Incorporating Concept Hierarchies Into Usage Mining Based Recommendations

Amit Bose - University of Minnesota
Kalyan Beemanapalli – University of Minnesota
Jaideep Srivastava - University of Minnesota
Sigal Sahar - Intel Corporation

Presenter: **Kalyan Beemanapalli**

Outline

- Motivation and Background
- Domain Knowledge and Concept Hierarchy
- Similarity Model
- Recommendation Engine
- Experimental Setup
- Results
- Conclusion and Future Directions

Motivation

- Most Recommendation Engines are based on Usage Information
- Very few have explored the use of Domain Information in usage analysis (*Jia et al*)
- No generalized framework for incorporating domain information into Usage Analysis
- Other areas like Bioinformatics and Information Retrieval have made use of domain information successfully
- Recent studies have shown that structural and conceptual characteristics of a website play an important role in the quality of the recommendations provided by a recommendation engine (*Nakagawa et al*)
- Domain information helps in incorporating expert knowledge into usage analysis

Basic Approach

- Many user sessions are similar – locate these
- Form clusters of similar sessions - Define a similarity measure between sessions using all available data
- Represent each cluster using a click-stream tree (*Gündüz et al*)
- When generating recommendations, match the current user's session with the best cluster and recommend page(s) which are not part of the current user's session
- Make domain information (Concept Hierarchy) an integral part of this architecture.

Background

- Sequence Alignment
  - Example:

    Q1 = (P1, P2, P3, P4, P5)

    Q2 = (P2, P4, P5, P6)
  - Optimal alignment of the sequences

    __ **P2** __ **P4** **P5** **P6**

    P1 **P2** **P3** **P4** **P5** __
- Scoring Matrix
  - Example: 2 for a match, -1 for a mismatch, Alignment score = 2
  - Alignment can be very useful if scoring matrix is designed carefully

	P1	P2	P3	P4	P5	-
P2	1	2	1	1	-2	-4
P4	-1	0	1	2	-1	-3
P5	-4	-3	-2	-1	0	-2
P6	-5	-4	-3	-1	-1	-1
-	-5	-4	-3	-2	-1	0

Figure 1. Calculation of similarity score between sequences using dynamic programming

Scoring Matrix using Domain Knowledge

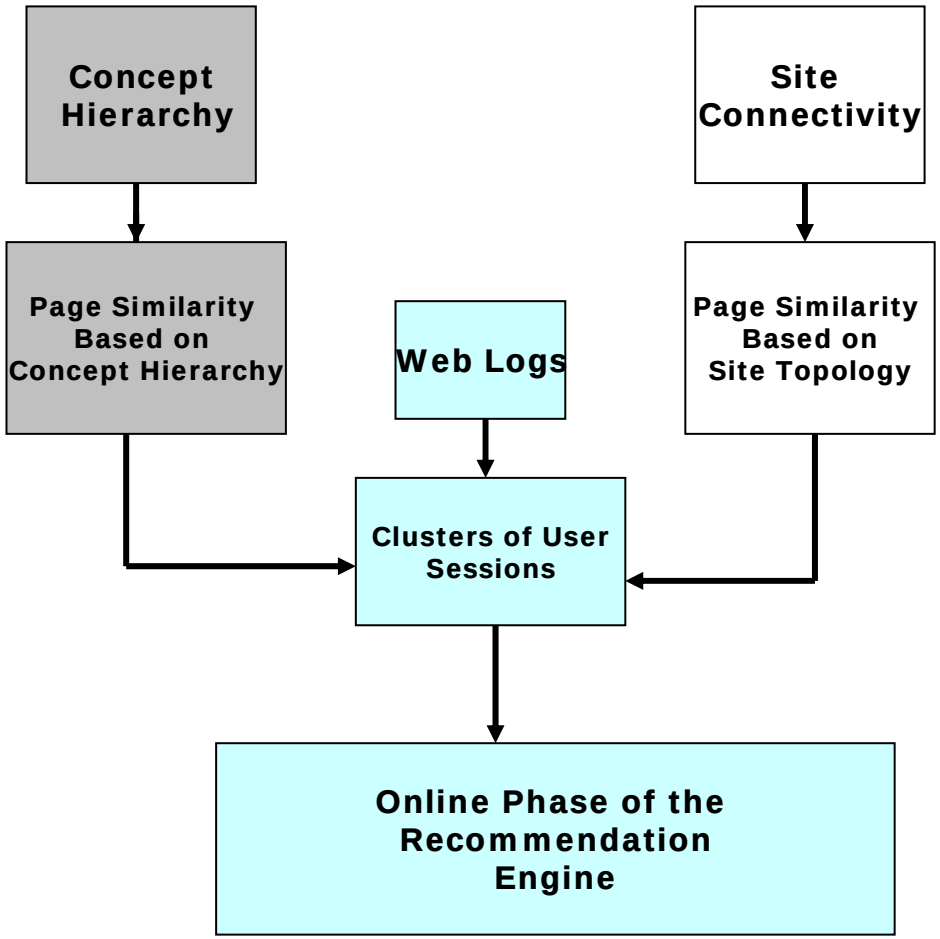
- Protein Sequence Alignment is the optimal alignment of two protein sequences
- A protein is a sequence of amino acids
- One can think of a protein as a sequence of characters – sequence alignment equivalent to optimal string match
- The problem of pair-wise sequence alignment is well studied; there exist solutions based on dynamic programming
- Use BLOSUM62(*Henikoff and Henikoff*) to determine the similarity between amino acids

