Context Aware Recommendation

Bamshad Mobasher

Center for Web Intelligence

School of Computing
DePaul University, Chicago

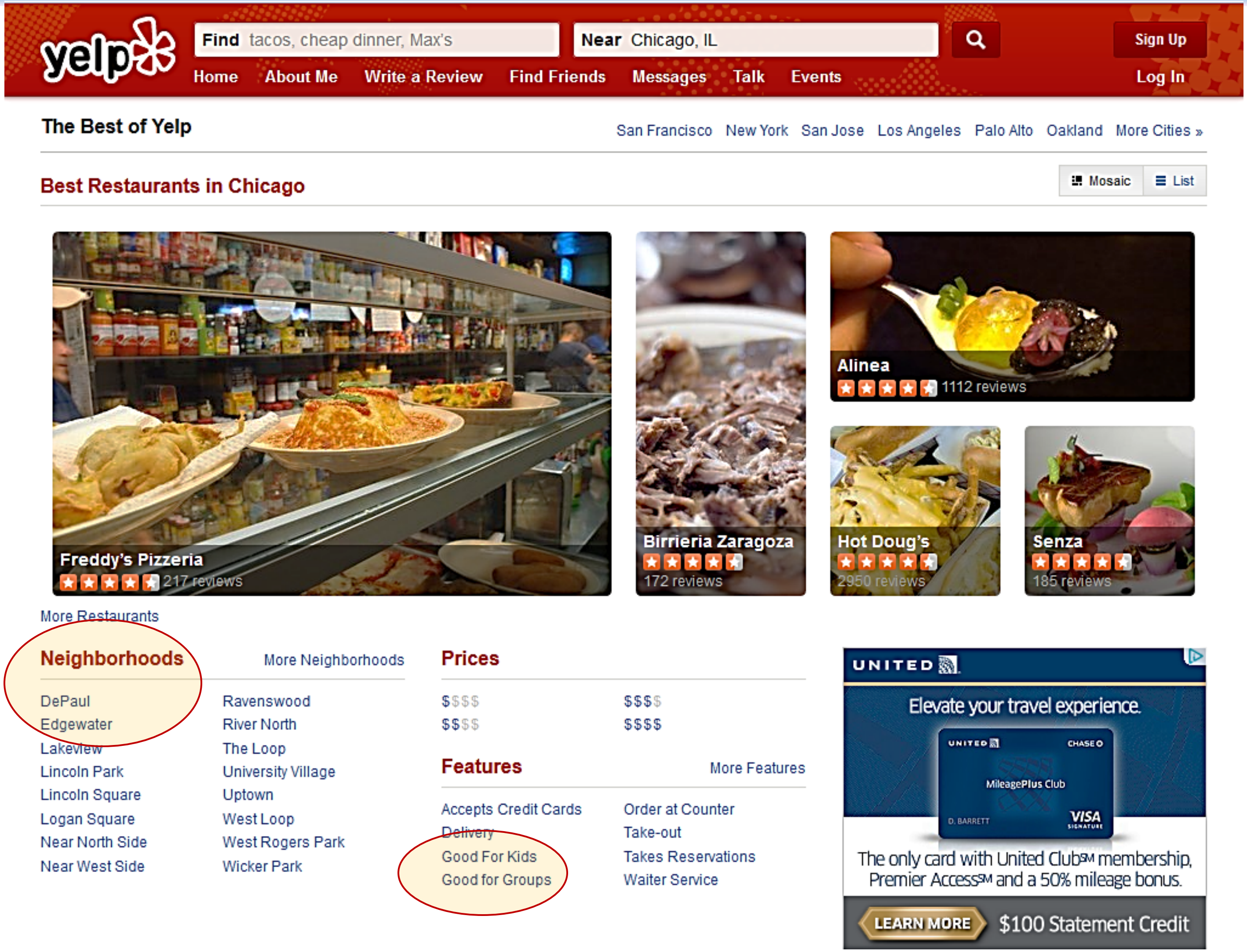
Outline

- **General views of context and their relevance to recommendation problem**
  - Key issues for context-aware system
  - Representational versus Interactional Views
- **Key Concepts in Contextual Aware Recommendation**
  - Architectures for integrating context in recommender systems
  - Highlighted Approaches in the Representational Framework
    - Item / User Splitting
    - Differential Contextual Modeling
    - Approaches based on Matrix Factorization
  - Interactional Context
    - Example Architecture: A Framework based on human memory
    - Highlighted Approach: Latent Variable Context Modeling

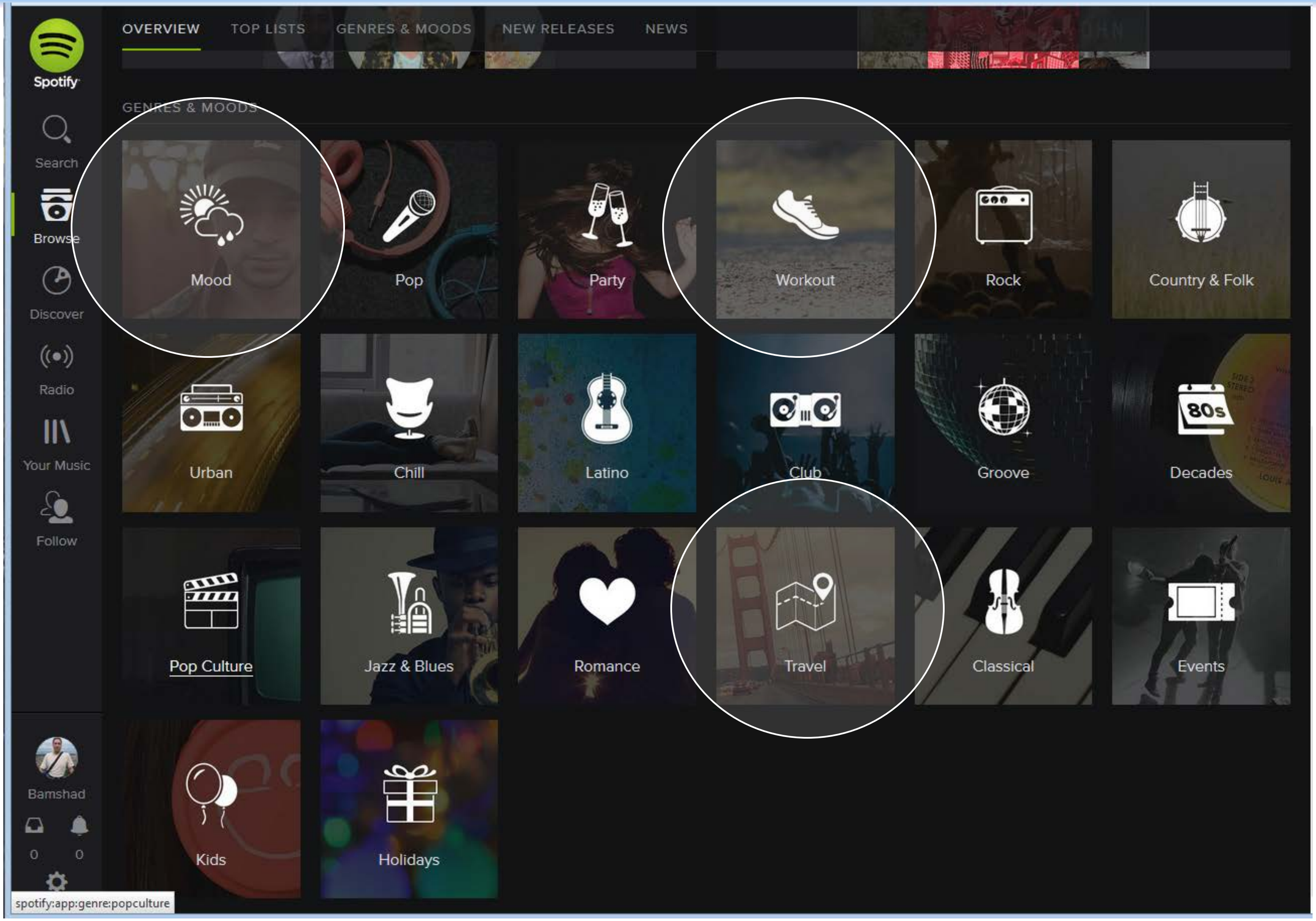
Context in Recommendation

- **Recommendation Scenario**
  - Dan's purchases on Amazon:
    - mystery-detective fiction "Da Vinci Code" (for himself)
    - "Python Programming" (for work)
    - "Green Eggs and Ham" (gift for his daughter)
  - How should we represent Dan's *interest in books*?
  - System needs to know the difference between children books and computer books, i.e., the contexts in which Dan interacts with the system
  - What should be recommended if Dan is reading reviews for a book on Perl Scripting?

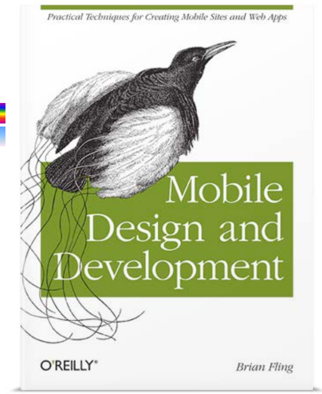
Context in Recommendation



Context in Recommendation



Types of Context



[Fling, 2009]

- **Physical context**
  - time, position, and activity of the user, weather, light, and temperature ...
- **Social context**
  - the presence and role of other people around the user
- **Interaction media context**
  - the device used to access the system and the type of media that are browsed and personalized (text, music, images, movies, …)
- **Modal context**
  - The state of mind of the user, the user’s goals, mood, experience, and cognitive capabilities.

