Contextual User Modeling for Recommendation

Bamshad Mobasher

Center for Web Intelligence

School of Computing
DePaul University, Chicago

2nd Workshop on Context Aware Recommender Systems
Barcelona, Spain, September 26, 2010

Context in Recommendation

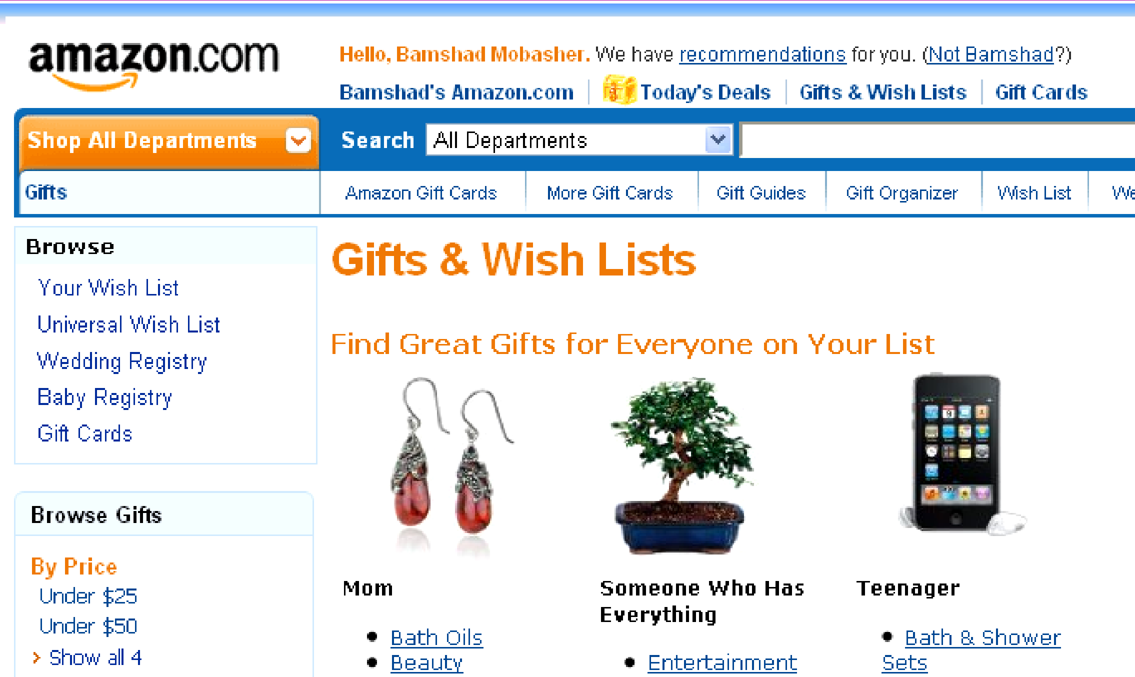
- **Recommendation Scenario**
  - Steve’s purchases on Amazon:
    - mystery-detective fiction “Da Vinci Code” (for himself)
    - “Python Programming” (for work)
    - “Green Eggs and Ham” (gift for his daughter)
  - How should we represent Steve’s *interest in books*?
  - System needs to know the difference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

Anatomy of a Famous Example

- **Jack buys a book on pregnancy for his friend Jill who is expecting**
- **The purchase becomes part of Jack's profile and in subsequent visits, he gets recommendations about baby cloths, child rearing, etc.**

- **Amazon' approach: distinguish between the task of gift buying versus buying items for oneself.**

Anatomy of a Famous Example



- Is this a problem of context?
- Or, a problem in user profiling?
- Are they the same thing?

- Goal of identifying gifts, is to exclude them from profile not to change context
- Once excluded, no context is used for the actual user

- Even if “gift” were to be taken as a context, it would have to be handled differently for each recipient

Relevant Questions

- "... it is difficult to find a relevant definition satisfying in any discipline. Is context a frame for a given object? Is it the set of elements that have any influence on the object? Is it possible to define context a priori or just state the effects a posteriori? Is it something static or dynamic? Some approaches emerge now in Artificial Intelligence. In Psychology, we generally study a person doing a task in a given situation. Which context is relevant for our study? The context of the person? The context of the task? The context of the interaction? The context of the situation? When does a context begin and where does it stop? What are the real relationships between context and cognition?"

- Bazire and Brezillon, 2005

Context in Recommendation

- **Recommendation Scenario - Revisited**
  - Steve's purchases on Amazon:
    - mystery-detective fiction "Da Vinci Code" (for himself)
    - "Python Programming" (for work)
    - "Green Eggs and Ham" (gift for his daughter)
  - context seems to be tied to a particular interaction of user with the system
  - To make things worse, context may change during one visit or interaction (context = tasks?)
  - System needs to distinguish between different longer-term interests

Outline

- **General views of context and their relevance to recommendation problem**
  - Representational versus Interactional Views
- **Architecture for Contextual User modeling**
  - Our emphasis: an interactional framework based on a model of human memory
- **Different Implementations of the Interactional model**
  - Contextual modeling based on semantic + behavioral cues
  - Contextual modeling based on latent variable models of user preferences
    - Context as Task
    - Demonstrating an approach to inferring contexts from behavioral data, using context in learning preference models, and predicting/tracking user contexts

Defining Context

- **Entities interact with their environment through "situated actions"**
  - "Any information that can be used to characterise the situation of entities." (Day 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)

- **Intensionality versus Extensionality**

Different Views of Context

- **Dourish (2004) distinguishes between two views of context:**
  - representational view and the interactional view
- **Representational view, makes four key assumptions:**
  - Context is a form of information, it is delineable, stable and separable from the activity
- **Implications:**
  - Context is information that can be described using a set of “appropriate” attributes that can be observed
  - These attributes do not change and are clearly distinguishable from features describing the underlying activity undertaken by the user within the context
  - No “situated action”

Different Views of Context

- **Interactional View of Context (Dourish, 2004)**
  - Contextuality is a relational property, i.e. some information may or may not be relevant to some activity
  - 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.
- **Drawback**
  - The "qualification problem" – similar to the outstanding problem from AI.

