Which App Will You Use Next? Collaborative Filtering with Interactional Context

Donghyuk Shin
Dept of Computer Science
UT Austin

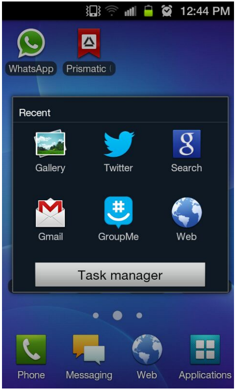
ACM RecSys'13
Oct 15, 2013

Joint work with N. Natarajan and I. S. Dhillon

Which App Will You Use Next?

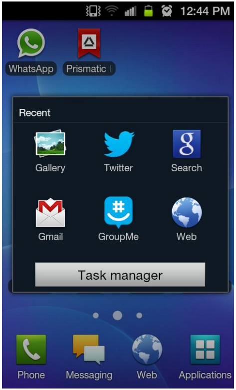
Which App Will You Use Next?

Given a sequence of apps used *recently*...



Which App Will You Use Next?

Given a sequence of apps used *recently*...



Recommend an app to use *next*

Which App Will You Use Next?

- Users click on items (apps) that are of interest in the *current* context
- Sequence of recently used items should help better zero in on the items the user would click next
- Applicable in settings where items are generally used repeatedly
  - listening to music from online radio
  - browsing products on shopping websites
  - ...

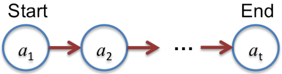
Problem Setting

Problem Setting

- Setting
  - Users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and items $\mathcal{I} = \{a_1, a_2, \ldots, a_{|\mathcal{I}|}\}$
  - Their history of sessions $\mathcal{S}$

Problem Setting

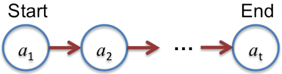
- Setting
  - Users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and items $\mathcal{I} = \{a_1, a_2, \ldots, a_{|\mathcal{I}|}\}$
  - Their history of sessions $\mathcal{S}$
- Session
  - Sequence of items accessed by a user over a certain contiguous period



  - Defined based on the type of activity
    e.g. articles read by an online user when she is signed in

Problem Setting

- Setting
  - Users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and items $\mathcal{I} = \{a_1, a_2, \ldots, a_{|\mathcal{I}|}\}$
  - Their history of sessions $\mathcal{S}$
- Session
  - Sequence of items accessed by a user over a certain contiguous period



  - Defined based on the type of activity
    e.g. articles read by an online user when she is signed in
- Problem

  Given a specific user $u \in \mathcal{U}$

  with session *in progress* $s = \langle a_{i_1}, a_{i_2}, \ldots, a_{i_t} \rangle$, for some $t \geq 1$

  Recommend the best candidate item $a_{i_{t+1}} \in \mathcal{I}$

Main Differences

- Past sessions — implicit feedback in the form of click sequences
- Session in progress — interactional context (Dourish, 2004)

Main Differences

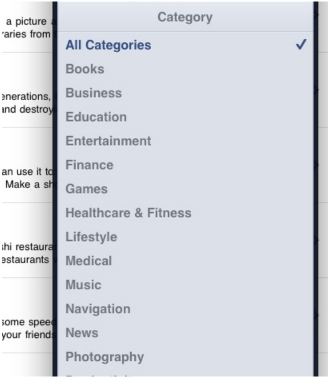
- Past sessions — implicit feedback in the form of click sequences
- Session in progress — interactional context (Dourish, 2004)

	Our problem	Traditional CF
Item interaction	Brief and repetitive	Usually once (watched/rated)
Feedback	Implicit (ordered clicks)	Explicit (ratings/comments)
Recommendations while user interacts	Dynamic	Static
Context	Interactional	Representational

Representational vs. Interactional Context

Representational Context:

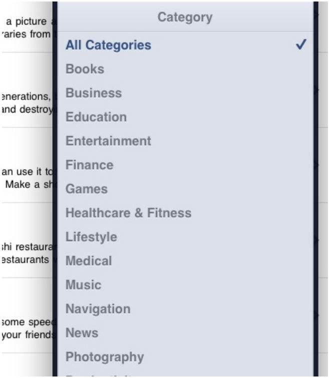
user specifies preferences globally



Representational vs. Interactional Context

Representational Context:

user specifies preferences globally



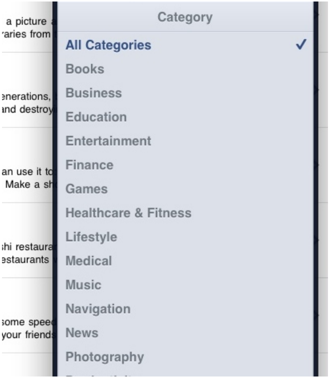
Interactional Context:

user preferences change between sessions

Representational vs. Interactional Context

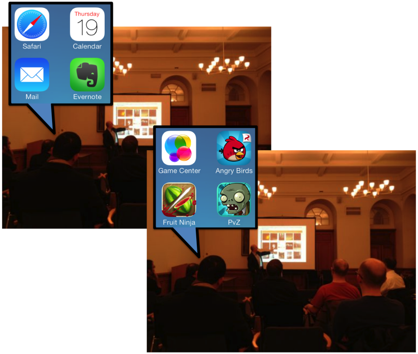
Representational Context:

user specifies preferences globally



Interactional Context:

user preferences change between sessions



Representational vs. Interactional Context

Representational Context:

- What most context-aware CF methods have used
- Form of information provided as attributes — delineable and stable e.g. location, time and interests
- Defined *before* user interaction and influences user action
- Context and activity are separable

Interactional Context:

- Relatively unexplored
- Scope of context is defined dynamically — should be inferred from user interaction
- Context *arises* from user interaction
- Context is a relational property — relevant to some particular activity

Proposed Method: iConRank

Implicit Feedback via Click Sequences

- Session data examples:

$\langle$browser, maps, messages$\rangle$
$\langle$mail, phone, settings, phone, browser, phone, browser, search$\rangle$
$\langle$phone, calendar, messages, camera$\rangle$
$\langle$settings, mail, browser, mail, browser$\rangle$
...

Implicit Feedback via Click Sequences

- Session data examples:

  ⟨browser, maps, messages⟩
  ⟨mail, phone, settings, phone, browser, phone, browser, search⟩
  ⟨phone, calendar, messages, camera⟩
  ⟨settings, mail, browser, mail, browser⟩
  ...

- User-item count matrix is *not* a good choice (Hu et al, 2008)
  - Need only ranking of items
  - Hard to gauge user preferences with clicks — no explicit scale
  - Want to model session history as sequences instead of counts

Implicit Feedback via Click Sequences

- Session data examples:

  ⟨browser, maps, messages⟩
  ⟨mail, phone, settings, phone, browser, phone, browser, search⟩
  ⟨phone, calendar, messages, camera⟩
  ⟨settings, mail, browser, mail, browser⟩
  ...

- User-item count matrix is *not* a good choice (Hu et al, 2008)
  - Need only ranking of items
  - Hard to gauge user preferences with clicks — no explicit scale
  - Want to model session history as sequences instead of counts
- Simple and effective way to model sequences → Markov models
  - State space: set of items $\mathcal{I}$
  - State transition probability matrix $M \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$
    - $M(i, j)$: probability that item j is clicked immediately after item i (first-order Markov model)
    - Estimated from training session data

Markov Modeling

Two extreme case Markov models:

- Individual-level
  - Estimated from session data of each user
  - Training data for a given user is too sparse to estimate $|\mathcal{I}| \times |\mathcal{I}|$ parameters
- Global-level
  - Estimated from all session data
  - Hard to capture diverse transition behavior of different users

Markov Modeling

Two extreme case Markov models:

- Individual-level
  - Estimated from session data of each user
  - Training data for a given user is too sparse to estimate $|\mathcal{I}| \times |\mathcal{I}|$ parameters
- Global-level
  - Estimated from all session data
  - Hard to capture diverse transition behavior of different users

Our approach:

- Cluster-level Markov model
  - Group users with common navigational patterns
  - Conforms with the basic idea of collaborative filtering

    "Combine preferences from *similar* users to make recommendations"