What the system knows about context

How Contextual Factors Change	Knowledge of the RS about the Contextual Factors		
	Fully Observable	Partially Observable	Unobservable
Static	Everything Known about Context	Partial and Static Context Knowledge	Latent Knowledge of Context
Dynamic	Context Relevance Is Dynamic	Partial and Dynamic Context Knowledge	Nothing Is Known about Context

See: Adomavicius, Mobasher, Ricci, and Tuzhilin. Context Aware Recommender Systems. *AI Magazine*, Fall 2011

Defining Context

- **Entities interact with their environment through "situated actions"**
  - "Any information that can be used to characterize the situation of entities." (Dey et al., 2001)

- **Context of an entity exist independently and outside of the entity's actions**
  - Everything that affects computation except its explicit input and output." (Lieberman and Selker, 2000)

Different Views of Context

- **Paul Dourish (2004) distinguished between two views of context**
- **Representational view:**
  - Context is information that can be described using a set of “appropriate” attributes that can be observed and are distinguishable from features describing the underlying activity undertaken by the user within the context
- **Interactional View of Context**
  - The scope of contextual features is defined dynamically, and is occasioned rather than static
  - Rather than assuming that context defines the situation within which an activity occurs, there is a cyclical relationship between context and activity:
    - Context gives rise to the activity and activity changes the context

Representational View: Assumptions & Implications

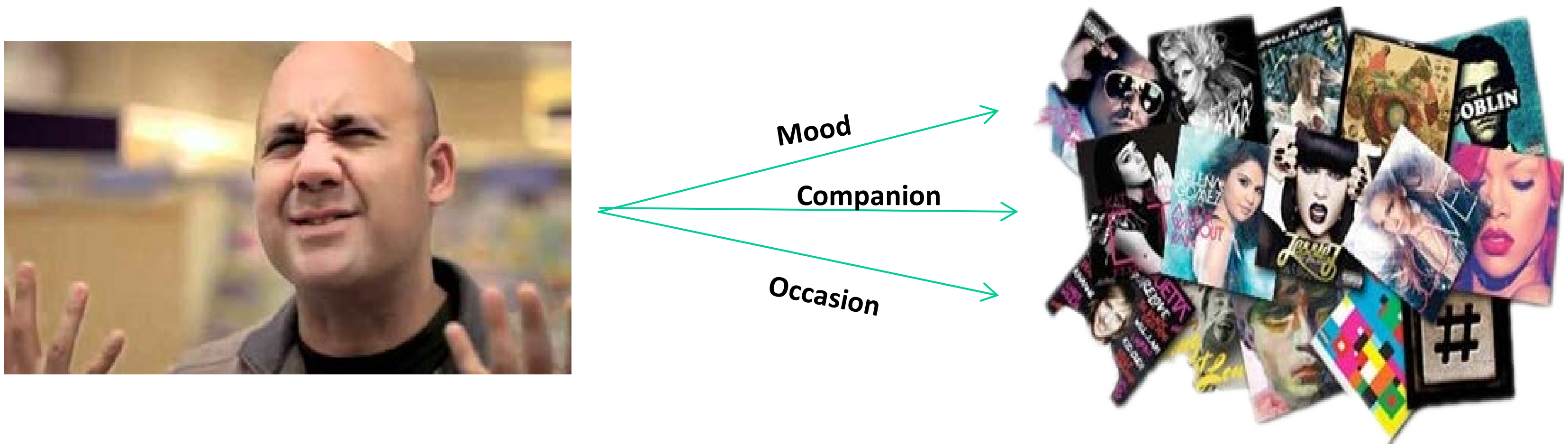
- **Context can be represented as an explicit, enumerated set of static attributes (i.e., it's "extensional")**
  - Typically attributes are predefined based on the characteristics of the domain and environment
    - E.g., time, date, location, mood, task, device, etc.
  - Contextual variable can have associated structure
    - E.g., Sunday < Weekend
- **Implications:**
  - Must identify and acquire contextual information as part of data collection before actual recommendations are made
  - Relevant contextual variables (and their structures) must be identified at the design stage
- **Drawbacks**
  - The "qualification problem" – as in AI & Knowledge Representation
  - Context is static. No "situated action"

Interactional View: Assumptions & Implications

- **Properties of Context**
  - Context gives rise to a behaivor that is observable, though context itself may not be observable (it’s “intensional”)
    - Context exists (usually implicitly) in relation to the ongoing interaction of the user with the system
  - not static
    - Can be derived: a stochastic process with d states $\{c_1, c_2, \ldots, c_d\}$ representing different contextual conditions
- **Context aware recommendation**
  - Explicit representation of context may not be as important as
    - recognizing behavior arising from the context
    - adapting to the needs of the user within the context
- **Drawback:** Ability to explain recommendations

Context-Aware RS (CARS)

- **Traditional RS:** Users × Items → Ratings
- **Contextual RS**: Users × Items × Contexts → Ratings



Example of Ratings Data in Representational Framework

User	Movie	Time	Location	Companion	Rating
U1	*Titanic*	Weekend	Home	Family	4
U2	*Titanic*	Weekday	Home	Family	5
U3	*Titanic*	Weekday	Cinema	Friend	4
U1	*Titanic*	Weekday	Home	Friend	?

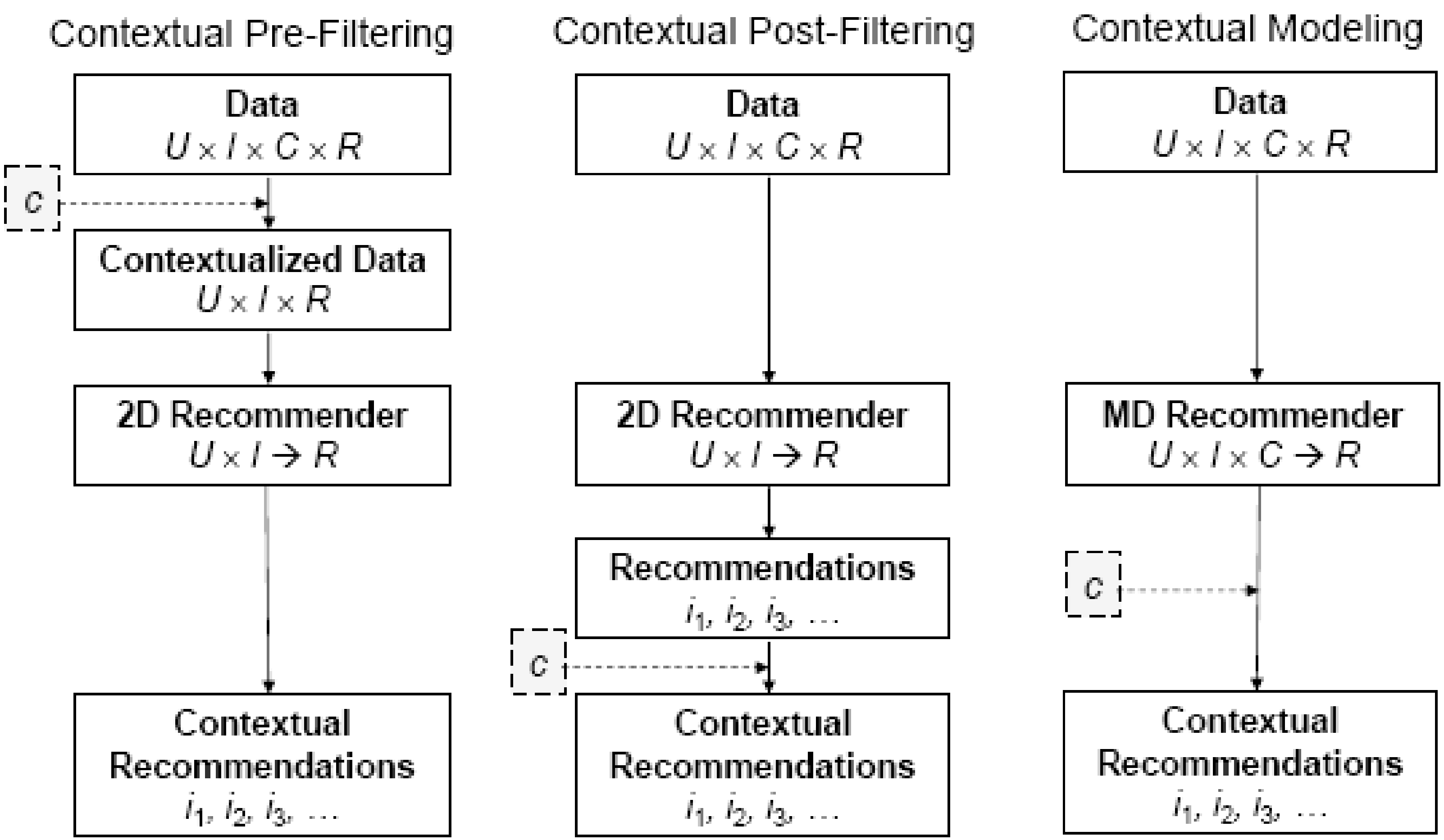
- But what constitutes context:
  - <Weekday, Home, Friend>?
  - A subset of contextual variables, e.g., <Time, Location>?
- How do we select or weigh the most relevant set of variables

CARS Architectural Models

- **Three types of Architecture for using context in recommendation (Adomavicius, Tuzhilin, 2008)**
  - Contextual Pre-filtering
    - Context information used to select relevant portions of data
  - Contextual Post-filtering
    - Contextual information is used to filter/constrain/re-rank final set of recommendations
  - Contextual Modeling
    - Context information is used directly as part of learning preference models
- **Variants and combinations of these are possible**

CARS Architectural Models

From Adomavicius, Tuzhilin, 2008



Contextual Pre-Filtering Challenges

- Context Over-Specification
  - Using an exact context may be too narrow:
    - Watching a movie with a girlfriend in a movie theater on Saturday
  - Certain aspects of the overly specific context may not be significant (e.g., Saturday vs. weekend)
  - Sparsity problem: overly specified context may not have enough training examples for accurate prediction
- Pre-Filter Generalization
  - Different Approaches
  - “Roll up” to higher level concepts in context hierarchies
    - E.g., Saturday → weekend, or movie theater → any location
  - Use latent factors models or dimensionality reduction approaches (Matrix factorization, LDA, etc.)