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

BLOSUM62 Matrix

How does this help us?

- A user session is a sequence of web pages.
- Any two user sessions can be optimally aligned to get alignment score – higher means more similar
- Challenge is to design an appropriate scoring (or similarity) matrix for the web domain
- Several ways possible to generate page-by-page similarity matrix:
  - Using Concept hierarchy of the web-site
  - Using Link structure of the web-site



Model for using Domain Knowledge

Quantifying Similarity

- Important ingredient in sequence alignment
- Two kinds of Similarity measures:
    1. Similarity between pages
    2. Similarity between sessions
- Defining similarity: two issues
    - What is the basis of similarity
    - How to calculate strength of this similarity
- Meaning of session alignment – find the best matching of user intents
- We use Domain knowledge to define similarity between pages and use this similarity to quantify similarity between sessions

Concept Hierarchy

- Web-site content organized and structured to reflect functional characteristics
- Hierarchy of abstractions – a common way of organizing content
- Different parts of the tree address different purposes; concepts more generally
- Concept hierarchy – content designer's view of the user intent
- Yahoo! Directory, Google Directory, and the hierarchy that can be obtained from Content Management Servers

Sample Concept Hierarchy

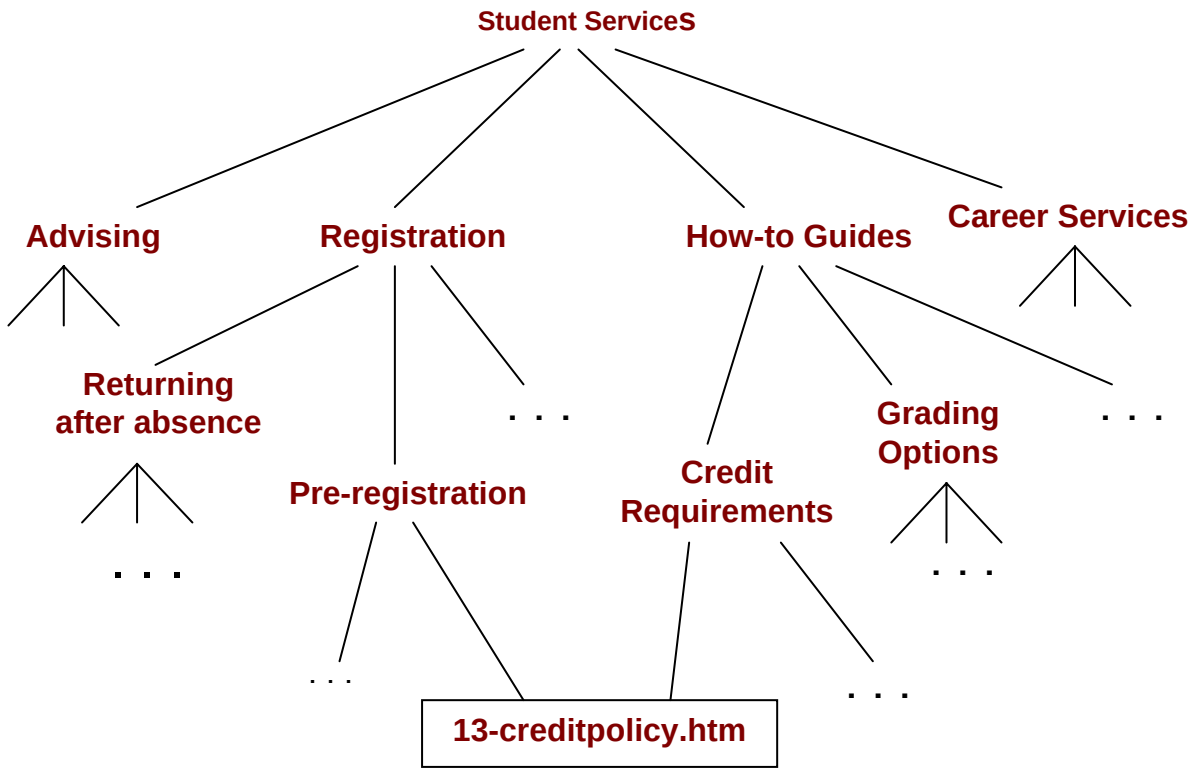


Figure 2. Example concept hierarchy for a university student-services website

Adapting Concept Hierarchy

- Simple edge-counting: assumes links span same distance
- Information theoretic model (Resnik, 1999)
- Associate probabilities with nodes
- Probability gives strength of concept; is monotone
- *Information content* of a node is defined as the negative logarithm of probability

$$I(n) = -\log p(n)$$

where p(n) is the probability assigned to node n

- Higher level nodes are less informative, root = 0

New Similarity Model – Based on Concept Hierarchy

- Probabilities calculated using usage information
- Increment frequency of page and its ancestors
- To gauge similarity between pages, find all subsuming ancestors
- Similarity = Maximum information content of all subsuming ancestors