Interactional View: Assumptions & Implications

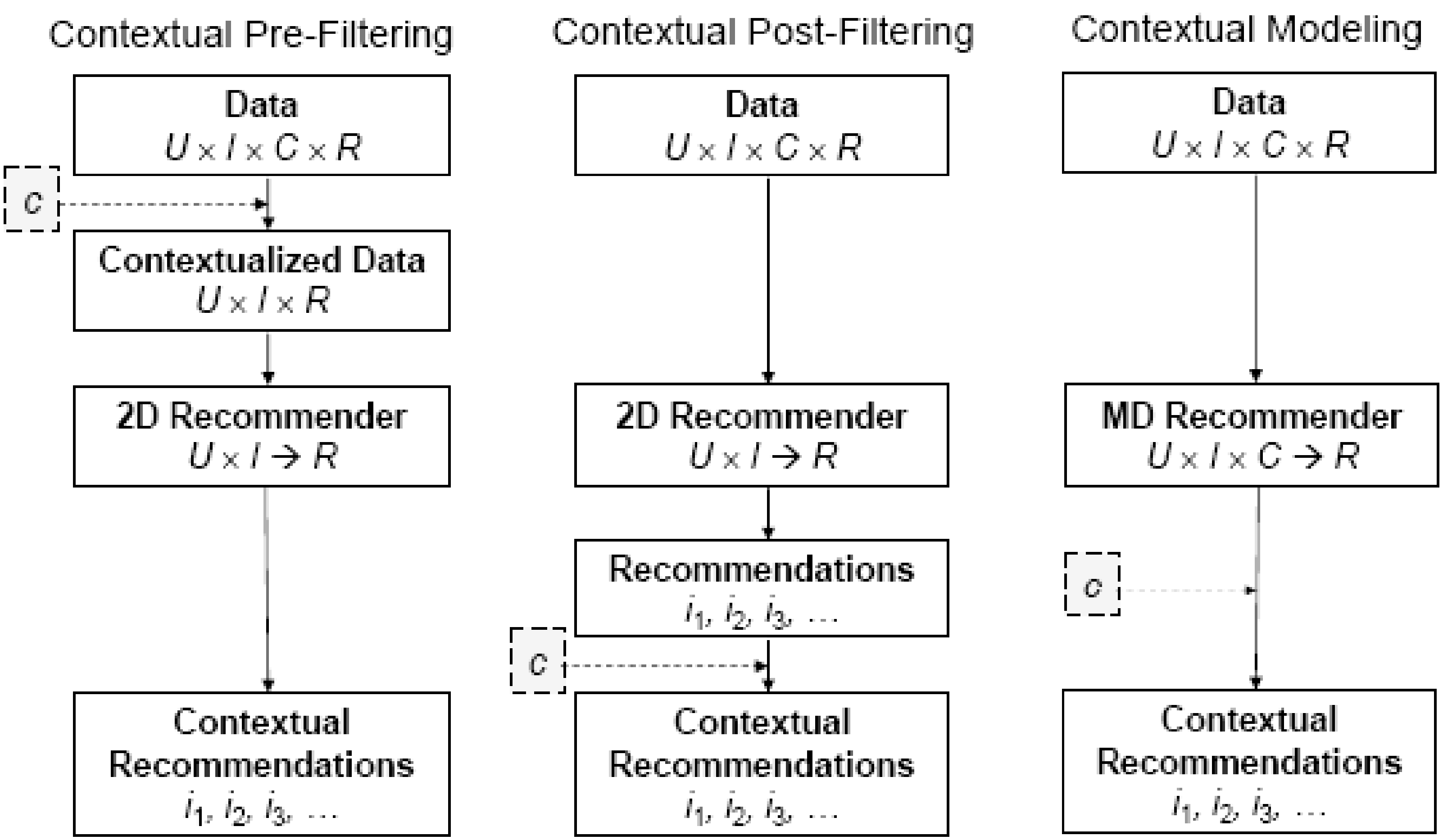
- **Properties of Context**
  - Context gives rise to a behavior that is observable, though context itself may not be observable (it's "intensional")
    - Context exists (usually implicitly) in relation to the ongoing interaction of the user with the system
  - not static
    - Can be derived: a stochastic process with d states $\{c_1, c_2, \ldots, c_d\}$
- **Context aware recommendation**
  - Integrate context in the process of learning user preference models ("Contextual User Modeling")
  - 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

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**
- **Originally introduced based on the representational view**
  - Though these architectures are also generally applicable in the interactional view

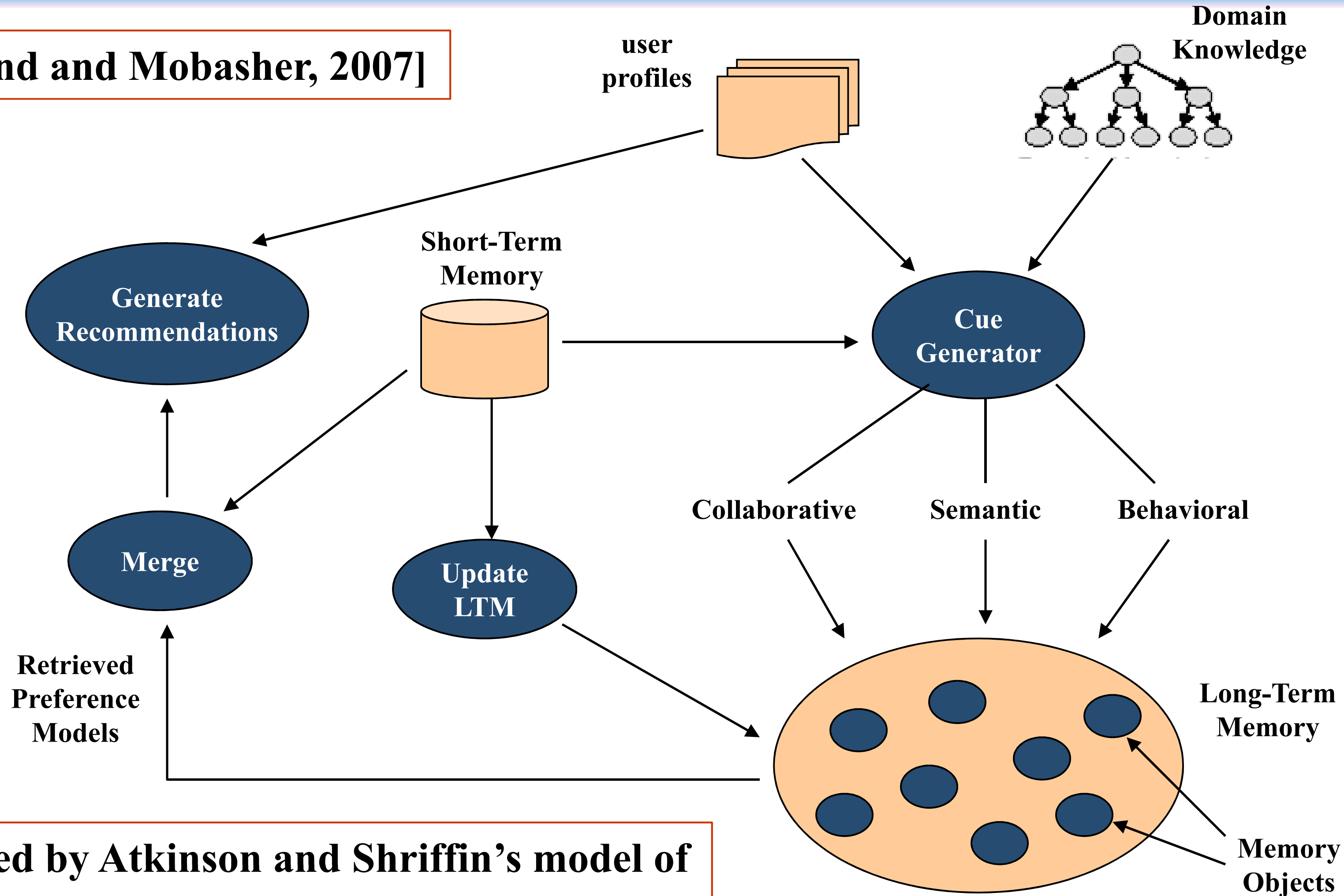
CARS Architectural Models

From Adomavicius, Tuzhilin, 2008



An Interactional Model 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:**
  - LTM objects can be organized based on ontological or semantic relationships
  - LTM preference models may be aggregate objects based on similarities among users
  - Identifying relevant LTM objects can be done in a variety ways (typically using appropriate similarity functions or probabilistic approaches)