Behavioral Clustering

- Each user u is represented by Markov transition probability matrix $M^{(u)}$

Behavioral Clustering

- Each user u is represented by Markov transition probability matrix $M^{(u)}$
- Distance measure between two probability distributions: KL-divergence

$$d_{KL}(x, y) = \sum_{i=1}^{p} x_i \log_2 \left(\frac{x_i}{y_i} \right)$$

Behavioral Clustering

- Each user u is represented by Markov transition probability matrix $M^{(u)}$
- Distance measure between two probability distributions: KL-divergence

$$d_{KL}(x, y) = \sum_{i=1}^{p} x_i \log_2 \left(\frac{x_i}{y_i} \right)$$

- Optimize k-means objective and compute centroid M_k of cluster π_k (Banerjee et al, 2005)

$$M_k = \frac{1}{|\pi_k|} \sum_{u \in \pi_k} M^{(u)}$$

Distance between user u and cluster π_k (centroid M_k)

$$d(u, \pi_k) = \frac{1}{|\mathcal{I}|} \sum_{i=1}^{|\mathcal{I}|} d_{KL}(M^{(u)}(i, \cdot), M_k(i, \cdot))$$

Neighborhood-based Model

Predicted rating $\hat{r}_{u,i}$ of user u and item i:

$$\hat{r}_{u,i} = b_{u,i} + \sum_{j \in \mathcal{N}(i)} w(i,j)(r_{u,j} - b_{u,j})$$

$\Rightarrow$ Weighted combination of "k-nearest neighbor" items (or users).

- $r_{u,i}$: rating of item i by user u
- $b_{u,i}$: rating bias — user and item bias
- $w(i,j)$: weights proportional to similarity between items i and j
- $\mathcal{N}(i)$: neighborhood set of item i — k-nearest neighbor items

Neighborhood-based Model

Predicted rating $\hat{r}_{u,i}$ of user u and item i:

$$\hat{r}_{u,i} = b_{u,i} + \sum_{j \in \mathcal{N}(i)} w(i,j)(r_{u,j} - b_{u,j})$$

$\Rightarrow$ Weighted combination of "k-nearest neighbor" items (or users).

- $r_{u,i}$: rating of item i by user u
- $b_{u,i}$: rating bias — user and item bias
- $w(i,j)$: weights proportional to similarity between items i and j
- $\mathcal{N}(i)$: neighborhood set of item i — k-nearest neighbor items

Two main changes

- Bias: introduce interactional context bias
- Weights (neighborhood): cluster-level Markov models

Incorporating Interactional Context

Introduce context bias $c_{u,i}$

$$f_{u,i} = b_{u,i} + \alpha \sum_{j \in \mathcal{N}(i)} w(i,j)(f_{u,j} - b_{u,j}) + (1-\alpha) c_{u,i}$$

- Context bias:

$$c_{u,i} = \begin{cases} 1, & \text{if item } a_i \text{ appears in current session } s \\ 0, & \text{otherwise} \end{cases}$$

- $\alpha \in [0,1]$: tradeoff between current context and information from past sessions
- Recurrence: want to estimate all item scores as session progresses

Incorporating Interactional Context

Introduce context bias $c_{u,i}$

$$f_{u,i} = b_{u,i} + \alpha \sum_{j \in \mathcal{N}(i)} w(i,j)(f_{u,j} - b_{u,j}) + (1-\alpha) c_{u,i}$$

- Context bias:

$$c_{u,i} = \begin{cases} 1, & \text{if item } a_i \text{ appears in current session } s \\ 0, & \text{otherwise} \end{cases}$$

- $\alpha \in [0,1]$: tradeoff between current context and information from past sessions
- Recurrence: want to estimate all item scores as session progresses

Let $z_{u,i} = f_{u,i} - b_{u,i}$ to remove the click (rating) bias:

$$z_{u,i} = \alpha \sum_{j \in \mathcal{N}(i)} w(i,j) z_{u,j} + (1-\alpha) c_{u,i}.$$

