A Survey on Feature Recommendation Techniques

Syeda Nazema Syed. Subhan
Dept. of CS & IT
Dr. Babasaheb Ambedkar Marathwada University
Aurangabad, India
e-mail: nazemaverda@ygmail.com

S. N. Deshmukh
Dept. of CS & IT
Dr. Babasaheb Ambedkar Marathwada University
Aurangabad, India
e-mail: sndeshmukh@hotmail.com

Abstract—Recommendation systems are a very common now days and it is used in a variety of applications. A recommender system that is designed to reduce the human effort of performing domain analysis. Domain analysis is the task in which we can find the commonality and difference between the different software's of same domain 'feature recommendation is very useful now a days. This approach relies on data mining techniques to discover common features across products as well as the relationship among these common features.

In this paper we used different techniques which are used for domain analysis and feature recommendation. This approach mines descriptions of product from publicly available online product descriptions, uses a text mining and a novel incremental diffusive clustering algorithm to discover features in specific domain , uses association rule mining to know latent relationships between the features within the products of same domain and uses KNN algorithm which generates a probabilistic feature model that represents commonalities, variant.

Keywords- *Domain Analysis, Recommendation System, Feature Extraction, kNN (k-Nearest-Neighbor), association rule mining, Incremental diffusive clustering algorithm.*

*****

I. INTRODUCTION

Domain Analysis is a process of identifying and documenting the commonalities and variables to a particular domain, it is starting phase of software development life-cycle to generate ideas for software. Many domain analysis techniques are proposed, such as the Feature Oriented Domain Analysis (FODA) [1] or the Feature Oriented Reuse Method (FORM) [2] success of this approaches is based on upon the document available for existing project repositories. Another method such as domain analysis and reuse Environment (DARE) [3]. These approaches generally assume that analysts utilize existing required documentation existing requirement specifications or competitors' product brochures and websites, and then manually or semi-manually evaluate the documentation to extract features. Domain analysis is a labour-intensive task in which related software systems are analysed to discover their common and variable parts. Result is dependent upon the expertise of available analyst's requirements specifications from previous related software/ products available. To solve these problems, researchers used data mining and natural language processing (NLP) techniques for automate the process of mining features from requirements specifications.

A recommender system can be defined as any system that guides a user in a personalized way to interesting or useful objects in a large space of possible options or that produces such objects as output [5].

Recommendation system can be classified into three main categories, content-based Filtering (CB) Content-based approach. In this approach, similar items to the once the user saw in past will be recommended to the user for new recommendations. For example, if a customer of softpedia.com bought watches of titan, similar watches (related to Titan) will be proposed in future recommendation sessions. Collaborative filtering (CF) in second approach, items that other people with similar tastes and preferences likes will be recommended. Third approach is hybrid approach which is use to overcome the limitations of both above approach hybrid approach are proposed that combines both approaches. Hybrid algorithms attempt to combine CB with CF. The combination of content with rating data helps capture more effective correlations between users or items, which yield more accurate recommendations. There are various important applications of recommendation system some of them given below

i. **Product Recommendation:** People purchase products from Amazon or similar on-line vendors strive each user written some suggestions of products that they might like to buy. This suggestion is based on the purchasing decisions written by similar customers or on other techniques, as shown in Figure 1, the user is awarded with similar items that other customers have like to buy.

ii. **Movie Recommendations**: Netflix offers movies recommendations for customer they might like. These recommendations are based on **ratings** of movies which provided by users, an example of both content with rating data is MovieExplain, which combines the rating user profile and the feature item profile to disclose the favourite features of users as shown in Figure2.

iii. **News Articles:** News article recommendation is based on reader interested article, interest of reader is identify on the basis articles that they have read in the past. The **similarity** will be based on the similarity of important words in the articles. The