Retrieving Preference Models Using Contextual Cues

- **Task of retrieving memory objects from LTM can be viewed as estimating:**

$$\Pr(L_i \mid STM) = \frac{1}{\Pr(STM)} \sum_j \Pr(L_i \mid CC_j) \bullet \Pr(STM \mid CC_j) \bullet \Pr(CC_j)$$

Where L_i are memory objects stored in LTM and CC_j are contextual cues generated from STM

- **The calculation $\Pr(STM|CC_j)$ would be highly dependent on the particular type of cue being used.**
  - $\Pr(STM|CC_j)$ may be estimated based on collaborative, semantic, or behavioral observations
  - E.g., $\Pr(STM|CC_j)$ *could be* a weight associated with a concept (such as the fraction of positive ratings in STM associated with items in a given category

Type of Contextual Cues

- **Collaborative Cues**
  - represent items as vectors over user ratings
  - Memory objects from LTM with preference models that have a similarity greater than a particular threshold are retrieved and used in the recommendation generation
- **Semantic Cues**
  - Retrieve LTM preference models based on semantic similarity with user preference model from the active interaction.
  - Assume the existence of an item knowledge base (or textual feature space for documents) and use item semantics to compute similarity between items.
- **Behavioral Cues**
  - Can use various implicit metrics for user preferences.
  - Similarity between these metrics computed for the active interaction and LTM preference models are used as the basis for retrieving objects from LTM.
  - Another approach is to extract latent factors that drive user choice, for example, impact values associated with item attributes extracted from an ontology, or factors derived from user actions representing various tasks or topics.

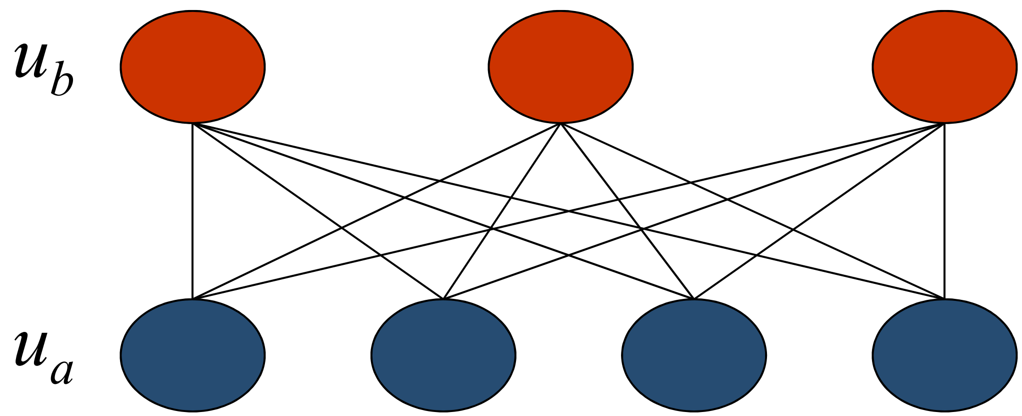
Characteristics of the Framework

- **Different, but not in contradiction to the three architectural models for contextual filtering**
- **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
    - This, in turn, 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)

Example Implementation 1: Contextual Collaborative Models

- **Inclusive Memory Model**
  - Uses all ratings in the LTM and STM of u_a to define neighbourhood
- **Temporal Memory Model**
  - Uses ratings from STM and the last k ratings from LTM
- **Contextual Memory Model**
  - Uses ratings from STM and those ratings from LTM rated within the same context as current context
- **Problem: filtering user profiles can exacerbate the usual problems with sparsity in CF**

Dealing with Sparsity: Generalized Cosine Max Similarity Metric

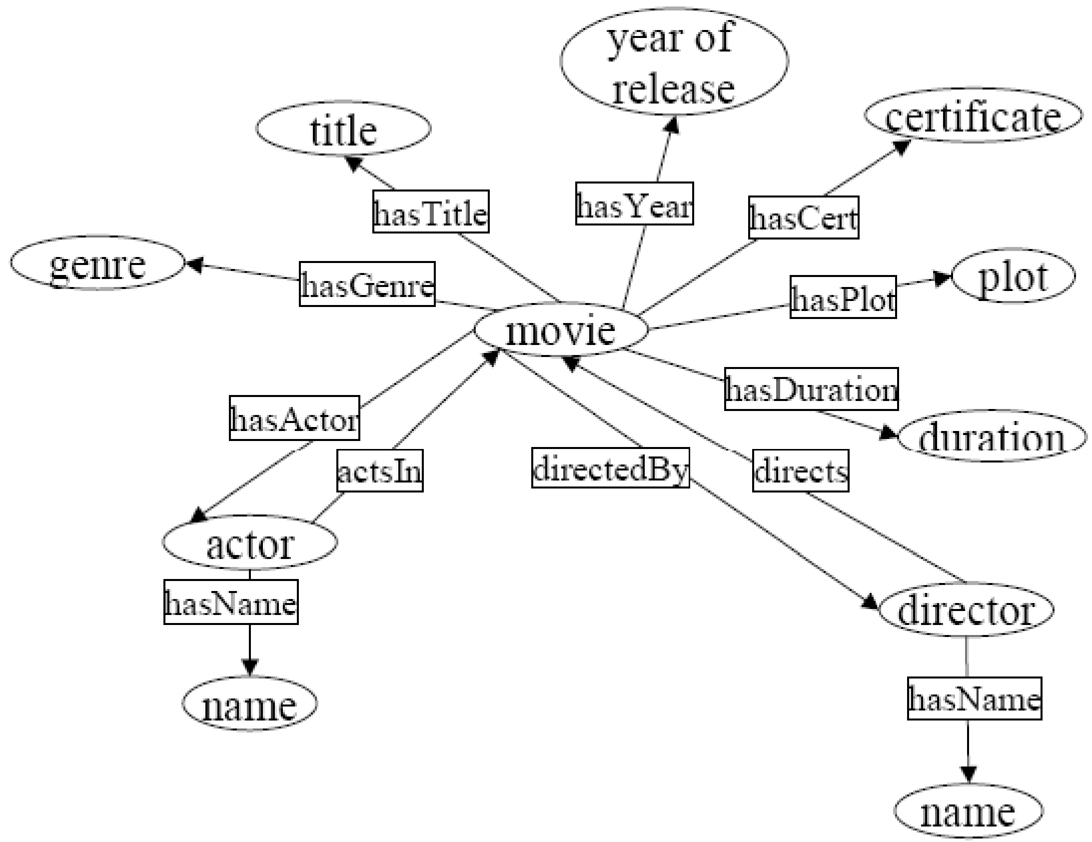


$$u_a \cdot u_b = \sum_{(i_j, i_t) \in S} r_a(i_j) \times r_b(i_t) \times s_{item}(i_j, i_t)$$

