Context-Aware Service Recommendation

[Recommender System and Service Computing]

Yueshen Xu

Xidian University; Zhejiang University

xyshzjucs@zju.edu.cn
ysxu@xidian.edu.cn
Background
- Why do those Web services need to be recommended?
- What’s the challenge?

Context-aware Recommendation
- What kind of context information can we use?
- Basic method

Context-aware feature learning
- User context-aware features learning
- Service service-aware features learning

Experiment
- Experimental Result
- Performance Comparison

Reference
Web Service

A Web service is a self-describing programmable application used to achieve interoperability and accessibility over a network—‘Services Computing’, Zhang, L. et al.

Other similar component

- Open API:
  - Douban, Sina Weibo, RenRen, 51.com
  - Amazon, Facebook, MySpace

Popularity

- Amazon Relational Database Service
- Amazon Simple Storage Service

They are almost the same

General restricted

WS = WSDL + Interface + SOAP

XDU & ZJU
Background

Cloud Computing → the number of Web services exploding

Decision Basis

QoS (response time, throughput, availability etc.)

Problem

Everyone wants to invoke the one whose qos is the best.

Is there a service suitable for everyone? No. → Why? → Context

So the real problem is: which one should I invoke?

Personalization & Prediction
Background

Formalization of the Problem

- Personalized QoS prediction
- User-Service Invocation Matrix

<table>
<thead>
<tr>
<th></th>
<th>service1</th>
<th>service2</th>
<th>service3</th>
<th>service4</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>q_{11}</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>user2</td>
<td>✗</td>
<td></td>
<td>q_{22}</td>
<td>✗</td>
</tr>
<tr>
<td>user3</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>user4</td>
<td>q_{41}</td>
<td>✗</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user5</td>
<td>✗</td>
<td>✗</td>
<td>q_{53}</td>
<td>✗</td>
</tr>
</tbody>
</table>

Challenge

- Data Sparsity → Cold Start Problem
- Large Scale → Feasibility

Similar with rating prediction in e-commerce systems

Sufficient Features + Effective Model

Cold-Start Problem
Context-aware recommendation

Contextual Features Selection

➢ Basis: factors dominating QoS: *physical configuration*

Feature Quantification

➢ User-User: *Geographical Distance*
➢ Service-Service: *Service Provider*

Why?
Context-aware recommendation

Feature Quantification (Con.)

Observation

- For the exactly same service or user:
  ✓ Users in different locations usually experience different QoS
  ✓ Users in the same location usually experience similar QoS

Assumption:

- Users located nearly with each other have similar IT infrastructure
- Services provided by the same company have similar physical configuration

Reasonable?

Users in city, town, community, just like browsing in Internet.

Services operated by different providers usually offer different QoS
Services operated by the same providers usually offer similar QoS
Context-aware recommendation

- Assumption: Reasonable?

Web service invocation scenario

- Algorithm: the basic model
  - Collaborative Filtering/Matrix Factorization ← traditional recommender system
Related Work → Collaborative Filtering

- Explore similar users and services through their historical invocation records → much like recommender systems

Whose invocation is similar with mine?

- Pearson Correlation Coefficient → Similarity Computation

\[
Sim(a,u) = \frac{\sum_{i \in I} (q_{ai} - \bar{q}_a)(q_{ui} - \bar{q}_u)}{\sqrt{\sum_{i \in I} (q_{ai} - \bar{q}_a)^2} \sqrt{\sum_{i \in I} (q_{ui} - \bar{q}_u)^2}}
\]

\[
Sim(i, j) = \frac{\sum_{u \in U} (q_{ui} - \bar{q}_i)(q_{uj} - \bar{q}_j)}{\sqrt{\sum_{u \in U} (q_{ui} - \bar{q}_i)^2} \sqrt{\sum_{u \in U} (q_{uj} - \bar{q}_j)^2}}
\]

Predicted Results

- Self Average + Weighted Average

\[
p_{ui} = \bar{u} + \frac{\sum_{u_a \in S(u)} Sim(u, u_a)(r_{u_a} - \bar{u}_a)}{\sum_{u_a \in S(u)} Sim(u, u_a)}
\]

\[
p_{ui} = \bar{i} + \frac{\sum_{i_k \in S(i)} Sim(i, i_k)(r_{i_k} - \bar{i}_k)}{\sum_{i_k \in S(i)} Sim(i, i_k)}
\]