Figure 1. Example of Recommendation in Amazon

Our Justified Recommendation			
[Movie id]	[Movie Title]	[The reason is]	[because you rated]
1526	Witness (1955)	Ford, Harrison (I)	21 movies with this feature
1273	Code of Night (1994)	Wills, Bruce	7 movies with this feature
1004	Geronimo: An American Legend (1993)	Hackman, Gene	7 movies with this feature
1442	Scarlet Letter, The (1995)	Oldman, Gary	7 movies with this feature
1044	Paper, The (1994)	Close, Glenn	7 movies with this feature
693	Casino (1995)	De Niro, Robert	6 movies with this feature
274	Sabrina (1995)	Pollack, Sydney	6 movies with this feature
1092	Dear God (1996)	Kinnear, Greg	5 movies with this feature

Figure 2. Example of Recommendations in MoviExplain

same principles apply to recommending blogs from among the millions of blogs available, videos on YouTube, or other sites where content is provided regularly. Feature recommendation system it is beneficial in single application projects. Consider an analyst selecting requirements for a new audio player software system might evaluate the features provided by existing products. The identification and reuse of domain property could potentially reduce cost of development, time to electing features, improve quality, and increase product competitiveness. In this approach initial product description is taken as input, analyse this description and generate related features. For recommendation there are different algorithms of data mining are used as given in below section.

Domain analysis techniques are frequently available to organizations with existing products in the targeted domain. On the other hand, the online description of product on different websites Repositories mean that partial Descriptions of hundreds of thousands of products are now publicly available on the WebPages for particular domain. These product descriptions can be used in place of actual requirements to construct a new product of same domain.

II. DATA SETS

As we study previously proposed feature recommendation system in that we know the data is collected from different web sites.The online Descriptions of hundreds of thousands of products are now publicly available on the WebPages Collecting the data from different web sites by using web crawler. Data is collected from softpedia.com, softonic.com, googleapps.com etc. For experimental purposes, they mined from the 20 Softpedia product categories shown in Table I. for extracting important data from huge collection of data they proposed novel Incremental diffusive clustering algorithm[7].

Table 2 shows a chunk of the feature which are manually identified from softpedia.com. One which contain

TABLE I. PRODUCT CATEGORY

Product Category	Product Count	Feature Count
Antivirus	165	40
Audio-Players	283	46
Bookmark-Managers	92	33
Browsers	205	41
Coding-languages-Compilers	154	50
Compression-tools	159	40
Debuggers-Decompilers-Dissasemblers	118	46
Digital-Photo-Tools	314	46
Download-Managers	207	41
FTP-Servers	59	38
File-Sharing	278	44
File-managers	260	46
Multimedia-IPOD-tools	97	34
Office-suites	50	52
Password-Managers-Generators	281	38
Popup-Ad-Spyware-Blockers	211	35
Search-engine-tools-submitting	198	41
Text-editors	333	48
Video-Recording	150	37
WebCam	129	40

core features such as Automatic Updates and malware Protection. This table was manually compiled by researchers at DePaul University through inspecting Antivirus product listing from softpedia.com. It therefore only includes features listed on softpedia [11].

III. FEATURE MINING

Features must initially be mined from product specifications which are available from web pages. Negar Hariri et-al presented in there paper where they used 117,265 different products from 21 Softpedia categories. The description of the product is present as a huge collection of data so descriptions are parsed into sentences to form *feature descriptors* and then pre-processed stemming and stop-word removal.

Each feature descriptor is then converted into a vector space representation using the term frequency and inverse document frequency (TF–IDF) approach. As many

 ISSN: 2321-8169
1662 - 1667

products having similar features, but these features described in different ways, the descriptors must be clustered into consistent clusters such that each cluster resemblance to software features. The similarity of a pair of feature descriptors can be found by using the cosine similarity of their corresponding term frequency and inverse document frequency (TF–IDF) vectors. This similarity measure can be used in any intercourse clustering algorithm such as K - means, Kmedoid [18], or spherical K-means [20] to group similar feature descriptors. Then they proposed new approach i.e. *incremental diffusive clustering* (IDC) approach [7] for feature extraction into 1,135 clusters, By August, 2010, they included over 796,536 different software product, with over 1.5 billion downloads and categorized under 9 primary groupings, 292 sub categories, and 1,096 product types.