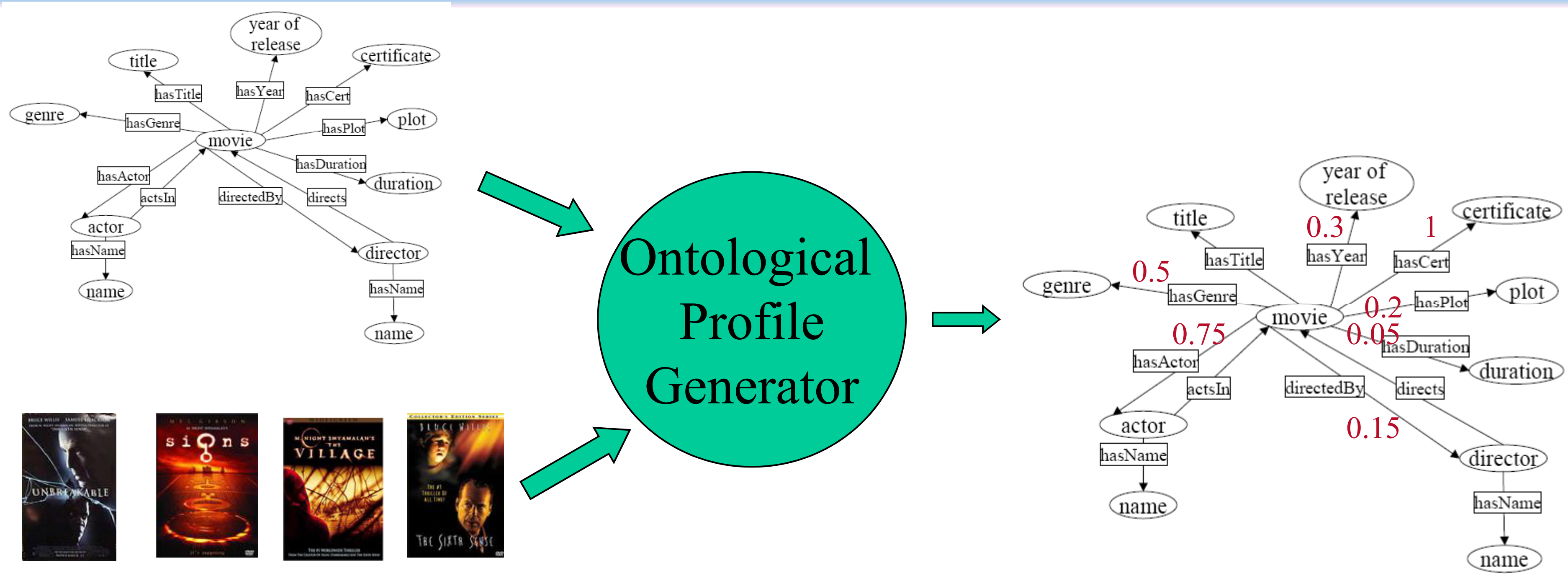
Similarity of i_j and i_t depends on the context of the user's interactions

Example Implementation 2: Item Knowledge Bases and Context

- **Given user behavioural data and an item knowledge bases**
- **Discover different user behaviors that can be associated with different user interaction contexts**



A High Level View



- **One visitor may have multiple such profiles**
  - If they are distinct enough, they would represent a different context for the user visit
  - Clustering of these profiles using identified 27 distinct clusters (contexts) within 15,000 user visits

Measuring Impact

- **Defined based on an observed (f(x)) and an expected distribution (g(x)) of instances of the concept**
  - The greater the divergence (Kullback-Leibler) between these distributions, the greater the impact
- **Assumes g(x) to be uniform**
  - All instances on the concept are equally likely to be viewed by the user

$$imp_u = 1 - \frac{H(f(x))}{\log s}$$

  - s is the number of unique instances of the concept and H(f(x)) is the entropy of f(x)

Measuring Impact (II)

- **impI assumes g(x) is the likelihood of instances of the concept being viewed within a random sample**
  - Simulated
    - using the item knowledge base, assuming each item has an equal probability of being selected
    - Popularity of the item across all users (takes temporal factors into account)

$$imp_I = \frac{\sum_{x \in D_I} f(x) \log \frac{f(x)}{\hat{f}(x)}}{-\log(min \hat{f}(x))}$$

Some Evaluation Results

Algorithm	Precision	Recall	F1
RandomNeighbour	80.4%	1.7%	0.033
Traditional CF	80.45%	8.22%	0.149
ContextualRecommender	83.8%	10.38%	0.184
ContextualSemanticRecommender	84.3%	10.81%	0.191

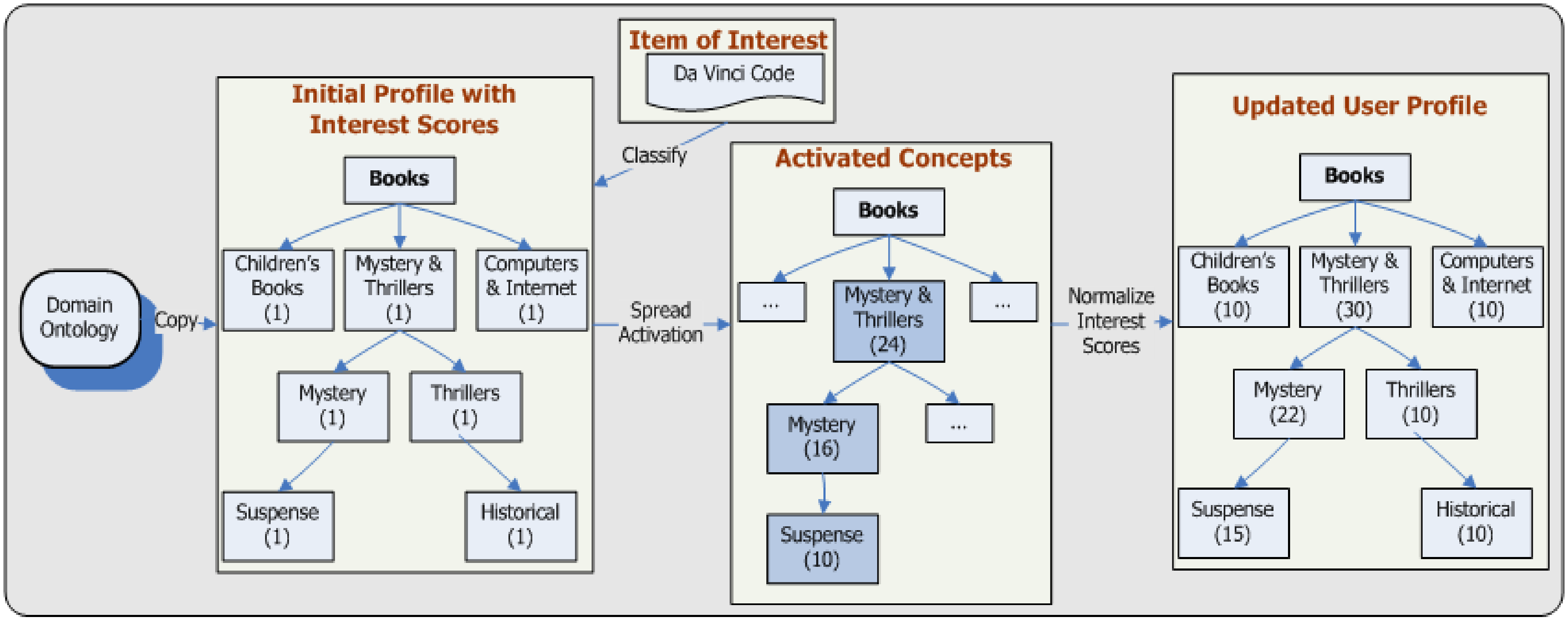
Example Implementation 3: Concepts as Context Models