$$S(n_1, n_2) = \max_{a_i \in A} \{ I(a_i) \}$$

where **A** – **Common Ancestor of pages belonging to concepts n1 and n2**

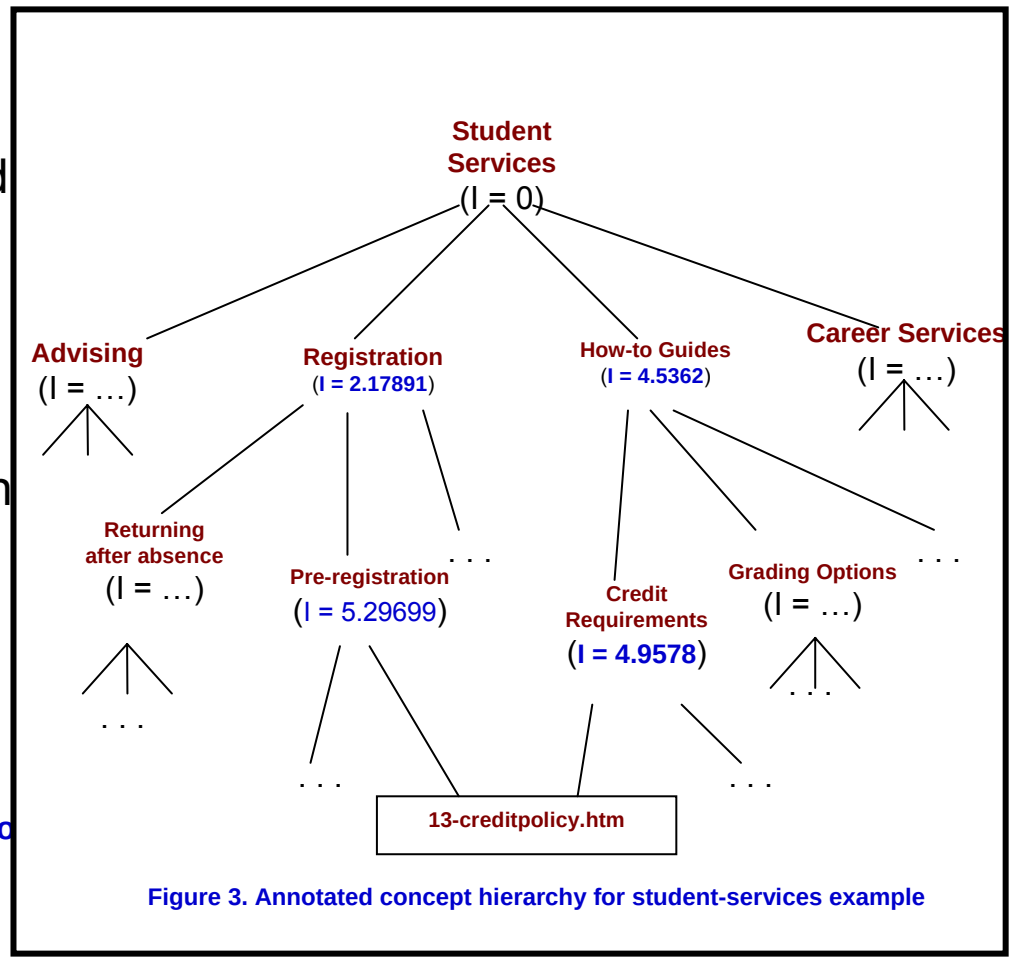


Figure 3. Annotated concept hierarchy for student-services example

Normalization of Similarity Values

- Information Content, being a logarithm, lies in the range of 0 to ∞
- The range needs to be normalized to use for calculating alignment scores of sessions
- The values are normalized between -1(maximum penalty) to 1 (maximum reward)
- Thus the normalized similarity score between page nodes n_1 and n_2 is given as

$$Sim(n_1, n_2) = \begin{cases} \dfrac{S(n_1, n_2)}{I_M} - 1 & if\ S(n_1, n_2) \le I_M \\ \dfrac{S(n_1, n_2) - I_M}{I_{MAX} - I_M} & otherwise \end{cases}$$

Where I_M and I_{MAX} are the median and maximum values of the information contents of all concept nodes in the hierarchy

