

**A New Technique For
Intelligent Web Personal Recommendation**

OSSAMA HASHEM KHAMIS EMBARAK

Submitted for the Degree of Doctor of Philosophy

Heriot-Watt University

School of Mathematical and Computer Sciences (MACS)

October 2011

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

ABSTRACT

Personal recommendation systems nowadays are very important in web applications because of the available huge volume of information on the World Wide Web, and the necessity to save users' time, and provide appropriate desired information, knowledge, items, etc. The most popular recommendation systems are collaborative filtering systems, which suffer from certain problems such as cold-start, privacy, user identification, and scalability. In this thesis, we suggest a new method to solve the cold start problem taking into consideration the privacy issue. The method is shown to perform very well in comparison with alternative methods, while having better properties regarding user privacy.

The cold start problem covers the situation when recommendation systems have not sufficient information about a new user's preferences (the user cold start problem), as well as the case of newly added items to the system (the item cold start problem), in which case the system will not be able to provide recommendations. Some systems use users' demographical data as a basis for generating recommendations in such cases (e.g. the Triadic Aspect method), but this solves only the user cold start problem and enforces user's privacy. Some systems use users' 'stereotypes' to generate recommendations, but stereotypes often do not reflect the actual preferences of individual users. While some other systems use user's 'filterbots' by injecting pseudo users or bots into the system and consider these as existing ones, but this leads to poor accuracy.

We propose the *active node method*, that uses previous and recent users' browsing targets and browsing patterns to infer preferences and generate recommendations (node recommendations, in which a single suggestion is given, and batch recommendations, in which a set of possible target nodes are shown to the user at once). We compare the active node method with three alternative methods (Triadic Aspect Method, Naïve Filterbots Method, and MediaScout Stereotype Method), and we used a dataset collected from online web news to generate recommendations based on our method and based on the three alternative methods. We calculated the levels of novelty, coverage, and precision in these experiments, and we found that our method achieves higher levels of novelty in batch recommendation while achieving higher levels of coverage and precision in node recommendations comparing to these alternative methods. Further, we develop a variant of the active node method that incorporates semantic structure elements. A further experimental evaluation with real data and users showed that semantic node recommendation with the active node method achieved higher levels of novelty than non-semantic node recommendation, and semantic-batch recommendation achieved higher levels of coverage and precision than non-semantic batch recommendation.

*This Thesis is dedicated to my Family, Parents
and my Brothers*

ACADEMIC REGISTRY

Research Thesis Submission



Name:	OSSAMA HASHEM KHAMIS EMBARAK		
School/PGI:	School of Mathematical and Computer Sciences (MACS)		
Version: <i>(i.e. First, Resubmission, Final)</i>	First	Degree Sought (Award and Subject area)	Doctor of Philosophy (Computer Science)

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

- 1) The thesis embodies the results of my own work and has been composed by myself
- 2) Where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
- 3) The thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
- 4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
- 5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.

Signature of Candidate:		Date:	/ / 2011
-------------------------	--	-------	----------

Submission

Submitted By <i>(name in capitals)</i> :	
Signature of Individual Submitting:	
Date Submitted:	/ / 2011

For Completion in the Student Service Centre (SSC)

Received in the SSC by <i>(name in capitals)</i> :			
<i>Method of Submission</i> <i>(Handed in to SSC; posted through internal/external mail):</i>			
<i>E-thesis Submitted (mandatory for final theses)</i>			
Signature:		Date:	/ / 2011

Publications arising from this thesis

- Embarak, O., Corne, D. "A Method for Solving the Cold Start Problem in Recommendation Systems", 7th international conference on innovations in information technology (Innovations'11) Communication. Abu Dhabi, UAE, pp. 239–244, 2011.
- Embarak, O., Corne, D. "Integration of Users Preferences and Semantic Structure to Solve the Cold Start Problem", 7th international conference on innovations in information technology (Innovations'11) Communication. Abu Dhabi, UAE, pp. 245–250, 2011.
- Embarak, O., Corne, D. "Semantic Structure for E-Commerce Applications", 4th international conference on Developments in E-Systems Engineering - DeSE2011-Track 03: e-Business and Management innovations. Dubai, UAE, 2011, Pending.
- Embarak, O., Corne, D. "Preventing the Privacy Problem via Integration of Users Preferences and Semantic Structure", 4th international conference on Developments in E-Systems engineering - DeSE2011-Special Session: Advanced Interaction Technology. Dubai, UAE, 2011, Pending.
- Embarak, O., Corne, D. "Detecting Vicious Users in Recommendation Systems", 4th international conference on Developments in E-Systems Engineering - DeSE2011-Track 03: e-Business and Management innovations. Dubai, UAE, 2011, Pending.
- Embarak, O., Corne, D. "Using Semantic of ontologies for solving cold start in E-Commerce applications", ICITST 2011, 6th International Conference on Internet Technology and Secured Transactions - Multimedia & Web Services. Abu Dhabi, UAE, 2011, Pending.
- Embarak, O., Corne, D. "Feedback waves for Robustness analysis in Recommendation Systems", ICITST 2011, 6th International Conference on Internet Technology and Secured Transactions - Multimedia & Web Services. Abu Dhabi, UAE, 2011, Pending.

<i>Table of Contents</i>	<i>Page</i>
1. Introduction	1
1.1 Web personal recommendation, the cold start problem, and web privacy issues...	2
1.2 Web personal recommendation goals.....	3
1.3 Attributes of different approaches used for web personal recommendation	5
1.4 Collaborative filtering techniques.....	8
1.5 Recent trends and challenges of web usage personalization.....	9
1.6 Current techniques for solving the cold start problem.....	10
1.7 Motivation.....	11
1.8 Research statement	12
1.8.1 Problem definition.....	12
1.8.2 Research assumption.....	13
1.8.3 Method and contributions.....	13
1.9 Thesis structure.....	14
2. Background and Literature Review	16
2.1 Introduction.....	17
2.2 Web personalization meaning, stages, and aims.....	20
2.3 Categorization of web personal recommendation systems.....	23
2.3.1 Rule - based systems.....	22
2.3.2 Content based systems.....	23
2.3.3 Collaborative filtering systems.....	24
2.3.4 Hybrid systems.....	25
2.4 Collaborative filtering systems: associated techniques.....	26
2.4.1 Collaborative filtering techniques for memory-based systems.....	26

A. Similarity calculation.....	26
B. From similarities to recommendations.....	28
C. Computing the top-N items.....	29
2.4.2 Collaborative filtering techniques for model-based systems.....	30
A. Clustering-based collaborative filtering.....	30
B. Association rule based collaborative filtering.....	33
C. Sequential rule collaborative filtering.....	35
2.4.3 Graph theoretic collaborative filtering.....	37
2.4.4 Hybrid collaborative filtering systems	38
2.4.5 Summary.....	40
2.5 Previous personalization and recommendation systems.....	41
2.6 Conclusion.....	43
2.7 Recent trends in web usage personalization.....	43
2.8 Current web personalization challenges.....	46
2.8.1 The cold start problem.....	46
A) Demographic based recommendation.....	47
B) Stereotype recommendation.....	48
C) Case-based recommendation.....	49
D) Attributes-based recommendation.....	49
2.8.2 The Scalability problem.....	52
2.8.3 The Privacy problem.....	53
A) Privacy risks.....	54
B) Principles of applying fair information practice.....	55
C) Approaches used to reduce personalization privacy risks.....	56
2.8.4 The Diversity problem.....	59

2.8.5 The Robustness problem.....	60
2.8.6 The Data Sparseness problem.....	60
2.9 Evaluating web personalization systems.....	61
2.10 A novel approach to the cold start problem.....	63
2.10.1 Basic terminologies and concepts.....	63
2.10.2 Understand users' behavior and goals.....	64
2.10.3 Select the best routes (the best routes must survive).....	65
2.10.4 Recommending the latest valuable items.....	65
2.11 Summary.....	66
 3. The Active Node Technique.....	68
3.1 Introduction.....	69
3.2 Description and explanation of the Active node technique	72
3.2.1 Data collection and cleaning.....	74
3.2.2 Creation of sequential maximal sessions	74
A) Rules used to generate sequential maximal forward sessions.....	75
B) Algorithm for creating sequential maximal forward sessions.....	76
C) Calculate each session's time duration.....	77
3.2.3 Evaluation and absorption of maximal sessions.....	78
A) Significance of a sequential maximal sessions.....	78
B) Calculation of session page weights.....	83
C) Absorption process (<i>sessions absorbing other sessions that are subsets</i>).....	85
3.2.4 The Integrated routes profile	89
A) Algorithm for creating integrated routes	90
B) Abstract users profiling.....	91

C) Validity of the integrated routes profile	92
D) Incorporating new added items in the recommendation process.....	93
3.2.5 The recommendation process.....	96
A) Node recommendation rules.....	96
B) Batch recommendation rules.....	97
C) New items recommendation rules.....	98
D) The recommendation algorithm.....	99
E) Switching between node and batch recommendation.....	100
3.3 Evaluation Methods.....	100
3.3.1 Novelty level.....	100
3.3.2 Precision and coverage levels.....	101
A) Node recommendation evaluation methods.....	101
Precision and coverage levels in node recommendation.....	102
B) Batch recommendation evaluation methods.....	103
Precision and coverage levels in batch recommendation.....	104
C) New items evaluation methods.....	105
3.4 Summary.....	106
 <i>4. A Collaborative Filtering System Based on the Active Node Technique.....</i>	<i>107</i>
4.1 Introduction.....	108
4.2 Implementation of a system based on the active node method.....	109
4.2.1 Context of the proposed system	109
4.2.2 Data collection and preparation.....	111
A) Data collection and cleaning.....	111
B) Data Preparation.....	114

4.2.3 Pattern discovery phase using ANT.....	116
A) Evaluating the significance of maximal sessions.....	116
B) Absorption process.....	120
C) The Integration process.....	121
4.2.4 The Recommendation Phase.....	124
4.3 Alternative methods for solving the cold start problem	127
4.3.1 The Naïve Filterbot model.....	127
4.3.2 The Triadic Aspect Model.....	128
4.3.3 MediaScout stereotype model.....	133
4.4 Description of Experiments.....	134
4.4.1 Website chosen for online evaluation experiments	134
4.4.2 Methods and metrics for evaluation	137
4.4.3 Experimental Results.....	141
A) Level of novelty.....	141
B) Level of coverage.....	143
C) Level of precision.....	144
4.4.4 Conclusion.....	146
4.5 Summary.....	147
 <i>5. Augmenting the Active Node Technique with Semantic Information</i>	 <i>148</i>
5.1 Introduction.	149
5.2 Merging the active node technique with a semantic structure.	151
5.3 Updating items attributes within a semantic structure	154
5.3.1 Exploiting RDF/RDFS to support the concept of personal recommendation ..	155
5.3.2 The semantic update process.....	155

5.4 Some further detail, and dealing with the cold start problem	156
5.4.1 Item preference parameters in RDF statements.....	157
5.4.2 Dealing with new users and new items using semantic integrated routes.....	164
5.5 The basic ideas behind node and batch recommendations in the semantic ANT...	165
5.5.1 Prioritization of recommendations.....	168
5.5.2 The semantic ANT node and batch recommendation algorithms.....	171
5.6 Comparison and Evaluation.....	172
 6. Conclusion & Future Work.....	182
<hr/>	
6.1 The summary.....	183
6.1.1 The active node technique and the cold start problem.....	185
6.1.2 The active node technique and user privacy issues	185
6.1.3 Domain independence	187
6.2 Conclusions.....	187
6.3 Future work.....	188
 7. Appendices.....	191
<hr/>	
A. Technical user click streams analysis report.....	193
A.1 Access Resources.....	193
A.1.1 Top Access Pages.....	193
A.1.2 Single Accessed Pages.....	193
A.1.3 Number of Hits Per Page.....	194
A.1.4 Top Entry Pages.....	197
A.1.5 Top Exit Pages.....	198
A.2 Visitors Activities.....	198

A.2.1 Top Visitors by Number of Visits.....	198
A.2.2 Visitors who visit once.....	199
A.2.3 Repeated visitors.....	200
A.2.4 Average duration per visitors	200
A.2.5 Average visits duration for all visitors	201
A.2.6 Top Visitors by Duration (top twenty).....	201
A.2.7 Number of unique visitors.....	202
A.3 Site Navigation.....	205
A.3.1 Visitors popular paths through the web site.....	205
A.3.2 Max Path Length.....	242
A.3.3 Min Path Length.....	242
B. Suggested method modules.....	243
B.1 Data Flow Diagram Level (1)	243
B.2 System Flow Chart.....	244
B.3 Data preparation Flow Chart.....	245
B.4 System pattern discovery flow chart.....	246
B.5 System recommendation flow chart.....	247
C. Abbreviations.....	248
D. A glossary of terms.....	250
References.....	253

Figure 2.1: A classification of types of web-mining activity.....	17
Figure 2.2: The user's interaction with the web, adapted from Ackerman, 1997, p.22.....	18
Figure 2.3: The high-level web usage mining process (Mobasher, Dai et al. 2001)..	19
Figure 2.4: Demographic filtering (Drachsler et al., 2007).....	47
Figure 2.5: Case- based recommendation (Drachsler et al., 2007).....	49
Figure 2.6: Attribute-based recommendation (Kalz et al., 2008).....	50
Figure 3.1: Simple example to show user selections in red, and in yellow the selected candidates for recommendations.....	72
Figure 3.2: User online path(s) shows the extent of the overlapping between Individual and collective users desires.....	73
Figure 3.3: A website viewed as a network of nodes.....	75
Figure 3.4: Algorithm for creating maximal forward sessions from user's click stream.....	77
Figure 3.5: Region of acceptance and rejection.....	80
Figure 3.6: Region of acceptance and rejection with new added item.....	82
Figure 3.7: A user significant maximal forward session.....	84
Figure 3.8: Super and sub session.....	85
Figure 3.9: One session absorbs another session that is a subset of it.....	86
Figure 3.10: Absorption algorithm.....	88
Figure 3.11: Integrated route creation.....	89
Figure 3.12: Integrated routes algorithm.....	90
Figure 3.13: Users sessions profiling.....	91
Figure 3.14: Generating a virtual link to a new added item.....	95
Figure 3.15: Different stored routes.....	97
Figure 3.16: A simple illustration of batch recommendation.....	98

Figure 3.17: Recommendation algorithm.....	99
Figure 3.18: Different routes used for node recommendation evaluation.....	102
Figure 3.19: Different items used for node recommendation evaluation.....	102
Figure 3.20: Target set TS used for batch recommendation evaluation (these are items that were selected by users in a training phase – see Chapter 5 – after visiting node <i>D</i>)......	103
Figure 3.21: Evaluation set for batch recommendation.....	104
Figure 4.1: Context diagram for web personal recommendation system.....	109
Figure 4.2: General model for collaborative system based on the active node technique.....	110
Figure 4.3: Data collection and preparation phase.....	111
Figure 4.4: Server log file raw data format.....	112
Figure 4.5: A Cleaned Log file.....	113
Figure 4.6: Active node online and offline phases.....	116
Figure 4.7: A visualization of the process that generates integrated routes from user click streams.....	124
Figure 4.8: Candidate items for node recommendation.....	125
Figure 4.9: Candidate items for batch recommendation.....	126
Figure 4.10: AlArabiya.net website main interface.....	135
Figure 4.11: Users' demographical features.....	136
Figure 4.12: User online maximal path and the expected target set.....	137
Figure 4.13: Complete maximal path.....	137
Figure 4.14: Novelty of recommendations.....	142
Figure 4.15: Coverage of recommendations.....	144
Figure 4.16: Precision of recommendations.....	145
Figure 5.1: Semantic web recommendation cycle.....	152
Figure 5.2: Semantic active node.....	153

Figure 5.3: Update items attributes in semantic structure.....	154
Figure 5.4: Main items' impact values and the associated virtual items' relative weights.....	156
Figure 5.5: Example using RDF statements.....	158
Figure 5.6: Describing a semantic item.....	158
Figure 5.7: Insert item preference parameters in RDF statements.....	159
Figure 5.8: Integrated collected preferences within the semantic structure.....	160
Figure 5.9: A web item's semantic code.....	161
Figure 5.10: A web item with its virtual linked nodes in semantic format.....	162
Figure 5.11: Semantic virtual web sites.....	163
Figure 5.12: Recommendation cycle based on semantic active node technique.....	164
Figure 5.13: Items in semantic and virtual relations.....	166
Figure 5.14: Two main items in a semantic relationship.....	166
Figure 5.15: Generating node recommendations for node X.....	168
Figure 5.16: Sample of the generated semantic classes.....	173
Figure 5.17: A web node's semantic properties.....	174
Figure 5.18: News node as a super and sub classes.....	174
Figure 5.19: A node associated with its properties.....	175
Figure 5.20: A node in semantic and virtual relationships.....	176
Figure 5.21: An XML structure of the generated semantic ontology.	176
Figure 5.22: Semantic and non-semantic active node novelty.....	178
Figure 5.23: Semantic and non-semantic active node coverage.....	179
Figure 5.24: Semantic and non-semantic active node precision.....	180
Figure 6.1: Illustrating a user's online session.....	186

Table 2.1: Advantages and disadvantages of different recommendation methods with reference to the cold-start problem.....	52
Table 3.1: Significance calculation example.....	80
Table 3.2: Significant and insignificant sessions.....	81
Table 3.3: Calculate the significance of a session with new element.....	82
Table 3.4: Significant and insignificant sessions with new items.....	83
Table 3.5: Relative weight of the items in the session of Figure 4.7.....	84
Table 3.6: Super and sub session items relative weights	85
Table 3.7: Example of recalculation of items' weights after absorption.....	87
Table 4.1: Sample of maximal forward sessions created by the implemented system.....	115
Table 4.2: Some calculated impact values.....	117
Table 4.3: Selected significant sessions.....	118
Table 4.4: Relative weights of items in different sessions.....	119
Table 4.5: Duplicated significant sessions.....	119
Table 4.6: Absorbed sessions.....	120
Table 4.7: Sample of created integrated routes.....	123
Table 4.8: Sample of generated node recommendations.....	125
Table 4.9: Sample of generated batch recommendations.....	126
Table 4.10: Example item features.....	129
Table 4.11: Users' demographic triples.....	130
Table 4.12: Users per categories.....	130
Table 4.13: Users-Items weight matrix.....	131
Table 4.14: Distribution of features against users.....	131
Table 4.15: Distribution of features against Items.....	132

Table 4.16: Online maximal session.....	138
Table 4.17: Target Sets associated with the maximal session in Table 5.16.....	138
Table 4.18: Match between target sets and recommendation sets.....	139
Table 4.19: Calculating precision.....	140
Table 4.20: Novelty values for different methods.....	141
Table 4.21: Coverage values for different methods.....	143
Table 4.22: Precision values for the tested recommendation methods.....	144
Table 4.23: Comparing the active node technique with alternative techniques.....	146
Table 5.1: Main-to-virtual item priority levels.....	169
Table 5.2: Example showing main-to-main priority levels.....	170
Table 5.3: Semantic and non-semantic active node percentage of novelty.....	177
Table 5.4: Semantic and non-semantic active node percentage of coverage.....	178
Table 5.5: Semantic and non-semantic active node percentage of precision.....	180

Chapter 1

Introduction

1.1 Web personal recommendation, the cold start problem, and web privacy issues.

Nowadays, web users interact with many web personal recommendation systems in e-learning, e-commerce, e-news, e-media, e-travel guide, and so forth. These systems aim to find the most interesting and valuable information for web users by suggesting items of interest to users based on their explicitly or implicitly collected preferences. Many approaches are used to create recommendation systems, the collaborative filtering approach is the most successful and widely used. However, systems based on a collaborative filtering approach suffer from several problems such as the cold start problem (for example if a new user visits Amazon web site for first time or if a new item is added to the site, then the Amazon system becomes unable to generate sensible recommendations). The privacy problem (which reflects the users' concerns regarding the misuse of their collected personal data), the scalability problem, the diversity problem, etc. In this thesis, we suggest a method that provides high quality recommendations, and solves the cold start problem, also taking into account the privacy problem.

The cold start problem is divided into the user cold-start problem and the item cold-start problem. *The user cold-start problem* happens when there is a new user in the system for whom no rating information is available (e.g. using Amazon for first time). Hence, a collaborative filtering system does not have enough information to estimate similarity between him/her and the others, so that the system will be unable to make recommendations or only create poor recommendations. While *the item cold-start problem* occurs when there is no rating information for a new added item to the web (e.g. a new book added to Amazon), and hence measuring similarity between the new added items and the old items becomes very difficult. Therefore, the system won't be able to recommend any new added item until it can measure similarity between this new item and the old ones (Park and Chu, 2009).

Personal recommendation systems employs data mining and/or collaborative filtering to predict contents (or items) that likely to be of interest to users. These systems can be particularly effective when the user identifies himself explicitly to the web site, e.g. e-commerce web sites are increasingly introducing personal features in order to build and retain relationships with customers and increase the number of purchases made by each customer. Although, individuals appreciate personalization and find it useful, but

personalization raises a number of *privacy* concerns ranging from user discomfort with a computer inferring information about them based on their purchases, to concerns about identity thieves, or the government gaining access to the users' profiles. In some cases, users provide personal data to a web site in order to receive personalized services despite their privacy concerns, while in other cases; users may turn away from a site because of privacy concerns. Our work focuses on finding a proper framework for a collaborative filtering system, which avoids the cold start problem, and considers privacy concerns.

This chapter is divided into nine sections. *Firstly*, as shown earlier; we introduced web personalization, the cold start problem, and the privacy problem. *In the next section*, we demonstrate some web personal recommendation goals. *In section three*, we explicate different classification criteria for approaches used for web personal recommendation. *In section four*, we briefly explain some techniques used for collaborative filtering. *In section five*, we explore some recent trends in web usage personalization. *In section six*, we briefly sum up current techniques used to solve the cold start problem. *In section seven*, we reveal our motivation. *In section eight*, we demonstrate the research question and area of the thesis. *Finally*, in section nine we show the thesis structure.

1.2 Web personal recommendation goals

Recommendation systems are widely used on the internet, they are used for movies, news, e- learning, e-commerce, and travel web systems...etc. Many goals are achieved from adapting recommendation systems in real life; we will summarize some of these (inter-related) goals as follows.

1. Increase online purchases. Good recommendation systems can significantly increase the likelihood of a customer making a purchase.
2. Provide proper recommendations. Online users get recommendations for many of their everyday activities including movies, interesting travels, music concert .etc.
3. Save users' time. Recommendations that help users to find what they looking for in less time.

4. Understand users' real desires. These systems record actual user behavior, and therefore, they capture objective and real knowledge about their users.
5. More offers available. An increased range of products and information become available to customers through recommended items or information to users.
6. User directed sites. Online contents become adaptable, and reflect the actual desires and preferences of users.
7. Suggest useful courses. E-learning recommendation systems provide appropriate course recommendations based on users' levels and based on similar users, which is useful for the e-learning process as well as for learners.
8. Novel recommendations. Since recommendation systems adapt based on collected data about users' desires, which change from time to time, then these systems will be able to provide proper and novel recommendations.
9. Create trust. By understanding users well, a system can provide unexpected and novel recommendations to them which are appropriate, and make users feel comfortable and trusting in the system.
10. Always up-to-date. The collected attributes of users help developers to automatically update the system with information, products etc that reflect current users' needs.
11. Improve an organization's web site structure (e.g. by ensuring most recommended items are easier to find).
12. Increase loyalty. When users feel that the system knows what they like and what they do not like, and provide proper recommendations to them, this makes them more loyal to the web system.

The ability of personal recommendation systems to provide its goals depend on the efficiency of collected data about users' desires. These collected data take different shapes based on different approaches. Explanation of these approaches is provided in the following section.

1.3 Attributes of different approaches used for web personal recommendation.

Web personal recommendation approaches are classified here based on the way the different approaches used to different aspects of the recommendation task. For example, recommendation models sometimes predict a rating for items not currently rated by the user and then show highly-rated items to the user; in contrast, selection based recommendation models select the N most relevant items for a user and show only the selected N items (Anand and Mobasher, 2005). There are many different classification of web personal recommendation systems, but in this section we will summarize some selected classification criteria.

Implicit Vs Explicit data collection

Implicit data collection. Users are not providing their preferences explicitly but their preferences are inferred from their selections and click streams. For example, a user's search queries and purchase history such as in (Linden et al., 2003) Amazon.com recommendation which use item-to-item collaborative filtering for generating recommendations. Several systems use users' click streams (stored in log files) to infer interests or preferences using association rules such as in (Mobasher et al., 2001; Srivastava et al., 2000). Some systems use both content structure and user behavior for generating more accurate recommendations such as (Eirinaki et al., 2005). Other systems use personal agents to collect preferences and generate recommendations (Good et al., 1999).

Explicit data collection. Personalization is considered as a conversational process that needs explicit interaction with the users as they search for items of interest. This is a form of case-based reasoning, and several systems are case-based systems (Burke, 2000; Lorenzi and Ricci, 2005; McGinty and Smyth, 2005) which use critiquing to improve the performance of recommendation process. Some systems use explicit ratings feedback, where the user must rate all recommended items based on their fit to his desires (Ginty and Smyth, 2002). In preference feedback, the user chooses one of the recommendations that best suits his requirements, and then uses his selection (feedback) to recommend similar items (Burke et al., 1997 ; Fesenmaier et al., 2003; Shimazu, 2002).

Duration

Task-focused personalization is a rule-based system the most appropriate way of providing task-focused personalization is to make recommendations based on actions a user has taken while performing a task. For example, if a user purchases a ticket to Egypt at a travel web site then the web site might suggest books about pyramids, tours and travel in Egypt, etc. Such personalization is based on information provided by or inferred from the user during the current session or while completing the current task. Also known as memory based system, traditional collaborative filtering (Maes, 1994) and content based systems (Yang, 1994), these are examples of memory based approaches which can use associative networks (O'Riordan and Sorensen, 1995) and also use ontology profiles (Sieg et al., 2005).

Profile-based personalization. Many personalization systems develop profiles for users and explicitly add provided or inferred information about users each time they return to the site cookies (Eirinaki and Vazirgiannis, 2003) or IP addresses (Kosala and Blockeel, 2000) are used to recognize returning visitors automatically and retrieve their stored profiles, or users may be asked to login to the site. This is also known as a model-based approach, where users' patterns are collected and stored in their profiles and then used to generate recommendations. Several systems are model-based systems such as (Sieg et al., 2007) who create an ontological profile from a user's search to provide personalization, (Chen et al., 2007) who create a private dynamic user's profile to provide content recommendation, and (Shokri et al., 2009) who create aggregate offline profiles to provide collaborative filtering recommendations.

User involvement

User-initiated personalization. Some sites offer users the option of selecting customizations and display packages of interest, or news related to topics the user has selected. Users might also select their preferred page layout for information or the number of items they want displayed (Mulvenna et al., 2000), or they might provide information about their display and bandwidth constraints and ask to have a site optimized accordingly (Good et al., 1999).

System-initiated personalization. Some sites attempt to personalize content for every user, even if users do not request customized features and take no explicit actions to request personalization (McGinty and Smyth, 2005). In some cases, sites provide a way for users to cancel personalization (Weld et al., 2003).

Reliance on predictions

Prediction-based personalization. Some sites use users' explicit or inferred ratings to build users' profiles that can be compared with the profiles of other users. When users with similar profiles are discovered, the system predicts that they will have similar preferences and offers recommendations to one user based on the stated preferences of the others. Such systems are often referred to as *recommender* systems or *collaborative filtering* systems, e.g. (Resnick et al., 1994; Balabanovi and Shoham, 1997), which recommend items that are rated highly by users similar to the active user.

Content-based personalization. Some sites use the specific requests or other actions of a user to trigger automatic personalization. For example, if a user buys a book on web usage personalization, the site may suggest other books on web usage mining. In this case the site is not using ratings to predict other types of books the user might like to buy, but simply offering the user additional books on the same topics as the book he/she already bought. Such systems build individual 'like' and 'dislike' profiles for each user. The NewsWeeder system (Lang, 1995) creates users' profiles from feedback collected about their rating of articles on a scale of 1 to 5, then these profiles are used to recommend articles to users.

Item Vs User information

Item related information. Some systems create recommendations based on content description of items (Lang, 1995) and/or products domain ontology (Ghani and Fano, 2002); these systems generate profiles from unstructured data related to items.

User related information. These systems depend on users' demographical data (Pazzani, 1999), where demographical data are collected from users' home pages, and a text classifier which classifies users to learn characteristics of home pages associated with users who like a particular restaurant. In Lifestyle Finder (Krulwich, 1997), collected demographical data

directly from users, classify users into 62 demographic clusters, and then recommend items relevant to each user demographic cluster. Other systems collect a user's behaviors such as his online purchased items, or his online click streams stored on log files (Mobasher, 2005).

1.4 Collaborative filtering techniques

Many collaborative filtering techniques are used to provide personal recommendations; in this section I will summarize some of these techniques.

Traditional collaborative filtering. Collaborative filtering provides an alternative to content based filtering; collaborative filtering systems utilize the benefits of collected and created users profiles (Goldberg et al., 1992). Feedback collected from users is used to find likeminded users as well as to recommend items to the active user. Some collaborative systems use matrices to measure similarity between users using Pearson and Spearman Correlation (Resnick et al., 1994), using cosine angle distance (Sarwar et al., 2000), and some other systems use Mean-square difference and constrained Pearson correlation (Shardanand and Maes, 1995). In order to increase the accuracy of recommendation and reduce the size of neighborhoods, some systems use a threshold parameter to restrict selection to a subset of users who are in the neighborhood of the active user based on a predetermined threshold (Shardanand and Maes, 1995). Herlocker et al., 1999 proposed the use of significance weighting to measure how dependable is the measure of similarity between two users, where two users are significantly more similar if they share common interests in fifty items than if they share common interests in only three items.

Item-based collaborative filtering. This method usually involves building an item similarity matrix based on users' ratings of these items, and then recommending items with high similarity to the selected item by the active user. ***Clustering-based techniques*** are important in this context; item based clustering and user based clustering are the most common clustering methods. Various clustering algorithms are used including K-Means and user-based clustering (Ungar and Foster, 1998), hierarchical agglomerative clustering (O'Connor and Herlocker, 2001), item and user based clustering (Kohrs and Merialdo, 1999). Breese et al., 1998 described a mixture-resolving algorithm to cluster users based on their items ratings. In ***association and sequence rule based techniques***, association rules or

similar are learned and used to infer items to recommend (Padmanabhan and Tuzhilin, 1999; Silberschatz and Tuzhilin, 1996; Tan et al., 2004). Simple approaches do not consider the order in which items were accessed, while sequential pattern discovery considers the order of items when discovering frequently occurring item-sets (Baumgarten et al., 2000; Agrawal and Srikant, 1995; Mobasher et al., 2002). In sophisticated versions of this method, the ratings of items are transferred into a directed graph, where nodes represent users and edges represent predicted ratings of a user based on the ratings of another user (Aggarwal et al., 1999). Many systems use a mixture of these previously mentioned collaborative filtering techniques to avoid some deficiencies of using only one technique (Nakagawa and Mobasher, 2003). In the next chapter, we will provide more details about these collaborative filtering techniques.

1.5 Recent trends and challenges of web usage personalization

Web personal recommendation systems aim to provide users with what they are looking for efficiently and in less time. Many approaches are used to achieve these goals such as: Individual vs collaborative, reactive vs proactive, user information vs item information, memory based vs model based, client side vs server side. Most current web personalization systems are collaborative filtering systems that depend on both user behavior and item ratings. Several challenges direct web personalization researches, such as the cold start problem that occurs when a new user or a new item has just entered the system. Privacy issues reflect users' irritation associated with the use of their personal data (collected by recommendation systems) by a third party. A scalability problem also occurs because of tremendous growth in users and items, which leads to the need for more computations and resources. A diversity problem also occurs with the diversity of items in the recommendation list, and this can badly affect users' satisfaction, especially in item based collaborative systems. A robustness problem occurs when an interested party intentionally influences item recommendation by inserting false ratings. We provide more discussions and explanations for some of these challenges in chapter two.

1.6 Current techniques for solving the cold start problem

As described previously, the cold start problem in collaborative personal recommendation systems happens when the system has insufficient information about the new user or about a new added item in the system, and hence the system becomes unable to generate recommendations for that user or involving that item, and may even generate inaccurate recommendation. The cold start problem is divided into the user cold start, the item cold start, and the system cold start problems. ***The user cold start problem*** happens when a new user enters the web system and then the system has no information about him/her in order to generate recommendations (e.g. a user enters the E-bay web site for his first time). ***The item cold start problem*** happens when a new item is added to the site and then the system has no information about the item ratings and hence will not be able to generate justified recommendations (e.g. a new item is added to the E-bay web site). ***The system cold start problem*** happens when the system starts working for the first time, and then the system has no information about item ratings and about user preferences, therefore it will not be able to generate proper recommendations.

Several methods are provided to solve the cold start problem; some systems use user demographical data, others use user stereotypes, or item attributes. In approaches that use demographical data, when the user enters the system for first time, then the system will request him to fill a form about his age, gender, income, religion, marital status, language, ownership (home, car, etc), social position, etc... . These data are used by the system to find similar users among users that already have a history using the system. Systems that create user stereotypes create a specific image with specific meaning about users (often held in common by people about another group), and then generate recommendations which could be directed to each stereotype category. Although the demographical base method and stereotype based method provide a solution to the cold start problem, both of them depend on users filling in forms about their personal data, and therefore are unsatisfactory in regard to privacy issues. Also they only solve the user cold start problem, but only as well as the generated user classifications reflect the actual desires of the new users.

Some systems used case-based recommendations, which determine each item's attributes, and then generate an item attributes similarity matrix between each new added item and the established items. Although this provides a solution to the item cold start problem, it does not

solve the user cold start problem. This also potentially leads to an over-personalization problem, and it often needs manual determination of each item's attributes. Some other systems are attribute based systems that collect data about both items and users; they keep full records about item attributes, and then, by forcing users to fill in forms or give their interests, they generate recommendations based on the match between item attributes and user interests. This method ignores user privacy and generates static recommendation, since user profiles are static, and levels of novelty in recommendation sets are very low.

In this thesis, we suggest a method to solve the cold start problem, which focuses on trying to infer users' real preferences, from the clues provided by 'integrated routes' (constructed from real click streams) without forcing users to fill in any forms, and without collecting any personal or demographical data. Therefore, we aim to solve the cold start problem (user and item), while bearing in mind the privacy concerns.

1.7 Motivation

The main goal for any web personalization system is to convey useful information to web site visitors. Collaborative filtering systems (CFS) create recommendation sets based on each user's historical collected data. Without collectig users' historical preferences, traditional collaborative filtering systems face a problem of creating recommendation set for any new visitor. The first visit represents a big issue in the web personalization context; as we explained earlier that the lack of available information about any new user puts him off the system before the system has been able to gather the required data to provide recommendations. Web Personalization can be particularly effective when the user identifies himself explicitly to the web site, but numerous privacy concerns arise ranging from users' discomfort with computers inferring information about them based on their purchases to concerns about identity thieves, or the government gaining access to personalization profiles.

In order to find solutions for the cold start and privacy problems we suggest using multiple abstract user profiles, as well as using semantic relationships and virtual relationships between items. Abstract profiles that reflect users' preferences will be created, such as front-end profiles, abstract back-end profiles, universal profile, and integrated routes profiles (all to be explained later in this thesis). As soon as a specific user visits the site, our

system automatically will create a front-end profile that will contain the active path of the current user; where the active path is the followed path by the online user. This profile is a temporary profile that should be removed as soon as the user exits from the site. During user movements from node to node, our system should create a recommendation set based on his online front-end profile comparing with the integrated maximal forward routes (an aggregate representation of routes undertaken by previous users), The suggested recommendation can be based on the current user's active location (node recommendation type) or based on his online path (batch recommendation type). *Node recommendation* mode will create a recommendation set based on the current active node (where the active node is the current visited node or page by the online user). While in *Batch recommendation*, we will create recommendations based on the online maximal path; where a maximal path refers to the online non-cyclic and sequential order of all visited nodes by a site user, where each item on the user online maximal path can be involved in the creation of recommendation set.

1.8 Research statement

This thesis proposes a new method for solving the cold start and privacy problem. The method is fully described, implemented and tested in later chapters. In this section, we briefly demonstrate the main directions and elements of our research.

1.8.1 Problem definition

How can we provide an appropriate solution for the cold start problem? Moreover, how can we handle the privacy problem? These represent the main challenges that we will try to solve in this research using the suggested technique. As we explained earlier, the cold start problem refers to new users with no interaction history and no profile, therefore the system becomes unable to personalize its interactions to the user. The same problem arises by adding a new item to the web site where systems cannot recommend the new added items before collecting a considerable history of item ratings. Although, web personalization systems try to predict the contents that are likely to be of interest to the visitors, users become more concerned about their privacy. Because of the computer's predictions and misuse of their collected data, so that creating a web personal recommendation system within the boundaries of privacy is one of our goals, so it is possible to identify users online based on

inferring their browsing targets, to generate recommendations while taking into consideration their privacy concerns.

Privacy protected personalization from first visit represents the main goal for this research. We try to achieve this goal using multiple abstract profiles instead of using personal data, generating recommendations based on an aggregated data structure capturing key elements of user's routes through the system. We should mention here that the user maximal path refers to non-cyclic sequential order of all visited nodes by a sit user, and the current path reflects the user current online visited maximal path.

1.8.2 Research assumption

We claim it is possible to provide privacy-protected recommendations from first time by using the assumption that *“when different users have similar paths through a site, they have similar browsing targets”*. Given a set of abstract users' click streams on a specific web site as inputs, and by implementing the suggested method; our system will provide recommendations for site visitors without forcing them to provide personal data.

Users' click-streams can be collected using servers log files or by using online data collection. Whatever the used data collection method, the collected data should be put in a format that is suitable for further processing. Discovering significant clickstream patterns will be based on the collected data in its abstract form, without identifying users; after processing these data, the system will create recommendations based on a process that matches the active user's click-streams against a stored integrated route profile built from previous abstracted route data.

1.8.3 Method and contributions

The approach we propose in this thesis is called the Active Node Technique (ANT). A summary of the ANT is as follows. *Firstly*, we collect loopless abstract maximal clickstream sessions. *Secondly*, we generate 'integrated routes' that represent the largest abstract loopless routes visited by abstract users through their clickstreams on the specific web site. *Thirdly*,

recommendation sets are generated based on visited subsets of the maximal online session, based on matching them to stored integrated routes.

In this thesis, we evaluate the ANT by comparing it to alternative methods. In evaluation experiments we calculate the *novelty*, level of *precision*, and level of *coverage* in the generated recommendation sets comparing with alternative techniques used for the cold start problem. The thesis also describes and evaluates a version of the ANT that is implemented, and suitably adapted, for a semantic web environment, using semantic ontology concepts.

Contributions; web personal recommendation can be described as any action that makes the web experience of a user customized to the user's preferences. In relation to solving the cold start problem and privacy problems, several ideas are provided in this thesis as follow.

- A novel solution to the cold start problem (the Active Node Technique), which is introduced and explained in chapter four and tested in chapter five.
- A novel technique to solve the privacy problem in personal recommendation systems (another aspect of the Active Node Technique), this is introduced and explained in chapter four and tested in chapter five.
- a novel way to measure recommendation novelty, as well as new coverage and precision evaluation formulas, which are introduced in chapter four and implemented in chapter five.
- A further demonstration and adaptation of the Active Node Technique in the semantic web context, which is explained and evaluated in chapter six.

1.9 Thesis structure

The thesis organized as follows. In **Chapter 1**, we introduced web personal recommendation, the cold start and privacy problems, web personal recommendation goals, classification criteria, recent trends and challenges, our motivation, and the research statement. In **Chapter 2**, we provide background and literature review for web personal recommendation systems, its techniques, challenges (specially the cold start problem and the privacy problem), and evaluation criteria. In **Chapter 3**, we describe the suggested active node technique and the suggested evaluation methods. In **Chapter 4**, we discuss the

proposed privacy-protected collaborative system model based on the active node technique, and describe the selected alternative methods for comparison, and we describe experiments and show results. In **Chapter 5**, we describe how to improve coverage and precision levels by a marriage between the suggested method and semantic ontology structures. In **Chapter 6**, we provide our conclusions and suggest future works.

Chapter 2

Background and Literature Review

2.1 Introduction.

Web mining refers to the applications of data mining that extract knowledge from web data including web documents, hyperlinks between documents, and usage logs of web sites. Therefore, web mining techniques focus on extracting knowledge about web contents, structure, and usage data. There is no difference between web content mining and general data mining, since it focuses on how to extract knowledge, whether the content was obtained from the web, a database, a file system, or through any other means. As shown in Figure 2.1, Web content can be varied, containing text and hypertext, image, audio, video, records, etc. Mining each of these media types is by itself a sub-field of data mining.

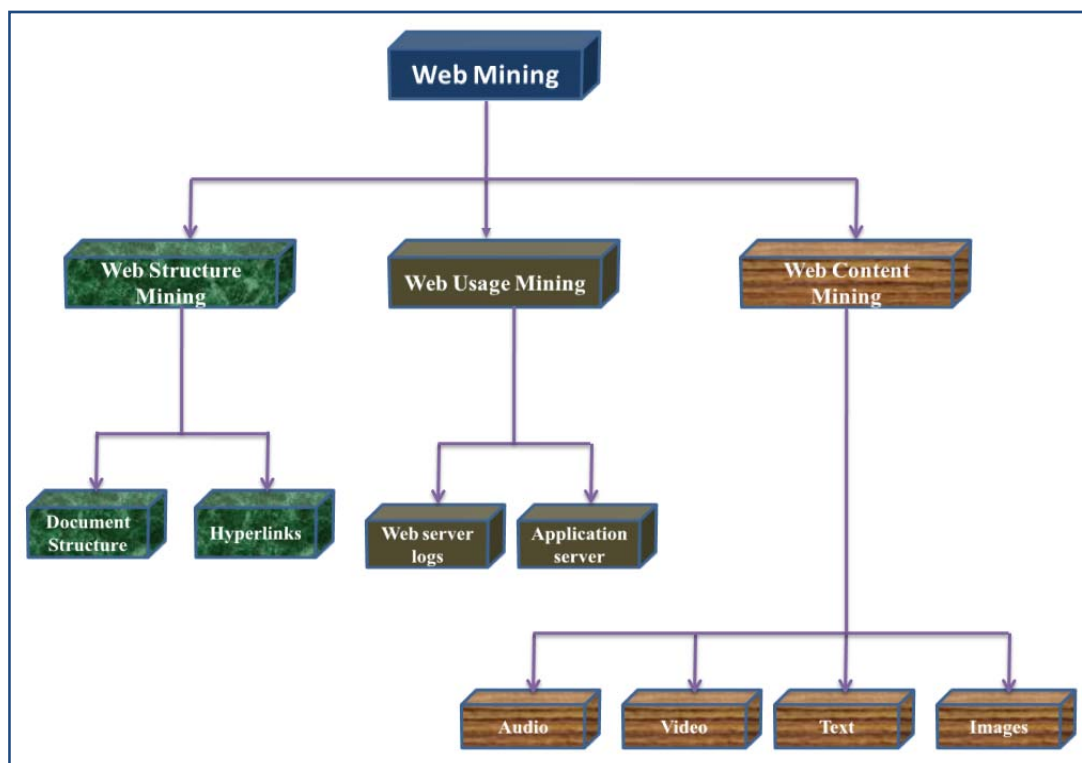


Figure 2.1: A classification of types of web-mining activity.

Web usage mining; as a data mining application that depends on the data collected from users' interactions with the web, has greatly concerned both academia and industry in recent years. Users' interaction patterns with the web are recorded in web data log files. The goal of web usage mining is to capture and model the behavioural patterns and profiles of users interacting with a web site (Kosala and Blockeel 2000). Groups of users with common needs or interests usually help to discover patterns as collection of pages or items that are accessed

frequently. Detecting user access patterns is useful in numerous applications: supporting web-site design decisions such as content and structure justifications (Perkowitz and Etzioni 2001); optimizing systems by enhancing caching schemes and load-balancing, making web-sites adaptive (Nakagawa and Mobasher 2003); supporting business intelligence and marketing decisions (Anand and Mobasher 2005); testing user interfaces, monitoring for security purposes, and more importantly in web personalization applications such as recommendation systems and target advertising (Etzioni and Perkowitz 2000).

Web server log files provide a list of page requests made to a given web server in which a request is addressed by, at least, the IP address of the machine placing the request, the date and time of the request, and the URL of the page requested. Using such data it is possible to reconstruct the user navigation sessions within the web site, where a session consists of a sequence of web pages viewed by a user in a given time window. A log entry is automatically added each time a request for a resource reaches the web server. While this may reflect the actual use of the resources on a site, it does not record behaviour like frequent backtracking or frequent reloading of the same resource when the resource is cached by the client browser or a proxy as shown in Figure 2.2.

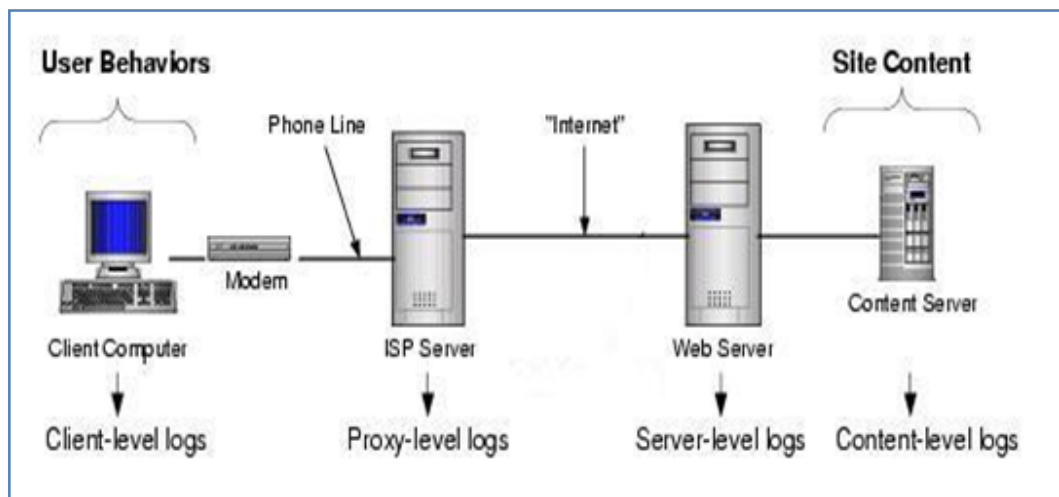


Figure 2.2: The user's interaction with the web, adapted from Ackerman, 1997, p.22.

It is important to note that the entries of all users are mixed in the log, which contains simply ordered chronological events, although one single page request from a user may generate multiple entries in the server log. One major problem in web log mining is how to

identify unique users and associate users with their access log entries. The web usage mining process is divided into three phases as shown in Figure 2.3, *data pre-processing* (used to select, clean, and prepare the log raw data), *pattern discovery* (application of data mining algorithms, such as association rules, sequence analysis, etc.), and *pattern analysis* (evaluation of yielded patterns to seek unknown and useful information) (Mobasher 2005). There are two distinct directions; in the first approach, we map user sessions onto relational tables and an adapted version of standard data mining techniques, such as mining association rules. In the second approach, statistical techniques are developed and invoked directly on the log data (Borges and Levene 2004).

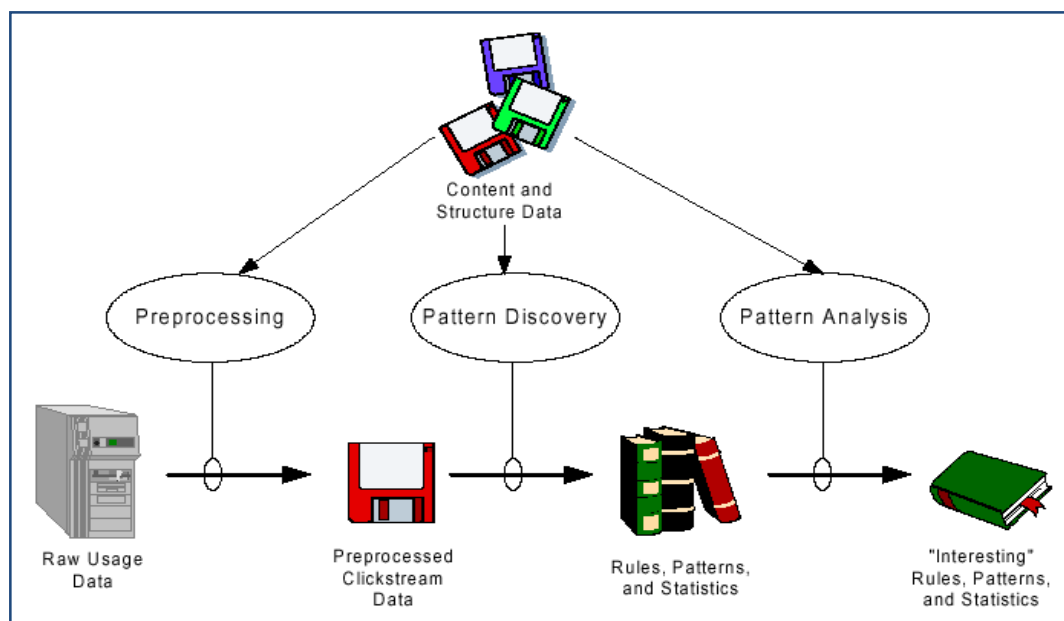


Figure 2.3: The high-level web usage mining process (Mobasher, Dai et al. 2001).

Personalization typically employs data mining to predict the content that is likely to be of interest to visitors. It can be effective if we can determine the interests of users (visitors, or customers). Hence, collecting data about users is necessary for recommendation; this is done explicitly by asking the user, or implicitly by observing user behavior, or by a combination of these. Creating users' profiles based on the collected data is useful for personalization purposes.

2.2 Web personalization meaning, stages, and aims.

One of the important applications of web usage mining is *web personalization*. The idea is to exploit usage data to customize a web site for individuals by identifying the needs and preferences of each individual user, and customize the content and/or structure of the web information system based on the user needs. Han and Kamber 2006 defined web usage mining as automatic discovery and analysis of patterns in click-stream and associated data collected or generated because of user interactions with web resources on one or more web sites. Several goals are achieved from using web usage mining such as:

1. Capturing, modeling, and analyzing the behavioural patterns and profiles of users interacting with a web site.
2. Discovering user patterns that can be used for decision-making.
3. Gaining better understanding of site visitors' needs and desires.
4. Creating a more efficient or useful organization for the web sites, and to do more effective marketing campaigns.

Based on previously mentioned ideas, we can define *Web personalization* as the process of automatically and efficiently customizing web sites to fulfill users' requirements.

Web personalization systems consist of three phases (Mobasher 2007) ***Data Preprocessing***, ***Pattern Discovery***, and ***Recommendations***. Two types of data can be collected for the data preprocessing phase: ***explicit data*** and/or ***implicit data***. **Explicit** data refers to data collected using online registration forms where users themselves enter such data to the system, although some users deliberately provide incorrect data. **Implicit** data refers to the data collected using stored log files that can be collected from server or proxy levels, where all users' click streams are recorded in log files so that it expresses the actual user movements in the web site.

The main goal of the ***data preprocessing phase*** is to put data in a suitable form required for the pattern discovery phase, while the ***Pattern discovery phase*** aims to discover each user's patterns to detect his/her interests. In particular, web usage-mining techniques (such as clustering, association rule mining, and navigational pattern mining, etc.) rely on offline pattern discovery from user transactions. In the ***Recommendation phase***, an agent suggests a

set of items for each online visitor based on his/her profile or based on similarities between the user and others.

Personalization systems mainly depend on the data collected about customers, which reflect the interests of the users and their interactions with applications and items. Different personalized systems differ, not only in the algorithms used to generate recommendations, but also in the manner in which user profiles are built. Content-based and rule-based personalization systems generally build an individual profile of user interests that is used to tailor future interactions with only that user. Personalization systems that are content-based filtering systems require extraction of item features from the item description, or extraction of information about relationships between items. Generating users' profiles in such systems can be explicit, or implicit, where the system observes the user behavior and then uses various heuristics to classify items as interesting or non-interesting to that user (Mladenic 1996), while explicit profile creation depends on the user assigning ratings to items or manually identifying positive and negative examples (Pazzani and Billsus 1997). A major disadvantage of approaches based on user profiles is the lack of serendipity as recommendations are very focused on the user's previous interests, as well as the system depending on the availability of content descriptions of the items being recommended.

E-commerce web systems sometimes depend on demographical profiling which reflects personal demographical attributes, and may include some computed attributes such as total amount spent as well as the frequency of purchases or visits. Although some systems use demographical data for personal recommendation, demographical data are difficult to collect, violate privacy concerns, and often provide poor quality recommendations, which do not reflect the actual interests of visitors (Pazzani 1999).

Traditional collaborative filtering uses users' ratings to create individual profiles, while non-traditional collaborative filtering systems rely on user-to-user similarities, where profiles are represented as a vector of ratings providing the user's preferences on a subset of items. Where an active user's profile is used to find other users with similar preferences, these similar users are said to be in the same 'neighborhood'. On the other hand, hybrid collaborative filtering approaches utilize both content and user rating data to create user profiles (Melville, Mooney et al. 2002). Other systems may use ontological data for user

profiling and thus may require a more complex representation than the flat representations used in standard approaches (Ziegler, McNee et al. 2005).

Regardless of the algorithm used for web personalization, data collected for user profiling, as indicated before, can be collected implicitly or explicitly. Explicit data collection needs user participation and is collected using online registration or survey forms, or by providing personal and financial information during a purchase. However, implicit data collection uses click streams or other types of behavioural data that does not require users to devote time for participation, and mostly users do not know that they are being monitored. Many e-commerce systems, such as Amazon.com, monitor customer's online purchase activity and use the collected information to create user profiles (Lee, and Cheung 2009).

Generally, web usage personalization aims to capture and model the behavior patterns and profiles of users interacting with a web site, understand behavior characteristics of visitors or user segments, improve the structure and /or content of the site, and recommend a set of useful objects (pages) to the current user(s) (active user).

2.3 Categorization of web personal recommendation systems.

Different approaches used for personalization purposes lead to different methods for creating visitors' profiles and predictions. Personalization systems can be categorized into three types: *Rule based systems*, *Content filtering systems*, and *Collaborative filtering systems*. A brief description of each is explained as follows:

2.3.1 Rule-based systems.

Such systems rely on manual or semi-automatic decision rules to generate recommendations for users. Most E-commerce web sites implement rule-based personalization where the web site administrator plays a vital role in specifying personalization rules that are highly domain dependent and reflect the specific objectives of the business web site. Users' profiles are created explicitly by asking users to fill in online forms (Pazzani 1999); the data collected about users is often demographical or personal (Srivastava, Mobasher et al. 2000), the content served to users is affected by pre-specified rules that relate stereotypical ideas of appropriate recommendations based on demographic

and personal profile (Pazzani and Billsus 2007). The advantages and disadvantages of rule-based systems are shown below.

Advantages of rule-based systems

- 1- Rule-based user profiles are very simple;
- 2- Users participate in creating their own profiles;
- 3- Appropriate for online advertising campaigns;
- 4- The site manager has much control over the content served to a particular user;
- 5- Faster than other personalization systems.

Disadvantages of rule-based systems.

- 1- The data collected may be incorrect;
- 2- Explicit data collection often represents a burden to the users;
- 3- Users' profiles are static;
- 4- Users' profiles are subjective;
- 5- The rules are highly domain dependent and reflect particular market objectives;
- 6- The system performance degrades over time as the profiles age;
- 7- Users' profiles that are based on demographic data are less accurate than those based on item content or usage data;
- 8- Profiles are individual in nature so that each user profile is used to tailor future interactions with only that user;
- 9- Rules are created manually, and reflect the administrator's preferences more than users' preferences.

2.3.2 Content-based systems.

Such systems rely on well-known information retrieval techniques (Pazzani and Billsus 2007). The items are represented by a set of features, or attributes that characterize the item. Meanwhile, each user profile is individual in nature and created from features associated with items in which the user has previously expressed interest (Krulwich and Burkey 1996). A recommendation agent makes the comparison between the features extracted from unrated items with the content description in the user profile. Items that the agent considers as matching the user profile are recommended to that user (Mobasher 2005). Some e-commerce applications represent both user and item features as vectors of weighted attributes (Pazzani

and Billsus 1997), and then compare user and item vectors. Some systems create user profiles only from features of items previously rated by the active user (Micarelli, Sciarrone et al. 2007). The advantages and disadvantages of content-based systems are shown below.

Advantages of content-based filtering systems

- 1- Recommendations are based on the similarity between user profiles and item attributes, not on pre-determined rules.
- 2- Site administrator biases have little effect on the items recommended.
- 3- User profiles are dynamic.
- 4- Profiles contain objective data.

Disadvantages of content-based filtering systems

- 1- Less user participation in his/her profile creation.
- 2- Requires knowledge of document modeling techniques using information retrieval and filtering.
- 3- The system tends to over-personalize the item recommendations since user profiles are usually solely based on the same user's previous ratings of items.
- 4- Sometimes content-based systems suggest the same items several times to the same user.
- 5- The extraction of document features requires nontrivial computational effort and may also give unreliable results.

2.3.3 Collaborative filtering systems.

Such systems rely on web usage mining that use users' click-stream data automatically collected online, or stored on the server (or proxy) log files. Some systems start by cleaning historical log files, then discover patterns and create user profiles (Herlocker, Konstan et al. 1999). Several techniques are used for pattern discovery or classification of users or items (Zhang and Jiao 2007), such as clustering (Burke 2000), association rules (Nakagawa and Mobasher 2003), and sequential patterns (Mobasher, Dai et al. 2002). The recommendation agent in such systems matches the ratings of a current user for objects with those of similar users (nearest neighbors) in order to produce a set of recommended items that the active user

has not visited yet (Mobasher 2007). The advantages and disadvantages of collaborative filtering system are shown below.

Advantages of collaborative filtering systems

- 1- Increase the span of recommendations since user recommendations are based on similar users.
- 2- Suggest unexpected items to users.
- 3- Profiles are dynamic in nature and represent the actual interests of the user.
- 4- There is no site manager control over the suggested items.
- 5- The actual item features are not part of the profile.

Disadvantages of collaborative filtering systems

- 1- Unacceptable recommendation latency with the increase in the number of items and users.
- 2- Similarity computations sometimes become complex.
- 3- Cold start personalization is poor since creating each user profile requires multiple visits by the user.
- 4- No user participation in creating his/her profile.
- 5- Privacy issues.

2.3.4 Hybrid systems.

Because of several drawbacks and deficiencies that are difficult to overcome within the confines of a single recommendation approach, several researchers have tried to mix between these approaches to avoid such obstacles and gain the benefits of both (Burke 2002). The most common form of hybrid recommender combines content-based and collaborative filtering. Nakagawa and Mobasher 2003 proposed a hybrid recommendation system that switched between different recommendation systems based on the degree of connectivity of the site, and implement binary weights on page views within user transactions. They used association rule discovery and sequential pattern discovery; in association rule discovery, they ignored the sequence of page views and identify users by their IP address. They used a fixed size sliding window for creating a recommendation set to the current user, allowing only the last n visited pages to influence the recommendations. A main problem in this

system is the lower coverage due to the large number of states that the system should manipulate to generate recommendations. In addition, the system was unable to generate recommendations for new users or even for the new added items, and ignored all but the last n visited pages of the current session. Gunawardana and Meek 2009 described unified Boltzmann machines, which are probabilistic models that combine collaborative and content information in a coherent manner in order to utilize benefits from content-based and collaborative-based approaches to provide more accurate recommendation.

2.4 Collaborative filtering systems: associated techniques.

Different techniques are used in different collaborative systems (memory-based, model-based, and hybrid collaborative filtering systems). Collaborative filtering (*CF*) systems use the collected preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. In this section we attempt to present a comprehensive survey of CF techniques.

2.4.1 Collaborative filtering techniques for memory-based systems

Memory-based CF algorithms depend on the collected user-item preferences stored in the database to identify the neighbourhood of the active user and then to provide recommendations. Memory-based CF algorithms use the following steps: calculate the similarity w_{ij} between users i and j , then predict a set of items that represent a bag of recommendations, and then find the top- N recommendation using the k most similar users or items (Sarwar, Karypis et al. 2001).

A. Similarity calculation.

Item-based similarity calculates similarity between item i and item j depending on the users' ratings for both items, and hence compute similarity between the two items w_{ij} based on the two co-rated values. In contrast user-based similarity calculates the similarity $w_{u,v}$, between users u and v who have both rated the same items. Different similarity metrics are used, and the following section provide more details about some of these methods.

Correlation-based similarity calculation

Pearson correlation is used to find the similarity between two users $w_{u,v}$ or between two items $w_{i,j}$, which simply measures the strength of the correlation between user's ratings of two items, or between two user's ratings of a set of items (Melville, Mooney et al. 2002). Equation 2.1 shows the Pearson correlation between two users.

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (2.1)$$

Where the $i \in I$ sums of items that both the users u and v have rated, and \bar{r}_u is the average rating made by user u for this set of items and \bar{r}_v is the average rating made by user v .

For item-based similarity, equation 2.2 is used.

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad (2.2)$$

Where U is the set of users who have rated both items i and j , and $r_{u,i}$ is the rating of user u on item i , and \bar{r}_i is the average rating of item i by those users.

Many other correlation-based similarities computations are used in different systems such as: *constrained Pearson correlation*, which uses median instead of mean rates, *Spearman rank correlation*, which use ranks instead of absolute ratings, and *Kendall's τ correlation*, which uses relative ranks to calculate the correlation (Herlocker, Konstan et al. 2004).

Vector-cosine based similarity.

This can be used to find similarity between two items on the basis of vectors of word frequencies form in text descriptions of the items (Salton and McGill 1983). In this context, cosine angle can be used in collaborative filtering systems by treating user or item attribute vectors as document frequency vectors.

Formally, if R is the $m \times n$ user-item matrix, then the similarity between two items, i and j , is defined as the cosine of the n dimensional vectors corresponding to the i^{th} and j^{th} column of the matrix R . *Vector cosine similarity* between items i and j is calculated by equation 2.3.

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| * \|\vec{j}\|} \quad (2.3)$$

where “ \cdot ” denotes the dot-product of the two vectors. In order to compute similarity for n items, then an $n \times n$ similarity matrix is computed (Sarwar, Karypis et al. 2000).

For example, if the vector $\vec{A} = \{x_1, y_1\}$ and vector $\vec{B} = \{x_2, y_2\}$, the vector cosine similarity between \vec{A} and \vec{B} is computed as in equation 2.4.

$$w_{A,b} = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} = \frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}} \quad (2.4)$$

Many other similarity measurements are used such as: *conditional probability-based similarity*, *adjusted cosine similarity* (Deshpande and Karypis 2004) .

B. From similarities to recommendations

Recommendation systems that calculate similarities between users or items use a subset of nearest neighbors (of the active user), and then calculate weighted aggregate ratings to generate predictions for the active user (Herlocker, Konstan et al. 1999).

Weighted sum of others' ratings.

In order to find a predicted rating for the active user a , for a certain item i , the weighted average of all item ratings are sometimes used as shown by equation 2.5.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|} \quad (2.5)$$

Where $\sum_{u \in U}$ sums over all users $u \in U$ who have rated item i , and \bar{r}_a and \bar{r}_u are the average ratings for the user a and user u over all other rated items, and $w_{a,u}$ is the weighted distance between the user a and user u .

Simple weighted average.

Item-based recommendation systems can use the *simple weighted average* to predict the rating, $P_{u,i}$, for user u of item i (Sarwar, Karypis et al. 2001), as shown by equation 2.6,

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|} \quad (2.6)$$

where $\sum_{n \in N}$ sums over all rated items $n \in N$ for user u , $r_{u,n}$ is the rating for user u of item n and $w_{i,n}$ is the weight between items i and n .

C. Computing the top- N items.

Top- N recommendation aims to recommend a set of N top-ranked items that are expected to be of most interest to the active user. A top- N recommendation technique analyzes the user-item matrix to discover relations between different users or items and use them to compute the best recommendations.

Top- N recommendation (user based).

Sarwar, Karypis et al. 2000 used the *Pearson correlation* to find the k most similar users to the active user, then they use the user-item matrix R to identify a set of items C that are of most interest to the k neighbours. The recommendation system recommends the top- N most frequent items in set C to the active user. Although top- N recommendation is able to provide appropriate recommendations, but it has some deficiencies related to scalability and real-time performance (Jamali and Ester 2009).

Top-N recommendation (item based).

Jamali and Ester 2009 tried to solve the scalability problem in user-based top- N recommendation systems by computing the k most similar items for each item. Moreover, they identify the set C as candidates for recommended items by taking the union of the k most similar items and removing items already visited by the active user to get a subset U . Then, they calculate the similarities between each item of the set C and the subset U . The resulting set of items in C , only the top- N items are provided as a recommendation.

Several other memory-based techniques are being used for recommendation purposes, such as *Default Voting*, which calculates pairwise similarity from the ratings that both users have rated (Sarwar, Karypis et al. 2000). However, this provides recommendations based on users' ratings, but will not work with too few votes; also, it focuses on the intersection similarity set, which neglects much of the user's rating history. Breese, Heckerman et al. 1998 used 'negative preference' for the unobserved ratings and then computes the similarity between users on the resulting ratings data. Chee, Han et al. 2001 used the average vote of a small group as a default vote to extend each user's rating history. Herlocker, Konstan et al. 1999 found a small intersection sets by reducing the weight of users that have fewer than 50 items in common.

2.4.2 Collaborative filtering techniques for model-based systems

Model based collaborative filtering systems depend on a two-stage process for generating recommendations. *The first stage* is offline, where online users' behaviors (e.g. from historical log files) are mined in order to discover user patterns. *The second stage* is online or real time, where a recommendation set is created based on the active user's profile. Several techniques used by collaborative systems for creating users' profiles, discovering users' patterns, and making recommendations are given in (Mobasher 2007).