- **Ontological user profile is an instance of the reference ontology**
  - E.g., Amazon's Book Taxonomy
  - Each concept is annotated with an interest score
- **Whenever the system acquires new evidence about user interests, such as page views or explicit ratings, the user profile is updated with new interest scores**
- **System is designed to maintain and update the ontological user profiles based on the user behavior and ongoing interaction**
- **Profile Normalization**
  - Relative importance of concepts in the profile reflect the changing interests and varied information contexts of the user

Updating User Context by Spreading Activation

- **Interest score**
  - Indicates the importance of a concept for the user
  - Gets incremented or decremented based on the user's behavior over many interactions
- **Spreading Activation**
  - Ontological User Profile is treated as the semantic network
  - Interest scores updated based on activation values
  - Initial set of concepts is assigned an initial activation value based on similarity to user's short-term interests
  - Activate other concepts based on a set of weighted relations
    - Relationship between adjacent concepts is determined based on the degree of overlap
  - Obtain a set of concepts and their respective activations

Profile Updating Illustrated



Augmenting Collaborative Recommendation with the Context Model

- **Collaborative Filtering with Ontological Profiles**
  - User similarities are computed based on their interest scores across ontology concepts, instead of their ratings on individual items
    - This also helps broaden the recommendations and alleviate typical problems with CF: “cold start,” “diversity,” “serendipity”
  - Additional filtering is performed by selecting only neighbors that have significant interest in the concept of the “target item”
    - This helps in identifying the relevant “information access context” and improves accuracy

Ontology-Based Collaborative Recommendation

- **Semantic Neighborhood Generation**
  - Compare the ontological user profiles for each user to form semantic neighborhoods
  - Euclidean Distance

$$\mathrm{distance}_{u,v} = \sqrt{\sum_{j \in C} \left(IS(C_{j,u}) - IS(C_{j,v})\right)^2}$$

- C - set of all concepts in the reference ontology
- $IS(C_{j,u})$ – interest score for concept C_j for target user u
- $IS(C_{j,v})$ – interest score for concept C_j for target user v

  - Normalize the distance
  - Calculate similarity based on the inverse of the normalized distance

Ontology-Based Collaborative Recommendation

- **Prediction Computation**
  - Compute the prediction for an item *i* for target user *u*
    - Select most similar *k* neighbors
    - Concept-based filtering on the neighbors
  - Variation of Resnick's standard prediction formula

$$\boxed{p_{u,i} = \bar{r}_u + \frac{\sum_{v \in V} sim_{u,v} * (r_{v,i} - \bar{r}_v)}{\sum_{v \in V} sim_{u,v}}}$$

In fact a function of user, item and concept

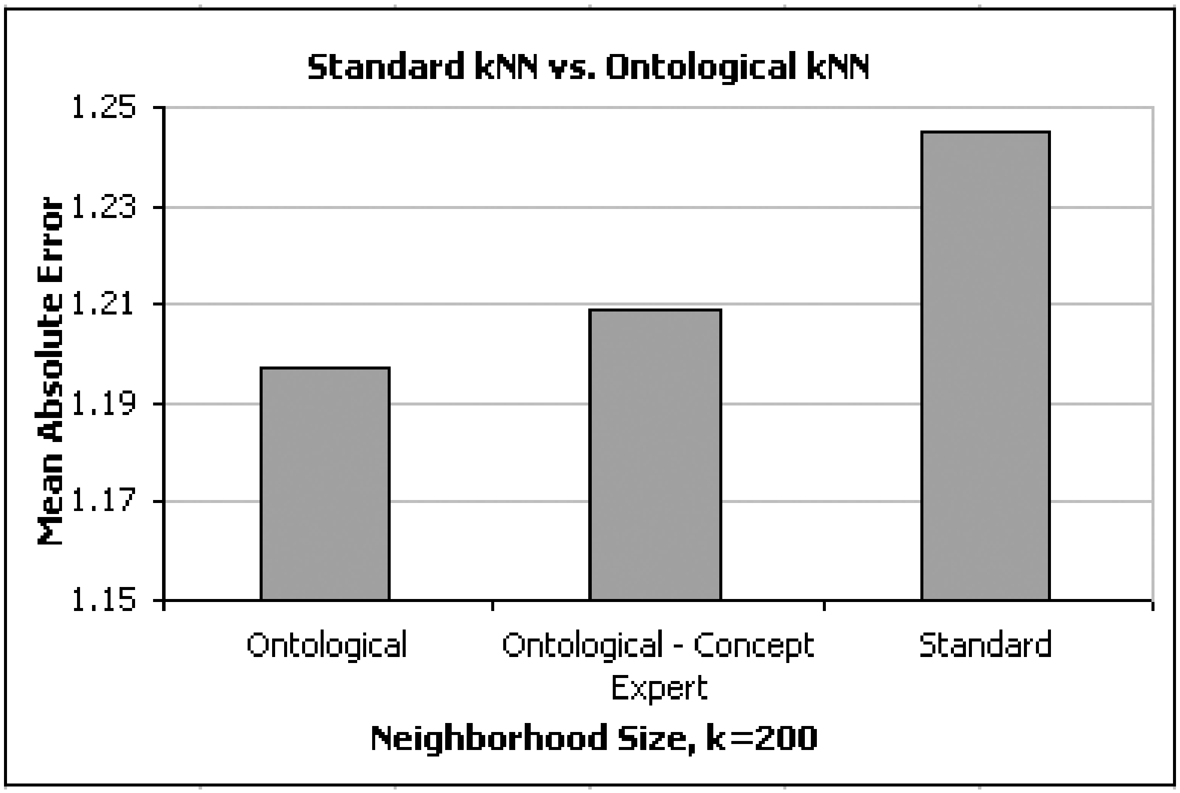
- We use concept-based mean ratings for the target user and specific neighbors
- *V* – set of *k* similar users

Experimental Setting

- **Reference Ontology**
  - Amazon's Book Taxonomy
    - ISBN – unique identifier for each book
    - Category, title, URL, and editorial reviews
    - 4,093 concepts and 75,646 distinct books
- **Evaluation using the book ratings collected by Ziegler**
  - 4-week crawl from the BookCrossing community
  - 72,582 book ratings belonging to users with 20 or more ratings
  - Training data utilized for spreading activation
  - Test data used for predicting ratings
  - K-fold cross validation, $k = 5$

Experimental Results

- **Mean Absolute Error, k=200**
  - ANOVA significance test with 99% confidence interval, *p-Value < 0.01 (6.9E-11)*



Experimental Results

- **Recommendation Diversity**
  - Improved Personalization
  - Improved Surprisal