Recommendation Engine Architecture

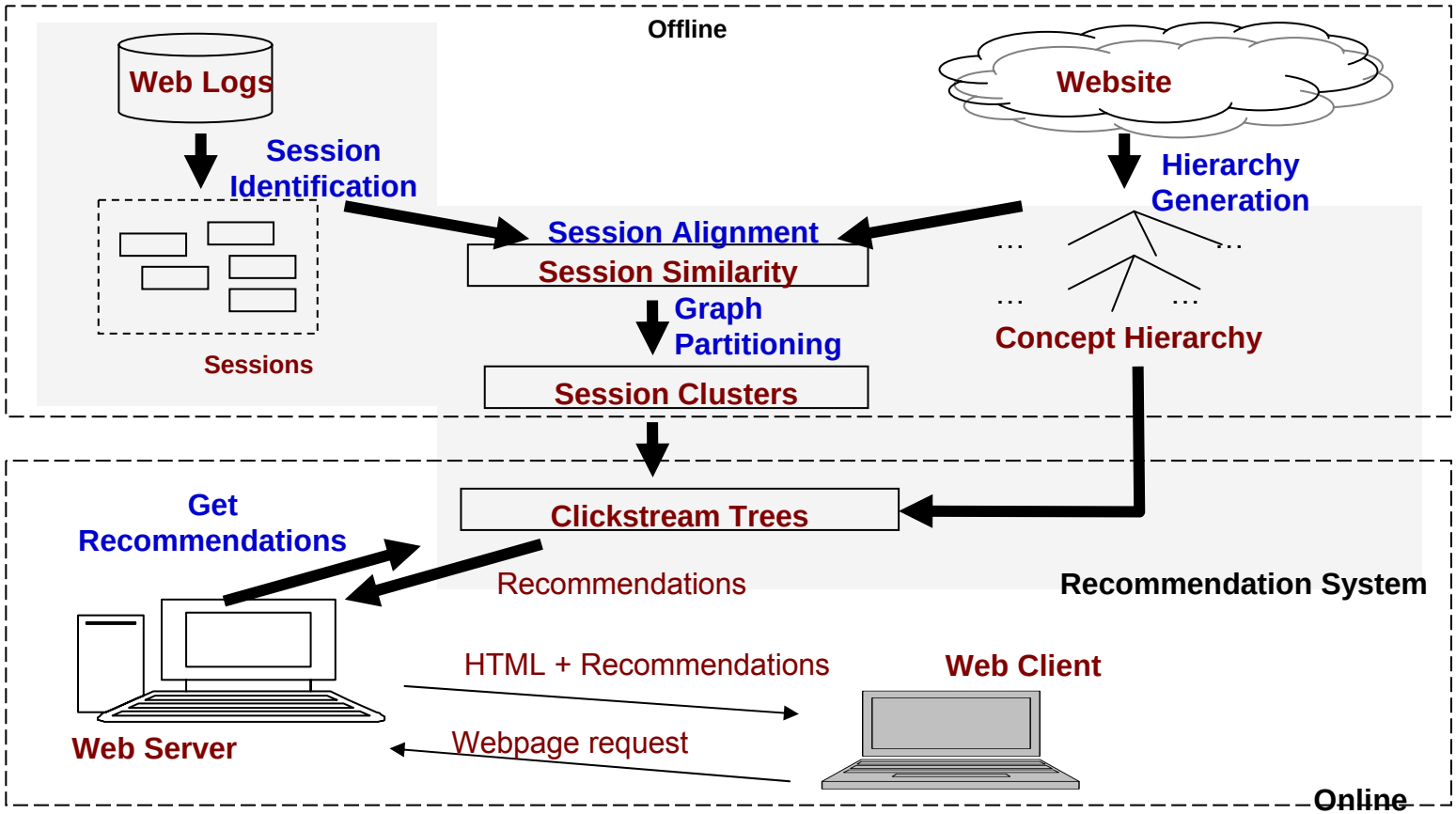


Figure 1. The Recommender System

Recommendation Engine – Online Phase

- This is the online phase of the recommendation Engine architecture
- The current user session is matched against the sessions in the clusters which are ending with the same page as the online session
- Calculate the pairwise similarity score between each of the these matching sessions with the online session. Define the recommendation score
- Recommend the top n pages
- The calculation of recommendation score can be as simple as the similarity score itself or something complex
- A Sample click stream tree is shown in the figure