A. Clustering-based collaborative filtering.

Clustering aims to divide a data set into groups where inter-cluster similarities are minimized while the similarities within each cluster are maximized. Generally, clustering methods can be divided into three different categories (Han and Kamber 2006):

- Partition clustering creates k partitions of a given data set, and each partition represents a cluster; the k -means algorithm is a common partitioning method.
- Hierarchical clustering builds a tree-based clustering. In a top-down approach, it starts from the whole data set of items as a single cluster and recursively partitions this data set. In a bottom-up approach, hierarchical clustering will start from individual items as clusters and iteratively combine smaller clusters into larger clusters.
- Model-based clustering uses a mathematical model to discover the best fit between data points, and usually it specified as a probability distribution.

Different collaborative system use different clustering methods which sometimes cluster items based on interest scores, or cluster users based on the characteristics of their behaviour. In **item-based clustering**, items are clustered based on the similarity of ratings from all users for these items (O'Connor and Herlocker 2001). Each item-based cluster center is represented by an M -dimensional vector $C_i^{(K)} = (q_1, q_2, q_3, \dots, q_m)$, where each q_k is the average ratings by user U_k of items within the cluster. In **user-based clustering**, users are clustered based on the similarity of their ratings of items; each cluster center $C_j^{(U)}$ is represented by an n -dimensional vector, $C_j^{(U)} = (R_1, R_2, R_3, \dots, R_n)$, where R_i is the average item rating for item T_i by users in cluster k (Borges and Levene 2004). Several factors are used to determine each item's weight within profiles, such as the path or link distance from pages to the current user location within the site, or also the rank based on whether the item is significant (or not) to the user. The recommendation system calculates the similarity of an active user's profile with other users' profiles to discover the top N matches are then used to produce a recommendation set (Sarwar, Karypis et al. 2002).

Generally, user-based clustering group users based on the similarity of their profiles in a matrix UP , while item-based clustering makes a clustering based on the similarities of the interest scores for these items across all users, or based on similarity of their attributes or their content features. Ungar and Foster 1998 used k -means for item and user-based clustering. O'Connor and Herlocker 2001 used agglomerative hierarchical clustering for item-based clustering as a means of reducing the dimensionality of the rating matrix. In this context, they use Pearson's correlation coefficient to calculate the similarity of column

vectors from the items ratings matrix, and then create smaller ratings matrices that are used for predictions. Kohrs and Merialdo 1999 used hierarchical clustering for user-based and item-based clustering. Borges and Levene 2004 used mixtures resolving algorithms to cluster users based on their item ratings.

A typical user-based clustering starts with the matrix UP of user profiles and then partitions UP into k groups of profiles where each group's members are similar to each other and different from other groups' members. This partitioning process can be based on common navigational behavior or interest shown in various items. The resulting user segmentation is used to find neighborhoods of the active user as well as to find recommendations for the active user (Mobasher, Dai et al. 2002). In order to determine similarity between a target user and a user segment, the centroid vector of each cluster is computed and used as the aggregate representation of the user segment. Each cluster C_k has centroid vector v_k which is computed as: $v_k = \frac{1}{|C_k|} \sum u_n$, where u_n is the vector in UP for a user profile $u_n \in C_k$. Hence to create appropriate recommendations for an active user u and target item i , they need to find the most similar neighborhood (with a profile v_k) of the active user, and then a prediction score can be computed for item i and user u as in equation 2.7.

$$P_{u,i} = \bar{s}_u + \frac{\sum_{v \in V} sim(u,v)(s_v(i) - \bar{s}_v)}{\sum_{v \in V} |sim(u,v)|} \quad (2.7)$$

where $P_{u,i}$ refers to the prediction score for user u and item i . V is the set of k most similar segments, $s_v(i)$ is the weight of i in the neighbor segment v , \bar{s}_u and \bar{s}_v are the average interest scores over all items for user u and segment v , and $sim(u, v)$ is the similarity between user u and segment v .

Perkowitz and Etzioni 2000 used an algorithm called *PageGather* to discover significant groups of pages based on user access patterns, they used a complete link to cluster pages based on users clicks, they represent pages as nodes and then edges between two nodes are added if the corresponding pages occur in more than a certain number of sessions. Hence, all connected components within the graph grouped into one cluster. Each cluster's nodes are recommended in a new index page using a hyperlink to each cluster item. Nasraoui, Krishnapuram et al. 2002 used fuzzy clustering approach where any item may be considered

as belonging to more than one cluster at the same time. Some clustering methods do not consider the sequential order of visited items, but other clustering algorithms take this into account. E.g. (Strehl 2002) used graph-based algorithm to cluster web subsequences transactions.

B. Association rule based collaborative filtering.

Association rules serve as a useful tool for discovering correlations among items in a large database. They explore the probability that when certain items are visited in a session, certain other items will also be visited in the same session (Sandvig, Mobasher et al. 2007). An association rule is typically of the form $X \rightarrow Y$, where X and Y are two disjoint sets of items. An interpretation of the association rule in business trading situation is that when a customer buys items in X , the customer will also buy items in Y . Two important functions are used for mining association rules, the *support* function and the *confidence* function.

Support indicates the frequencies of the patterns occurring in the rule. This algorithm finds groups of items that occur together in many transactions (e.g. sessions). These groups of items are referred to as a frequent item sets. Given a transaction database T (i.e. a record of many sessions, each session t being a set of items visited) and a set of items I_i , the support of I_i is defined as in equation 2.8.

$$\sigma(I_i) = \frac{|\{t \in T : I_i \subseteq t\}|}{|T|} \quad (2.8)$$

In association rule building algorithms, a minimum level of support is needed to guide the generation of new rules at each iteration (Mobasher and Burke, 2008). As well as needing to find rules with a certain level of support (which means they will be useful often, instead of rarely used), association rules also need to have a suitable level of confidence.

Confidence refers to the accuracy of the implication of the association rule. If the confidence is high, then the rule is more reliable. An association rule r is an expression of the form $X \Rightarrow Y (\sigma_r, \alpha_r)$, where X and Y are item sets. The *confidence* for the rule r , σ_r , is given by

$$\sigma(X \cup Y) / \sigma(X) \quad (2.9)$$

This represents the conditional probability that Y occurs in a transaction given that X has occurred in that transaction.

While it's possible to restate the *support* for the rule r , σ_r , as in equation 2.10.

$$\sigma_r = \sigma(X \cup Y) \quad (2.10)$$

This represents the probability that X and Y occur together in a transaction (Mobasher and Burke, 2008). In the classic a priori algorithm and most algorithms that derive from it, a minimum support S and minimum confidence C must be satisfied, as the algorithm proceeds to find larger and more interesting rules.

Sarwar, Karypis et al. 2000 used association rules in an e-commerce recommendation system, where the preferences of the user were matched against the items in the antecedent X of each rule, and all stored matching rules with sufficient confidence were used to recommend N items to the active user. Although association rules help to find appropriate recommendations, this does not work well when the dataset is sparse. Fu, Budzik et al. 2000 tried to solve this problem in two different ways. *Their first solution* is to rank all matching rules calculated by the degree of intersection between the antecedent rule and the items in the user's active session, and then to generate the top k recommendations. *Their second solution* is to find "close neighbors" who have similar interests to a target user and make recommendations based on the close neighbor's history.

Recommendation agents generate association rules (among both users and items) for each user, and then if support is greater than a pre-specified threshold, then the system generates recommendations based on user association, else it uses item associations. Association-based algorithms use a sliding window w that is decreased iteratively until a match with the antecedent of a rule is found. The main problem here is that the sliding window does not reflect the sequential sequences of selected item by specific user since it lose its earlier items with the increase of its length, as well being time consuming since it requires repeated search through the rule-base. Alternatively, to association rules, some systems use data structures (such as directed acyclic graphs) to store discovered item sets in order to generate more efficient recommendations in less time than generating association rules.

Aggarwal et al. 2001 created a directed acyclic graph of frequent item sets, which uses different levels reflecting the depth of each item in the graph starting from 0 to k , where k is the maximum size among all frequent item sets. Each node at depth d in the graph corresponds to an item set I of size d and is back-linked to item sets of size $d-1$ that contain I at level $d-1$, and forward-linked to item sets of size $d+1$ that contain I at level $d+1$. All item sets are sorted in lexicographical order before being inserted into the graph, and the user's active session is also sorted in the same manner to be able to match different orderings of an active session with frequent item sets. In order to find candidate items for recommendation, matches between the active user session window, w , with all previously discovered frequent item sets of size $|w| + 1$ containing the current session window by performing a depth-first search of the frequent item set graph to the level $|w|$. Confidence values of the corresponding association rules are calculated, and if a match is found, the child (singleton) of the matched items in w are used to generate candidate recommendations.

C. Sequential rule collaborative filtering

Sequential patterns are important in collaborative filtering and refer to common patterns found in the order in which users visit a set of items and/or pages (Eirinaki and Vazirgiannis 2003). The discovery of sequential patterns allows us to predict the next pages that might be accessed by the active user based on the previously accessed pages (Zhou, Hui et al. 2004).

Sequential patterns can represent non-contiguous frequent sequences in the underlying set of transactions or sessions. In contagious sequential patterns, each pair of adjacent elements must appear consecutively in a transaction t , which supports the pattern. Given a transaction set T (e.g. a set of user sessions) and a set $S = \{S_1, S_2, \dots, S_n\}$ of frequent sequential (respectively, contiguous sequential) patterns.

The *support* of each pattern S_i is defined as in equation 2.11.

$$\sigma(s_i) = \frac{|\{t \in T : s_i \text{ is (contiguous) subsequence of } t\}|}{|T|} \quad (2.11)$$

The *confidence* of the rule $X \Rightarrow Y$, where X and Y are (contiguous) sequential patterns defined as,

$$\alpha(X \Rightarrow Y) = \frac{\sigma(X \circ Y)}{\sigma(X)} \quad (2.12)$$

Where \circ denotes the concatenation operator.

Schechter, Krishnan et al. 1998 created contiguous sequential patterns by capturing frequent navigational paths that reflect users' behaviors stored in log files. As we mentioned before, the sequential patterns reflect ordering of visited pages or selected items, while association rule mining focus on the presence of items within a user session rather than the order in which they occur. Spiliopoulou and Faulstich 1998 represented contiguous navigational sequences in a tree structure and created an *aggregate tree*. In their context, they extract transactions from a collection of web logs and transform them into sequences to create the tree that is used later for generating recommendations. Sequential patterns are typically stored in a single tree structure where nodes represent items and the root represents the empty sequence. Mobasher, Dai et al. 2002 used a fixed size-sliding window w over the current transaction for recommendation generation, requiring a tree to be generated with maximum depth only $|w| + 1$. The length of the created sequential tree can be controlled through support and confidence thresholds, but the site characteristics such as site topology and degree of connectivity have a significant impact on the usefulness of sequential patterns over non-sequential (association) patterns (Nakagawa and Mobasher 2003). Additionally, collaborative systems that depend on contiguous sequential patterns are more valuable in page pre-fetching applications where it is the intent to predict the immediate next page to be accessed rather than generating candidates for recommendations (Mobasher, Dai et al. 2002).

Sarukkai 2000 designed a system to predict the *next* user action based on a user's previous surfing behavior; a probabilistic model was used to predict subsequent visits using the sequences of page-views in the user's session. This approach models a user's navigational activity as a Markov chain, represented as a 3-tuple $\langle A, S, T \rangle$ where A is a set of all possible actions, S is the set of states, and T is the transition probability matrix that stores the probability that a user will perform an action $a \in A$ when the process is in a state $s \in S$. The probability of a transition from state s_i to state s_j is denoted by $T = [p_{i,j}]_{n \times n}$, and the order of the Markov model corresponds to the number of prior events used in predicting a future event. Therefore, given a set of paths R , the probability of reaching a state s_j from a state s_i

via a (non-cyclic) path $r \in R$ is given by: $p(r) = \sum P_{k,k+1}$, where k ranges from i to $j-1$. The probability of reaching s_j from s_i is the sum over all paths: $P(j|i) = \sum_{r \in R} P(r)$.

Borges and Levene 1999 used a Markov model to discover high-probability user navigational paths in a Web site. Deshpande and Karypis 2004 used selective Markov models that only store some of the states within the model and consider it as a solution to the coverage problem (the difficulty of representing correct transition probabilities when the number of states is high); they used pruning algorithm to prune out states that cannot be expect to be accurate predictors. Three parameters were used for the pruning process: support, confidence, and estimated error.

Although contiguous sequential pattern mining can provide higher prediction accuracy, but many problems arise when using this technique such as lower coverage, and high complexity due to the large number of states.

2.4.3 Graph theoretic collaborative filtering

Mirza 2001 presented a graph-theoretic model that casts recommendation as a process of ‘jumping connections’ in a graph. Moreover, he presented an algorithmic framework drawn from random graph theory and outlines an analysis for one particular form of jump called a ‘hammock’; he used two datasets collected over the internet to demonstrate the validity of his approach. Huang, Chung et al. 2002 created a graph-based recommender system for a digital library that naturally combines the content-based and collaborative approaches; they find high-degree book-book, user-user, and book-user associations. The system was tested and they found that the system gained improvement with respect to both precision and recall by combining content-based and collaborative approaches.

A graph-theoretic approach for collaborative filtering was used to build a directed graph with vertices representing users and edges denoting the degree of similarity between them by (Mirza, Keller et al. 2003). In order to predict user u ’s rating of item i , we need to find a directed path from user u to a user who has rated item i . In other words, a path should exist from user u_i to u_j if user u_j can be used to find predictions for user u_i . In order to predict if a particular item i_k will be of interest to user u_i , (Mirza 2001) system calculates the shortest

path from user u_i to any user u_j who has rated item i_k , and the predicted rating for the item i_k by user u_i generated as a mapping function from user u_i to u_j .

2.4.4 Hybrid collaborative filtering systems

Different techniques are being used for recommendation, but each one has its own limitations. Some researchers see that creating hybrid collaborative filtering systems helps not only to reduce these limitations (found in individual techniques), but also to utilize the benefits gained from these separate techniques. The most common form of hybrid systems are combinations of collaborative and content based models; some other hybrid systems include demographical data along with collaborative filters, while some other systems combine semantic knowledge with usage data for recommendation. In this section we will discuss some of these hybrid systems.

Integration between content-based features and usage data

Hybrid systems that depend on such integration generate recommendations not only based on similar users, but also based on the content similarity of these pages to the pages which user has already visited. Users' profiles are represented as concept vectors that reflect their interests in particular concepts or topics. Therefore, these systems usually create a content-enhanced profile, containing the semantic features of the underlying items as well as mapping each item or page in a user profile to one or more content features extracted from the items (Mobasher 2007).

Ansari, Essegaier et al. 2000 proposed a Bayesian preference model that statistically integrates user preferences, user and item features, and expert evaluations. In addition, they used sampling parameter estimation from the full conditional distribution of parameters and they achieved better performance than pure collaborative filtering. Eirinaki, Vazirgiannis et al. 2003 used content features extracted from web pages to enhance usage data. Information retrieval techniques were used to extract pages features, and then the features were mapped to a predefined concept hierarchy. The users' navigational behaviors were represented in the form of clusters or association rules, which were then used as the recommendation basis for each user or group of users, resulting in a broader semantic set of recommendations.

Haase, Ehrig et al. 2004 created semantic user profiles from usage and content information to provide personalized access to bibliographic information on a Peer-to-Peer bibliographic network. The user's semantic profile is created from the expertise (such as website developers), recent queries, recent relevant instances and a set of weights for the similarity function. Ghani and Fano 2002 created a recommender system based on a custom-built knowledge base using product semantics, and they extracted attributes from the online marketing text, describing the products browsed. Girolami and Kabán 2003 created a probabilistic model based on the content information of each user's items of interest, and then the system makes predictions for unvisited or unrated items based on the content information of these items. The individual models were combined under a hierarchical Bayesian framework.

Popescul, Ungar et al. 2001 used a mixture model of hidden variables to handle three-way co-occurrence data including users, items, and content features. The proposed model was used to discover the hidden relationships among users, items and attributes, but several limitations were found in this approach since the three-way observation data is very sparse, and needs to be generated subjectively from other observation data.

Integration between structured semantic knowledge and usage data

Although the combination of content and usage data improves the performance of recommendation systems, keyword-based approaches cannot capture more complex semantic relationships among objects and properties associated with these objects. For example, potentially valuable relational structures among objects such as relationships between students, courses, and instructors, may be missed if we only rely on the description of these entities using sets of keywords. In order to recommend different types of complex objects using their underlying properties and attributes, the system must be able to rely on a characterization of user segments and objects, not just based on keywords, but at a deeper semantic level using the domain ontologies for the objects.

Middleton, Shadbolt et al. 2004 created an ontological profile for each user that relies on a topic hierarchy; they used available ontologies based on personnel records and user publications. Kearney, Anand et al. 2005 combined web usage data with semantic knowledge in order to get a deeper understanding of users' behaviors, therefore they capture the impact

of provided domain knowledge on the user's behavior and then create an ontological profile for each user. A mapping between each page (within user sessions) to the proper concepts in the ontology is performed, and then specific instances are generalized to an Ontological Profile (OP). Hence, vectors of pages over a set of concepts are built, where each dimension measures the degree to which the page belongs to the corresponding concept.

Integration between link structure and usage data

Some web personalization systems rely on the hyperlink structure of the web site to provide recommendations. Nakagawa and Mobasher 2003 created a hybrid recommendation system that switched between different recommendation algorithms based on the degree of connectivity in the site and the current location of the user within the site. They found that in a highly connected web site with short navigational paths, non-sequential models perform well by achieving higher overall precision and recall than sequential pattern models. They used a logistic regression function as a switching criterion to select the best recommendation model for the target user. The similarity function compares sessions containing pages that are different but structurally related. Li and Zaïane 2004 found navigational patterns of users using a user's access history and the content of visited pages, as well as the connectivity between the pages on a web site. The users' visits are called "missions", where a mission is a sub-session with a consistent goal, determined based on the content similarity of the pages within the session. In order to generate navigational patterns, users' missions are clustered and enhanced with their linked neighborhood, and then when a visitor starts a new session, the session is matched with these clusters to generate a recommendation list.

2.4.5 Summary

Several techniques are used for web personalization starting from those depend on rules which pre-specified by the site administrator (rule-based approach – usually associated with marketing campaigns, where a specific contents are conveyed to the user or a set of users based on specific rules). Some systems depend on filtering the content of visited pages to determine the users' interests, and then based on the created profile for each user; a recommendation agent creates a set of recommendations for that user. Collaborative filtering systems try to utilize the benefits of profiles of many users. The profile of each user is useful

not only for that user but also for others in the neighbourhood, which will be used by the recommendation agent to create a set of suggestions for that user.

In this section, we demonstrated different techniques used by collaborative-based systems; in next section, we demonstrate a mixture of previous personalization and recommendation systems.

2.5 Previous personalization and recommendation systems

Several systems use the content-based filtering approach for personalization. Pazzani 1999 developed a system which classifies web pages based on specific features, and then asks users to rate their interests based on these features. A user profile is created from previously ranked features on a particular topic to distinguish between interesting and non-interesting features for each user. They classify web pages using a naïve Bayes classifier to predict future pages as potentially interesting to the user. Users provide an initial profile to determine which pages are interesting and which are not, and the initial profiles are updated gradually based on users' visits. The main advantages of this system are the simplicity and the user's participation in creating his/her profile. The system depends only on item selection and is purely based on the user's previous ratings of items stored in their user profile, but it does not take into account changes in the user's interests. Schwab, Kobsa et al. 2000 created user profiles from implicit observations, using naïve Bayes, and create a technique for selecting features for a specific user based on the deviation of feature values from the norm. There is less user participation in recommendations, and recommendation are solely based on the user's previous rating; the main disadvantage of the system is that the required time for capturing the features is too high. Generally, users are pleased with personalization if the recommendation agent provides useful but unexpected items to them, but most content-based systems recommend items that have been previously recommended to the users due to their static profiles and the extracted features from web pages (Schwab, Kobsa et al. 2000).

Collaborative filtering systems assume the users with common interests in the past (known as consumed items feedback) will have similar tastes in the future, so they try to find other likeminded users and create a recommendation set of items consumed (or visited) by those likeminded users but not consumed (or visited) by the current (active) user. Herlocker,

Konstan et al. 1999 proposed the use of a significance weighting to measure how dependable the measure of similarity between two users. Herlocker et al. found that two users are considered equally similar regardless of whether they had two rated items or fifty items, so that neighbours based on small samples produced a bad prediction of the active user interests. They proposed the use of variance weighting to consider the variability of items' values within the session; a low weight means that most users have a similar rating for the item and so it is more difficult to discriminate between users. To solve this problem Herlocker et al. used a scale of ratings. Breese, Heckerman et al. 1998 proposed the use of inverse user frequency where items less frequently rated are given a lower weight.

Sarwar et al., 2001 built an item-based system by creating an item similarity matrix $IS[j, i]$ that shows the similarity between items i_j and items i_i . Such similarity is not based on the items' features (as in content-based filtering systems) but based on users' ratings of the items. The recommendation process predicts the rating for items not previously rated by the user, but by computing a weighted sum of the ratings of items in the item neighbourhood of the target item, consisting of only those items previously rated by the user (Sarwar, Karypis et al. 2001). Several systems used a clustering approach; some of these are item-based clustering and the others are user-based clustering or a combination of the two. Kohrs and Merialdo 1999 used top down hierarchical clustering for users and items; two cluster hierarchies were captured, one of these was based on item ratings by the user and the other is based the user ratings of items. The predicted rating of an item for the active user was generated using a weighted average of cluster centre coordinates for all clusters from the root cluster to the appropriate leaf node of each of the two hierarchies. The weights were based on the intra-cluster similarity of each of the cluster.

Newman, Asuncion et al. 2007 created a Google news recommender system, which combines three different algorithms: collaborative filtering using MinHash clustering, Probabilistic Latent Semantic Indexing (PLSI), and co-visitation counts. Although the system provides news recommendations, it does not solve the cold-start problem for new users. Even though ratings from new users can be updated in near real-time by their algorithm, it still needs to wait until new users provide ratings or clicks before making recommendations.

Gabrilovich, Dumais et al. 2004 provided personalized news feeds for users by measuring news novelty in the context of stories the users have already read. Micarelli, Gasparetti et al.

2007 built personalization models for short-term and long-term user needs based on user actions instead of traditional information retrieval (IR) techniques. Speretta and Gauch 2005 created users' profiles from their query histories and used these profiles to re-rank the results returned by an independent search engine by giving more importance to the documents related to topics contained in the user profile. The TaskSieve system designed by (Ahn, Brusilovsky et al. 2007) to utilize benefits of collected feedback to create a feedback-based profile for personalized search. A personalized service may not only be based on the active user's behaviours, it can benefit from similar users' behaviors, as well as from the homogeneous groups of consumers by using *a priori* segmentation. Krulwich 1997 and Pazzani 1999 group consumers on the basis of demographic and socioeconomic variables, and statistical models are estimated within each of those groups, and recommendations are based on demographic classes inferred from users' personal attributes.

2.6 Conclusion:

As noted before, recommendation systems can be mostly categorised into rule-based systems, content-based filtering, and collaborative filtering systems. The collaborative filtering systems are the most commonly used models for personalization purposes. But, although traditional collaborative filtering systems generate successful recommendations, they suffer from several problems such as the ***cold start problem*** where a user should visit web site several times before the system is being able to discover his/her preferences. There is also the ***latency problem***, where recommendations to the current active user may take too much time due to system load and the number of processes required for generating a recommendation set. There is also the ***privacy problem***, and the ***scalability problem***, and other challenges that we will explore later in this chapter.

2.7 Recent trends in web usage personalization

Most currently, web personalization systems are collaborative systems or hybrid systems that combine content-based and usage-based systems. Some of the most recent systems use a reactive approach (David, Carstea et al. 2010). These systems deal with personalization as a conversational process that requires explicit interactions with the user in the form of queries or feedback. A list of recommendations is provided to user, and then he should choose one of the recommendations that best suit his requirements, thereby refining his interests to help the

recommendation process. Other systems use a proactive approach (Chao, Yang et al. 2011), where the system learns user preferences and provides recommendations based on the learned information. These systems provide the user with recommendations that the user may choose to select or ignore. In this case, it is not necessary for the user to provide explicit or implicit feedback to the system for the recommendation process, and feedback is not central to the recommendation process.

Talabeigi, et al. 2010 tried to find a solution to the problem of information overload on the Internet. They created a dynamic Web page recommender system based on asynchronous cellular learning automata (ACLA) which continuously interacts with the users and learns from his behavior. They need to update periodically extracted pattern and rules in order to make sure they still reflect the trends of users or the changes of the site structure or content. However, their system did not overcome the privacy problem, and they did not provide a proper explanation about the system performance, as well as it is not clear the precision level per period since the update of users' patterns done periodically.

Erkin, Beye et al. 2010 encrypted the privacy sensitive data ; in order to solve the privacy problem, and generate recommendations by processing them under encryption. With this approach, the service provider learns no information on any user's preferences or the recommendations made. The proposed method is based on homomorphic encryption schemes and secure multiparty computation (MPC) techniques, but the level of accuracy of provided recommendation is not measured in order to prove the effectiveness of the proposed system. Zhan, Hsieh et al. 2010 provide a model for protecting privacy in collaborative recommender systems designed to hide individual user records from the system itself, but they do not tackle the risk that individual actions can be inferred from temporal changes in the system's recommendations and do not appear to provide high protection against such threat. Kaleli, and Polat 2010 propose a naïve Bayesian classifier based collaborative filtering (CF) over a P2P topology where the users protect the privacy of their data using masking, which is analogous to randomization. However, they did not indicate how these masks are regenerated as well as how exactly user identified to the system in the future visits. In privacy-preserving, data (Han, Ng et al. 2009) produced a host of secure computation protocols such as singular values decomposition, but this decomposition process consumes time and make the model more complex.

Park and Chu 2009 they used filterbots method as a model to represent relationships between a user's demographic information and an item's metadata. The set of filterbots that was used by the system was also fine-tuned, and was injected into user-user (or item-item) matrix to find similarity between users (or items), and then generate recommendations. Nevertheless, this model cannot be used with the new system cold start when the system is new and there are no ratings from any user for any item. As well as the system used filterbots and this did not reflect the actual users' preferences and only reflect average ratings of demographic groups. Zhang, Chuang et al. 2010 used the user-tag-object tripartite graphs, they suggest a recommendation algorithm that use social tags. Although the suggested model provides more personalized recommendation when the assigned tags belong to more diverse topics, but the suggested algorithm is particularly effective for small-degree objects, Therefore they don't consider the growth of the system since introducing new users or items may involve the cold start problem for them.

Wei and Park 2009 proposed a system for recommending an item for a user. Therefore, the suggested system constructs one or more user profiles, where each user profile is represented by a user feature set. In addition, it constructs one or more item profiles, where each item profile is represented by a set of item feature. The system receives historical item ratings given by one or more users, and then the system generates one or more preference scores by modeling at least one relationship among the user profiles, the item profiles and the historical item ratings. Nevertheless, the system cannot generate recommendation in case of system cold start as well as the privacy problem is still valid in the model. Personal recommendation systems strive to adapt their services (adv, news, movies, items, etc.) to personal users by using both content and user information. The cold start problem stills a challenging because the provided web service is featured with dynamically changing pools of contents, rendering traditional collaborative filtering methods inapplicable. As well as the scale of most web services of practical interest calls for solutions that is fast in learning and computation. Lihong, Wei et al. 2010 modeled personalized recommendation of news articles as a contextual bandit problem. Their approach used a learning algorithm sequentially selects articles to serve users based on contextual information about users and articles, while simultaneously adapting its article-selection strategy based on user-click feedback to maximize total user clicks, but still the privacy problem valid in the system, as well as the system will not generate recommendations until it receive users feedbacks.

2.8 Current web personalization challenges

Web recommendation systems providing fast and accurate recommendations will attract customers as well as achieve benefits to companies, but these systems face many challenges, which we will discuss in this section.

2.8.1 The cold start problem

Several challenges direct web personalization research; one of these challenges is known as the first visit or the cold start and latency problem (Schein, Popescul et al. 2002). A web personalization system should have some information available about a new visitor, to present items of interest to the new user and promote his future interaction. Hence, a new user with no interaction history with a site will not receive any suggestions or recommendations, i.e. the system is unable to personalize its interactions with this new user. Therefore, the lack of useful information about the visitor puts him/her off the system until the system is able to collect the required data to start generating appropriate and interesting recommendations to the new visitor. A similar problem arises when a new item is added to the web site; systems that depend only on item ratings cannot recommend the new item before a considerable amount of history with that item has been collected. This problem is known as the new Item or new item latency problem. A collaborative filtering system provides no value to the first user who rates the new item. Drachslar, Hummel et al. 2007 tried to solve this problem using demographic profiles by collecting data explicitly, so that the new user was classified demographically and he/she would receive recommendations similar to others in the same demographical group. Schein, Popescul et al. 2002 tried to solve the cold start problem by creating a profile for each new user and initialized it using the proprieties of a peer-to-peer network using profiles of similar peers in the semantic neighborhood to initialize the profile of a new peer; this problem is discussed in more details in the next chapter. Seung-Taek et al., 2009 used predictive feature-based regression models that leverage all available information about users and items, such as user demographic information and item content features, to tackle the cold-start problem.

As we explored before, researchers divide this problem into the user cold-start problem and the item cold-start problem. Many researchers tried to solve the cold start problem using different methods as explore in the following section.

A) Demographic based recommendation.

Some researchers use demographical data to find initial similarities between site visitors. Demographic data refers to specific user characteristics such as age, gender, income, religion, marital status, language, ownership (home, car, etc), and social position, etc. Demographic data can be used as initial characteristics for creating recommendations and solve the user cold start problem, i.e. providing recommendations when the system does not yet have any information about the user ratings. This is illustrated by figure 2.4, taken from (Drachsler et al., 2007).



Figure 2.4: Demographic filtering (Drachsler et al., 2007).

The red user is new to the system, and demographically matches the user who likes item A; therefore, the system will recommend item A to this new user. Although such data can be useful for creating initial recommendations, demographic profiling creates generalizations about groups of people. Many individuals within these groups will not conform to these generalized profiles; demographic information is aggregated and probabilistic information about groups, not about specific individuals. Also, users are required to fill in a form or in some other way provide their demographic information, which causes annoyance for users and also does not take privacy into consideration. Furthermore, these profiles will be static manner and need to be updated frequently, which becomes boring for users (Nguyen et al., 2007). Lam et al., 2008 used a ‘User-Info Aspect’ model (also called triadic aspect model) that depends on users’ demographic information such as age, gender, and job. Although this model provides a solution to the user cold start problem, but it did not provide a solution to the item cold start problem; also demographical data does not reflect the actual preferences of users. We will provide more description about this method in chapter five.

B) Stereotype recommendation.

A stereotype is defined as a simplified and/or standardized conception or image with specific meaning, often held in common by people about another group (Sollenborn and Funk, 2002). A stereotype may be a conventional and oversimplified conception, opinion, or image based on the assumption that there are common attributes held by members of a specific group. It may be a positive or negative, also it is typically formed by limited knowledge about the group, or false association between two variables. For example, the English people are stereotyped as inordinately proper, prudish, and stiff, while stereotypes about the Arabs and Muslims present in Western societies and American media, literature, theatre and other creative expressions, present them as billionaires, bombers, and shepherds. Such stereotypes are mostly incorrect.

Some recommendation systems use such stereotypes for creating initial recommendations for the new users (Shani et al., 2007). The Naïve Filterbots algorithm proposed by (Park et al., 2006) is used to inject ‘pseudo users’ or bots into the system; these bots rate items algorithmically according to attributes of items or users, for example according to who acted in a movie, or according to the average of some demographic of users. Ratings generated by the bots are injected into the user-item matrix along with actual user ratings, and then standard collaborative filtering algorithms are applied to generate recommendations. Although it is useful for creating an initial recommendation, this approach may refuse to recognize a distinction between an individual and the group to which he or she belongs. At the same time, to classify a person with specific group of people, it should collect data about his/her ethnics or his/her personal data. Therefore, such systems ask users to fill explicitly a form about his personal data and/or collect data based on his location using for example his IP address; therefore if he is in Egypt for example then he is from the Middle-East, therefore he/she is an Arab, and therefore either a billionaire, bombers, or shepherd. This does not reflect the reality, and also ignores privacy concerns since it depends on collecting personal data.

C) Case- based recommendation.

In this approach, items with highest correlation to the items the user has liked before are recommended. Subsequently, when a new item is added to the web site, the system must find similarities between the new item and the old items, as shown by figure 2.5. As soon as a user visits the web site, the system automatically will recommend items, including newly-added items, with high similarity to the visited item (Smyth, 2007).

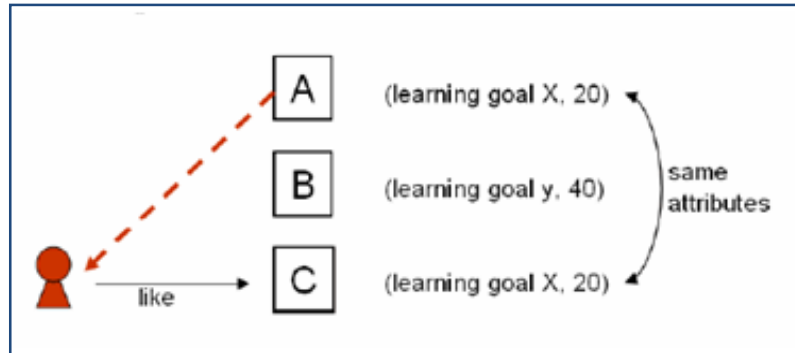


Figure 2.5: Case- based recommendation (Drachsler et al., 2007).

The user known that he like item C. Item A is a newly added, and has similar attributes to item C, so item A will be recommend to the user. Finding related items in this way requires pre-determination of item attributes, and this leads to static views of relationships between items. Although this method solves the item cold start problem, it does not solve the user cold start problem; a significant drawback is that this requires the determination of each item's attributes, and this will often need to be done manually.

D) Attributes-based recommendation.

Attribute-based recommendation systems collect data about both users and items attributes, as shown by figure 2.6. Thus, when a new item is added to the web site or a new user visits the web site, the system collects information about the item's specifications and attributes, and will usually ask the new user to fill in a form to create his or her profile attributes.

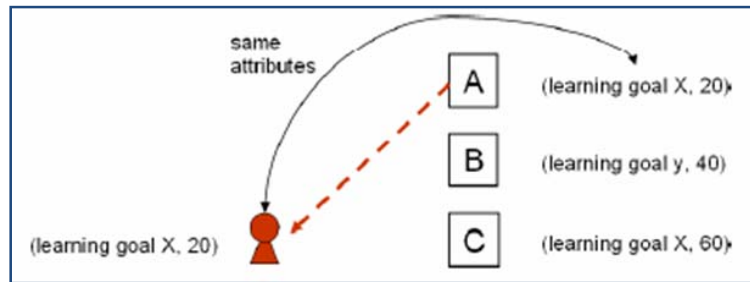


Figure 2.6: Attribute-based recommendation (Kalz et al., 2008).

In systems; which depend on attributes to generate recommendation, collect items attributes and users attributes and then generate recommendation based on mapping between both attributes. Fig. 2.6 shows an item A with attributes learning goal X, and its contents is suitable for visitors of age under 20, while we have a new visitor with attributes learning goal X, and age 20, therefore item A is recommended, since an existing user with similar attributes likes this item. The attribute-based technique solves both cold start problems (item and user) but the main disadvantages of this technique are that the user and item profiles are static, and data collection is done by requiring users to fill in forms or give their interests by selecting from specific pre-prepared categories. In addition, there are problems of over-personalization, since the system sometimes will recommend the same items to the same user. Such systems will not be able to generate recommendation until data has been collected about the new user's preferences, and will not be able to recommend new items unless the new item's specifications and attributes are provided. Table 2.1 shows a summarized comparison between the methods we have discussed so far.

Table 2.1 summarizes the advantage and disadvantage of previous solutions.

Method	Assumption	Advantages	Disadvantages
Demographic data	Users with similar demographic data have the same tastes	1- No user cold start problem. 2- Very easy and simple.	1- Static profiles 2- Reflect groups but not individuals 3- Privacy issues 4- Users annoyance from filling in forms 5- Item cold start problem
Stereotype	All people of the same stereotype are similar and have the same tastes	1- No user cold start problem. 2- Very easy and simple	1- Illusory correlation 2- Biased 3- Represents people entirely in terms of narrow assumptions. 4- May refuse to recognize a distinction between an individual and the group to which he or she belongs. 5- Item cold start problem
Case – base reasoning	If user likes a specific item then he/she will like similar items. Recommends new but similar items	1. No content analysis 2. Domain independent 3. No cold start related to new items.	1. Cold start related to new user. 2. New added item attributes must be determined before being involved in recommendation 3. Sparsity 4. Sometimes recommend the same items

Method	Assumption	Advantages	Disadvantages
Attributes-based technique	Recommend items based on the match between item attributes and user attributes.	1.No cold start problem 2.Mapping between users; profiles and items attributes are simple.	1. Static profiles 2. Does not learn 3. Require regular maintenance 4. Over personalization 5. Force users' to fill forms 6. Privacy problem and/or IP address problem. 7. Suitable to information that can be described by categories such as media like audio, video, etc.

Table 2.1: Advantages and disadvantages of different recommendation methods with reference to the cold-start problem.

2.8.2 The Scalability problem

With tremendous increase in the numbers of existing users and items, which leads to increase in the number of candidate items for recommendations, traditional Collaborative Filtering (CF) algorithms will suffer serious scalability problems, with computational resource requirements going beyond practical or acceptable levels. For example, with millions of online customers (M) and millions of distinct available items (N), a CF algorithm with the complexity of $O(n)$ may already be too large. Also, many systems need to react immediately to online requirements and make recommendations for all users regardless of their purchases and ratings history, which demands high scalability from a CF system (Linden, Smith et al. 2003). Some systems tried to solve this problem by limiting the number of users that are compared when making predictions for the active user. Tang, Winoto et al. 2005 proposed that the rating of the other items by a user should provide enough information to support the target item's predicted rating by that user. Spiliopoulou, Mobasher et al. 2003

proposed the use of heuristics to limit the number of items considered for the movie recommendation domain, so that he suggests the using of temporal features of items (year of release of a movie) to limit the set of candidate movies for recommendation. Sarwar, Karypis et al. 2001 used memory-based *CF* algorithms, such as the item-based *Pearson correlation CF* algorithm to achieve satisfactory scalability. Instead of calculating similarities between all pairs of items, item-based *Pearson CF* calculates the similarity only between the pair of co-rated items by a user. Xue, Lin et al. 2005 used model-based *CF* algorithms, such as clustering *CF* algorithms, to address the scalability problem by seeking users for recommendation within smaller and highly similar clusters instead of the entire database. Several researchers have tried to deal with the issue of scalability, but there are still a lot of challenges in scalability arising from the domain dependency of web personalization.

2.8.3 The Privacy problem

Privacy is defined as the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively (Kienle, 2008). Personalization typically employs data mining and/or collaborative filtering to predict content that is likely to be of interest to individual users. Personalization can be particularly effective when the user identifies himself or herself explicitly to the web site. E-commerce web sites are increasingly introducing personalized features in order to build and retain relationships with customers and increase the number of purchases made by each customer. Individuals appreciate personalization and find it useful. Nevertheless, personalization raises a number of privacy concerns ranging from user discomfort with the computer inferring information about them based on their purchases, to concerns about identity thieves. In some cases, users will provide personal data to a web site in order to receive personalized services despite their privacy concerns; in other cases, users may turn away from a site because of privacy concerns (Ackerman et al., 1999).

Privacy is one of the most current challenges in web personalization systems. All web personalization approaches collect data explicitly or implicitly about visitors to enable them to personalize the user's experience. Creating and maintaining users' profiles represents one of the main tasks in web personalization. However, users become more concerned about their privacy because of the computers' predictions about and potential misuse of data collected

about them. Inadvertently this reveals personal information to other users, when cookies are used for authentication or to access a user's profile, anyone who uses a particular computer may have access to the information in that user's profile (Tsow, Kamath et al. 2007). This leads to concerns such as family members learning about gifts that may have been ordered for them and co-workers learning about an individual's health or personal issues. In addition, when profiles contain passwords or "secret" information that is used for authentication at other sites, someone who gains access to a user's profile on one site may be able to subsequently gain unauthorized access to a user's accounts both online and offline (Arlein, Jai et al. 2000).

The possibility that someone who does not share the user's computer may gain unauthorized access to a user's account on a personalized web site (by guessing or stealing a password, or for example because they work for an e-commerce company) raises similar problems and worries. However, while family members and co-workers may gain access inadvertently or due to curiosity, other people may have motives that are sinister. Thieves, for example, may find profile information very useful (Chellappa and Shivendu 2007). Several systems tried to handle the privacy issue by using pseudonymous profiles (Hansen, Schwartz et al. 2008), client-side profiles (Chen, Han et al. 2007), task-based personalization (Fischer-Hübner 2002), or by putting users in control (Potter 2006), but still privacy represents one of the big issues in web personalization.

A. Privacy risks

Personalization; especially e-commerce personalization, leads to a number of risks to user privacy. The computer's ability to make predictions about users' habits and interests may represent a privacy risk because such predictions may be used unwisely, and perhaps reveal information that the users thought other people did not know about them (Ramakrishnan et al., 2001). The computer may inadvertently reveal personal information to other users who use the same computer. When cookies are used for authentication or access to a user's profile, anyone who uses a particular computer may access to the information in a user's profile. This leads to concerns such as family members learning about gifts that may have been ordered for them and co-workers learning about an individual's health or personal

issues. In addition, when profiles contain passwords or “secret” information that is used for authentication at other sites, someone who gains access to a user’s profile on one site may be able to subsequently gain unauthorized access to the user’s other accounts, both online and offline (Arlein et al., 2000). The possibility that someone who does not share the user’s computer may gain unauthorized access to a user’s account on a personalized web site (by guessing or stealing a password, or because they work for an e-commerce company, for example) raises similar concerns. However, while family members and co-workers may gain access inadvertently or due to curiosity, other people may have motives that are far more sinister. Thieves, for example, may find profile information too useful.

Finally, the possibility that information stored for use in personalization may find its way into a government surveillance application is becoming increasingly real. This places users of these services at an increased risk of being subject to government investigation, even if they have done nothing wrong.

B. Principles of applying fair information practice.

Several principles have been developed for protecting privacy when using personal information (Cranor, 2002). Nevertheless, we should mention here that as restrictions on the data collected about users increase, the efficiency level of personalization is decreased. Some principles associated with this tradeoff are as follows:

1. *Collection restriction*. Data collection and usage should be limited. This means that personalization systems should collect only data that they need, and not every possible piece of data that they might find a need for in the future.
2. *Data Quality*. Data should be used only for purposes of which it is relevant, and it should be accurate, complete, and kept up-to date.
3. *Purpose design*. Data controllers should specify up front how they are going to use data, and then they should use that data only for the specified purposes. In the context of personalization, this suggests that users be notified up front when a system is collecting data to be used for personalization (or any other purpose). Privacy policies are often used to explain how web sites will use the data they collect.

4. *Use constraint.* Data should not be used or disclosed for purposes other than those disclosed under the purpose specification principle, except with the consent of the data subject or as required by law. This suggests that data collected by personalization systems should not be used for other purposes without user consent. This also suggests that sites that want to make other uses of this data develop interfaces for requesting user consent.
5. *Security Safeguards.* Data should be protected with reasonable security safeguards. In the context of web usage personalization especially ecommerce personalization, this suggests that security safeguards be applied to stored personalization profiles and that personalization information should be transmitted through secure channels.
6. *Openness.* Data collection and usage practices should not be a secret. In the context of ecommerce personalization, this suggests that users be notified up front when a system is collecting data to be used for personalization. Users should be given information about the type of data being collected, how it will be used, and who is collecting it. It is especially important that users be made aware of implicit data collection.
7. *Individual Participation.* Individuals should have the right to obtain their data from a data controller and to have incorrect data erased or amended. This suggests, as with the data quality principle, that users given access to their profiles and the ability to correct them and remove information from them.
8. *Accountability.* Data controllers are responsible for complying with these principles. In the context of ecommerce personalization, this suggests that personalization system implementers and site operators should be proactive about developing policies, procedures, and software that will support compliance with these principles.

C. Approaches used to reduce personalization privacy risks

Several approaches have been developed to design systems that reduce privacy risks and make privacy compliance easier; in this section, we will demonstrate such approaches with a critical view.

Pseudonymous Profiles. An individual's name and other personally identifiable information are not needed in order to provide personalized services. For example,

recommender systems typically do not require any personal information in order to make recommendations. If personal information is not needed, personalization systems can be designed so that users are identified by pseudonyms rather than their real names. This reduces the chance that someone who gains unauthorized access to a user's profile will be able to link that profile with a particular individual (Kobsa, 2003). Although it does not eliminate this risk, because maybe someone who gains access to a user's account by using his/her computer or by learning his/her user name and password may be able to gain access to a pseudonymous profile. Nonetheless, pseudonymous profiles are a good way to address some privacy concerns.

In addition, companies may find it significantly easier to comply with some privacy laws when they store only pseudonymous profiles rather than personally identifiable information. For increased privacy protection, sites that employ pseudonymous profiles should make sure that this profile information is stored separately from web usage logs that contain IP addresses and any transaction records that might contain personally identifiable information. Using pseudonymous profiles is therefore still risky since many other privacy issues are still applicable.

Client -Side Profiles, another option for reducing privacy concerns associated with user profiles and satisfying some legal requirements is to store these profiles on the user's client (computer) rather than on a web server. This will ensure that the profiles are accessible only by the user and those who have access to his/her computer. Client-side profiles may be stored in cookies that are replayed to a web site that uses them to provide a personalized service and immediately discards them. The information stored in these profiles should be encoded or encrypted so that it is not revealed in transit and it is inaccessible to viruses or other malicious programs that may look for personal information stored in cookies. Some systems use client side software that users can install to be used as intermediate between web site and client (WK-XO and RUJ, 2005). Although, these procedures help to reduce privacy concerns, many concerns remain applicable; users can turn off such cookies, and also any other people who have access to a user's computer may gain access to their information; also, most users reject using client-side software agents. Using client-side profiles by storing specific data in cookies is also still risky, since some users prevent cookies as well delete them regularly

from their machines to save their resources or avoid malicious software that collects data from cookies.

Putting Users in Control. This refers to the ability to develop systems that give users ability to control the collection and the use of their information. Users should be able to control what information is stored in their profile, the purposes for which it will be used, and the conditions (if any) under which it might be disclosed. They should also be able to control if personalization takes place. In some cases, the law may require such controls therefore a number of e-commerce web sites give users access to their profiles. However, it is not clear that many users are aware of this, and reports from operators of some personalization systems indicate that users rarely take actions to pro-actively customize their online experiences (Mont et al., 2003). Alternative applications that would require less foresight on the part of users can allow them to specify privacy preferences as part of the transaction process. Thus, when a user enters a credit card number and shipping address, that user would also be prompted to decide whether this transaction should be excluded from their profile. In addition, this user might establish a default setting that would apply to all his/her purchases unless indicated otherwise, or even specify general policies. Putting users in control or allowing them to specify their preferences causes them some annoyance; in addition, they will always tend to receive static, unchanging recommendations based on their previously specified preferences.

Generally, whether you are a visitor¹ or a user², the system should have sufficient information about your preferences to generate recommendations. In this thesis, we differentiate between visitors³ based on their real online actions and behaviours that reflect their desires. Collecting visitors' online click streams in a specific way (see next chapter) helps our recommendations systems to generate recommendations for visitors even if they are new users. We consider users by their online actions; therefore every time they visit the web site, they will receive up-to-date recommendations that will be different from time to time. Generating such non-static recommendation; based on users' dynamic and varying desires without asking them for any personal information, increases the loyalty level between

¹ A visitor in this context is one who visits the web site for the first time (temporary user)

² A user in this context is one who usually visits this web site (a permanent user)

³ In this research we use both visitor and user terms as synonymous

the user and the web site. Users usually have specific desires when they are browsing any web site; therefore, we make the assumption that their desires are detectable from their online click streams. As we will describe later, we process the online clickstream to obtain a ‘maximal forward session’; these maximal sessions are stored and used to generate ‘integrated routes’, which represent stretchable routes through the web site that reflect acquired desires (i.e. incorporating recommendations) from all users who have used that site. These integrated routes reflect neither specific persons nor categories of persons, but reflect the abstract patterns of desires learned from all site visitors.

2.8.4 The Diversity problem

The diversity of items in the recommendation set affects user satisfaction; Bradley and Smyth 2001 tried to evaluate the effect of diversification on user satisfaction, applied to item-based and user-based collaborative filtering. The study concluded that introducing diversity affects user satisfaction largely in item-based collaborative models, while it has no measurable effect on user-based collaborative filtering. Diversity was measured as the average distance between the candidate recommendation and all items currently within the recommendation set. McCarthy, Reilly et al. 2005 tried introducing diversity into recommendation sets by balancing similarity of an item to the target with the diversity of the current items within the recommendation set.

Since, web personalization aims to provide useful, contextually appropriate information and services to the user, we must obviously discover the user’s browsing context. The user context is used to predict his or her behaviour so that the system can better serve his or her requirements. It is usually assumed that user behaviour is predictable from past interactions, so that we use previous interactions that were undertaken within the same context and use them to predict the needs of that user. Some systems used client-side web agents that allow the user to interact with a concept classification hierarchy to define the context of the query terms provided; the agent uses a portion of the hierarchy to expand the initial search query, effectively adding user intent to the query. Contextual retrieval also represents a challenge in information retrieval and personalization research (Weitzner, Hendler et al. 2006).

2.8.5 The Robustness problem

Several web personalization systems depend on item ratings provided by users to generate social recommendations. Users may give many positive recommendations for their own materials and negative recommendations for their competitors. In other words, item recommendations can be significantly influenced by intentionally inserting false ratings for a subset of items. This kind of problem is known as an attack. O'Mahony, Hurley et al. 2004 identified two types of attacks: ***push attacks*** are aimed at promoting a particular item by increasing its ratings for a larger subset of users, and ***nuke attacks*** are aimed at reducing the predicted ratings of an item so that it is recommended to a smaller subset of users. Attacking users can use several models: the ***average attack model*** assumes that the attacker knows the average rating for each item in the database and assign values randomly distributed around this average, except for target item (Burke, Mobasher et al. 2005). The ***random attack model*** forms profiles by associating a positive rating for the target item with random values for the other items (Lam and Riedl 2004). Bell and Koren 2007 used a comprehensive approach to the robust attacks problem by removing global effects at the data normalization stage of their neighbour-based collaborative filtering system. The study of attack models and their impact on recommendation algorithms can lead to designing more robust and trustworthy personalization systems. Many questions are raised by the attack concept. Do attacks affect all types of recommendation systems (rule based, content based, collaborative-based systems)? Is it possible to avoid attacks by depending only on agents for ratings? Are the attacks domain dependent?.

2.8.6 The Data Sparseness problem

Sparsity refers to the fact that as the number of items increases only a small percentage of items will be rate by users. Consequently, many pairs of customers will have no item ratings in common and even those that do will not have a large number of common ratings. In addition, the nearest neighbour computation resulting from this scenario will not be accurate, and hence a low rating for an item would not imply that similar items will not be recommended (Anand and Mobasher 2005). The ***low coverage*** problem occurs as a result of the sparsity problem, when the numbers of users' ratings are very small compared with the large number of items in the system, then recommendation system will be unable to generate

recommendations for them; therefore coverage is defined as the percentage of items that the algorithm could provide recommendations for. Some systems used dimensionality reduction techniques by removing insignificant users or items from the user-item matrix (Billsus and Pazzani 1998). Ziegler, Lausen et al. 2004 created users profiles via inference of super-topic score and topic diversification to overcome the sparsity problem. Su and Khoshgoftaar 2006 used model-based collaborative filtering to address the sparsity problem by providing more accurate predictions for sparse data. Huang, Chen et al. 2004 used model-based collaborative filtering techniques that tackle the sparsity problem and include the *association retrieval technique*, which applies an associative retrieval framework and related spreading activation algorithms to explore transitive associations among users through their rating and purchase history. As we can see, various different techniques can be used for solving the sparsity problem, but this usually means discarding a set of users or items, which may lead to loss of useful information and hence affect recommendation quality.

2.9 Evaluating web personalization systems

A successful web personalization system is one that accurately predicts user needs and fulfills these needs. Many criteria are used to evaluate web personal recommendation systems; some are related to the algorithm used to generate recommendations, while others are used to evaluate provided recommendation sets. Therefore, evaluation of web personalization systems needs to consider a number of different issues (Spiliopoulou, Mobasher et al. 2003) such as:

- **User satisfaction:** users are satisfied if the system is pleasant to use; we can detect this from remarks said by the user during the test, or by using a questionnaire.
- **Efficiency:** the resources required to achieve personalization goals, for example if the required times to achieve the task are limited.
- **Effectiveness:** if the user's objectives are achieved, and if they can fulfill their individual needs.
- **Coverage:** is the system able to suggest appropriate recommendations to all users and for all items in an appropriate time? Also we may measure the percentage of the universe of items that the recommendation system is capable of recommending.

Alternatively, measure the percentage of recommended items that were really of interest to the users, rather than considering the complete universe of items.

- **Utility:** the utility of a recommended item can be calculated in various ways. E.g. the distance of the recommended item from the current page, referred to as navigation distance.
- **Robustness:** this measure the extent to which an attack can affect a recommender system.
- **Performance:** these measure the response time for a given recommendation algorithm and how easily it can scale to handle a large number of concurrent requests for recommendations.
- **Precision:** measures the probability that a selected item is really relevant to the user.
- **Recall:** measures the probability that a relevant item is selected.
- **Power of attack:** measure the average change in the gap between the predicted and target rating for the target item.

The evaluation process will be different based on the approach used, and may differ from system to another. Goldberg, Roeder et al. 2001 created accuracy metrics for a prediction task with numeric ratings, including mean squared error of predicted ratings. Massa and Avesani 2004 calculated the mean absolute error for each user and then averaged over all users to evaluate the system as a whole. Recommendation systems tend to have lower errors when predicting users' ratings. Mobasher, Dai et al. 2002 measured precision and recall in order to evaluate if the selected items are relevant as well as to evaluate that the degree to which relevant items are selected. Herlocker, Konstan et al. 2004 measured coverage by calculating the percentage of the universe of items that the recommendation system is capable of recommending. O'Mahony, Hurley et al. 2004 measured the power of attack by calculating the average change in the gap between the predicted and target rating for the target item, where the power of attack metric assumes that the goal of the attack is to force item ratings to a target rating value. Generally, the evaluation process leads to recommend the system or recommend modifications. E.g. what is bad in the system and why? How good is the system?. With the new semantic web concept, the used evaluation criteria need to be match with the semantic structure. Therefore, it is important to use more evaluation criteria such as the **integrability**; which evaluate the ability of the

system to integrate with collected recommendations from different web application. **Shareability** refers to the ability to share used ontologies data resources between different applications that provide different services and/or products. **Expandability or extensibility**: which imply extending or adding to the system by adding any new ontology concepts to specific web site and keep the harmony of these concepts by creating relationships for any new added item(s), such system changes may occur to fit the changes needed and/or desired for the used environment. We evaluated our method against the other alternative methods based on levels of precision, coverage, and novelty. We see that these are the most important criteria for evaluating any recommendation systems in order to ensure the accuracy, coverage, and novelty of provided recommendations. Moreover it is possible to evaluate our method against the other alternative methods based on robustness, power of attack where our method avoid such attacks by using the significance equation as we will indicate in the coming chapters.

2.10 A novel approach to the cold start problem

Our method aims to maintain a click stream based data structure that represents the collective browsing behaviour of all users. We use this data structure to provide information about how users might be thinking when they are browsing the site. In particular, our central assumption is that two users with similar click streams will be looking for similar things. Therefore, we will deal with users based on what they are seeking when they are browsing, which are expressed by their online selections and by the paths that they follow. Consequently, we assume that *“when different users have similar paths through a site, they have similar browsing targets”*. In this case, there is no requirement to calculate similarities between users or similarity matrices; instead, we assume that issues of similarity are compiled into the integrated data routes data-structure that we maintain.

2.10.1 Basic terminology and concepts.

In order to achieve a good understanding of online users, we collect their online behaviour s in the form of ‘maximal forward session’; which reflects a loopless set of visited or selected items in sequential manner. where we consider each online selected item, read topic, browsed

page, purchased item, ...etc. as a node of interest to this specific user. A user can select any node during his online visit and then he/she can move from one node to another, and his or her selections are collected to represent his or her online session. A session is considered maximal when it begins to loop (a cycle), or if it has reached a specific predetermined length. A cycle appears because a user has revisited any previously visited nodes in the current session; when this happens, the sequence of nodes up to and not including the revisited node is saved, and a new session is started. Therefore, we define a user maximal session as a sequence of loopless contiguous selected nodes. These collected 'maximal sessions' can of course be of varying lengths, but in this thesis we generally only consider sessions of lengths between two and twenty, i.e. $2 \leq L \leq 10$. So, when a session has visited 10 nodes without looping, we consider that to be a completed session and then another session starts. In order to reduce computational complexity, we absorb all sessions into 'super sessions'. Basically, this means that when a user session visits only nodes *B*, *C* and *D* in that order, and there is already a stored session that has visited *A*, *B*, *C*, *D* and *E* in that order, the smaller session is absorbed into the stored larger 'super session'. As we will see, this means that parameters and weights of the stored session will change to reflect the history of users using that pathway in the web site. In this way, we get the benefits of collecting sessions from many users, finding a map of the user's interests in the form of paths through the web site. We should mention here that the terms session, route, and path are used as synonyms in this thesis. We used non-cyclic maximal sessions in order to avoid selecting any item in recommendation set while the current user has recently visited. We should mention here that we integrate smaller sessions into a larger integrate routes to find the maximal expected paths which users expected to follow while they browse the website. Therefore, we can define the integrated route as a user-visited path that consists of one or more integrated maximal forward sessions. As well as we limited the length of collected maximal session into the length ten in order to avoid any delay in the recommendation stage. However, we expect to evaluate the system performance with larger session lengths as indicated in our future works.

2.10.2 Understand users' behaviour and goals.

Any user has his or her individual desires and goals when he is browsing a specific site. Somehow, he or she may discover some new knowledge during his browsing, which may

change his or her desires; we call these ‘acquired desires’. Our system will give recommendations that try to satisfy the individual desires by finding connections between the individual desires and the acquired desires. The suggested method will help to predict acquired desires by determining which path the user might follow in his online trip on the web site, given the stored data structures representing other users’ paths through the system, i.e. if a user starts on the route A to D , the system will notice that several other users who started this way went on to visit nodes H and J . For many of the previous users, these may have been acquired desires, not present when they began browsing. But if the weights for this continuation of the path are strong enough, it makes sense to recommend these nodes to our active user, since he or she may already have these desires or be likely to acquire them. Using these concepts we do not need to use users’ personal data, login, or IP address, etc. The system will handle the user as an ‘abstract user’ who follows a specific path on the web site.

2.10.3 Selecting the best routes (the best routes must survive)

The main goal for finding maximal routes through the web site is to find the longest maximal valuable paths that visitors have while they are browsing the web site. In addition, to utilize the benefits of collected users’ sessions, the system will merge small sessions into stored larger sessions. Contextually, all low weighted sessions (short routes, and/or with very little time spent at most nodes) tend to be ignored in our approach, and the remaining stored sessions are the only those that seem to represent significant user interest. By having such impact for considering what sessions and session data to use, the time required for creating recommendations will be reduced.

2.10.4 Recommending the latest valuable items

Our method will recommend the latest valuable items by using users’ online maximal sessions in association with the latest highly-weighted integrated routes (integrated routes refers to the main data-structure that summarizes the behaviour from previous users. The system will provide unknown items to the visitors as recommendations based on the match between his or her individual desires and the acquired desires from similar users. The match

is automatically detected when a specific user goes on a specific path, so that we consider him similar to all users who went through the same path, and the system will give him or her recommendations based on highly weighted pages on that path. As we will explain later, we provide two types of recommendation. The first, *node recommendation*, generates recommendations based on the selected active node. The second, *batch recommendation*, generates recommendations based on the online visited path.

2.11 Summary

Different methods are used to solve the cold start problem. All these methods try to create an initial profile (for a new user, or a new item), but such systems suffer from the following disadvantages:

- 1- Privacy concerns arise since these systems impose a burden on users to fill in forms or otherwise convey their personal data.
- 2- Initial profiles are static and do not reflect the actual situation of the web site (user-based recommendation).
- 3- When the web site is changed by adding new pages, this requires recreation of the initial profile (item-based recommendation).
- 4- User trust of recommendations will be low, since often the user will receive the same initial recommendations.
- 5- New items involve not only the newly added items on the system, but also the ‘old’ items that have never been recommended before to users. Existing systems have problems with both types of new item.

Privacy problems arise as soon as we collect personal information about site visitors; if we try to use different method to identify users’ interests which do not need personal data then we skip privacy problem. Therefore, the method suggested in this thesis will use users’ browsing targets, inferred from the match between their active click-stream and stored abstract click-streams from other users, to identify their current desires and potentially desires that they will acquire on this website, and also to avoid the necessity to create initial profiles, which are static, biased, and time consuming.

In chapter three, we will explain and describe our method, concepts, stages, and its associated algorithms.

Chapter 3

The Active Node Technique

3.1 Introduction.

The main goal of web personalization and recommendation systems is to equip users with what they are looking for on a particular web site. Site visitors provide large number of choices during their browsing; consequently, we face a large number of selections that reflects users' preferences. Therefore, we find different groups of users with different preferences that reflect their browsing targets on the web site, but these groups are not fully separated. Moreover, members of a particular group are not required to carry exactly the same interests; also, members of different groups are not necessarily carrying different interests. In other words, it might be a member of a specific group likes something that is also liked by a member of a quite separate group, and preferences might change over time.

As indicated in chapter two, several researchers have used demographic data (e.g. the triadic aspect model) or stereotype data (e.g. the naïve filterbots method) to generate initial profiles for users. These profiles cannot be trusted to be a reflection of the actual interests of users while they are online; they only depend on demographic data or stereotypes that categorize users into different common categories. In addition, other researchers used case-based recommendation, which depends on generating initial profiles for all site items and categorizing items into different groups. Therefore, when a user browses an item of a specific category, then all other items in that category will be provided as recommendations, which leads to an over-personalization problem. Also, it is time-consuming to determine each item's attributes; this is often done manually. In attribute-based recommendation methods, researchers create initial profiles for both users and items, where initial profiles of items reflect the items' attributes, while initial profiles of users reflect their interests, and then a match between items' attributes and users' preferences is performed to generate recommendations. However, this method is unsuitable from the viewpoint of privacy since it imposes users with the burden of filling in forms about their personal data; also the created profiles remain static and need to be updated from time to time to reflect changes in users interests; also, generated recommendations depend on static attributes and soon do not reflect the actual preferences of users.

In the method proposed in this thesis, we will focus on users' browsing targets during their trip on a particular web site. Therefore if user W is online and follows a forward path (a sequence of visited items) which is a subset of a stored integrated route that contains items A

and C , then we can say that user W is interested in that path and we will recommend items A and C for him or her. We can assume that if two users follow the same path then these two users have browsing similarity (similar interests) and we might recommend items later on that path without the need for recalculating similarity or creating initial profiles. It should be clear that we identify users by the browsing targets that we infer, not by any personal data. Hence, users will receive recommendations based on their online behaviour and selections.

Providing recommendation for new users; using the presented concept, is valid where any new user who enters a web site will be able to follow their own choice of specific path, thereby expressing his/her thinking; therefore the system will be able to provide recommendations based on the paths followed. Again, since users are identified by their online browsing targets, therefore the privacy problems using this concept will vanish even with the inferred information since the system will provide recommendation based on browsing target but not based on users' personal profiles. We should ask ourselves one more question: what about new added items? We consider as new added items not only items that have been very recently added to our web site, but also all items that have never been visited before and therefore have no selection history that helps with recommendations. As we will explain later, new items are given an *impact weight* that is calculated according to the link structure of the web site – this enables it to be added to recommendation sets.

Overall, we assume that *Users who go through a specific path have similar interests as represented by the nodes of this path*. So, if we have a stored representation of a path (maybe integrated from many users), and our active user's path so far matches part of this path, *then we believe the active user should inherit benefits from this stored path*. That is items found later on the stored path will be suitable recommendations for the active user.

In the following sections, we will provide more descriptions and explanations of an elaboration and implementation of the suggested method. In the ***data collation and cleaning*** section, we describe how to collect data using online data collection or by using historical log files, both of which require data cleaning to remove irrelevant data. Therefore, users' log files or online collected data represent the inputs to this stage, and the outputs are a set of cleaned logs or cleaned users click streams in the format¹ that we need. In the ***sequential***

¹we collect page name, start time on the page, and end time on the page

maximal session creation section, we explain rules used for creating users' maximal sessions as well as the algorithm used throughout our implementation; the inputs to this stage are a set of cleaned users' click streams and the outputs are a set of users' maximal sessions. We then have to **evaluate and absorb created maximal sessions**, which is necessary to calculate the significance of each session and remove all insignificant maximal sessions (only significant session must survive). Where a user's session is a subsequence of a session already in the stored profiles, all such sub sessions are absorbed into the stored super sessions (in order to reduce the storage space without losing quality of data). We then update old sessions to reflect the current significance of this session for the current visitors. After updating we calculate the new relative weights of each node in its associated session. Therefore, we now have a set of significant and weighted sessions. In the **integration process**, we try to utilize the benefits of created sessions by drawing on the abstract users' browsing interests. Therefore, the input to this stage is a set of significant and weighted sessions; which are collected from the previous stage, and then by using an integration algorithm we get a set of significant and relatively weighted routes; these routes can be used for matching with the active user's clickstream, and to select candidates for recommendations.

At the **recommendation stage**, for any new user we can provide two types of recommendation: node recommendations, and batch recommendation; we describe node and batch recommendation as well as describe the algorithm used for generating candidates for recommendations, and how it is possible to switch between node and batch recommendation methods. The **evaluation stage** shows different evaluation criteria that we used to evaluate our suggested method, comparing it with alternative methods.