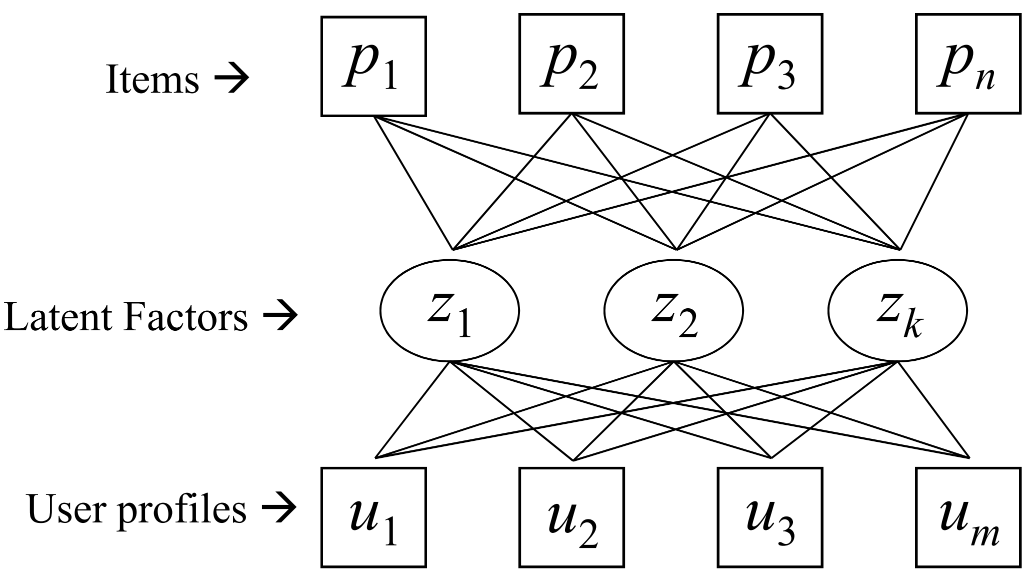
Algorithm	Personalization, d(20)	Surprisal/Novelty, I(20)
Standard kNN	0.922	6.544
Ontological kNN	0.975	7.286
ANOVA p-value	1.9417E-276	4.9221E-181

Inferring (and Predicting) Context: Latent Variable Models

- **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.
  - Particularly useful when dealing with applications involving user's performing informational or functional tasks.
  - Contexts correspond to tasks and are derived as latent factors in the observational data collected in the short-term memory.
  - Probabilistic Latent Semantic Analysis (PLSA) can be used to automatically learn and characterize these tasks, as well as the relationships between the tasks and items or users.
  - An algorithm based on Bayesian updating to discover individual user's task transition patterns and generating ***task-level user models***.

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.**



Advantage of PLSA:

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

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

PLSA Model – Behavioral Observations

- **represented as a matrix (UP_{mxn})**
- **each entry UP_{ij} corresponds to a weight of item j within a user interaction i. The weight can be binary, or based on the various implicit or explicit measures of interest.**

	p1	p2	p3	p4	p5	. . .
User 1	1	0	0	1	1	. . .
User 2	0	1	1	0	1	. . .
…	…	…	…	…	…	…

Note: similar models can be built using other types of observation data, e.g., <users, query terms>, <pages, keywords>, etc.

PLSA Model

- **Consider each single observation (u_i, p_j) as a generative process:**
  - 1. select a user session u_i with $\Pr(u_i)$
  - 2. select a latent factor z_k associated with u_i with $\Pr(z_k \mid u_i)$
  - 3. given the factor z_k, pick an item p_j with $\Pr(p_j|z_k)$
  - each observation is represented as

$$\Pr(u_i, p_j) = \sum_{z_k} \Pr(u_i) \bullet \Pr(z_k \mid u_i) \bullet \Pr(p_j \mid z_k)$$

  - using Bayes' rule, we can also obtain:

$$\Pr(u_i, p_j) = \sum_{z_k} \Pr(z_k) \bullet \Pr(u_i \mid z_k) \bullet \Pr(p_j \mid z_k)$$

Model Fitting

- **Maximizing the likelihood of usage observation**

$$L(U,P)=\sum_{u_i,p_j} UP_{ij}\bullet\log\Pr(u_i,p_j)=\sum_{u_i,p_j} UP_{ij}\bullet\log\sum_{z_k}\Pr(z_k)\Pr(u_i\mid z_k)\Pr(p_j\mid z_k)$$

- **Estimating parameters using Expectation-Maximization (EM) algorithm**

- **Output of the EM algorithm:**
  - **$\Pr(u_i \mid z_k)$, $\Pr(p_j \mid z_k)$, $\Pr(z_k)$**

- **Using Bayes' rule, we can also obtain:**
  - **$\Pr(z_k \mid u_i)$, $\Pr(z_k \mid p_j)$**

Context-Based Patterns

- **Identify user segments**
  - For each context z_k, find top users with the highest $\Pr(u|z_k)$ as a user segment.
  - Applications: collaborative recommendation; market segmentation
- **Identify characteristic items or users w.r.t. each task**
  - Characteristic items: $\{p_{ch} : \Pr(p_{ch}|z_k)*\Pr(z_k|p_{ch}) \geq \alpha\}$
  - Characteristic user sessions: $\{u_{ch} : \Pr(u_{ch}|z_k)*\Pr(z_k|u_{ch}) \geq \theta\}$
  - Applications: task/context based search; user or item classification
- **Identify a given user's context**
  - For each user (interaction) u, find tasks with the highest $\Pr(z|u)$
  - Allows for performing higher-level behavioral analysis (based on discovered tasks or contexts)

Methodological Note

- **Two Web navigational data sets**
  - CTI: about 21,000 sessions, 700 pages, 30 tasks (defined), user history length set to 4+, 20,000+ features
  - Realty Data: about 5,000 sessions, 300 properties, user history length set to 4+, 8000+ features
- **Experimental Methodology:**
  - Measure the accuracy of our recommendation system, compare it to a standard recommendation system based on first-order Markov model
  - Use "hit ratio" to measure recommendation accuracy
  - Hit ratio:
    - Given a test session, use the first k items to generate a top-N recommendation set.
    - If this set contains the $k+1^{th}$ item of the test session, we consider it a hit.
    - HitRatio = totalHits / totalSessions

      (averaged over 10 runs in 10-fold cross-validation)