Context-aware recommendation

XDU & ZJU
Context-aware recommendation

**Probabilistic Matrix Factorization (PMF)**

\[ p(Q \mid U, S, \sigma_Q^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} N\left( q_{ij} \mid U_i^T S_j, \sigma_Q^2 \right) \]

\[ p(U \mid \sigma_U^2) = \prod_{i=1}^{m} N(U_i \mid 0, \sigma_U^2 I) \]

\[ p(S \mid \sigma_S^2) = \prod_{j=1}^{n} N(S_j \mid 0, \sigma_S^2 I) \]

\[ p(U, S \mid Q) \propto p(Q \mid U, S) \times p(U) \times p(S) \]

\[ p(U, S \mid Q) = \prod_{i=1}^{m} \prod_{j=1}^{n} N\left( q_{ij} \mid U_i^T S_j, \sigma_Q^2 \right) \]

\[ \times \prod_{i=1}^{m} N(U_i \mid 0, \sigma_U^2 I) \]

\[ \times \prod_{j=1}^{n} N(S_j \mid 0, \sigma_S^2 I) \]

\[ \ln p(U, S \mid Q, \sigma_Q^2, \sigma_U^2, \sigma_S^2) \]

\[ - \frac{1}{2\sigma_Q^2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (q_{ij} - U_i^T S_j)^2 \]

\[ - \frac{1}{2\sigma_U^2} \sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_S^2} \sum_{j=1}^{n} S_j^T S_j \]

\[ - \frac{1}{2} \left( \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \right) \ln \sigma_Q^2 + MD \ln \sigma_U^2 + ND \ln \sigma_S^2 + C \]

**Generative Model**

**Statistical Machine Learning**

**Maximize Posterior Probability (MAP)**

**Bayesian Inference**

Logarithm Transformation

Maximization Minimization
Context-aware recommendation

- **Probabilistic Matrix Factorization**
  - MF can factorize the high-rank user-service invocation feature space into the joint low-rank feature space
  - $Q = \{q_{ij}\}: m \times n$ user-service invocation matrix
  - $U \in \mathbb{R}^{d \times m}$, $S \in \mathbb{R}^{d \times n}$: user and service feature matrices

  $q_{ij} \approx U_i^T S_j$ Inner product of two $d$-rank feature vectors

  $$\min_{U,S} \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (q_{ij} - U_i^T S_j)^2$$

  Minimization function

  $$\min L = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n I_{ij} (q_{ij} - U_i^T S_j)^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_S}{2} \|S\|_F^2$$

  Objective Optimization Function

  How can the context information be merged into the basic model?
Context-aware recommendation

### Matrix Decomposition
- Tri-angle
- LU
- QR
- Spectral
- SVD

### Matrix Factorization
- Basic MF
- Non-negative
- PMF
- BPMF
- pLSA, LDA

---

XDU & ZJU
Context-aware features learning

----User Side

**User Neighborhood Identification**
- Latitude and Longtitude Nearest Neighbors (KNN)

**Similarity Computation**
- Distinguish each neighbor’s significance
- The similarity of user *i* and user *j* can be defined as

\[ Sim_{ij} = \exp(-d_{ij}) \]

- \( Sim_{ij} = 1 \): i and j live in the exactly same place
- \( Sim_{ij} \to 0 \): i and j live extremely far

\[ w_{il} = \frac{Sim_{il}}{\sum_{g \in KNN} Sim_{ig}} \]

\[ q_{KNN \cdot j} = \sum_{l=1}^{K} w_{il} U_i^T S_j \quad (U_i \in KNN \text{ of user } i) \]
Context-aware features learning

User Side

- Reasonable Combination
  \[ q_{ij} \approx \hat{q}_{ij} = \alpha U_i^T S_j + (1 - \alpha) \sum_{l=1}^{k} w_{il} U_l^T S_j \]
  Learned from his/her neighbors’ invocation experience

- Final Objective Function
  \[ L = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (q_{ij} - (\alpha U_i^T S_j + (1 - \alpha) \sum_{l=1}^{k} w_{il} U_l^T S_j))^2 + \frac{\lambda_U}{2} \| U \|_F^2 + \frac{\lambda_S}{2} \| S \|_F^2 \]
  \[ \text{Global Minimum} \rightarrow \text{Closed-form solution} \rightarrow \text{NP Hard} \]
  \[ \text{(Stochastic) Gradient Descent} \rightarrow \text{Local Minimum} \]

\[
\begin{align*}
\frac{\partial E}{\partial U_i} &= \alpha \sum_{j=1}^{n} I_{ij} S_j (\alpha U_i^T S_j + (1 - \alpha) \sum_{l=1}^{k} w_{il} U_l^T S_j - q_{ij}) + \lambda_U U_i \\
\frac{\partial E}{\partial S_j} &= \sum_{i=1}^{m} I_{ij} (\alpha U_i^T S_j + (1 - \alpha) \sum_{l=1}^{k} w_{il} U_l^T S_j - q_{ij}) \times (\alpha U_i + (1 - \alpha) \sum_{l=1}^{k} w_{il} U_l) + \lambda_S S_j
\end{align*}
\]
Context-aware features learning

--- Service Side

- **Service Neighborhood Identification**
  - Operated by the same service provider