3.2 Description and explanation of the Active node technique

All users' click streams are stored, and integrated into an abstract profile called the *integrated routes profile*. When a new user is browsing the site, his or her click streams are matched with the stored integrated routes to discover what we assume to be his or her target paths, and then the system can start to generate recommendations based on these target paths. We consider a web site visitor as having particular targets that we call *individual desires*. As illustrated in Figure 3.1, he or she browses the site, and if we can find a match between the individual desires and the abstract collective users' desires in the integrated routes profile, in this case, we being able to solve the cold start problem.

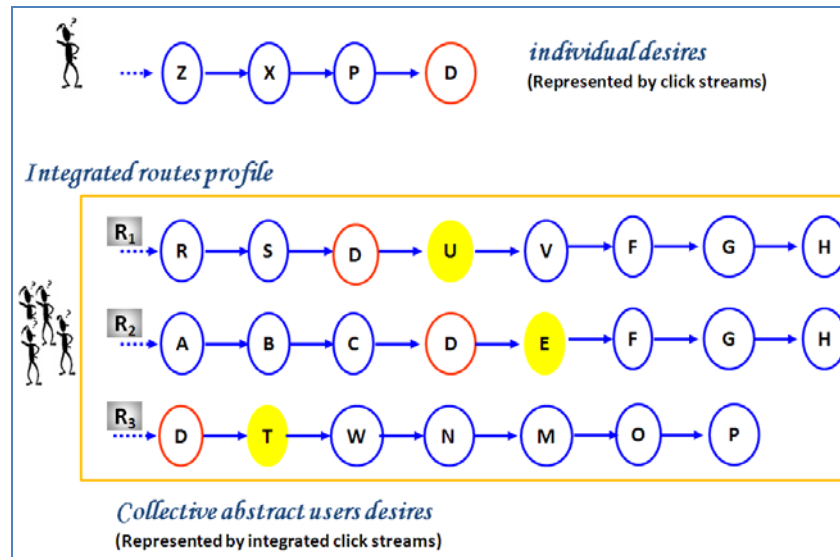


Figure 3.1: Simple example to show user selections in red, and in yellow the selected candidates for recommendations.

Hence, the integrated routes profile represents interests for all abstract visitors, regardless of their identification data, and the new visitor (again, not identified) is able to benefit from this to receive promising recommendations. We illustrate this again in another way in Figure 3.2, which emphasizes the potential overlap between the active user's desires or browsing targets, and those of previous visitors.

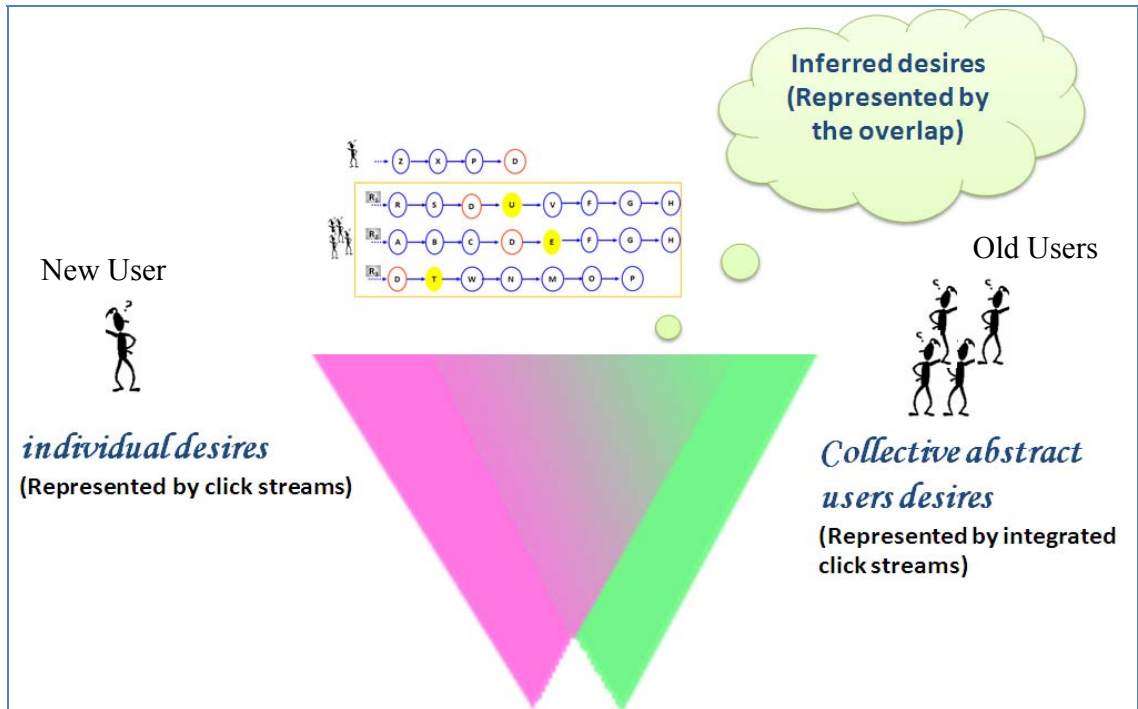


Figure 3.2: User online path(s) shows the extent of the overlapping between Individual and collective users desires.

We restate the basic idea here in order to introduce some terminology that we use. The method depends on the assumption that we can infer a user's online browsing targets; this is one by taking into account each node visited in the user's path. At any time, the page (sometimes referred to as 'item', since it may be a description of a particular product) that the user is currently viewing is considered as the *active node*, while the current online maximal path (to be defined later) is considered to be the current *active path*. We provide a system flow chart and data flow diagram for the method in appendix B. Meanwhile, the following sections explain the different stages in the active node technique.

We use the term 'profile' to refer to a database table which is used to store users' click streams. As illustrated in figure 3.13, we store four profiles (tables), three of them are temporary, storing sessions that are removed as soon as the processing and calculations are completed; these are the front-end profile, back-end profile, and universal profile. The front-end profile is used in data collection and cleaning (section 3.2.1) for temporarily maintaining collected users' click streams (selected nodes, start time, end time). The back-end profile is used to temporarily store sequential maximal sessions for online abstract users (section

3.2.2); these temporary maximal sessions need to be put into a proper format by calculating the time the user has spent on each node and the session's total duration. The universal profile is also a temporary profile that is used to maintain absorbed sessions from the back-end profile; as shown later in section 3.2.3. Only one profile is permanently stored, called the *integrated routes profile* which is used to store integrated routes (created from universal profile data) that reflect the integrated preferences 'abstract' (unidentified) users, and this profile is used to find candidates for recommendations (as shown in section 3.2.4).

3.2.1 Data collection and cleaning

Usage data can be collected from data in the server log files or by online data collection. Log files can be collected on several levels, such as the server level, proxy level, or client level. The server log files provide a list of page requests made to a given web server in which a request is addressed by, at least, the IP address of the requesting machine, the date and time of the request, the URL of the requesting page, and number of bytes, status, method, and other items related to the log file format. From this information, it is possible to reconstruct the user's historical navigation sessions within the web site (a 'session' consists of a sequence of web pages viewed by a user in a specific visit). Not all log file data are important for our purposes, therefore such log files should be cleaned and only the required data will be captured and used in the next stage. Users' behaviours can be captured while they are online, therefore only the required data will be collected and processed directly into the suitable form before storage.

3.2.2 Creation of sequential maximal sessions

Whatever the way that data has been collected, it should be in a suitable form required for processing, which is the *sequential maximal forward session* form. A user session refers to all pages accessed by that user during a single visit to a specific site. Therefore, from the cleaned log files or through online data collection we will get a set of sessions S where:

$$S = \{s_1, s_2, \dots, s_j, \dots, s_n\} \quad (3.1)$$

containing n sequential maximal forward sessions, where each session s_j consists of s_j^p pages. A specific session s_j is an ordered list of triples (p_j, t_j, w_j) , where p_j denotes the page title, t_j is the time spent on that page, and w_j is an associated weight.

$$s_j = ((p_1^j, t_1^j, w_1^j), (p_2^j, t_2^j, w_2^j), \dots, (p_l^j, t_l^j, w_l^j)) \quad (3.2)$$

Where $l \geq 2$, and we refer to each set of non-cyclic sequential order triples as a sequential maximal forward session. In a stored maximal forward session, the aim is that it should be long enough to be useful, and also not include repetitions. In this way we aim for a compact representation of the user's interests.

A) Rules used to generate sequential maximal forward sessions.

1. Loops should not occur in a maximal forward session; therefore we generate a maximal session from the user's click stream as soon as a repeated node appears.
2. The length of a stored maximal session is limited to 10. That is, if the user's clickstream sequence has visited 10 different pages, then (even if the next page visited is again different and does not introduce a loop), we store this session and start a new one. The main reason for having this limit is to reduce delays in being able to generate recommendations.

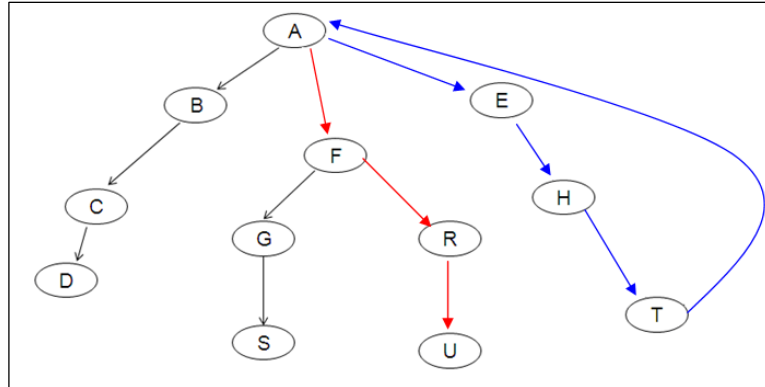


Figure 3.3: A website viewed as a network of nodes.

As illustrated in figure 3.3, if a user's online click stream show that he visits the following nodes in sequence, {AEHTAFRU}, then our method will create two maximal forward sessions $s_1 = \{AEHT\}$, and $s_2 = \{AFRU\}$. It is clear that with the appearance of a cycle (at the second visit to node A), a session is terminated and stored, and a new session is started, therefore we get a forward session.

These rules can be used to create maximal forward sessions using online data collection or by using log files. Using online data collection, as soon as a specific user starts browsing the web site, our system will collect the required information regarding the visited pages and time spent per page. Then the system will create sessions based on the concept of maximal

forward session and store these temporarily in the *front-end profile*, F_P , which will contains several different maximal sessions likely to be of varying length i.e.

$$F_P = \{s_1, s_2, \dots, s_n\} \quad (3.3)$$

Where s_1 is the first online maximal session, and s_k is the last online maximal session of the current user on his current visit. It is important to remember that we are dealing with abstract visitors and do not use their personal data or IP addresses.

B) Algorithm for creating sequential maximal forward sessions

This algorithm represents the rules described in the preceding subsection, which is used to create contiguous sequential maximal sessions from users' click streams, therefore the inputs to this algorithm are a set of cleaned log files or a set of users' online click streams, and the output is a set of contiguous sequential maximal sessions. The following steps illustrate how we created the maximal sessions; using the algorithm shown by figure 3.4.

1. Initialize an empty maximal session s , current active node "*page*", maximal session length $l = 0$, and end-session as a Boolean variable initially *False* but becomes *True* when the user exits from this web site.
2. Read the next page P from the user's click stream
3. If P is null, this means the user has left the site and then we terminate and store this session as a new maximal session, as long as l is greater than or equal to 2.
4. If P already appears in the current session, then a cycle is found; the current maximal session is terminated and stored only if l is greater than or equal to 2, and then a new maximal session is started.
5. Add P to the maximal session S , and increment length l .
6. If the current maximal session length l has reached 10, then the maximal session is terminated and stored, and then a new maximal session is started.

Figure 3.4 shows the more precise algorithm for creating sequential maximal sessions.

```

1. Begin
2. Set  $s = \{\}$  // declare an empty maximal session
3. Page = "" // declare page variable
4.  $l = 0$  // length of session
5. Set end_session=False // declare a Boolean variable to check end of current
   session
6. Do
   Page = read visited page name
   If (Page==Null) then
     end_session=True
   End if

   If Not_in ( $s$ , Page) &&  $l < 10$  // function to detect repeated nodes
      $s \leftarrow s \cup \text{Page}$ 
      $l++$ 
   Else end_session=True
     Create maximal session
     Store maximal session // store maximal session in the front-end
   profile
      $S = \{\}$  // restart a new session
   End if
   Loop while end_session==False
7. End

```

Figure 3.4: Algorithm for creating maximal forward sessions from user's click stream.

C) Calculate each session's time duration

All maximal sessions collected via the algorithm in Figure 3.4 are initially stored in the front-end profile, and then transferred to the back-end profile for further processing. At this point, the time spent by the user at each node is calculated from online collected data, as the difference between the node's start time and end time. The representation of a maximal forward session in the back-end profile includes the duration at each node, and also the total

duration for the session. We consider the time of termination of the session to be the end time of the last node in that session.

3.2.3 Evaluation and absorption of maximal sessions

In this section we will demonstrate the following components of the active node technique:

1. How we determine the significance of each session (sessions considered insignificant will be discarded).
2. How we calculate and update weights for the pages in a session.
3. How a small session is absorbed into a larger ones that may be already stored, which includes it as a sub-sequence.

A) *Significance of a sequential maximal sessions*

In this stage we will evaluate the significance of each session. This depends on what we name the *impact* value of the pages in a session. Each item (page) has an impact value, and we explain this next.

Calculating the impact of an item

The impact value of a page is initialized to zero for all pages on the web site. When sessions have been recorded and stored, we can then calculate the impact for each page, which is basically the average time spent by users on that page. We should mention here that for any recently added item, its impact value will be set to zero until the item has become selected by users. Equation (3.4) is used to calculate each item's impact value.

$$\text{Impact}(x_j) = \frac{\sum_{i=1}^k \text{time}(x_i)}{k} \quad (3.4)$$

where the numerator refers to the total time on item x_i by site users over all session (k of them) which contained it. This becomes updated, as we will see, as new maximal forward sessions are generated.

Calculating the significance of a session

We consider a session to be significant if it will make clear differences to the integrated routes already stored. The significance of a session is also an estimate of how much it reflects the users' real interests, and it depends on the time spent by the user during that session. However, we expect that sessions with very low durations and also sessions with very high durations are likely to be invalid, because there seems to be a good chance that the user was being inattentive in both cases. We reject outliers because it reflects abnormal actions by some users on the website in order to affect the system performance and recommendations. Equation 3.5 is used to calculate each item's significance value.

$$Sig(s_j) = \frac{\sum_{i=1}^n time(x_i) - Min_I}{Max_I - Min_I} \quad (3.5)$$

Where,

$Sig(s_j)$, is the significance of a specific session,

$\sum_{i=1}^n time(x_i)$, the session's total time duration

Max_I , the highest impact value of items from that session

Min_I , the lowest impact value of items in that session.

We now look at a worked example. In Table 3.1, for each of 5 sessions, we see the time spent by the user on the items A, B, E and G in that session (e.g. in session 2 the user spent 6 seconds, 4 seconds, 11 seconds, and 14 seconds respectively on these items. The Impact column shows impact values for each time (these are assumed to be based on previous collected data about abstract users' visits). The bottom row of the table shows significance values calculated according to equation 3.5. Each session has a significance value, and we can see that they are varying.

We will now discard the sessions whose significance value is likely to be untrustworthy – these are the ones with too low or too high significance. Figure 3.5 shows the regions of acceptance and rejection of significance values. A session will only be retained in the integrated routes profile if its significance is in the region of acceptance, while sessions with very low and very high significance value reflect an attack, therefore we omit such sessions.

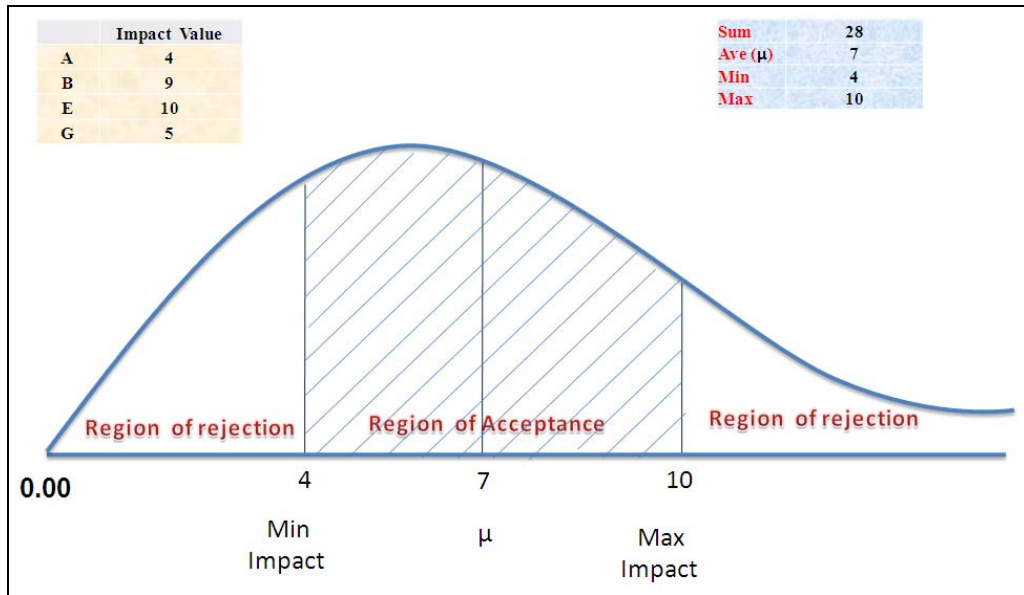


Figure 3.5: Region of acceptance and rejection.

A session elements	Impact	S_1	S_2	S_3	S_4	S_5
A	4	5	6	2	15	15
B	9	6	4	3	30	9
E	10	8	11	4	7	10
G	5	8	14	2	40	7
Sum	28	27	35	11	92	41
Ave (μ)	7	6.75	8.75	2.75	23	10.25
Min	4					
Max	10					
S		3.83	5.17	1.17	14.67	6.17

Table 3.1: Significance calculation example.

It may be helpful to work through the Table 3.1 example in more detail. Note that all items' impacts are greater than zero which means that all session items have been visited before by web site users. Given recent new sessions $A \rightarrow B \rightarrow E \rightarrow G$ from five users, as shown in the table, we can calculate the significance of session s_1 using equation 3.5 as follows:

$$Sig(s_1) = \frac{27 - 4}{10 - 4} = \frac{23}{6} = 3.83$$

So, the significance of that session for users so far is 3.83, which is lower than the minimum threshold, therefore this session will not be taken into consideration in further processing.

Considering the region of acceptance in Figure 3.5, the end result after calculating the significances of these sessions is shown in Table 3.2.

A session elements	THRESHOLD	S ₁	S ₂	S ₃	S ₄	S ₅
A	4	5	6	2	15	15
B	9	6	4	3	30	9
E	10	8	11	4	7	10
G	5	8	14	2	40	7
Sum	28	27	35	11	92	41
Ave (μ)	7	6.75	8.75	2.75	23	10.25
Min	4					
Max	10					
S		3.83	5.17	1.17	14.67	6.17
Significance		Low (Region of rejection)	Moderate (Region of acceptance)	Low (Region of rejection)	Extreme (Region of rejection)	Moderate (Region of acceptance)
Consider it		No	Yes	No	No	Yes

Table 3.2: Significant and insignificant sessions.

Only sessions S₂ and S₅ will be considered, and all integrated routes that this session is a subset of will be affected and their weights will be updated; the impact values for the elements of the session will also be updated.

If any element in the session is new or visited for the first time, then its impact value will be zero. In this case the minimum threshold for acceptance will be zero. On the other hand, if all elements of a session are visited for first time, then the minimum and the maximum threshold values will be equal to zero and in this situation there is no need to calculate significance and we will accept the session. The worked example in Table 3.3 shows a session with an element visited for the first time.

A session elements	THRESHOLD	S ₁	S ₂	S ₃	S ₄	S ₅
A	4	5	6	2	15	15
B	0	6	4	3	30	9
E	10	8	11	4	7	10
G	5	8	14	2	40	7
Sum	19	27	35	11	92	41
Ave (μ)	4.75	6.75	8.75	2.75	23	10.25
Min	0.00					
Max	10					
S		2.70	3.50	1.10	9.20	4.10

Table 3.3: Calculate the significance of a session with new element.

As shown in table 3.3, element **B** has impact equal to zero which means it has not been visited before, therefore the region of acceptance for this session will be as shown in Figure 3.6.

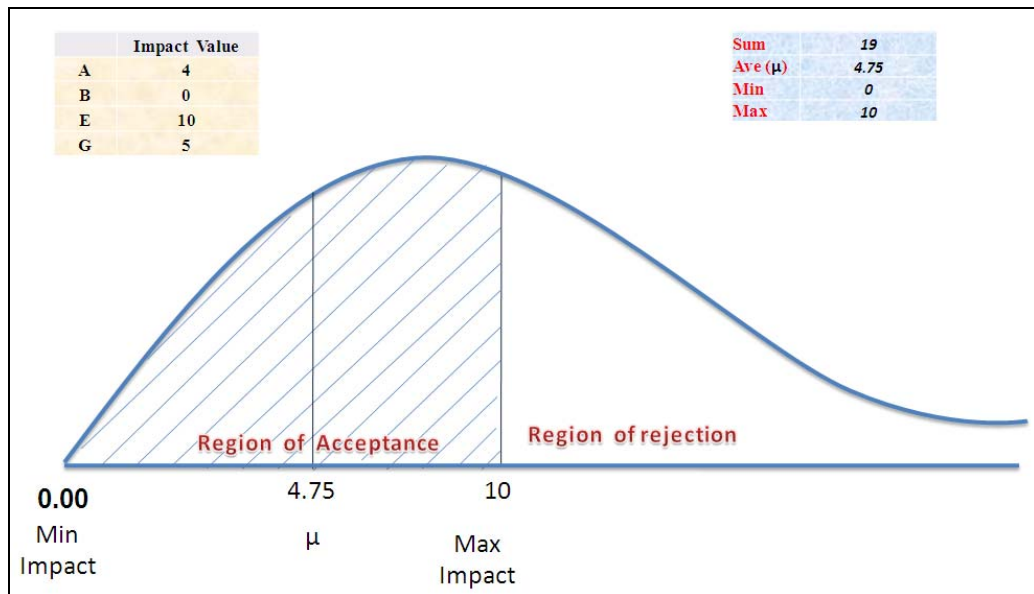


Figure 3.6: Region of acceptance and rejection with new added item.

We can calculate the significance of each session again using equation 3.5. E.g. the significance of session s_4 is calculated as follows:

$$Sig(s_4) = \frac{105 - 0}{10 - 0} = \frac{105}{10} = 10.50$$

In this case the session s_4 significance is too high for us to consider as valid, and our method will reject it, while the other users' sessions are moderate therefore we will consider them as shown by table 3.4.

A session elements	THRESHOLD	S ₁	S ₂	S ₃	S ₄	S ₅
A	4	5	6	2	20	15
B	0	6	4	3	30	9
E	10	8	11	4	15	10
G	5	8	14	2	40	7
<i>Sum</i>	<i>19</i>	<i>27</i>	<i>35</i>	<i>11</i>	<i>105</i>	<i>41</i>
<i>Ave (μ)</i>	<i>4.75</i>	<i>6.75</i>	<i>8.75</i>	<i>2.75</i>	<i>26.25</i>	<i>10.25</i>
<i>Min</i>	<i>0.00</i>					
<i>Max</i>	<i>10</i>					
<i>S</i>		<i>2.70</i>	<i>3.50</i>	<i>1.10</i>	<i>10.50</i>	<i>4.10</i>
<i>Significance</i>		<i>Moderate</i> (Region of acceptance)	<i>Moderate</i> (Region of acceptance)	<i>Moderate</i> (Region of acceptance)	<i>Extreme</i> (Region of rejection)	<i>Moderate</i> (Region of acceptance)
<i>Consider it</i>		<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>

Table 3.4: Significant and insignificant sessions with new items.

B) Calculation of Session Page Weights

Previously, we have calculated an impact value for each item (taking into account all sessions it has appeared in), and shown how this leads to calculations of significance values for each session. After this, we calculate the relative weight of each item in its associated session. This is simply the proportion of that session's total duration that was spent on this particular item. Hence we use equation (3.6) to calculate the relative weight of an item with respect to a particular session.

$$W(x^{s_k}) = \frac{time(x^{s_k})}{time(s_k)} \quad (3.6)$$

$W(x^{s_k})$ refers to the weight for page x in a maximal session s_k , and

$time(x^{s_k})$, refers to the total time spent by the user on item x in session s_k , and

$time(s_k)$, refers to the total spent time on the maximal session s_k .

Figure 3.7 shows a significant maximal forward session, with each item labeled with the time spent on that session. Table 3.5 shows the associated relative weight of each item in this session.

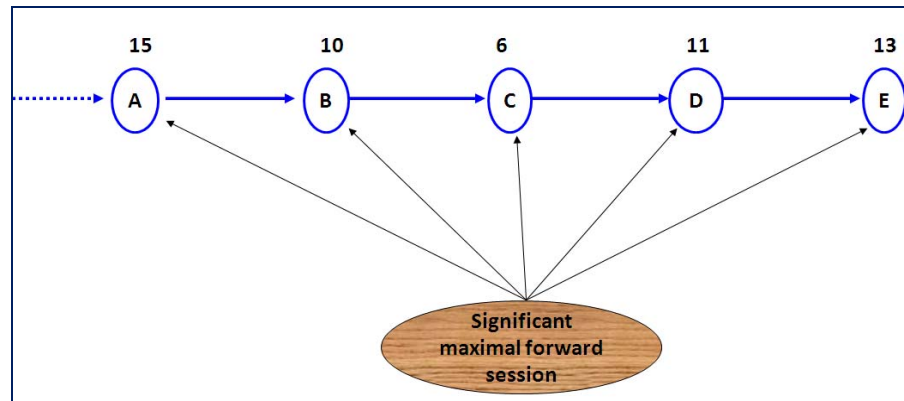


Figure 3.7: A user significant maximal forward session.

We should recall that sessions are of different lengths, where some of these sessions are subsets of other (super) sessions. Therefore sessions that are subset of other sessions will be absorbed by the latter in order to reduce the computations required to find candidates for recommendations.

page	spent time/ Minute user X	Page Weight
A	15	0.27
B	10	0.18
C	6	0.11
D	11	0.20
E	13	0.24
	55	1.00

Table 3.5: Relative weight of the items in the session of Figure 3.7.

This absorption process is based on relative weights, rather than absolute times spent on the pages, which gives a fairer picture of the importance of a page when the values are updated during the absorption process.

C) Absorption process (sessions absorbing other sessions that are subsets)

In the absorption process (AP), if $P(S_k)$ is the ordered set of pages visited in session S_k , then whenever we have $P(S_i) \subseteq P(S_j)$, the integrated route profile (IRP) will only Store S_j with appropriately recalculated weights. Therefore, as soon as an absorption case is detected, we will update the larger session and remove the smaller one.

Consider the two sessions described in Table 3.6, which shows the page weights for each of two sessions.

(Super session)		(Sub session)	
page	Page Weight	page	Page Weight
A	0.21	B	0.30
B	0.27	C	0.45
C	0.14	D	0.25
D	0.20		1.00
E	0.18		
	1.00		

Table 3.6: Super and sub session items relative weights.

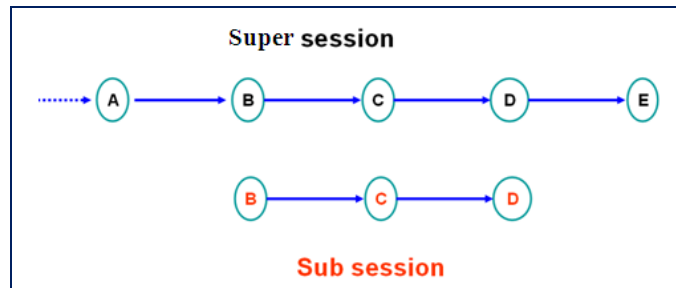


Figure 3.8: Super and sub session

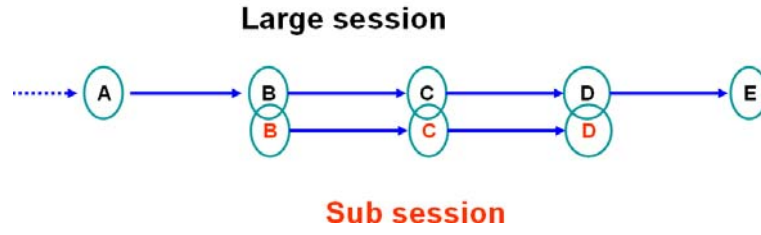


Figure 3.9: One session absorbs another session that is a subset of it.

As illustrated in Figures 3.8 and 3.9, the sub session is absorbed by the larger one, and then we need to update the relative weights of the super session items, to reflect how their importance to abstract users. This absorption process is done offline, during a pattern discovery phase, and it is performed on the abstract sessions stored in the back-end profile (which becomes empty after absorption is completed) and the universal-profile, hence all super sessions will be stored in the universal profile.

Absorption steps

Given each new session in the back-end profile, the following steps represent the absorption process.

1. Find a ‘super-session’ in the universal profile that should absorb this session.
2. If no absorption case is found for this session, then it is stored as a standalone session in the universal profile, and we return to step 1 for the next session
3. Calculate temporary weights for the session,
4. Recalculate the relative weights in the super-session.
5. Update the super-session in the universal profile,

We provide more detailed description about the absorption process as follows.

1. Finding an absorption case for the selected session

As we indicated before, an absorption case exists when a new session is a sub-sequence of session that already exists in the universal profile

2. Calculating temporary weights

The length of the super-session will be greater than or equal to the length of the session being absorbed. In either case, we can update the super-session weights as follows:

- A. Suppose $P(S_i) \subseteq P(S_j)$. For each item in $P(S_j) - P(S_i)$, the temporary weight is equal to the weight of that item in S_j .
- B. If an item is in $P(S_i) \cap P(S_j)$, then its temporary weight is the mean of its weights in S_i and S_j .

$$\text{Temporary Weight of item (x)} = \begin{cases} \text{PSW(x)} \therefore \text{Super session weight of an item x,} \\ \quad x \in \text{super session, and } x \notin \text{sub session} \\ \\ (\text{PSW(x)} + \text{SSW(x)})/2 \therefore \text{Average weight of node} \\ \quad \text{x, } x \in \text{sub session weight, and } x \in \text{super session} \end{cases} \quad (3.7)$$

As soon as we calculate the temporary weight, the weights of the updated sessions are renormalized, so that the total session weight is 1.

$$\text{Updated weight}(x) = \frac{\text{Temporary weight}(x)}{\text{Total temporary weight}} \quad (3.8)$$

The following table 3.7 shows the output of this recalculation process for the two sessions in table 3.6.

page	Larger session Weight L	Sub Session Weight S	Temporary Weight	Recalculated weight
A	0.21	-	0.210	0.176
B	0.27	0.3	0.285	0.238
C	0.14	0.45	0.295	0.247
D	0.2	0.25	0.225	0.188
E	0.18	-	0.180	0.151
Total	1	1	1.195	1.000

Table 3.7: Example of recalculation of items' weights after absorption.

The Absorption algorithm

Figure 3.10 shows detailed pseudocode for the absorption process, where the inputs to the absorption process are a set of generated maximal forward sessions, and the outputs are a set of significant and relatively weighted super sessions.

```

1. Begin
2. Set S={all maximal forward sessions stored in the back-end profile}
3. Initialize session counter = 0 // a counter for the Universal profile sessions
4. Initialize i = 1 // a counter for the back-end profile session
5. While S // S is not empty
    Read session  $S_i$  // read a session number i
    Match = no
    Counter = 1
    // compare selected back-end profile session with
    // universal profile sessions
    While NOUP // Not end Of Universal Profile
        If Super_Sub ( $S_i$ , UP (counter)) // detect absorption case
            Match = Yes
            If Super ( $S_i$ )
                Update_Weight ( $S_i$ )
                Update_UP( $S_i$ )
            Else
                Update_Weight ( UP (counter) )
                Update_UP ( UP (counter) )
            End if
        End if
        Counter ++
    Loop
    // if a session has no super session then store it to the universal profile,
    // and the session is a super session of itself
    If Match = no then
        Store ( $S_i$ )
    End if
    i++
Loop
6. Empty(BEP) // Empty the back end profile sessions
7. End

```

Figure 3.10: Absorption algorithm.

We capture all significant sessions from the back-end profile, and then compare each to the universal profile sessions using the Super_sub function. If an absorption case is found

then we update the super-session and update/store it to the universal profile; if the selected session has no absorption case (not matched) then we store it to the universal profile, and then select another session from the back-end profile. After finishing the absorption process, all back-end profile sessions should be removed.

3.2.4 The Integrated Routes Profile

Sessions in the universal profile are next used to update the main datastructure at the heart of our technique – this is the *Integrated Routes Profile*. The main goal of the integrated routes profile is to represent typical user's paths through the site in a flexible and compact way, supporting the generation of recommendations, while minimizing computation time (for recommendation generation) and storage needs. We can combine two sessions in the universal profile into an *integrated route* if there is an intersection between the beginning of one and the end of the other. If we have the two sessions for example as shown by figure 3.11 in the universal profile, then we get an integrated route as shown by the same figure.

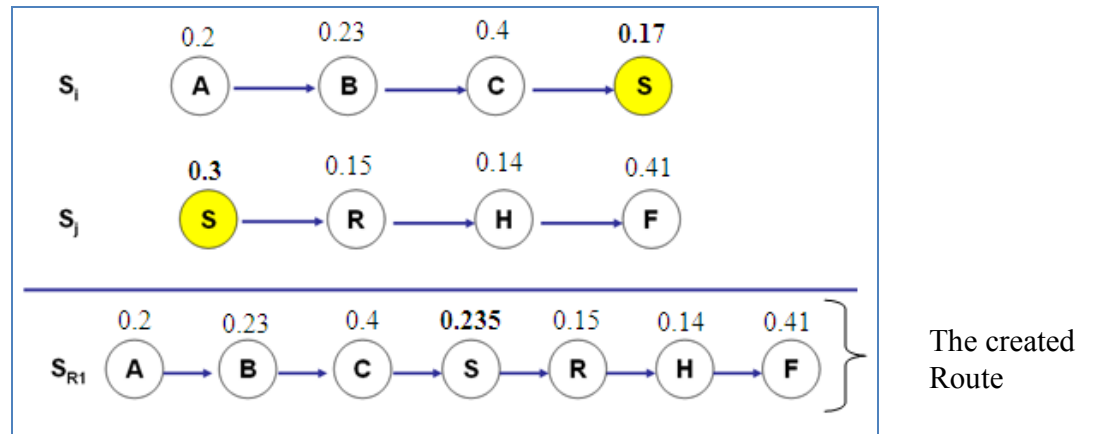


Figure 3.11: Integrated route creation.

If the same route is found already in the integrated routes profile, the absorption process amounts simply to updating its weights. It is necessary to mention here that the number of routes created in this way must be less than or equal to the number of sessions in the universal profile. The following steps outline how integrated routes are created.

1. Select a session from the universal profile
2. Find an integration case involving this session; this happens if the beginning/end of the selected session (in step one) matches with the end/beginning of any other session in the universal profile.
3. Given that an integration case has been found, create the integrated route.

4. Store the created integrated route in the integrated route profile (or if the same route is found in the IRP, then update it in IRP).
5. While there are more unprocessed sessions in the universal profile, return to step 2.

A) Algorithm for creating integrated routes

Sessions stored in the universal profile (which no longer contain any cases for absorption) are used as inputs for integration processing in order to generate integrated routes as outputs. The created integrated routes will be used for generating recommendations for abstract users. Figure 3.12 shows pseudocode of the algorithm used for creating integrated routes.

```

1. Begin
2. Set  $S = \{\text{all sessions in the universal profile}\}$ 
3. Initialize session counter = 1
4. Initialize  $i = \text{counter} + 1$ 
5. Declare Match as Boolean
6. While Not EOUP // not end of the universal profile
    Read a session  $S$  of number "counter"
    Match = no
     $i = \text{counter} + 1$ 
    While Not EOS // not end of sessions  $S$ 
        If  $\text{GetEnd}(S(\text{counter})) \equiv \text{GetBegin}(S(i))$ 
            Route =  $\text{IR}(S(\text{counter}), S(i))$  // create a route
             $\text{IRP} \xleftarrow{\text{Store / Update}} \text{Route}$  // store a route
            Match = yes
        Else If  $\text{GetEnd}(i) \equiv \text{GetBegin}(\text{counter})$ 
            Route =  $\text{IR}(S(i), S(\text{counter}))$  // create a route
             $\text{IRP} \xleftarrow{\text{Store / Update}} \text{Route}$  // store a route
            Match = yes
        End if
        Increment  $i$ 
    Loop
    If Match = no then  $\text{IRP} \xleftarrow{\text{Store / Update}} S(\text{counter})$  // store the session itself
    Increment counter
Loop
7. End

```

Figure 3.12: Integrated routes algorithm.

As shown in Figure 3.12, when a session becomes integrated, we remove it from the universal profile. If a session does not become integrated, then it is stored as a standalone

session in the integrated routes profile, and in this case too it is removed from the universal profile, (see System pattern discovery flow chart in appendix B). Thus, the integrated routes profile becomes the sole stored data structure that summarizes abstract users' usage of the site.

B) Abstract users profiling

As indicated earlier, we collect abstract click streams, and then each visited node (selected item) is captured and stored in the front-end profile in association with its start and end time in order to create user's maximal sessions. Then these maximal sessions are transferred into the back-end profile and stored with each node's name and duration. Absorption processing is done to these maximal sessions in the back-end profile, then the absorbed sessions are stored in the universal profile, which is used later for integration processing, and then all integrated sessions are stored in the integrated routes profile; which is a profile for all users and used to select candidates for recommendations. It is important to mention that maximal sessions creations and recommendations generation are done online, while the absorptions, session significance evaluation, and integrations processes are done offline. Figure 3.13 shows the full sequence of this process.

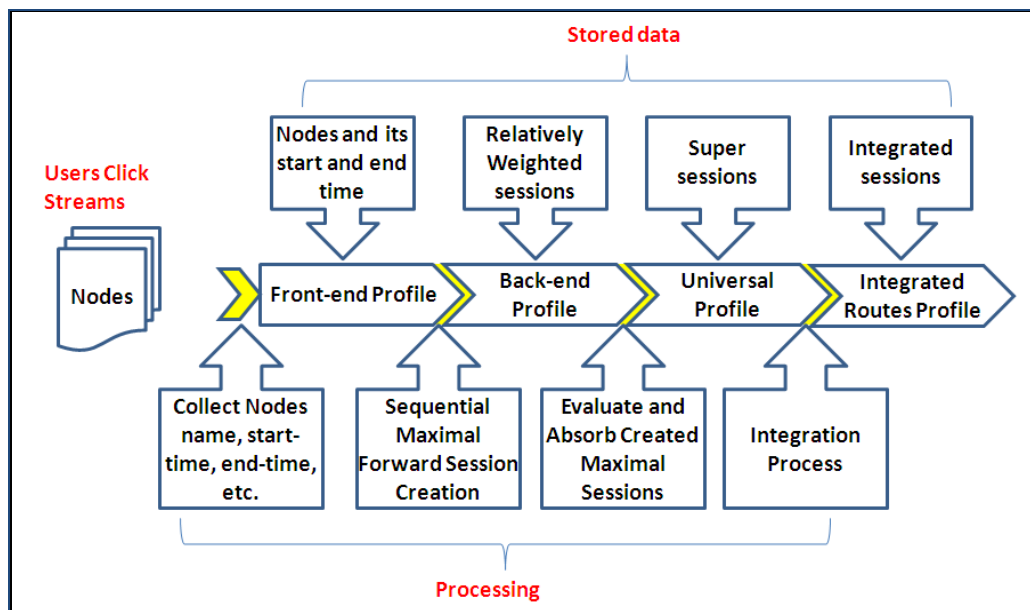


Figure 3.13: Users sessions profiling.

C) Validity of the integrated routes profile

Is the integrated routes profile valid and useful? Traditional collaborative systems collect users' preferences and then measure the similarity between users. For any changes in a user's preferences, such systems must recalculate the similarity. The required time for calculating and recalculating similarity can be problematic. In addition, these systems will not be able to give appropriate recommendations for any new user until the system collects the required information about his/her preferences. Our method tries to find users' preferences via the integrated route profile (IRP) and will not recalculate similarity but will update the previously stored routes, and users who follow specific paths will inherit recommendations from this path in the IRP, based on the implicit similarities in preferences that have appeared from users who have followed the same path previously. Using the IRP in this way achieves the following benefits:

- 1- No requirement to re-calculate similarity matrices or similar data-structures with every change in users' data.
- 2- Reduced time required to find candidates for recommendation, since the creation of integrated routes reduce the number of stored sessions and hence reduces the required time to generate a recommendation set. Also, recommendations are found by following the integrated route matched by the user's current session, with no need to constantly compare similarity with many stored user profiles (for example).
- 3- More flexibility for creating recommendations, especially batch recommendation since we look for a larger sessions which the current online user session is a subset of (see section 3.2.5).
- 4- Helps to solve the cold start problem, since a specific user can visit the web site starting from any node, which will already be involved in some routes in the IRP.
- 5- Storage requirements are low, since our system will store only the integrated routes; back-end, front-end, universal profiles' data will be deleted on completion of the integration process.

The current user's click streams will be used to determine his/her path, and then the system will make recommendations based on the match between his/her current path and the stored integrated routes (the acquired desires). Therefore storing longer routes is important to recommend a variety of highly weighted nodes on that path.

Of course, these benefits may be matched by some disadvantages. The active node technique depends on the integrated routes profile to make recommendations, and when this is created there is much loss of information that does not happen in other kinds of system. In a system that stores all users' browsing data, the computation time and storage requirements are problematic, but such a system is always able to find the most similar previous browsing pattern, and this may lead to more accurate recommendations in some circumstances. However we hypothesise that our system maintains the ability to provide appropriate recommendations, and this is tested in later chapters.

D) Incorporating new added items in the recommendation process

When a new item is added to the web site, we would like to infer a suitable weight for that item so that it might be recommended appropriately. Therefore, we make use of the link structure that arises when the item is added. There will be always be at least one link on the site to the new item, from items (pages) already in the system (e.g. when a new book is added to Amazon, it will be linked from a 'New Books' page, as well as other pages relating to its category). To infer a suitable weight for this item, we use a '*virtual weight*', which reflects the expected weight of new items by all site visitors who have preferences relating to this new item, as shown by Figure 3.14. We consider all hyperlinks between nodes as e , where $e=1$ if the hyperlink (effectively, this is a semantic relationship) is found, else $e=0$. Also, every item appears or selected in sequential manner with any other item stored in the integrated route has a real weight w . Let N be the new item, and let $X=\{x_1, x_2, x_3, \dots, x_n\}$ be the set of items that link to N . Let A be the active node. When there is a path $A \rightarrow x_i$ for any x_i in X , then we can calculate a virtual weight for the link $A \rightarrow N$.

The following formula 3.9 is used to calculate the virtual weight between the active node A and the new item N .

$$W_v(A \rightarrow N | A \rightarrow x_i) = e.w_r(A, x_i) \cdot \frac{\text{Impact}(x_i)}{\text{Impact}(A)} \quad (3.9)$$

Substituting the equation for the impact calculation (equation 3.5), this becomes

$$W_v(A \rightarrow N | A \rightarrow x_i) = e.w_r(A, x_i) \cdot \frac{\frac{\sum_{i=1}^n \text{time}(x_i)}{n}}{\frac{\sum_{j=1}^k \text{time}(A_j)}{k}} \quad (3.10)$$

Which simplifies to:

$$W_v(A \rightarrow N | A \rightarrow x_i) = e.w_r(A, x_i) \cdot \frac{k \sum_{i=1}^n \text{time}(x_i)}{n \sum_{j=1}^k \text{time}(A_j)} \quad (3.11)$$

$$W_v(A \rightarrow N | A \rightarrow x_i) = \frac{e.w_r(A, x_i) \cdot k}{n} \cdot \frac{\sum_{i=1}^n \text{time}(x_i)}{\sum_{j=1}^k \text{time}(A_j)} \quad (3.12)$$

where $w_r(A, x_i)$ is the real weight between active item A and item x_i , n represents the number of times the item x_i is found in the integrated routes, k represents the number of times of times the item A is found in the integrated routes, $\sum_{i=1}^n \text{time}(x_i)$ represent the spent time by all site visitors on item x_i which is stored in the integrated routes, and $\sum_{j=1}^k \text{time}(A_j)$ represent the spent time by all site visitors on item A , which is also stored in the integrated routes.

If the collected data for items are ratings, rather than spent time by users (this can be true in variations of the active node technique); then we can calculate virtual weight between item N and item A via x_i , using the equation 3.13:

$$W_v(A \rightarrow N | A \rightarrow x_i) = \frac{e \cdot w_r(A, x_i) \cdot k}{n} \cdot \frac{\sum_{i=1}^n R(x_i)}{\sum_{j=1}^k R(A_j)} \quad (3.13)$$

Where $W_v(N | A \rightarrow x_i)$ is the virtual weight between the active node A and the new added item N via x_i , and $e=1$ if the hyperlink (semantic relationship) is found between items x_i and N , else $e=0$. On other hand, $w_r(A, x_i)$ refers to the number of times items A and x_i appear (e.g. are purchased) together, while k refers to the number of users who have rated item A , and n refers to the number of users who have rated item x_i . $\sum_{i=1}^n R(x_i)$ represents the total of ratings by all users for item x , while $\sum_{j=1}^k R(A_j)$ represent the total of ratings done by all users for item A .

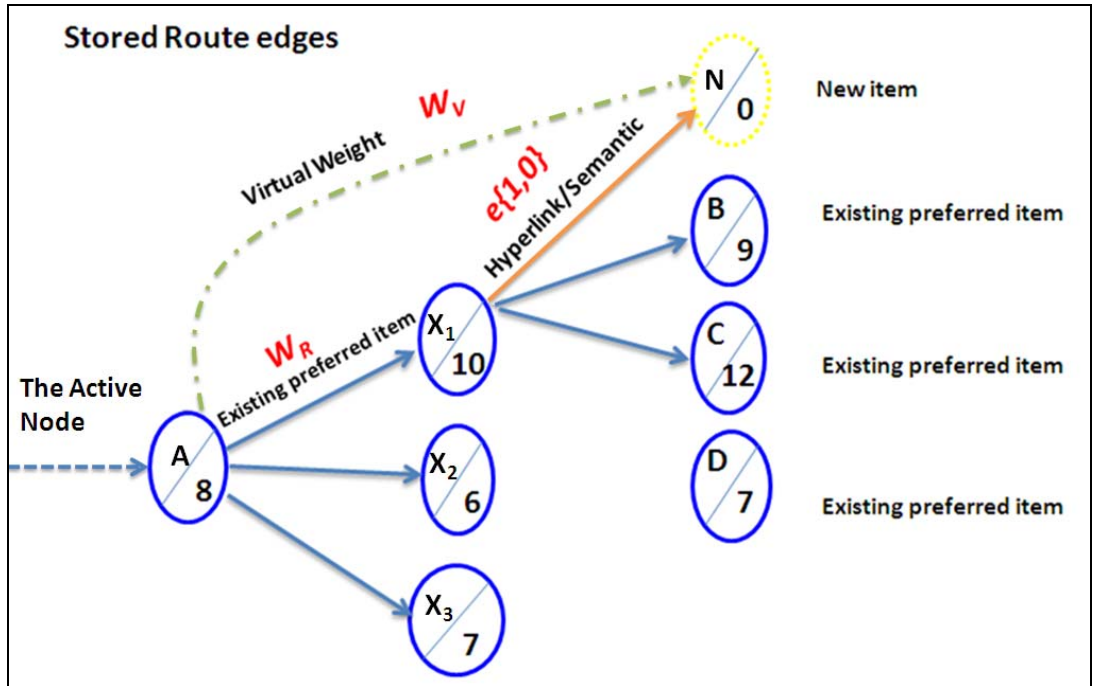


Figure 3.14: Generating a virtual link to a new added item.

We can predict an average virtual weight for any new item by considering all hyperlink relationships using the following formula 3.14

$$\overline{W}_v(A \rightarrow N) = \frac{\sum_{i=1}^n W_v(A \rightarrow N \mid A \rightarrow x_i)}{n} \quad (3.14)$$

The virtual weight is not arbitrary; its value is based on users' browsing preferences and depends on the real weights and impacts of items that are related to the new item.

3.2.5 The recommendation process

Two types of recommendation can be generated for new users based on the integrated routes profile and the users' online maximal forward session. These are *batch* recommendations and *node* recommendations. In node recommendation, the system will create a set of recommendations based on nodes that are directly linked from the current active node. In batch recommendation, however, the set of recommendations will be generated using the top N highly weighted nodes further on in the integrated route that currently matches the user session (this may include many paths).

A) Node recommendation rules

The primary rule used for generating node recommendations is represented by equation (3.15).

$$Find(x_i \mid A \xrightarrow{e_j} x_i \subset IR_j) \quad (3.15)$$

In section 3.15, A refers to the user's active node, and x_i refers to any item that can be reached directly (i.e. via a hyperlink) from A , and is also stored in an integrated route IR_j sequentially immediately after A . All such x_i are candidates for recommendations, and only the top n items are selected for recommendation. For example, if we have the following four routes in the integrated routes profile as show in figure 3.15, and we detect the current user maximal online session as $A \rightarrow B \rightarrow C \rightarrow D$.

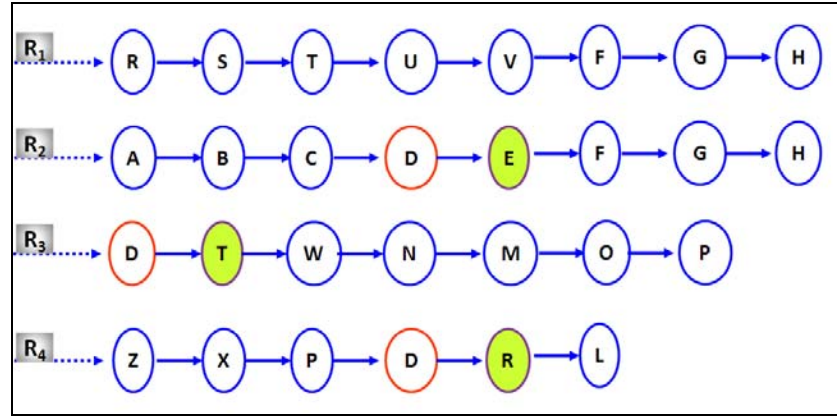


Figure 3.15: Different stored routes.

then, using node recommendation, since the active node is D , the system will consider nodes E , T , R as candidate nodes for recommendation but their associated relative weights will determine which one(s) will be selected for recommendation and which will not.

B) Batch recommendation rules

In batch recommendation, we collect candidate items for recommendation from further along the integrated routes, but at the same time require a more extensive match with the user's current session. Rule (3.16) is used to generate candidate items for batch recommendation.

$$Find(x_i \mid CMP \subset IR_j) \quad (3.16)$$

Where CMP refers to the user's current online maximal path, and IR_j refer to stored integrated routes that contain CMP as a subsequence. Again, the top n candidates will be selected for recommendation.

Consider the example shown in Figure 3.16, in which a new user's online maximal path is $A \rightarrow B \rightarrow C \rightarrow D$, and we have two integrated routes. Only the second route will be considered because it is a super-sequence of the user's online maximal route, and then our candidate nodes for batch recommendation are E , F , G , H , which represent the expected browsing targets of the user, and the system will select, for example E and H because of their high relative weights.

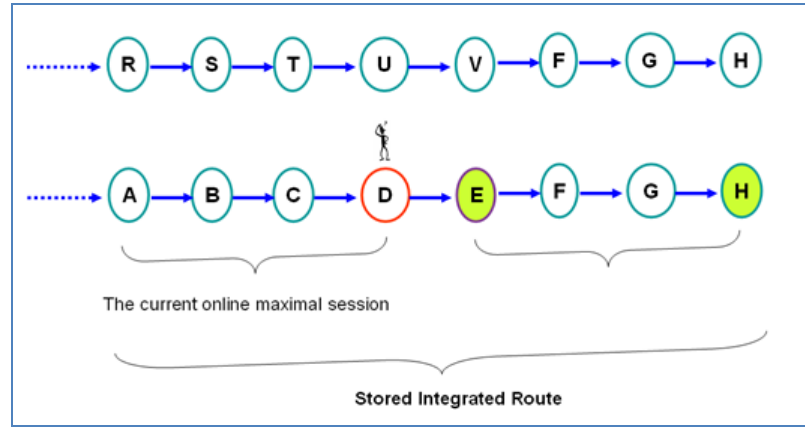


Figure 3.16: A simple illustration of batch recommendation.

C) New items recommendation rules

We consider newly added items, as well as existing items on the website that have never been visited, as being ‘new’ items. Initially, new items have impact values set to zero. All selected candidates for recommendation, in both node and batch recommendation, are checked to see if they have a direct link to any new item (any node with impact equal to zero), and then by implementing the virtual weight equation (3.12), we can select the top N items for recommendation, which may include some new items. Returning again to figure 3.16, we considered nodes E and H because of their high relative weight. We now use $D \rightarrow E$ and $G \rightarrow H$ to calculate a virtual weight for any new items linked from E or from H respectively.

We can summarize the steps used for collecting candidate items for recommendation as follows:

1. Initialize empty sets for collecting candidate recommendation items;
2. Read the current user’s online maximal path;
3. Find the integrated routes that are super-sequences of the current user’s maximal online path.
4. Capture all candidate items for node recommendations, and calculate related new items’ virtual weights. Also collect candidates for batch recommendation and calculate related new items’ virtual weights.

5. If the batch recommendation set is not empty, then the recommendation set will contain the batch recommendations and the associated new items. Otherwise it will contain the node recommendations and the associated new items.
6. The top n weighted items will be provided as recommended items in association with the top k new items.

D) The recommendation algorithm

Figure 3.17 shows pseudocode for the algorithm used for collecting candidate items for recommendation; this algorithm depends on the online user maximal path and the integrated routes profile as inputs in order to generate the recommendation set.

```

1. Begin
2. initialize  $NR=\{\}$  node recommendation subset
3. initialize  $BR=\{\}$  batch recommendation subset
4. initialize  $NW=\{\}$  zero weight nodes (new added items)
5. initialize  $RS=\{\}$  empty recommendation set
6. Read Current Maximal Path “CMP”
7. Read last node “X” in the online maximal session (the requested active node)
8. While not end of IRP // not end of integrated route profile
    Read route  $R_j$  // read first rout in the integrated route profile
    If  $CMP \subset R_j$  Then
        Let T be the top n weighted nodes in  $R_j$ 
         $BR \leftarrow BR \cup T$  // top n weighted nodes
         $NR \leftarrow NR \cup T$  // top n weighted nodes
         $NW = \{l_1, l_2, \dots, l_k\}$ , such that there is a link from x to  $l_i$ , and each  $l_i$  has zero weight
    End if
Loop
9. If BR not empty then
     $RS = BR \cup NW$ 
Else
     $RS = NR \cup NW$ 
End if
10. Display RS
11. End

```

Figure 3.17: Recommendation algorithm.

Where,

CMP: Current User Online Maximal Path

X: Last Node Name in the Maximal Online Session

R: An Integrated Route
IRP: The Integrated Routes Profile
RS: Recommendation set

E) Switching between node and batch recommendation

Our method gives high flexibility to switching between node and batch recommendation. If the node recommendation set is empty, then the system automatically switches to batch recommendation. In addition, if the system detects a recommendation set with too many nodes, only the top N weighted nodes can be recommended to the user. New item(s) find a chance of being in the recommendation set using the suggested method as indicated earlier.

3.3 Evaluation Methods

In the next chapter, we describe experiments aimed at evaluating the active node technique and also to compare it with selected alternative techniques. In this section we describe the metrics that are used in the evaluation experiments. In short, we measure the **novelty, precision, and coverage** of generated recommendation sets. In node recommendation, the target set represents the items in integrated routes that have a link from the user's active node. In batch recommendation, the target sets represent items (not visited yet by the current user) stored later in the integrated routes that contain the current user maximal path as a sub-sequence. Novelty reflects the ability of the system to provide unknown or unexpected items in the displayed recommendations. Coverage reflects the extent to which the system draws its recommendations from the whole target set – if the same small set of items are recommended repeatedly, for example, this shows poor coverage. Finally, precision tries to measure how much of the recommended items are appropriate recommendations for the user. The following subsections provide more description of each evaluation metric.

3.3.1 Novelty level

It is important to define novelty in recommendation systems; when these systems recommend items that the user was not aware of, then the system provides novel items. Providing repeated items is meaningless for users, and hence the system should make the user aware of unknown items. We calculated the novelty of generated recommendations based on the following steps.

1. Collect generated recommendations for each user
2. Find repeated recommended items between different recommendations
3. Find novelty percentage using formula 3.17.

If the system generates recommendation sets $R_1, R_2, R_3, \dots, R_n$. The total number of distinct recommended items is $\left| \bigcup_{i=1, \dots, k} R_i \right|$, and the total number of recommended items including repeats is $\sum_{i=1}^k |R_i|$. Then the level of novelty can be calculated as follows:

$$Novelty = \frac{\left| \bigcup_{i=1, \dots, k} R_i \right|}{\sum_{i=1}^k |R_i|} \quad (3.17)$$

We calculated the level of novelty for the active node method as well as for the other alternative methods, as shown in chapter four.

3.3.2 Precision and coverage levels.

In this section, we demonstrate how we calculate coverage and precision of provided recommendations sets.

A) Node recommendation evaluation methods

In node recommendation, we will use the generated node recommendations to calculate levels of coverage and precision, compared against the current active node target items stored in the integrated routes. The following two figures 3.18 and 3.19 show the expected candidates for active node D . In addition, evaluate diversity of generated recommendations for different users in different online maximal sessions, as well as how the system can provide up-to-date recommendations in the same active node.

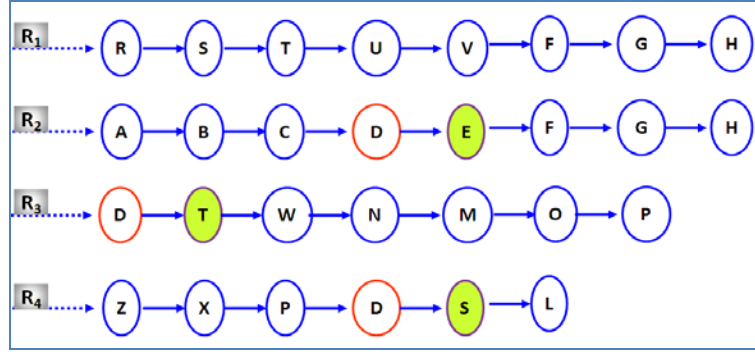


Figure 3.18: Different routes used for node recommendation evaluation.

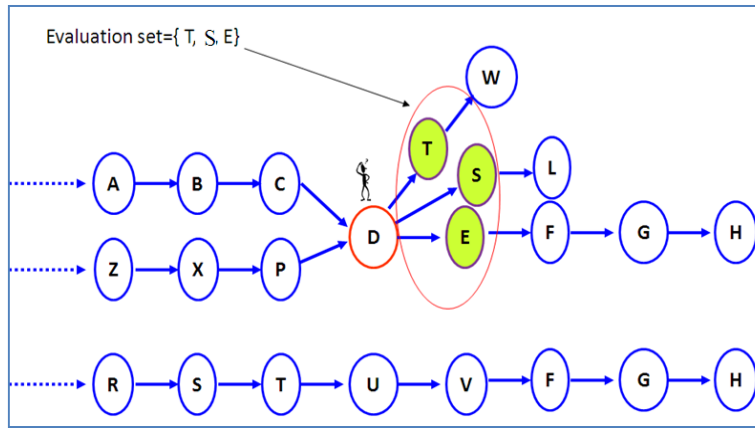


Figure 3.19: Different items used for node recommendation evaluation.

Precision and coverage levels in node recommendation

A level of coverage is used to measure percentage of items provided as node recommendation and appear in the target sets to the total number of items in the target set (selected items by user during a training phase, which did not involve recommendations to the users, are used as the target set). While precision level measure percentage of items provided as node recommendation and appear in the target set to the total number of recommended items in recommendation set. Let R the set of generated recommendations (in node recommendation mode) where $R = \{r_1, r_2, r_3, \dots, r_n\}$. Let TS the set of target set and $TS = \{x_1, x_2, x_3, \dots, x_k\}$, where target set TS contains target item x_i that has a link from the active node. Then we can calculate the coverage and precision as follows:

$$Coverage = \frac{\sum_{i=1}^n |R_i \cap TS_i|}{\sum_{j=1}^k |TS_j|} \quad (3.18)$$

Where $\sum_{i=1}^n |R_i \cap TS_i|$ represents number of items found in both recommendation set and target set. While $\sum_{j=1}^k |TS_j|$ represents the total number of items in the target set.

$$Precision = \frac{\sum_{i=1}^n |R_i \cap TS_i|}{\sum_{j=1}^k |R_j|} \quad (3.19)$$

Where $\sum_{i=1}^n |R_i \cap TS_i|$ represents number of items found in both recommendation set and target set. While $\sum_{j=1}^k |R_j|$ represents the total number of items in the all recommendation sets.

B) Batch recommendation evaluation methods

Batch recommendation evaluation will depend on the stored integrated routes, and batch recommendations will generally be a superset of node recommendations. Figures 3.20 and 3.21 show a user maximal session, the expected target set (shown in figure 3.20), and the used evaluation set to select candidates for recommendation as shown by figure 3.21.

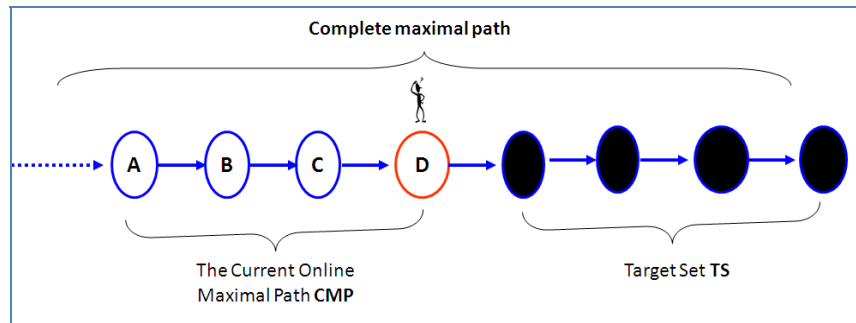


Figure 3.20: Target set TS used for batch recommendation evaluation (these are items that were selected by users in a training phase – see Chapter 4 – after visiting node D).

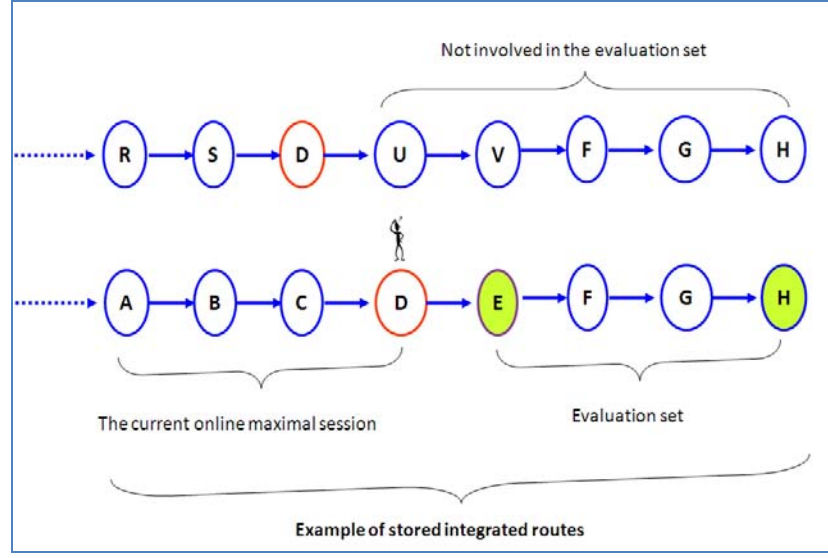


Figure 3.21: Evaluation set for batch recommendation.

In batch recommendation, we will collect the evaluation set from the created integrated routes and compare it to the user's target set.

Precision and coverage levels in batch recommendation

Level of coverage measures the amount of items provided in the batch recommendation set that appear in the target sets, as a percentage of the total number of items in the target sets (stored integrated routes; which the current user maximal path is subset of, are used as target sets). Let R the set of generated recommendations (in batch recommendation mode) where $R = \{r_1, r_2, r_3, \dots, r_n\}$. Let TS the set of target sets $TS = \{ts_1, ts_2, ts_3, \dots, ts_k\}$, where the target set ts_i contains all expected browsing target items stored in the integrated routes i . where the current user maximal path is a subset of the integrated routes i . Then we can calculate the coverage and precision as follows:

$$Coverage = \frac{\sum_{i=1}^n |R_i \cap TS_i|}{\sum_{j=1}^k |TS_j|} \quad (3.20)$$

Where $\sum_{i=1}^n |R_i \cap TS_i|$ represents number of items found in both recommendation sets and

target sets. While $\sum_{j=1}^k |TS_j|$ represents the total number of items in all target sets.

While precision measures the participation level of each recommendation set in its associated target set, the accuracy level is calculated using equation (3.21).

$$Precision = \frac{\sum_{i=1}^n |R_i| \cdot \frac{|R_i \cap TS_i|}{|TS_i|}}{\sum_{j=1}^k |R_j|} \quad (3.21)$$

Where $|R_i|$ represents number of items in the recommendation set i . $|R_i \cap TS_i|$ the number of item found in both recommendation set number i and target set number i . $|TS_i|$ the total number of items in the target set i . and $\sum_{j=1}^k |R_j|$ the total number of items provided to user in all generated recommendation sets.