Examples of Inferred Tasks

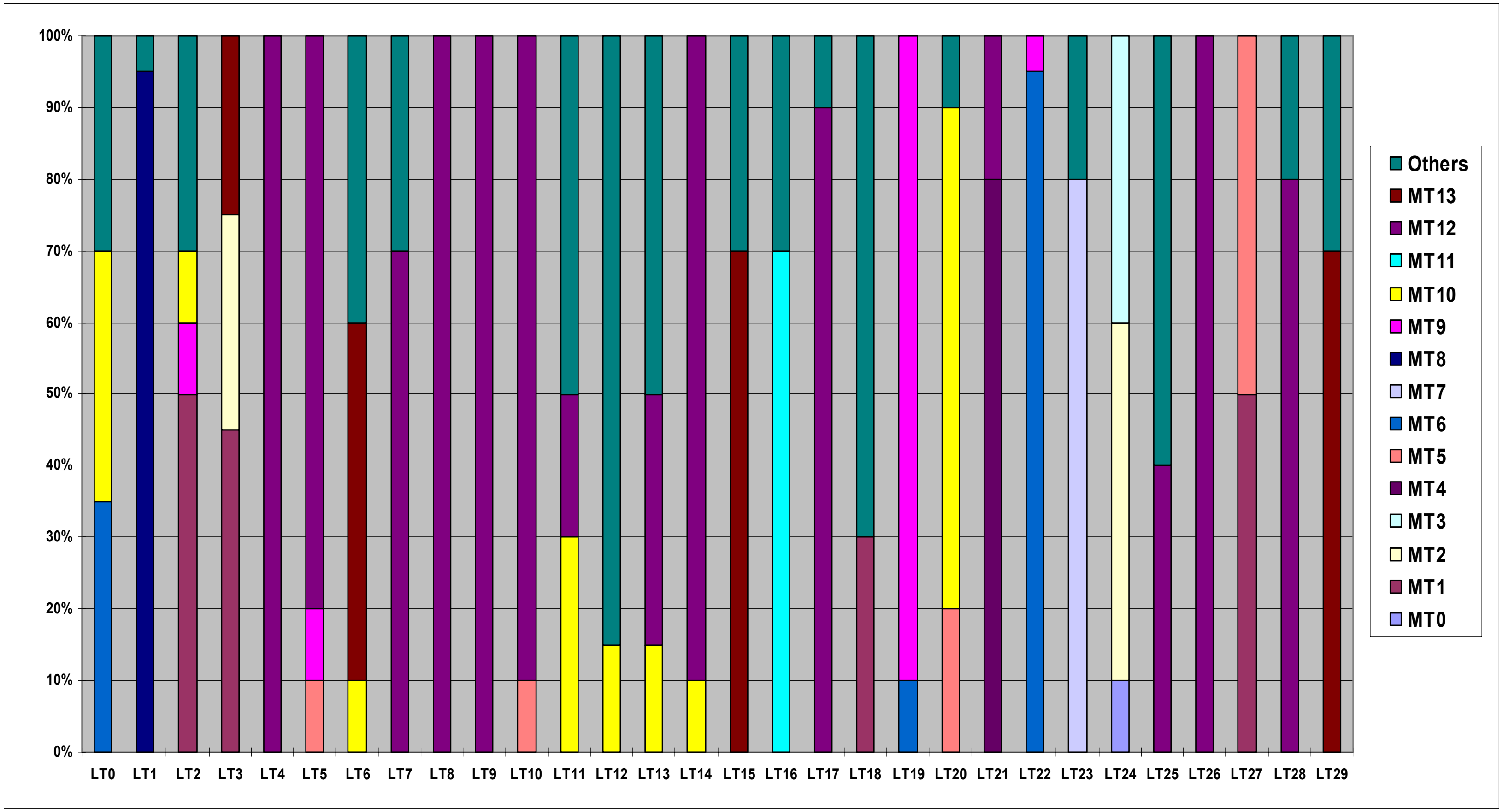
	A real user session (page listed in the order of being visited)
1	Admission main page
2	Welcome information – Chinese version
3	Admission info for international students
4	Admission - requirements
5	Admission – mail request
6	Admission – orientation info
7	Admission – F1 visa and I20 info
8	Application – status check
9	Online application - start
10	Online application – step 1
11	Online application – step 2
12	Online application - finish
13	Department main page

Top tasks given this user – Pr(task | user)	
Task 10	**0.4527**
Task 21	**0.3994**
Task 3	0.0489
Task 26	0.0458

PageName
Department main page
Admission requirements
Admission main page
Admission costs
Programs
Online application – step 1
…
Admission – international students

PageName
Online application – start
Online application – step1
Online application – step2
Online application - finish
Online application - submit
…
Department main page

Distribution of Learned Tasks



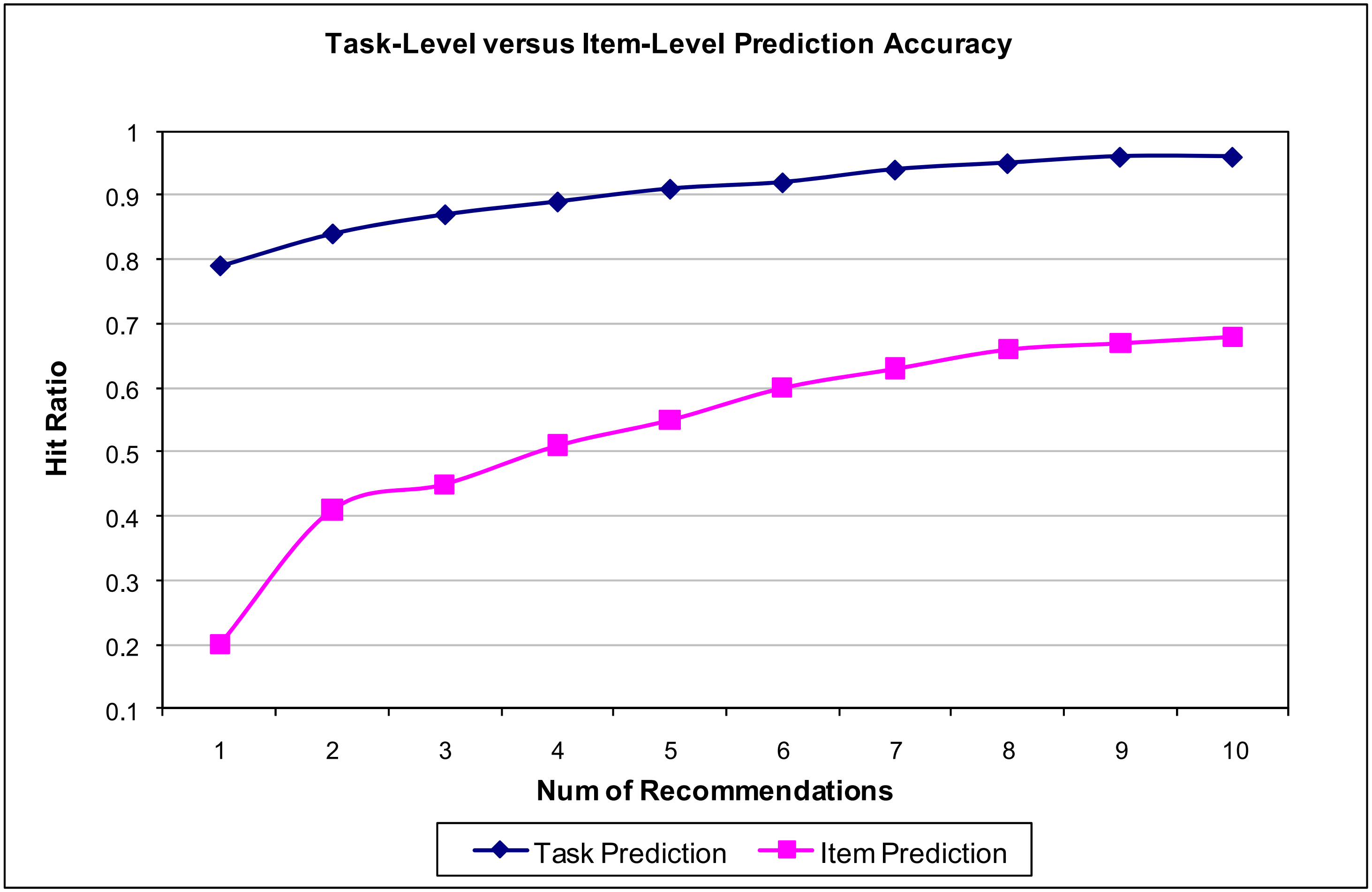
MT0 – MT13 were actual tasks, commonly performed by users on the Web site, selected manually by domain experts

Task-Level User Tracking?

A real user session involving tasks 10 (Admissions Info.) and Task 21 (Online Application)										
1	2	3	4	5	6	7	8	9	10	11
T10: 0.6	T21: 0.4									
	T10: 0.7	T21: 0.3								
		T10: 0.8	T21 :0.2							
			T10: 0.8	T21 :0.2						
				T10: 0.6	T21: 0.4					
					T10: 0.3	T21: 0.7				
						T10: 0.1	T21: 0.9			
							T10: 0	T21: 1		
								T10: 0.2	T21: 0.8	

Sliding window, W (with $|W| = 4$) moves from the beginning to the end of this user session. Top 2 tasks and the corresponding values for Pr(task | W) are shown.