This algorithm used a heuristic approach to calculate the number of clusters. This algorithm tends to outperform other algorithms, including K-means, spherical K-means, and latent Dirichlet allocation (LDA) [19] for

TABLE II. A SAMPLE OF FEATURES COLLECTE FINDTHEBEST.COM ANTIVIRUS PRODUCTS DESCRIPTION

	Ad-Aware Free	Ad-Aware Personal	Ad-Aware Pro Security	Ad-Aware Total	Advanced System Care	Advanced System Care	Advanced System Care	Advanced System Care	Advanced System Care	Advanced SystemCare 8	Airy AntiSpyware	Airy AntiSpyware Pro	Airy Free AntiSpyware	Avast Free Antivirus	Avast Premier	avast! Free Antivirus	avast! Free Antivirus for	avast! Internet Security	avast! Pro Antivirus	Avetix	Avetix Free	Avetix Pro	AVG Antivirus	AVG AntiVirus for Mac	AVG AntiVirus Free	AVG Anti-Virus Free	AVG Antivirus	AVG Antivirus 2.5	AVG Internet Securit	AVG Internet Security	Avira Antivirus Pro	Avira Free Antivirus	Avira Free Antivirus for	Avira Free Antivirus for	Avira Free Mac Security	Avira Internet Security	Avira Internet Security	Avira Internet Security	Avira Ultimate
Anti_Dialer														•		•				•						•													
Anti_Rootkit				•							•	•	•	•		•			•	•		•				•	•			•		•				•	•		
Anti_Spam	•		•	•		•	•			•		•	•	•	•			•	•			•				•			•	•							•		•
Anti_Spyware/Adware		•	•	•	•	•	•			•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•		•
AntiVirus_Protection	•	•	•	•	•	•	•		•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•		•		•
Automatic_Updates	•	•	•	•		•	•		•					•		•		•	•	•	•	•	•	•		•	•		•				•			•	•		•
Boot_Disk	•																																						
Boot_Time_Scan																•		•	•																				
Bot_Blocker	•												•																										
Easy_&_user_friendly_interface	•		•						•		•		•	•		•	•	•	•	•					•		•		•			•				•	•		•
Email_Protection	•	•	•	•													•	•		•	•					•	•		•	•	•	•				•	•		•
Email_Support	•	•	•	•							•		•			•	•	•	•	•	•				•				•						•	•	•		•
FAQ's	•	•	•	•									•			•		•	•	•					•		•		•		•						•		
File_Encryption														•		•			•	•						•	•			•					•				•
File_Shredding				•										•		•	•		•						•	•	•		•	•									•
Firewall	•	•	•	•									•	•	•			•	•		•								•	•	•	•				•	•		•
Game_Mode	•	•	•	•														•	•	•							•		•							•	•		
Identity_Protection		•	•	•				•					•	•	•	•	•	•	•		•		•				•		•	•					•			•	•
Instant_Message_Protection		•		•										•		•	•		•	•					•		•		•										
Keyloggers														•					•	•	•																		

clustering requirements [11], each feature required to be meaningfully and proper named. One approach uses the *medoid* as the name. The medoid is the descriptor that is most representative of the feature's theme. The medoid is find by first computing the cosine similarity between each descriptor and the centroid of the cluster and then summing up the different weighted values in the descriptor's term vector for all values above a certain threshold which is already define. Then scores are normalized and then added together for each descriptor. The descriptor is selected as the feature name whose score is highest. This approach produces proper meaningful names. As a feature based on the theme of *update, disk, malware protection, update support.*

IV. FEATURE RECOMMENDATION PROCESS

The process of recommendation system and component required in feature recommendation is given in below paragraph. Figure 3 illustrates the overall process, which consisting of an initial training phase as well as a usage phase. In the first step screen scrapper utility used to extracted features from online product descriptions and the feature recommender is trained, while in the next steps the trained system makes recommendations based on an initial description of the product provided by a requirements analyst or other users of the system.