Contextual Post-Filtering

- Contextual Post-Filtering is generally heuristic in nature
  - Basic Idea: Treat the context as an additional constraint
  - Many different approaches are possible
- Example: Filtering Based on Context Similarity
  - Can be represented as a set of features commonly associated with the specified context
  - Adjust the recommendation list by favoring those items that have more of the relevant features
  - Similarity-based approach (but the space of features may be different than the one describing the items)
- Example: Filtering Based on Social/Collaborative Context Representation
  - Mine social features (e.g., annotations, tags, tweets, reviews, etc.) associated with the item and users in a given context *C*
  - Promote items with frequently occurring social features from *C*

Contextual Recommendation Approaches

- Many different approaches in recent years, both in representational and interactional frameworks
  - Extensions of standard collaborative filtering
    - *CF after Item / user splitting pre-filters*
    - *Differential Context Modeling*
  - Heuristic distance-based approaches
    - Extend items-item, user-user similarities to contextual dimensions
    - Requires, possibly domain specific, similarity/distance metrics for various contextual dimensions
  - Approaches based on matrix/tensor factorization
    - Model the data as a tensor and apply higher-order factorization techniques (HoSVD, HyPLSA, etc) to model context in a latent space
    - *Context-Aware Matrix Factorization*
  - *Probabilistic latent variable context models*

Highlighted Approach: Item / User Splitting

Context-Aware Splitting Approaches

- **Generally based on contextual pre-filtering**
  - May be combined with contextual modeling techniques
- **Goal: produce a 2D data set that incorporates context information associated with preference scores**
  - Advantage: can use a variety of well-known traditional recommendation algorithms in the modeling phase
  - Disadvantages:
    - Determining the variables based on which to split
    - May lead to too much sparsity
- **There are three approaches to splitting:**
  - Item Splitting (Baltrunas et al., 2009, RecSys)
  - User Splitting (Baltrunas et al., 2009, CARS)
  - UI Splitting (Zheng et al., 2013)

Item Splitting and User Splitting

- **Item Splitting**
  - Assumption: the nature of an item, from the user's point of view, may change in different contextual conditions (values of contextual variables)
  - Hence we may consider the item as multiple items – one for each contextual condition
- **User splitting**
  - It may be useful to consider one user as multiple users, if he or she demonstrates significantly different preferences in different contexts
- **Good deal of recent work on these approaches:**
  - L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. RecSys 2009
  - L. Baltrunas and X. Amatriain. Towards time-dependent recommendation based on implicit feedback. RecSys 2009 Workshop on CARS
  - A. Said, E. W. De Luca, and S. Albayrak. Inferring contextual user profiles – improving recommender performance. RecSys 2011 Workshop on CARS

Example: Item Splitting

User	Movie	Rating	Time	Location	Companion
U1	M1	3	Weekend	Home	Friend
U1	M1	5	Weekend	Theater	Spouse
U1	M1	?	Weekday	Home	Family

Assume Location (Home vs. Theater) is the best split condition

User	Item	Rating
U1	M11	3
U1	M12	5
U1	M11	?

M11: M1 seen at home; M12 = M1 seen not at home

Example: User Splitting

User	Movie	Rating	Time	Location	Companion
U1	M1	3	Weekend	Home	Friend
U1	M1	5	Weekend	Theater	Alone
U1	M1	?	Weekday	Home	Family

Assume Companion (Family vs. Non-Family) is the best split condition

User	Item	Rating
U12	M1	3
U12	M1	5
U11	M1	?

U11: U1 saw the movie with family; U12 = U1 saw the movie alone or with a friend

User-Item (UI) Splitting

- **New approach combining User and Item splitting**
  - The process is simply an application of item splitting followed by user splitting on the resulting output
  - Y. Zheng, B. Mobasher, R. Burke. Splitting approaches for Context-aware Recommendation. (To appear)

- **Using the same conditions as previous example:**

Item Splitting

User	Item	Rating
U1	M11	3
U1	M12	5
U1	M11	?

User Splitting

User	Item	Rating
U12	M1	3
U12	M1	5
U11	M1	?

UI Splitting

User	Item	Rating
U12	M11	3
U12	M12	5
U11	M11	?

Determining the Best Conditions for Splitting

- **Impurity Criteria**
  - Statistical criteria to evaluate whether items are being rated significantly differently under an alternative contextual condition
  - e.g. the location → home vs. not-home

- **Commonly used criteria**
  - t_{mean} (t-test)
  - t_{prop} (z-test)
  - t_{chi} (chi-square test)
  - t_{IG} (Information gain)

- **Thresholds:**
  - P-value is used to judge significance

Splitting Criteria

- **Example: t_{mean}**
  - Uses the two-sample *t* test and computes how significantly different are the means of the ratings in the two rating subsets, when the split *c* (context condition) is used
  - $$t_{mean} = \left| \frac{\mu_{i_c} - \mu_{i_{\bar{c}}}}{\sqrt{s_{i_c}/n_{i_c} + s_{i_{\bar{c}}}/n_{i_{\bar{c}}}}} \right|$$
  - *S* is the rating variance, and *n* is the number of ratings in the given contextual condition, *c* and $\bar{c}$ denote alternative conditions
  - The bigger the *t* value of the test is, the more likely the difference of the means in the two partitions is significant
  - This process is iterated over all contextual conditions

Splitting Criteria Example

- t_{mean} ; condition: time= weekend and not weekend

$$t_{mean} = \left| \frac{\mu_{i_c} - \mu_{i_{\bar{c}}}}{\sqrt{s_{i_c}/n_{i_c} + s_{i_{\bar{c}}}/n_{i_{\bar{c}}}}} \right|$$

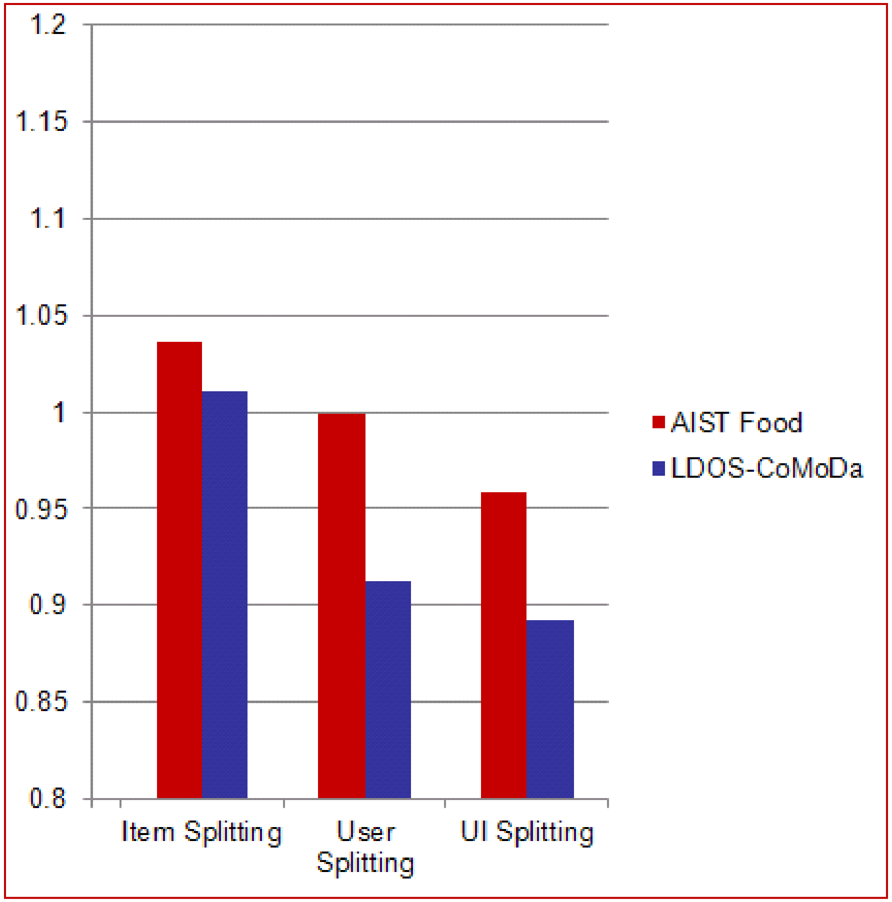
User	Item	Rating	Time	Location	Companion
U1	T1	3	Weekend	Home	Sister
U1	T1	5	Weekend	Cinema	Girlfriend
U2	T1	4	Weekday	Home	Family
U3	T1	2	Weekday	Home	Sister