Task/Context Prediction



Contextual Modeling: The Maximum Entropy Framework

- **A statistical model widely used in language learning, text mining (Rosenfeld 1994, Berger et al. 1996).**
  - Estimate probability distribution from the data
  - Labeled training data used to derive a set of constraints for the model that characterize class-specific expectations in the distribution
  - Constraints represented as expected values of “features” which are real-valued functions of examples
  - Goal: find a probability distribution which satisfies all the constraints imposed on the data while maintaining maximum entropy

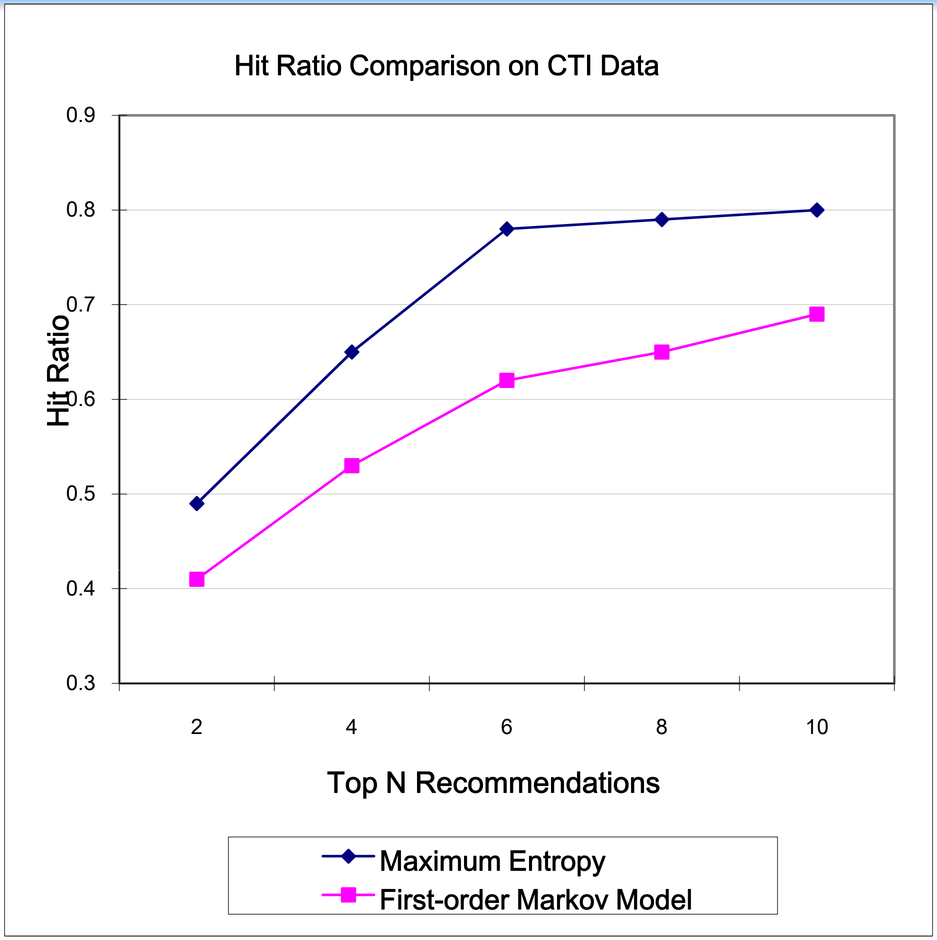
- **Advantage:**
  - integration of multiple sources of knowledge or multiple constraints without subjective assumptions or intervention.
  - In this case we use ME to integrate learned contextual information into the preference modeling process.

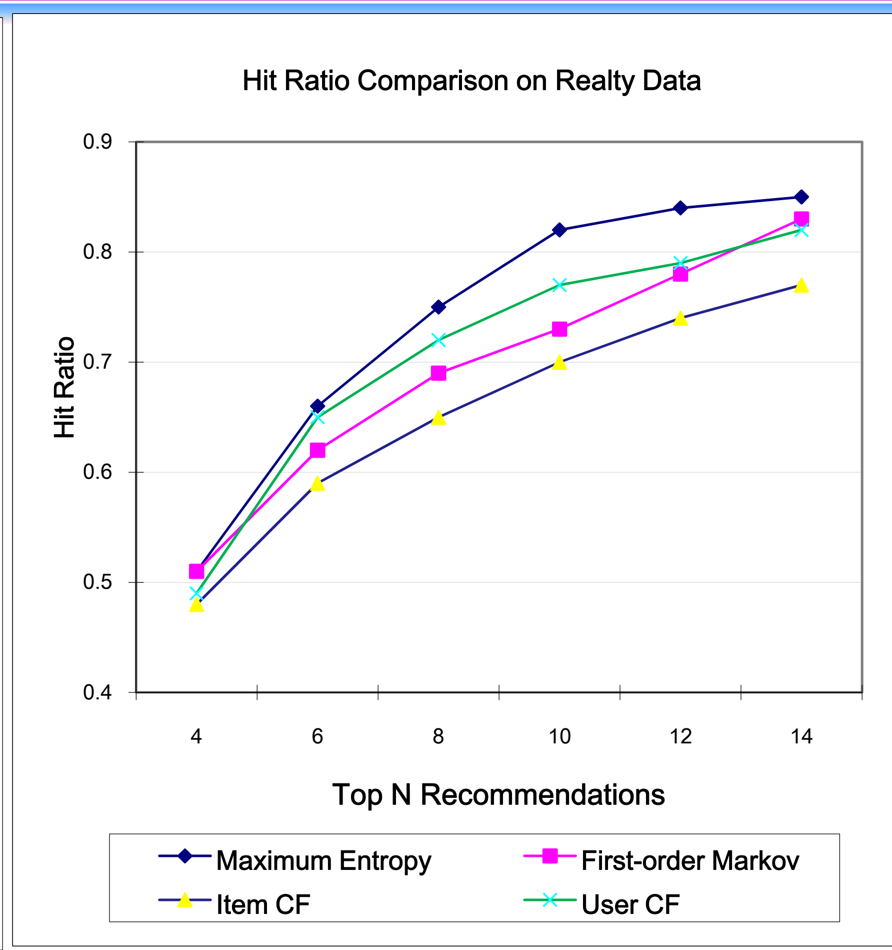
Using the Maximum Entropy Model

- **The basic Maximum Entropy framework:**
  - First, Identify a set of feature functions that will be useful for the desired task (e.g., prediction or classification)
  - Then, for each feature:
    - Measure its expected value over the training data
    - Take this expected value to be a constraint for the model distribution

- **In our model**
  - Define two sets of features
    - 1. features based on item-level transitions in users interactions
    - 2. features based on task-level transitions in user interactions
  - Max. Ent. Framework uses both sets of features to compute $\Pr(p_d \mid H(u))$

Experimental Results





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) & effective (but ad hoc) ways of dealing with the qualification problem
  - **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 and user activity can be particularly effective in contextual user modeling

Still many unanswered questions ….

?

But, all will be revealed in the panel discussion ☺