The first phase mining product description from online software product repositories. For example, feature descriptors could be retrieved from Softonic.com which contains a large collection of software products. In the next step, clustering algorithm used to groups features and generates an appropriate name for each feature. Final step, a product-by-feature matrix and a feature itemset graph (FIG) based on the relationships between products and the mined features are both constructed. Then kNN algorithm is used for recommending features .The trained recommender system can be used to generate recommendations based on an initial textual description of the product provided by the requirements or domain analyst.

Following is the list of component used in feature extraction and recommendation.

 ISSN: 2321-8169
1662 - 1667

[1] **Screen scraper:** It is first component of system used for mining descriptors from online product description.

[2] **Incremental diffusive clustering:** It is second component of recommendation system used for which clusters descriptors into features and extracts meaningful names for each cluster. This clustering algorithm is capable of identifying both most prominent and hidden themes from descriptors to perform well for the task of clustering requirements [13].

[3] **Product by-feature matrix:** which is generated as a by-product of the feature and products, as some work of recommendation used matrix factorization techniques [14].

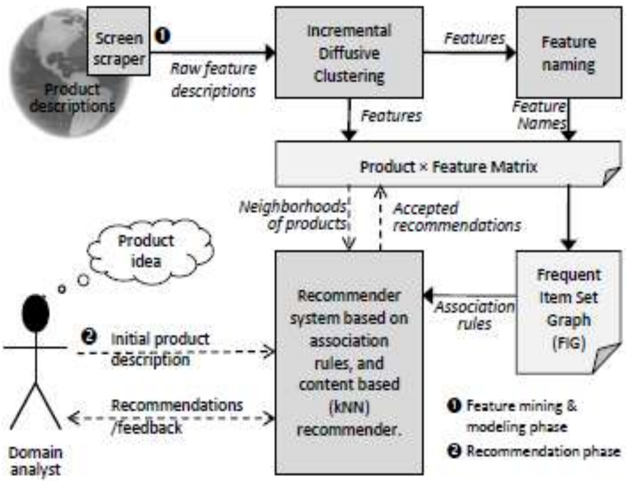

Figure 3. Feature extraction and recommendation

[4] **Frequent itemset graph (FIG):** From this feature association rule can generated [15, 16].

[5] **Recommender system:** The final component is the recommender system itself system utilizes both the Product-by-feature matrix and the FIG to generate on demand feature recommendations.

The feature recommendation process is start when a user provides a short textual product description. This description is automatically and analyzed tokenization, stemming, and removal of stop word to create a partial product profile. Then it work in flow as shown in figure 3[7] it produce recommendation of feature which is shown in figure 4[7]

V. Feature Recommendation Algorithm

Feature Recommendations can be achieved in two different ways. First, is the feature recommender which used as a Content Based recommender, it includes the standard kNN for feature recommendations. Second one is an association rule which is used to identify hidden connection between features and products Recommendations.

The aim of the feature recommendation system is to provide recommendations for software with a given set of initial features. The feature recommender can be trained by using a binary product-by-feature matrix, based upon the clusters created by IDC. Product-by-feature matrix it is a matrix that contain relationship between product and feature, the word binary is used here to indicate the relationship between product and feature i.e. 0,1. In binary product-by-feature matrix P gives the number of products (117,265), F is the number of identified features (1,135), m of i ,j is equal to 1 if the feature j includes a descriptor originally mined from the product i[1]. This matrix is referred as the feature pool for next process, it is further used to generate feature recommendations. By using this matrix various collaborative filtering algorithms, such as neighbourhood-based techniques which is user-based kNN and item-based kNN as well as matrix factorization approaches such as BPRMF [21], can be exploited to produce recommendations. For a new product p with a set of

Figure 4. Example of Feature Recommendation System

features Fp, the recommendation algorithm computes a recommendation score for each of the features which are not in Fp and presents to the users the top N (where N is the number of recommendations) features with the highest recommendation scores.