C) New items evaluation methods

As discussed before, all new added items, as well as old items never visited before, are considered as new items; all these items are initialized with zero impact. Let NT the set of new items involved in the training phase (see Chapter 4), and let T be the set of such items that are selected by users during their browsing. Then we can calculate a coverage level for new items simply as in equation (3.22),

$$Coverage = \frac{|T|}{|NT|} \quad (3.22)$$

In other words, the coverage level for new items shows the proportion of new items selected from the whole number of new items input in the training phases. The **precision level** for new items measures the proportion of new items involved in the target set that have been involved in recommendation sets. Let W be the set of new items involved in generated recommendation sets, where $W = \{w_1, w_2, w_3, \dots, w_n\}$ and w_i is a set of new items involved in recommendation set i . Let T be the set of new nodes selected by users through their browsing. All new added items to the website begin with zero impact value, and then when users select it, then its impact value increase, and as clue it will appear in the integrated routes. In the context, it reflects that users trust some of the suggested new items $|T|$ and hence they select it. Then we calculate precision for the new added items as follows:

$$Precision = \frac{|T|}{\sum_{i=1}^n |w_i|} \quad (3.23)$$

Where $|T|$ refers to the total number of trusted items, and $\sum_{i=1}^n |w_i|$ represents the total number of new items involved in generated recommendation sets.

3.4 Summary

In order to address the cold start problem in a way that considers privacy concerns, we suggest the active node technique (ANT) as a method to collect users' abstract click streams, as a way to lead to appropriate and useful recommendations for any user. Collected abstract click streams are used to create abstract integrated routes, which in turn will be used to generate the delivered recommendation sets to site visitors regardless of their personal data. We showed how to collect abstract loop-less sessions (maximal sessions) that show the abstract users' preferences, and we showed our approach to evaluate and store selected maximal sessions, as well as the approach to integrating smaller sessions into larger ones for a more compact representation. The integrated routes profile (IRP) stores integrated routes, which each represent a maximally sized abstract loop-less route, which aggregates visits by abstract users on the specific web site. These routes are used to find candidates for recommendations. We also presented the evaluation metrics (novelty, coverage, and precision) that we will use to evaluate our method and compare it to alternative methods as shown in the next chapter.

Chapter 4

A Collaborative Filtering System Based on the Active Node Technique

4.1 Introduction.

Web recommendation and personalization systems aim to help users to find what they are looking for in less time and with high accuracy, by suggesting items or information from the huge amount of information available. Such systems are now implemented in many different areas such as E-commerce, E-learning, E-business, etc. However, web personal recommendation systems face many challenges; one of these challenges is known as the cold start problem. There are various approaches that have been suggested for solving the cold-start problem; as indicated in chapter three. Some techniques depend on **demographical** data such as the *Triadic Aspect Model* suggested by (Lam et al., 2008b), where they used users' information (such as age, gender, and job) to find initial similarity between users. Some systems depend on the **stereotype** image in order to create initial ratings, such as *Naïve Filterbots*, suggested by (Park et al., 2006). In Park et al.'s approach, the filterbot algorithm injects pseudo users or bots into the system; these bots rate items algorithmically according to attributes of items or users, for example according to who acted in a movie, or according to the average of some users demographic. Ratings generated by the bots are injected into the user-item matrix along with actual user ratings. Then standard CF algorithms are applied to generate recommendations. Park and Chu, 2009 collected users' demographical information (e.g. age, gender) to generate initial profiles for users and hence each user is represented by a set of features, while they also represent each item by a set of features; then they find affinities between users' features and items' features. Meanwhile, some other systems depend on item-based similarity to generate recommendations, as explained in chapter three.

The rest of this chapter is organized as follows. In section 4.2, we describe a practical implementation of the approach described in chapter three, we provide a practical implementation of data collection and cleaning processes associated with the technique, we explain how to create integrated routes, and then how to generate recommendations. In section 4.3, we describe three alternative methods (*Naïve Filterbots*, *Triadic Aspect Model*, and *item-based model*). In section 4.4, we describe our experiments (data sets, experiment design, and method of evaluation), and present our experimental results. In section 4.5, we provide our summary and conclusions.

4.2 Implementation of a system based on the active node method.

In this section we describe the implementation in broad terms using a system model approach; this is based on a graphical representation that describes the problem to be solved and the system that is to be developed to achieve specific goal(s) or objectives (Delaney and Brown, 2002). A System model is used for system analysis purposes to understand the different prospective parts of the system, which we demonstrate in the following subsections.

4.2.1 Context of the proposed system.

A Context diagram is useful to view how the system will work with its subsystems. Figure 4.1 shows the context diagram for our system.

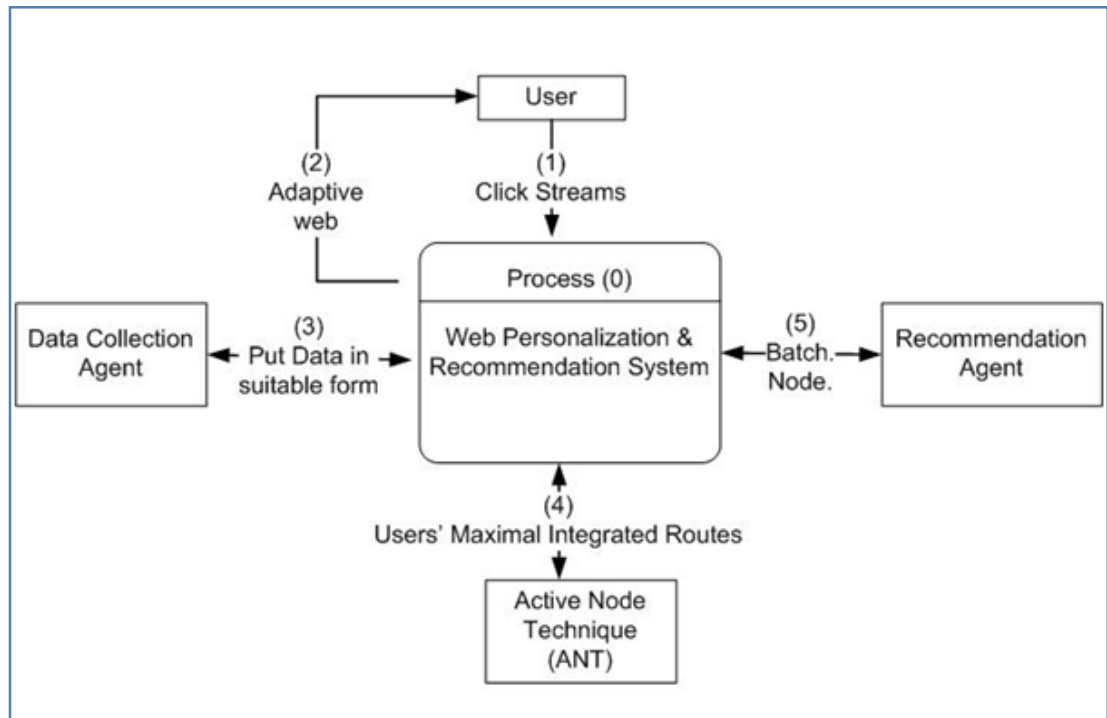


Figure 4.1: Context diagram for web personal recommendation system.

As shown in figure 4.1, the system's main function is to provide recommendations to web visitors, and we have three subsystems (modules) each one having its own inputs, processing, and outputs. The first module is a data collection agent that collects users' clicks streams and filters them to find valuable information, and transforms these data into a suitable form (maximal sequential sessions) that will serve further processing. The second module is the active node method, that is used to discover users' significant integrated routes. The third

module is the recommendation module, which uses the discovered integrated routes profile to generate recommendations. Two types of recommendations can be provided to users: batch recommendations (nodes that may be of interest to the user, from anywhere on the web site) and/or node recommendations (nodes of interest chosen only from the nodes directly linked from the user's current page). The inputs to the data collection module are the users' click streams, while the outputs are the representations of these as sequential maximal sessions that are then input to the active node technique. The active node technique module then outputs the integrated routes profile, that then becomes the input to the recommendation agent; the outputs of the recommendation module are recommendations sets provided to users.

The suggested system follows a familiar and general web personalization and recommendation architecture. This architecture consists of three stages. The first is the data collection stage; where we can collect data online or use log files, and transfer it into the database. The second stage, generally called pattern discovery, is where we will use the active node technique (ANT) to discover integrated routes of users' preferences. The third stage is recommendation, whether it is node recommendation or batch recommendation. Figure 4.2 shows the general structure of the suggested system phases using online data collection.

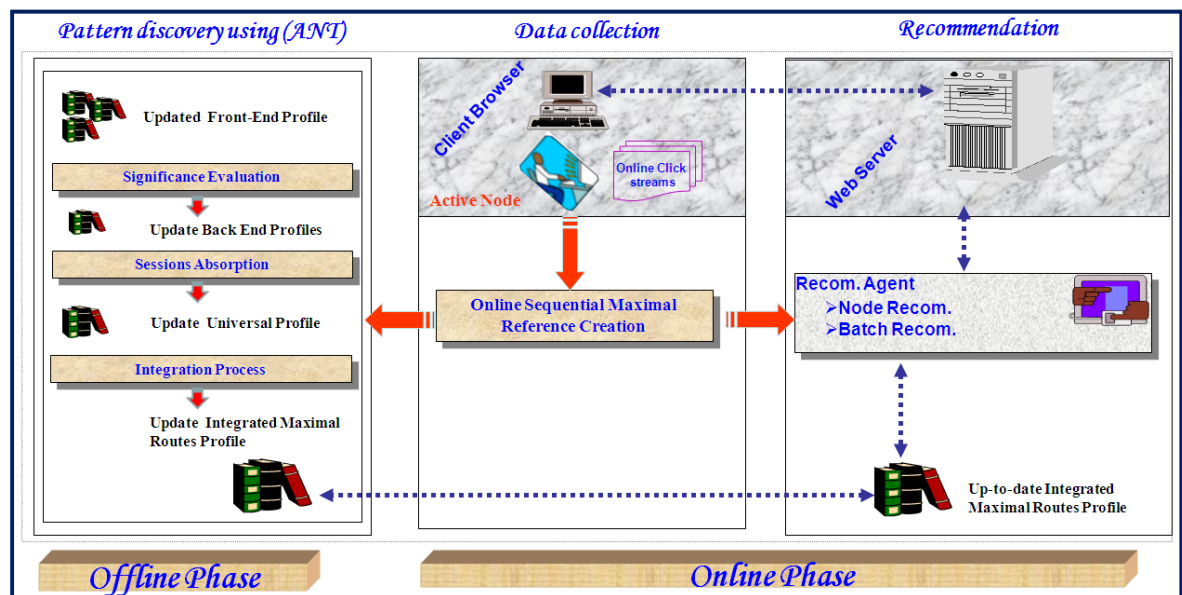


Figure 4.2: General model for collaborative system based on the active node technique.

In the following sections, we will demonstrate more explanation of different model phases.

4.2.2 Data collection and preparation.

The inputs of this phase may include the web server logs or registration files (if we will use log historical files), or online data collected from users' click-streams. The outputs are the users' maximal sessions. The goal of this phase is to remove irrelevant data that will not serve the further processing of the active node technique. Figure 4.3 shows that this phase consist of three processes, which are data collection and cleaning, maximal session creation, and creation and/or updating of the front-end profile.

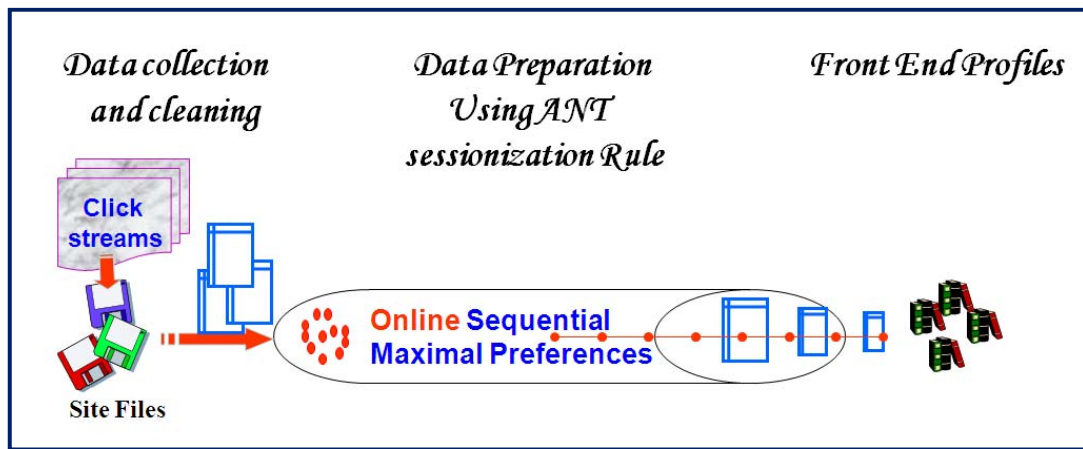


Figure 4.3: Data collection and preparation phase.

A) Data collection and cleaning.

The usage data can be collected using historical click streams stored in log files, these log files can be collected from the server side, client side, and/or proxy servers, each of which differ in terms of typical data formats. Such log files must be cleaned and converted into data abstractions suitable for further processing. Server log files provide a list of the page requests (or selected items) made to a given web server; a request is characterized by the IP address of the requested machine, the date and time of the request, the URL of the requested page, DNS, bytes, status, method, and other items related to the log file format. These log files store all events related to the web site, hence containing much that is irrelevant or not desired for the active node technique. Figure 4.4 illustrates the server log file format.

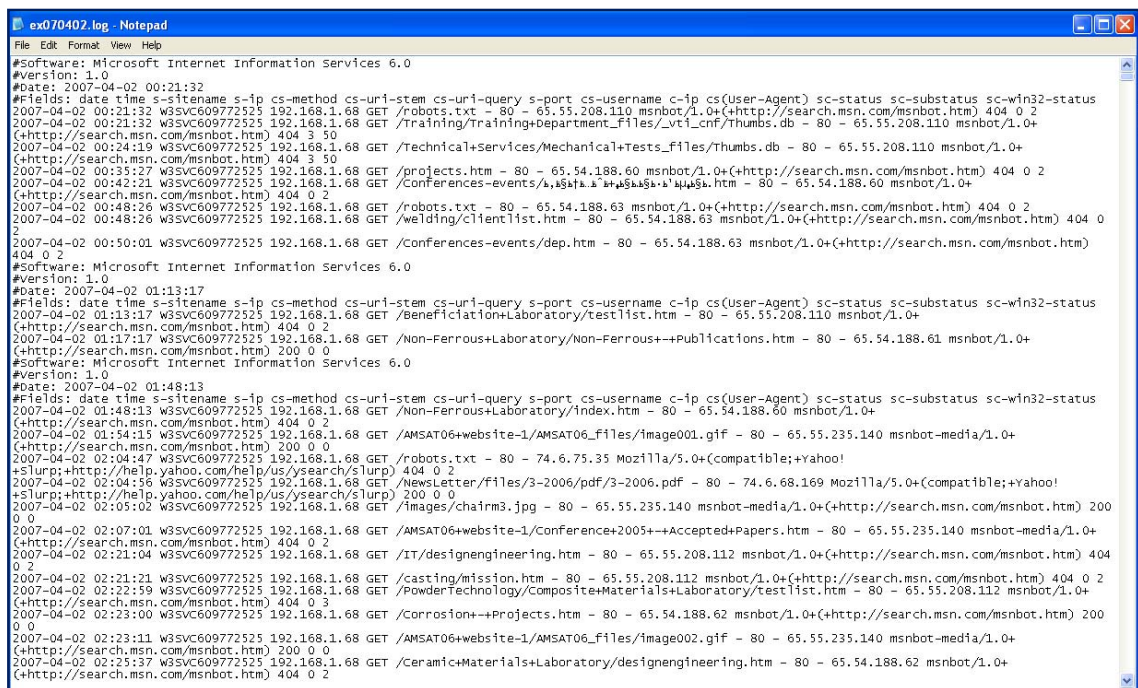
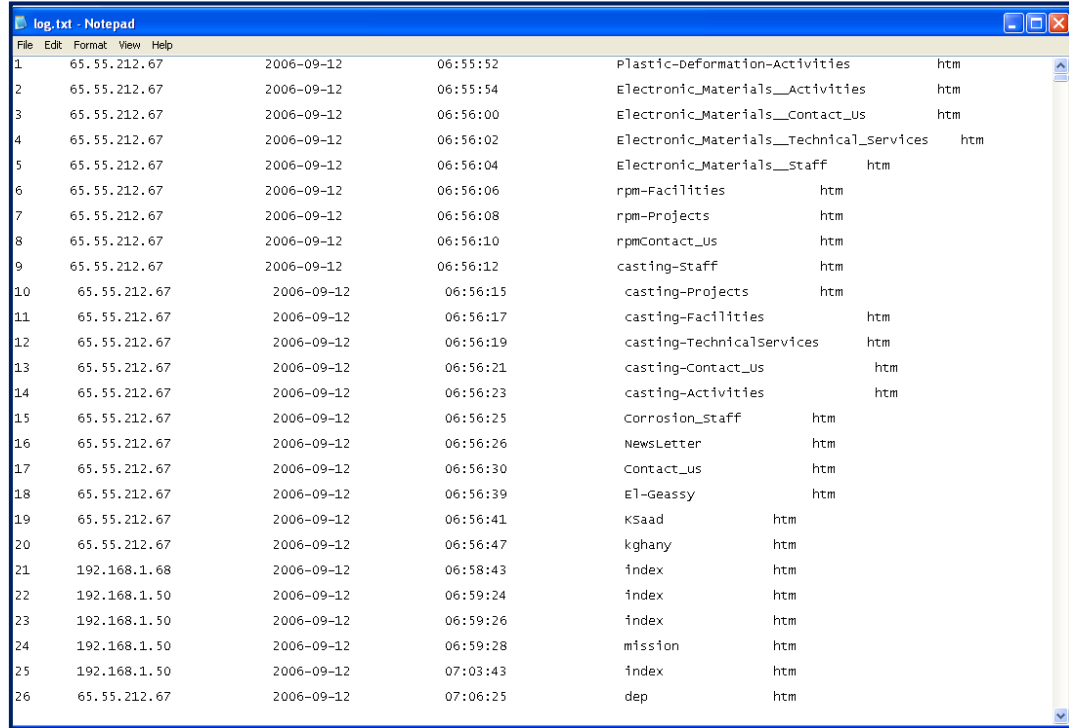


Figure 4.4: Server log file raw data format.

Data cleaning involves removing all irrelevant and erroneous items and capturing only useful data. The discovered association or reported statistics are only useful if the data represented in the server log gives an accurate picture of the user accesses to the web site. The HTTP protocol requires a separate connection for every file requested from the web server. Therefore, a user's request to view a particular page often results in several log entries since graphics and scripts are downloaded in addition to the HTML file. In most cases, only the log entry of the HTML file, ASP files, and Xsp files request are relevant and should be kept for the user session file. Generally, a user does not explicitly use all of the graphics that are on the web page, but they are automatically downloaded due to the HTML tags.

The main aim of web usage mining is to get an accurate picture of the user behavior; it does not make sense to include file requests that the user did not explicitly request. Elimination of the items deemed irrelevant can reasonably be accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as GIF, JPEG, TXT, PDF, JPG, etc. can be removed. In addition, the common scripts such as "count.cgi" can also be removed. This task is very important to the personalization and

recommendation process because all next tasks depend on the outputs of this stage. Only the records that will serve the purpose of the personalization will be extracted from log files, as displayed in the cleaned log file shown by figure 4.4.



Line	IP	Date	Time	Page	Ext
1	65.55.212.67	2006-09-12	06:55:52	Plastic-Deformation-Activities	htm
2	65.55.212.67	2006-09-12	06:55:54	Electronic_Materials__Activities	htm
3	65.55.212.67	2006-09-12	06:56:00	Electronic_Materials__Contact_Us	htm
4	65.55.212.67	2006-09-12	06:56:02	Electronic_Materials__Technical_Services	htm
5	65.55.212.67	2006-09-12	06:56:04	Electronic_Materials__Staff	htm
6	65.55.212.67	2006-09-12	06:56:06	rpm-Facilities	htm
7	65.55.212.67	2006-09-12	06:56:08	rpm-Projects	htm
8	65.55.212.67	2006-09-12	06:56:10	rpmContact_Us	htm
9	65.55.212.67	2006-09-12	06:56:12	casting-Staff	htm
10	65.55.212.67	2006-09-12	06:56:15	casting-Projects	htm
11	65.55.212.67	2006-09-12	06:56:17	casting-Facilities	htm
12	65.55.212.67	2006-09-12	06:56:19	casting-TechnicalServices	htm
13	65.55.212.67	2006-09-12	06:56:21	casting-Contact_Us	htm
14	65.55.212.67	2006-09-12	06:56:23	casting-Activities	htm
15	65.55.212.67	2006-09-12	06:56:25	Corrosion_Staff	htm
16	65.55.212.67	2006-09-12	06:56:26	NewsLetter	htm
17	65.55.212.67	2006-09-12	06:56:30	Contact_us	htm
18	65.55.212.67	2006-09-12	06:56:39	EL-Geassy	htm
19	65.55.212.67	2006-09-12	06:56:41	KSaad	htm
20	65.55.212.67	2006-09-12	06:56:47	kghany	htm
21	192.168.1.68	2006-09-12	06:58:43	index	htm
22	192.168.1.50	2006-09-12	06:59:24	index	htm
23	192.168.1.50	2006-09-12	06:59:26	index	htm
24	192.168.1.50	2006-09-12	06:59:28	mission	htm
25	192.168.1.50	2006-09-12	07:03:43	index	htm
26	65.55.212.67	2006-09-12	07:06:25	dep	htm

Figure 4.5: A Cleaned Log file.

From this information, it is possible to make statistical analysis of the site visitors; we collected log files for a period of six weeks from <http://www.cmr.di.sci.eg> and performed a statistical analysis as shown by the historical log analysis report in appendix A. In addition, it is possible to reconstruct the users' navigation click streams into the form required for the next stages. The usage data also can be collected online using users' click-streams; therefore, we can collect usage data in the format required for further processing without the need for log files and hence without the need for additional data cleaning. In addition, generation of maximal forward sessions while users are online will help in estimating the duration for the last page in a session (which is problematic in recommendation systems that depend on the collected data from log files). We calculate the last node time duration as the difference between that page's start time and the time recorded when terminating the maximal session function.

As the user moves from one active node to another, the system will collect required data such as the requested page (active node), and time spent per page. When an online session has reached maximal length, the system will store it in the front-end profile, and at the same time the system sends the created maximal session to the recommendation agent for generating recommendation sets that should then be displayed to the user on the requested page.

B) Data Preparation

Users' online click streams will not be useful until put in the form required for the next processing steps, therefore the collected data should be sessionized into maximal forward session formats using the suggested active node rules and the algorithm for creating maximal sessions, as explained in chapter three .

Sequential maximal sessions creation

The system will collect the user's click streams and, whenever a loop is created or the session length reaches ten nodes, or the user terminates the session, then the system will create a sequential maximal session and restart a new maximal session (using suggested rule and algorithm for generate sequential maximal sessions discussed in chapter three). Table 4.1 shows a sample of maximal forward sessions created during operation of the implemented system.

node	starttime	endtime	duration
Begin			
action to stem job losses.aspx	11:46:00 PM	11:46:06 PM	6
Consumer.aspx	11:46:06 PM	11:46:15 PM	9
Finance.aspx	11:46:15 PM	11:46:29 PM	14
Oil Market.aspx	11:46:29 PM	11:46:39 PM	10
ECB At Loss.aspx	11:46:39 PM	11:46:43 PM	4
End	11:46:43 PM		43
Begin			
action to stem job losses.aspx	11:52:54 PM	11:53:00 PM	6
Honeymoon.aspx	11:53:00 PM	11:53:08 PM	8
Gene Study.aspx	11:53:08 PM	11:53:16 PM	8
Finance.aspx	11:53:16 PM	11:53:25 PM	9
Food Safety.aspx	11:53:25 PM	11:53:34 PM	9
End	11:53:34 PM		40
Begin			
action to stem job losses.aspx	11:55:35 PM	11:55:51 PM	16
Politics US Economy.aspx	11:55:51 PM	11:55:57 PM	6
Nato and Russia.aspx	11:55:57 PM	11:56:03 PM	6
Iraq Voted.aspx	11:56:03 PM	11:56:42 PM	39
US Foreign Policy.aspx	11:56:42 PM	11:56:51 PM	9
End	11:56:51 PM		76

Table 4.1: Sample of maximal forward sessions created by the implemented system.

During the creation of maximal session, collected click streams will be sent sequentially to the recommendation agent. In context, a specific user online session may match many stored integrated routes, and only highly weighted items will be selected for recommendation. The recommendation engine will be able to create recommendation sets based on changes in his/her online session, and hence, recommendation sets will change from node to another.

Front-end profile creation

As soon as a user enters the web site, our system will collect his maximal session(s) and store it (them) in the front-end profile, where the system will temporarily store users' paths (maximal session pages that a specific user accesses during his/her current visit) and the time spent per page. The front-end profile is used later for further processing by the active node technique to determine the significance level for each session. As soon as a user leaves the web site, all collected data about his/her online maximal visited sessions will be evaluated by the significance function, and then significant maximal sessions will move to the back-end

profile, while insignificant maximal sessions will be removed and the front-end profile created for this user will be deleted.

4.2.3 Pattern discovery phase using ANT

The sequence of maximal sessions output from the previous phase represents the inputs to the pattern discovery phase. Items' impact values will be calculate, and the absorption process will be invoked, leading to recalculation of impact values and weights in sessions that have absorbed sub-sessions. Figure 4.6 shows the active node online and offline phases, the online phases, which have been discussed and explained already; in the following section, we will provide descriptions and explanations of the offline phase processes in our implementation.

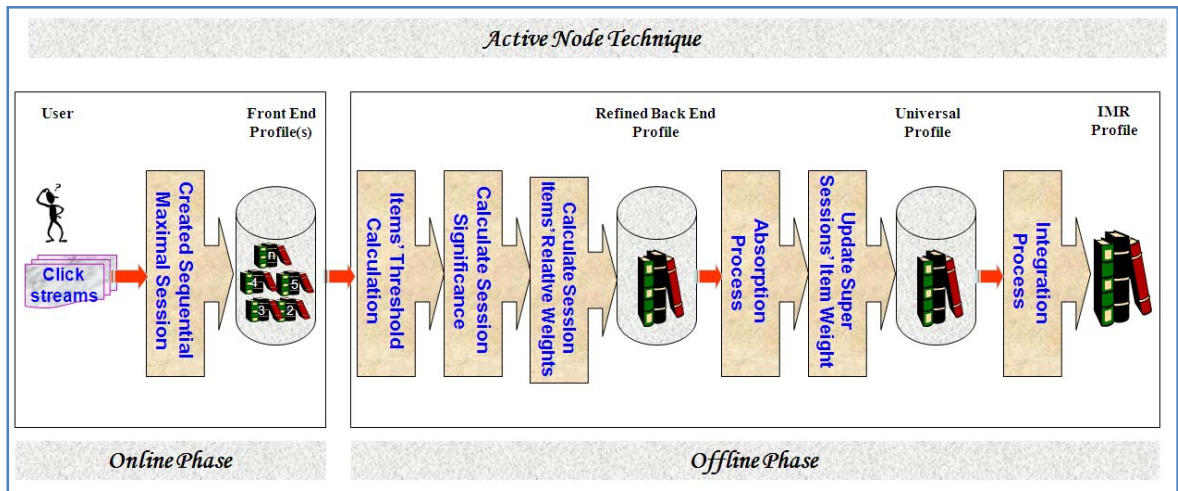


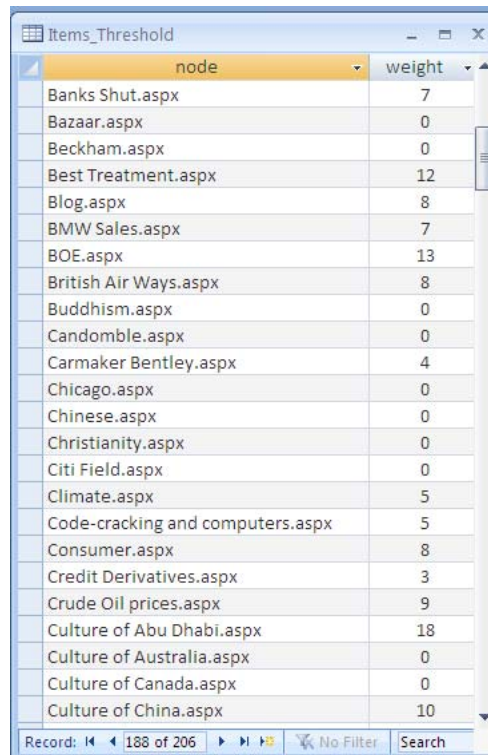
Figure 4.6: Active node online and offline phases.

A) Evaluating the significance of maximal sessions

Not all maximal sessions are deemed valuable; only 'significant' sessions will be selected for further processing while the others will be removed. Evaluating a session's significance requires the calculation of the impact values for all items.

Items impact calculation

As we indicated in chapter three , all items' impact values are initialized to zero. Users move from item to item during their visit on the website, and the time spent by the user on each item is stored in association with the maximal session data structure. Using these time durations, we can calculate the impacts of items using the relevant equations in chapter three. The impact value of a specific item represents the average time spent by all site visitors on that item. Table 4.2 shows some calculated impact values.



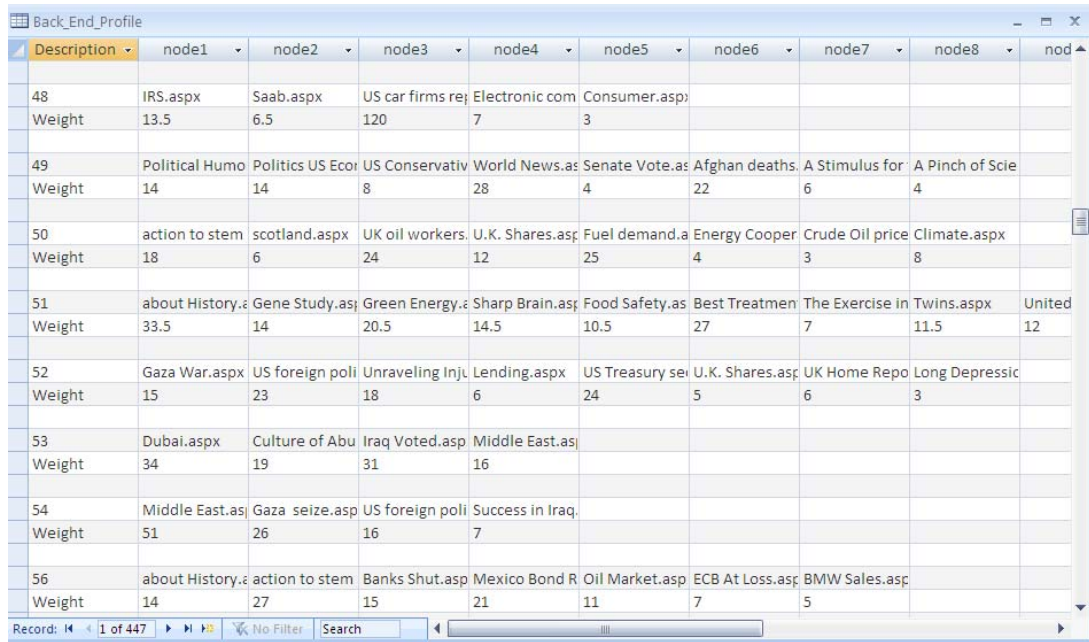
node	weight
Banks Shut.aspx	7
Bazaar.aspx	0
Beckham.aspx	0
Best Treatment.aspx	12
Blog.aspx	8
BMW Sales.aspx	7
BOE.aspx	13
British Air Ways.aspx	8
Buddhism.aspx	0
Candomble.aspx	0
Carmaker Bentley.aspx	4
Chicago.aspx	0
Chinese.aspx	0
Christianity.aspx	0
Citi Field.aspx	0
Climate.aspx	5
Code-cracking and computers.aspx	5
Consumer.aspx	8
Credit Derivatives.aspx	3
Crude Oil prices.aspx	9
Culture of Abu Dhabi.aspx	18
Culture of Australia.aspx	0
Culture of Canada.aspx	0
Culture of China.aspx	10

Table 4.2: Some calculated impact values.

As can be seen in the example of Table 4.2, several items have zero impact, which means that these items are currently new (or not yet visited during any sessions that were considered significant).

Calculating a session's significance value

Using the significance equation in chapter three , and the calculated impact values, we can eliminate non-significant sessions and only the significant ones will be selected for absorption and then for integration processes. Table 4.3 shows a back-end profile with only valuable sessions selected from the front-end profile.



Description	node1	node2	node3	node4	node5	node6	node7	node8	node9
48	IRS.aspx	Saab.aspx	US car firms re	Electronic com	Consumer.aspx				
Weight	13.5	6.5	120	7	3				
49	Political Humo	Politics US Ecor	US Conservativ	World News.as	Senate Vote.as	Afghan deaths	A Stimulus for	A Pinch of Scie	
Weight	14	14	8	28	4	22	6	4	
50	action to stem	scotland.aspx	UK oil workers	U.K. Shares.asp	Fuel demand.a	Energy Cooper	Crude Oil price	Climate.aspx	
Weight	18	6	24	12	25	4	3	8	
51	about History.e	Gene Study.as	Green Energy.e	Sharp Brain.as	Food Safety.as	Best Treatmen	The Exercise in	Twins.aspx	United
Weight	33.5	14	20.5	14.5	10.5	27	7	11.5	12
52	Gaza War.aspx	US foreign poli	Unraveling Inj	Lending.aspx	US Treasury ser	U.K. Shares.asp	UK Home Repo	Long Depressic	
Weight	15	23	18	6	24	5	6	3	
53	Dubai.aspx	Culture of Abu	Iraq Voted.asp	Middle East.as					
Weight	34	19	31	16					
54	Middle East.as	Gaza seize.asp	US foreign poli	Success in Iraq					
Weight	51	26	16	7					
56	about History.e	action to stem	Banks Shut.asp	Mexico Bond R	Oil Market.asp	ECB At Loss.asp	BMW Sales.asp		
Weight	14	27	15	21	11	7	5		

Table 4.3: Selected significant sessions.

Clearly, the number of sessions in the back-end profile will typically be lower than number of sessions in front-end profiles.

Calculating relative weights of session items

After transferring all significant sessions to the back-end profile, all sessions' items relative weights should be calculated using the items' weight equation discussed in chapter three. An item's weight reflects the importance of that item in a given session. An item that appears in several sessions will have different relative weights for each session the item appears in, but it will have only one impact value. Table 4.4 shows an example of a set of items (nodes) with different relative weights in different sessions.

Description	node1	node2	node3	node4	node5	node6	node7	node8
1	Small Business	England.aspx	Culture of Engl					
Weight	2%	5%	93%					
2	England.aspx	Middle East.aspx	Political Humo	Politics US Ecor				
Weight	33%	33%	17%	17%				
3	US Foreign Pol	US Treasury secretary.a	IBM Jobs.aspx	Long Depressic				
Weight	17%	36%	19%	28%				
4	Politics US Ecor	action to stem job losse	Senate Vote.as	Gaza War.aspx				
Weight	46%	45%	6%	3%				
5	Azeri radar.aspx	Gaza seize.aspx	action to stem					
Weight	15%	82%	3%					
7	action to stem	England.aspx	Political Humo	Gaza War.aspx	Iraq Voted.asp	Fuel demand.a		
Weight	13%	4%	15%	13%	15%	39%		
8	action to stem	England.aspx	A Stimulus for	Energy.aspx	Iraq Voted.asp			
Weight	29%	17%	29%	17%	9%			
9	US Foreign Pol	United States Culture.a						
Weight	86%	14%						

Table 4.4: Relative weights of items in different sessions.

As soon as we have calculated relative weights, we should scan the back-end profile for any duplicated sessions. If any duplication is found then the duplicates are merged together with relative weights averaged. A duplication can only happen between sequential maximal sessions of the same size and with the same sequence of items. For example, if we have the three sessions of size 4 in Table 4.5, we can remove duplicates and replace with a single maximal session with the weights in the rightmost column.

Abstract user X	Abstract user Y	Abstract user Z	Merged weights
$A \rightarrow B \rightarrow C \rightarrow D$	$A \rightarrow B \rightarrow C \rightarrow D$	$A \rightarrow B \rightarrow C \rightarrow D$	$A \rightarrow B \rightarrow C \rightarrow D$
WEIGHT	WEIGHT	WEIGHT	Average weight
A 0.2	A 0.25	A 0.10	0.18
B 0.3	B 0.40	B 0.25	0.32
C 0.3	C 0.25	C 0.15	0.23
D 0.2	D 0.10	D 0.50	0.27

Table 4.5: Duplicated significant sessions.

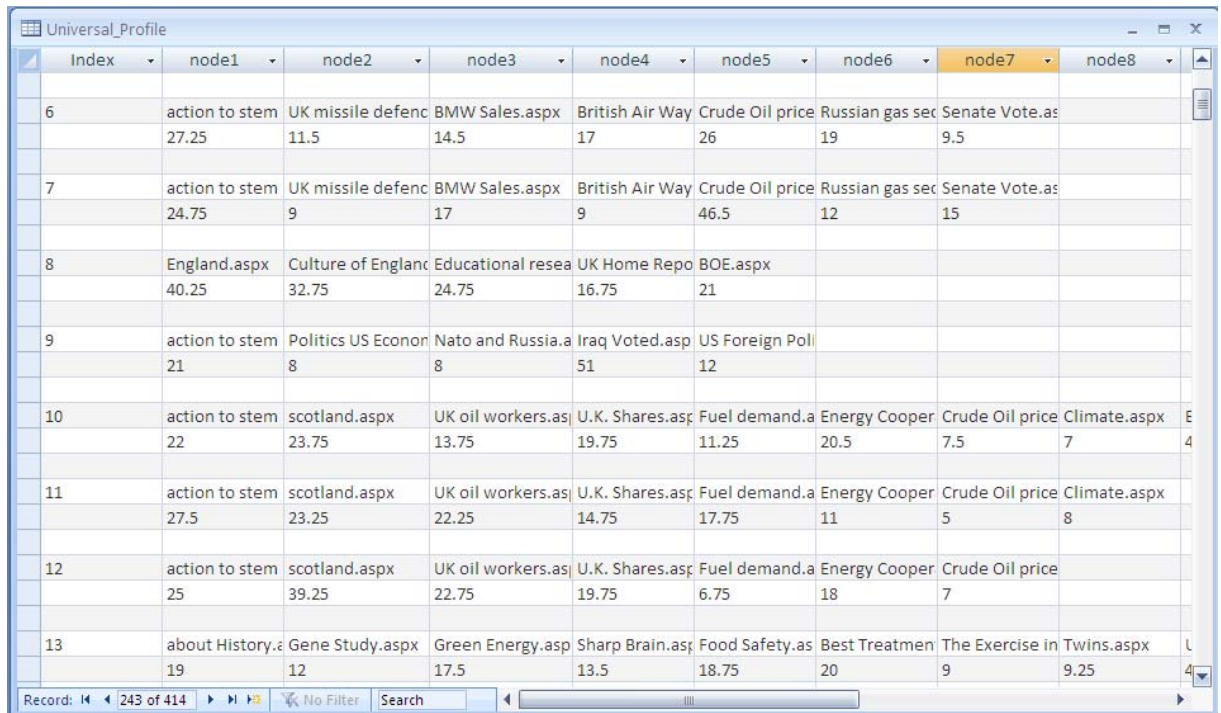
Back-end profile creation

At this stage, we have a back-end profile containing only significant sessions, with no duplicate sessions, and all relative weights correctly calculated.

B) Absorption process

As indicated in chapter three, the main goal of this process is to reduce the number of maximal sessions that remain in the back-end profile, without any significant loss of information relevant to the generation of recommendations. Therefore, we now detect any cases in which we have one session that is a strict super-sequence of another session, for each such case we retain only the ‘super-session’, after appropriately recalculating its items’ relative weights (please see section 4.2.3).

The steps of the absorption process were discussed in chapter three; they are used to detect absorption cases and then calculate false weights to update super-sessions. After finishing the absorption process, all back end profile sessions will be removed. Table 4.6 shows a sample of absorbed sessions in the universal profile.



Index	node1	node2	node3	node4	node5	node6	node7	node8
6	action to stem	UK missile defenc	BMW Sales.aspx	British Air Way	Crude Oil price	Russian gas sec	Senate Vote.as	
	27.25	11.5	14.5	17	26	19	9.5	
7	action to stem	UK missile defenc	BMW Sales.aspx	British Air Way	Crude Oil price	Russian gas sec	Senate Vote.as	
	24.75	9	17	9	46.5	12	15	
8	England.aspx	Culture of England	Educational resea	UK Home Repo	BOE.aspx			
	40.25	32.75	24.75	16.75	21			
9	action to stem	Politics US Econon	Nato and Russia.a	Iraq Voted.asp	US Foreign Poli			
	21	8	8	51	12			
10	action to stem	scotland.aspx	UK oil workers.as	U.K. Shares.asp	Fuel demand.a	Energy Cooper	Crude Oil price	Climate.aspx
	22	23.75	13.75	19.75	11.25	20.5	7.5	7
11	action to stem	scotland.aspx	UK oil workers.as	U.K. Shares.asp	Fuel demand.a	Energy Cooper	Crude Oil price	Climate.aspx
	27.5	23.25	22.25	14.75	17.75	11	5	8
12	action to stem	scotland.aspx	UK oil workers.as	U.K. Shares.asp	Fuel demand.a	Energy Cooper	Crude Oil price	
	25	39.25	22.75	19.75	6.75	18	7	
13	about History.e	Gene Study.aspx	Green Energy.asp	Sharp Brain.asp	Food Safety.as	Best Treatmen	The Exercise in	Twins.aspx
	19	12	17.5	13.5	18.75	20	9	9.25

Table 4.6: Absorbed sessions.

Updating super-session items' relative weights

Super session items' relative weight should be updated using temporary weights, as discussed in chapter three. The suggested algorithm for absorption explained in chapter three is used as well as the temporary weight method. It should be clear that the session items' relative weights, (shown in table 4.6) need not sum to one, while session items shown in the back-end profile; (displayed in table 4.4) should sum to one within a session since they reflect the relative importance of the items in a session for a specific user.

The Universal profile

All super sessions are stored in the universal profile. Each item has a specific relative weight in its super session; the items' relative weights are used to prioritize items in the candidate set for recommendations. All super sessions stored in the universal profile are used for creating integrated routes, and as soon as the integrated routes are created, all super sessions in the universal profile should be delete.

C) The Integration process

The suggested integration rule and algorithm explained in chapter three are used to create integrated routes. In this process, we aim to utilize benefits of the created super maximal sessions on the universal profile by finding larger 'elastic' maximal routes. For example if we have the following super maximal sessions on the universal profile:

$D \rightarrow J \rightarrow R \rightarrow S$ with weights 0.3, 0.4, 0.1, 0.2

$C \rightarrow F \rightarrow R \rightarrow S$ with weights 0.4, 0.1, 0.1, 0.4

$S \rightarrow H \rightarrow Z$ with weights 0.2, 0.6, 0.2

$A \rightarrow B \rightarrow C \rightarrow D$ with weights 0.2, 0.2, 0.2, 0.4

To derive a larger maximal route from these sessions we follow these steps:

- 1- Set a counter of the number of sessions in the universal profile (in our example Count=4).

2- If the end node of any session represents the beginning node of any other session(s), we should create an integrated route. In the previous example we will get the following:

$D \rightarrow J \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.3, 0.4, 0.1, **0.2**, 0.6, 0.2

$C \rightarrow F \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.4, 0.1, 0.1, **0.3**, 0.6, 0.2

Where the relative weight of item **S** will become $(0.2+0.2)/2$ and $(0.2+0.4)/2$ respectively in these new larger sessions.

Our system should then remove a merged session such as $S \rightarrow H \rightarrow Z$, decreasing the Counter by one (now, in our example, **Count**=3) and the remaining sessions are:

$D \rightarrow J \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.3, 0.4, 0.1, **0.2**, 0.6, 0.2

$C \rightarrow F \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.4, 0.1, 0.1, **0.3**, 0.6, 0.2

$A \rightarrow B \rightarrow C \rightarrow D$ with weights 0.2, 0.2, 0.2, 0.4

Again, we will look for sessions where the end node of one is the beginning node of any other. We find this to be the case for the first and third of the above, so we will create the integrated route:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow J \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.2, 0.2, 0.2, **0.35**, 0.4, 0.1, 0.2, 0.6, 0.2.

The remaining sessions are:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow J \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.2, 0.2, 0.2, **0.35**, 0.4, 0.1, 0.2, 0.6, 0.2.

$C \rightarrow F \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.4, 0.1, 0.1, **0.3**, 0.6, 0.2

Again, we will look for sessions where the end node of one is the beginning node of any other, no match case found then the created integrate routes are

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow J \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.2, 0.2, 0.2, **0.35**, 0.4, 0.1, 0.2, 0.6, 0.2.

$C \rightarrow F \rightarrow R \rightarrow S \rightarrow H \rightarrow Z$ with weights 0.4, 0.1, 0.1, **0.3**, 0.6, 0.2

We should mention here that although we have two different weights for node **C** but for different sessions, such integrated routes will be useful for batch recommendation. All these created maximal routes should be stored on the integrated maximal routes profile.

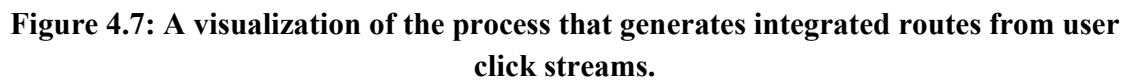
Integrated sequential maximal routes profile.

As we explained previously, the remaining integrated maximal routes will be stored in the integrated routes profile and all super sessions in the universal profile should be removed. Integrated routes should be updated from time to time with new information, and only integrated routes will be maintained for recommendations. Duplication is not allowed between any two integrated routes. Therefore in the next iterations if any new created integrated routes cause duplication in the integrated profile then our system should update the existing maximal routes items' relative weights; otherwise, the system will store the received maximal routes to the integrated route profile. Table 4.7 shows a sample of created integrated routes.

Route#	node1	node2	node3	node4	node5	node6	node7
1	Nato and Russia.aspx	action to stem job los	scotland.aspx	UK oil workers.asp	U.K. Shares.aspx	Fuel demand.aspx	Energy Coop
	61	23	7	7	22	12	23
2	Manufacturing Shrink.	BOE.aspx	RBS.aspx	action to stem job	scotland.aspx	UK oil workers.aspx	U.K. Shares.
	49	25	22	6	7	7	22
3	Energy.aspx	action to stem job los	scotland.aspx	UK oil workers.asp	U.K. Shares.aspx	Fuel demand.aspx	Energy Coop
	55	26	7	7	22	12	23
4	Manufacturing Shrink.	IRS.aspx	action to stem job los	scotland.aspx	UK oil workers.aspx	U.K. Shares.aspx	Fuel deman
	7	81	10	7	7	22	12
5	Azeri radar.aspx	Gaza seize.aspx	action to stem job los	scotland.aspx	UK oil workers.aspx	U.K. Shares.aspx	Fuel deman
	15	82	5	7	7	22	12
6	Gaza War.aspx	US Foreign Policy.asp	Unraveling Injustice.a	Lending.aspx	US Treasury secretary	UK missile defence.ar	action to ste
	23	35.75	29.5	11.5	13	27	8
7	Nato and Russia.aspx	action to stem job los	scotland.aspx	UK oil workers.asp	U.K. Shares.aspx	Fuel demand.aspx	Energy Coop
	61	28	6	24	12	25	4
8	Manufacturing Shrink.	BOE.aspx	RBS.aspx	action to stem job	scotland.aspx	UK oil workers.aspx	U.K. Shares.
	49	25	22	11	6	24	12

Table 4.7: Sample of created integrated routes.

In previous sections we showed how we collect users' maximal session, how we absorb such sessions to create super sessions, and then how we generate integrated routes. Figure 4.7 shows a simple visualisation of how we proceed from users' click streams to integrated maximal routes.



Node recommendations aim to create recommendations of good nodes to visit, from those that are directly linked to the active node. A batch recommendation, however, will be a set of suggested nodes, which could be anywhere on the site (that is, they do not have to be available in a hyperlink at the active node); batch recommendations represent the top N highly weighted nodes on the user's expected future path, which in turn is based on his/her current maximal session and its match with the integrated route profile. The rules and algorithms suggested in chapter three are used to collect candidate nodes for

recommendation. Figure 4.8 illustrates a node recommendation scenario; node D is the active node in the figure while nodes T , R , and E are the candidate nodes, as well as any stored new items with a physical link to these four candidate items. As soon as the candidate items are determined, only items of higher relative weight are given high priority and selected for recommendation, while newly-added items are given moderate priorities and also selected for recommendation.

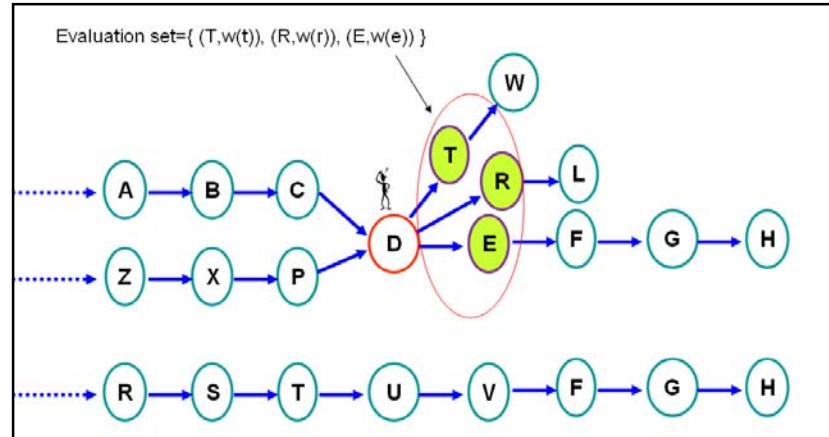


Figure 4.8: Candidate items for node recommendation.

Table 4.8 shows sample of generated node recommendations in the implemented system.

Visit_Number	node1	node2	node3	node4	node5	node6	node7	node8	node9
1023	ECB At Loss	Manufacturing							
1024	Success in Iraq	Iraq Voted	US Foreign Poli	Unraveling Inju	US Conservativ				
1025	Educational techno	Electronic learn							
1026	Success in Iraq	Iraq Voted	US Foreign Poli	Unraveling Inju	US Conservativ				
1027	Nato and Russia	Senate Vote							
1028	Success in Iraq	Iraq Voted	US Foreign Poli	Unraveling Inju	US Conservativ				
1029	Sharp Brain	The Exercise in							
1030	US Treasury secreta	US focus shifts	Unraveling Inju	Nato and Russi	World News	Iraq Voted			
1031	Fuel demand	Middle East	US Foreign Poli	Success in Iraq	US foreign poli				
1032	A Stimulus for the F	Lending							
1033	Afghan deaths	Carmaker Bent	Gaza War	US Conservativ	Nato and Russi				
1034	Oil Market	England	US Treasury se	UK missile defe	UK oil workers	Lending	Senate Vote		
1035	Success in Iraq	Iraq Voted	US Foreign Poli	Unraveling Inju	US Conservativ				
1036	US Treasury secreta	Success in Iraq	Gaza War	Unraveling Inju	United States C	Gaza seize	Finance	President Is Ju	
1037	Energy Cooperator	Russian gas sec							
1038	HSBC	Long Depressic	Tourism Slump	Success in Iraq	Lending	Finance	Qatari GTL Proj	Energy	
1039	ECB At Loss	Hot Earth							
1041	U.K. Shares								
1042	Gene Study	Unraveling Inju							
1043	Flight Simulator	Mystery of Billi	Finance	United States C	Consumer	Politics US Eco	Till Children D	Honeymoon	England
1044	British Air Ways	Oil Market	Tourism Slump						
1045	ECB At Loss	Manufacturing							
1046	IRS	BOE	Oil Market	Energy Cooper	US Treasury se				

Table 4.8: Sample of generated node recommendations.

In batch recommendation, all items of high relative weight in the future path can be selected as high priority candidate for recommendation, and new items related to those candidate items can be selected for recommendation with moderate priority. A batch

recommendation scenario is illustrated in Figure 4.9. In the figure, nodes *E*, *F*, *G*, and *H* are all candidates for batch recommendation, while only *E* and *H* are included in the recommendation set owing to their higher weights.

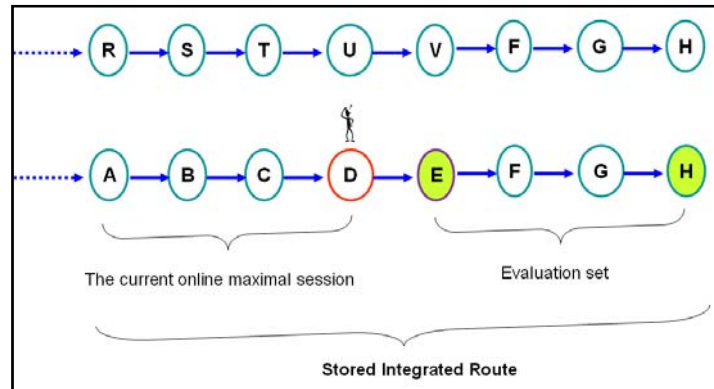


Figure 4.9: Candidate items for batch recommendation.

Table 4.9 shows a sample of batch recommendations generated in the implemented system.

Visit_Number	node1	node2	node3	node4	node5	node6
1062	Startup Costs	Manufacturing Shrini	IBM jobs	US Treasury secret	Mexico Bond Risk	Manufacturing Shr
1063	IBM jobs	Banks Shut	Tourism Slump	ECB At Loss	Long Depression	
1064	Small Business	Banks Shut	Tourism Slump	ECB At Loss	BMW Sales	Climate
1065	US focus shifts to A	Russia missiles	UK oil workers	Nato and Russia		
1066	Poor Feet	Sauna	The Exercise in Var			
1067	Best Treatment	Sauna	The Exercise in Var	Twins	Dental	
1068	Russia missiles	Startup Costs	Small Business			
1069	Startup Costs	Oil Market	Small Business	US Treasury secret	Mexico Bond Risk	Manufacturing Shr
1070	Mexico Bond Risk	Succession	Banks Shut	Manufacturing Shr	BMW Sales	British Air Ways
1071	Gaza War	Russia missiles	Old Forests, New F	UK oil workers	US focus shifts to A	Senate Vote
1072	Qatari GTL Project	Startup Costs	Small Business	Mexico Bond Risk	Crude Oil prices	ECB At Loss
1073	Long Depression	Startup Costs	Oil Market	Small Business	US Treasury secret	Crude Oil prices
1074	Manufacturing Shri	Succession	Banks Shut	ECB At Loss	Manufacturing Shri	BMW Sales
1075	Qatari GTL Project	Startup Costs	Oil Market	US Treasury secret	Small Business	Crude Oil prices
1076	IBM jobs	Banks Shut	Tourism Slump	ECB At Loss	BMW Sales	Energy Cooperatio
1077	Sharp Brain	Food Safety	Best Treatment	Sauna	The Exercise in Var	Twins
1078	Qatari GTL Project	Startup Costs	Mexico Bond Risk	Manufacturing Shri	Small Business	Banks Shut
1079	Sharp Brain	Food Safety	Green Energy	Dental		
1080	Best Treatment	Sauna	Dental			
1081	Mexico Bond Risk	Startup Costs	Small Business	Crude Oil prices	US Treasury secret	ECB At Loss
1082	US Treasury secret	Long Depression	IBM jobs	London	A Stimulus for the United States Cult	

Table 4.9: Sample of generated batch recommendations.

4.3 Alternative methods for solving the cold start problem

The active node technique (ANT) depends on previous users' visits to build integrated routes and then to generate recommendations for new users; newly-added items are given special treatment to promote their recommendation, but this treatment is centred on their relationships with more well-established items on the site. Many alternative methods are used and/or researched for solving the cold start problem; as indicated in chapter three. In this section we will present four of these alternative methods, each of which we use later as comparative methods when we evaluate the ANT.

4.3.1 The Naïve Filterbot model

Park et al., 2006 implemented the *Naïve Filterbots* algorithm, this method injects 'pseudo users' or bots into the system. These bots rate items according to attributes of items or users, for example according to the average rating of some demographically similar users. Once the filterbots are defined and injected into the user-item matrix, the system will treat them like any existing users, and treat their ratings of items as valid ratings. Standard CF algorithms are then applied to generate recommendations. This method is an extension of *RipperBots* proposed by (Good et al., 1999), in which filterbots were automated agents that rated all or most items using information filtering techniques. As soon as a bot injects ratings into the system, used user-based and item-based algorithms are used to calculate predicted ratings for items. The user-based algorithm depends on the *Pearson correlation* coefficient to measure similarity between users as follows:

$$sim(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u) \cdot (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_i (r_{v,i} - \bar{r}_v)^2}} \quad (4.1)$$

Where $sim(u,v)$ is the similarity between users u , and v , while $r_{u,i}$, and $r_{v,i}$ are the ratings of item i by both users u and v . In addition \bar{r}_u represents the user u average rating for all items, and \bar{r}_v represents the user v average rating for all items, and $I_u \cap I_v$ is the set of items that have been rated by both users u and v .

A modified similarity formula $sim'(u,v)$ is used if the intersection $I_u \cap I_v$ is small:

$$sim'(u,v) = \frac{\min(|I_u \cap I_v|, \gamma)}{\gamma} * sim(u,v) \quad (4.2)$$

Then, the predicted rating of item j for user u is calculated as follows:

$$p_{u,j} = \bar{r}_u + \frac{\sum_{v \in U} sim'(u,v) * (r_{v,j} - \bar{r}_v)}{\sum_{v \in U} |sim'(u,v)|} \quad (4.3)$$

The item-based algorithm depended on *adjusted cosine similarity* to calculate similarity between items as follows:

$$sim(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) * (r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} * \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}} \quad (4.4)$$

Where $sim(i,j)$ represents the similarity between items i and j . If the number of users who rate items is small, then a modified similarity is calculated. The predicted rating of the item i for user u is then:

$$p_{u,i} = \bar{r}_i + \frac{\sum_{j \in I_u} sim(i,j) (r_{u,j} - \bar{r}_j)}{\sum_{j \in I_u} |sim(i,j)|} \quad (4.5)$$

Where \bar{r} represents the average rating of item i , and $r_{u,j}$ represents the rating of item j done by user u .

4.3.2 The Triadic Aspect Model

Lam et al., 2008a suggested a method using users' demographical information such as age, gender, and job, originally suggested by (Hofmann, 1999). Given a set of items $Y = \{y_1, y_2, \dots, y_k\}$ and a set of users $U = \{u_1, u_2, \dots, u_k\}$, a basic data element is a triple (u, y, r) where u is a user, y is an item, and r is the rating of item y by user u . Another key data element is the *triple* (a, g, j) which represents a user, representing the features age, gender and job respectively – an example set of users is in table 4.11. In the triadic aspect method, each user (or category of users) is also considered to be represented by a vector of latent variable values, where each latent variable corresponds to a feature of an item. For example, on a

news website, the first value might represent the user's interest in sport, the second might represent his interest in economics, etc. The triadic aspect model works by calculating estimates of how a user will rate a certain item based on its features, using the historical data about categories of users and their ratings. The key equation used to work this out is equation 4.6 (we give an example later to explain it). This gives a rating $R(z | a, g, j)$ for feature z , given that the user has the demographic triple (a, g, j) .

$$R(z | a, g, j) = \frac{R(z)R(a | z)R(g | z)R(j | z)}{\sum_{z'} R(z')R(a | z')R(g | z')R(j | z')} \quad (4.6)$$

To predict a user's rating for an item y that has a set of features Z , equation 4.7 sums over Z the products of $S(y, z)$ and $R(z | u)$, where $S(y, z)$ is item y 's share of the total ratings we have for items with feature z , and $R(z | u)$ is calculated by equation 4.6.

$$\text{Rating of } y \text{ with feature set } Z \text{ by user } u \text{ with triple } (a, g, j) = \sum_{z \in Z} S(y, z)R(z | a, g, j) \quad (4.7)$$

In the following, we work through an example to demonstrate how we implement the triadic aspect model.

Suppose that we have items A, B, C, D and E and a set of users where each user visit one or more of these items, and either implicitly or explicitly assigns a rating to the items they visit. In our context, users set a rating implicitly, since we use the time that the user spends on that item as their rating. Each item has a set of features. The features, for each item, are one or more categories that describe that item. These can be assigned according to the website's directory or link structure, or according to semantic information (e.g.in RDF statements). These *features* are also the latent variables. Table 4.10 shows the features assumed in this example for items A to E.

Item	A	B	C	D	E
Features (latent variables)	Politics Economic	Action, Adventure War Politics	Sports Football Tennis	Business Technology Electronics Football	Economic Business Technology

Table 4.10: Example item features.

We assume a set of 11 users $U = \{u_1, \dots, u_{11}\}$, for each of whom we have the demographic data age, gender and job (e.g. user u_1 has *triple* (25, male, Student)). In this example, Table 4.11 shows the information for our users, using the coding shown in figure 4.11.

User	Age	Gender	Job
u_1	20	0	0
u_2	30	0	1
u_3	0	1	1
u_4	30	0	0
u_5	0	1	1
u_6	20	0	0
u_7	0	1	1
u_8	0	1	1
u_9	20	0	0
u_{10}	0	1	3
u_{11}	30	0	1

Table 4.11: Users' demographic triples.

From the information in Table 4.11, we can place the users into demographic categories as shown in Table 4.12.

Triple(a,g,j)	n(a,g,j)	User
(20,0,0)	3	u_1, u_6, u_9
(30,0,1)	2	u_2, u_{11}
(0,1,1)	4	u_3, u_5, u_7, u_8
(30,0,0)	1	u_4
(0,1,3)	1	u_{10}
Total	11	

Table 4.12: Users per categories.

Finally, Table 4.13 tells us the ratings that each user has made for the items (among A, B, C, D and E) that they have visited.

User/Item	A	B	C	D	E
u_1	0.7	-	-	0.55	0.47
u_2	0.42	-	0.45	-	-
u_3	-	0.72	-	0.85	0.63
u_4	0.35	-	-	0.44	-
u_5	-	0.45	-	-	0.66
u_6	0.36	-	0.67	0.34	0.72
u_7	-	0.84	-	0.76	-
u_8	0.23	-	-	-	0.88
u_9	-	0.34	-	0.53	-
u_{10}	0.59	-	-	0.45	-
u_{11}	-	0.37	-	0.71	0.34

Table 4.13: Users-Items click-streams matrix.

Therefore, for example, user u_1 belongs to a category of users who have interests in (Politics, Economic, Business, Technology, Electronics, and Football). We can now look at the information so far considering each user category (defined by the rows in table 4.12) and individual feature separately. The outcome is given in Tables 4.14 and 4.15. An entry in Table 4.13 shows the average rating provided by users in that user category (the row) for items whose features include the specified feature (column).

(a,g,j)	n(a,g,j)	Politics	Economic	Action	Business	Adventure	War	Sports	Football	Tennis	Technology	Electronics	Totals	Users
(20,0,0)	3	1.4	2.25	0.34	2.61	0.34	0.34	0.67	2.09	0.67	2.61	1.42	14.74	u_1, u_6, u_9
(30,0,1)	2	0.42	0.76	-	1.05	-	-	-	0.71	-	1.05	0.71	4.7	u_2, u_{11}
(0,1,1)	4	2.24	2.4	2.01	3.78	2.01	2.01	0.45	2.06	0.45	3.78	1.61	22.8	u_3, u_5, u_7, u_8
(30,0,0)	1	0.35	0.35	-	0.44	-	-	-	0.44	-	0.44	0.44	2.46	u_4
(0,1,3)	1	0.59	0.59	-	0.45	-	-	-	0.45	-	0.45	0.45	2.98	u_{10}
Totals	11	5	6.35	2.35	8.33	2.35	2.35	1.12	5.75	1.12	8.33	4.63	47.68	

Table 4.14: Distribution of features against users.

In contrast, Table 4.15 shows us how item ratings are distributed over features. The entries in row D are all 4.63, which are the total ratings we have for item D , while the column titles show the total ratings we have had for items whose feature set includes that column.

Items/ Features	Politics	Economic	Action	Business	Adventure	War	Sports	Football	Tennis	Technology	Electronics	total
A	2.65	2.65	-	-	-	-	-	-	-	-	-	5.3
B	2.72	-	2.72	-	2.72	2.72	-	-	-	-	-	10.88
C	-	-	-	-	-	-	1.12	1.12	1.12	-	-	3.36
D	-	-	-	4.63	-	-	-	4.63	-	4.63	4.63	18.52
E	-	3.7	-	3.7	-	-	-	-	-	3.7	-	11.1
Total	5.37	6.35	2.72	8.33	2.72	2.72	1.12	5.75	1.12	8.33	4.63	49.16

Table 4.15: Distribution of features against Items.

We will now show how we use equation 4.6 to calculate $p(\text{Business}, (30, 0, 1))$. That is, we use equation 4.6 for the case where the latent variable is 'Business' and the user category is $(30, 0, 1)$. First, we just work through the calculation of $R(z)R(a|z)R(g|z)R(j|z)$, Where z is business, a is 30, g is 0 and j is 1. This gives us the numerator of equation 4.6 for $R(\text{Business}, (30, 0, 1))$. Keeping (a, g, j) the same throughout, we then calculate $R(z, (30, 0, 1))$ for each of the other features, and sum them to have the denominator. In the end, we have $R(\text{Business}, (30, 0, 1))$, which gives us an estimate for how a user in that category would rate an item with feature 'Business'.

So, the calculation becomes $R(\text{Business})R(30|\text{Business})R(0|\text{Business})R(1|\text{Business})$

Where:

$$\begin{aligned}
\circ \quad R(\text{Business}) &= \frac{8.33}{49.16} = 0.1694 \\
\circ \quad R(30|\text{Business}) &= \sum_{z=\text{"Business"}} \frac{R(30 \text{ and Business})}{R(\text{Business})} = \frac{1.05}{8.33} + \frac{0.44}{8.33} = 0.1789 \\
\circ \quad R(0|\text{Business}) &= \sum_{z=\text{"Business"}} \frac{R(0 \text{ and Business})}{R(\text{Business})} = \frac{2.61}{8.33} + \frac{1.05}{8.33} + \frac{0.44}{8.33} = 0.492 \\
\circ \quad R(1|\text{Business}) &= \sum_{z=\text{"Business"}} \frac{R(1 \text{ and Business})}{R(\text{Business})} = \frac{1.05}{8.33} + \frac{3.78}{8.33} = 0.5798
\end{aligned}$$

...in which the details of the calculation can be explained by reference to table 4.14.

