.k.a. Item-based FedRec)

Large overlap of items of the two rating matrices

Vertical Federated Recommendation
(a.k.a. User-based FedRec)

Large overlap of users of the two rating matrices
Category 1: Horizontal Federated Recommendation

Large overlap of items of the two rating matrices
Horizontal Federated Recommendation: Case 1

Example: movie recommendation with data from individual users
Federated Collaborative Filtering [Ammad et al. 2019]

**Intuition:** decentralized matrix factorization, each user profile is updated locally, item profiles are aggregated and updated by server.

Loss function:
\[
\min_{U,V} \frac{1}{M} \sum_{i,j} (r_{ij} - (u_i, v_j))^2 + \lambda \|U\|_F^2 + \mu \|V\|_F^2
\]

Update function:
\[
\begin{align*}
\hat{u}_i^t &= u_i^{t-1} - \gamma \nabla_{u_i} F(U^{t-1}, V^{t-1}) \\
\hat{v}_j^t &= v_j^{t-1} - \gamma \nabla_{v_j} F(U^{t-1}, V^{t-1})
\end{align*}
\]

User local updates → Server updates

Gradients from users
Federated Collaborative Filtering [Ammad et al. 2019]

Pros: user data is decentralized.
Cons: no MPC (plaintext gradients).

Training Process:
1. Server initializes item profiles, parties initializes user profiles;
2. Sever distributes item profiles to parties;
3. Parties locally update user profiles with item profiles;
4. Parties send item profile gradient updates to server;
5. Server updates item profile.
**Horizontal Federated Matrix Factorization [Chai et al. 2019]**

**Intuition:** Item profile gradients are encrypted by HE. Semi-honest server securely aggregates encrypted item profiles gradients, and knows nothing about the profile content.

---

**Training Process:**

1. **Server initializes and encrypts item profiles:**
   - Server initializes and encrypts item profiles.

2. **Server distributes encrypted item profiles to parties:**
   - Server distributes encrypted item profiles to parties.
   - Parameter Sharing: the latest $C_V$ are kept available for all users’ download.

3. **Parties locally update user profiles with encrypted item profiles:**
   - Parties locally update user profiles with encrypted item profiles.
   - Parties send encrypted item profile gradient updates to server.

4. **Server securely aggregates item profile gradients and updates item profiles:**
   - Server securely aggregates item profile gradients and updates item profiles.

Security of secure aggregation protocol is guaranteed [Bonawitz et al. 2017].

---

Horizontal Federated Recommendation: Case 2

Example: movie recommendation with data from two different groups of users.

Solution to Case 2:
Item profiles are securely aggregated by server, group of user profiles are locally updated by parties.
Category 2: Vertical Federated Recommendation

Large overlap of users of the two rating matrices
Vertical Federated Recommendation: Case 1

Example: Shared users, different items

Party A

Party B

book recommendation

movie recommendation

No data exchange
Vertical Federated Matrix Factorization [Chai et al. 2019]

**Intuition:** User profile gradients are encrypted and securely aggregated by semi-honest server, item profiles are updated locally.
Vertical Federated Matrix Factorization [Chai et al. 2019]

Training Process:
1. Server initializes and encrypts user profiles; parties initialize item profiles.
2. Server distributes encrypted user profiles to parties;
3. Parties locally update item profiles with decrypted user profiles; Parties send encrypted user profile gradient to server;
4. Server securely aggregates user profile gradients and update user profiles.

Security of secure aggregation protocol is guaranteed [Bonawitz et al. 2017].
Vertical Federated Recommendation: Case 2

Example:
Shared users
different features

Book recommendation auxiliary data from third-parties

<table>
<thead>
<tr>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Georgia</td>
<td>2018.5</td>
</tr>
<tr>
<td>Florida</td>
<td>2019.1</td>
</tr>
<tr>
<td>Hawaii</td>
<td>2017.3</td>
</tr>
<tr>
<td>Kansas</td>
<td>2018.5</td>
</tr>
<tr>
<td>Georgia</td>
<td>2018.10</td>
</tr>
<tr>
<td>Florida</td>
<td>2019.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Party A</th>
<th>Party B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sports</td>
<td>Photography</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
Federated Factorization Machine [Zheng et al. 2019]

**Intuition:** cross-features between A and B are useful, but features are sensitive. Federated factorization machine computes these cross-party cross-features and their gradients under encryption.