Similarity and Neighborhood

Let $w(i,j) = M(j,i)$

$$z_{u,i} = \alpha \sum_{j \in \mathcal{N}(i)} w(i,j) z_{u,j} + (1-\alpha) c_{u,i}$$

$$\Downarrow$$

$$\mathbf{z}_u = \alpha M^T \mathbf{z}_u + (1-\alpha)\mathbf{c}_u$$

Similarity and Neighborhood

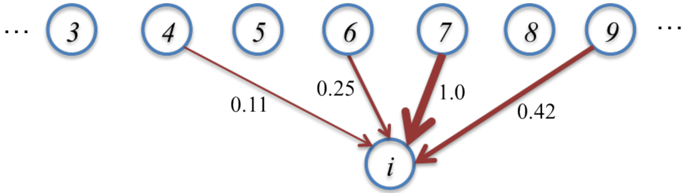
Let $w(i,j) = M(j,i)$

$$z_{u,i} = \alpha \sum_{j \in \mathcal{N}(i)} w(i,j) z_{u,j} + (1-\alpha) c_{u,i}$$

$$\Downarrow$$

$$\mathbf{z}_u = \alpha M^T \mathbf{z}_u + (1-\alpha)\mathbf{c}_u$$

- Items that tend to transition to i more often receive higher similarity
- $\mathcal{N}(i)$ corresponds to items that transition to i in the Markov graph



Personalized PageRank

Normalize $\mathbf{z}_u$ to sum to 1

$$\mathbf{z}_u = (\alpha M + (1-\alpha)\mathbf{1}\mathbf{c}_u^T)^T \mathbf{z}_u$$

Personalized PageRank!

using Markov graph M and personalization vector $\mathbf{c}_u$

Personalized PageRank

Normalize $\mathbf{z}_u$ to sum to 1

$$\mathbf{z}_u = (\alpha M + (1-\alpha)\mathbf{1}\mathbf{c}_u^T)^T \mathbf{z}_u$$

Personalized PageRank!

using Markov graph M and personalization vector $\mathbf{c}_u$

Recall $z_{u,i} = f_{u,i} - b_{u,i}$

- Since focus is on item transitions, we set $b_{u,i}$ to be user u's transition probabilities of item a_{i_t} to other items
- Final score

$$f_{u,i} = z_{u,i} + M^{(u)}(i_t, i)$$

where $M^{(u)}$ is the transition probability matrix of user u

iConRank Algorithm

1. (**Offline step**) Cluster users $\mathcal{U}$ with past sessions $\mathcal{S}$ into k clusters using k-means algorithm. Compute transition matrix M_k for each cluster k:

$$M_k = \frac{1}{|\pi_k|} \sum_{u \in \pi_k} M^{(u)}$$

2. Compute personalized PageRank

$$\mathbf{z}_u = (\alpha M_k + (1-\alpha)\mathbf{1}\mathbf{c}_u^T)^T \mathbf{z}_u$$

   with current $\mathbf{c}_u$ and $k = \pi(u)$ is the cluster user u is assigned to
3. Compute scores $f_{u,i} = z_{u,i} + M^{(u)}(i_t, i)$
4. Rank items using $f_{u,i}$ and return the top-N items as recommendations for $a_{i_{t+1}}$

iConRank Algorithm

- Efficiency
  - Clustering step is done *offline* — independent of current session
  - Use the *linearity property* of personalized PageRank:

$$\mathbf{z}_u(\beta \mathbf{v}_1 + (1-\beta)\mathbf{v}_2) = \beta \mathbf{z}_u(\mathbf{v}_1) + (1-\beta)\mathbf{z}_u(\mathbf{v}_2)$$

  where $\mathbf{z}_u(\mathbf{v})$ is the PageRank using personalization vector $\mathbf{v}$

    - Pre-compute $\mathbf{z}_u(\mathbf{e}_{i_j})$, then $\mathbf{z}_u(\mathbf{c}_u) = \sum_{j=1}^{t} c_{u,i_j} \cdot \mathbf{z}_u(\mathbf{e}_{i_j})$
    - Only need to pre-compute for apps installed on a device

- Cold start
  - No current session: use uniform distribution as $\mathbf{c}_u$
  - New user: use global-level Markov model instead of M_k

Experimental Results

Datasets

Two real-life datasets:

- `Apps`:
  - Proprietary dataset obtained from a manufacturer of mobile devices[1]
  - Session is started when screen=on and ended when the screen=off and at least one minute has elapsed ('home' app removed from dataset)
- `LastFM`:
  - Publicly available dataset from `last.fm`
  - Session ended if there is no other artist streamed within an hour from the last artist

Statistic	Apps	LastFM
# of users	17,062	941
# of items (apps/artists)	9,583	98,412
# of training sessions	1,167,171	644,001
# of testing sessions	459,899	95,038
Average session length	6.53	18.33
Median session length	3	7

[1] *The data provided to us was highly anonymized and contained only generic identifiers that cannot be correlated or traced back to actual users*

Behavioral Clusters

- Cluster Stats

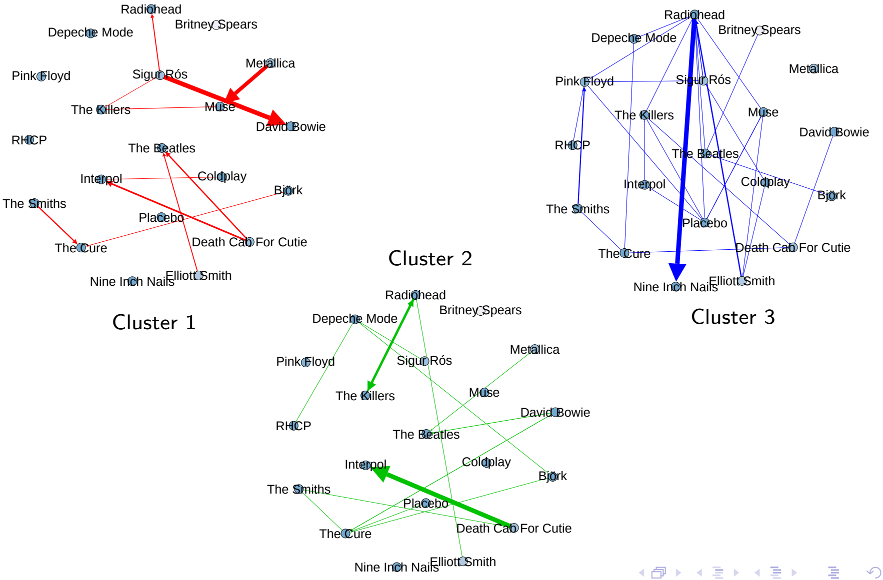
Dataset	Cluster	Users	nnz(M_k)	Session Length Avg.	Session Length Median
Apps	1	4,695	68,967	6.37	3
	2	5,711	73,121	6.32	3
	3	6,656	78,963	6.83	3
LastFM	1	327	1,733,056	19.15	7
	2	320	1,500,913	18.55	7
	3	294	1,227,933	17.01	7

- Top transitions in Apps dataset

Cluster1	Cluster2	Cluster3
(contacts,phone)	(phone,dialer)	(phone,dialer)
(message,contacts)	(*search,browser*)	(1,phone)
(contacts,dialer)	(*mail,browser*)	(phone,2)
(*settings,phone*)	(message,mail)	(phone,3)
(*data,settings*)	(*calendar,mail*)	(4,phone)
(*camera,photo*)	(*browser,video*)	(5,phone)
(*contacts,settings*)	(*phone,browser*)	(message,mail)

Behavioral Clusters

Transitions between top-20 artists



Evaluation

Compared methods:

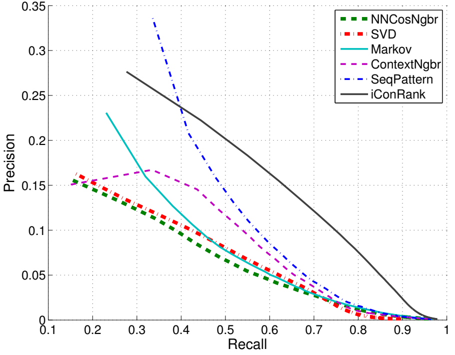
- Methods using user-item count matrix:
  - NNCosNgbr: neighborhood-based model using cosine similarity
  - SVD: singular value decomposition
- Methods using user's current session:
  - **iConRank**: our proposed method
  - Markov: global-level Markov model
  - ContextNgbr: same as NNCosNgbr except that the k-nearest neighbors are computed from items in the current session
  - SeqPattern: sequence mining algorithm (Hariri et al, 2012)