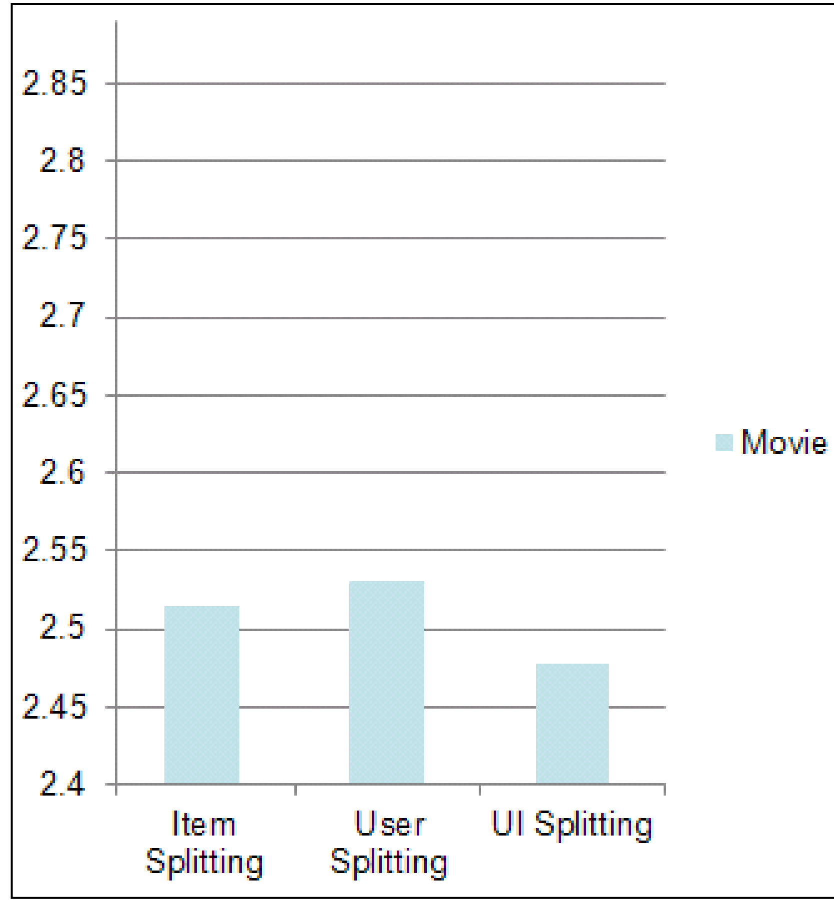
- mean1=4 , mean2 =3 , s1 =1 , s2 =1 , n1= 2, n2=2
- Impurity criteria t_{mean} = (4-3)/1 = 1
- P-value of t-test used to determine significance (0.05 as threshold)

Example Results

	Food Data	Movie Data
Ratings	6360	1010
Users	212	69
Items	20	176
Contexts	Real hunger (full/normal/hungry) Virtual hunger	Time (weekend, weekday) Location (home, cinema) Companions (friends, alone, etc)
Contexts-linked Features	User gender food genre, food style, food stuff	User gender, year of the movie
Density	Dense in contexts	Sparse in contexts

Example Results (RMSE)





Results based on splitting followed by Matrix Factorization (discussed next)

Highlighted Approach:
Differential Context Modeling

Differential Context Modeling

User	Movie	Time	Location	Companion	Rating
U1	*Titanic*	Weekend	Home	Family	4
U2	*Titanic*	Weekday	Home	Family	5
U3	*Titanic*	Weekday	Cinema	Friend	4
U1	*Titanic*	<u>Weekday</u>	<u>Home</u>	<u>Friend</u>	?

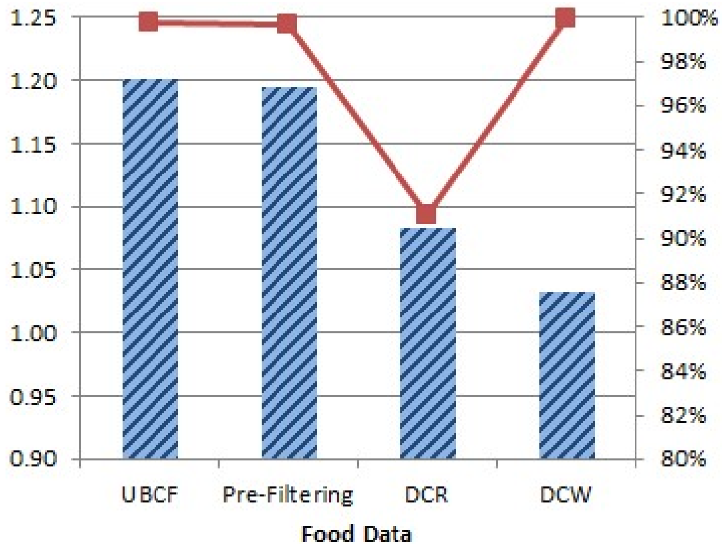
- Context Matching ➔ only the exact context <Weekday, Home, Friend>?
- Context Selection ➔ use only the most relevant contexts
- Context Relaxation ➔ relax set of constraints (dimensions) defining the context, e.g. use only time
- Context Weighting ➔ use all dimensions, but weight them according to co-occurrence relationships among contexts

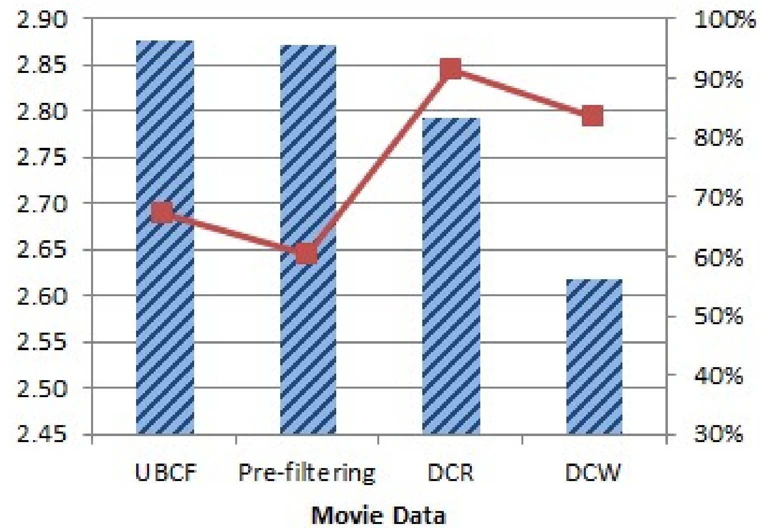
Differential Context Weighting

User	Movie	Time	Location	Companion	Rating
U1	*Titanic*	Weekend	Home	Friend	4
U2	*Titanic*	Weekday	Home	Friend	5
U3	*Titanic*	Weekday	Cinema	Family	4
U1	*Titanic*	<u>Weekday</u>	<u>Home</u>	<u>Family</u>	?

- **Goal: Use all dimensions, but weight them based on the similarity of contexts**
  - Assumption: the more similar two contexts are, the more similar the ratings will be in those contexts
  - Similarity can be measured by Weighted Jaccard similarity $J(c, d, \sigma) = \frac{\sum_{f \in c \cap d} \sigma_f}{\sum_{f \in c \cup d} \sigma_f}$
  - Example:
    - c and d are two contexts (two red regions in the Table)
    - σ is the weighting vector <w1, w2, w3> for three dimensions.
    - Assume they are equal weights, w1 = w2 = w3 = 1
    - J(c, d, σ) = # of matched dimensions / # of all dimensions = 2/3

Predictive Performance





Blue bars are RMSE values, ***Red lines*** are coverage curves

Findings:

1) t-test shows DCW works better significantly in movie data
2) DCW can further alleviate sparsity of contexts
3) DCW offers better coverage over baselines!

Differential Context Modeling

- **Some relevant work**
  - Y. Zheng, R. Burke, B. Mobasher. "Differential Context Relaxation for Context-aware Travel Recommendation". In EC-WEB, 2012 [DCR]
  - Y. Zheng, R. Burke, B. Mobasher. "Optimal Feature Selection for Context-Aware Recommendation using Differential Relaxation". In ACM RecSys Workshop on CARS, 2012 [DCR + Optimizer]
  - Y. Zheng, R. Burke, B. Mobasher. "Recommendation with Differential Context Weighting". In UMAP, 2013 [DCW]
  - Y. Zheng, R. Burke, B. Mobasher. "Differential Context Modeling in Collaborative Filtering". In SOCRS-2013, DePaul University, Chicago, IL, May 31, 2013 [DCM]
- **Future Work**
  - Try other similarity measures of contexts instead of the simple Jaccard
  - Introduce semantics into the similarity of contexts to further alleviate the sparsity of contexts, e.g., Rome is closer to Florence than Paris
  - Parallel PSO to speed up optimizer
  - Additional recommendation algorithms on DCR or DCW

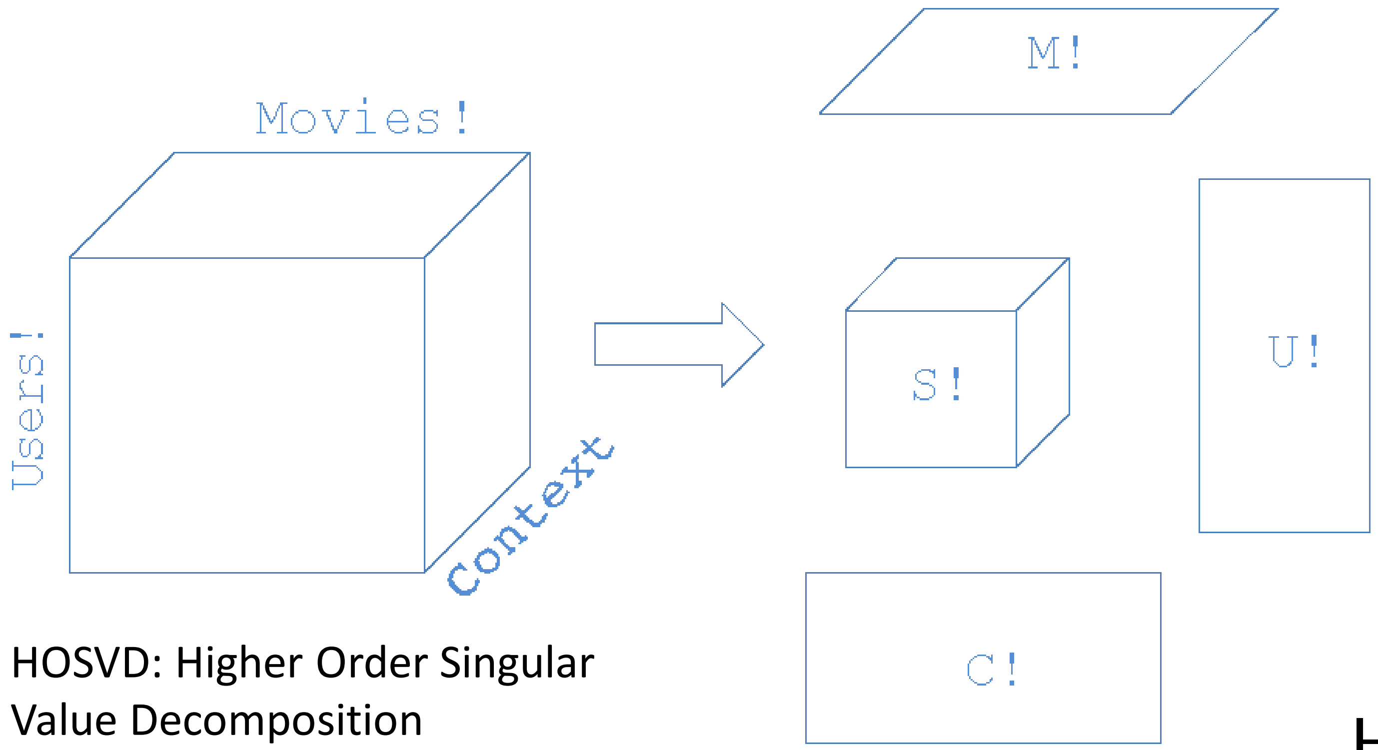
Highlighted Approach:

Contextual Modeling using Matrix Factorization

Contextual Modeling via Factorization

- Recent approaches to contextual modeling attempt to fit the data using various regression models
  - Prominent examples: Tensor Factorization (TF), Factorization Machines (FM)
- Tensor Factorization
  - Extends the two-dimensional matrix factorization problem into an multi-dimensional version of the same problem
  - Multi-dimensional matrix is factored into lower-dimensional representation, where the user, the item and each contextual dimension are represented with a lower dimensional feature vector
- TF can introduce a huge number of model parameters that must be learned using the training data
  - the number of model parameters grow exponentially with the number of contextual factors

Recall: Tensor Factorization



HOSVD: Higher Order Singular Value Decomposition

HOSVD Model

$U \in \mathbb{R}^{n \times d_U}$, $M \in \mathbb{R}^{m \times d_M}$ and $C \in \mathbb{R}^{c \times d_C}$

$S \in \mathbb{R}^{d_U \times d_M \times d_C}$

$$R[U, M, C, S] := L(F, Y) + \Omega[U, M, C] + \Omega[S]$$

Context-Aware Matrix Factorization

- **Recall the difference between MF and BiasMF:**
  - Standard MF: Predict(u, i) = $q_i^{\top} p_u$
  - BiasMF: Predict(u, i) = $q_i^{\top} p_u + \mu + b_u + b_i$
  - b_u and b_i are user bias and item bias respectively, μ is the global mean rating

- **Context-Aware MF**
  - CAMF replaces the simple item bias, b_i, by the interaction between item and contextual conditions
    - Predicted rating will now be also a function of contextual conditions, c_1, …, c_k giving rise to a particular context
    - The item bias is modeled by how that item is rated in different contexts
    - i.e., sum of biases for the given item across all contextual conditions, c_1, …, c_k
    - Different levels of granularity in aggregating item biases can lead to different variants of CAMF

Recall: Factorization Machines

- **Approach: Treat input as a real-valued feature vector**
  - Model both linear and pair-wise interaction of k features (i.e. polynomial regression)
  - Traditional machine learning will overfit
  - Factor pairwise interactions between features
  - Reduced dimensionality of interactions promote generalization
  - Different matrix factorizations become different feature representations
  - Tensors: Additional higher-order interactions
- **Combines “generality of machine learning/regression with quality of factorization models”**

Recall: Factorization Machines

- **Two categorical variables (u, i) encoded as real values:**

Feature vector $\mathbf{x}$

	User				Movie				
	A	B	C	...	TI	NH	SW	ST	...
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...

- **FM becomes identical to MF with biases:**

$$f(\mathbf{x}) = b + w_u + w_i + \mathbf{v}_u^T \mathbf{v}_i$$

From Rendle (2012) KDD Tutorial

Recall: Factorization Machines

- **Can easily map different factorization models in FM:**

Recommender Data

(A,TI,H,{C},5)
(A,NH,S,{},3)
(A,SW,N,{B,C},1)
(B,SW,N,{A,C},4)
(B,ST,H,{},5)
(C,TI,S,{A},1)
(C,SW,H,{A,B},5)

→

Feature vector $\mathbf{x}$

	User			Movie				Mood			Watched with		
	A	B	C	TI	NH	SW	ST	S	N	H	A	B	C
$\mathbf{x}^{(1)}$	1	0	0	1	0	0	0	0	0	1	0	0	1
$\mathbf{x}^{(2)}$	1	0	0	0	1	0	0	1	0	0	0	0	0
$\mathbf{x}^{(3)}$	1	0	0	0	0	1	0	0	1	0	0	0.5	0.5
$\mathbf{x}^{(4)}$	0	1	0	0	0	1	0	0	1	0	0.5	0	0.5
$\mathbf{x}^{(5)}$	0	1	0	0	0	0	1	0	0	1	0	0	0
$\mathbf{x}^{(6)}$	0	0	1	1	0	0	0	1	0	0	1	0	0
$\mathbf{x}^{(7)}$	0	0	1	0	0	1	0	0	0	1	0.5	0.5	0

Target y

5	$y^{(1)}$
3	$y^{(2)}$
1	$y^{(2)}$
4	$y^{(3)}$
5	$y^{(4)}$
1	$y^{(5)}$
5	$y^{(6)}$

Context-aware recommendation data (left side) is transformed into a prediction problem from real-valued features by encoding the categorical and set categorical variables.

From Rendle SIGIR, 2011

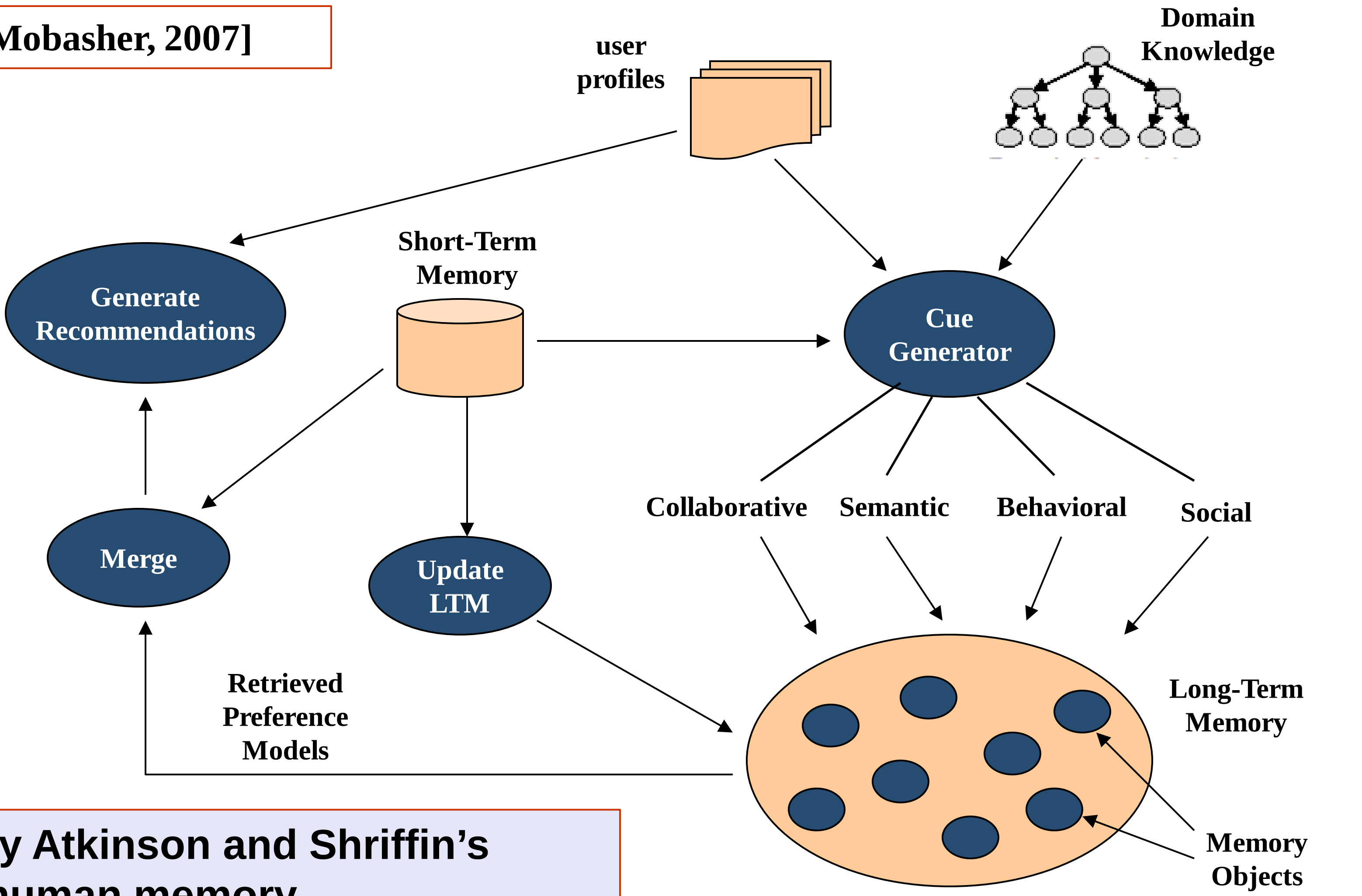
Some Additional References for MF Based Approaches

- Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model. KDD 2008
- Yehuda Koren, Collaborative filtering with temporal dynamics. KDD 2009
- A. Karatzoglou , X. Amatriain , L. Baltrunas , N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. RecSys 2010
- L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. RecSys 2011
- L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Luke, and R. Schwaiger. InCarMusic: Context-aware music recommendations in a car. ECWeb 2011
- Rendle, Gantner, Freudenthaler, Schmidt-Thieme: Fast context-aware recommendations with factorization machines. SIGIR 2011

The Interactional View of Context

An Interactional Framework for Contextual Recommendation

[Anand and Mobasher, 2007]



Inspired by Atkinson and Shriffin's model of human memory

Contextual Recommendation Generation

- **Explicit or implicit preferences for items from the active interaction are stored in the STM**
- **Contextual cues are derived from this data and used to retrieve relevant preference models from LTM**
  - Relevant = belong to the same context as the active interaction
- **Merged with STM preferences and used to predict preferences for unseen items**
- **New Observations used to update preference models in LTM**
- **Lots of variations on how LTM preference models are organized**
  - based on ontological or semantic relationships
  - aggregate models based on similarities among users
  - based on connections in social or information networks
  - etc.