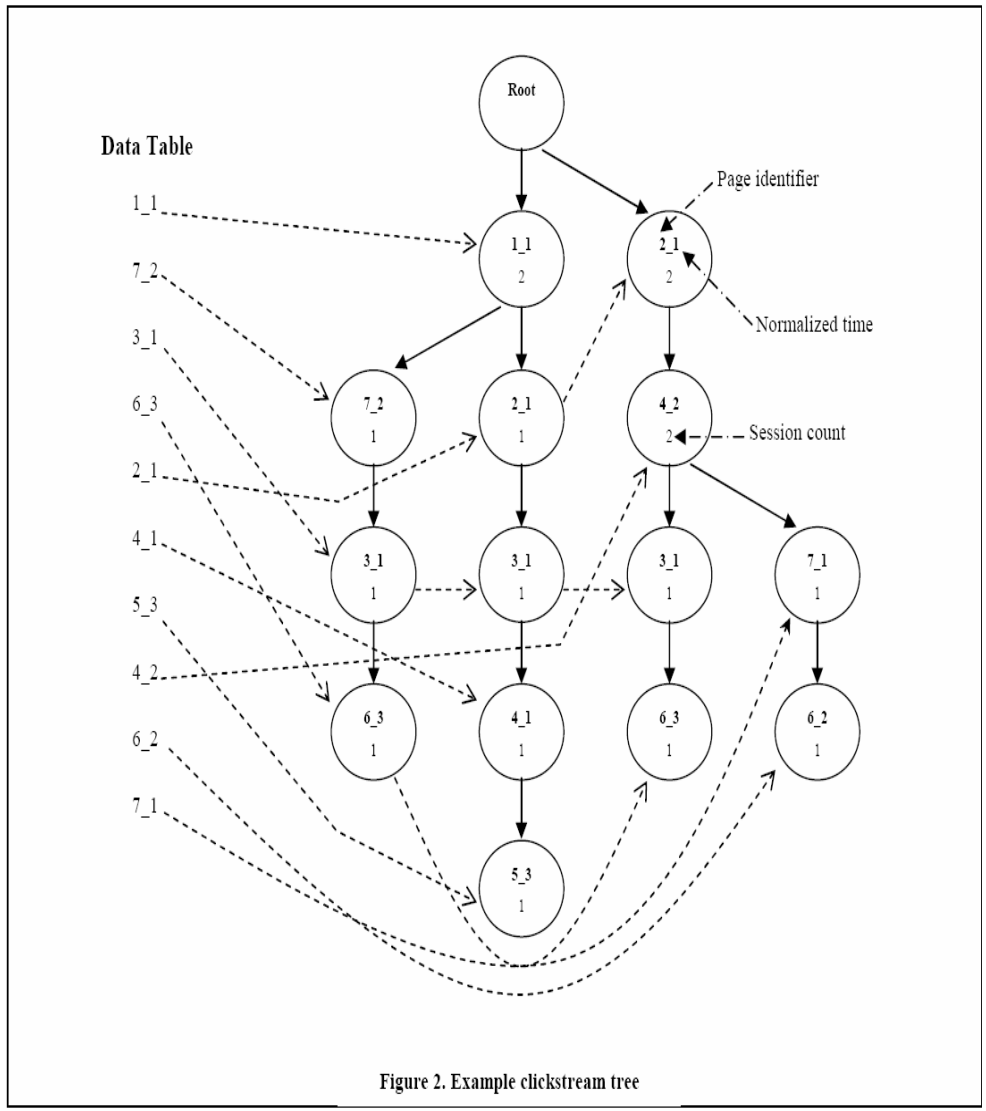


Figure 2. Example clickstream tree

Experimental Setup

- Experiments carried out on web-server logs obtained from CLA website
- The website serves over 14,500 students in nearly 70 majors and minors
- Contains about 1500 unique web pages
- After removing the noise sessions, obtained about 50,000 sessions
- Used a portion of the cleaned logs as training sessions and remaining as test sessions
- The performance was measured using various metrics.

Metrics

- *Predictive Ability (PA)*: **Percentage of pages in the test sessions for which the model was able to make recommendations. This is a measure of how useful the model is.**
- *Prediction Strength (PS)*: **Average number of recommendations made for a page.**
- *Hit Ratio (HR)*: **Percentage of *hits*. If a recommended page is actually requested later in the session, we declare a hit. The hit ratio is thus a measure of how good the model is in making recommendations.**
- *Click Reduction (CR)*: **Average percentage click reduction. For a test session *(p1, p2,…, pi…, pj…, pn)*, if *pj* is recommended at page *pi*, and *pj* is subsequently accessed in the session, then the click reduction due to this recommendation is:** (j-i)/i
- *Average Recommendation Rank (AR)*: **Average rank of a hit.. If a recommendation is a hit, then the rank of the recommendation is the rank of that hit. The lower the rank of a hit, the better the quality of recommendation.**