Our expression therefore works out to $(0.0678)(0.1789)(0.492)(0.5798) = 0.00346$. To finish calculating equation 4.6 we now simply work out the denominator, which is the sum of similar calculations, one for each feature (each column of Table 4.14).

Let us suppose that this results in a value for $R(\text{Business}, (30, 0, 1))$ of 0.021. and suppose we have also done the same for all latent variables (features) and all user categories. So, we now have $R(\text{Feature} | \text{UserCategory})$, for every feature and user category.

We can now use equation 4.7 to predict the rating that some new user u_{12} would provide for any given item. Suppose a new user has triadic triple $(30, 0, 1)$, and we want to know how they will rate item E , which has feature set $Z = \{\text{Business}, \text{Economics}, \text{Technology}\}$. We now use equation 4.7, which sums $S(E, z)R(z | 30,0,1)$ for each element in Z . The $R(z | 30,0,1)$ terms are calculated as we have seen, while the $S(x_5, z)$ are calculated using the information in Table 4.15. For example, $S(E, \text{Business}) = 3.7/8.33 = 0.442$, which is item E 's share of the ratings received so far for all items that include Business as a feature.

4.3.3 MediaScout stereotype model

MediaScout system proposed by (Shani et al., 2007) used a stereotype approach by combining elements from both content-based and collaborative filtering. They created a set of 'stereotype' content-based profiles, using an affinity vector of stereotypes as the user profile. Moreover, they classified new users into clusters through an interactive questionnaire, generated automatically from the stereotypes after each update, while existing users are automatically classified to new stereotypes through the update process and do not need to go through the questionnaire again. A relevance value is calculated based on the match between item profiles and the stereotype profile; this relevance value is used to generate recommendations. This model follows the following steps.

1. Generate initial stereotype profiles manually (an informed developer will be able to identify the key features relevant for people when they choose which movie to see), by identifying a set of attributes (features) and assigning relevance values *relevance* (i, s) for various pre-identified attributes s extracted from different items i_s . For example, maybe for a specific movie the actor X will get a relevance value 0.85,

while the director of the movie Y may get a relevance value 0.9, and both X , and Y are pre-determined features of this movie.

2. When a new user enters the system, he or she answers a set of simple questions, such as whether he likes an actor, or is asked to choose their preferences from a list of movies. A profile, based on a vector of affinities with a set of stereotypes, is then calculated for this user.
3. Given a particular item, the relevance value of a media item to each stereotype can be calculated, and then the relevance of that item to any particular user can be estimated, as follows:

$$relevance(i, u) = \sum_{s \in stereotypes} v(s)relevance(i, s) \quad (4.8)$$

Where $relevance(i, u)$ refers to the relevance value of item i to user u . while $relevance(i, s)$ refers to the relevance value of item i to stereotype s .

4.4 Description of Experiments

In order to evaluate the validity and value of the suggested Active Node Technique (ANT) in comparison with alternative methods (Naïve Filterbots Model, Triadic Aspect Model, and MediScout Stereotype Model), we first collected and used a set of users' preferences (section 4.4.1). Then, by implementing the ANT and the four mentioned alternative methods, we obtained recommendation sets (a different recommendation set for each method) which were used in the evaluation process as described in section 4.4.2. Experimental results that provided in section 4.4.3 are showing the real implementations of the different methods.

4.4.1 Website chosen for online evaluation experiments

The standard data sets used in the evaluation of many collaborative filtering algorithms are the MovieLens, Book-Crossing and/or Jester joke dataset, but these datasets do not provide the website structure, or semantic relationships between objects, that are required to properly test and validate the ANT. Therefore, we set up a separate web site with a copy of the

Alarabiya¹, then colleagues had been asked to use it in different training sessions, and then the collected click streams had been used to test our method comparing to alternative methods. AlArabiya.net is the leading news channel in the Arab world. The English news version of this website was launched in August 2007 to build a bridge between the channel and English-speakers who are interested in the Middle East. AlArabiya.net is one of the most reliable sources of news and analysis about the Middle East catering to readers all over the world; figure 4.10 shows the AlArabiya.net website main interface.

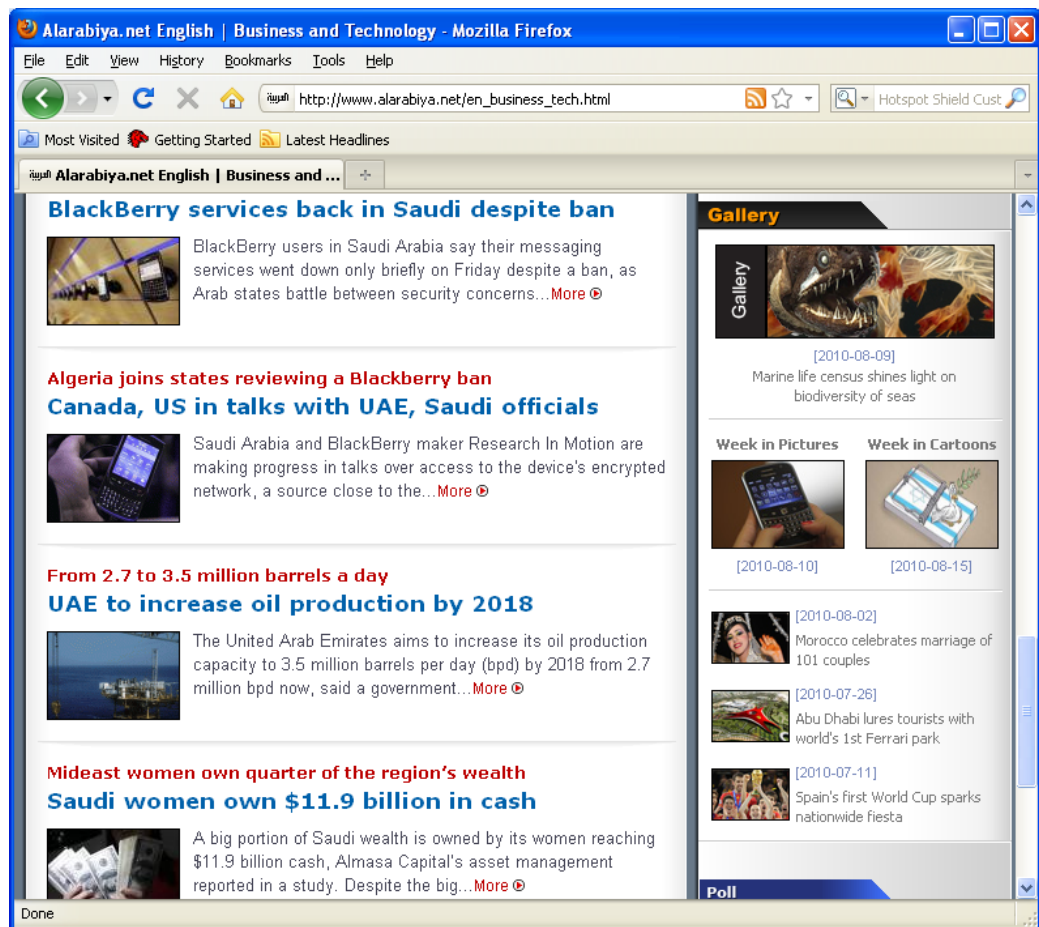


Figure 4.10: AlArabiya.net website main interface.

Datasets were collected in four phases². In the first phase log data are cleaned and used to generate integrated routes (for the ANT method) and the time spent time per page is considered as the page weight that reflects how valuable the page seems to be to site visitors. For the *stereotype method* we created an initial stereotype profile with specific relevance

¹ An online news web site <http://www.alarabiya.net/english/>

² Every phase of data collection was a three months time period

values related to different topics (news, sports, business, technology, etc), while the cleaned data was used to generate user affinity stereotypes profile, with time spent time per page considered as the relevance value. For the *naïve Filterbots method*, we first created a user-item matrix with cleaned data associated with each item's weight (average time spent by users on that item). Then, we created user-feature bots that were used to generate items ratings (using average weight) for any new user based on the demographic data of users (age, profession, gender, etc). Then, we inject these filterbots (like any other existing user associated with its ratings or spent time) into the user-item matrix (we implemented naïve filterbots for the user based approach only), and then implemented user based algorithm to calculate predictions and find recommendation sets. For the *demographic based model*, we collected age, gender, and job for all users involved in the training session (264 users are involved in the training process), and then by implementing the *Triadic Aspect Model*, we calculated item ratings predictions using demographic features, as shown in figure 4.11.

Gender	
0	Male
1	Female

Age	
0	<20
20	20-29
30	30-39
40	40-49
50	50-59
60	60+

Professions	
0	Student
1	Tutor
2	Assistant manager
3	Customer services
4	Helpdesk
5	Developer
6	Business man
7	Accountant
8	Inspector
9	Data entry
10	Internal Auditor
11	Sales man

Figure 4.11: Users' demographical features.

In the second phase, we provided recommendations based on each method to users, and collected their subsequent selections, and then the collected data were used to update the test data. *In the third phase*, we again provided recommendations to users based on each of the

different methods under study, and again collected the users' selections. Recommendation sets collected in the second and third phases, along with the users' selections are used for evaluation.

4.4.2 Methods and metrics for evaluation

We aim to evaluate the *novelty* of recommended items, as well as the *precision* and *recall* of recommendations; therefore, we used the novelty formula from chapter three, as well as the suggested formulae for precision and recall for both node recommendation and batch recommendations. In this section, we show examples of the way we calculated precision and recall. Figure 4.12 illustrates the case of a specific user who visits node D as part of the session $A \rightarrow B \rightarrow C \rightarrow D$. While the user is at D (D is the active node), the recommendation system (whether this is the ANT or one of the comparative techniques) makes a set of recommendations – this is the recommendation set (RS). In due course, however, the user will continue his or her session and actually visit a series of new nodes. The set of nodes that the user actually visits is called the Target set (TS). To evaluate the recommendations made at the point when the user is at node D, the recommendation set generated at that point must be compared with the target set (nodes actually visited after that point).

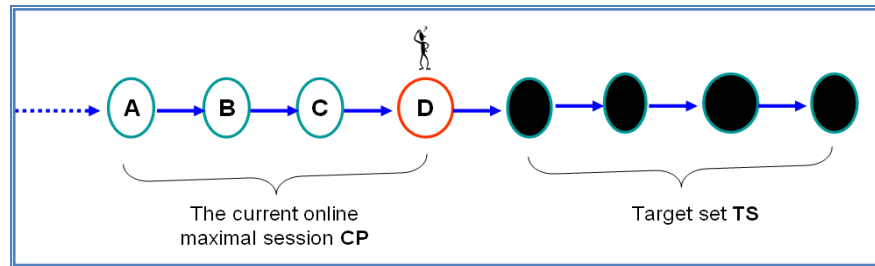


Figure 4.12: User online maximal path and the expected target set.

Using the active node method, the system will take the current maximal online path (CP), and then by implementing rules of node and batch recommendation, will generate a recommendation set (RS) which will be delivered to the user. The user's subsequent movements from node to node are recorded, and become the target set (TS) that will be stored to complete the user's maximal path, as shown in figure 4.13.

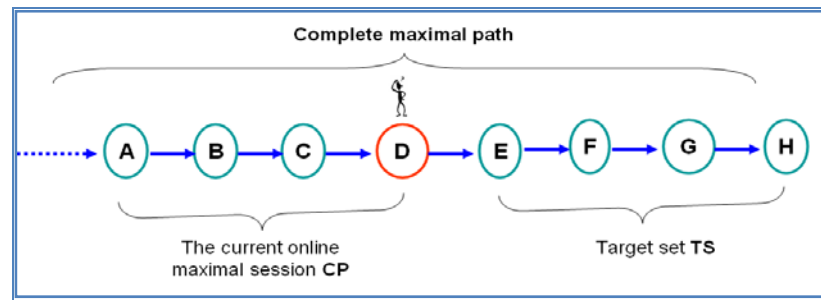


Figure 4.13: Complete maximal path.

We now work through an example to help explain the aspects of the evaluation process that have been discussed so far. Table 4.16 displays an online maximal session. The user has visited these nodes, in order from left to right.

<i>a</i> <i>maximal</i> <i>session</i>	about History.aspx	action to stem job losses.aspx	Banks Shut.aspx	Mexico Bond Risk.aspx	Manufacturing Shrink.aspx	Oil Market.aspx
---	-----------------------	--------------------------------------	--------------------	-----------------------------	------------------------------	--------------------

Table 4.16: Online maximal session.

Given that Table 4.16 shows the complete maximal session, Table 4.17 illustrates the target sets associated with each node.

Maximal	about History.aspx	action to stem job losses.aspx	Banks Shut.aspx	Mexico Bond Risk.aspx	Manufactur ing Shrink.aspx	Oil Market.as px
TS ₁	action to stem job losses.aspx	Banks Shut.aspx	Mexico Bond Risk.aspx	Manufacturi ng Shrink.aspx	Oil Market.asp x	
TS ₂	Banks Shut.aspx	Mexico Bond Risk.aspx	Manufacturing Shrink.aspx	Oil Market.aspx		
TS ₃	Mexico Bond Risk.aspx	Manufacturin g Shrink.aspx	Oil Market.aspx			
TS ₄	Manufacturing Shrink.aspx	Oil Market.aspx				
TS ₅	Oil Market.aspx					

Table 4.17: Target sets associated with the maximal session in table 4.16.

For example, TS₃ is the target set associated with the point at which the active node was the third node in the session (the “Banks Shut” page). Table 4.18 now shows the

recommendation sets (generated by the ANT) associated with each node in the session. Analogous to the way TS_3 is defined, RS_3 is the set of recommendations generated by the system at the point when the user is at the third node (“Banks Shut”) in the session. As we can see from table 4.18, the ANT made 6 separate page recommendations, and 3 of these (which are highlighted in yellow) correspond to nodes in the target set TS_3 – i.e. these nodes were actually visited by the user later in that session.

K	1	2	3	4	5	-			Σ
$ R_i \cap TS_i $	4	3	3	2	1				13
TS	5	4	3	2	1	-			15
Maximal	about History.aspx	action to stem job losses.aspx	Banks Shut.aspx	Mexico Bond Risk.aspx	Manufacturing Shrink.aspx	Oil Market.aspx			
RS_1	Oil Market	Qatari GTL Project	action to stem job losses	Startup Costs	Sharp Brain	Banks Shut	Mexico Bond Risk	Small Business	8
RS_2	Banks Shut	Manufacturing Shrink	Small Business	Mexico Bond Risk	Crude Oil prices	Qatari GTL Project	Startup Costs		7
RS_3	Qatari GTL Project	Oil Market	Small Business	Mexico Bond Risk	US Treasury secretary	Manufacturing Shrink			6
RS_4	Qatari GTL Project	Startup Costs	Oil Market	Small Business	US Treasury secretary	Manufacturing Shrink			6
RS_5	Qatari GTL Project	Startup Costs	Oil Market	Long Depression	Crude Oil prices				5

Table 4.18: Match between target sets and recommendation sets.

Again making use of the example illustrated in Table 4.18, we can calculate the level of coverage as follows,

$$Coverage = \frac{\sum_{i=1}^n |R_i \cap TS_i|}{\sum_{j=1}^k |TS_j|} = \frac{13}{15} = 0.866$$

Coverage in this case comes to 0.866, which we round up and denote as 87%. Note that the level of coverage is calculated on the basis only of the first N-1 elements of the maximal session.

K	1	2	3	4	5	Σ
Match(TS,RS)	4	3	3	2	1	13
TS	5	4	3	2	1	15
	80%	75%	100%	100%	100%	
RS	8	7	6	6	5	32
Participation level	6.4	5.25	6	6	5	28.65
Accuracy level						90%

Table 4.19: Calculating precision.

The precision value is then calculated as follows:

$$Precision = \frac{\sum_{i=1}^n |R_i| \cdot \frac{|R_i \cap TS_i|}{|TS_i|}}{\sum_{j=1}^k |R_j|}$$

The calculation of precision for our ongoing example is illustrated in Table 4.19 and also below.

$$Precision = \frac{8 \times \left(\frac{4}{5}\right) + 7 \times \left(\frac{3}{4}\right) + 6 \times \left(\frac{3}{3}\right) + 6 \times \left(\frac{2}{2}\right) + 5 \times \left(\frac{1}{1}\right)}{32} = \frac{28.65}{32} = 0.895$$

As we have indicated before, in the first phase of the evaluation process, we perform a training session to generate initial profiles for the stereotypes model, Triadic Aspect model, and demographical data model, as well as to generate integrated routes for the active node method. In the second phase, all of recommendation sets generated by the different methods are collected and then the system updated, while the system is updated with the new

experience. Finally, in the third phase, a new training session is created, using collected data from the previous two phases as the test set.

4.4.3 Experimental Results

In this section, we demonstrate the calculated evaluation metrics for the four different evaluation methods, and discuss the results.

A) Level of novelty

Table 4.20 summarizes the calculated average novelty value for different stages of users' experience with the system (based on the number of node visits) and for each recommendation method studied. These results are also shown graphically in Figure 4.14.

Novelty								
	Number of Node Visits							
Method	≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Active Node (Batch Recommendation)	0.75	0.7	0.69	0.64	0.67	0.72	0.76	0.82
Active Node (Node Recommendation)	0.65	0.63	0.59	0.55	0.54	0.6	0.62	0.65
Triadic Aspect Model	0.6	0.56	0.51	0.44	0.43	0.43	0.33	0.3
Stereotype Model	0.5	0.49	0.41	0.39	0.3	0.34	0.35	0.27
Naïve FilterBots Model	0.66	0.65	0.6	0.57	0.55	0.53	0.54	0.49

Table 4.20: Novelty values for different methods.

Table 4.20 shows that ANT batch recommendations achieve the highest level of novelty. This can be explained in terms of the fact that recommendations from the ANT are based on integrated routes, which brings in information from other users. ANT node recommendations have lower novelty lower than ANT batch recommendations, since node recommendation candidates are restricted to those nodes with virtual or hyperlink relationship to the current active node. The Naïve filterbots approach achieves a high level of novelty, similar to that of ANT node recommendations, but with increasing time using the system, the level of novelty declines. This is probably because, when using the Naïve filterbots approach, re-injected

ratings are not too much differ from previously injected ratings¹ (we used items average time duration as a default prediction). Although naïve filterbots started with high level of novelty (all site nodes injected with a false ratings and hence the available non repeated candidates for recommendation was high too) we should mention here that these injected ratings are rule based, and do not strongly reflect actual users' ratings. The Stereotype and Aspect models also start with a high level of novelty (but lower than ANT and naïve filterbots); however, their novelty scores dramatically decline, since both models (stereotype and triadic aspect) classify users based on stereotypic and demographic data, without effective ways to adapt changes in interests.

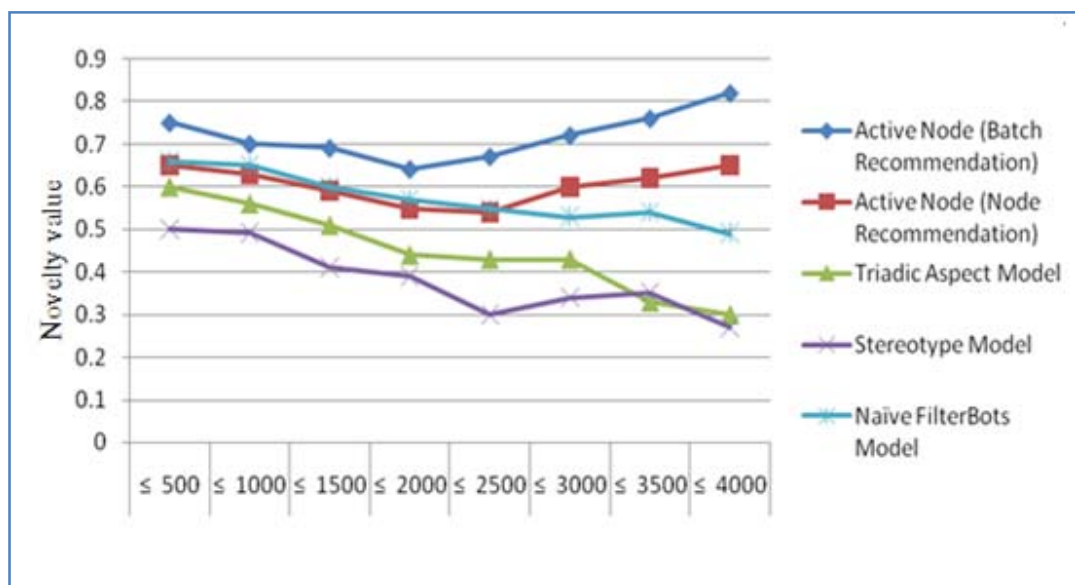


Figure 4.14: Novelty of recommendations.

¹ In the implementation, we used items time duration as ratings

B) Level of coverage

The idea of the coverage metric is to measure how well the recommended sets cover the items in the target sets; we used the coverage formula explained in chapter three, and Table 4.21 shows the calculated coverage value for each method used in the experiments described in this chapter.

Coverage								
	Number of Visits							
Method	≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Active Node (Batch Recommendation)	0.54	0.58	0.63	0.65	0.69	0.69	0.73	0.77
Active Node (Node Recommendation)	0.9	0.84	0.8	0.79	0.82	0.85	0.89	0.93
Triadic Aspect Model	0.55	0.59	0.62	0.63	0.66	0.7	0.69	0.65
Stereotype Model	0.5	0.52	0.56	0.58	0.6	0.62	0.62	0.59
Naïve FilterBots Model	0.2	0.23	0.3	0.25	0.24	0.19	0.29	0.3

Table 4.21: Coverage values for different methods.

As shown by Table 4.21, and the graphical view in Figure 4.15, ANT node recommendation achieved a high level of coverage. ANT batch recommendation achieved lower coverage than node recommendation, but its coverage increased as the number of visited nodes increased. The increase in the number of visitors leads to enhancement in the stored integrated routes, which also leads to increasing the provided candidates for recommendations. Therefore, in the first iterations, the number of integrated routes was lower than in the later sessions. The Triadic aspect method achieves coverage values similar to ANT batch recommendation initially, but its coverage declines over time.

Both the naïve filterbots and stereotype models achieved lower coverage than the ANT. In naïve filterbots, all non rated items are injected with false ratings, which consider all site nodes as visited, and hence a large number of candidate items for recommendation are available, increasing the chances of missing items in the target sets (this also explain its high level of novelty). The Stereotype model achieves a lower level of coverage than ANT because the system uses implicit feedback that considers all non-selected items from a

specific user as not-liked items. Hence, some of the items in target sets may vanish from recommendation sets after a certain number of iterations.

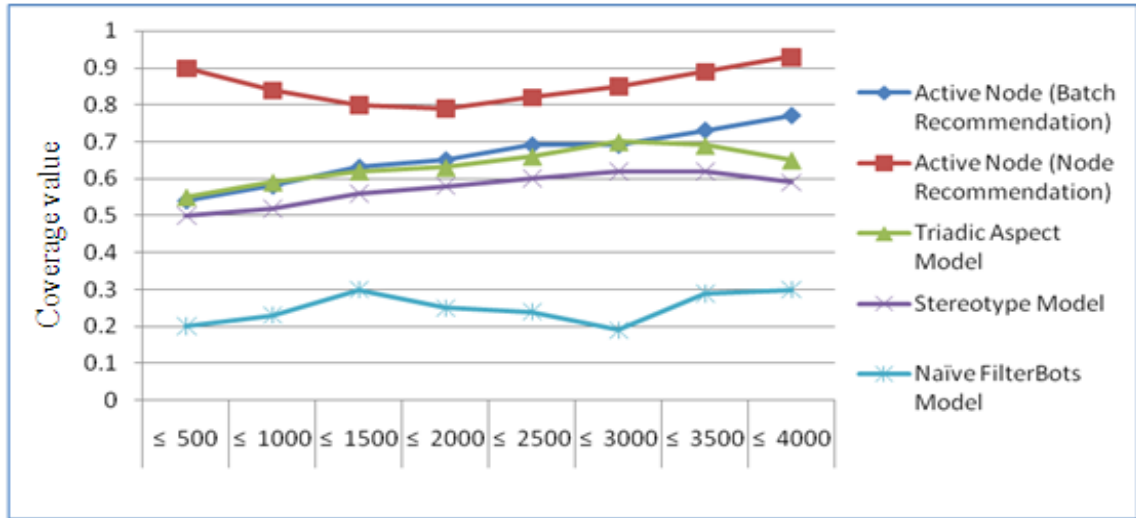


Figure 4.15: Coverage of recommendations.

C) Level of precision

Precision measures the accuracy of recommended sets; we use the precision formula explained in chapter three. Table 4.22 shows the calculated precision values for each of the recommendation methods implemented in our experiments, again with the values shown at different stages of time (measured in terms of number of visits). It is clear that as the coverage increases, the precision also increases.

Precision								
	Number of Visits							
Method	≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Active Node (Batch Recommendation)	0.51	0.55	0.60	0.62	0.66	0.66	0.70	0.74
Active Node (Node Recommendation)	0.87	0.81	0.77	0.76	0.79	0.82	0.86	0.90
Triadic Aspect Model	0.52	0.56	0.59	0.60	0.63	0.67	0.66	0.62
Stereotype Model	0.47	0.49	0.53	0.55	0.57	0.59	0.59	0.56
Naïve FilterBots Model	0.17	0.20	0.27	0.22	0.21	0.16	0.26	0.27

Table 4.22: Precision values for the tested recommendation methods.

Figure 4.16 shows the data from Table 4.22 in graphic form. ANT node recommendation achieved the highest precision level. In the case of the ANT, high novelty (as we saw earlier) does not conflict with high coverage and precision, since we measure novelty as a function of the items within recommendation sets, while coverage and precision calculations concern the match between recommendation sets and target sets. We should also mention here that the basis of recommendations generated by the ANT was the users' (and other users') clickstreams using the site, while in the Triadic Aspect model, recommendations were based on latent demographical parameters, and in the stereotype model they were based on stereotypical user categories. Finally, in the naïve filterbots model, recommendations were based on the ratings made by injected filterbots, which tend to favour specific highly weighted items.

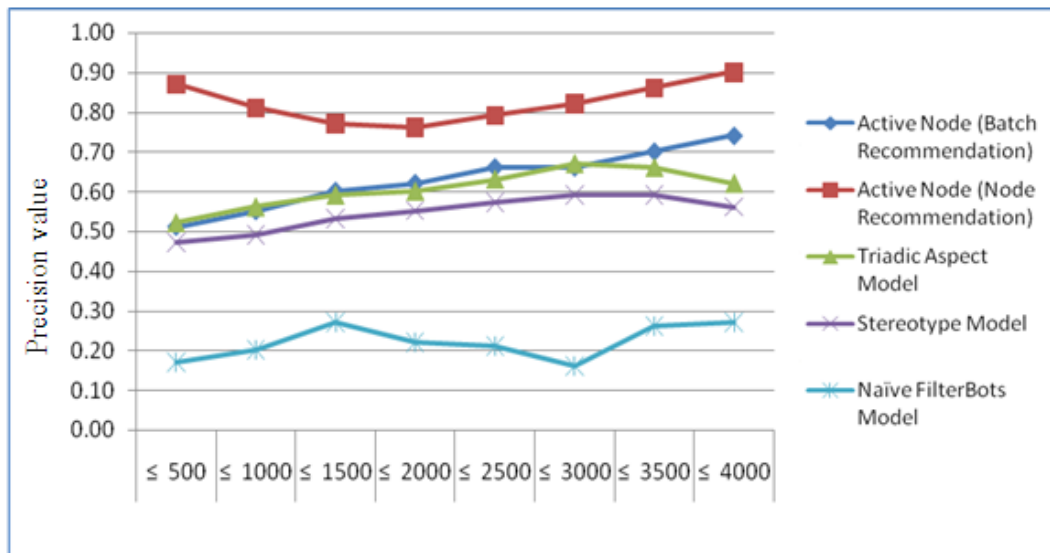


Figure 4.16: Precision of recommendations.

4.4.4 Conclusion

Each recommendation method that we have tested in this chapter has its own approach to solving the cold start problem, as well as (of course) its own approach to providing appropriate recommendations of items for existing users. The Naïve filterbots model provides a high level of novelty, but a quite low level of coverage and precision. The stereotype model achieved the worst level of novelty, but it achieved a higher level of coverage and precision. The Triadic Aspect model, which depends on user demographical data, achieved quite low levels of novelty, but it achieved high levels of coverage and precision. The ANT method achieved the highest levels of novelty (in batch mode), and achieved the highest levels of coverage and precision (in node recommendation mode) Table 4.23 provides a summary of the observations that can be made in comparing the ANT and the alternative methods investigated.

Criteria	Active Node (Batch Recommend ation)	Active Node (Node Recommend ation)	Triadic Aspect Model	Stereotype Model	Naïve FilterBots Model
Novelty	Highest level	Middle	Lower level	Lowest level	Middle
Coverage	Middle	Highest level	Middle	Lower level	Lowest level
Precision	Middle	Highest level	Middle	Lower level	Lowest level
Privacy	No privacy concern	No privacy concern	Privacy concern	Privacy concern	Privacy concern (user based filterbots)
Type of collected data	Users sequential preferences	Users sequential preferences	Users demographica l data	Users feedbacks and preferences	Items ratings /users ratings
The way we Collect data	Implicit data collection	Implicit data collection	Explicit data collection	Explicit data collection	Explicit data collection
Comparing method	Power of thinking	Power of thinking	Demographic al data	Stereotype data	Filterbots (injected false ratings)
Express users preferences	High	High	Low	Low	Very Low
Robust problem	Controlled	Controlled	High	High	Very high

Table 4.23: Comparing the active node technique with alternative techniques.

If collaborative filtering systems are built based on the ANT, they are effectively privacy-protected, since recommendations are made on the basis only of anonymized and processed click streams. As we can see in the above results, the ANT can lead to recommendations with strong levels of novelty, precision and coverage, despite not collecting any personal data from users. Also, the ANT techniques were either the best, or intermediate, on each evaluation measure, while each of the alternative techniques was worst in one or more of the measures. The results suggest that the ANT is a promising technique for use in recommendation systems.

In contrast, regarding privacy concerns, the alternative methods collected users' demographical data as well as other data. In the naïve filterbots model, users are identified by their IP addresses. Of course, if a version of the naïve filterbots model was used that depended only on abstract ratings then it becomes privacy protected. However in that case the robustness problem (direct recommendations to specific items) remains in the naïve filterbots model. Meanwhile, the Demographical method does not validate entered demographical data, which is sometimes does not reflect the user's true demographical data.

4.5 Summary

The active node technique (ANT) provides two different recommendation methods, *node recommendation*, and *batch recommendation*. In this chapter we have compared the ANT with four alternative approaches to recommendation. Our experiments showed that batch recommendation achieved the highest level of novelty, and node recommendation achieved the highest level of coverage and precision. The ANT techniques compare very well against the four alternatives tested here. Nevertheless, we only used one website and a limited number of users. Although there was nothing special or biased about the chosen website, it could be argued that a full evaluation of the ANT would require a range of different sites, with different amounts of online users. It would be interesting to see how the performance of the ANT varies in different situations, especially in comparison with other methods. Largely we leave the latter concerns to future work. In the remainder of this thesis, however, we consider how the levels of novelty for ANT node recommendations, and the levels of coverage and precision for ANT batch recommendations, may be improved. This leads us to a version of the approach which uses semantic information, which is described in the next chapter.

Chapter 5

Augmenting the Active Node Technique with Semantic Information

5.1 Introduction.

The huge amount of services and information provided on the internet imposes the necessity to find tools and mechanisms for saving users' time and money in the task of finding those services and information that are relevant to the user. Mechanisms providing relevant services to customers, by combining their own personal preferences with the preferences of like-minded other users, has been part of the improvements in web technology as we moved away from web 1.0 (where the web was merely 'readable' - characterized by static data and limited interaction between users and websites) to web 2.0 (where the web is increasingly 'writable' - allowing for much greater interaction and social networking). We now move towards web 3.0, representing the 'executable' phase in web development, providing dynamic web applications, interactive services, and machine-to-machine interaction, where the user's machine can automatically search the web to find the best options given the user's preference and desires, without consuming too much time. One of the growing elements in the upcoming web 3.0, helping to enable much of the new functionality, is the set of technologies and techniques that give us the *semantic web*.

The semantic web is defined by (Lee, B. 2010) as "a web of data with meaning in the sense that a computer program can learn enough about what the data means to process it". Berners-Lee defines the Semantic Web as "a web of data that can be processed directly and indirectly by machines" (Antoniou and Van Harmelen, 2004). The implementation of web semantic concept means that converting web from the web of documents to the web of data, which allows data to be shared and reused across application, as well as increase the accuracy of information retrieval on the web. Semantic data can be represented by variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as Resource Description Framework (RDF) Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain.

Generally, site visitors would like to find products and services more quickly and with less effort. Also, marketers want to provide the right message to the right individual at the right time, and they want to help customers and visitors to find the products and services they are interested in, and to promote similar products or services that the customers may not otherwise have searched for. Therefore, personal recommendation systems provide a solution

by customizing web sites based on users' preferences or via the preferences of like-minded users; however there are several problems with these systems, such as the cold start problem, privacy issues, and the scalability problem, which decrease the level of accuracy of recommended items. In this thesis, we suggest the active node technique to solve the cold start problem, as well as the privacy problem, while maintaining high quality and appropriate recommendations in all circumstances.

Current recommendation systems provide recommendations for items to website visitors, but often these recommended items are irrelevant and poorly chosen. Similarly, current search engines retrieve too many pages for most queries, with many or most of the retrieved pages being irrelevant to the user's requirements. The main goal of the *semantic web* is to provide more professional knowledge management systems by allowing the extraction of useful and meaningful information from web repositories.

Personal recommendation systems on the web aim to provide useful information to site visitors; the semantic web will clearly provide a proper environment and structure for such recommendations, and assist users in their day-to-day online activities. The semantic web tries to deal with the *content* of web pages, and the content generally available in the web as a whole. Semantic web systems aim to improve machine processing of web pages based on content information. It is natural to think that, by using semantic web techniques, we can make recommendations more accurate. Also, this will help the scalability of recommendation systems, since semantic methods should help to select only those items with relevance to the user's needs.

In the context of recommendations, we see that the semantic web, based on the current collection of languages and structures mentioned above, still does not provide certain functionality as we can see from the following:

1. Static mode recommendations; although the semantic structure of a website may help to provide better recommendations, these are still static, and will tend to be based on a pre-specified and fixed semantic structure.
2. Semantic structures do not help us to identify preference or priority among different classes of objects.
3. Semantic structures do not help us in finding recommendations for relevant nodes that are not integrated into the semantic structure.

In the coming sections, we will describe how the ANT can be implemented in a way that takes advantage of the semantic structures provided by websites, and we will also provide suggestions to overcome the limitations listed above.

The rest of this chapter is organized as follows. In section 5.2, we give a broad introduction to how we integrate the ANT in a semantic web context. In section 5.3, we broadly describe how we update the semantic information on the website during user visits; this includes subsections discussing properties added to the RDF/RDFS structures, clarifying the overall update method. In section 5.4, we provide some detail about the implementation of the new semantic properties discussed in section 5.3, including examples from RDF files, and in this section we also discuss how the semantic ANT handles the user and item cold start problems. In section 5.5 we are finally able to provide detail of the semantic ANT recommendation algorithms; this section starts by describing the basic idea of the semantic ANT node and batch recommendation approaches, and then it gives details of how recommendations are prioritised, and then specifies the steps of the algorithms. Finally, section 5.6 describes experiments that evaluate the semantic ANT recommendation methods and compare them with the non-semantic approaches.

5.2 Merging the active node technique with a semantic structure.

Using the ANT technique in an appropriate way in the context of a semantic web application should lead to recommendations that are more accurate and allow inference of additional useful recommendations for users based on mapping their browsing behaviour to the properties of recommended objects. Hence, using both users' click streams and the semantic descriptions of web pages allows the system to recommend items to users, not only based on like-minded users, but also based on page content. To achieve this, we would like in some way to merge the ANT's integrated routes with the semantic structure of the website.

The basic approach that we adopt is as follows. In the semantic version of ANT, first of all, the ANT is in many ways unchanged: as before, we collect data from user's click streams, and go through all of the steps to generate and maintain the integrated route profile, including the important elements of the impact values associated with each node in the integrated route profile. However, in the Semantic version of ANT, these impact values, as well as information about related pages (neighbours in the integrated route structure) are to

be treated as semantic information, and will be included in the semantic markup of the nodes themselves. The main difference between the previous ANT implementation and the semantic version is how recommendations are generated. Since nodes (the actual web pages, perhaps containing a news article or describing a product for sale) now have extra semantic information within them, we can use this information to generate recommendations directly. Recommendations are now made partly (and *indirectly*) via the integrated routes profile, since they will be made on the basis of semantic information contained in the active node's web page, which got there on the basis of the integrated route profile. Additional kinds of recommendation will also be made that take advantage of the other semantic information within the pages.

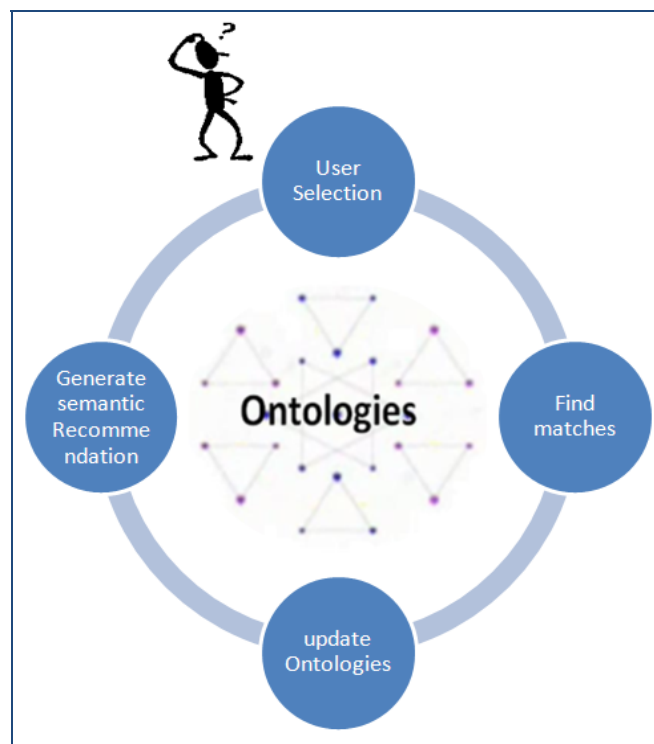


Figure 5.1: Semantic web recommendation cycle.

As shown in Figure 5.1, as the user starts his/her journey within the web site, the collected click streams will reflect the user's desires and goals, and updated information about the integrated routes will be used to update the semantic information at each node; the recommendation agent will be able to generate recommendations based on these, and also based on other aspects of the website's semantic structure (e.g. an ontology and information

about the positions of visited pages within the ontology). Obviously, to be able to generate up-to-date recommendations, our system must be able to update the semantic information.

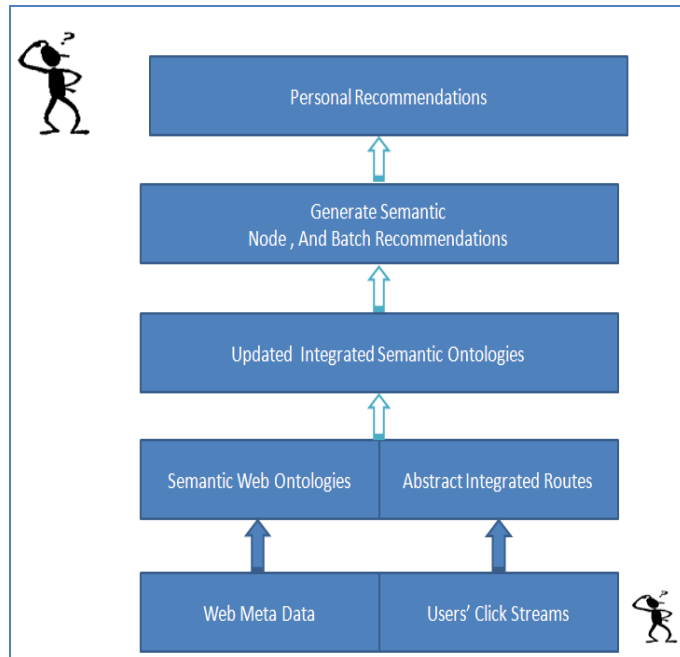


Figure 5.2: Semantic active node.

The semantic version of ANT (semantic ANT) is therefore a composite of semantic ontologies and the ANT that has been described in previous chapters. Users' click streams are used to create integrated routes (as before). The integrated routes are used to update the semantic information within the pages, and in turn are used to generate node and batch semantic recommendations, as shown by figure 5.2.

5.3 Updating item attributes within the semantic structure

The semantic ANT involves continual updating of semantic information within the web pages of the site under consideration. We will update the relative weights; those involved in the ANT calculations as described in previous chapters, the virtual links, and the priority levels for the virtual links.

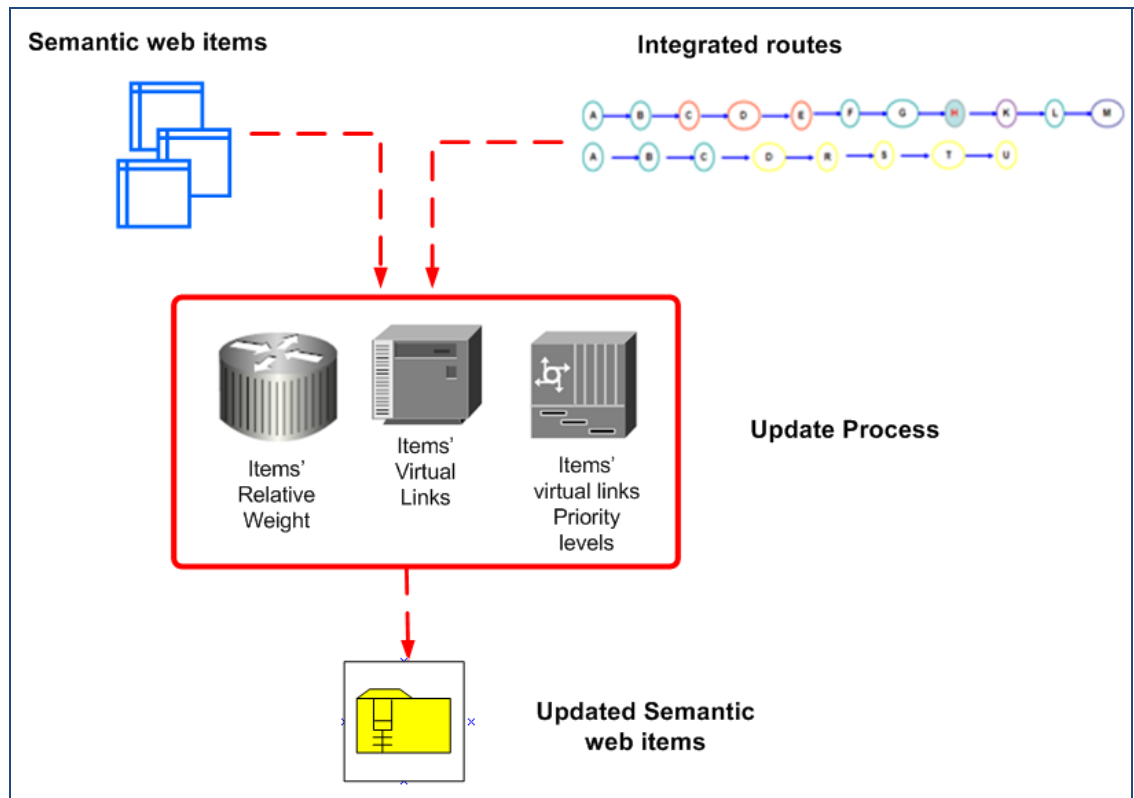


Figure 5.3: Update items attributes in semantic structure.

As illustrated in Figure 5.3, we use integrated routes to update the semantic information, so that information within the pages will continue to reflect an updated picture of the actual importance of each item to site visitors, which in turn will affect the generated recommendations.

5.3.1 Exploiting RDF/RDFS to support the concept of personal recommendation.

In the context of generating recommendation and/or personalization, as well as to find a solution to the cold start problem using semantic web concept and the active node method, we suggest adding elements to RDF/RDFS as follows:

1. **rdfs:impact**, this expresses the impact value of a web item based on users' preferences in the integrated route; this value will be updated regularly with every update iteration.
2. **rdfs:virtuallylinkedto**, this relates an item to other items that are 'virtually-linked' to it based on users' interest. A virtual link from A to B, in this context, indicates that B follows A in the integrated routes profile. It is important to mention here that for any new item this property will be empty. As soon as the item is selected by visitors in association with any other items and stored in the integrated route profile, this will be updated. What do we mean by main item? Every item associated with virtual items is a main item (or we can call it as a main class) for those virtual items.
3. **rdfs:weight**, this is used to maintain the relative weight of a virtual item based on users' interests.
4. **rdfs:priority**, this is used to maintain priority levels. A priority level can be one of: Pass, Merit, Distinction. The calculation of priority levels will be shown later.

5.3.2 The semantic update process.

During the update process, the system should perform the following steps for each web item:

1. Update the item's impact value using the calculated impact value as explained in chapter 3.
2. Find this item's virtually linked items, and the relative weights of each virtually linked item.
3. Calculate priority levels for each virtually linked item.

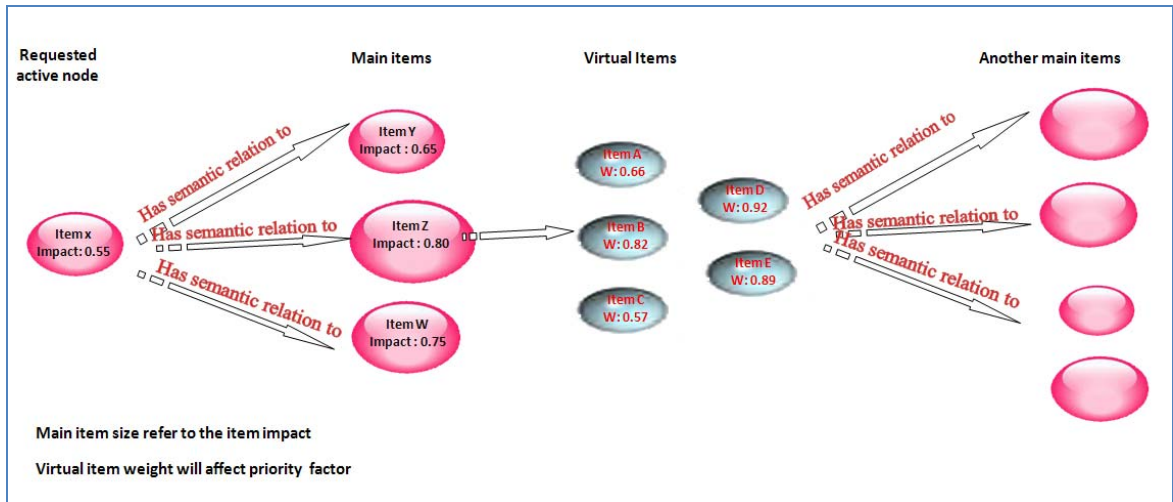


Figure 5.4: Main items' impact values and the associated virtual items' relative weights.

Figure 5.4 illustrates the relationships between 'main' items and virtual items in the semantic ANT context. In the figure, X *has semantic relation to* Y, which means that an item X is in the same semantic domain as item Y. This relates to the semantic structures that are already present at the website, independent of the recommendation system. For example, the website may have an ontology, and the location of main item X will be noted in (usually) RDF markup associated with X's page. Since items Y, Z and W have the same position in the ontology (e.g. they may sell the same category of product as item X), they are linked by this semantic relationship. However, each item also has an impact value inherited by the calculations done as part of the ANT. Note that, among the items which X has semantic relationship with, item Z has the higher impact, which means that it represents higher preferences among the site visitors. If item X is the active node, the semantic ANT will be likely to use as candidate Z. Also each item has virtual links – these reflect the links in the integrated route profile generated by the ANT. Note that item Z is virtually linked to A, B, C, D, and E, and these links have relative weights 0.65, 0.82, 0.57, 0.92, and 0.89 respectively. If X is the active node, then not only Z is a likely recommendation that will be made, but also items virtually linked to Z, and these recommendations will also take account of the weights of the virtual items, and also their priority values. How priority values are generated is indicated later in section 5.5.

As soon as the update process is complete, the system should automatically update each virtual item's priority level. An automatic tool or internal resource description framework

(RDF) method to update virtual items' priority levels is therefore required. It is important to mention here that every virtual item is also a 'main' item to another virtual one.

5.4 Some further detail, and dealing with the cold start problem

As we explored before, the user cold-start problem happens when there is a new user in the system. In addition, the 'item' cold start problem can happen when recently introduced items have no ratings. Several classical mechanisms have been tried to solve these cold start problems, but it still represents a subject of controversy and debate. We have suggested the use of active node technique as a methodology to solve the cold start problem, while at the same time respecting concerns about users' privacy, since the ANT does not depend on users' personal data to provide recommendations. In this section, we will demonstrate how we deal with cold start and related issues in the context of the semantic ANT.

5.4.1 Item preference parameters in RDF statements.

Currently the meaning of web contents is not generally machine accessible and this represents a major obstacle to providing better support to site visitors. The semantic web can provide relevant support for personalization and recommendation systems since it deals with web contents in a way that can be processed by machines, and provides suitable structures for personalization and recommendation. Moving towards the semantic web is going faster, and several languages have appeared, such as XML, RDF, OWL (the current semantic web – related W3C standards).

As indicated earlier in this chapter, we would like to add more features within semantic ontologies to help in providing recommendations based on users' preferences. The example in figure 5.5 shows how can we use RDF to describe web item(s) or thing(s), and this is shown in a more accessible form in figure 5.6.

```

<?xml version="1.0"?>
<rdf:RDF
  xml:base="file:///C:/Documents%20and%20Settings/ALIAA/My%20Documents/Altova/SemanticWorks2009/SemanticWo
rksExamples/Shop_Example.rdf" xmlns:cols="http://www.Web-implementation/Active-Node-Technique/main#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Cyber-Shot">
    <cols:camera-Pixels>8.1</cols:camera-Pixels>
    <cols:item-color> Silver</cols:item-color>
    <cols:model>C9059a</cols:model>
    <cols:price>49.99</cols:price>

    <rdfs:subClassOf>
      <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#At-and-T"/>
    </rdfs:subClassOf>

  </rdf:Description>

```

Figure 5.5: Example using RDF statements.

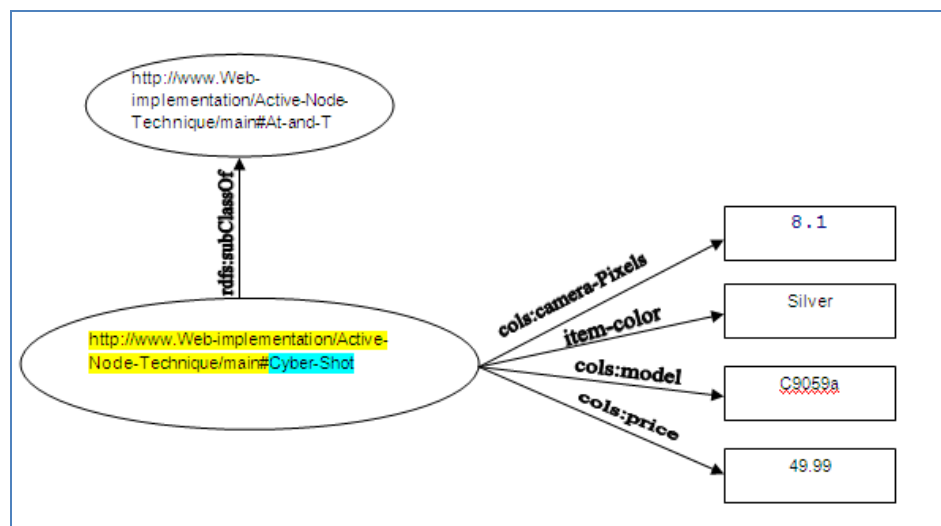


Figure 5.6: Describing a semantic item.

Figure 5.6 shows us the description-based representation of a web item using a semantic structure, which indicates the item's properties as well semantic relationships. We would like to add more features to the item to reflect users' preferences for the item, as shown in figure 5.7, and again in more accessible form in figure 5.8.

```

<?xml version="1.0"?>
<rdf:RDF
  xml:base="file:///C:/Documents%20and%20Settings/ALIAA/My%20Documents/Altova/SemanticWorks2009/SemanticWorks
  Examples/Shop_Example.rdf" xmlns:cols="http://www.Web-implementation/Active-Node-Technique/main#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Cyber-Shot">
    <cols:camera-Pixels>8.1</cols:camera-Pixels>
    <cols:item-color> Silver</cols:item-color>
    <cols:model>C9059a</cols:model>
    <cols:price>49.99</cols:price>

    <rdfs:subClassOf>
      <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#At-and-T"/>
    </rdfs:subClassOf>

    <rdfs: impact >0.00</rdfs: impact >
    <rdfs:weight>0.0</rdfs:weight>

    <rdf:Description rdf:about="http://www.w3.org/2000/01/rdf-schema#Virtuallylinkedto">
      < rdf:collection >

        < /rdf:collection >
    </rdf:Description >

  </rdf:Description>

```

Figure 5.7: Insert item preference parameters in RDF statements.

As shown previously in this chapter, we want to insert preference parameters in the created semantic items structure, which will express the relative importance of items to users as well as provide details of virtually-related items.

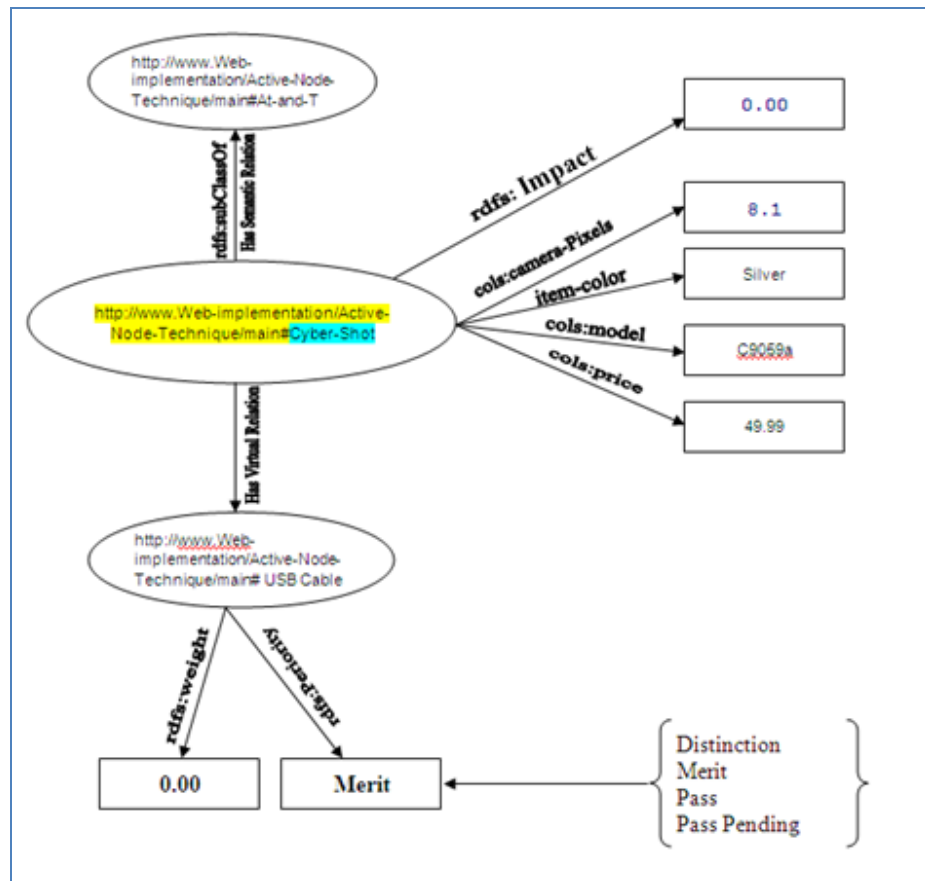


Figure 5.8: Integrated collected preferences within the semantic structure.

As we can see in figure 5.8, the main item has the semantic relation “subclassof”, and a virtual relation that expresses users’ preferences (based on what is collected and stored in integrated routes). In addition, the item’s impact and weight values (here initialized to zero) are present, along with the item’s priority level. More explanation is provided in the forthcoming sections.

We now show a more complete version of the OWL-created code for the “Cyber-shot” mobile phone example in figure 5.9.

```
<?xml version="1.0"?>
<rdf:RDF
  xml:base="file:///C:/Documents%20and%20Settings/ALIAA/My%20Documents/Altova/SemanticWorks2009/SemanticWork
sExamples/Shop_Example.rdf" xmlns:cols="http://www.Web-implementation/Active-Node-Technique/main#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Cyber-Shot">
    <cols:camera-Pixels>8.1</cols:camera-Pixels>
```

```

<cols:item-color>
  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Silver">
    <rdf:type>
      <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
    </rdf:type>
  </rdf:Description>
</cols:item-color>

<cols:model>C9059a</cols:model>
<cols:price>49.99</cols:price>
<rdf:type>
  <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
</rdf:type>

<rdfs:subClassOf>
  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#At-and-T"/>
</rdfs:subClassOf>

<owl:oneOf rdf:parseType="Collection">
  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Sony-Ericsson"/>
</owl:oneOf>

</rdf:Description>

```

Figure 5.9: A web item's semantic code.

The written code for the **Cyber-Shot** mobile phone in figure 5.9 shows a semantic description of the phone but no description of preference related or recommendation-related information – i.e. no information relating to the users who have browsed this item. We add more features to the written code to express such information, and then we can rewrite this code to appear something like what is shown in figure 5.10.

```

<?xml version="1.0"?>
<rdf:RDF
  xml:base="file:///C:/Documents%20and%20Settings/ALIAA/My%20Documents/Altova/SemanticWorks2009/Sema
  nticWorksExamples/Shop_Example.rdf" xmlns:cols="http://www.Web-implementation/Active-Node-
  Technique/main#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-Technique/main#Cyber-Shot">
    <cols:camera-Pixels>8.1</cols:camera-Pixels>
    <cols:item-color>
      <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
        Technique/main#Silver">
      </rdf:Description>
    </cols:item-color>
    <cols:model>C9059a</cols:model>
    <cols:price>49.99</cols:price>
    <rdf:type>
      <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
    </rdf:type>
    <rdfs:subClassOf>
      <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
        Technique/main#At-and-T"/>
    </rdfs:subClassOf>
    <owl:oneOf rdf:parseType="Collection">
      <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
        Technique/main#Sony-Ericsson"/>
    </owl:oneOf>
  </rdf:Description>

```

```

</owl:oneOf>
<rdf: impact >0.70</rdf: impact >
<rdf:weight>0.0</rdf:weight>
<rdf:Description rdf:about="http://www.w3.org/2000/01/rdf-schema#Virtuallylinkedto">
  <rdf:type>
    <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:type>
  <rdf:collection >
    <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
    Technique/main# Plantronics
    Voyager Bluetooth Headset ">
      <rdf:weight >0.56 </rdf:weight>
      <rdf:Periority> pass pending </rdf:priority>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
    Technique/main# USB Cable ">
      <rdf:weight >0.85 </rdf:weight>
      <rdf:Periority> Merit </rdf:priority>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
    Technique/main# Car Charger">
      <rdf:weight >0.90 </rdf:weight>
      <rdf:Periority> Merit </rdf:priority>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.Web-implementation/Active-Node-
    Technique/main# CaseCrown ">
      <rdf:weight >0.95 </rdf:weight>
      <rdf:Periority> Distinction </rdf:priority>
    </rdf:Description>
  </rdf:collection >
</rdf:Description>
</rdf:Description>
</rdf:RDF>

```

Figure 5.10: A web item with its virtual linked nodes in semantic format.

We think that in the future users will not only receive recommendations from web systems associated with particular sites, but users will be given more flexibility not only as users but as a developers too. Users will be able to program and maintain web sites in some way to match their desires. We do not refer to the currently available ways to customize sites for interface designs or colours, etc., but we refer to the contents of the website, which should be fully adaptable for any specific user.

We must differentiate between two cases. *Firstly*, creating recommendations that are restricted to the site contents. Such systems will generate recommendations from the items or contents that available in the site itself, hence recommendations can be generated and provided to users associated with the same site structure. *Secondly*, web sites in which it will be possible to capture recommendations and items from different web sites and not restricted to specific site resources; here, we can generate virtual web sites with preferences. What we mean by a user in future being able to program their web experience, is such an ability to generate a user-specific virtual web (as illustrated in Figure 5.11).

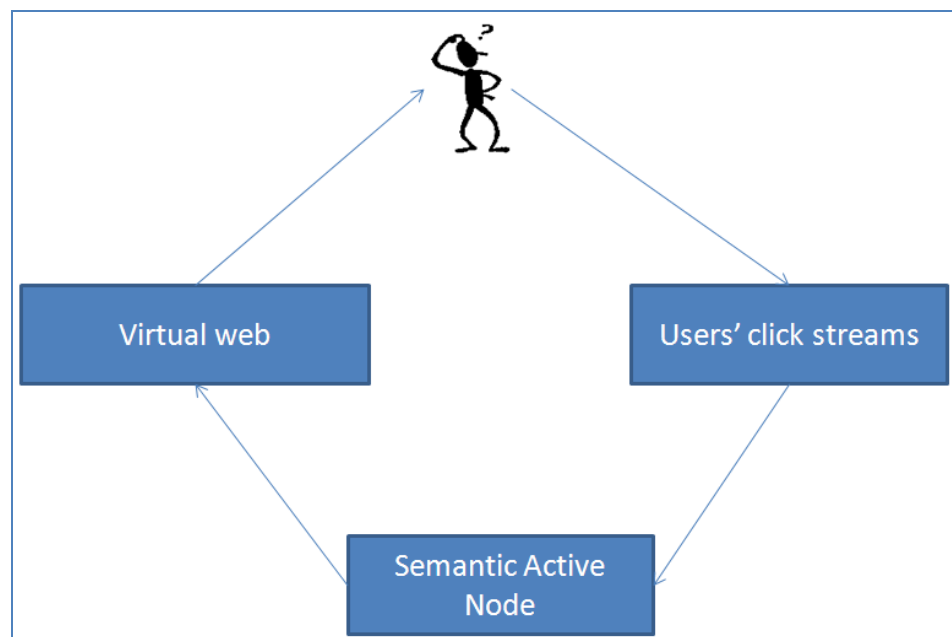


Figure 5.11: Semantic virtual web sites.

5.4.2 Dealing with new users and new items using semantic integrated routes.

Using the semantic ANT, users can participate in the creation of web contents by updating the semantic web with their preferences. As mentioned before, as soon as we create integrated routes, our system will update items' semantic descriptions by showing related virtual items (based on the integrated routes), and augment these with priority factors. Only useful items with high priority will then be involved in the recommendations. The virtual items and their weights and priorities will be revised from time to time, hence these items will not remain forever but will be updated based on users' interests. Therefore, users' activities will affect the contents shown to them.

For any new user *firstly*, the system will determine his/her selected main item (active node), and then the system will check the priority levels of all virtually linked items associated with his/her main item. *Secondly*, the system is now ready to find items that are semantically related to those virtual ones of distinct priority levels. *Thirdly*, the recommendation agent will capture the top N semantic items found in the second phase, and provide these as a recommendation set, as illustrated in Figure 5.12.

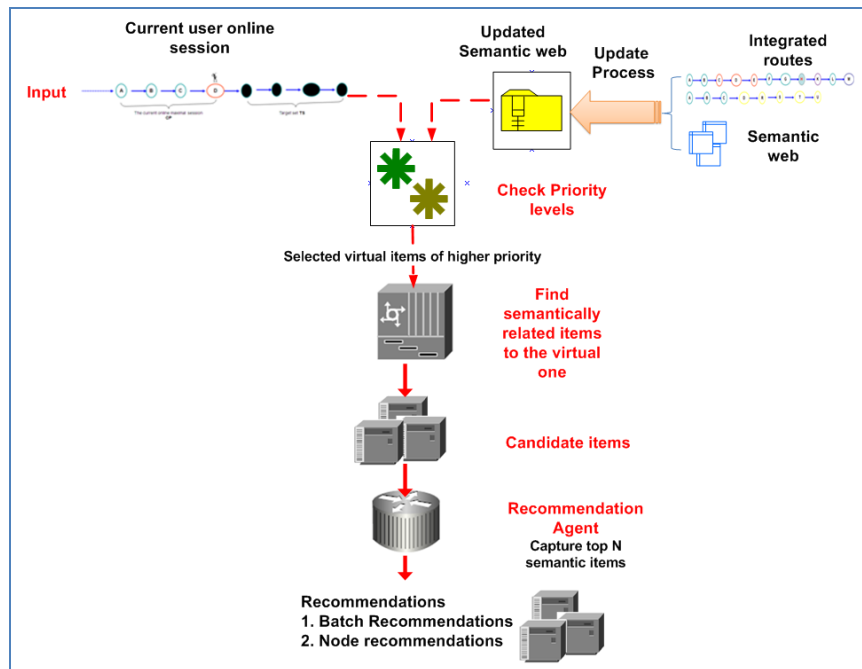


Figure 5.12: Recommendation cycle based on semantic active node technique.

New user's desires can be inferred through his/her online maximal sessions and the stored item preference parameters. Newly added items will normally be involved in the website's built-in semantic structure, and therefore they will immediately have a chance to be selected along with any other semantically related items.

5.5 The basic ideas behind node and batch recommendations in the semantic ANT.

As we have discussed, the integrated routes profile of the ANT provides links between items (pages) that we call *virtual* links in the context of the semantic ANT. So, if $A \rightarrow B$ is within the integrated route profile (generated in the normal way from site visits), we consider that A has a virtual link to B, and (as we have also discussed), this will be reflected in the RDF metadata associated with A. Meanwhile, the RDF statements (or other semantic technology used) will reveal that other items (e.g. C) have content in the same semantic category as A. In all those cases, we consider that there is a *semantic link* between A and (in this case) C. As illustrated in figure 5.13 and figure 5.15, in general there may be paths such as $A \rightarrow X \rightarrow Y$, where $A \rightarrow X$ is a semantic link and $X \rightarrow Y$ is a virtual link, and vice versa. The basic idea of node and batch recommendations; which we further explain below, is to consider recommending items that are only 1 or 2 links apart from the active node, but, in node recommendation we prefer virtual links followed by semantic links, and in batch recommendation we prefer semantic links followed by virtual links.