A. *FEATURE RECOMMENDATIONS USING STANDARD KNN*

For recommending feature different algorithms are used, such as product-based kNN, feature-based kNN method. K-Nearest Neighbor (kNN) learning strategy perform well in forum recommendations [38,41]. Two well-known methods are also used for recommendation against the Standard KNN i.e. feature-based kNN [45] and matrix factorization using Bayesian personalized ranking method (BPRMF) [46].

 ISSN: 2321-8169
1662 - 1667

Feature-based kNN method is also a neighbourhood model similar to product-based kNN. Feature-based kNN gives the prediction on the basis of feature neighbourhood, matrix factorization is another method which record both product and feature it is useful for binary data .By using kNN algorithm we can computes a feature-based similarity of a new product and all existing products. The highest priority k similar products are considered as neighbours of the new product. Then using the information from highest priority K neighbours recommends the features.

Content based feature recommendation have the some advantages i.e. Domain knowledge not Needed, Adaptive quality improves over time, Implicit feedback sufficient but it also have some limitations i.e. New user ramp-up problem, Quality dependent on large Historical data set. Stability verses plasticity problem.

Negar Hariri & Carlos Castro-Herrera are used cosine similarity to find the similarity of the new product and the existing product [11]. Negar Hariri et-al used equivalent of the cosine similarity to find similarity of new and existing products features [7]. Informally, the similarity is a numerical measure of the degree to which the two objects are alike. It is usually non-negative and are often between 0 and 1, where 0 means no similarity, and 1 means complete similarity [40]. There are different formulas for calculating similarities between the new product, existing product such as Euclidean Distance [42], Pearson Correlation [43], Jaccard coefficient [44], Cosine similarity.

B. *FEATURE RECOMMENDATIONS USING ASSOCIATION RULES*

Association rules can sometimes be used to generate recommendations from relatively scattered product profiles, which can be use if a user provides limited information for an initial product description. Various Association rule techniques are available such as the Apriori algorithm [27] ,TREE-PROJECTION [33], fp-growth [30], PERFECT HASHING AND PRUNING (PHP) [31], by using association rules we may identify set of items based on concurrence of item in transactional database. In feature recommendation system association rule mining used for finding the relationship between features with in products. Association rule mining developed for supporting market basket analysis to analyze products that customers tend to purchase at the same time.

For example, Association rule milk, butter => bread the meaning of this rule is that the customers who buy milk and butter are also likely to buy bread. For build recommender system in different domain association rule mining has been used such as in e-commerce, intelligent we applications [21], [32], [29]. Association rule has two Parts, first finding frequent feature sets and then discovery of frequent feature sets from association rule which satisfy a minimum confidence. H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli used a Apriori algorithm in their work for feature recommendation [7], apriori uses generate and test approach mean it generates candidate itemset and tests if they are frequent than select. it scan database multiple time . Then Negar Hariri et-al proposed fp_growth algorithm [35] in association rule mining. Fp_growth algorithm firstly generate fp_tree in that root node is null and then generating the nodes that contain features as well as support count of that feature. Fp_tree depends on how the items are ordered.

After generating association rule features can be recommended for new products when provide short textual description by finding all the matching rules. Frequent features can be sorted in directed acyclic graph [36], [37] for reducing search time. The graph is organized from level 0 to level N, where N is the maximum size from all frequent discovered frequent feature set.

Figure 5[7] depict a subset of graph for features related to antivirus software. Each node represents a frequent feature set and the number associated with it shows the support value of the feature set. For example, the 4 features , network intrusion detection (NP) and disk scan for finding malware (DS), Spyware Protection (SP), DF occur together in 0.935 of total products and are consider as a frequent feature set, if this subset graph is only considered then the feature recommended by the system are NP,DS,SP,DF .