Performance measure: recall at top-N (Cremonesi et al, 2010)

Results

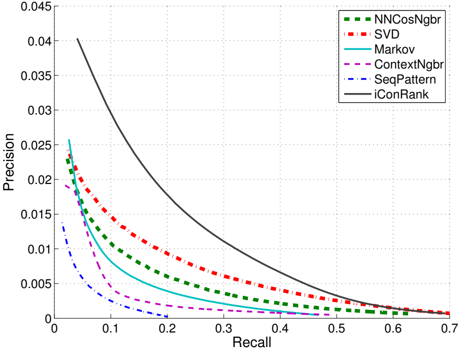
- Apps

Method	Recall@N			
	$N=5$	$N=10$	$N=15$	$N=20$
NNCosNgbr	0.4301	0.5478	0.6167	0.6636
SVD	0.4574	0.5853	0.6480	0.6851
Markov	0.4592	0.5744	0.6370	0.6754
ContextNgbr	0.5266	0.6248	0.6739	0.7045
SeqPattern	0.5517	0.6451	0.6899	0.7223
iConRank	**0.6701**	**0.7927**	**0.8386**	**0.8632**



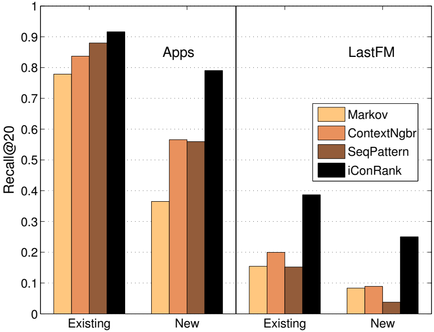
- LastFM

Method	Recall@N			
	$N=5$	$N=10$	$N=15$	$N=20$
NNCosNgbr	0.0691	0.1044	0.1328	0.1560
SVD	0.0810	0.1286	0.1633	0.1922
Markov	0.0631	0.0905	0.1113	0.1285
ContextNgbr	0.0597	0.0775	0.0884	0.0971
SeqPattern	0.0371	0.0536	0.0656	0.0748
iConRank	**0.1277**	**0.1882**	**0.2304**	**0.2633**

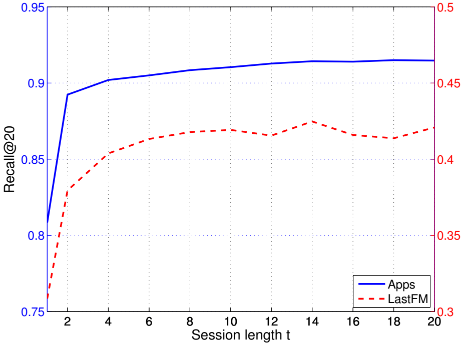


Results

- Performance on existing and new items



- Performance as more of the current session revealed



Conclusions

- Collaborative filtering with interactional context
- Applicable in many click-based interactive systems
- Proposed novel method: iConRank
  - Personalized and dynamic recommendations given the current session (interactional context) — recommendations are updated as the user interacts with the system
  - Behavioral clustering — recommendations using past item transitions and transitions from users with similar navigational patterns
  - Superior recommendations than other competitive methods on two real-life datasets
  - Scalable and can be efficiently implemented on devices with limited processing power

Thank You!

References

[1] P. Dourish. *What we talk about when we talk about context*. Personal and Ubiquitous Computing, 2004.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. *Clustering with bregman divergences*. JMLR, 2005.

[3] Y. Hu, Y. Koren, and C. Volinsky. *Collaborative filtering for implicit feedback datasets*. ICDM, 2008.

[4] P. Cremonesi, Y. Koren, and R. Turrin. *Performance of recommender algorithms on top-N recommendation tasks*. ACM RecSys, 2010.

[5] N. Hariri, B. Mobasher, and R. Burke. *Context-aware music recommendation based on latent topic sequential patterns*. ACM RecSys, 2012.

[6] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu. *Fast app launching for mobile devices using predictive user context*. MobiSys, 2012.