- **Reasonable Combination**
  
  \[ q_{ij} \approx \hat{q}_{ij} = \alpha U^T_i S_j + (1-\alpha) \frac{1}{|C(j)|} \sum_{c \in C(j)} U^T_i S_c \]

  Regulating Factor

  Learned from its neighbors' invocated experience

- **Final Objective Function**

  \[
  L = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (q_{ij} - (\alpha U^T_i S_j + (1-\alpha) \frac{1}{|C(j)|} \sum_{c \in C(j)} U^T_i S_c))^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_S}{2} \|S\|_F^2
  \]

  *(Stochastic) Gradient Descent \rightarrow Local Minimum*

  the same with the computation process of user-side
Context-aware features learning

--- Combination

- **Model Unification == Ensemble Learning**
  - Model Fusion: too parameters, too complicated
  - Result Aggregation: simple and effective

- **Result Aggregation**
  - Strategy 1: Fixed Proportion
    \[ q_{ij} \approx \hat{q}_{ij} = \lambda \times p_{user-side} + (1 - \lambda) \times p_{service-side} \]
    \( \lambda: \) a fixed decimal
    - too simple, too naive
  - Strategy 2: Dynamic Proportion
    \[ q_{ij} \approx \hat{q}_{ij} = w_i \times p_{user-side} + w_j \times p_{service-side} \]
    \[ w_i = \frac{con_i \times \lambda}{con_i \times \lambda + (1 - \lambda) \times con_j}, \quad w_j = \frac{con_j \times (1 - \lambda)}{con_i \times \lambda + (1 - \lambda) \times con_j} \]
Context-aware features learning

--- Combination

A Unified Framework

- Offline and Online
- Preparation for System Building
- It needs further improvement

Building the system has been on the schedule

Update when the info in databases has been updated

Update when the offline step has been updated
## Experiment

### Preparation

- **Dataset:** real world dataset
- **Matrix Density:** 5%~20%
- **Evaluation Metrics:** RMSE and MAE
- **Result:** average of multiple testing

<table>
<thead>
<tr>
<th>Approach</th>
<th>Matrix Density (MD)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>MAE</td>
<td>RMSE</td>
<td>MAE</td>
</tr>
<tr>
<td>UPCC</td>
<td>1.6940</td>
<td>0.7229</td>
<td>1.6101</td>
<td>0.6473</td>
</tr>
<tr>
<td>IPCC</td>
<td>1.6346</td>
<td>0.7347</td>
<td>1.5439</td>
<td>0.6610</td>
</tr>
<tr>
<td>UIPCC</td>
<td>1.5837</td>
<td>0.7043</td>
<td>1.4744</td>
<td>0.6208</td>
</tr>
<tr>
<td>RegionKNN</td>
<td>1.5681</td>
<td>0.6958</td>
<td>1.4554</td>
<td>0.6092</td>
</tr>
<tr>
<td>Basic-MF</td>
<td>1.4952</td>
<td>0.7251</td>
<td>1.3127</td>
<td>0.5902</td>
</tr>
<tr>
<td>EMF_F</td>
<td>1.4700</td>
<td>0.7162</td>
<td>1.2955</td>
<td>0.5737</td>
</tr>
<tr>
<td>NIMF</td>
<td>1.4023</td>
<td>0.6316</td>
<td>1.2702</td>
<td>0.5384</td>
</tr>
<tr>
<td>SN-EMF</td>
<td>1.3935</td>
<td>0.6273</td>
<td>1.2689</td>
<td>0.5363</td>
</tr>
<tr>
<td>UN-EMF</td>
<td>1.3850</td>
<td>0.6139</td>
<td>1.2666</td>
<td>0.5359</td>
</tr>
<tr>
<td>U-EMF</td>
<td>1.3503</td>
<td>0.5840</td>
<td>1.2548</td>
<td>0.5171</td>
</tr>
</tbody>
</table>

Large Improvement v.s. Large computation

Offline → Online (linear)

Our methods

---

2017/6/15

XDU & ZJU
**Impact of $\alpha$**

- The parameter $\alpha$ controls the individual contributions of the user/service and their neighbors to the predicted value
- Neighbors’ Contributions are dominating

**Impact of $\lambda$**

- The parameter $\lambda$ regulates the weight of user-side model and service-side model
- User’s context is more important
  - Why?
## Impact of TopK

- The parameters $\text{TopK(S)}$ and $\text{TopK(U)}$ determine the number of neighbors involved in the learning process.
- Too few or too much neighbors are both not so proper.
Future Work

- Explore more efficient strategies for model aggregation
- Trying to find conduct the unification method of model fusion (*model fusion*)
- Build a practical service recommender system
- Refining the similarity function between users
- Collecting a real world dataset by ourselves
- Others
Reference


Thank You !
Q&A

Context-Aware Web Service Recommendation