**Prediction function**

\[ f([x_p^{(A)}; x_q^{(B)}]) = f(x_p^{(A)}) + f(x_q^{(B)}) + \sum \langle v_i^{(A)}, v_j^{(B)} \rangle x_{p,i}^{(A)} x_{q,j}^{(B)} \]

**Cross features between A and B** are useful; e.g., “location x sports” can be a strong indicator for predicting Georgia user’s preference to sports movies.

Federated Factorization Machine [Zheng et al. 2019]

Training Process

1. Parties initialize models
2. Party B sends encrypted partial prediction, partial loss and partial feature gradients to party A
3. Party A sends encrypted error and partial feature gradients to party B
4. Parties send encrypted and masked gradients to server
5. Server decrypts and sends back
6. Parties unmask and update models

Security of semi-honest MPC protocol is guaranteed [Goldreich et al. 1987].
Federated Factorization Machine [Zheng et al. 2019]

**Inference Process:** encrypted prediction on party A’s features + encrypted prediction on A&B features + encrypted prediction on party B’s features.

1. Party A and B compute encrypted intermediate results
2. Server aggregates the encrypted intermediate results and decrypts
3. Server sends plain-text prediction to party A
What If Different Users and Items at the Same Time?

Transfer Federated Recommendation

Data from A

Data from B

Users

Items
Category 3: Transfer Federated Recommendation

Example: movie and book recommenders with different groups of users
Matrix Tri-factorization [Li et al. 2009]

**Intuition:** similar users/items can be clustered into groups, and there exist group correspondences across parties.

**User cluster Indicator** $U$

**Item cluster Indicator** $V$

**Codebook** $\Sigma$

Li et al. Transfer Learning for Collaborative Filtering via a Rating-Matrix Generative Model, ICML, pp.617-624.
Federated Matrix Tri-factorization [Tan et al. 2019]

**Intuition:** codebooks as group correspondences are used for transfer, they are encrypted and securely aggregated by semi-honest server, and user/item profiles are updated by parties.

**Training Process**

1. Server initializes and encrypts codebook; Parties initializes user and item profiles;

2. Server distributes encrypted codebook to parties;

3. Parties update user and item factors by decrypted codebook;

4. Parties compute codebook gradients and send encrypted gradients to server;

Server securely aggregates encrypted codebook gradients and updates codebook.

Security of secure aggregation protocol is guaranteed [Bonawitz et al. 2017].
Application 1: Horizontal Federated Movie Recommendation

Recommender keeps user data on local devices, protects privacy while achieving lossless performance.

Application 2: Vertical Federated News Feeds Recommendation

Recommender leverages auxiliary user data to address cold start and improve performance.

User’s Internet browsing behaviors from 3rd-party

Finance News Feeds Recommendation

- PV: 21%
- UV: 22%
- CTR: 11%
An Industrial Grade Federated Learning Framework

support federated learning architectures and secure computation of any machine learning algorithms

Get started
FATE: Open-sourced Framework (https://github.com/FederatedAI/FATE)

- **FATE-Flow**
  - Flow DAG Parser
  - Lifecycle Manager
  - Multi-Party Task Scheduler

- **FATE-Serving**
  - Federated Inference
  - Model Manager
  - Version Control

- **FATE-Board**
  - FL Visualization
  - Monitoring
  - Log Manager

---

**Federated ML**: Federated Machine Learning Core Component

**EggRoll**: Distributed Computing Framework

**Data Injection**
- Access Interface: HIVE, MySQL, Level DB
- Format Adapter: Amazon S3, CSV, HBASE, HDFS, ......

**Device**
- CPU Clusters
- GPU Clusters
- Android/IOS
IEEE Standard P3652.1 – Federated Machine Learning

➢ **Title:**
  - Guide for Architectural Framework and Application of Federated Machine Learning

➢ **Scope:**
  - Description and definition of federated learning
  - The types of federated learning and the application scenarios to which each type applied
  - Performance evaluation of federated learning
  - Associated regulatory requirements

**Call for participation**
- More info: [https://sagroups.ieee.org/3652-1/](https://sagroups.ieee.org/3652-1/)

IEEE Standard Association is an open platform and we are welcoming more organizations to join the working group.
Summary

• Recommender systems can be improved with more data
• Yet privacy and security needs to be addressed
• Federated learning to bridge decentralized data in recommendation
  • Vertical Federated Recommendation (a.k.a. user-based FedRec)
  • Horizontal Federated Recommendation (a.k.a. item-based FedRec)
  • Transfer Federated Recommendation
• FedRec is an underexplored area with a lot of opportunities