Characteristics of the Framework

- **The Framework Emphasizes**
  - The distinction between local, transient preference models in STM and the long-term established models in LTM
  - The importance of user's interaction with the system in deriving contextual cues
  - The mutually reinforcing relationship between user activity and the context model
    - Emphasizes the dynamic nature of context
- **Does Not Emphasize**
  - Explicit knowledge-based representation of contextual attributes
  - A rigid formulation of contextual modeling approaches
    - Very general framework and many implementations are possible (we will look at several next)

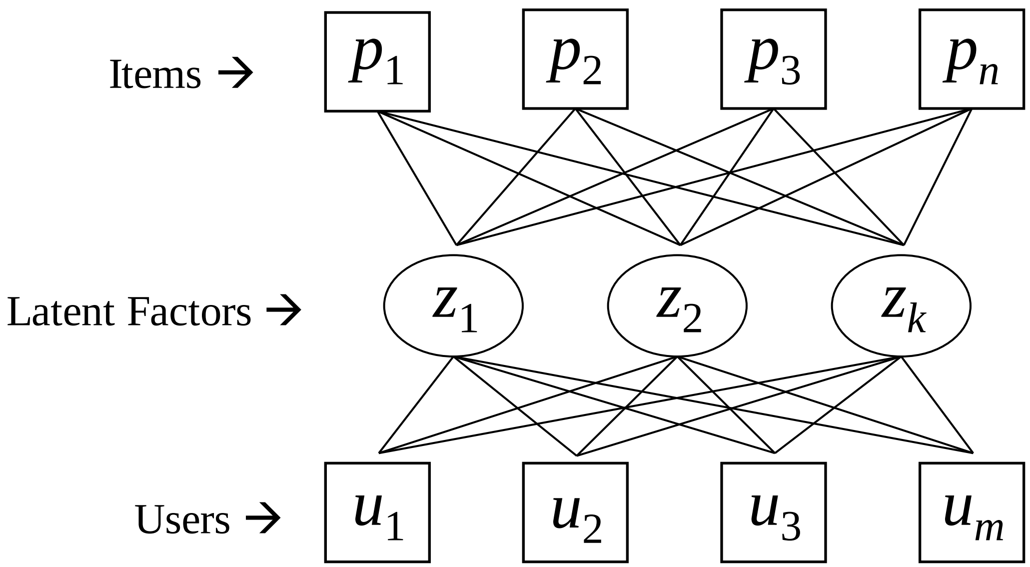
Highlighted Approach:
Latent Variable Context Models

Latent Variable Context Model

- **Generative approach to modeling user context**
- **Basic assumption:**
  - users' interactions involve a relatively small set of contextual states that can "explain" users' behavior at different points during their interactions
- **Have been used effectively in applications involving user's performing informational or functional tasks**
- **Contexts correspond to tasks/activities and are derived as latent factors from the observed user data**
- **Latent variable models such as PLSA or LDA can be used to automatically characterize these "contexts," as well as their relationships to items or users**

Latent Variable Models

- **Assume the existence of a set of latent (unobserved) variables (or factors) which "explain" the underlying relationships between two sets of observed variables.**



Advantages:

Probabilistically determine the association between each latent factor and items, or between each factor and users.

In navigational/interactional data, the latent factors correspond to distinguishable patterns usually associated with performing certain informational or functional tasks. Context = Task!

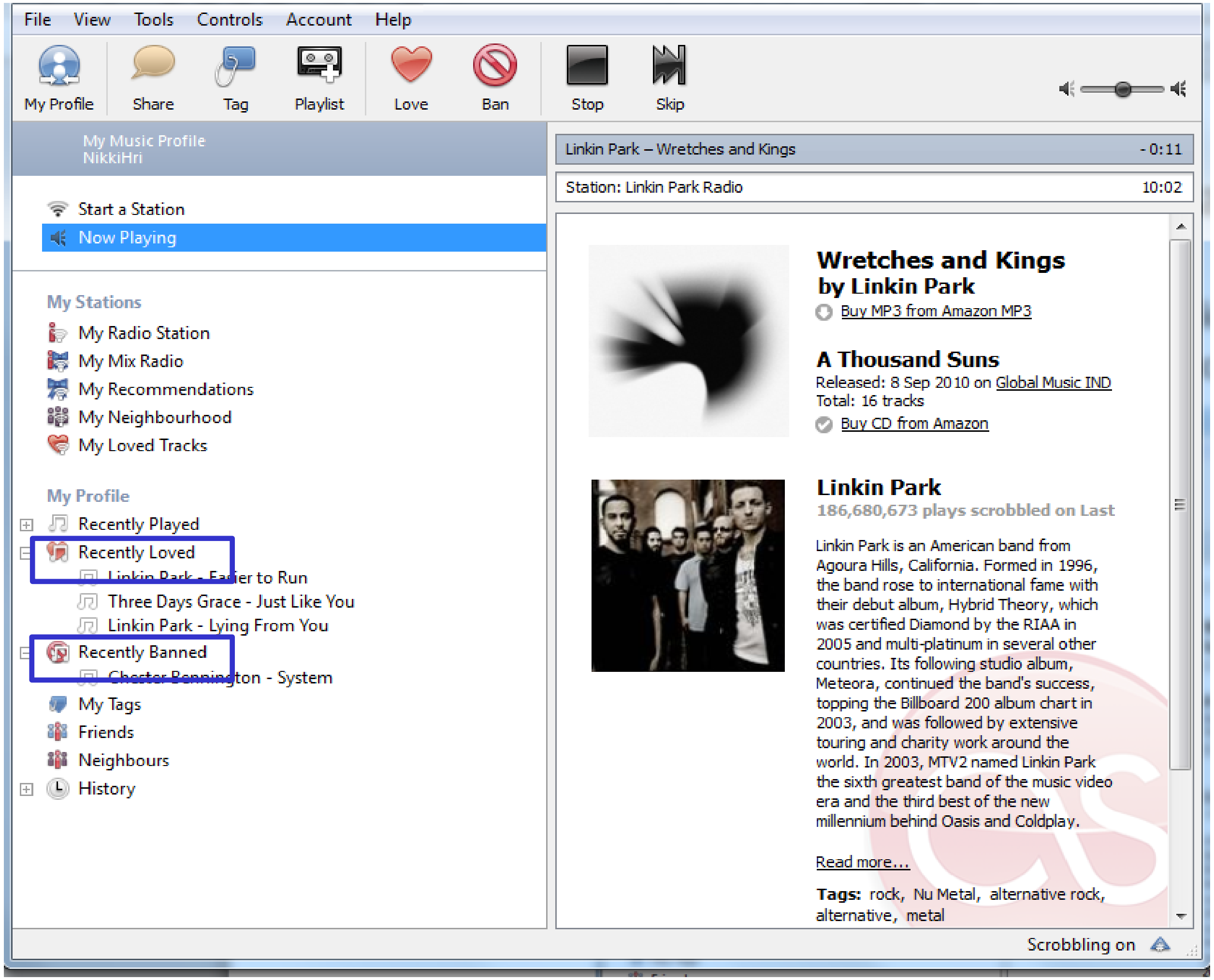
Example Implementation: Inferring Latent Contexts From Social Annotation

Hariri, Mobasher, Burke, RecSys 2012

- **Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns**
- **In the domain of music recommendation:**
  - Context is usually not fully observable & may depend on many factors
    - Types of user activity (exercising, relaxing, driving, dancing)
    - User’s moods or emotional states
    - Occasion or social setting
  - Contextual information is dynamic and should be inferred from users' interactions with the system such as:
    - Liking/disliking/skipping songs
    - Creating different stations by selecting different track seeds or artists
- **Different applications:**
  - Song recommendation, Playlist generation, Playlist recommendation