In order to generate semantic ANT node recommendations, we first find items that have a virtual link to the active node (this is effectively the same as what happens in the non-semantic version of ANT, since a 'virtual link' in this context means a direct link in the integrated route profile). In this chapter we call this a 'Main-to-virtual' link, and each such link has main-to-virtual priority levels that we calculate, and hence select items of high priority as candidates. Let V be this set of virtual links with high priority. In the semantic ANT we now go further and also enrich the candidate recommendation set by adding items that have semantic links with the items in V. Again, each of these semantic links, from an item in V to another item on the site, has a priority level. These are what we call 'main-to-main' priority levels, and again, only high priority ones are added to the recommendation set. The use of V provides realistic and useful node recommendations, just as in the non-semantic

version of ANT. The use of nodes semantically related to V provides extra novelty but still likely to be appropriate recommendations, since they are related closely in content to V.

In semantic ANT batch recommendation, we focus more on semantically related nodes, since the semantics of the active node is probably a good clue to the overall interests and targets of the user. However we also need to add novelty and useful recommendations guided by the integrated routes profile. The method for semantic ANT batch recommendation is therefore to also consider small paths $A \rightarrow X \rightarrow Y$, where A is the active node, but now the first link $A \rightarrow X$ is a semantic link, and the link $X \rightarrow Y$ is a virtual link (deriving from the integrated routes). We can imagine the relationships between ‘main’ items (with semantic relationships between each other) and associated virtual items as shown in figure 5.13 and figure 5.14.

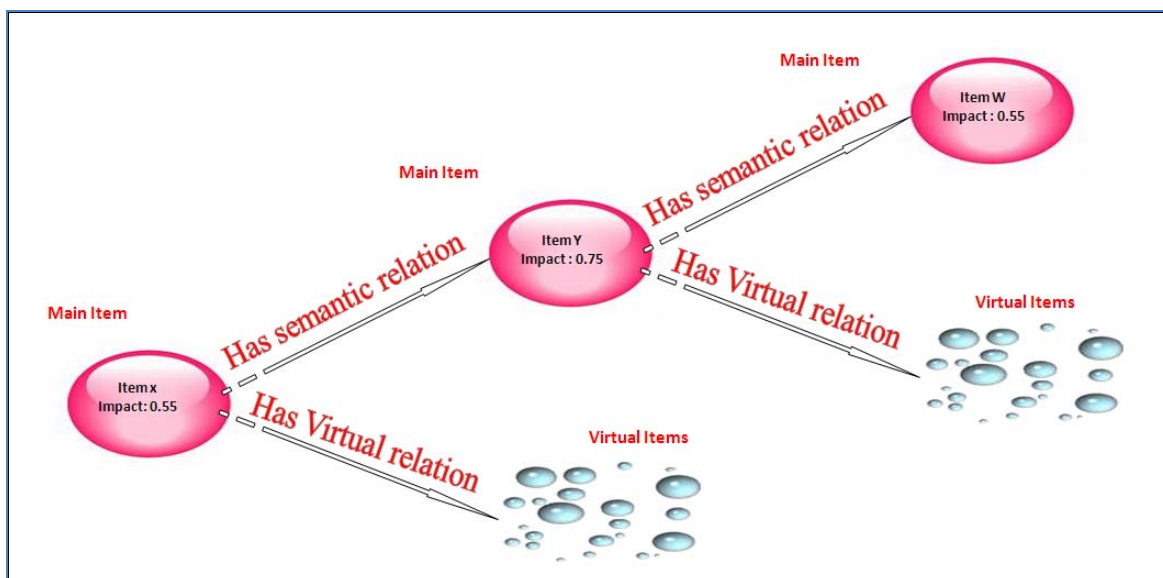


Figure 5.13: Items in semantic and virtual relations.

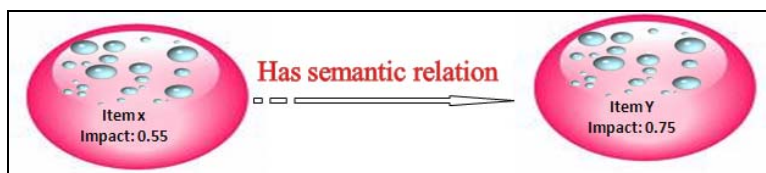


Figure 5.14: Two main items in a semantic relationship.

Figure 5.13 illustrates a case where we have semantic links and virtual links. It is important to recall that, from the viewpoint of recommendations from the current active node, the

semantically related items are associated with impact values, while the virtually related items are associated with weights. Impact values are calculated based on the level of involvement of an item in all integrated routes, while the *weight* of a virtual item, in context, is calculated based on the level of engagement, in the integrated route profile, between that virtual item and some other item. In other words, impact reflects the relative importance of an item to all site visitors, while weight reflects the relative importance of an item to the subset of site visitors who visit both this virtual item and another specific main item.

In section 5.5.1, we explained how to calculate the priority levels of virtual items. We prefer to pick items of higher priority, *Distinction* (D) and *Merit* (M), to generate recommendations. If we have no D or M priorities, then we can decrease the acceptable priority level to *Pass* (P), if again, we have no P priorities, then we are probably handling a new or rarely visited item, and we will generate recommendation based on semantic relationships only.

In all cases, recommendations will be generated based on the higher priority virtually related items first, gradually reducing the priority level until we reach *N* recommended items. Figure 5.15 shows an example in which we are generating recommendations for main item X (the active node) which has a virtual relation to item A of priority D, as well as virtual links to other items with lower priorities. Also it has a semantic link to item Y, and indirect semantic links to H, L, J and K (these are direct semantic links to the virtually linked item A). When generating recommendations for the user at node X, we consider only the virtual items that have high priority (in this case only A), and should pick all of the linked ‘main’ items that have high impact values (in our example, items (H, 0.70), (L, 0.90), (J, 0.80), (K, 0.55)).

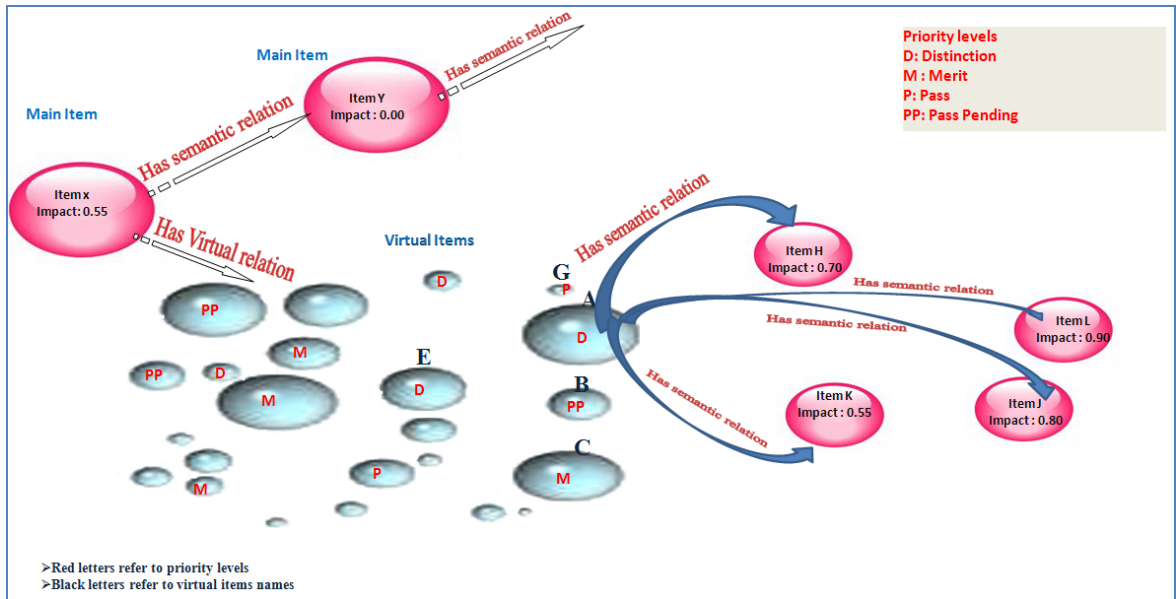


Figure 5.15: Generating node recommendations for node X.

5.5.1 Prioritization of recommendations.

In this section we will try to clarify how the various kinds of items are prioritized when making semantic ANT node recommendations in relation to a specific active node X. In this context, we need to separately consider ‘main to virtual’ links $X \rightarrow Y$ (where the ‘main’ item X is the active node, and it has a virtual link to Y), and ‘main to main’ semantic links $X \rightarrow Y$, where X is the active node and it is semantically linked to Y, or where X is any other node, and it has a semantic link to Y (e.g. X has a virtual link from the active node, but in this context is now a ‘main’ item, so we consider its impact value). We thus separately consider: ‘Main to virtual’ priority levels, ‘Main to main’ priority levels, and finally ‘collective prioritization’, which simply considers how we prioritise item Z as a recommendation for active node X in the context when we have a priority level for $X \rightarrow Y$ and a priority level for $Y \rightarrow Z$.

Main-to-virtual items priority levels

If X is the active node (or the ‘main item’), and it has a virtual link to Y, we use the following rules to generate priority values for the virtually linked items:

- If $W(\text{virtual item } Y) < \text{impact}(X)$ Then

Priority level (PL) = Pass pending

- If $W(\text{virtual item } Y) \geq \text{impact}(X)$, and $\text{impact}(X) > 0$, Then

$$\text{Priority Factor (PF)} = \frac{W(\text{virtual item}(y)) - \text{impact}(x)}{\text{impact}(x)}$$

$$\text{Priority Level (PL)} = \begin{cases} \text{Pass} & \text{if } 0.00 \leq \text{PF} < 0.50 \\ \text{Merit} & \text{if } 0.50 \leq \text{PF} < 0.75 \\ \text{Distinction} & \text{if } 0.75 \leq \text{PF} \leq 1.00 \end{cases}$$

where, $W(\text{virtual item } Y)$ refers to relative weight of item y, which should be greater than zero; $\text{impact}(X)$ refer to calculated impact value of main item x, and *Priority levels* Pass pending, Pass, Merit, Distinction express very low, low, medium, and high priority levels respectively.

Table 5.1 illustrates this by showing the priority levels for a list of virtual items associated with main item X.

Priority levels	Virtual items (Weight)				
Main item (Impact)	A, 0.97	B, 0.50	C, 0.72	G, 0.65	E, 0.87
X, 0.55	D	PP	P	P	M

Table 5.1: Main-to-virtual item priority levels.

Main-to-main items priority levels

When an item X has a semantic link to item Y, we find a priority levels for Y as follows:

➤ **If** $impact(Y) < impact(X)$ **Then**

Priority level (PL) = Pass pending

➤ **If** $impact(Y) \geq impact(X)$ **Then**

$$Periority\ Factor\ (PF) = \frac{impact(y) - impact(x)}{impact(x)} + 0.5$$

$$Priority\ Level\ (PL) = \begin{cases} \text{Pass} & \text{if } 0.00 \leq PF < 0.50 \\ \text{Merit} & \text{if } 0.50 \leq PF < 0.75 \\ \text{Distinction} & \text{if } 0.75 \leq PF \leq 1.00 \end{cases}$$

Using the situation illustrated in figure 5.15, but where we now consider A to be a ‘main item’ with impact of 0.60, we use the above rule to get the main-to-main priority levels for H, J, K and L as shown in Table 5.2.

Priority levels	Main items (Impact)			
Main item (Impact)	H, 0.70	J, 0.80	K, 0.55	L, 0.90
A, 0.60	P	P	PP	M

Table 5.2: Example showing main-to-main priority levels.

Collective prioritization

To complete the picture, we now have to consider how to prioritize each item Z which is linked in two steps to the active node X, where $X \rightarrow Y$ is a virtual or semantic link and $Y \rightarrow Z$ is also a virtual or semantic link. In general, after the calculations done above the first link will have priority either D, M, P or PP, and the second link will also have priority either D, M, P, or PP. When we consider both links together, in order, it will have any of the 16 priority profiles, ranging through D-D, D-M, ..., P-PP, PP-PP. E.g. “D-P” means that the link $X \rightarrow Y$ as priority D and the link $Y \rightarrow Z$ has priority P. In order to place a priority order

over such items, Z, that are two links away, we simply prioritize these profiles as follows (highest priority first): D-D, D-M, M-D, M-M, D-P, M-P, D-PP, M-PP, P-D, P-M, P-P, P-PP, PP-D, PP-M, PP-P, PP-PP. However we should note that using the priorities D-D ... M-M only was implemented in the algorithm specifications and tests discussed later. Restricting to priorities of M-M or above was sufficient in these tests, however in theory variations of the approach could use lower priority recommendations as and when necessary.

5.5.2 The semantic ANT node and batch recommendation algorithms.

Now we can specify the algorithm we use for semantic ANT node recommendation. Suppose that the active node is A. If we are in *node recommendation* mode, then:

1. Find the set of items (V) that are virtually linked to A and have priority D or M.
2. If V is empty (A must be a new or rarely visited item), then consider only items that are semantically related to A and have priority D or M, and place these into set R.
3. If V is not empty, find items of priority D or M that are semantically related to the items in V; let these items be set R.
4. If R is empty, call the *batch recommendation* algorithm.

Following this, the set of items that constitute possible node recommendations for A are the items in set R.

We can similarly specify semantic ANT batch recommendation as follows. If the active node is A, and we are in batch recommendation mode, then:

1. Find the set of items (S) that are semantically linked to A and have priority D or M.
2. Find items of priority D or M that are virtually linked to the items in S; let these items be set R.
3. If R is empty, call the semantic node recommendation algorithm.

Following this, the set of items that constitute possible semantic batch recommendations for A are the items in set R.

In both cases, the system then chooses from the set R to generate recommendations for the user. The system will generate the top *N* recommendations, and if there are less than *N* items

in R , then only these will be shown. If there are more than N items in the set R , then *collective prioritization* as described above will come into play, and the top N will be chosen, breaking ties randomly.

5.6 Comparison and Evaluation.

In order to evaluate the semantic ANT approach, we set up a separate semantic structure website for the collected nodes from the Alarabiya website. We first determined the main classes or domains (news, shopping, sports, business, technology, etc) of the site. Each domain's properties were generated and each item in each domain was generated along with its associated properties which are node date, title, hasvirtualrelation, nodeWeight, and impact. Each item of a specific domain is a subclass of another domain based on the OWL language structure (the semantic relations created in the OWL syntax).

The generated semantic structure was converted into XML file format to be used for further processing, and then as soon as users browse the web site, we collected their clickstream data in the standard way (as with the ANT) to generate integrated routes, as well as updating the semantic information in the generated XML file in the ways that have been described. In a training session, 264 users were involved, and we evaluated novelty, precision, and coverage for the generated recommendations for both semantic ANT, and non-semantic ANT. and the results were as shown below.

Before we show the results, we show some examples illustrating the semantic structures involved in the implementation of this experiment. In figure 5.16 we see an example of some of the classes generated in order to implement the semantic ANT.

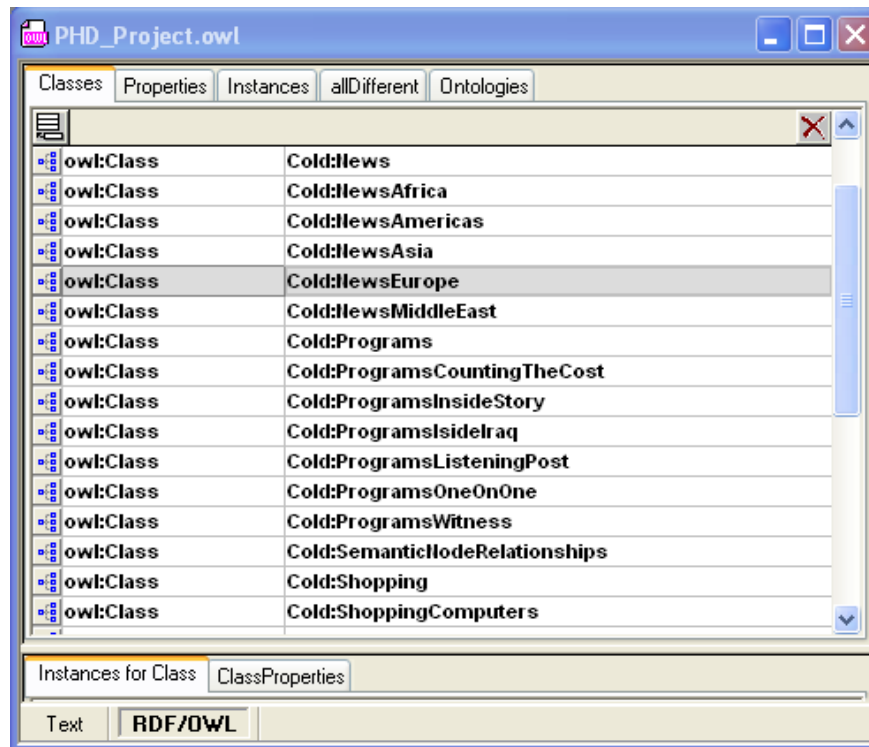


Figure 5.16: Sample of the generated semantic classes.

Meanwhile, Figures 5.17, 5.18, 5.19 show a super class called “News” that is a subclass of “WebNode”, and “NewsEurope” which is a sub class of “News”, and “PanicInTheEurozone” which is an instance of the “EuropeNews” class.

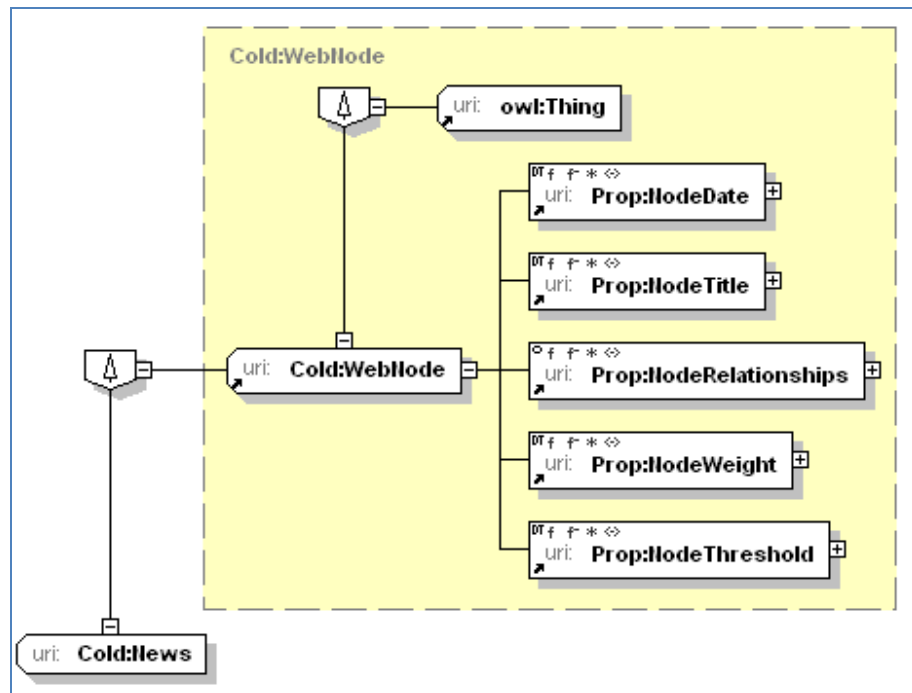


Figure 5.17: A web node's semantic properties.

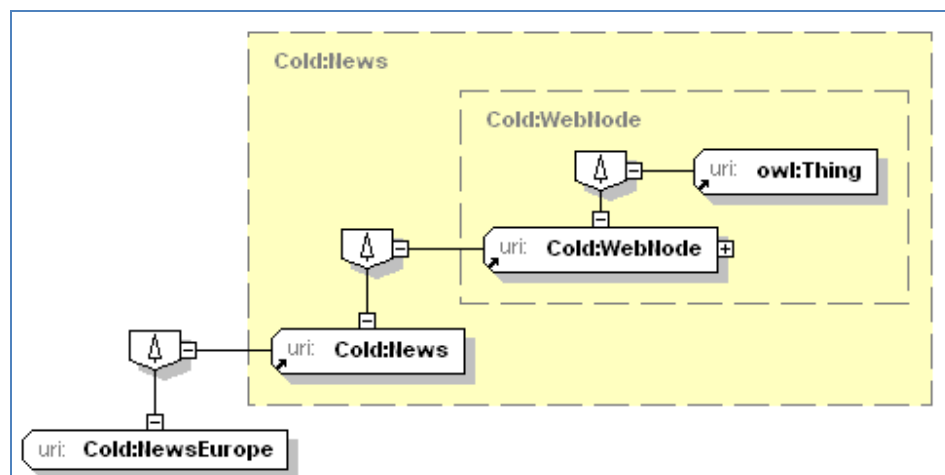


Figure 5.18: News node as a super and sub classes.

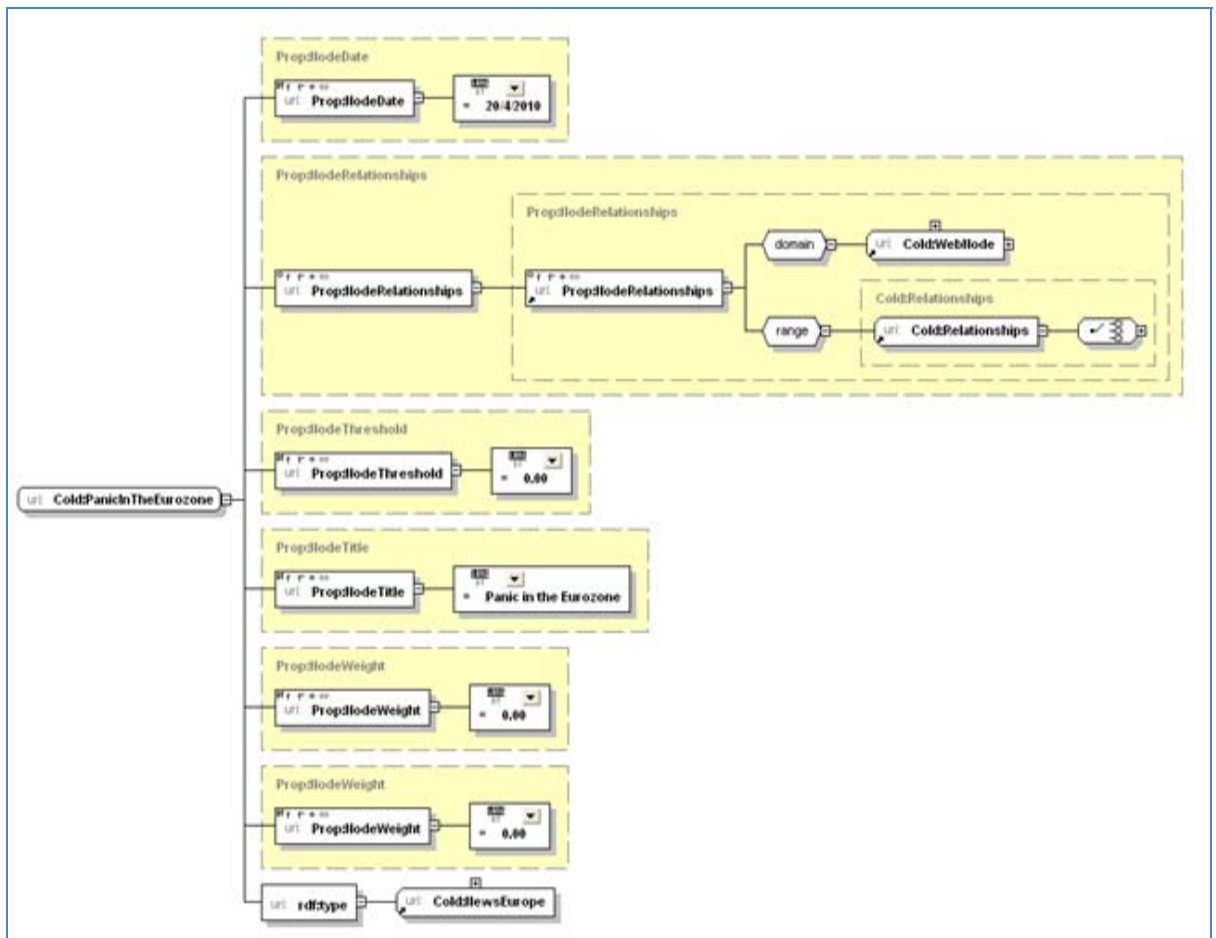


Figure 5.19: A node associated with its properties.

Figure 5.19 shows node-associated properties; as shown, the node's initial impact and weight are given value zero, and then, based on users' preferences these values will change. Each node has semantic and virtual relationships, where an item's semantic value is affected by its calculated impact, while the item's virtual value is affected by the calculated item weight in its integrated route. Figure 5.20 shows an item with its semantic and virtual relationships.

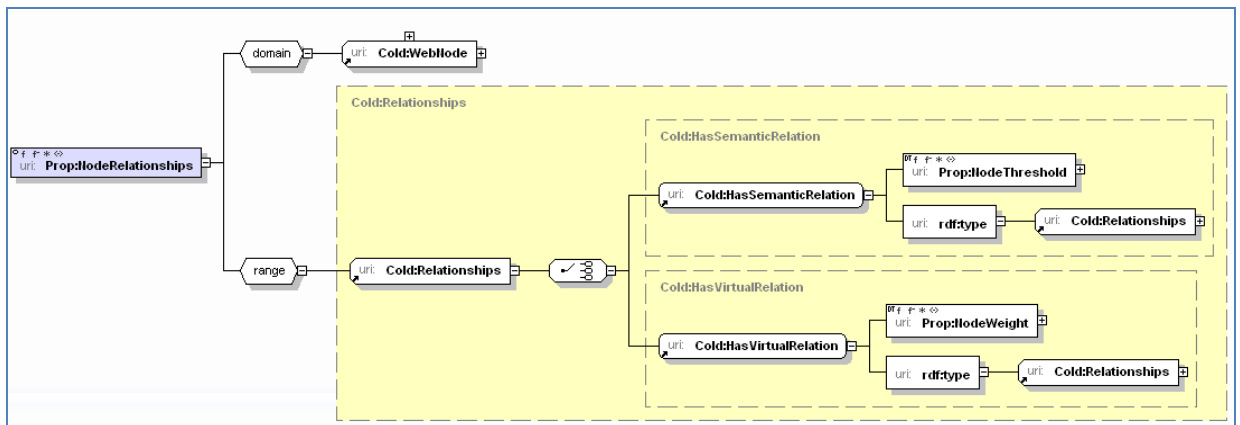


Figure 5.20: A node in semantic and virtual relationships.

The generated XML for the semantic ontology structures are then used for further processing in order to update according to users' preferences, and then to generate recommendations using the semantic and virtual relationships.

Figure 5.21: An XML structure of the generated semantic ontology.

Tables 5.3, 5.4, and 5.5 show the results of our comparison study between the previous (non-semantic) ANT and the semantic ANT in terms of novelty, coverage, and precision on the Alarabiya website.

Novelty									
Methodology		Number of Visits							
		≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Non-semantic	Active Node (Batch Recommendation)	0.75	0.7	0.69	0.64	0.67	0.72	0.76	0.82
	Active Node (Node Recommendation)	0.65	0.63	0.59	0.55	0.54	0.6	0.62	0.65
Semantic	Active Node (Batch Recommendation)	0.8	0.79	0.75	0.76	0.84	0.86	0.87	0.89
	Active Node (Node Recommendation)	0.77	0.75	0.78	0.79	0.74	0.77	0.79	0.83

Table 5.3: Semantic and non-semantic active node percentage of novelty.

As shown by table 5.3 and by the figure 5.22, the novelty values of the semantic ANT method are better than that of the non-semantic ANT. Both batch and node recommendations from the semantic ANT achieved higher novelty than the non-semantic ANT recommendations, but the biggest difference is between the node recommendations – that is the improvement of semantic ANT node recommendations over non-semantic ANT node recommendations is higher than the difference between semantic ANT batch and non-semantic ANT batch. This is not very surprising, since semantic ANT node recommendations include all of the next-step nodes from the integrated routes (just as with the non-semantic ANT), but then add to this extra candidates via semantic links. It is important to see now if this extra novelty comes with any degradation in precision or coverage.

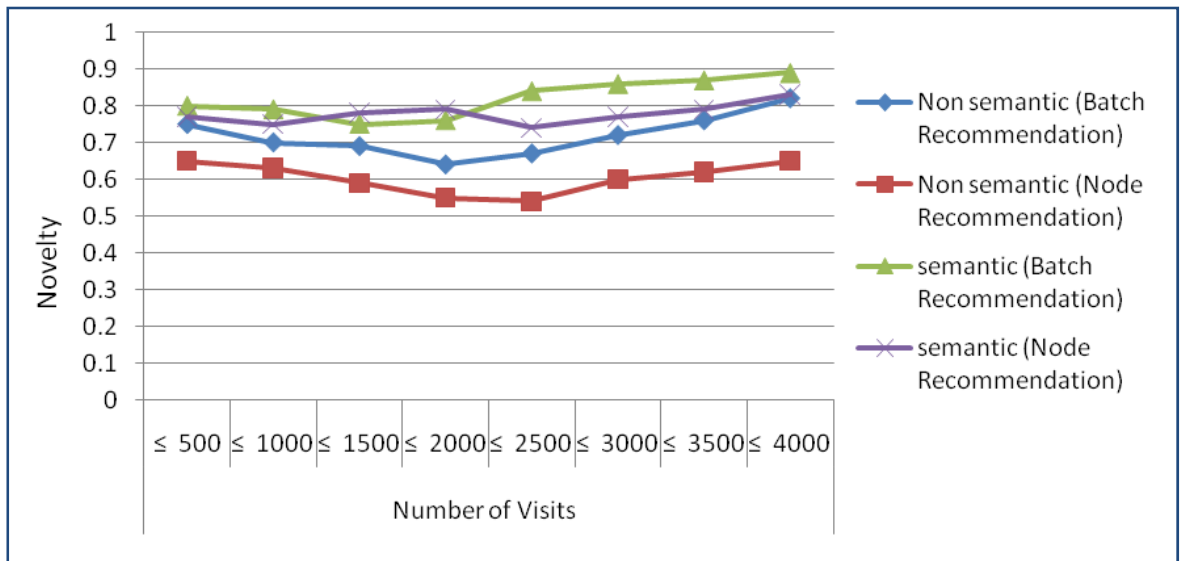


Figure 5.22: Semantic and non-semantic active node novelty.

Table 5.4 and figure 5.23 show the coverage levels for semantic and non-semantic active node recommendations. Clearly the semantic ANT node recommendations achieved better coverage than the other approaches, with semantic ANT batch recommendations in third place.

Coverage									
Methodology		Number of Visits							
		≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Non-semantic	Active Node (Batch Recommendation)	0.54	0.58	0.63	0.65	0.69	0.69	0.73	0.77
	Active Node (Node Recommendation)	0.9	0.84	0.8	0.79	0.82	0.85	0.89	0.93
Semantic	Active Node (Batch Recommendation)	0.75	0.73	0.77	0.71	0.74	0.76	0.79	0.83
	Active Node (Node Recommendation)	0.94	0.93	0.89	0.95	0.93	0.88	0.94	0.96

Table 5.4: Semantic and non-semantic active node percentage of coverage.

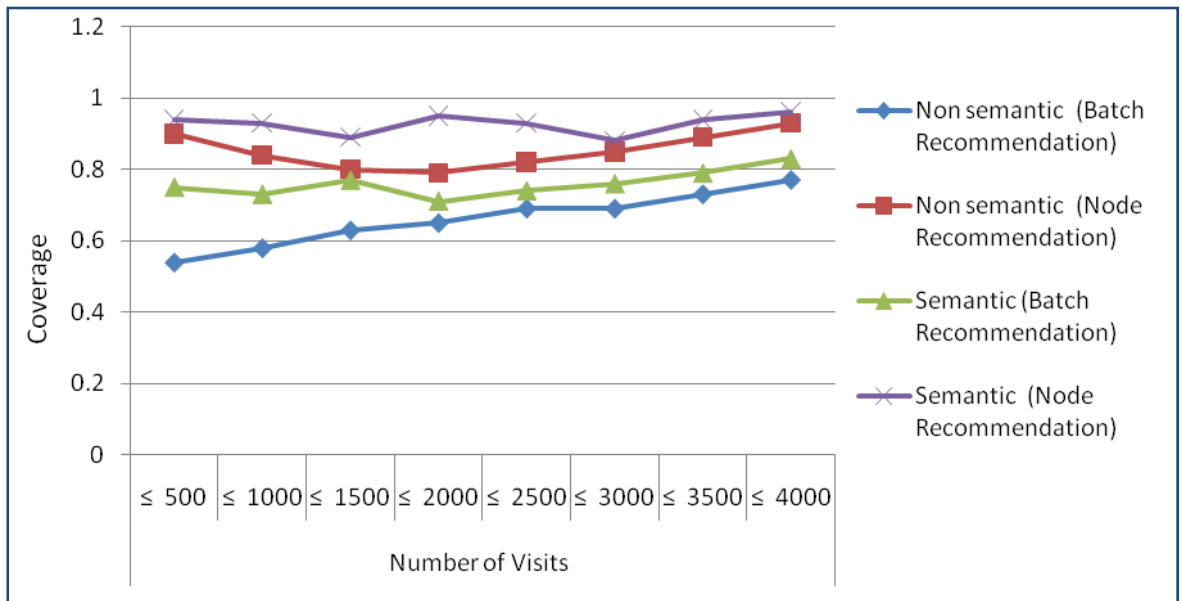


Figure 5.23: Semantic and non-semantic active node coverage.

With coverage levels increased in the semantic node ANT recommendations, we can expect precision to be increased too. As we can see in table 5.5 and figure 5.24, this is the case. It is worth remembering that novelty levels are fairly independent of coverage and precision, since the novelty of a recommended item is based on how much the item is repeatedly recommended to the same user during his visit, while coverage and precision are based on the match between recommended items and the target sets. Coverage and precision are therefore related, of course, where coverage indicates how much of the user's actual visited pages (in the training period) are covered in the recommendation sets, while a high precision means that not many items were recommended that were not also visited.

Precision									
Methodology		Number of Visits							
		≤ 500	≤ 1000	≤ 1500	≤ 2000	≤ 2500	≤ 3000	≤ 3500	≤ 4000
Non-semantic	Active Node (Batch Recommendation)	0.51	0.55	0.60	0.62	0.66	0.66	0.70	0.74
	Active Node (Node Recommendation)	0.87	0.81	0.77	0.76	0.79	0.82	0.86	0.90
Semantic	Active Node (Batch Recommendation)	0.72	0.70	0.74	0.68	0.71	0.73	0.76	0.80
	Active Node (Node Recommendation)	0.91	0.90	0.86	0.92	0.90	0.85	0.91	0.93

Table 5.5: Semantic and non-semantic active node percentage of precision.

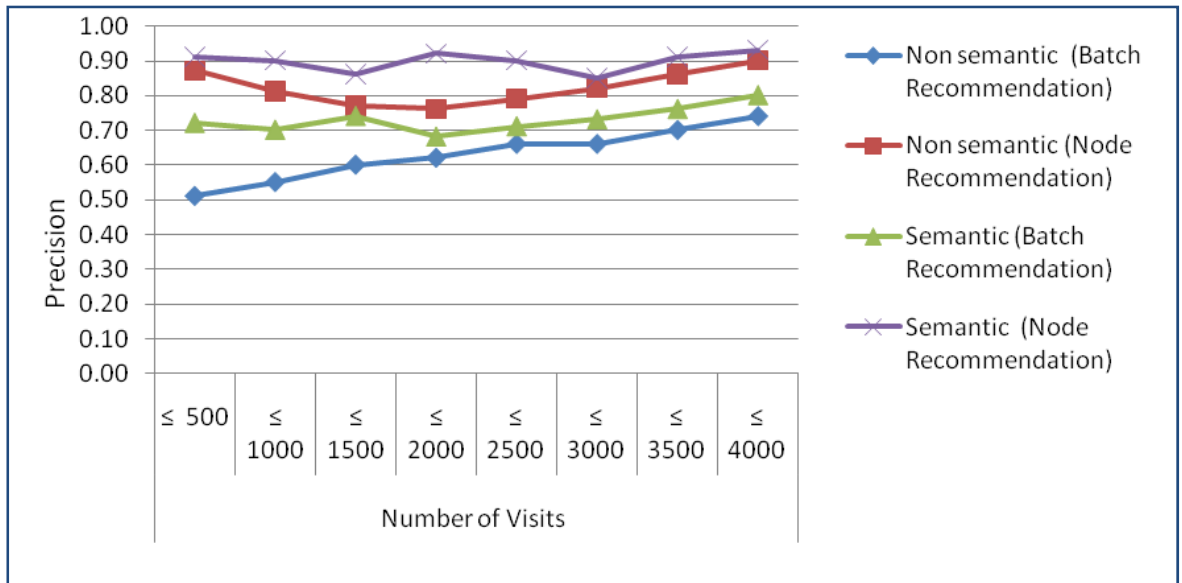


Figure 5.24: Semantic and non-semantic active node precision.

The evaluation experiment suggests that semantic ANT recommendation is quite successful in comparison to the non-semantic ANT (and therefore by implication its performance is strong compared to the alternative methods tested in chapter 4). In particular semantic ANT node recommendation seems to be the best-performing method. The semantic ANT batch recommendation method performs better than the non-semantic version of batch

recommendation, but not as well as the non-semantic version of node recommendation. The basic idea of semantic ANT batch recommendation is that the semantic category of the user's current node is a good clue to their general browsing targets, so the recommendations are based mostly on the semantic links from the current node. However it turns out that this does not have particularly strong performance. This could be because the basic idea is only sometimes true, and in other times it provides misleading directions. Alternative versions of this will be worth studying. On the other hand, semantic ANT node recommendation, which maintains a key part of the non-semantic ANT node recommendations (recommending the higher priority next-step links from the integrated routes profile), and then further enriches them with semantic linked nodes, seems to have been a promising idea. Again, this could be further explored in future work.

Chapter 6

Conclusion & Future Work

6.1 The summary

Internet users currently face problems of information overload due to rapid growth in the volume of information and the number of web users. Therefore, helping online users to receive appropriate items and information in reasonable time is becoming a critical issue in web applications. In this thesis, we aimed to address two important problems, the cold start problem and the privacy problem. We highlighted the different methodologies used to solve each problem and demonstrated criticisms of the previous approaches. We then described and evaluated the Active Node Technique (ANT) which achieves good recommendation (in terms of novelty, coverage and precision), without violating privacy concerns.

As mentioned before, web personalization refers to the process of automatically customizing the content and/or the structure of a web site to the specific and individual needs of each user without asking for his needs explicitly. This has been achieved by taking advantage of the user's navigational behaviour, revealed through processing of the web usage logs and/or by using users click streams. In particular, the ANT approach implicitly discovers web usage patterns that emerge from the whole collection of users that visit the site, and the recommendations that arise from ANT adapt and change overtime as users' interests (collectively) adapt and change over time. As we have seen when evaluating the approach, this leads to appropriate recommendations both for new users (and new items) and for established users. The ANT approach; introduced in this thesis, is therefore recommended as a solution to the cold start and privacy problems for providing web users with personal recommended items, i.e. web personal recommendation.

In more detail, we first explained the framework for data collection, which leads to collecting 'maximal online sessions' that are sequences of visited items (pages) and contain no repeats. We then discussed and presented how to try to ensure that only 'significant' maximal sessions are kept for further processing and use. To reduce the storage requirements, without a significant negative effect on the value of the stored information, we use an absorption process (if a session is a sub-sequence of another session, we only store the latter session), and we try to make sure that the relative weights of items are modified in an appropriate way during this process. The resulting 'Integrated routes' is used to infer the future paths that may be followed by users, given their current browsing behavior.

The integrated routes profile can be used for two types of recommendation: batch recommendation (a kind of ‘jumping ahead’ recommendation) and node recommendation (focused on the likely next nodes that user’s might visit from the current page). In batch recommendation, N items of higher relative weights are recommended to the user, where these items come from points ahead on the continuation of the user’s current path, as suggested in the integrated routes profile. In node recommendation, N items of higher relative weights are selected for recommendation, but these are restricted to ‘next steps’ from the current active node.

In section 1.8.3, we indicated the main contributions of the thesis. We provide them again below, indicating where in the thesis they have been described and justified.

- A novel solution to the cold start problem (both items and user cold start), which is introduced and explained in chapter three, and tested against three other alternative methods in chapter four. This is the Active Node Technique (ANT).
- The same technique also serves to solve the privacy problem in personal recommendation systems, in the sense that good recommendations are provided, without the need to ask for and/or use user IP addresses or any personal user data; this is introduced and explained in chapter three and tested in chapter four.
- Metrics are introduced to measure recommendation novelty, as well as coverage and precision, which are introduced in chapter three and implemented in chapter four.
- We provide a novel way to improve recommendations in the context of a semantic web environment, in the form of a way to combine the ANT with semantic web structures. This was explained and evaluated in chapter five.

The remainders of this chapter is as follows: In section 6.1.1, we summarise how the ANT is used to solve the cold start problem. In section 6.1.2 we indicate how is the ANT provides good inferences about users’ browsing targets, without using their personal data, and then in section 6.1.3, we argue that the ANT is domain independent. In section 6.2, we provide our overall conclusions, and in section 6.3, we consider a selection of important avenues of future research.

6.1.1 The active node technique and the cold start problem.

The user cold start problem happens when a new user visits the web; in traditional recommender systems, this is a problem since the system has no data about his/her preferences. When using the ANT, however, the system already has an integrated routes profile built from many previous user visits. The new visitor will follow a specific path(s) on the web site during his/her visit, and the ANT will quickly be able to generate useful recommendations based on the match between the user's browsing behavior and the stored integrated routes.

The item-based cold start problem happens when new item(s) are added to the web site. In traditional systems, since these items have not been rated or visited, it is problematic to include them in recommendations. In the ANT, we solve this problem by using the link structure on the website. New items will inherit (in essence) the weights of established items that link directly to them, and also new items are promoted among the recommendation set, to help generate experience and valid ratings for them. In the case of the semantic ANT, the inbuilt semantic links provide extra help in ensuring that appropriate recommendations are made for new items.

6.1.2 The active node technique and user privacy issues

In some recommendation systems, user identification is necessary to distinguish among different users. However this introduces many difficulties such as a *single IP address / multiple server sessions*, where internet service providers (ISP_s) have a pool of proxy servers that users use to access the web. A single proxy server may have several users accessing a web site potentially over the same IP address. *Multiple IP address / single user* also causes problems, where the same user may take several IP addresses on each request. In addition, a user that accesses the web from different machines will have a different IP address from one session to another, while a user that uses more than one browser even on the same machine will appear as multiple users.

Users can also be distinguished by using demographical data through registration and authentication mechanisms, or by using client side cookies. But cookies are often disabled or

deleted. It is possible to use a combination of IP address and any other available information that helps to distinguish between users.

Using the ANT, however, there is no need to collect personal information (name, age, and address), or user IP address. We only detect his/her online web maximal sessions (in the current visit only), and then match these to stored integrate routes. Figure 6.1 illustrates a user during his online maximal session; the ANT will treat the user as an abstract user, and then the system will generate proper recommendations based on inferring his or her browsing targets based on the current session and the stored integrated routes. If and when a specific user deletes all cookies or changes his or her IP address, this has no effect on the ANT.

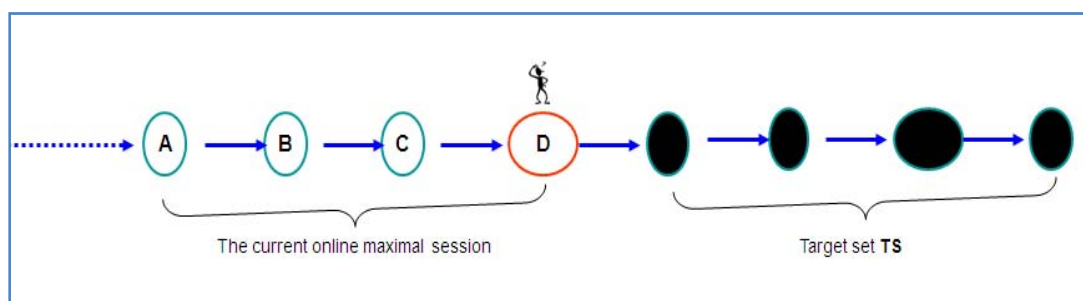


Figure 6.1: Illustrating a user's online session.

In general, a privacy problem arises from any method used to identify users, particularly if this is personal data or even the IP address. The ANT does not need to identify users, in such a way, but it makes the most out of the information that a user naturally and easily provides in terms of their sequence of page visits (and the associated time duration information) on the site. The main research question that we have examined is whether this alone is sufficient to provide good recommendations, and we have found, by evaluating and comparing with methods that use demographic data, that the recommendations provided do compete very favourably against other methods that are intrusive in relation to privacy concerns. To summarise and comprehensively state the ways in which the ANT does not isolate privacy concerns, we provide the following points:

1. No personal data are collected when using the ANT.
2. Users receive recommendations based on their online maximal selections, therefore he/she will receive recommendation only when they are online.

3. The data collected and stored by the ANT relates only to user's sequences of page visits, and contains nothing that can identify users.

6.1.3 Domain independence.

The evaluation of the proposed technique has been done in the context of a news website, however we wish to argue that the ANT provides a framework to generate appropriate recommendations in any domain prevalent in web services applications, such as E-Learning, E-commerce, News web applications, and so on. There is nothing domain-dependent about the ANT processes, and we believe it is intuitively reasonable to suppose that it is applicable on any type of website. For example, in e-commerce applications and all similar application, we can use batch recommendations as a good choice for generating recommendation, since these tend to help save the user time in finding what they ultimately wish to purchase. For example, mobile phones have a semantic relationship to headphones, chargers, and batteries, etc., therefore when using batch recommendations in the semantic ANT context, the candidates that will be used to generate recommendations will be from those semantically related items. For E-learning applications, node recommendation is arguably the best choice, since they provide appropriate 'next steps' that are validated as good choices via the integrated route profile. Also, in the semantic ANT context, these recommendations will be based on semantically related nodes as well as virtually linked nodes. For example, if someone studies a C++ course, he/she can receive a recommendation to read about other programming language such as C#, visual C...etc , as well as recommendations to read journals or magazines about programming challenges. Although, the ANT is a domain independent, but it might needs some adaptation in the implementation steps e.g. using ratings instead of using spent time per page, then we can use the suggested ratings equations 4.13. In medical websites we can use the ANT semantic structure, where diagnosing of specific diseases; which required some medical tests, are semantically related to the diagnosing of some another diseases with another medical tests.

6.2 Conclusions

Several techniques have been used to solve the cold start problem as indicated before, but these techniques each provide only part of the solution. Some techniques solve the items cold start problem, but not the user cold start problem, or *vice versa*, or in some cases the solution

to these problems suffer from privacy issues. The proposed active node technique overcomes several drawbacks of other techniques, and provides a framework for the cold start problem (item cold start and user cold start), as well as taking privacy concerns well into consideration. Several benefits are accomplished by implementing the active node technique that can be summarized as follows.

1. Solving the user cold start problem (by providing appropriate inferences of browsing targets, thanks to the stored integrated routes, very quickly after the new user has started browsing)
2. Solving the item cold start problem, via various aspects of both the ANT and the semantic ANT that pay special attention to promoting relevant new (or unvisited) items.
3. Low computation time overhead during the recommendation phase and low storage requirements compared with many other methods.
4. Flexibility to use either node or batch recommendations, including combining and switching between the two types.
5. Adaptation of recommendations, which are kept fresh and up to date with good levels of novelty.
6. Avoiding privacy concerns.
7. Achieving good quality recommendations (as shown in the experiments in this thesis), at the same time as achieving the other benefits above.

6.3 Future work

While there are many open research problems in personal recommendation systems, this thesis suggests answers to several questions related to the cold start and privacy problems in these systems. However there remain many open questions, some arise from how the performance of ANT may vary in different contexts, and some arise from how such systems can best take advantage of the new opportunities provided by semantic web technologies. We briefly consider below three broad issues that we find of particular interest. Respectively, these concern further evaluation of the ANT in different environments, significant extension to the ANT to make it more adaptive to work well in different environments, and the continued opportunities arising as the semantic web grows.

1. The performance of recommendation and personalisation systems is important, especially in the context of e-commerce applications, since the revenue of a site depends on how well it can maintain the interest of users, and also save their time in finding things of interest to them. Even in a non e-commerce website, the retention and constant stimulation of users is still important, since many such websites gain revenue from advertising within their pages. We have evaluated the ANT using just one specific website (see section 5.4), and involving a limited number of users. We have also argued that, from an intuitive viewpoint, the system should work well in other types of website. However it will be revealing for ourselves or others to do further future work that evaluates the ANT technique (and in comparison with other techniques), on different types of website. Different types of website provide different contexts in which the relative performance of the ANT might vary. For example, if the sitemap is broad and shallow (paths tend to be brief, and individual pages have many links), the integrated routes will be short, and there will be more emphasis on the impact values and weight values to ensure good recommendations. On other websites, if the number of visitors is quite small, there is not much information in the integrated routes, and the performance of ANT might be little or no better than other methods. It would be interesting to know how the performance of the ANT depends on the numbers of users and visits. Even with too many users and visits, when the preferences are very wide and varied, in some cases the integrated routes profile could be confused, and unable to offer well-targeted recommendations for the current user.

There is one issue in which we have made some progress in ongoing work. In some contexts it is a common problem that some users behave in a way that misleads (maybe deliberately) the recommendation system, applying false ratings or exercising misleading click-streams. In the ANT that we have described, there are features built in to the session-significance calculations (ignoring too short or too long sessions, or sessions where some page durations are too long or too short) that help to keep only sessions likely to be valid. In some recent work (tested on the MovieLens database) (Embarak and Corne, 2011) we explore an extension to this which considers the variance in the ratings supplied by a user in different domains of the site, and which classifies users in a number of classes (e.g. ‘untrustworthy’) – this has worked well in

the Movielens database, in the context where user's supply ratings. The method could also be incorporated in the ANT, adapted to classify users based on the variance and amount of significant and insignificant sessions they create.

2. The semantic ANT method described in chapter five included many design decisions which could be varied and explored. In the node and batch recommendations respectively, candidates for recommendation were only taken from virtual links followed by semantic links, or semantic links followed by virtual links. It would be possible to go to further depth in the (combined semantic and virtual) link graph. An interesting idea worth considering is to explore adaptive ANT and adaptive semantic ANT methods. For example, in the adaptive semantic ANT, there are parameters that control the priority levels given to different parts of the link graph. These parameters can change over time by a reinforcement learning approach, guided by trying to achieve high amounts of user visits on recommended pages. The same approach can be used for all of the parameter choices that we have fixed so far in our exploration of the ANT and the semantic ANT. For example, an adaptive basic ANT can adapt over time the threshold values that it uses for determining whether or not a session is significant.
3. The semantic ANT and the possibilities of the semantic web offer many future directions. One of these is the issue of integrating the information across different ontologies from different websites, or maybe between different parts of the same website (dealing with different categories of products). A related problem is in the consistency of ontologies – e.g. two different website may sell only mobile phones and accessories, but using completely different terms and structures for their ontologies. Another general problem is scalability – as more and more websites exploit ontology information within their pages and metadata, the opportunities grow for integrating and reasoning across these different sites, but the techniques used for integration and meta-reasoning clearly need to be scalable. There are several research efforts that go towards all of these directions from different angles – we note that the needs of and opportunities for recommendation and personalisation systems should be seriously considered in these efforts.

Appendices

A. Technical user click streams analysis report

A.1 Access Resources

- A.1.1 Top Access Pages
- A.1.2 Single Accessed Pages
- A.1.3 Number of Hits Per Page
- A.1.4 Top Entry Pages
- A.1.5 Top Exit Pages

A.2Visitors Activities

- A.2.1 Top Visitors by Number of Visits
- A.2.2 Visitors who visit once
- A.2.3 Repeated visitors
- A.2.4 Average duration per visitors
- A.2.5 Average visits duration for all visitors
- A.2.6 Top Visitors by Duration
- A.2.7 Number of unique visitors

A.3Site Navigation

- A.3.1 Visitors popular paths through the web site
- A.3.2 Max Path Length
- A.3.3 Min Path Length

B. Suggested methodology modules

- B.1 Data Flow Diagram Level (1)**
- B.2 System Flow Chart**
- B.3 Data preparation Flow Chart**
- B.4 System pattern discovery flow chart**
- B.5 System recommendation flow chart**

C. Abbreviations

A. Technical user click streams analysis report

A.1 Access Resources

A.1.1 Top Access Pages

Top Access Pages are pages that mostly accessed by visitors

<i>Top Access Pages</i>	Page Title	Number of hits
	intro	193
	index	108
	dep	50
	chairman	48
	Manufacturing_Technology_Department	31
	Advanced_Materials_Department	31
	mpm07	29
	Minerals_Processing_and_Technology_Department	27
	Registration	24
	Metals_Technology_Department	24
	Training_Department	23
	AMSAT	23
	Contact_us	20
	culture_program	19
	mission	18
	IT	18
	important_dates	18
	Corrosion_Laboratory	18
	Conferences	18
	library	16
	Training_Courses	15

A.1.2 Single Accessed Pages

Single Accessed Pages are pages that are accessed only once by visitors

<i>Single Accessed Pages</i>	Page Title	Number of hits
	Bahgat	1
	Beneficiation_Activities	1
	Beneficiation_Contact_Us	1
	Beneficiation_Staff	1
	casting-Projects	1
	Chemical_and_Electrometallurgy_-_Activities	1
	Electronic_Materials_Technical_Services	1
	Facilities	1
	index_golive	1
	IT-contact	1
	KhalidHafez	1

	Minerals_Characterization_Projects	1
	Plastic-Deformation-Facilities	1
	Plastic-Deformation-Publications	1
	Plastic-Deformation-TechnicalServices	1
	Publications	1
	Pyrometallurgy	1
	PyrometallurgyContact_Us	1
	Pyrometallurgy-Projects	1

A.1.3 Number of Hits Per Page

Represent number of click streams for web site pages

	Page Title	Number of hits
<i>Number of Hits Per Page</i>	intro	193
	index	108
	dep	50
	chairman	48
	Advanced_Materials_Department	31
	Manufacturing_Technology_Department	31
	mpm07	29
	minerals_Processing_and_Technology_Department	27
	Metals_Technology_Department	24
	Registration	24
	AMSAT	23
	Training_Department	23
	Contact_us	20
	culture_program	19
	Conferences	18
	Corrosion_Laboratory	18
	important_dates	18
	IT	18
	mission	18
	library	16
	topics	15
	Training_Courses	15
	Composite	14
	contact	13
	submission	13
	casting-Staff	12
	Electronic_Materials_Laboratory	12
	Minerals_Characterization_Laboratory	12
	NonFerrous_Laboratory	12
	Technical_Services	12
	Corrosion_Equipments	11
	italy	11
	Nanostructured_Materials_Laboratory	11

AMSAT06	10
Exhibition	10
Nanostructured_Materials_Staff	10
NewsLetter	10
Instructions	9
Plastic_Deformation_Laboratory	9
RegForm	9
Accompanying	8
Ceramic_Materials_Laboratory	8
CulturalProgram	8
Devices	8
GUC	8
organizing	8
rpm-Staff	8
SubAndinst	8
Training_Courses_-2004-2005	8
Beneficiation_Laboratory	7
Chemical_and_Electrometallurgy_Laboratory	7
Composite-Staff	7
Corrosion_Activities	7
rpm	7
schedule	7
Steel_Technology_Laboratory	7
casting	6
Ceramic_Materials_-Staff	6
Corrosion_Staff	6
Mechanical_Tests	6
rpmContact_Us	6
Steel_Technology_Staff	6
welding	6
عطاءات ومناقصات	6
Ceramic_Materials_-Projects	5
Composite-Projects	5
CulturalProgramTour	5
Electronic_Materials_Facilities	5
Information	5
Minerals_Characterization_Activities	5
NewsLetter-3-2006-pic	5
PowderTechnologyLaboratory	5
rpm-Activities	5
rpm-Projects	5
Steel_Technology_Activities	5
Activities	4
Beneficiation_Technical_Services	4
casting-Contact_Us	4
casting-Publications	4
Composite-Facilities	4
Corrosion_Projects	4
Corrosion_Publications	4

Electronic_Materials_Staff	4
index_sub_front	4
Minerals_Characterization_Staff	4
NonFerrous_Activities	4
Postconferencetours	4
Pyrometallurgy-Staff	4
rpm-Publications	4
rpm-TechnicalServices	4
Staff	4
Steel_Technology_Projects	4
welcome	4
2circular	3
Beneficiation_Facilities	3
Beneficiation_Projects	3
Beneficiation_Publications	3
Chemical_and_Electrometallurgy_-Contact_Us	3
Chemical_and_Electrometallurgy_-Projects	3
Composite-Publications	3
ContactUs	3
Electronic_Materials_Activities	3
index_front	3
index_sub	3
index_sub_dream	3
IT-staff	3
Minerals_Characterization_Facilities	3
NonFerrous_Staff	3
Plastic-Deformation-Projects	3
Plastic-Deformation-Staff	3
rpm-Facilities	3
TechnicalServices	3
welding-Projects	3
AccompanyingPersonsProgram	2
casting-Activities	2
casting-Facilities	2
casting-TechnicalServices	2
Composite-Activities	2
Corrosion_Contact_Us	2
Electronic_Materials_Contact_Us	2
Electronic_Materials_Projects	2
Electronic_Materials_Publications	2
index_dream	2
index_sub_golive	2
IT22	2
kghany	2
Minerals_Characterization_Contact_Us	2
Minerals_Characterization_Publications	2
Minerals_Characterization_Technical_Services	2
Nanostructured_Materials_Activities	2
Nanostructured_Materials_Projects	2

	NonFerrous_Facilities	2
	NonFerrous_Projects	2
	NonFerrous_Publications	2
	Steel_Technology_Facilities	2
	welding-Activities	2
	welding-Facilities	2
	welding-Publications	2
	Bahgat	1
	Beneficiation_Activities	1
	Beneficiation_Contact_Us	1
	Beneficiation_Staff	1
	casting-Projects	1
	Chemical_and_Electrometallurgy_-_Activities	1
	Electronic_Materials_Technical_Services	1
	Facilities	1
	index_golive	1
	IT-contact	1
	KhalidHafez	1
	Minerals_Characterization_Projects	1
	Plastic-Deformation-Facilities	1
	Plastic-Deformation-Publications	1
	Plastic-Deformation-TechnicalServices	1
	Publications	1
	Pyrometallurgy	1
	PyrometallurgyContact_Us	1
	Pyrometallurgy-Projects	1
	Pyrometallurgy-Publications	1
	Steel_Technology_Contact_Us	1
	welding-Contact_Us	1

A.1.4 Top Entry Pages

Top Entry Pages and Top Exit Pages for site visitors (the highest start pages for all sessions)

	start Page Title	Number of hits
<i>Top Entry Pages</i>	intro	108
	index	46
	chairman	19
	mpm07	13
	library	5
	dep	4
	Corrosion_Equipments	4
	Conferences	4
	عطاءات ومناقصات	4
	italy	3

	Training_Department	3
	IT	2
	Devices	2
	Composite	2
	Contact_us	2
	Registration	2

A.1.5 Top Exit Pages

The highest end pages for all sessions)

<i>Top Exit Pages</i>	End Page Title	Number of hits
	index	65
	intro	41
	organizing	6
	topics	6
	library	5
	mission	5
	important_dates	5
	chairman	5
	Training_Courses - 2004-2005	5
	Nanostructured_Materials_Laboratory	4
	NewsLetter	4
	عطاءات ومناقصات	4
	Training_Department	4
	Manufacturing_Technology_Department	4
	Corrosion_Equipments	3
	Contact_us	3

A.2 Visitors Activities

A.2.1 Top Visitors by Number of Visits

<i>Number of Visits per Visitor(Top fifteen)</i>	User IP	Number of Visits	Total Duration/per minute
	65.54.188.61	12	75
	65.54.188.62	11	95
	192.168.2.30	10	100
	65.54.188.60	8	51
	143.248.110.60	6	35
	65.55.212.65	6	76
	66.249.65.174	6	45
	65.214.44.45	5	145
	128.194.135.94	5	63
	192.168.3.25	4	35

	65.55.208.94	4	23
	65.55.208.95	3	25
	65.55.208.96	3	20
	62.178.10.113	3	120
	65.54.165.36	3	35
	38.113.234.181	2	15

A.2.2 Visitors who visit once

One session visit, sample of twenty visitors

	User IP	Number of Visits	Total Duration/per minute
<i>Visitors who visit once</i>	192.168.4.22	1	3
	192.168.4.26	1	5
	192.168.4.35	1	9
	192.168.4.81	1	1
	192.168.5.16	1	5
	193.194.69.210	1	8
	193.194.83.169	1	7
	193.194.92.197	1	3
	193.227.29.230	1	1
	193.227.30.5	1	6
	193.47.80.42	1	5
	193.48.246.16	1	20
	195.229.236.214	1	1
	195.43.3.70	1	1
	195.97.22.113	1	14
	195.97.225.3	1	2
	196.2.124.252	1	4
	196.202.24.213	1	1
	196.202.35.195	1	2
	196.202.62.3	1	6

A.2.3 Repeated visitors

More than one session or visit, sample of twenty visitor

<i>Repeated Visitors</i>	User IP	Number of Visits	Total Duration/per minute
	65.54.188.61	12	75
	65.54.188.62	11	95
	192.168.2.30	10	100
	65.54.188.60	8	51
	143.248.110.60	6	35
	65.55.212.65	6	76
	66.249.65.174	6	45
	65.214.44.45	5	145
	128.194.135.94	5	63
	192.168.3.25	4	35
	65.55.208.94	4	23
	65.55.208.95	3	25
	65.55.208.96	3	20
	62.178.10.113	3	120
	65.54.165.36	3	35
	38.113.234.181	2	15
	192.168.1.53	2	8
	64.71.164.125	2	77
	192.168.1.57	2	8
	192.168.2.130	2	9

A.2.4 Average duration per visitors

Total duration per visitor/no of visits per visitor (sample of twenty visitor)

<i>Average visitor visits Duration (minutes per visit)</i>	User IP	Number of Visits	Average visit duration (minute / visit)
	65.54.188.61	12	6
	65.54.188.62	11	8
	192.168.2.30	10	10
	65.54.188.60	8	6
	143.248.110.60	6	5
	65.55.212.65	6	12
	66.249.65.174	6	7
	65.214.44.45	5	29
	128.194.135.94	5	12
	192.168.3.25	4	8
	65.55.208.94	4	5
	65.55.208.95	3	8
	65.55.208.96	3	6
	62.178.10.113	3	40

	65.54.165.36	3	11
	38.113.234.181	2	7
	192.168.1.53	2	4
	64.71.164.125	2	38
	192.168.1.57	2	4
	192.168.2.130	2	4

A.2.5 Average visits duration for all visitors

Total duration for all visits/Total Number of visits

<i>Average visits durations(Total visit durations per minute/Total Number of Visits)</i>	Total Number of Visits	Total visit durations per minute	Average minutes per visit
	159	2004	12.6

A.2.6 Top Visitors by Duration (top twenty)

<i>Top Visitors by Duration</i>	IP	Number of Visits	Duration for all visits
	65.214.44.45	5	145
	62.178.10.113	3	120
	192.168.2.30	10	100
	65.54.188.62	11	95
	62.119.73.3	1	79
	64.71.164.125	2	77
	65.55.212.65	6	76
	65.54.188.61	12	75
	128.194.135.94	5	63
	192.168.2.83	1	61
	65.54.188.60	8	51
	66.249.65.174	6	45
	62.114.59.241	1	43
	66.249.90.136	2	40
	65.54.165.36	3	35
	192.168.3.25	4	35
	143.248.110.60	6	35
	65.55.208.95	3	25
	65.55.208.94	4	23
	193.48.246.16	1	20

A.2.7 Number of unique visitors

<i>Number of unique visitors</i>	No	IP	No	IP
	1	12.157.224.18	2	122.152.129.9
	3	128.175.228.117	4	128.194.135.94
	5	129.187.155.67	6	130.130.37.13
	7	130.237.66.30	8	130.238.20.217
	9	130.54.130.227	10	130.83.203.237
	11	132.170.202.87	12	132.178.125.104
	13	133.1.118.92	14	133.1.218.200
	15	134.102.61.246	16	137.226.10.198
	17	137.73.98.160	18	139.18.188.106
	19	143.248.110.60	20	143.248.226.227
	21	144.122.1.201	22	147.228.41.187
	23	150.82.52.143	24	152.105.242.235
	25	153.96.72.2	26	158.169.9.14
	27	161.252.96.193	28	172.215.174.231
	29	192.168.1.15	30	192.168.1.53
	31	192.168.1.57	32	192.168.2.130
	33	192.168.2.133	34	192.168.2.139
	35	192.168.2.144	36	192.168.2.147
	37	192.168.2.18	38	192.168.2.19
	39	192.168.2.224	40	192.168.2.243
	41	192.168.2.30	42	192.168.2.32
	43	192.168.2.44	44	192.168.2.60
	45	192.168.2.77	46	192.168.2.81
	47	192.168.2.83	48	192.168.2.84
	49	192.168.2.85	50	192.168.3.19
	51	192.168.3.25	52	192.168.4.156
	53	192.168.4.20	54	192.168.4.22
	55	192.168.4.26	56	192.168.4.35
	57	192.168.4.42	58	192.168.4.81
	59	192.168.4.82	60	192.168.5.16
	61	192.168.5.18	62	192.168.5.30
	63	193.140.142.10	64	193.145.249.213
	65	193.194.69.210	66	193.194.83.169
	67	193.194.92.197	68	193.194.92.202
	69	193.227.29.230	70	193.227.30.5
	71	193.47.80.42	72	193.48.246.16
	73	195.229.236.214	74	195.229.236.217
	75	195.229.236.218	76	195.24.134.69
	77	195.43.0.250	78	195.43.3.70
	79	195.97.22.113	80	195.97.225.3
	81	196.2.124.252	82	196.202.101.168
	83	196.202.111.179	84	196.202.16.88
	85	196.202.24.213	86	196.202.35.195
	87	196.202.5.198	88	196.202.56.125
	89	196.202.57.152	90	196.202.62.3
	91	196.202.65.52	92	196.202.75.215

