Ranking in Context-Aware Recommender Systems

Minsuk Kahng, Sangkeun Lee, Sang-goo Lee

School of Computer Science and Engineering, Seoul National University, South Korea

{minsuk, liza183, sglee}@europa.snu.ac.kr

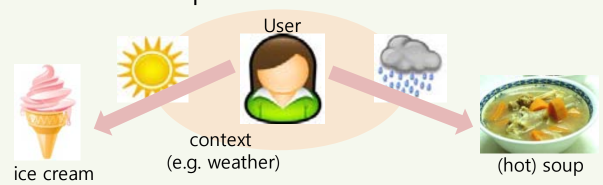
Abstract

- Propose a novel context-aware recommendation method
- Present five types of features by slicing multi-dimensional data, and combine these features using learning-to-rank

Problem

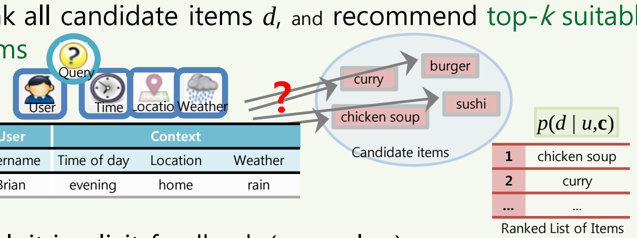
Context-Aware Recommendation?

- Use contextual information (e.g. location, time, weather) to estimate user's preference for recommendation



Problem Formulation

- Given a query (user u and current context $\mathbf{c}$), rank all candidate items d, and recommend top-k suitable items



User	Context		
Username	Time of day	Location	Weather
Brian	evening	home	rain

- Exploit implicit feedback (usage log)
  - e.g. accessing a webpage, listening to a song, watching a video

	User	Context			Item
	Username	Time of day	Location	Weather	Food
1	Tom	morning	office	cloudy	sushi
2	Karen	afternoon	downtown	rain	soup
3	Matt	evening	home	sunny	burger
4	Brian	evening	office	rain	soup

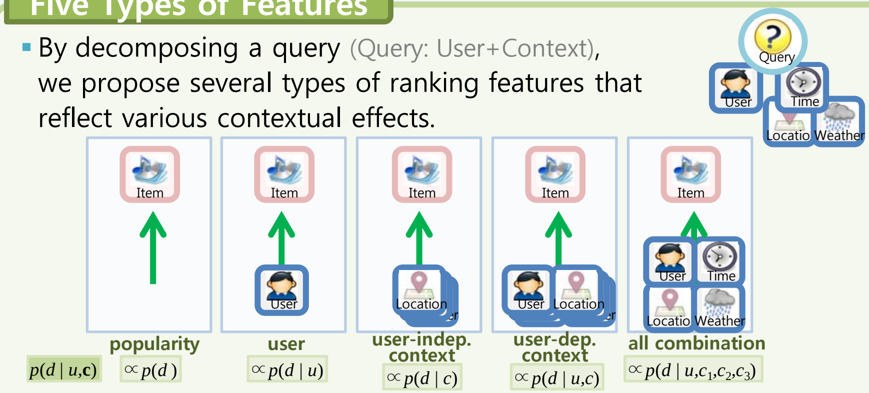
Motivation

- Context can influence users' item preferences in various ways depending on the domain and application.
  - e.g. No context, user-independent, a set of contextual values, …
- Try to capture various contextual effects.

Approach

Five Types of Features

- By decomposing a query (Query: User+Context), we propose several types of ranking features that reflect various contextual effects.

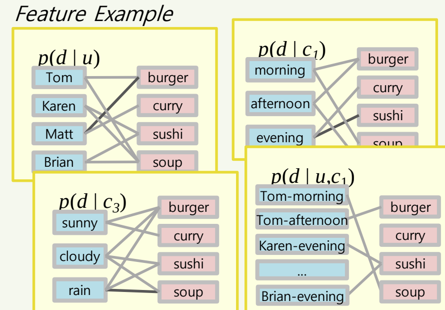


$$p(d \mid u,\mathbf{c}) = f\big(p(d),\ p(d \mid u),\ p(d \mid c_1),\ p(d \mid u,c_1),\ p(d \mid u,c_1,c_2,c_3)\big)$$

Make features by slicing multi-dimensional data (cube)

User	Context			Item
Username	Time of day	Location	Weather	Food
Tom	morning	home	rain	soup
Tom	afternoon	School	sunny	burger
Karen	evening	downtown	cloudy	sushi
Karen	afternoon	school	cloudy	soup
Matt	morning	home	cloudy	burger
Matt	afternoon	school	rain	burger
Matt	evening	downtown	sunny	curry
Brian	afternoon	school	rain	soup
Brian	evening	downtown	rain	sushi

Feature Example



Retrieval model

- Adopt an existing LDA-based retrieval model [Wei & Croft, SIGIR, '06]
- Capture past history & similar user/context using latent topic

$$p(u,v|d) = \lambda\left(\frac{N_d}{N_d+\mu}\boxed{p_{ML}(u,v|d)} + \left(1-\frac{N_d}{N_d+\mu}\right)p_{ML}(u,v|C)\right) + (1-\lambda)\boxed{p_{LDA}(u,v|d)}$$

Combine Features

- Combine the features using *learning to rank*
- Choose to use *Ranking SVM.*
  - Learn the weight of each feature $y(\mathbf{x}_i) = w_1 \times f_1(d_1,q) + w_2 \times f_2(d_1,q) + \ldots + w_K \times f_K(d_1,q)$
- For pairwise preference, some negative examples are randomly sampled.
  - For each record, there is only one positive feedback (item).

Experiments

Dataset

Evaluate with two real-world datasets

1. Bugs Music – music listening log
   - 3 contextual variables: time of day, date, weather
   - The system suggests new songs.
2. Foursquare – place check-in log
   - 3 contextual variables: GPS, time of day, date
   - Collected dataset using Twitter API.
   - The system suggests top-k places for a query.

How to evaluate?

Measure – Hit ratio, NDCG

- Check if the model produces a list that includes the item in the record

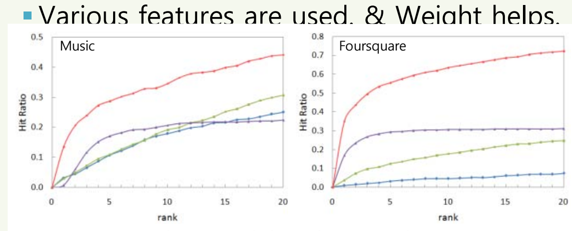
Methods in Comparison

- Compare to three baseline methods – popularity (1st feature), user only (CF) (1st, 2nd), reduction-based approach [Adomavicius *et al.*, TOIS, '05]

Results

Outperforms several baseline methods.

- Various features are used. & Weight helps.



20th international world wide web conference
28th march - 1st april 2011
hicc, hyderabad, india

SEOUL NATIONAL UNIVERSITY