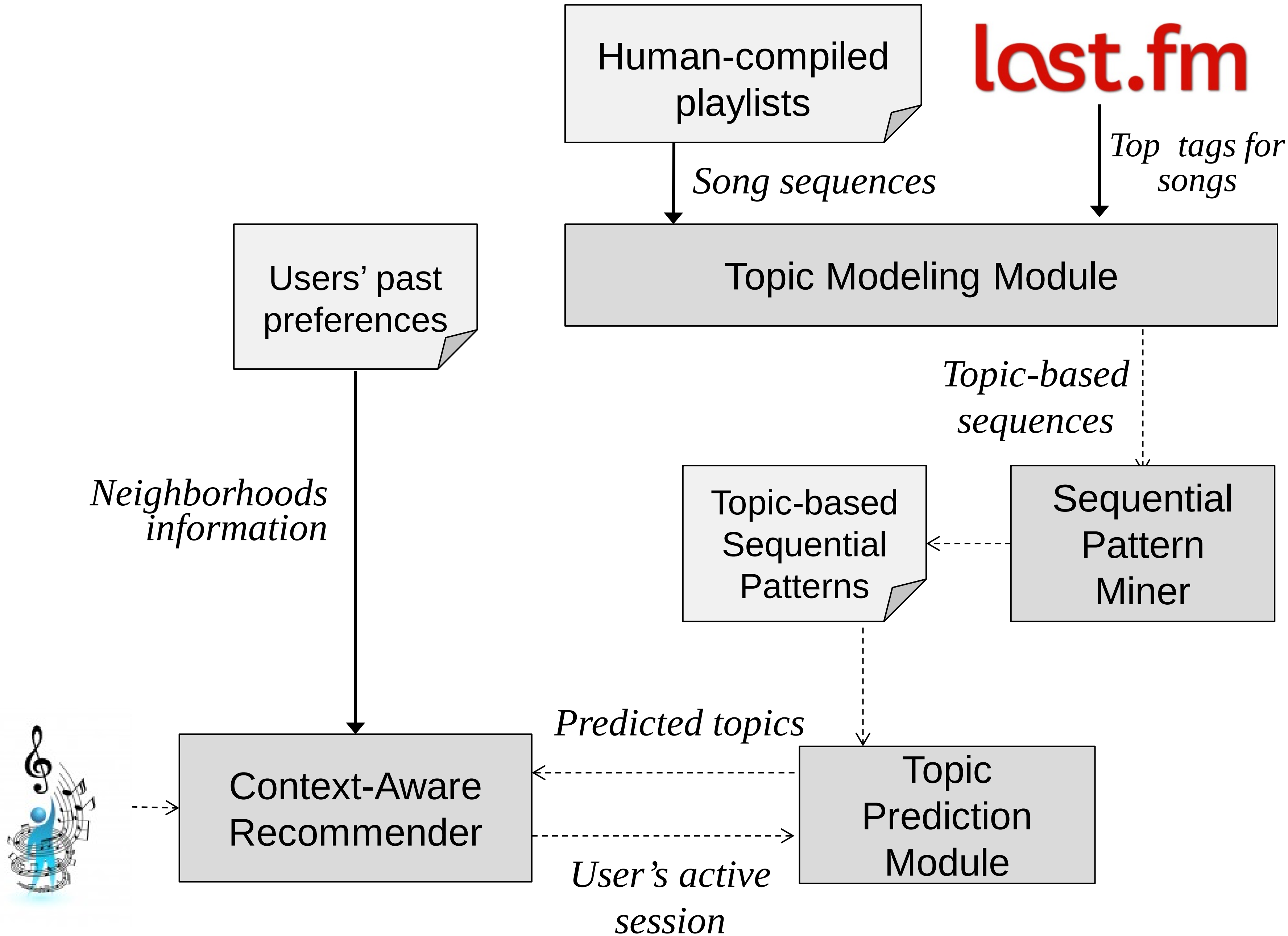
User Interactions

Context is reflected in the sequence of songs liked/disliked or played by the user in her current interaction with the system



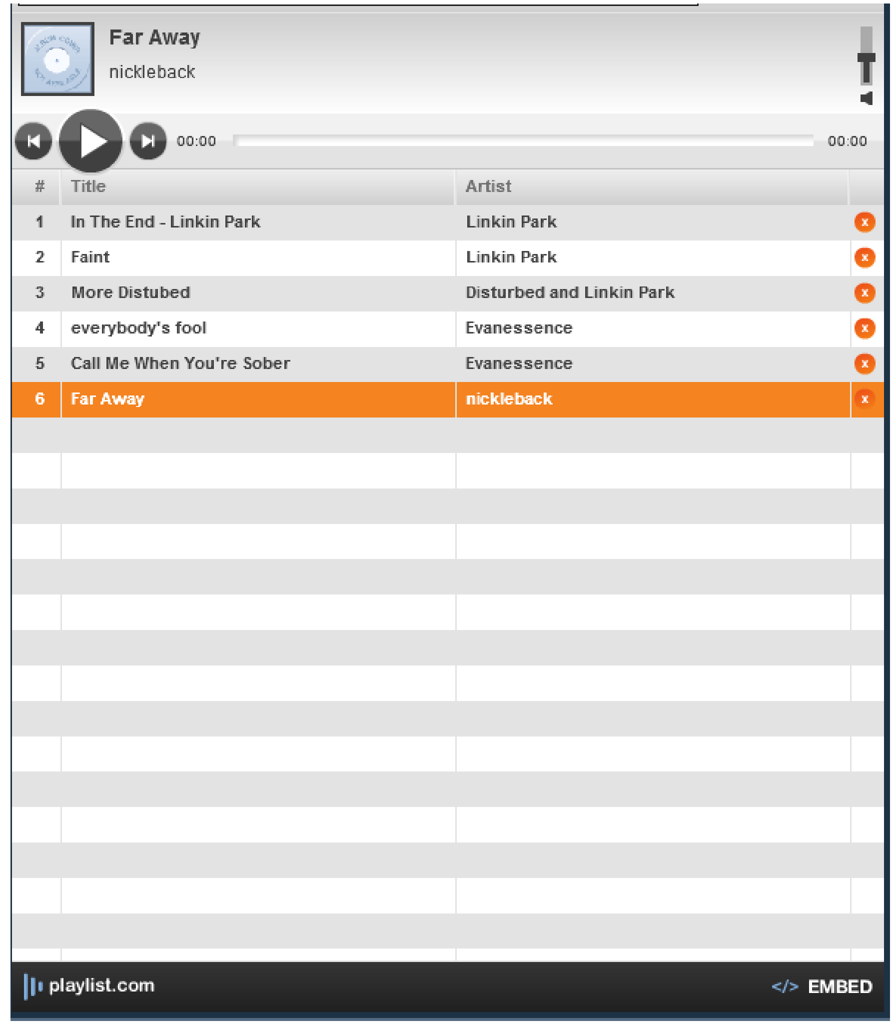
Goals of the Contextual Music Recommendation Framework

- Infer users' contextual states based on the most recent sequence of songs liked, played, or added to a playlist
- Utilize sequential information in user's history of interaction to identify and predict changes in context
- Adapt the system's recommendations to user's interest corresponding to these changes in context


Human-compiled playlists
last.fm
Song sequences
Top tags for songs
Users' past preferences
Topic Modeling Module
Topic-based sequences
Neighborhoods information
Topic-based Sequential Patterns
Sequential Pattern Miner
Predicted topics
Context-Aware Recommender
Topic Prediction Module
User's active session

Example Usage Scenario: Playlist Generation

1. The user selects an initial sequence of songs for the playlist.
2. The system *infers* user's context and recommends a set of songs.
3. The user adds one of the recommendations (or a new song outside the recommendation set) to the playlist
4. The system updates its knowledge about the user's preferences before the next interaction



Topic Modeling for Song Context Representation

- **We use LDA topic modeling approach to map user's interaction sequence to a sequence of latent topics**
  - Better at capturing more general trends in user's interests
- **The latent topics are generated from the top most frequent tags associated with songs**
  - Tags obtained from social tagging Web sites such as last.fm.
  - Tags may characterize song features, user’s situation, mood, etc.
  - For LDA, songs are taken as documents and tags as words
  - After fitting the topic model for *K* topics, the probability distribution over topics can be inferred for any given song
  - For each song, the set of dominant topics are selected that have probabilities higher than a specific threshold value

Top Most Frequent Tags for a Sample of Topics

Topic#1	Topic#2	Topic#3	Topic#4	Topic#5	Topic#6	Topic#7
ambient	latin	death	60s	chill	beautiful	electronic
instrumental	world	thrash	oldies	downtempo	sad	electronica
soundtrack	streamable	black	roll	chillout	mellow	house
classical	spanish	heavy	50s	christmas	melancholy	techno
beautiful	para	doom	rockabilly	lounge	acoustic	tranc
age	bossa	brutal	top	electronic	chill	electro
chillout	fusion	melodic	500	trip-hop	soft	bass
experimental	musica	california	radio	electronica	slow	drum
movie	que	power	rolling	trip	melancholic	ambient
atmospheric	nova	progressive	1960s	ambient	favourite	beat
world	brazilian	gods	rhythm	hop	chillout	idm
ethereal	african	seixas	time	easy	ballad	experimental
chill	party	speed	elvis	cool	singer-songwriter	club
calm	brasil	swedish	soundtrack	sexy	life	minimal
electronic	espanol	old	american	radio	easy	party

An Example Playlist (Mapped to Tags and Topics)

Time	Popular Tags	Dominant Topics
1	Singer-songwriter, mellow, relaxing, chill, male vocalist, easy listening, acoustic, 00's, guitar, rock, happy	6
2	Singer-songwriter, chill, acoustic, mellow, rock, summer, surf, male vocalist, pop, relaxing, guitar, happy	6
3	singer,-songwriter, indie rock, folk, acoustic, mellow, chill out, relaxing, bittersweet, lo-fi	6, 20, 23
4	Alternative rock, ballads, calm, beautiful, nice, soundtrack, favorites	6, 28
5	Electronic, electronica, French, chill out, trip-hop, ambient, down-tempo, sexy, 90s, alternative, easy listening, guitar, mellow, relax, female vocal	7, 5
6	Soundtrack, 90s, alternative, atmospheric, female vocalist, indie, dreamy	23
7	Singer-songwriter, acoustic, chill, alternative, rock, male vocalist, easy listening, driving	6, 25
8	Cover, Beatles cover, rock, 90s, soundtrack, brass , pop rock, alternative, rock, folk, brass	30, 18
9	Indie, rock, acoustic, 90s, cover, mellow, pop, folk, dreamy, singer-songwriter, sad-core, summery, sweet, alternative rock, female vocalist	6, 20