93	196.202.8.203	94	196.202.97.113
95	196.204.6.140	96	196.205.128.224
97	196.205.219.254	98	196.205.230.45
99	196.205.233.122	100	196.205.241.96
101	196.205.35.9	102	196.206.120.170
103	196.218.112.91	104	196.218.114.79
105	196.218.12.216	106	196.218.154.163
107	196.218.156.49	108	196.218.19.110
109	196.218.19.188	110	196.218.19.242
111	196.218.19.72	112	196.218.190.39
113	196.218.21.204	114	196.218.23.97
115	196.218.29.70	116	196.218.36.142
117	196.218.46.111	118	196.218.56.16
119	196.219.111.169	120	196.219.129.231
121	196.219.140.212	122	196.219.141.141
123	196.219.153.161	124	196.219.153.168
125	196.219.153.85	126	196.219.156.150
127	196.219.163.111	128	196.219.164.75
129	196.219.184.95	130	196.219.194.132
131	196.219.194.192	132	196.219.196.200
133	200.10.161.160	134	203.144.143.9
135	203.199.213.131	136	203.200.35.35
137	211.229.145.44	138	211.25.50.12
139	212.117.73.42	140	212.12.244.228
141	212.138.47.13	142	212.24.224.17
143	212.24.224.18	144	213.131.70.13
145	213.154.91.36	146	213.158.177.106
147	213.181.224.27	148	213.186.167.139
149	213.212.233.122	150	213.42.21.75
151	213.6.85.145	152	217.139.56.2
153	217.52.88.25	154	217.53.105.101
155	217.53.80.188	156	217.53.83.119
157	217.54.192.179	158	218.219.224.206
159	218.82.144.120	160	219.136.75.106
161	220.227.207.35	162	222.14.78.218
163	38.113.234.181	164	41.196.176.36
165	41.222.70.194	166	41.250.51.84
167	58.22.131.13	168	59.92.51.39
169	62.114.101.248	170	62.114.159.196
171	62.114.34.151	172	62.114.57.174
173	62.114.59.235	174	62.114.59.241
175	62.114.59.245	176	62.114.59.37
177	62.117.33.11	178	62.119.73.3
179	62.139.80.40	180	62.139.86.20
181	62.140.74.77	182	62.149.114.19
183	62.150.176.65	184	62.178.10.113
185	62.241.139.35	186	62.241.145.213
187	62.241.151.161	188	62.68.255.200
189	62.68.57.121	190	64.71.164.125

191	65.214.44.45	192	65.54.165.35
193	65.54.165.36	194	65.54.188.60
195	65.54.188.61	196	65.54.188.62
197	65.54.188.63	198	65.55.208.109
199	65.55.208.111	200	65.55.208.112
201	65.55.208.113	202	65.55.208.114
203	65.55.208.90	204	65.55.208.91
205	65.55.208.92	206	65.55.208.93
207	65.55.208.94	208	65.55.208.95
209	65.55.208.96	210	65.55.208.97
211	65.55.212.65	212	65.55.235.140
213	66.232.124.38	214	66.249.65.115
215	66.249.65.174	216	66.249.65.193
217	66.249.72.7	218	66.249.90.136
219	67.169.58.234	220	68.151.114.132
221	68.50.118.183	222	71.127.36.180
223	72.36.146.50	224	74.124.192.201
225	74.139.203.227	226	74.14.252.159
227	80.11.150.47	228	80.169.156.244
229	81.10.87.249	230	81.169.235.173
231	81.183.142.130	232	81.21.97.8
233	81.31.160.26	234	82.103.138.223
235	82.146.166.137	236	82.194.62.227
237	82.198.177.182	238	82.201.170.62
239	82.201.179.127	240	82.201.221.95
241	82.201.222.7	242	82.201.243.109
243	82.201.244.195	244	82.201.255.108
245	82.89.230.197	246	83.101.150.116
247	84.0.219.228	248	84.255.187.178
249	84.36.12.151	250	84.36.147.227
251	84.36.150.157	252	84.36.158.237
253	84.36.17.122	254	84.36.2.215
255	84.36.20.228	256	84.36.28.232
257	84.54.27.5	258	85.103.2.173
259	85.249.139.82	260	87.101.244.9
261	87.118.112.30	262	88.116.163.106

A.3 Site Navigation

A.3.1 Visitors popular paths through the web site

<i>12.157.224.18</i>	No of Visits	Path _ Time per Seconds _
	1	AMSAT ==6=> contact ==5=> CulturalProgram
<i>122.152.129.9</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> index
<i>128.175.228.117</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==12=> index ==48=> Nanostructured_Materials_Laboratory ==7=> Advanced_Materials_Department ==6=> Electronic_Materials_Laboratory ==1=> intro ==29=> Electronic_Materials_Facilities ==2=> Electronic_Materials_Publications ==19=> Electronic_Materials_Facilities ==4=> Electronic_Materials_Laboratory ==7=> Advanced_Materials_Department ==13=> Ceramic_Materials_-_Staff ==7=> intro ==10=> Composite ==7=> Composite-Staff ==17=> Composite ==4=> Advanced_Materials_Department ==14=> Metals_Technology_Department ==5=> intro ==9=> Corrosion_Staff ==1=> intro ==6=> Manufacturing_Technology_Department ==8=> PowderTechnologyLaboratory ==2=> Staff
<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==42=> AMSAT
<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	2	Conferences ==37=> AMSAT06 ==45=> mission ==48=> NewsLetter ==87=> IT ==47=> Contact_us ==43=> chairman ==43=> mpm07 ==300=> Training_Department ==42=> dep ==46=> CulturalProgram ==46=> Exhibition ==84=> Registration ==45=> Instructions
<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	3	italy ==219=> library ==300=> contact ==300=> NewsLetter-3-2006-pic ==218=> Devices ==124=> Training_Courses ==44=> Manufacturing_Technology_Department
<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	4	Advanced_Materials_Department ==126=> Information ==42=> CulturalProgramTour ==41=> schedule ==44=> Postconferencetours ==45=> SubAndinst ==44=> RegForm

<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	5	italy ==244=> library
<i>128.194.135.94</i>	No of Visits	Path _ Time per Seconds _
	6	Devices ==172=> Manufacturing_Technology_Department ==44=> Minerals_Processing_and_Technology_Department ==300=> contact ==300=> Advanced_Materials_Department ==48=> Metals_Technology_Department ==44=> RegForm ==43=> AccompanyingPersonsProgram ==42=> Information ==42=> CulturalProgramTour ==43=> schedule ==44=> Postconferencetours ==42=> SubAndinst
<i>129.187.155.67</i>	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==7=> important_dates ==4=> culture_program ==5=> organizing
<i>130.130.37.13</i>	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==21=> important_dates

<i>130.237.66.30</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> Minerals_Processing_and_Technology_Department ==5=> Chemical_and_Electrometallurgy_Laboratory ==95=> chairman ==26=> Contact_us
<i>130.238.20.217</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==146=> dep ==3=> Advanced_Materials_Department ==4=> Nanostructured_Materials_Laboratory ==3=> Nanostructured_Materials_Staff ==28=> Nanostructured_Materials_Publications
<i>130.54.130.227</i>	No of Visits	Path _ Time per Seconds _
	1	chairman ==4=> intro ==21=> AMSAT ==69=> index ==26=> topics ==37=> important_dates
<i>130.83.203.237</i>	No of Visits	Path _ Time per Seconds _
	1	index ==8=> dep ==19=> Ceramic_Materials_Laboratory ==5=> Ceramic_Materials_-_Staff
<i>132.170.202.87</i>	No of Visits	Path _ Time per Seconds _
	1	chairman ==300=> index
<i>132.178.125.104</i>	No of	Path _ Time per Seconds _

	Visits	
	1	intro ==4=> index ==300=> Corrosion_Laboratory ==4=> Corrosion_Staff ==15=> chairman ==42=> AMSAT ==17=> mpm07 ==111=> organizing ==151=> Accompanying ==25=> submission ==45=> welcome
<i>133.1.118.92</i>	No of Visits	Path _ Time per Seconds _
	1	chairman ==55=> index
<i>133.1.118.92</i>	No of Visits	Path _ Time per Seconds _
	2	chairman ==2=> intro ==9=> index
<i>133.1.218.200</i>	No of Visits	Path _ Time per Seconds _
	1	index ==2=> intro
<i>134.102.61.246</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> index
<i>137.226.10.198</i>	No of Visits	Path _ Time per Seconds _
	1	submission ==10=> culture_program ==13=> Accompanying ==46=> important_dates
<i>137.73.98.160</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index
<i>137.73.98.160</i>	No of Visits	Path _ Time per Seconds _
	2	chairman ==300=> Advanced_Materials_Department ==11=> Steel_Technology_Laboratory
<i>139.18.188.106</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==67=> library
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _
	1	index ==4=> intro
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _
	2	intro ==300=> chairman ==3=> intro ==4=> index
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _
	3	intro ==300=> index ==2=> intro
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _
	4	index ==300=> intro ==5=> index ==19=> intro ==300=> index ==3=> intro
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _
	5	index ==2=> intro
<i>143.248.110.60</i>	No of Visits	Path _ Time per Seconds _

	6	intro ==300=> index ==2=> intro
143.248.110.60	No of Visits	Path _ Time per Seconds _
	7	index ==3=> intro
143.248.110.60	No of Visits	Path _ Time per Seconds _
	8	index ==5=> intro
143.248.110.60	No of Visits	Path _ Time per Seconds _
	9	index ==2=> intro ==300=> index ==3=> intro
143.248.110.60	No of Visits	Path _ Time per Seconds _
	10	intro ==300=> index
143.248.226.227	No of Visits	Path _ Time per Seconds _
	1	chairman ==300=> index
143.248.226.227	No of Visits	Path _ Time per Seconds _
	2	index ==2=> intro
143.248.226.227	No of Visits	Path _ Time per Seconds _
	3	index ==3=> intro
144.122.1.201	No of Visits	Path _ Time per Seconds _
	1	index ==63=> Contact_us ==32=> Metals_Technology_Department ==4=> Manufacturing_Technology_Department ==0=> Advanced_Materials_Department ==102=> Minerals_Characterization_Laboratory ==89=> Beneficiation_Laboratory ==268=> Corrosion_Laboratory ==2=> Plastic_Deformation_Laboratory ==0=> NonFerrous_Laboratory ==80=> NonFerrous_Staff ==37=> Plastic-Deformation-Staff ==17=> Steel_Technology_Staff ==60=> Ceramic_Materials_Laboratory ==1=> Electronic_Materials_Laboratory ==0=> Nanostructured_Materials_Laboratory ==125=> welding ==1=> rpm ==4=> Staff ==36=> casting-Staff
147.228.41.187	No of Visits	Path _ Time per Seconds _
	1	intro ==126=> dep ==41=> Metals_Technology_Department ==7=> Corrosion_Laboratory ==28=> Corrosion_Equipments
150.82.52.143	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==20=> important_dates
152.105.242.235	No of	Path _ Time per Seconds _

	Visits	
	1	intro ==24=> index
153.96.72.2	No of Visits	Path _ Time per Seconds _
	1	index ==8=> Manufacturing_Technology_Department ==22=> PowderTechnologyLaboratory ==7=> Staff ==30=> casting-Staff ==29=> casting-Contact_Us ==14=> rpm ==4=> rpm-Facilities ==30=> kghany ==18=> rpmContact_Us ==300=> mission
153.96.72.2	No of Visits	Path _ Time per Seconds _
	2	Conferences ==42=> Training_Department
153.96.72.2	No of Visits	Path _ Time per Seconds _
	3	intro ==5=> index
158.169.9.14	No of Visits	Path _ Time per Seconds _
	1	Composite ==23=> mission
161.252.96.193	No of Visits	Path _ Time per Seconds _
	1	culture_program ==39=> topics
172.215.174.231	No of Visits	Path _ Time per Seconds _
	1	IT ==300=> mission ==12=> chairman ==174=> GUC ==50=> AMSAT ==192=> Technical_Services
172.215.174.231	No of Visits	Path _ Time per Seconds _
	2	NewsLetter-3-2006-pic ==300=> italy
192.168.1.15	No of Visits	Path _ Time per Seconds _
	1	index ==300=> important_dates ==81=> organizing
192.168.1.53	No of Visits	Path _ Time per Seconds _
	1	important_dates ==2=> topics ==2=> submission ==3=> Accompanying
192.168.1.53	No of Visits	Path _ Time per Seconds _
	2	index ==164=> intro
192.168.1.53	No of Visits	Path _ Time per Seconds _
	3	intro ==5=> index ==11=> Manufacturing_Technology_Department ==6=> casting-Staff ==13=> casting-Publications ==104=> culture_program ==3=> topics ==21=> organizing ==34=> topics ==116=> GUC ==62=> italy
192.168.1.53	No of Visits	Path _ Time per Seconds _
	4	intro ==5=> index
192.168.1.57	No of	Path _ Time per Seconds _

	Visits	
	1	chairman ==41=> dep ==5=> Manufacturing_Technology_Department ==11=> casting-Staff ==34=> GUC ==77=> IT
192.168.1.57	No of Visits	Path _ Time per Seconds _
	2	intro ==5=> index
192.168.1.57	No of Visits	Path _ Time per Seconds _
	3	intro ==7=> index
192.168.1.57	No of Visits	Path _ Time per Seconds _
	4	intro ==300=> NewsLetter ==86=> intro
192.168.2.130	No of Visits	Path _ Time per Seconds _
	1	intro ==125=> Advanced_Materials_Department ==14=> Ceramic_Materials_Laboratory ==12=> Ceramic_Materials_-_Staff ==61=> Ceramic_Materials_-_Projects
192.168.2.130	No of Visits	Path _ Time per Seconds _
	2	intro ==43=> index ==41=> dep ==20=> Metals_Technology_Department ==6=> NonFerrous_Laboratory ==47=> intro ==5=> index ==10=> Minerals_Processing_and_Technology_Department ==23=> Manufacturing_Technology_Department ==18=> casting-Staff ==38=> PowderTechnologyLaboratory ==26=> welding ==15=> rpm ==6=> rpm-Staff ==38=> Beneficiation_Laboratory ==44=> Minerals_Characterization_Laboratory
192.168.2.133	No of Visits	Path _ Time per Seconds _
	1	index ==67=> Advanced_Materials_Department ==7=> Composite
192.168.2.139	No of Visits	Path _ Time per Seconds _
	1	index ==34=> culture_program ==4=> submission ==69=> Conferences
192.168.2.139	No of Visits	Path _ Time per Seconds _
	2	index ==60=> dep ==6=> chairman ==88=> casting-Staff
192.168.2.139	No of Visits	Path _ Time per Seconds _
	3	intro ==3=> index
192.168.2.144	No of Visits	Path _ Time per Seconds _
	1	intro ==25=>

		Minerals_Processing_and_Technology_Department
192.168.2.144	No of Visits	Path _ Time per Seconds _
	2	intro ==17=> Minerals_Processing_and_Technology_Department ==19=> Minerals_Characterization_Laboratory
192.168.2.147	No of Visits	Path _ Time per Seconds _
	1	index ==9=> Advanced_Materials_Department ==5=> Electronic_Materials_Laboratory ==300=> Nanostructured_Materials_Laboratory
192.168.2.18	No of Visits	Path _ Time per Seconds _
	1	index ==145=> Conferences ==42=> GUC ==51=> Manufacturing_Technology_Department ==3=> casting ==63=> casting-Publications ==38=> casting-Contact_Us
192.168.2.18	No of Visits	Path _ Time per Seconds _
	2	index ==48=> Advanced_Materials_Department ==21=> Minerals_Processing_and_Technology_Department ==10=> Manufacturing_Technology_Department ==300=> library ==169=> Devices
192.168.2.19	No of Visits	Path _ Time per Seconds _
	1	intro ==92=> Devices ==86=> dep ==34=> library ==32=> NewsLetter
192.168.2.224	No of Visits	Path _ Time per Seconds _
	1	intro ==14=> Technical_Services
192.168.2.243	No of Visits	Path _ Time per Seconds _
	1	intro ==26=> dep ==4=> Metals_Technology_Department ==6=> Plastic_Deformation_Laboratory ==300=> GUC
192.168.2.30	No of Visits	Path _ Time per Seconds _
	1	intro ==5=> index ==300=> intro ==4=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	2	index ==300=> intro ==4=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	3	intro ==300=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	4	intro ==300=> index ==300=> intro ==10=> index ==300=> intro

192.168.2.30	No of Visits	Path _ Time per Seconds _
	5	intro ==5=> index ==300=> intro ==300=> index ==300=> intro ==7=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	6	intro ==5=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	7	intro ==7=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	8	intro ==9=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	9	intro ==5=> index ==300=> intro ==4=> index ==300=> intro ==5=> index ==300=> intro ==4=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	10	intro ==8=> index ==300=> intro ==5=> index ==300=> intro ==5=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	11	intro ==300=> index ==300=> intro ==6=> index
192.168.2.30	No of Visits	Path _ Time per Seconds _
	12	intro ==300=> index ==300=> intro ==6=> index
192.168.2.32	No of Visits	Path _ Time per Seconds _
	1	intro ==40=> dep ==4=> Advanced_Materials_Department ==2=> Composite ==11=> Composite-Facilities ==9=> Composite-Publications ==15=> Composite-Staff ==18=> Ceramic_Materials_Laboratory ==5=> Ceramic_Materials_-_Staff ==43=> Ceramic_Materials_-_Projects ==29=> Minerals_Processing_and_Technology_Department ==3=> Beneficiation_Laboratory ==29=> Beneficiation_Publications
192.168.2.44	No of Visits	Path _ Time per Seconds _
	1	intro ==12=> index
192.168.2.60	No of Visits	Path _ Time per Seconds _
	1	index ==14=> Minerals_Processing_and_Technology_Department ==3=> Beneficiation_Laboratory ==4=>

		Beneficiation Staff
192.168.2.60	No of Visits	Path _ Time per Seconds _
	2	Training Department ==70=> عطاءات ومناقصات
192.168.2.60	No of Visits	Path _ Time per Seconds _
	3	intro ==4=> index
192.168.2.60	No of Visits	Path _ Time per Seconds _
	4	intro ==10=> index
192.168.2.60	No of Visits	Path _ Time per Seconds _
	5	intro ==300=> index
192.168.2.77	No of Visits	Path _ Time per Seconds _
	1	index ==18=> Metals_Technology_Department ==4=> Corrosion_Laboratory ==11=> Corrosion_Equipments
192.168.2.81	No of Visits	Path _ Time per Seconds _
	1	index ==300=> عطاءات ومناقصات ==47=> Metals_Technology_Department ==10=> Steel_Technology_Laboratory ==8=> Steel_Technology_Staff ==300=> Steel_Technology_Projects
192.168.2.83	No of Visits	Path _ Time per Seconds _
	1	chairman ==155=> dep ==300=> Minerals_Processing_and_Technology_Department ==300=> Minerals_Characterization_Laboratory ==31=> Minerals_Characterization_Activities ==135=> Minerals_Characterization_Staff ==300=> Corrosion_Laboratory ==5=> Corrosion_Activities ==300=> Plastic_Deformation_Laboratory ==300=> Steel_Technology_Activities ==300=> Advanced_Materials_Department ==300=> Ceramic_Materials_-_Projects ==7=> Ceramic_Materials_-_Staff ==105=> Electronic_Materials_Laboratory ==13=> Electronic_Materials_Activities ==85=> Electronic_Materials_Staff ==109=> Nanostructured_Materials_Activities ==41=> Nanostructured_Materials_Staff ==300=> PowderTechnologyLaboratory ==300=> casting-Activities ==84=> casting-Staff ==101=> rpm ==7=> rpm-Activities ==115=> rpm-Staff
192.168.2.84	No of Visits	Path _ Time per Seconds _
	1	intro ==88=> dep ==3=> Minerals_Processing_and_Technology_Department ==3=> Minerals_Characterization_Laboratory

192.168.2.84	No of Visits	Path _ Time per Seconds _
	2	intro ==15=> Minerals_Processing_and_Technology_Department ==10=> Minerals_Characterization_Laboratory ==18=> Minerals_Characterization_Activities ==7=> Minerals_Characterization_Staff ==16=> Minerals_Characterization_Technical_Services ==39=> Advanced_Materials_Department ==23=> Nanostructured_Materials_Laboratory ==5=> Nanostructured_Materials_Staff ==63=> Manufacturing_Technology_Department ==89=> Corrosion_Laboratory ==2=> Corrosion_Activities ==7=> Corrosion_Staff
192.168.2.85	No of Visits	Path _ Time per Seconds _
	1	Metals_Technology_Department ==8=> Corrosion_Laboratory ==9=> Corrosion_Staff
192.168.2.85	No of Visits	Path _ Time per Seconds _
	2	index ==98=> IT ==24=> chairman
192.168.3.19	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> index
192.168.3.25	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> index
192.168.3.25	No of Visits	Path _ Time per Seconds _
	2	index ==300=> intro ==5=> index
192.168.3.25	No of Visits	Path _ Time per Seconds _
	3	intro ==4=> index ==300=> intro ==3=> index ==300=> intro ==300=> index
192.168.3.25	No of Visits	Path _ Time per Seconds _
	4	intro ==4=> index ==300=> intro ==2=> index
192.168.3.25	No of Visits	Path _ Time per Seconds _
	5	intro ==159=> dep ==19=> Minerals_Processing_and_Technology_Department ==31=> Metals_Technology_Department ==108=> Manufacturing_Technology_Department ==300=> intro
192.168.3.25	No of Visits	Path _ Time per Seconds _
	6	intro ==3=> index
192.168.4.156	No of Visits	Path _ Time per Seconds _
	1	dep ==13=> Manufacturing_Technology_Department

		==17=> Plastic_Deformation_Laboratory ==26=> Steel_Technology_Laboratory ==191=> Composite ==137=> Electronic_Materials_Staff ==133=> Minerals_Processing_and_Technology_Department ==20=> Chemical_and_Electrometallurgy_Laboratory ==5=> Pyrometallurgy ==47=> Pyrometallurgy-Staff ==126=> Composite-Activities ==162=> Ceramic_Materials_- Projects
192.168.4.20	No of Visits	Path _ Time per Seconds _
	1	index ==9=> intro
192.168.4.20	No of Visits	Path _ Time per Seconds _
	2	index ==0=> intro
192.168.4.20	No of Visits	Path _ Time per Seconds _
	3	index ==115=> Metals_Technology_Department ==6=> Corrosion_Laboratory ==11=> Corrosion_Staff ==11=> Corrosion_Activities ==4=> Corrosion_Equipments
192.168.4.22	No of Visits	Path _ Time per Seconds _
	1	intro ==196=> Minerals_Processing_and_Technology_Department ==24=> Beneficiation_Laboratory
192.168.4.26	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index
192.168.4.35	No of Visits	Path _ Time per Seconds _
	1	intro ==177=> dep ==7=> Manufacturing_Technology_Department ==263=> Metals_Technology_Department ==11=> Corrosion_Laboratory ==2=> Corrosion_Activities ==5=> Corrosion_Equipments ==31=> Chemical_and_Electrometallurgy_Laboratory ==19=> Advanced_Materials_Department ==6=> Composite ==12=> Composite-Facilities ==10=> Composite-Projects ==8=> Composite-Publications
192.168.4.42	No of Visits	Path _ Time per Seconds _
	1	intro ==9=> mpm07 ==5=> topics ==7=> important_dates ==5=> organizing
192.168.4.81	No of Visits	Path _ Time per Seconds _
	1	index ==0=> intro ==0=> chairman ==64=> intro
192.168.4.82	No of Visits	Path _ Time per Seconds _
	1	intro ==9=> index

<i>192.168.5.16</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==9=> index
<i>192.168.5.16</i>	No of Visits	Path _ Time per Seconds _
	2	intro ==2=> index
<i>192.168.5.16</i>	No of Visits	Path _ Time per Seconds _
	3	intro ==5=> index
<i>192.168.5.16</i>	No of Visits	Path _ Time per Seconds _
	4	index ==300=> intro
<i>192.168.5.18</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==5=> index
<i>192.168.5.30</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> index
<i>193.140.142.10</i>	No of Visits	Path _ Time per Seconds _
	1	index ==46=> chairman
<i>193.145.249.213</i>	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==10=> culture_program ==2=> submission ==16=> organizing ==25=> topics
<i>193.194.69.210</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==129=> dep ==87=> Manufacturing_Technology_Department ==300=> welding
<i>193.194.83.169</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==14=> index
<i>193.194.83.169</i>	No of Visits	Path _ Time per Seconds _
	2	intro ==35=> mission ==51=> dep ==31=> Corrosion_Laboratory ==21=> Corrosion_Staff ==70=> Corrosion_Publications ==114=> Corrosion_Projects ==90=> Advanced_Materials_Department ==12=> Nanostructured_Materials_Laboratory
<i>193.194.92.197</i>	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==58=> submission ==83=> important_dates ==98=> topics
<i>193.194.92.202</i>	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==1=> intro
<i>193.227.29.230</i>	No of Visits	Path _ Time per Seconds _

	1	mpm07 ==108=> topics
193.227.30.5	No of Visits	Path _ Time per Seconds _
	1	index ==174=> Training_Courses_-_2004-2005 ==177=> Manufacturing_Technology_Department ==9=> welding
193.47.80.42	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==300=> Beneficiation_Publications
193.48.246.16	No of Visits	Path _ Time per Seconds _
	1	intro ==20=> index ==22=> mpm07 ==69=> AMSAT ==300=> dep ==23=> Advanced_Materials_Department ==19=> Electronic_Materials_Laboratory ==20=> Electronic_Materials_Projects ==213=> Electronic_Materials_Activities ==44=> Electronic_Materials_Staff ==173=> Nanostructured_Materials_Staff ==115=> Composite ==108=> Composite-Staff ==31=> Composite-Publications ==58=> Ceramic_Materials_Laboratory
195.229.236.214	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==99=> intro
195.229.236.217	No of Visits	Path _ Time per Seconds _
	1	library ==6=> intro
195.229.236.218	No of Visits	Path _ Time per Seconds _
	1	library ==4=> intro
195.24.134.69	No of Visits	Path _ Time per Seconds _
	1	intro ==12=> index ==8=> dep
195.43.0.250	No of Visits	Path _ Time per Seconds _
	1	chairman ==0=> intro
195.43.3.70	No of Visits	Path _ Time per Seconds _
	1	index ==80=> GUC
195.43.3.70	No of Visits	Path _ Time per Seconds _
	2	intro ==3=> index
195.97.22.113	No of Visits	Path _ Time per Seconds _
	1	Minerals_Characterization_Activities ==200=> Minerals_Characterization_Facilities ==300=> Minerals_Characterization_Contact_Us ==238=> italy ==42=> AMSAT ==39=> mpm07 ==65=>

		NewsLetter
195.97.225.3	No of Visits	Path _ Time per Seconds _
	1	welding-Facilities ==178=> welding-TechnicalServices
196.2.124.252	No of Visits	Path _ Time per Seconds _
	1	index ==0=> chairman ==1=> AMSAT ==6=> dep ==96=> topics ==12=> Registration ==3=> Exhibition ==18=> Instructions ==49=> mission ==5=> contact ==2=> important_dates ==42=> Contact_us ==20=> culture_program
196.202.101.168	No of Visits	Path _ Time per Seconds _
	1	intro ==14=> index
196.202.111.179	No of Visits	Path _ Time per Seconds _
	1	chairman ==0=> intro
196.202.16.88	No of Visits	Path _ Time per Seconds _
	1	Training_Department ==24=> Training_Courses ==11=> Training_Courses - 2004-2005
196.202.24.213	No of Visits	Path _ Time per Seconds _
	1	index ==4=> intro ==56=> عطاءات ومناقصات
196.202.35.195	No of Visits	Path _ Time per Seconds _
	1	intro ==47=> Training_Department ==21=> mpm07 ==35=> submission ==8=> culture_program ==3=> Accompanying ==8=> organizing
196.202.5.198	No of Visits	Path _ Time per Seconds _
	1	chairman ==1=> intro
196.202.56.125	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==1=> intro
196.202.57.152	No of Visits	Path _ Time per Seconds _
	1	Devices ==57=> Mechanical_Tests
196.202.62.3	No of Visits	Path _ Time per Seconds _
	1	chairman ==279=> dep ==2=> intro ==89=> mission ==0=> intro
196.202.65.52	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> Training_Department ==40=> Training_Courses - 2004-2005
196.202.75.215	No of Visits	Path _ Time per Seconds _
	1	intro ==47=> index

196.202.8.203	No of Visits	Path _ Time per Seconds _
	1	Conferences ==99=> Contact_us ==0=> intro
196.202.97.113	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index
196.204.6.140	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index
196.205.128.224	No of Visits	Path _ Time per Seconds _
	1	intro ==185=> Manufacturing_Technology_Department ==232=> AMSAT
196.205.219.254	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==264=> topics
196.205.230.45	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==159=> topics ==142=> culture_program ==29=> Accompanying
196.205.233.122	No of Visits	Path _ Time per Seconds _
	1	intro ==57=> index
196.205.241.96	No of Visits	Path _ Time per Seconds _
	1	عطاءات ومناقصات ==3=> intro
196.205.35.9	No of Visits	Path _ Time per Seconds _
	1	Composite ==52=> Composite-Staff ==48=> Ceramic_Materials - Staff
196.206.120.170	No of Visits	Path _ Time per Seconds _
	1	عطاءات ومناقصات ==1=> intro
196.218.112.91	No of Visits	Path _ Time per Seconds _
	1	intro ==7=> index
196.218.114.79	No of Visits	Path _ Time per Seconds _
	1	library ==152=> NewsLetter
196.218.12.216	No of Visits	Path _ Time per Seconds _
	1	chairman ==7=> intro ==116=> index
196.218.154.163	No of Visits	Path _ Time per Seconds _
	1	dep ==71=> Training_Courses ==8=> Training_Courses - 2004-2005 ==57=> Registration ==67=> Metals_Technology_Department ==10=> Minerals_Processing_and_Technology_Department ==7=> Minerals_Characterization_Laboratory

		==13=> Minerals_Characterization_Facilities
196.218.156.49	No of Visits	Path _ Time per Seconds _
	1	Contact_us ==39=> intro ==3=> عطاءات ومناقصات
196.218.19.110	No of Visits	Path _ Time per Seconds _
	1	intro ==36=> index
196.218.19.188	No of Visits	Path _ Time per Seconds _
	1	intro ==9=> index ==300=> intro
196.218.19.242	No of Visits	Path _ Time per Seconds _
	1	intro ==59=> index
196.218.19.72	No of Visits	Path _ Time per Seconds _
	1	intro ==13=> index
196.218.19.72	No of Visits	Path _ Time per Seconds _
	2	intro ==7=> index ==300=> intro
196.218.190.39	No of Visits	Path _ Time per Seconds _
	1	intro ==30=> index
196.218.21.204	No of Visits	Path _ Time per Seconds _
	1	index ==1=> intro ==7=> dep ==6=> Manufacturing_Technology_Department
196.218.23.97	No of Visits	Path _ Time per Seconds _
	1	intro ==14=> index
196.218.29.70	No of Visits	Path _ Time per Seconds _
	1	intro ==159=> index
196.218.36.142	No of Visits	Path _ Time per Seconds _
	1	Minerals_Characterization_Publications ==2=> intro
196.218.46.111	No of Visits	Path _ Time per Seconds _
	1	index ==300=> Contact_us
196.218.56.16	No of Visits	Path _ Time per Seconds _
	1	عطاءات ومناقصات ==45=> intro ==0=> mission
196.219.111.169	No of Visits	Path _ Time per Seconds _
	1	Contact_us ==22=> dep ==11=> Manufacturing_Technology_Department ==10=> welding
196.219.129.231	No of Visits	Path _ Time per Seconds _

	1	intro ==0=> Conferences ==70=> intro ==23=> IT ==8=> intro ==10=> AMSAT ==19=> intro ==15=> mpm07
<i>196.219.140.212</i>	No of Visits	Path _ Time per Seconds _
	1	index ==6=> intro ==45=> Conferences ==300=> Training_Department ==57=> intro
<i>196.219.141.141</i>	No of Visits	Path _ Time per Seconds _
	1	italy ==300=> Training_Department ==44=> Training_Courses _ 2004-2005 ==28=> Registration
<i>196.219.153.161</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==13=> index
<i>196.219.153.168</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==10=> index
<i>196.219.153.85</i>	No of Visits	Path _ Time per Seconds _
	1	intro ==15=> index
<i>196.219.156.150</i>	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==300=> submission ==300=> organizing
<i>196.219.163.111</i>	No of Visits	Path _ Time per Seconds _
	1	chairman ==5=> intro
<i>196.219.164.75</i>	No of Visits	Path _ Time per Seconds _
	1	chairman ==36=> intro ==10=> Advanced_Materials_Department ==6=> intro ==0=> Composite ==121=> Composite-Projects ==38=> Composite-Staff ==65=> mission ==9=> intro ==6=> Metals_Technology_Department ==0=> intro ==6=> NonFerrous_Laboratory ==0=> intro ==33=> NonFerrous_Publications ==67=> NonFerrous_Facilities ==29=> intro ==7=> Manufacturing_Technology_Department ==0=> intro ==47=> TechnicalServices ==58=> intro ==0=> casting ==9=> casting-Facilities ==54=> casting-Staff ==16=> casting-Publications ==39=> casting-Projects ==14=> intro ==0=> rpm ==7=> rpm-Staff ==77=> intro
<i>196.219.184.95</i>	No of Visits	Path _ Time per Seconds _
	1	Conferences ==5=> intro
<i>196.219.194.132</i>	No of Visits	Path _ Time per Seconds _
	1	Activities ==55=> GUC
<i>196.219.194.192</i>	No of Visits	Path _ Time per Seconds _

	1	Composite-Projects ==21=> mission ==6=> chairman ==29=> library
196.219.196.200	No of Visits	Path _ Time per Seconds _
	1	index ==4=> intro ==22=> عطاءات ومناقصات
200.10.161.160	No of Visits	Path _ Time per Seconds _
	1	index ==86=> dep ==189=> Advanced_Materials_Department ==17=> Electronic_Materials_Laboratory ==33=> Nanostructured_Materials_Laboratory
200.10.161.160	No of Visits	Path _ Time per Seconds _
	2	dep ==16=> Manufacturing_Technology_Department ==221=> Advanced_Materials_Department ==15=> Nanostructured_Materials_Laboratory
203.144.143.9	No of Visits	Path _ Time per Seconds _
	1	intro ==1=> Steel_Technology_Staff
203.199.213.131	No of Visits	Path _ Time per Seconds _
	1	Technical_Services ==4=> Devices
203.200.35.35	No of Visits	Path _ Time per Seconds _
	1	Minerals_Characterization_Laboratory ==59=> Minerals_Characterization_Staff
211.229.145.44	No of Visits	Path _ Time per Seconds _
	1	chairman ==15=> dep ==29=> Steel_Technology_Laboratory ==7=> Steel_Technology_Staff ==203=> NonFerrous_Laboratory ==27=> NonFerrous_Projects ==41=> Steel_Technology_Projects ==101=> index
211.25.50.12	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==61=> important_dates ==12=> submission ==36=> organizing ==300=> mpm07 ==268=> important_dates ==97=> topics
212.117.73.42	No of Visits	Path _ Time per Seconds _
	1	Information ==142=> intro ==3=> chairman
212.12.244.228	No of Visits	Path _ Time per Seconds _
	1	index ==25=> Manufacturing_Technology_Department
212.12.244.228	No of Visits	Path _ Time per Seconds _
	2	intro ==25=> index ==173=> Contact_us
212.138.47.13	No of	Path _ Time per Seconds _

	Visits	
	1	chairman ==1=> intro ==31=> dep ==9=> Minerals_Processing_and_Technology_Department ==34=> Beneficiation_Laboratory
212.24.224.17	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> Training_Department ==300=> Training_Courses ==80=> Training_Courses_-2004-2005
212.24.224.18	No of Visits	Path _ Time per Seconds _
	1	intro ==69=> index ==16=> Training_Department
213.131.70.13	No of Visits	Path _ Time per Seconds _
	1	index ==10=> Contact_us ==1=> intro ==300=> dep
213.154.91.36	No of Visits	Path _ Time per Seconds _
	1	chairman ==2=> intro
213.158.177.106	No of Visits	Path _ Time per Seconds _
	1	Plastic-Deformation-Publications ==76=> Plastic-Deformation-Facilities ==116=> intro ==73=> Plastic-Deformation-Staff
213.181.224.27	No of Visits	Path _ Time per Seconds _
	1	intro ==110=> dep ==7=> Manufacturing_Technology_Department
213.181.224.27	No of Visits	Path _ Time per Seconds _
	2	intro ==9=> index ==13=> dep ==8=> Manufacturing_Technology_Department ==7=> rpm
213.186.167.139	No of Visits	Path _ Time per Seconds _
	1	welding-Projects ==24=> index
213.212.233.122	No of Visits	Path _ Time per Seconds _
	1	contact ==141=> culture_program
213.42.21.75	No of Visits	Path _ Time per Seconds _
	1	index ==165=> mission ==14=> technical_services ==10=> training_department ==151=> conferences ==0=> it ==32=> amsat ==1=> newsletter
213.6.85.145	No of Visits	Path _ Time per Seconds _
	1	Registration ==131=> Training_Courses_-_2004-2005
217.139.56.2	No of Visits	Path _ Time per Seconds _
	1	chairman ==1=> intro ==34=> mission
217.52.88.25	No of	Path _ Time per Seconds _

	Visits	
	1	topics ==93=> Accompanying ==64=> important_dates ==102=> contact
217.53.105.101	No of Visits	Path _ Time per Seconds _
	1	intro ==73=> library
217.53.80.188	No of Visits	Path _ Time per Seconds _
	1	intro ==130=> Metals_Technology_Department ==33=> mission
217.53.80.188	No of Visits	Path _ Time per Seconds _
	2	library ==156=> chairman
217.53.83.119	No of Visits	Path _ Time per Seconds _
	1	intro ==20=> index ==29=> dep ==22=> Manufacturing_Technology_Department ==5=> Metals_Technology_Department ==6=> Advanced_Materials_Department ==25=> ContactUs ==165=> Corrosion_Laboratory ==7=> Corrosion_Staff
217.54.192.179	No of Visits	Path _ Time per Seconds _
	1	Training_Department ==82=> Training_Courses_-2004-2005 ==79=> Registration
218.219.224.206	No of Visits	Path _ Time per Seconds _
	1	intro ==47=> dep ==49=> Minerals_Processing_and_Technology_Department ==8=> Metals_Technology_Department ==15=> Advanced_Materials_Department ==18=> Manufacturing_Technology_Department ==148=> rpm ==49=> Composite ==28=> Electronic_Materials_Laboratory ==26=> Nanostructured_Materials_Laboratory ==18=> Corrosion_Laboratory ==8=> Corrosion_Activities ==52=> NonFerrous_Laboratory ==32=> Steel_Technology_Activities
218.82.144.120	No of Visits	Path _ Time per Seconds _
	1	index ==218=> library
219.136.75.106	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==6=> index
220.227.207.35	No of Visits	Path _ Time per Seconds _
	1	intro ==34=> Minerals_Processing_and_Technology_Department ==8=> Chemical_and_Electrometallurgy_Laboratory ==151=> Chemical_and_Electrometallurgy_

		Activities
222.14.78.218	No of Visits	Path _ Time per Seconds _
	1	dep ==13=> IT
38.113.234.181	No of Visits	Path _ Time per Seconds _
	1	IT ==300=> NewsLetter
38.113.234.181	No of Visits	Path _ Time per Seconds _
	2	library ==300=> dep ==300=> Technical _ Services
41.196.176.36	No of Visits	Path _ Time per Seconds _
	1	عطاءات ومناقصات ==2=> intro
41.222.70.194	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> chairman ==4=> dep ==9=> Metals _ Technology _ Department ==8=> Steel _ Technology _ Laboratory ==6=> Steel _ Technology _ Staff ==39=> Corrosion _ Laboratory ==7=> Corrosion _ Staff ==21=> NonFerrous _ Laboratory ==41=> Plastic _ Deformation _ Laboratory ==68=> Plastic-Deformation-Projects ==20=> Manufacturing _ Technology _ Department ==16=> Staff ==44=> casting ==7=> casting-Staff ==239=> kghany ==114=> Advanced _ Materials _ Department ==7=> Nanostructured _ Materials _ Laboratory ==5=> Nanostructured _ Materials _ Staff ==48=> Electronic _ Materials _ Laboratory ==5=> Electronic _ Materials _ Staff
41.250.51.84	No of Visits	Path _ Time per Seconds _
	1	topics ==19=> culture_program ==7=> important _ dates
58.22.131.13	No of Visits	Path _ Time per Seconds _
	1	intro ==1=> Training _ Department
59.92.51.39	No of Visits	Path _ Time per Seconds _
	1	intro ==20=> index ==10=> dep ==7=> Advanced _ Materials _ Department ==46=> Nanostructured _ Materials _ Laboratory ==47=> Ceramic _ Materials _ Laboratory ==3=> Ceramic _ Materials _ Staff ==19=> Composite
62.114.101.248	No of Visits	Path _ Time per Seconds _
	1	Registration ==45=> Training _ Courses _ 2004-2005
62.114.159.196	No of Visits	Path _ Time per Seconds _
	1	intro ==133=>

		Minerals_Processing_and_Technology_Department ==16=> Minerals_Characterization_Laboratory ==117=> Corrosion_Laboratory ==139=> Staff ==47=> casting ==20=> casting-Staff ==20=> welding ==30=> rpm
62.114.34.151	No of Visits	Path _ Time per Seconds _
	1	GUC ==43=> intro ==84=> index ==161=> Training_Department
62.114.57.174	No of Visits	Path _ Time per Seconds _
	1	intro ==52=> index
62.114.59.235	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==300=> topics ==57=> organizing
62.114.59.241	No of Visits	Path _ Time per Seconds _
	1	intro ==23=> index ==108=> italy ==58=> index ==24=> dep ==93=> Minerals_Processing_and_Technology_Department ==27=> Minerals_Characterization_Laboratory ==140=> Minerals_Characterization_Staff ==152=> Beneficiation_Projects ==300=> Pyrometallurgy- Staff ==40=> Chemical_and_Electrometallurgy_Laboratory ==41=> Chemical_and_Electrometallurgy_-_Projects ==117=> Corrosion_Laboratory ==25=> Corrosion_Projects ==65=> NonFerrous_Laboratory ==39=> NonFerrous_Projects ==44=> NonFerrous_Staff ==76=> Plastic-Deformation- Projects ==41=> Plastic-Deformation-Staff ==188=> Steel_Technology_Projects ==300=> Steel_Technology_Staff ==26=> Advanced_Materials_Department ==20=> Composite ==49=> Composite-Staff ==34=> Ceramic_Materials_Laboratory ==23=> Ceramic_Materials_-_Projects ==37=> Ceramic_Materials_-_Staff ==134=> Nanostructured_Materials_Laboratory ==56=> Nanostructured_Materials_Staff ==40=> Manufacturing_Technology_Department ==201=> casting-Staff ==30=> welding ==46=> rpm
62.114.59.245	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==300=> topics ==57=> submission ==58=> culture_program ==24=> important_dates ==130=> contact ==300=> Accompanying
62.114.59.37	No of Visits	Path _ Time per Seconds _
	1	index ==8=> intro
62.117.33.11	No of	Path _ Time per Seconds _

	Visits	
	1	intro ==45=> index ==48=> chairman
	No of Visits	Path _ Time per Seconds _
62.119.73.3	1	mission ==0=> dep ==0=> index ==1=> Technical_Services ==0=> chairman ==0=> Training_Department ==1=> Conferences ==0=> IT ==7=> AMSAT06 ==0=> mpm07 ==0=> NewsLetter ==300=> Minerals_Processing_and_Technology_Department ==0=> Metals_Technology_Department ==0=> Advanced_Materials_Department ==1=> AMSAT ==0=> Manufacturing_Technology_Department ==300=> mission ==300=> chairman ==300=> Technical_Services ==6=> Mechanical_Tests ==0=> Devices ==300=> Training_Department ==0=> Registration ==0=> Training_Courses ==300=> IT ==300=> italy ==0=> GUC ==0=> Conferences ==300=> Contact_us ==300=> mpm07 ==0=> contact ==1=> topics ==0=> important_dates ==0=> culture_program ==211=> amsat ==0=> AMSAT06 ==1=> contact ==264=> NewsLetter-3-2006-pic ==300=> Beneficiation_Laboratory ==0=> Minerals_Processing_and_Technology_Department ==0=> Minerals_Characterization_Laboratory ==3=> Chemical_and_Electrometallurgy_Laboratory ==238=> Corrosion_Laboratory ==0=> NonFerrous_Laboratory ==0=> Metals_Technology_Department ==2=> Steel_Technology_Laboratory ==0=> Plastic_Deformation_Laboratory ==159=> Ceramic_Materials_Laboratory ==0=> Composite ==0=> Advanced_Materials_Department ==1=> Electronic_Materials_Laboratory ==0=> Nanostructured_Materials_Laboratory ==277=> Manufacturing_Technology_Department ==2=> casting ==0=> PowderTechnologyLaboratory ==4=> rpm ==300=> culture_program ==0=> contact ==0=> AMSAT ==1=> topics ==0=> important_dates ==2=> Instructions ==0=> Registration ==0=> 2circular ==300=> Mechanical_Tests
62.139.80.40	No of Visits	Path _ Time per Seconds _
	1	Composite-Facilities ==117=> chairman ==0=> intro
62.139.86.20	No of Visits	Path _ Time per Seconds _
	1	welcome ==66=> Instructions
62.140.74.77	No of Visits	Path _ Time per Seconds _
	1	intro ==31=> index
62.149.114.19	No of	Path _ Time per Seconds _

	Visits	
	1	NewsLetter ==30=> NewsLetter-3-2006-pic
62.150.176.65	No of Visits	Path _ Time per Seconds _
	1	Contact_us ==15=> intro
62.178.10.113	No of Visits	Path _ Time per Seconds _
	1	chairman ==26=> index ==300=> library ==10=> AMSAT ==300=> Technical_Services ==13=> Training_Department ==8=> IT ==237=> Conferences ==233=> Contact_us ==300=> italy ==78=> AMSAT06 ==300=> rpm-Activities ==2=> rpm-Projects ==11=> rpm-TechnicalServices ==8=> rpm-Publications ==4=> rpm-Staff ==12=> rpmContact_Us ==300=> intro
62.178.10.113	No of Visits	Path _ Time per Seconds _
	2	mission ==2=> dep ==55=> AMSAT ==300=> Technical_Services ==87=> Training_Department ==24=> IT ==38=> Conferences ==71=> Contact_us ==7=> NewsLetter ==241=> GUC ==14=> italy ==4=> AMSAT06 ==267=> rpm-Activities ==127=> rpm-Projects ==94=> rpm-TechnicalServices ==63=> rpm-Publications ==18=> rpm-Staff ==2=> rpmContact_Us ==300=> intro ==36=> index ==11=> rpm-Facilities ==300=> mission ==68=> dep ==166=> AMSAT ==300=> Training_Department ==0=> Technical_Services ==8=> IT ==3=> Conferences ==2=> Contact_us ==12=> mpm07 ==11=> AMSAT06 ==46=> NewsLetter ==300=> rpm-Projects ==3=> rpm-TechnicalServices ==8=> rpm-Publications ==6=> rpm-Staff ==8=> rpmContact_Us
62.178.10.113	No of Visits	Path _ Time per Seconds _
	3	intro ==40=> index ==15=> rpm-Facilities ==300=> mission ==2=> dep ==40=> AMSAT ==300=> Technical_Services ==13=> Training_Department ==14=> IT ==9=> Conferences ==5=> Contact_us ==6=> NewsLetter ==300=> italy ==6=> AMSAT06 ==87=> rpm-Activities ==1=> rpm-Projects ==2=> rpm-TechnicalServices ==1=> rpm-Publications ==1=> rpm-Staff ==4=> rpmContact_Us ==300=> Minerals_Characterization_Laboratory ==18=> Beneficiation_Laboratory ==6=> Steel_Technology_Laboratory ==1=> NonFerrous_Laboratory ==2=> Plastic_Deformation_Laboratory ==1=> Corrosion_Laboratory ==3=> Electronic_Materials_Laboratory ==1=> Nanostructured_Materials_Laboratory ==5=>

		Metals_Technology_Department ==3=> Advanced_Materials_Department ==13=> Manufacturing_Technology_Department ==5=> CulturalProgram ==1=> 2circular ==2=> Registration ==1=> Exhibition ==0=> Instructions ==298=> Mechanical_Tests ==28=> Devices ==1=> Registration ==7=> Training_Courses ==1=> NewsLetter-3-2006-pic ==292=> welcome
64.71.164.125	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index ==300=> mpm07 ==300=> chairman ==101=> IT ==143=> AMSAT06 ==79=> dep ==93=> Contact_us ==300=> Advanced_Materials_Department ==13=> welcome ==198=> AMSAT ==300=> Mechanical_Tests ==81=> Training_Courses ==300=> library ==300=> Devices
64.71.164.125	No of Visits	Path _ Time per Seconds _
	2	Training_Courses _ 2004-2005 ==300=> NewsLetter-3-2006-pic ==300=> CulturalProgram ==279=> RegForm ==300=> schedule ==113=> SubAndinst ==300=> Postconferencetours ==104=> CulturalProgramTour ==170=> AccompanyingPersonsProgram
65.214.44.45	No of Visits	Path _ Time per Seconds _
	1	chairman ==300=> rpm-Activities ==300=> rpm- Facilities
65.214.44.45	No of Visits	Path _ Time per Seconds _
	2	rpm-Projects ==300=> Corrosion __ Activities ==300=> Corrosion __ Equipments ==300=> rpmContact_Us ==300=> Corrosion __ Projects ==300=> casting-Projects
65.214.44.45	No of Visits	Path _ Time per Seconds _
	3	welding-Projects ==300=> Corrosion __ Contact_Us ==300=> welding-Contact_Us ==300=> Exhibition ==300=> Registration ==300=> welding-Facilities ==300=> Corrosion __ Publications ==300=> casting- Facilities ==300=> welding-Publications ==300=> welding-Activities
65.214.44.45	No of Visits	Path _ Time per Seconds _
	4	Nanostructured_Materials __ Staff ==300=> intro ==300=> Registration ==300=> NewsLetter ==300=> casting-Staff ==300=> casting-Activities
65.214.44.45	No of Visits	Path _ Time per Seconds _

	5	Composite-Projects ==300=> Composite-Facilities ==300=> Information ==300=> Devices ==300=> casting-Contact_Us ==300=> casting-Publications ==300=> CulturalProgram ==300=> casting-TechnicalServices ==300=> Training_Courses_ - 2004-2005
65.54.165.35	No of Visits	Path _ Time per Seconds _
	1	AMSAT06 ==254=> Technical_Services ==0=> IT ==8=> chairman ==13=> Contact_us ==1=> Training_Department ==8=> Conferences ==3=> mission
65.54.165.35	No of Visits	Path _ Time per Seconds _
	2	Instructions ==0=> 2circular
65.54.165.36	No of Visits	Path _ Time per Seconds _
	1	welcome ==300=> Training_Courses ==300=> Registration
65.54.165.36	No of Visits	Path _ Time per Seconds _
	2	library ==300=> Mechanical_Tests ==300=> Minerals_Processing_and_Technology_Department ==300=> Advanced_Materials_Department ==300=> Manufacturing_Technology_Department
65.54.165.36	No of Visits	Path _ Time per Seconds _
	3	IT22 ==21=> IT-staff ==300=> Exhibition
65.54.165.36	No of Visits	Path _ Time per Seconds _
	4	Training_Courses_ - 2004-2005 ==0=> Training_Department ==6=> Training_Courses
65.54.188.60	No of Visits	Path _ Time per Seconds _
	1	NonFerrous Activities ==300=> IT22
65.54.188.60	No of Visits	Path _ Time per Seconds _
	2	Plastic-Deformation-Projects ==300=> welding-Contact_Us
65.54.188.60	No of Visits	Path _ Time per Seconds _
	3	Composite-Projects ==300=> Steel_Technology_Projects ==300=> index_sub_dream ==13=> index_sub_golive
65.54.188.60	No of Visits	Path _ Time per Seconds _
	4	index_sub ==0=> index_golive
65.54.188.60	No of Visits	Path _ Time per Seconds _
	5	welcome ==300=> rpm-Activities

65.54.188.60	No of Visits	Path _ Time per Seconds _
	6	Minerals_Characterization__Facilities ==300=> rpm-Publications
65.54.188.60	No of Visits	Path _ Time per Seconds _
	7	Minerals_Characterization__Publications ==300=> Electronic_Materials__Projects
65.54.188.60	No of Visits	Path _ Time per Seconds _
	8	Mechanical_Tests ==300=> index_sub_golive ==300=> index_sub_dream
65.54.188.60	No of Visits	Path _ Time per Seconds _
	9	NonFerrous__Staff ==300=> NonFerrous__Activities ==72=> NonFerrous__Technical_Services
65.54.188.61	No of Visits	Path _ Time per Seconds _
	1	index_sub_dream ==300=> index_sub ==0=> index_sub_front
65.54.188.61	No of Visits	Path _ Time per Seconds _
	2	Pyrometallurgy-Staff ==300=> IT-staff ==300=> KSaad
65.54.188.61	No of Visits	Path _ Time per Seconds _
	3	TechnicalServices ==300=> ContactUs ==7=> Facilities ==300=> Chemical_and_Electrometallurgy_-_Projects ==0=> Chemical_and_Electrometallurgy_-_Contact_Us ==300=> Corrosion_Staff
65.54.188.61	No of Visits	Path _ Time per Seconds _
	4	Training_Courses ==50=> Registration ==300=> Exhibition
65.54.188.61	No of Visits	Path _ Time per Seconds _
	5	Pyrometallurgy-Publications ==16=> PyrometallurgyContact_Us ==300=> welding-Facilities
65.54.188.61	No of Visits	Path _ Time per Seconds _
	6	NonFerrous_Laboratory ==300=> Training_Department
65.54.188.61	No of Visits	Path _ Time per Seconds _
	7	Beneficiation__Publications ==300=> Pyrometallurgy
65.54.188.61	No of	Path _ Time per Seconds _

	Visits	
	8	casting-TechnicalServices ==300=> library
65.54.188.61	No of Visits	Path _ Time per Seconds _
	9	Conferences ==300=> AMSAT06
65.54.188.61	No of Visits	Path _ Time per Seconds _
	10	casting-Contact_Us ==2=> casting-Staff ==300=> Electronic_Materials__Contact_Us ==0=> Electronic_Materials__Facilities ==2=> Electronic_Materials__Technical_Services
65.54.188.61	No of Visits	Path _ Time per Seconds _
	11	Devices ==300=> Technical_Services
65.54.188.61	No of Visits	Path _ Time per Seconds _
	12	index_dream ==300=> index_sub_dream ==0=> index_sub_front ==3=> index_sub
65.54.188.62	No of Visits	Path _ Time per Seconds _
	1	index ==300=> index_golive ==0=> index_front ==300=> index
65.54.188.62	No of Visits	Path _ Time per Seconds _
	2	IT-contact ==300=> Nanostructured_Materials__Projects ==300=> Electronic_Materials__Activities
65.54.188.62	No of Visits	Path _ Time per Seconds _
	3	Beneficiation__Contact_Us ==300=> Activities ==11=> Publications
65.54.188.62	No of Visits	Path _ Time per Seconds _
	4	Beneficiation__Projects ==0=> Beneficiation__Technical_Services ==2=> Beneficiation__Facilities ==300=> Preliminarylist
65.54.188.62	No of Visits	Path _ Time per Seconds _
	5	Plastic-Deformation-TechnicalServices ==0=> Plastic-Deformation-Staff ==300=> Postconferencetours ==300=> Nanostructured_Materials__Staff ==300=> Ceramic_Materials_-_Projects ==300=> Plastic_Deformation_Laboratory
65.54.188.62	No of Visits	Path _ Time per Seconds _
	6	index_sub_front ==300=> casting-Activities
65.54.188.62	No of Visits	Path _ Time per Seconds _
	7	Ceramic_Materials__Laboratory ==300=>

		rpmContact_Us ==300=> Composite
65.54.188.62	No of Visits	Path _ Time per Seconds _
	8	NewsLetter ==300=> Steel_Technology__Activities ==300=> rpm-Projects
65.54.188.62	No of Visits	Path _ Time per Seconds _
	9	chairman ==300=> index_sub_front ==300=> index
65.54.188.62	No of Visits	Path _ Time per Seconds _
	10	index_front ==0=> index ==300=> intro
65.54.188.62	No of Visits	Path _ Time per Seconds _
	11	italy ==300=> Corrosion _ Projects
65.54.188.63	No of Visits	Path _ Time per Seconds _
	1	Bahgat ==300=> IT-staff ==300=> Corrosion _ Publications
65.54.188.63	No of Visits	Path _ Time per Seconds _
	2	Composite-Publications ==300=> Beneficiation _ Laboratory
65.55.208.109	No of Visits	Path _ Time per Seconds _
	1	SubAndinst ==0=> schedule
65.55.208.111	No of Visits	Path _ Time per Seconds _
	1	Mechanical _ Tests ==299=> casting-Publications
65.55.208.112	No of Visits	Path _ Time per Seconds _
	1	Minerals _ Characterization __Activities ==300=> Minerals _ Characterization _ Technical _ Services
65.55.208.113	No of Visits	Path _ Time per Seconds _
	1	Training _ Department ==1=> NewsLetter
65.55.208.114	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> GUC
65.55.208.90	No of Visits	Path _ Time per Seconds _
	1	index_sub_golive ==12=> index_sub_dream
65.55.208.91	No of Visits	Path _ Time per Seconds _
	1	rpm-Staff ==0=> rpm-Publications
65.55.208.91	No of Visits	Path _ Time per Seconds _
	2	index_dream ==300=> casting
65.55.208.91	No of Visits	Path _ Time per Seconds _

	3	index_sub ==300=> index
65.55.208.91	No of Visits	Path _ Time per Seconds _
	4	NonFerrous__Activities ==2=> NonFerrous Technical Services
65.55.208.92	No of Visits	Path _ Time per Seconds _
	1	Pyrometallurgy-Staff ==300=> Minerals Characterization Laboratory
65.55.208.92	No of Visits	Path _ Time per Seconds _
	2	Chemical_and_Electrometallurgy _ Contact_Us ==1=> Chemical_and_Electrometallurgy _ Projects
65.55.208.92	No of Visits	Path _ Time per Seconds _
	3	Training_Courses ==300=> Electronic_Materials__Facilities ==55=> Electronic_Materials__Technical_Services
65.55.208.93	No of Visits	Path _ Time per Seconds _
	1	IT22 ==300=> ContactUs ==0=> TechnicalServices ==51=> Facilities
65.55.208.93	No of Visits	Path _ Time per Seconds _
	2	casting-Contact_Us ==0=> casting-Staff ==300=> index_sub_dream
65.55.208.94	No of Visits	Path _ Time per Seconds _
	1	Composite-Staff ==219=> Nanostructured_Materials_Publications
65.55.208.94	No of Visits	Path _ Time per Seconds _
	2	Corrosion__Contact_Us ==300=> RegForm ==300=> AMSAT
65.55.208.94	No of Visits	Path _ Time per Seconds _
	3	Electronic_Materials_Laboratory ==300=> Devices
65.55.208.94	No of Visits	Path _ Time per Seconds _
	4	Minerals_Processing_and_Technology_Department ==300=> index_sub_front ==0=> index_sub
65.55.208.95	No of Visits	Path _ Time per Seconds _
	1	CulturalProgramTour ==300=> Chemical_and_Electrometallurgy_Laboratory
65.55.208.95	No of Visits	Path _ Time per Seconds _
	2	Steel_Technology__Activities ==300=> Pyrometallurgy-Projects ==300=> welding-Publications

65.55.208.95	No of Visits	Path _ Time per Seconds _
	3	Beneficiation__Technical_Services ==300=> casting-Activities ==300=> index_front
65.55.208.96	No of Visits	Path _ Time per Seconds _
	1	Composite-Activities ==300=> IT-contact
65.55.208.96	No of Visits	Path _ Time per Seconds _
	2	Composite ==300=> Nanostructured_Materials_Activities ==300=> italy
65.55.208.96	No of Visits	Path _ Time per Seconds _
	3	Publications ==0=> Activities
65.55.208.96	No of Visits	Path _ Time per Seconds _
	4	index_front ==22=> index_golive
65.55.208.96	No of Visits	Path _ Time per Seconds _
	5	welding-Activities ==300=> Training_Department
65.55.208.97	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==300=> Corrosion_Publications ==300=> Nanostructured_Materials_Staff ==0=> Nanostructured_Materials_Projects ==2=> Chemical_and_Electrometallurgy_Activities
65.55.212.65	No of Visits	Path _ Time per Seconds _
	1	RegForm ==300=> Minerals_Characterization_Technical_Services ==5=> Minerals_Characterization_Activities ==300=> Electronic_Materials_Activities ==300=> Minerals_Characterization_Projects ==300=> index ==2=> Instructions ==1=> Exhibition ==1=> schedule ==1=> SubAndinst ==1=> Chemical_and_Electrometallurgy_Projects ==224=> Chemical_and_Electrometallurgy_Contact_Us ==300=> CulturalProgramTour ==3=> Postconferencetours
65.55.212.65	No of Visits	Path _ Time per Seconds _
	2	intro ==2=> Steel_Technology_Facilities ==2=> Steel_Technology_Activities ==1=> Steel_Technology_Contact_Us ==57=> Steel_Technology_Projects ==2=> Minerals_Characterization_Facilities ==10=> Electronic_Materials_Technical_Services ==1=> Electronic_Materials_Publications ==1=> Electronic_Materials_Facilities ==1=> Electronic_Materials_Projects ==2=> Electronic_Materials_Staff ==1=>

		Electronic_Materials__Contact_Us ==300=> IT ==300=> Steel_Technology_Staff_ElFawakhry
65.55.212.65	No of Visits	Path _ Time per Seconds _
	3	Training_Department ==300=> Registration ==0=> Training_Courses ==300=> mission ==300=> chairman ==300=> NewsLetter
65.55.212.65	No of Visits	Path _ Time per Seconds _
	4	Training_Courses _ _2004-2005 ==238=> Training_Courses ==215=> Registration
65.55.212.65	No of Visits	Path _ Time per Seconds _
	5	mpm07 ==300=> welcome
65.55.212.65	No of Visits	Path _ Time per Seconds _
	6	2circular ==300=> Exhibition ==1=> CulturalProgram
65.55.235.140	No of Visits	Path _ Time per Seconds _
	1	dep ==300=> chairman
65.55.235.140	No of Visits	Path _ Time per Seconds _
	2	library ==300=> Advanced_Materials_Department
66.232.124.38	No of Visits	Path _ Time per Seconds _
	1	topics ==0=> Registration ==2=> registration ==2=> regform ==2=> subandinst
66.249.65.115	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> mission
66.249.65.174	No of Visits	Path _ Time per Seconds _
	1	welding-Projects ==300=> index
66.249.65.174	No of Visits	Path _ Time per Seconds _
	2	intro ==300=> index
66.249.65.174	No of Visits	Path _ Time per Seconds _
	3	Composite ==300=> index ==300=> mission
66.249.65.174	No of Visits	Path _ Time per Seconds _
	4	mission ==300=> Registration ==300=> Composite- Staff
66.249.65.174	No of Visits	Path _ Time per Seconds _
	5	welding-Publications ==300=> welding-Activities ==300=> Alber
66.249.65.174	No of Visits	Path _ Time per Seconds _

	6	KhalidHafez ==300=> AMSAT
66.249.65.193	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> RegForm
66.249.65.193	No of Visits	Path _ Time per Seconds _
	2	AMSAT ==300=> index ==300=> mission
66.249.72.7	No of Visits	Path _ Time per Seconds _
	1	index ==300=> mission
66.249.90.136	No of Visits	Path _ Time per Seconds _
	1	index ==300=> Conferences ==83=> Minerals_Processing_and_Technology_Department ==24=> Minerals_Characterization_Laboratory ==151=> Technical_Services
66.249.90.136	No of Visits	Path _ Time per Seconds _
	2	chairman ==47=> mission ==31=> Technical_Services ==93=> IT ==77=> Contact_us ==26=> library ==50=> Training_Department ==45=> Minerals_Processing_and_Technology_Department ==18=> Metals_Technology_Department ==31=> Steel_Technology_Laboratory ==20=> Steel_Technology_Staff ==39=> Manufacturing_Technology_Department ==25=> rpm-Staff ==39=> NonFerrous_Laboratory ==81=> Minerals_Characterization_Laboratory ==3=> Beneficiation_Laboratory ==11=> Chemical_and_Electrometallurgy_Laboratory ==11=> Minerals_Characterization_Staff ==44=> Beneficiation_Staff ==74=> dep ==36=> Beneficiation_Facilities ==300=> Beneficiation_Projects ==53=> Beneficiation_Technical_Services ==18=> Beneficiation_Publications ==250=> Minerals_Processing_and_Technology_Department ==7=> Metals_Technology_Department ==13=> Corrosion_Laboratory ==3=> NonFerrous_Laboratory ==4=> Plastic_Deformation_Laboratory ==5=> dep ==19=> Beneficiation_Activities ==23=> Beneficiation_Facilities ==60=> Beneficiation_Technical_Services ==9=> Beneficiation_Publications ==56=> Corrosion_Activities ==13=> Corrosion_Equipments ==41=> Corrosion_Publications ==24=> Corrosion_Staff ==22=> NonFerrous_Activities ==9=> NonFerrous_Facilities ==81=>