Results

Model	Number of Recommendations Made = 10				
	Metrics				
	PA	PS	HR	CR	AR
RSM	93.42	9.82	**45.22**	**30.68**	**6.23**
SSM	97.50	9.81	**42.17**	**27.38**	**6.28**
CSM	97.27	9.80	**54.08**	**39.89**	**6.38**

Model	Number of Recommendations Made = 5				
	Metrics				
	PA	PS	HR	CR	AR
RSM	93.42	4.96	**35.13**	**33.14**	**3.12**
SSM	97.50	4.96	**31.23**	**27.56**	**3.59**
CSM	97.27	4.96	**42.56**	**38.87**	3.41

Conclusions

- Recommendation models based on usage information are incomplete as domain knowledge is ignored.
- Using domain knowledge, which represents the expert's opinion, the efficiency of the recommendation engine can be improved
- We designed a framework for integrating domain information with usage logs
- Demonstrated the utility of leveraging domain knowledge

Future Directions

- Improve the rank of the recommendations and Prediction strength.
- Incorporate other kinds of information like link structure of the web site.
- The recommendations need to be tested with domain experts and against real subjects in a lab environment
- Use browsing subsequence alignment rather than complete browsing sequence alignment
- Use Domain information to recommend pages which are not available with the usage logs

Questions and Suggestions