Sequential Pattern Mining and Topic Prediction

- **Using a training set of human-compiled playlists, sequential patterns are mined over the set of corresponding latent topic sequences**
  - Each pattern represents a frequent sequence of transitions between topics/contexts
- **Why Topic Level Sequential Patterns?**
  - Mining SPs on topics instead of songs is useful in capturing user interests based on common characteristics of the current context
  - Makes it easier to track and detect changes in the users' preferences due to changes in contextual states
  - Topic-based patterns are useful in managing the cold start problem: a new songs may still match topic-based patterns

Song Recommendation Based on Contextual Post-filtering

- **The predicted topics are used to contextualize the recommendations**

$$contextScore(h_u, s) = \frac{\sum_{t_i \in predictedTopics(h_u)} p(t_i|s)}{|predictedTopics(h_u)|}$$

  - *ContextScore*(h_u, *s*) represents the suitability of song *s* for the current context of user *u* (determined based on user's active session, h_u)

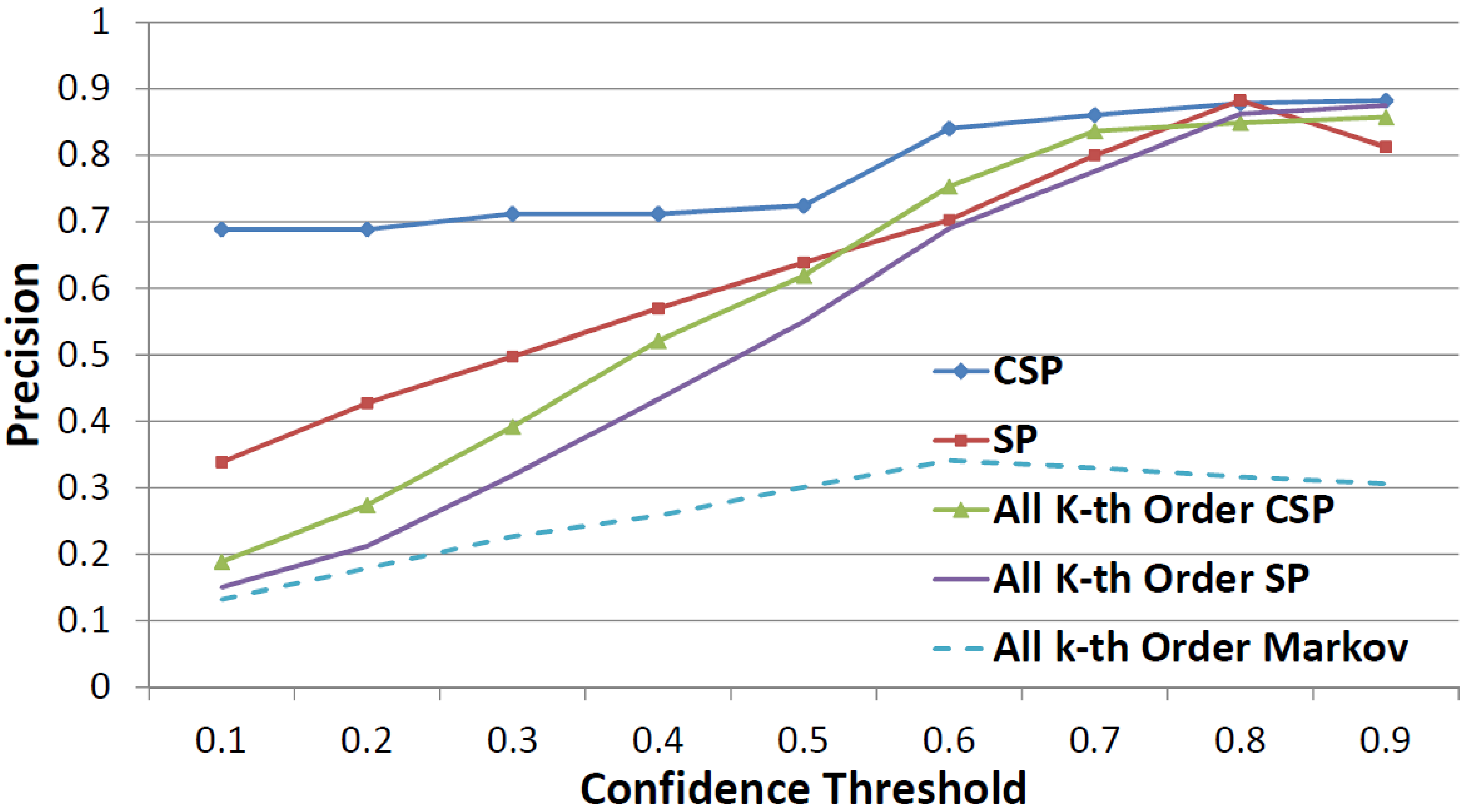
- **Next, recommendations are re-ranked using the contextual information**

$$predictionScore(s) = (contextScore(s) + \alpha_1) \cdot (CFScore(s) + \alpha_2)$$

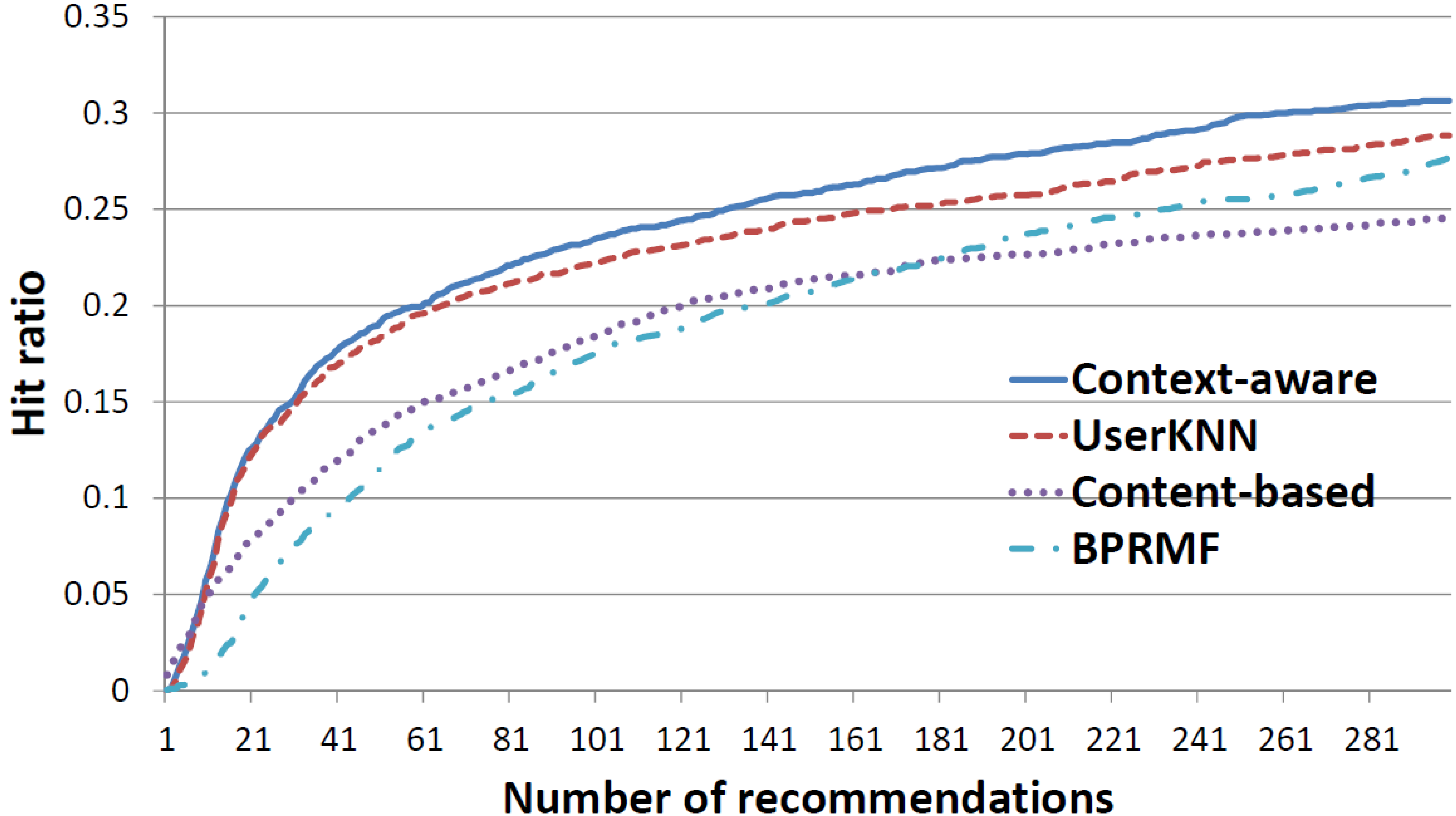
Evaluation

- **Dataset and Methodology:**
  - 28,963 user-contributed playlists from *Art of the Mix* website in January 2003
  - This dataset consists of 218,261 distinct songs for 48,169 distinct artists
  - Top tags were retrieved from the last.fm website for about 71,600 songs in our database
  - 48K songs with min. of 5 tags were used to build a 30-topic LDA model
  - The last *w* = 7 songs were selected as the user's active session, the last song was removed and the dominant topics associated with that song were used as target set

Topic Prediction Precision



Song Recommendation Performance



Conclusions

- **Incorporating context in recommendation generation *can* improve the effectiveness of recommender systems**
- **What does it take?**
  - **In representational models:** careful selection of relevant contextual attributes for the specific domain (the classic knowledge engineering task)
  - **In Interactional Models:** effective methods for extraction of contextual cues from user behavior & ways of coping with domains that don’t lend themselves to user interactions
- **Work on Interactional Models Suggests:**
  - observable behavior is “conditioned” on the underlying context
  - The context can be inferred (and predicted) effectively in certain kinds of applications
  - The integration of semantic knowledge or social cues with user activity can be particularly effective in contextual user modeling

?