		NonFerrous__Publications ==35=> Steel_Technology__Activities ==25=> Steel_Technology__Facilities ==34=> Steel_Technology__Projects
67.169.58.234	No of Visits	Path _ Time per Seconds _
	1	NewsLetter ==34=> AMSAT06 ==29=> mpm07 ==67=> Contact_us ==97=> IT ==33=> Training_Department ==32=> Technical_Services ==34=> chairman ==30=> mission
68.151.114.132	No of Visits	Path _ Time per Seconds _
	1	Training_Courses ==65=> mission ==14=> dep ==13=> welding ==80=> Metals_Technology_Department ==102=> Contact_us ==287=> dep
68.151.114.132	No of Visits	Path _ Time per Seconds _
	2	Registration ==38=> Training_Courses - 2004-2005
68.50.118.183	No of Visits	Path _ Time per Seconds _
	1	intro ==39=> dep ==20=> Nanostructured_Materials_Laboratory
71.127.36.180	No of Visits	Path _ Time per Seconds _
	1	chairman ==8=> intro
72.36.146.50	No of Visits	Path _ Time per Seconds _
	1	Registration ==4=> amsat ==1=> Accompanying ==2=> CulturalProgram ==2=> Exhibition ==5=> Information ==1=> Instructions ==2=> RegForm ==2=> Registration ==3=> SubAndinst ==1=> culture_program ==2=> important_dates ==2=> organizing
74.124.192.201	No of Visits	Path _ Time per Seconds _
	1	Ceramic_Materials_- _Projects ==0=> Ceramic_Materials_- _Publications
74.139.203.227	No of Visits	Path _ Time per Seconds _
	1	AMSAT ==300=> dep ==28=> Manufacturing_Technology_Department ==38=> rpm-Activities
74.14.252.159	No of Visits	Path _ Time per Seconds _
	1	Chemical_and_Electrometallurgy_- _Activities ==1=> intro
80.11.150.47	No of Visits	Path _ Time per Seconds _
	1	chairman ==3=> intro

80.169.156.244	No of Visits	Path _ Time per Seconds _
	1	chairman ==1=> intro
81.10.87.249	No of Visits	Path _ Time per Seconds _
	1	index ==300=> Technical_Services ==248=> Training_Department
81.169.235.173	No of Visits	Path _ Time per Seconds _
	1	عطاءات ومناقصات ==11=> intro
81.183.142.130	No of Visits	Path _ Time per Seconds _
	1	contact ==37=> culture_program
81.21.97.8	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==60=> culture_program ==19=> topics
81.31.160.26	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> index ==34=> chairman
81.31.160.26	No of Visits	Path _ Time per Seconds _
	2	chairman ==4=> intro ==47=> chairman ==62=> dep ==64=> Metals_Technology_Department
82.103.138.223	No of Visits	Path _ Time per Seconds _
	1	AccompanyingPersonsProgram ==18=> amsat ==2=> Accompanying ==5=> CulturalProgram ==1=> Exhibition ==31=> Instructions ==4=> RegForm ==9=> Registration ==2=> SubAndinst ==6=> contact ==1=> culture_program ==2=> important_dates ==3=> organizing ==12=> schedule ==16=> submission ==1=> topics
82.146.166.137	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==63=> intro
82.194.62.227	No of Visits	Path _ Time per Seconds _
	1	intro ==31=> index
82.198.177.182	No of Visits	Path _ Time per Seconds _
	1	index ==300=> dep
82.201.170.62	No of Visits	Path _ Time per Seconds _
	1	intro ==192=> casting ==5=> welding ==201=> عطاءات ومناقصات
82.201.179.127	No of Visits	Path _ Time per Seconds _
	1	intro ==16=> index ==18=> Training_Department ==29=> Training_Courses_-_2004-2005
82.201.221.95	No of	Path _ Time per Seconds _

	Visits	
	1	Training_Courses ==15=> Training_Courses_-2004-2005
82.201.222.7	No of Visits	Path _ Time per Seconds _
	1	Activities ==8=> عطاءات ومناقصات
82.201.243.109	No of Visits	Path _ Time per Seconds _
	1	mpm07 ==83=> Contact_us ==162=> dep ==46=> library ==129=> Plastic_Deformation_Laboratory ==77=> Manufacturing_Technology_Department ==27=> welding
82.201.244.195	No of Visits	Path _ Time per Seconds _
	1	Activities ==114=> عطاءات ومناقصات
82.201.255.108	No of Visits	Path _ Time per Seconds _
	1	chairman ==5=> intro ==74=> Minerals_Processing_and_Technology_Department ==29=> intro ==2=> Minerals_Characterization_Laboratory ==18=> Minerals_Characterization_Contact_Us ==127=> intro ==10=> Metals_Technology_Department
82.89.230.197	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==1=> intro
83.101.150.116	No of Visits	Path _ Time per Seconds _
	1	library ==300=> Training_Department
84.0.219.228	No of Visits	Path _ Time per Seconds _
	1	intro ==4=> Minerals_Characterization_Staff
84.255.187.178	No of Visits	Path _ Time per Seconds _
	1	Training_Courses_-2004-2005 ==42=> Training_Courses
84.36.12.151	No of Visits	Path _ Time per Seconds _
	1	intro ==300=> Contact_us
84.36.147.227	No of Visits	Path _ Time per Seconds _
	1	intro ==14=> index
84.36.150.157	No of Visits	Path _ Time per Seconds _
	1	IT ==287=> Nanostructured_Materials_Laboratory ==26=> Nanostructured_Materials_Staff ==42=> Nanostructured_Materials_Publications
84.36.158.237	No of Visits	Path _ Time per Seconds _
	1	Corrosion_Equipments ==5=> intro

84.36.17.122	No of Visits	Path _ Time per Seconds _
	1	intro ==12=> index ==41=> culture_program ==12=> organizing ==62=> important_dates ==11=> topics
84.36.2.215	No of Visits	Path _ Time per Seconds _
	1	intro ==0=> library
84.36.20.228	No of Visits	Path _ Time per Seconds _
	1	important_dates ==9=> culture_program ==16=> topics ==35=> Accompanying
84.36.28.232	No of Visits	Path _ Time per Seconds _
	1	intro ==47=> index ==276=> mission ==66=> library ==89=> Conferences ==7=> GUC ==49=> italy ==73=> Contact_us ==76=> Registration ==20=> Training_Courses ==8=> Training_Courses - 2004-2005
84.54.27.5	No of Visits	Path _ Time per Seconds _
	1	Corrosion _ Projects ==27=> intro
85.103.2.173	No of Visits	Path _ Time per Seconds _
	1	chairman ==59=> dep ==24=> Manufacturing_Technology_Department ==143=> mission
85.249.139.82	No of Visits	Path _ Time per Seconds _
	1	Registration ==2=> amsat ==1=> Accompanying ==2=> CulturalProgram ==0=> Exhibition ==5=> Instructions ==1=> RegForm ==1=> Registration ==2=> SubAndinst ==1=> contact ==1=> culture_program ==0=> important_dates ==2=> organizing ==1=> schedule ==1=> submission ==1=> topics
87.101.244.9	No of Visits	Path _ Time per Seconds _
	1	index ==176=> عطاءات ومناقصات
87.118.112.30	No of Visits	Path _ Time per Seconds _
	1	amsat ==1=> Accompanying ==0=> CulturalProgram ==2=> Exhibition ==2=> RegForm ==0=> Instructions ==1=> SubAndinst ==0=> Registration ==1=> culture_program ==0=> contact ==2=> important_dates ==0=> organizing ==1=> schedule ==0=> submission ==1=> topics
88.116.163.106	No of Visits	Path _ Time per Seconds _
	1	Corrosion _ Equipments ==23=> intro ==70=>

	Corrosion__Projects
--	---------------------

A.3.2 Max Path Length

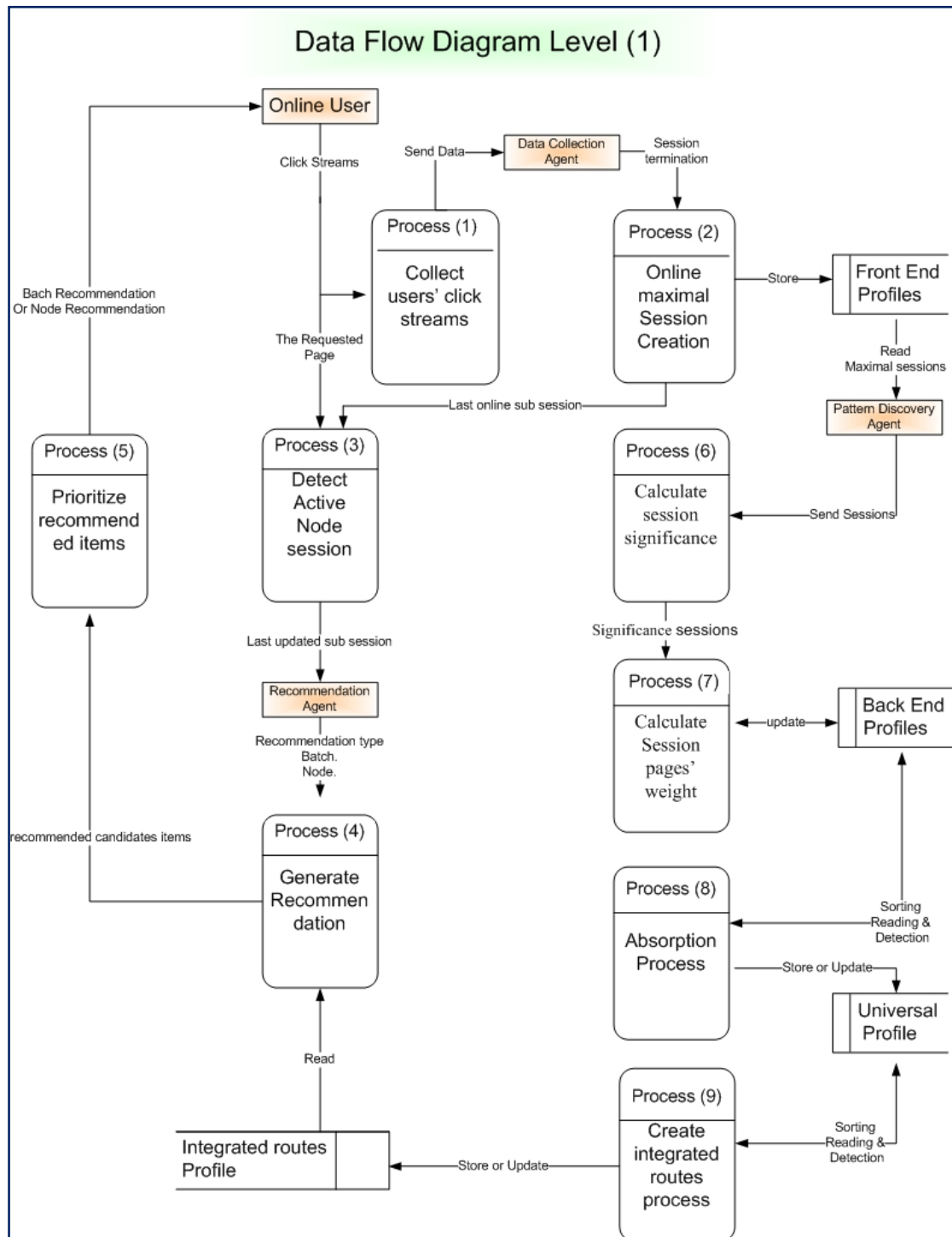
User_IP	User Max Path
62.114.59.241	<p>intro → index → italy → index → dep → Minerals_Processing_and_Technology_Department → Minerals_Characterization_Laboratory → Minerals_Characterization_Staff → Beneficiation_Projects → Pyrometallurgy-Staff → Chemical_and_Electrometallurgy_Laboratory → Chemical_and_Electrometallurgy_-_Projects → Corrosion_Laboratory → Corrosion__Projects → NonFerrous_Laboratory → NonFerrous__Projects → NonFerrous_Staff → Plastic-Deformation-Projects → Plastic-Deformation-Staff → Steel_Technology__Projects → Steel_Technology_Staff → Advanced_Materials_Department → Composite → Composite-Staff → Ceramic_Materials__Laboratory → Ceramic_Materials_-_Projects → Ceramic_Materials_-_Staff → Nanostructured_Materials_Laboratory → Nanostructured_Materials_Staff → Manufacturing_Technology_Department → casting-Staff → welding → rpm</p>

A.3.3 Min Path Length

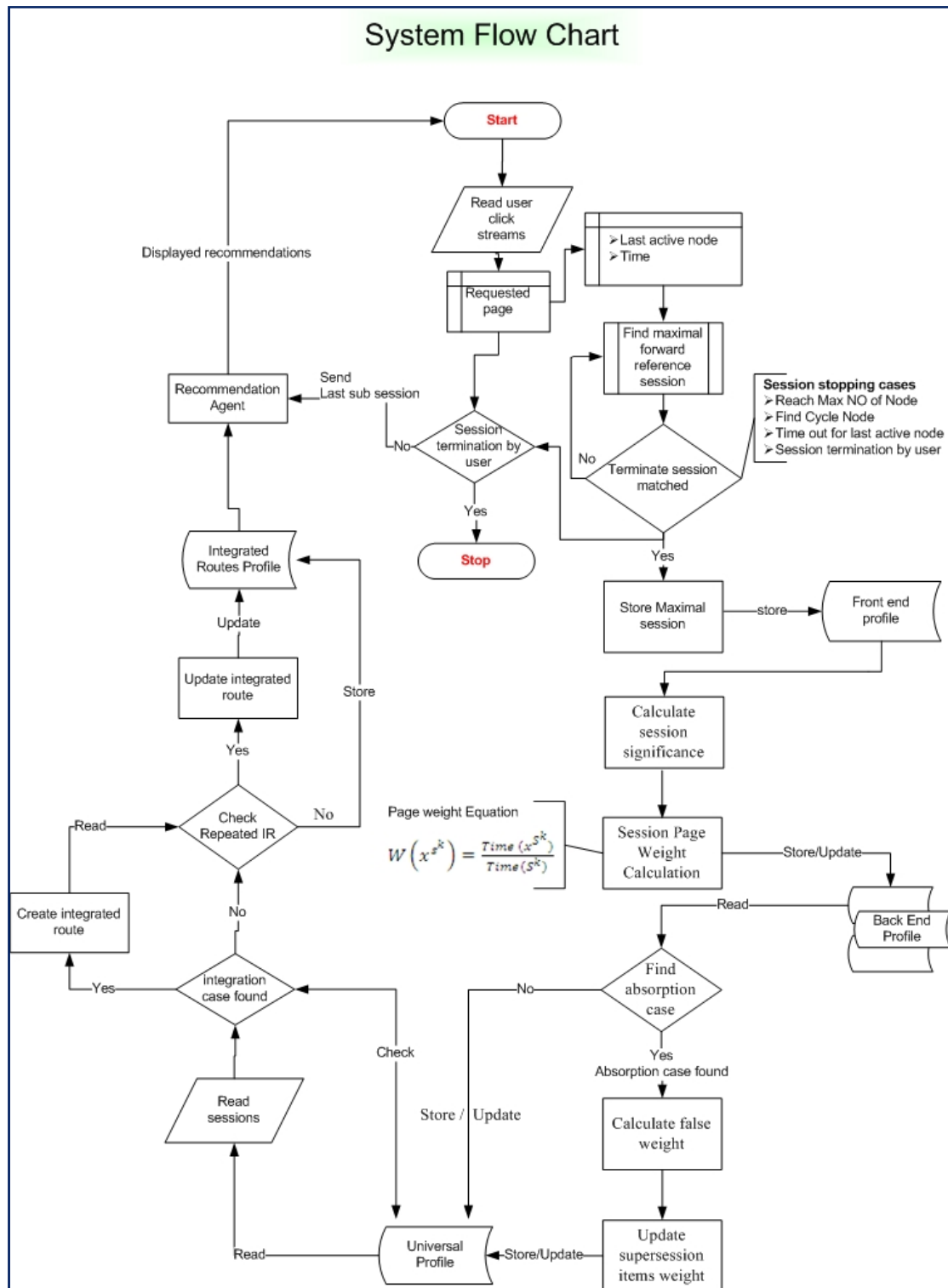
User_IP	User Min Path
122.152.129.9	intro → index

B. Suggested methodology modules

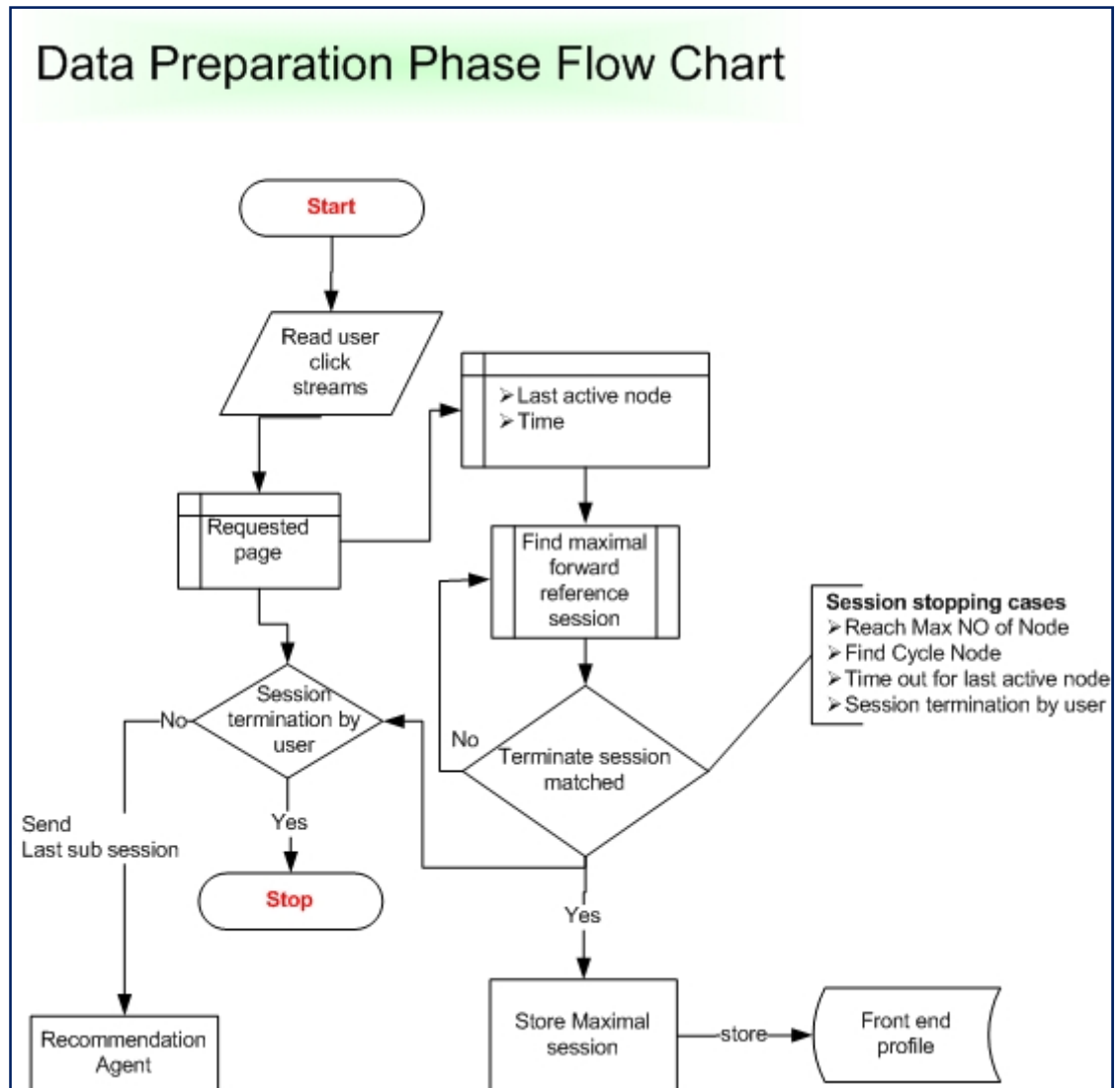
B.1 Data Flow Diagram Level (1)



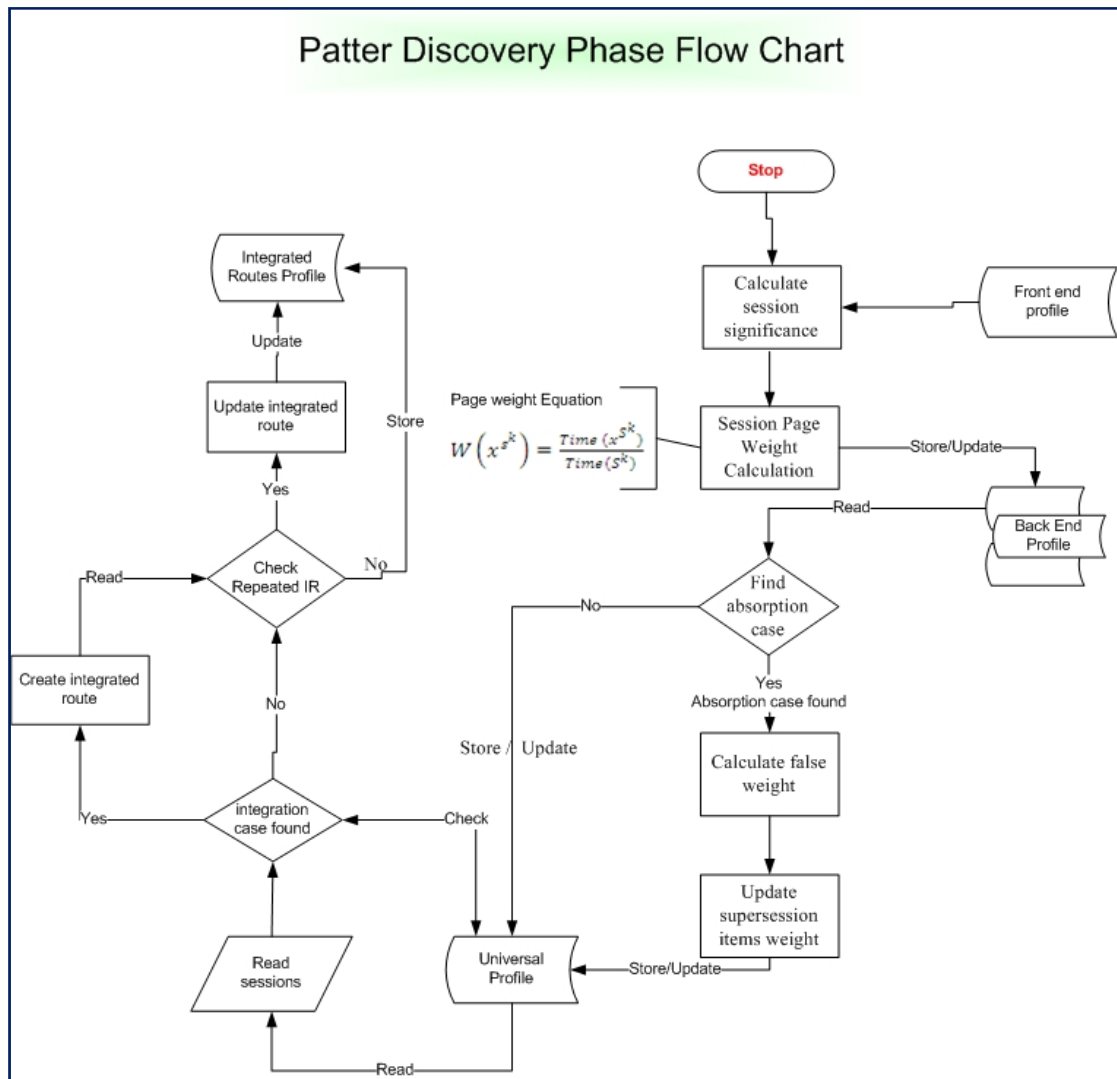
B.2 System Flow Chart



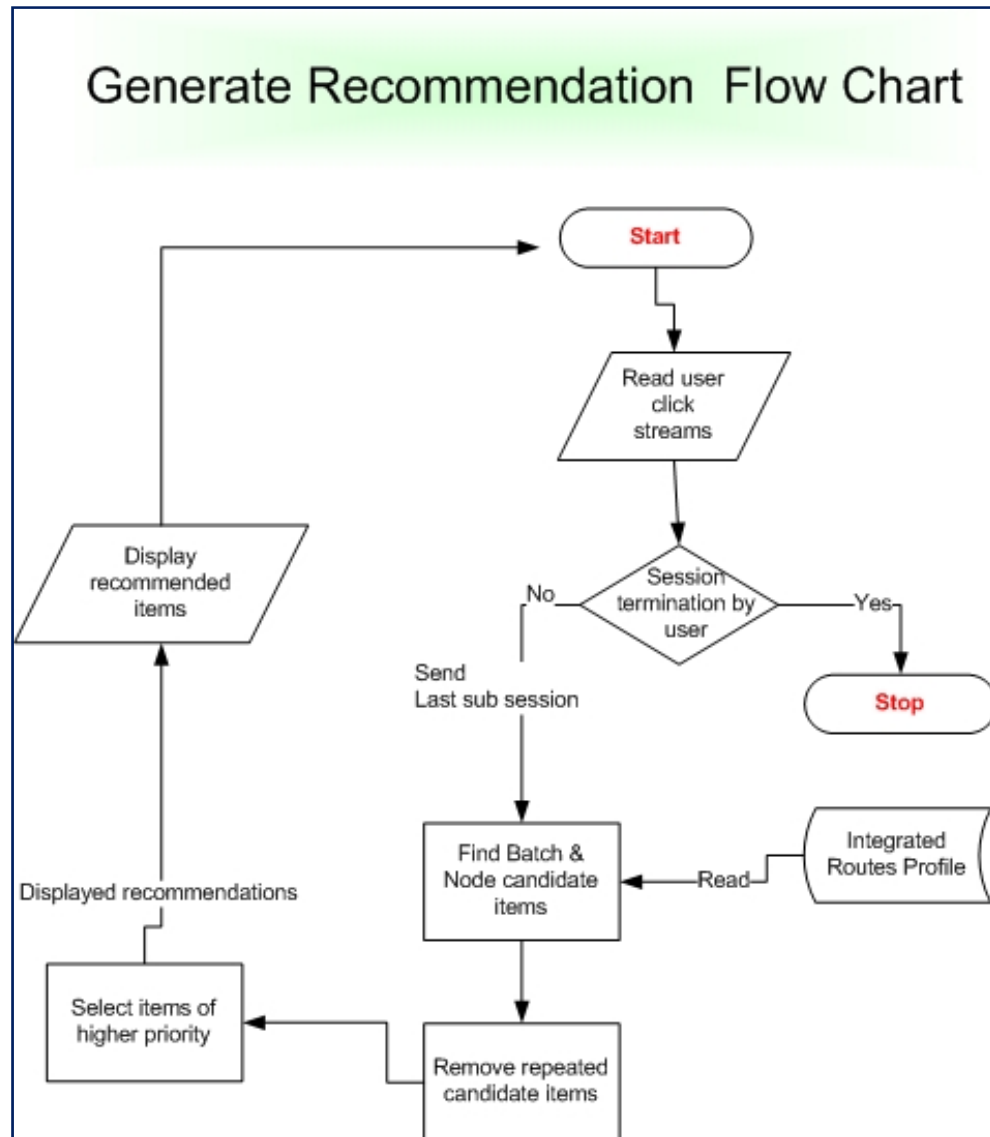
B.3 Data preparation Flow Chart



B.4 System pattern discovery flow chart



B.5 System recommendation flow chart



C. Abbreviations

<i>AN</i>	<i>Active Node</i>
<i>ANT</i>	<i>Active Node Technique</i>
<i>AP</i>	<i>Absorption Process</i>
<i>AR</i>	<i>Association Rule</i>
<i>BR</i>	<i>Batch Recommendation</i>
<i>CFS</i>	<i>Collaborative Filtering Systems</i>
<i>CL</i>	<i>Coverage Level</i>
<i>CMP</i>	<i>Current Online Maximal Path</i>
<i>CP</i>	<i>Current Maximal Online Path</i>
<i>CRS</i>	<i>Candidates Recommendation Set</i>
<i>CSP</i>	<i>Contiguous Sequential Patterns</i>
<i>DM</i>	<i>Data Mining</i>
<i>FW</i>	<i>False Weight</i>
<i>HTML</i>	<i>Hypertext Markup Language</i>
<i>HTTP</i>	<i>Hypertext Transport Protocol</i>
<i>IP</i>	<i>Internet Provider</i>
<i>IRP</i>	<i>Integrated Route Profile</i>
<i>MS</i>	<i>Maximal Sub Sessions Of Online Maximal Session</i>
<i>NNP</i>	<i>Null Weighted Nodes Profile</i>

<i>NR</i>	<i>Node Recommendation</i>
<i>NW</i>	<i>Null Weighted Recommendation Subset</i>
<i>OWL</i>	<i>Web Ontology Language</i>
<i>PC_S</i>	<i>Personal Computers</i>
<i>PF</i>	<i>Priority Factor</i>
<i>PL</i>	<i>Priority Level</i>
<i>PSW</i>	<i>Super Session Weight</i>
<i>RDF</i>	<i>Resource Description Framework</i>
<i>RDFS</i>	<i>Resource Description Framework Schema</i>
<i>RS</i>	<i>Recommendation Set</i>
<i>SAN</i>	<i>Semantic Active Node</i>
<i>SPW</i>	<i>Session Pages' Weight</i>
<i>SSW</i>	<i>Sub Session Weight</i>
<i>TS</i>	<i>Target Sets</i>
<i>UP</i>	<i>Universal Profile</i>
<i>URI</i>	<i>Universal Resources Identifier</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>W3C</i>	<i>World Wide Web Consortium</i>
<i>WWW</i>	<i>World Wide Web</i>
<i>XML</i>	<i>Extensible Markup Language</i>

D. A glossary of terms

<i>Term</i>	<i>Description</i>
The privacy problem	Reflects the users concerns regarding the misuse of their collected personal data.
The user cold-start problem	Happens when there is a new user in the system for who no rating information is available, and hence the system will unable to make recommendations or only create poor recommendations.
The item cold-start problem	Occurs when there is no rating information for a new added item to the web, and hence the system will unable to make recommendations or only create poor recommendations.
The system cold start problem	Occurs with the release of a new websites, where both user cold start and item cold start are applicable.
The Scalability problem	Computational resource required for generating recommendations are going beyond practical or acceptable levels with the increase in the number of candidate items for recommendations.
The Diversity problem	Reflect low level of users' satisfaction with a variety of items in the recommendation set.
The Data Sparseness problem	Refers to the fact that as the number of items increases only a small percentage of items will be rate by users.
Push attacks	Robustness problem: Promote a particular item by increasing its ratings for a larger subset of users.
Nuke attacks	Robustness problem: Reduce the predicted ratings of an item so that it is recommended to a smaller subset of users.
Average attack model	Robustness problem: assumes that the attacker knows the average rating for each item in the database and assigns values randomly distributed around this average, except for target item.
Random attack model	Robustness problem: form profiles by associating a positive rating for the target item with random values for the other items.

<i>Term</i>	<i>Description</i>
Implicit data collection	Users' preferences are inferred from their selections and click streams.
Explicit data collection	Explicit interaction with users in order to detect their interest
Task-focused personalization	Provides recommendations based on actions a user has taken while performing a task.
Profile-based personalization	Is a model-based approach, where users' patterns are collected and stored in their profiles and then used to generate recommendations
Web mining	Techniques focus on extracting knowledge about web contents, structure, and usage data.
Web usage mining	Reflects the data mining application that depends on the collected data from users' interactions with the web.
Rule-based systems	Such systems rely on manual or semi-automatic decision rules to generate recommendations for users.
Content-based systems.	Such systems rely on well-known information retrieval techniques which are used to find items features, or attributes, to be used later for generating recommendations.
Collaborative filtering systems	Such systems rely on web usage mining that use users' click-stream data to generate recommendations.
Hybrid systems.	Reflect the situation where recommendation system combines content and usage data to generate recommendation.
Clustering-based collaborative filtering	Aims to divide a data set into groups where inter-cluster similarities are minimized while the similarities within each cluster are maximized
Association rule based collaborative filtering	Reflect a useful tool for discovering correlations among items in a large database.

<i>Term</i>	<i>Description</i>
Front-end profile	Used to store online user's maximal sessions.
Back-end profile	Used to store maximal sessions items with its relative weights.
Universal profile	Used to store absorbed maximal sessions in the back-end profile.
Integrate route	A user-visited path that consists of one or more integrated maximal forward sessions.
Demographic data	Refers to specific user characteristics such as age, gender, income, religion, marital status, language, ownership (home, car, etc), and social position, etc.
Stereotype	Reflect a simplified and/or standardized conception or image with specific meaning, often held in common by people about another group.
Maximal forward session	Represents a loopless set of visited or selected items in sequential manner.
The active node	The online selected node /page.
The active path	The online visited path by current user.
Node recommendation	Generates recommendations based on the selected active node.
Batch recommendation	Generates recommendations based on the online visited path.

References

- ACKERMAN, M. S., CRANOR, L. & REAGLE, J. 1999. Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences. In: Proceedings of the 1st ACM conference on Electronic commerce. Denver, Colorado, USA, p.1-8.
- ADOMAVICIUS, G. & TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering, vol 17, Issue 6, p734-749.
- AGARWAL, R., AGGARWAL, C. & PRASAD, V. 2001. A tree projection algorithm for generation of frequent item sets. Journal of Parallel and Distributed Computing, Vol. 61, Issue 3., p. 350-371.
- AGGARWAL, C., WOLF, J., WU, K. & YU, P. 1999. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In: International Conference on Knowledge Discovery and Data Mining. San Diego, California, USA, p201-212.
- AGRAWAL, R. & SRIKANT, R. 1995. Mining sequential patterns. In: Proceedings of the 11th International Conference on Data Engineering. Taipei, Taiwan, p3-14.
- AHN, J., BRUSILOVSKY, P., GRADY, J., HE, D. & SYN, S. 2007. Open user profiles for adaptive news systems: help or harm?. In: Proceedings of ACM 16th international conference on World Wide Web. Banff, Alberta, Canada, p11 - 20.
- ANAND, S. & MOBASHER, B. 2005. Intelligent techniques for web personalization. Lecture notes in computer science. Springer-Verlag Publisher. Germany, p1-36.
- ANSARI, A., ESSEGAIER, S. & KOHLI, R. 2000. Internet recommendation systems. Journal of Marketing Research, Vol. 37, Issue 3, p363-375.
- ANTONIOU, G., & VAN HARMELEN, F. 2004. A semantic Web primer. Cooperative information systems. Cambridge, MIT Press. UK, p. 1-320.
- ARLEIN, R. M., JAI, B., JAKOBSSON, M., MONROSE, F. & REITER, M. K. 2000. Privacy-preserving global customization. In: Proceedings of the 2nd ACM

- conference on Electronic commerce. Minneapolis, Minnesota, USA, p176-184.
- BALABANOVIC, M. & SHOHAM, Y. 1997. Fab: content-based, collaborative recommendation. *Communications of ACM*, Vol. 40, Issue 3, p66-72.
- BAUMGARTEN, M., BÜCHNER, A., ANAND, S., MULVENNA, M. & HUGHES, J. 2000. User-Driven Navigation Pattern Discovery from Internet Data. In: MASAND, B. & SPILIOPOULOU, M. (eds.) *Web Usage Analysis and User Profiling*. Springer LNCS, Vol. 1836. Heidelberg , Berlin, p74-91.
- BELL, R. & KOREN, Y. 2007. Improved neighborhood-based collaborative filtering. In: *Proceedings of the KDD-Cup and Workshop at the 13th ACM*. San Jose, California, USA. P 7-14.
- BILLSUS, D. & PAZZANI, M. 1998. Learning collaborative information filters. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ,p46 - 54.
- BORGES, J. & LEVENE, M. 1999. Data Mining of User Navigation Patterns. In the *International Workshop on Web Usage Analysis and User Profiling, Lecture Notes In Computer Science*. Springer-Verlag, Vol. 1836 . London, UK, p92-111.
- BORGES, J. & LEVENE, M. 2004. A Dynamic Clustering-Based Markov Model for Web Usage Mining, *Computing Research Repository*, arXiv:cs/0406032v1.
- BRADLEY, K. & SMYTH, B. 2001. Improving Recommendation Diversity. In: *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science*. Ireland, p85-94.
- BREESE, J., HECKERMAN, D. & KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publisher. San Francisco, CA, USA, p43–52.
- BURKE, R. 2000. Knowledge-based recommender systems. In: *Encyclopedia of Library and Information Systems*. Marcel Dekker, Vol. 69. New York, USA, p.

- BURKE, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction Journal*, Vol. 12. Netherlands, p331-370.
- BURKE, R., HAMMOND, K., KULYUKIN, V., LYTINEN, S., TOMURO, N. & SCHOENBERG, S. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. Technical Report, University of Chicago. Chicago, IL, USA, p. 1-9.
- BURKE, R., MOBASHER, B. & BHAUMIK, R. 2005. Identifying attack models for secure recommendation. In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*. San Diego, California, USA, p19-25.
- CHAO, C., YANG, S., CHEN, P., & SUN, C. 2011. An Online Web Usage Mining System Using Stochastic Timed Petri Nets, In: *Proceeding of the Fourth International Conference on Ubi-Media Computing, u-media* , pp.241-246.
- CHEE, S., HAN, J. & WANG, K. 2001. Rectree: An efficient collaborative filtering method. In: *Data Warehousing and Knowledge Discovery (DaWaK)*. Springer, Lecture Notes in Computer Science, Vol. 2114. USA, p. 141-151
- CHELLAPPA, R. & SHIVENDU, S. 2007. Incentive design for free but no free disposal services: The case of personalization under privacy concerns. In: *The Workshop on the Economics of Information Security (WEIS)*. Atlanta , USA, P6-8.
- CHEN, T., HAN, W., WANG, H., ZHOU, Y., XU, B. & ZANG, B. 2007. Content Recommendation System Based on Private Dynamic User Profile. In: *Proceedings of the 6th International Conference on Machine Learning and Cybernetics*. Hong Kong, China, p. 2112–2118.
- CHO, Y. & KIM, J. 2004. Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Journal of Expert Systems with Applications*, Vol. 26, p. 233-246.

- CRANOR, L. 2002. I didn't buy it for myself—Privacy and ecommerce personalization. In: Proceedings of the ACM workshop on Privacy in the electronic society. Washington, DC, USA, p111 – 117.
- DAVID, N., CARSTEAN, C., PATRASCU, L., RATIU, I., MANDRU, L., 2010. Building solutions for web personalization. In: Proceeding of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases, World Scientific and Engineering Academy and Society, Stevens Point, Wisconsin, USA, p.201-210.
- DELANEY, D. & BROWN, S. 2002. Document Templates for Student Projects in Software Engineering. National University of Ireland. Maynooth, Co. Kildare, Ireland, p.70-300.
- DESHPANDE, M. & KARYPIS, G. 2004. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems, Vol. 22, p. 143-177.
- DESHPANDE, M. & KARYPIS, G. 2004. Selective Markov models for predicting Web page accesses. ACM Transactions on Internet Technology (TOIT), Vol. 4., p.163-184.
- FESENMAIER, D., RICCI, F., SCHAUMLLECHNER, E., WÖBER, K. & ZANELLA, C. 2003. DIETORECS: Travel advisory for multiple decision styles. Journal of Information and Communication Technologies in Tourism, Vol. 6., p.232–241.
- DRACHSLER, H., HUMMEL, H. & KOPER, R. 2007. Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. In: Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning & Exchange. Educational Technology Expertise Centre, Open University of the Netherlands. Netherlands, p.18-26.
- EIRINAKI, M. & VAZIRGIANNIS, M. 2003. Web mining for web personalization. in ACM Transactions on Internet Technology (TOIT). Vol. 3., p. 1-27.
- EIRINAKI, M., VAZIRGIANNIS, M. & VARLAMIS, I. 2003. SEWeP: using site semantics and a taxonomy to enhance the Web personalization process. ACM

Knowledge data discovery journal (KDD). Vol. 4., p. 99-108.

- EIRINAKI, M., VLACHAKIS, J. & ANAND, S. 2005. Ikum: An integrated web personalization platform based on content structures and user behavior. Intelligent Techniques for Web Personalization Journal (ITWP). Springer, Lecture Notes in Computer Science, Vol. 3169. Heidelberg, Berlin, p. 272-288.
- EMBARAK, O. & CORNE, D. 2011. Detecting Vicious Users in Recommendation Systems. 4th International Conference on Developments in E-Systems Engineering - DeSE2011. Abu Dhabi, UAE, 2011, Pending.
- ERKIN, Z., BEYE, M., VEUGEN, T., AND LAGENDIJK , R. 2010, Privacy enhanced recommender system. In Proceeding of 31st Symposium on Information Theory, Benelux, Rotterdam, pp. 35–42.
- ETZIONI, M. & PERKOWITZ, M. 2000. Towards adaptive Web sites: Conceptual frame work and case study. Artificial Intelligence Journal, Vol. 118. USA, p. 245-275.
- FESENMAIER, D., RICCI, F., SCHAUMLECHNER, E., WÖBER, K. & ZANELLA, C. 2003. Supporting travel decision making through personalized recommendation. Human-Computer Interaction Series, Designing personalized user experiences in eCommerce. Kluwer Academic Publishers. Norwell, USA, p. 231-251.
- FISCHER-HÜBNER, S. 2002. IT-security and privacy: Design and use of privacy enhancing security mechanisms. Lecture Notes In Computer Science, Vol. 1958. Simone Fischer-Hübner Karlstad University, Department of Computer Science. Karlstad, Sweden, p. 310-351.
- FU, X., BUDZIK, J. & HAMMOND, K. J. 2000. Mining navigation history for recommendation. In: Proceedings of the 5th international conference on Intelligent user interfaces. New Orleans, Louisiana, USA, p.106-112.
- GABRILOVICH, E., DUMAIS, S. & HORVITZ, E. 2004. News junkie: providing personalized newsfeeds via analysis of information novelty. In: Proceedings of the 13th international conference on World Wide Web. New York, USA, p.

- GHANI, R. & FANO, A. 2002. Building recommender systems using a knowledge base of product semantics. In 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. Malaga, Spain, p.10.
- GINTY, L. & SMYTH, B. 2002. Comparison-based recommendation. In: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning. Springer-Verlag, Vol. 2416. London, UK, p. 575 – 589.
- GIROLAMI, M. & KABÁN, A. 2003. On an equivalence between Probabilistic Latent Semantic Indexing (PLSI), and Latent Dirichlet Allocation (LDA). In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Association for Computing Machinery. Toronto, Canada, p. 433 - 434 .
- GOLDBERG, D., NICHOLS, D., OKI, B. & TERRY, D. 1992. Using collaborative filtering to weave an information tapestry. Communications of ACM, Vol. 35., Issue 12., p. 61-70.
- GOLDBERG, K., ROEDER, T., GUPTA, D. & PERKINS, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval Journal. Vol 4, Issue2., p.133-151.
- GOOD, N., SCHAFER, J., KONSTAN, J., BORCHERS, A., SARWAR, B., HERLOCKER, J. & RIEDL, J. 1999. Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference. Orlando, Florida, United States, p. 439 - 446.
- GUNAWARDANA, A. & MEEK, C. 2009. A unified approach to building hybrid recommender systems. In: Proceedings of the third ACM conference on Recommender systems. New York, USA, p.117-124 .
- HAASE, P., EHRIG, M., HOTH, A. & SCHNIZLER, B. 2004b. Personalized information access in a bibliographic peer-to-peer system. In: Proceedings of the

- AAAI Workshop on Semantic Web Personalization. AAAI Press. USA, p. 1-12.
- HAASE, P., EHRIG, M., HOTH, A. & SCHNIZLER, B. 2006. Personalized information access in a bibliographic peer-to-peer system. In: Peer-to-Peer and Semantic Web Decentralized Management and Exchange of Knowledge and Information. Springer. USA, p.143--158.
- HAN, J. & KAMBER, M. 2006. Data mining: concepts and techniques. Series in Data Management Systems. Morgan Kaufmann Publisher, ISBN.1558604898.
- HANSEN, M., SCHWARTZ, A. & COOPER, A. 2008. Privacy and identity management. IEEE Security and Privacy Magazine, Vol. 6 , Issue 2., P.38-45.
- HERLOCKER, J., KONSTAN, J., BORCHERS, A. & RIEDL, J. 1999. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. New York, USA, p. 230—237.
- HERLOCKER, J., KONSTAN, J., TERVEEN, L. & RIEDL, J. 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems Journal (TOIS), Vol. 22 , Issue 1., p. 5 – 53.
- HOFMANN, T. 1999. Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. Berkeley, California, USA, p.50 - 57
- HUANG, Z., CHEN, H. & ZENG, D. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Journal of Transactions on Information Systems (TOIS), Vol. 22 Issue 1., p.116 – 142.
- HUANG, Z., CHUNG, W., ONG, T. & CHEN, H. 2002. A graph-based recommender system for digital library. In: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries. Portland, Oregon, USA, p. 65-73.
- JAMALI, M. & ESTER, M. 2009. Using a trust network to improve top-N recommendation. In: Proceedings of the third ACM conference on Recommender systems. New York, USA, p.181-188.

- KALZ, M., DRACHSLER, H., VAN BRUGGEN, J., HUMMEL, H. & KOPER, R. 2008. Wayfinding services for open educational practices. *International Journal of Emerging Technologies in Learning (iJET)*, Vol. 3, Issue 2., p. 24-28.
- KALELI, C., POLAT, H. 2010. P2P collaborative filtering with privacy. *Turkish Journal of Electric Electrical Engineering and Computer Sciences*, Vol. 8, Issue 1., p101–116.
- KEARNEY, P., ANAND, S. & SHAPCOTT, M. 2005. Employing a domain ontology to gain insights into user behaviour. In: *Proceedings of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP05)*, in conjunction with the 19th international joint conference on Artificial Intelligence (IJCAI05). Edinburgh, UK, P.1-8.
- KOBSA, A. 2003. Pseudonymous yet Personalized Interaction with Websites that Utilize Network-wide User Modeling Services. In *HCIC Winter Workshop*, Winter Park. Colorado, USA, p1-37.
- KOHR, A. AND MERIALDO, B. 1999. Clustering for collaborative filtering applications. In: *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation*. IOS Press: Vienna, Austria, pp. 199–204.
- KOSALA, R. & BLOCKEEL, H. 2000. Web mining research: A survey. *ACM Knowledge Data Discover (KDD) Explorations Newsletter*, Vol. 2, Issue 1., p.1-15.
- KRULWICH, B. & BURKEY, C. 1996. Learning user information interests through extraction of semantically significant phrases. In: *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*. Stanford, California, USA, p.100–112.
- KRULWICH, B. 1997. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *ACM Artificial Intelligence Magazine*, Vol. 18 , Issue 2., p. 37-45.
- LAM, S. & RIEDL, J. 2004. Shilling recommender systems for fun and profit. In: *Proceedings of the ACM 13th international conference on World Wide Web*.

New York, USA, p. 393-402.

- LAM, X. N., VU, T., LE, T. D. & DUONG, A. D. 2008b. Addressing cold-start problem in recommendation systems. In: Proceedings of the ACM 2nd international conference on Ubiquitous information management and communication. Suwon, Korea, p. 208-211.
- LANG, K. 1995. Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning. Morgan Kaufmann publishers. San Mateo, CA, USA, p.331-339.
- LEE, B. 2010. Definition of semantic web in the glossary of W3C, <http://www.w3.org/People/Berners-Lee/Weaving/glossary.html>
- LEE, M., AND CHEUNG, C. 2009, User Satisfaction with an Internet-Based Portal: an Asymmetric and Non-linear Approach, Journal of the American Society for Information Science and Technology, Vol. 60, No. 1, pp. 111-122.
- LI, J. & ZAIANE, O. 2004. Using distinctive information channels for a mission-based Web recommender system. In: Proceedings of the 6th WEBKDD workshop: webmining and web usage analysis (WEBKDD04), in conjunction with the 10th ACM SIGKDD Conference (KDD'04). Seattle, USA, p.22-25.
- LIHONG, L., WEI, C., JOHN, L., AND ROBERT E. 2010. A contextual-bandit approach to personalized news article recommendation. In: Proceeding of 19th Intl. World Wide Web Conf. (WWW), New York, USA, p 661–670.
- LIN, W., ALVAREZ, S. & RUIZ, C. 2002. Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery Journal, Vol. 6, Issue 1., p. 83-105.
- LINDEN, G., SMITH, B. & YORK, J. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing Journal, Vol. 7, Issue 1., p.76-80.
- LORENZI, F. & RICCI, F. 2005. Case-based recommender systems: A unifying view. In: Intelligent Techniques for Web Personalization. Lecture Notes in Computer

Science, Volume 3169. Springer. Heidelberg, Berlin, p.89-113.

- MAES, P. 1994. Agents that reduce work and information overload. *Communications of ACM*, Vol. 37, Issue 7. New York, USA, p.30-40.
- MASSA, P. & AVESANI, P. 2004. Trust-aware collaborative filtering for recommender systems. In: *Proceedings of the International Conference on Cooperative Information Systems*. Agia Napa, Cyprus, p.492-508.
- MCCARTHY, K., REILLY, J., MCGINTY, L. & SMYTH, B. 2005. An Analysis of Critique Diversity in Case-Based Recommendation. In: *Proceedings of the Fifteenth International FLAIRS Conference*. San Mateo, CA, USA, p. 123-128
- MCGINTY, L. & SMYTH, B. 2005. Improving the performance of recommender systems that use critiquing. In: *Intelligent Techniques for Web Personalization*. Springer, Lecture Notes in Computer Science, Vol. 3169. Heidelberg, Berlin, p. 114-132.
- MELVILLE, P., MOONEY, R. & NAGARAJAN, R. 2002. Content-boosted collaborative filtering for improved recommendations. In: *18th national conference on Artificial intelligence*. American Association for Artificial Intelligence. Edmonton, Alberta, Canada, p.187-192.
- MICARELLI, A., GASPARETTI, F., SCIARRONE, F. & GAUCH, S. 2007. Personalized search on the world wide web. In: *The adaptive web: methods and strategies of web personalization*. Springer-Verlag. Heidelberg , Berlin, Lecture Notes In Computer Science, Vol.4321., p.195-230.
- MIDDLETON, S., SHADBOLT, N. & DE ROURE, D. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS) Journal*, Vol. 22 , Issue 1., p. 54 - 88.
- MIRZA, B., KELLER, B. & RAMAKRISHNAN, N. 2003. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, Vol. 20, Issue 2., p.131-160.

- MLADENIC, D. 1996. Personal WebWatcher: design and implementation. Technical Report, J. Stefan Institute. Ljubljana, Slovenia, p.1-6.
- MOBASHER, B., JIN, X., & ZHOU, Y. 2005. Task-Oriented Web User Modeling for Recommendation. In: Proceedings of the 10th International Conference on User Modeling (UM'05). Edinburgh, Scotland, p.109-118.
- MOBASHER, B. 2007. Data mining for web personalization. The adaptive web: methods and strategies of web personalization. Springer-Verlag. Heidelberg, Berlin, Lecture Notes in Computer Science, Vol. 4321, Issue 1., p. 90-135.
- MOBASHER, B., DAI, H., LUO, T. & NAKAGAWA, M. 2002. Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery Journal, Vol. 6, Issue 1. Kluwer Academic Publishers. Hingham, MA, USA, p.61-82.
- MOBASHER, B., DAI, H., LUO, T. & NAKAGAWA, M. 2001. Effective personalization based on association rule discovery from web usage data. In: Proceedings of the 3rd international workshop on Web information and data management. Atlanta, Georgia, USA, p. 9-15.
- MOBASHER, B., DAI, H., LUO, T. & NAKAGAWA, M. 2002. Using sequential and non-sequential patterns in predictive web usage mining tasks. In the Proceedings of the IEEE International Conference on Data Mining. IEEE Computer Society. Washington, DC, USA, p.669–672.
- MOBASHER, B., DAI, H., LUO, T., SUN, Y. & ZHU, J. 2000. Integrating web usage and content mining for more effective personalization. In: Proceedings of the 1st International Conference on Electronic Commerce and Web Technologies. Springer-Verlag, Lecture Notes In Computer Science, Vol. 1875. London, UK, p. 165-176.
- MOBASHER, B., JIN, X. & ZHOU, Y. 2004. Semantically enhanced collaborative filtering on the web. DePaul University, Center for Web Intelligence. Chicago, Illinois, USA.
- MONT, M., PEARSON, S. & BRAMHALL, P. 2003. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing

- services. In: Proceedings of the 14th International Workshop on Database and Expert Systems Applications. IEEE Computer Society. Washington, DC, USA , p. 377.
- MULVENNA, M., ANAND, S. & BÜCHNER, A. 2000. Personalization on the Net using Web mining: introduction. Communications of ACM, Vol. 43, Issue 8., p.122-125.
- NAKAGAWA, M. & MOBASHER, B. 2003. A hybrid web personalization model based on site connectivity. ACM Knowledge Data Discovery Journal (KDD), Vol. 6, Issue 1., p 59–70.
- NAKAGAWA, M. & MOBASHER, B. 2003. Impact of site characteristics on recommendation models based on association rules and sequential patterns. In: Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization. School of Computer Science, Telecommunication, and Information Systems. DePaul University. Chicago, Illinois, USA, p.1-8.
- NASRAOUI, O., KRISHNAPURAM, R., JOSHI, A. & KAMDAR, T. 2002. Automatic web user profiling and personalization using robust fuzzy relational clustering. In: E-commerce and intelligent methods, Physica-Verlag. Heidelberg, Pages 233-261.
- NEWMAN, D., ASUNCION, A., SMYTH, P. & WELLING, M. 2007. Distributed inference for latent Dirichlet allocation. Advances in Neural Information Processing Systems Journal, Vol. 20, Issue 1081–1088. Pennsylvania, USA, p. 17-24.
- NGUYEN, A., DENOS, N. & BERRUT, C. 2007. Improving new user recommendations with rule-based induction on cold user data. In: Proceedings of the ACM conference on Recommender systems. Minneapolis, USA, p. 121 - 128.
- NIU, L., YAN, X., ZHANG, C. & ZHANG, S. 2002. Product hierarchy-based customer profiles for electronic commerce recommendation. In: Proceedings of the 1st International Conference on Machine Learning and Cybernetics. Beijing, p.1075–1080.

- O'CONNOR, M. & HERLOCKER, J. 2001. Clustering items for collaborative filtering. ACM SIGIR Workshop on Recommender Systems, New Orleans, Louisiana, USA, p.1-4.
- O'MAHONY, M., HURLEY, N., KUSHMERICK, N. & SILVESTRE, G. 2004. Collaborative recommendation: A robustness analysis. ACM Transactions on Internet Technology (TOIT) Journal, Vol. 4, Issue 4. New York, USA, p.344-377.
- O'RIORDAN, A. & SORENSEN, H. 1995. An intelligent agent for high-precision text filtering. In: Proceedings of the fourth international conference on Information and knowledge management. Baltimore, Maryland, USA, p. 205-211.
- PADMANABHAN, B. & TUZHILIN, A. 1999. Unexpectedness as a measure of interestingness in knowledge discovery. Decision Support Systems Journal, Vol. 27, Issue 3. Amsterdam, The Netherlands, p.303-318.
- PARK, S., PENNOCK, D., MADANI, O., GOOD, N. & DECOSTE, D. 2006. Naïve filterbots for robust cold-start recommendations. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, USA, p. 699-705.
- PARK, S.-T. & CHU, W. 2009. Pairwise preference regression for cold-start recommendation. In: Proceedings of the third ACM conference on Recommender systems. New York, USA, p. 21-28.
- PAZZANI, M. & BILLSUS, D. 1997. Learning and revising user profiles: The identification of interesting web sites. Machine learning Journal, Special issue on multi-strategy learning, Vol. 27, Issue 3., p. 313-331.
- PAZZANI, M. & BILLSUS, D. 2007. Content-based recommendation systems. Lecture Notes In Computer Science, The adaptive web: methods and strategies of web personalization. Springer-Verlag. Heidelberg, Berlin, p. 325-341.
- PAZZANI, M. 1999. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review Journal, Vol. 13, Issue 5., p.393-408.

- PERKOWITZ, M. & ETZIONI, O. 2000. Towards adaptive web sites: Conceptual framework and case study. In: Proceedings of the 8th international conference on World Wide Web. Toronto, Canada, p. 1245-1258.
- PERKOWITZ, M. & ETZIONI, O. 2001. Adaptive web sites: Concept and case study. Artificial Intelligence Journal, Vol. 118, Issue 2., p.245-275.
- PITKOW, J. & PIROLI, P. 1999. Mining longest repeating subsequences to predict world wide web surfing. In: Proceedings of the 2nd conference on USENIX Symposium on Internet Technologies and Systems, Vol. 2. Boulder, Colorado, USA, p. 13-13
- POPESCU, A., UNGAR, L., PENNOCK, D. & LAWRENCE, S. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers. San Francisco, CA, USA, p. 437-444.
- POTTER, B. 2006. The Trusted Computing Revolution. Black Hat Journal. USA, p.1-20.
- POUWELSE, J., GARBACKI, P., WANG, J., BAKKER, A., YANG, J., IOSUP, A., EPEMA, D., REINDERS, M., VAN STEEN, M. & SIPS, H. 2008. Tribler: A social-based peer-to-peer system. Concurrency and Computation: Practice & Experience, Vol. 20, Issue 2., p. 127-138.
- RAMAKRISHNAN, N., KELLER, B., MIRZA, B., GRAMA, A. & KARYPIS, G. 2001. Privacy risks in recommender systems. IEEE Internet Computing Journal, Vol. 5, Issue 6., p. 54-62.
- RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P. & RIEDL, J. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the ACM conference on Computer supported cooperative work (CSCW '94'). Chapel Hill. North Carolina, USA, p.175-186.
- SALTON, G. & MCGILL, M. 1983. Introduction to modern information retrieval, ISBN:0070544840. McGraw-Hill, Inc.. New York, USA.

- SANDVIG, J., MOBASHER, B. & BURKE, R. 2007. Robustness of collaborative recommendation based on association rule mining. In: Proceedings of the ACM conference on Recommender systems (RecSys '07'). Minneapolis, MN, USA, p. 105-112.
- SANDVIG, J. MOBASHER, B. BURKE, R. 2008. A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms. IEEE Computer Society Technical Committee on Data Engineering. the National Science Foundation Cyber Trust program, p.1-11.
- SARUKKAI, R. 2000. Link prediction and path analysis using Markov chains. In: Proceedings of the 9th international World Wide Web conference on Computer networks : the International Journal of Computer and Telecommunications Networking. North-Holland Publishing Co. Amsterdam, The Netherlands, p. 377-386.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & REIDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. Hong Kong, China, p.285 – 295.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & RIEDL, J. 2000. Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM conference on Electronic commerce. Minneapolis, Minnesota, USA, p. 158-167.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & RIEDL, J. 2000. Application of dimensionality reduction in recommender system: a case study. In: Proceedings of The 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Web Mining for E-Commerce Challenges and Opportunities (WEBKDD'00'). Boston, Massachusetts, USA, p.1-12.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & RIEDL, J. 2002. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT'02'). Dhaka, Bangladesh, p.1-6.
- SCHECHTER, S., KRISHNAN, M. & SMITH, M. 1998. Using path profiles to predict HTTP requests. In: Proceedings of the 7th international conference on World

Wide Web. Elsevier Science Publishers. Brisbane, Australia, p.457-467.

- SCHEIN, A., POPESCU, A., UNGAR, L. & PENNOCK, D. 2002. Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. Tampere, Finland , p. 253-260..
- SCHWAB, I., KOBAS, A. & KOYCHEV, I. 2000. Learning about users from observation. In: Proceedings of AAAI Spring Symposium: Adaptive User Interface. Stanford University. California, USA, p.102-106.
- SHANI, G., MEISLES, A., GLEYZER, Y., ROKACH, L. & BEN-SHIMON, D. 2007. A Stereotypes-Based Hybrid Recommender System for Media Items. In: Proceedings of the AAAI Workshop on Web Recommendation. AAAI Press. USA, P1-8.
- SHARDANAND, U. & MAES, P. 1995. Social information filtering: algorithms for automating “word of mouth”. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, Addison-Wesley Publishing Co. New York, USA, p. 210-217.
- SHIMAZU, H. 2002. ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. Artificial Intelligence Review Journal, Vol. 18, Issue 4., p.223-244.
- SHOKRI, R., PEDARSANI, P., THEODORAKOPOULOS, G. & HUBAUX, J.-P. 2009. Preserving privacy in collaborative filtering through distributed aggregation of offline profiles. In: Proceedings of the third ACM conference on Recommender systems. New York, USA, p.157-164
- SIEG, A., MOBASHER, B. & BURKE, R. 2007. Web search personalization with ontological user profiles. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. Lisbon, Portugal, p. 525-534.
- SIEG, A., MOBASHER, B., BURKE, R., PRABU, G. & LYTINEN, S. 2005. Representing user information context with ontologies. In: Proceedings of HCI

International Conference. Las Vegas, Nevada, USA, p.210–217.

- SIEG, A., MOBASHER, B., BURKE, R. 2010. Ontology-Based Collaborative Recommendation. In: Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP), in conjunction with the International Conference on User Modeling, Adaptation, and Personalization (UMAP), BIG ISLAND OF HAWAII, p20-31.
- SILBERSCHATZ, A. & TUZHILIN, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, Issue 6., p. 970-974.
- SMYTH, B. 2007. Case-based recommendation. *The adaptive web: methods and strategies of web personalization*, Lecture Notes in Computer Science, Vol 2., p. 342-376 .
- SOLLENBORN, M. & FUNK, P. 2002. Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning. Springer-Verlag, Lecture Notes In Computer Science, Vol. 2416. London, UK, p. 395-405.
- SPERETTA, M. & GAUCH, S. 2005. Personalized search based on user search histories. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society. Washington, DC, USA, p.622 - 628.
- SPILIOPOULOU, M. & FAULSTICH, L. 1998. WUM: a tool for web utilization analysis. *the International Workshop on The World Wide Web and Databases*. Springer-Verlag, Lecture Notes In Computer Science, Vol. 1590. London, UK, p.184-203.
- SPILIOPOULOU, M., MOBASHER, B., BERENDT, B. & NAKAGAWA, M. 2003. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *Inform Journal on Computing*, Vol. 15, Issue 2., p.171-190.
- SRIVASTAVA, J., MOBASHER, B. & COOLEY, R. 2000. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, Vol. 43, Issue 8.,

p.142-151.

- STREHL, A. 2002. Relationship-based clustering and cluster ensembles for high-dimensional data mining. PhD thesis, The University of Texas at Austin. Texas, USA, P. 110-148.
- SU, X. & KHOSHGOFTAAR, T. 2006. Collaborative filtering for multi-class data using belief nets algorithms. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence. IEEE Computer Society Washington, DC, USA, p.497-504.
- TALABEIGI, M., FORSATI, R., MEYBODI, M., 2010. A dynamic web recommender system based on cellular learning automata. In Proceeding of 2nd International Conference in Computer Engineering and Technology (ICCET), Vol.7 , p755-761.
- TAN, P., KUMAR, V. & SRIVASTAVA, J. 2004. Selecting the right objective measure for association analysis. Information Systems Journal, Vol. 29 Issue 4., p.293-313.
- TANG, T., WINOTO, P. & CHAN, K. 2005. Scaling down candidate sets based on the temporal feature of items for improved hybrid recommendations. Intelligent Techniques for Web Personalization. Springer, Lecture Notes in Computer Science, Vol. 3169. Heidelberg, Berlin, p. 169-186.
- TSOW, A., KAMATH, S. & CAMP, L. 2007. A privacy-aware architecture for sharing web histories. IBM Systems Journal. Vol. 3. Indiana, USA, p.5-13.
- UNGAR, L. & FOSTER, D. 1998. Clustering methods for collaborative filtering. In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence (AAAI'98'). AAAI Press. Madison, Wisconsin, USA, p. 112-125.
- WEI, C. & PARK, S. 2009. Personalized recommendation on dynamic content using predictive bilinear models. In: Proceedings of the 18th International Conference on World Wide Web, ACM New York, USA, p. 691–700.

- WEITZNER, D., HENDLER, J., BERNERS-LEE, T. & CONNOLLY, D. 2006. Creating a policy-aware web: Discretionary, rule-based access for the world wide web. Web and Information Security. Idea Group Inc. Publisher. USA, p.201-208.
- WELD, D., ANDERSON, C., DOMINGOS, P., ETZIONI, O., GAJOS, K., LAU, T. & WOLFMAN, S. 2003. Automatically personalizing user interfaces. In: Proceedings of the 18th international joint conference on Artificial intelligence. Morgan Kaufmann Publishers Inc. Acapulco, Mexico, p.1613-1619.
- WK-XO, W. & RUJ, K. 2005. Privacy-Enhanced Personalization. In: Proceedings of the 10th international conference on user modeling. Edinburgh, Scotland, UK, p.225-241.
- XUE, G., LIN, C., YANG, Q., XI, W., ZENG, H., YU, Y. & CHEN, Z. 2005. Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. Salvador, Brazil, p.114-121.
- YANG, Y. 1994. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer. Dublin, Ireland, p.13-22.
- ZHANG, Y. & JIAO, J. 2007. An associative classification-based recommendation system for personalization in B2C e-commerce applications. Expert Systems with Applications. Vol. 33, Issue 2., p.357-367.
- ZHANG, Z., LIU, CHUANG., ZHANG, Y., AND ZHOU, T. 2010. Solving the cold-start problem in recommender systems with social tags. Europhysics Letters, Vol. 92, Number 2, p76-81.
- ZHOU, B., HUI, S. & CHANG, K. 2004. An intelligent recommender system using sequential web access patterns. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems. Piscataway, NJ, USA, p.393 - 398.
- ZIEGLER, C., LAUSEN, G. & SCHMIDT-THIEME, L. 2004. Taxonomy-driven computation of product recommendations. In: Proceedings of the 13th ACM

international conference on Information and knowledge management. ACM Association for Computing Machinery. Washington, D.C., USA, p. 406- 415.

ZIEGLER, C., MCNEE, S., KONSTAN, J. & LAUSEN, G. 2005. Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web. ACM Association for Computing Machinery. Chiba, Japan, p.22-32.