CONCLUSION

Domain Analysis is process of identifying and documenting the commonalities and variables to a particular domain. A recommender system is designed to reduce the

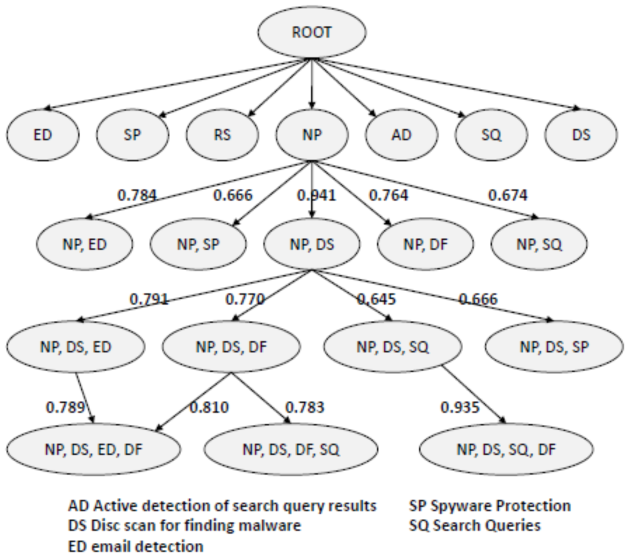

Figure 5. A Subset of Frequent Itemset Graph form Antivirus Software Features

human effort of performing domain analysis. Feature recommendation system is beneficial in single application projects. Domain analysis techniques are frequently

ISSN: 2321-8169
1662 - 1667

available to organizations with existing products in the targeted domain. The aim of the feature recommendation system is to provide recommendations for software with a given set of initial features. The feature recommender can be trained by using a binary product-by-feature matrix, Based upon the clusters created using IDC. Product-by-feature matrix its name sagest it is a matrix that contain relationship between product and feature, the word binary is Used here to indicate the relationship between product and feature.

On the basis of previously studied paper we conclude that when initial product description profile is sparse and have only few features then Association rule mining can be used which gives relatively accurate recommendations. Binary KNN Algorithm is much better than other algorithm, by using these algorithms we can find similar feature between new product and existing products by using some similarities formulas. Matrix factorization is also good techniques which consider feature as well as product but it is useful for binary data.

ACKNOWLEDGMENT

This review has been partially supported by Department of Computer Science and IT Dr. Babasaheb Ambedkar Marathwada University, Aurangabad. The views expressed here are those of the authors only.

REFERENCES

[1] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, “Feature-Oriented Domain Analysis (FODA) Feasibility Study,” Technical Report CMU/SEI-90-TR-021, Software Eng. Inst., 1990J.

[2] K.C. Kang, S. Kim, J. Lee, K. Kim, G.J. Kim, and E. Shin, “FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures,” Annals of Software Eng., vol. 5, pp. 143-168, 1998.

[3] W. Frakes, R. Prieto-Diaz, and C. Fox, “Dare: Domain Analysis and Reuse Environment,” Annals of Software Eng., vol. 5, pp. 125-141, 1998.

[4] K. Chen, W. Zhang, H. Zhao, and H. Mei. Anapproach to constructing feature models based onrequirements clustering. Requirements Engineering,IEEE International Conference on, 0:31–40, 2005.

[5] Burke. Knowledge-based recommender systems. Encyclopedia of Library and Information Systems, 69(32):180–200, 2000.

[6] P. Symeonidis et al., Recommender Systems for Location-based Social Networks, SpringerBriefs in Electrical and Computer Engineering, DOI 10.1007/978-1-4939-0286-6__2, © The Author(s) 2014.

[7] H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli, “On-Demand Feature Recommendations Derived from Mining Public Software Repositories,” Proc. 33rd Int’l Conf. Software Eng., p. 10, May 2011.

[8] “Recommender Systems” Prem Melville and Vikas Sindhwani IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.

[9] Resnick, P., and Varian, H.R. Recommender systems. Communications of the ACM, 40, 3 (1997), 56-58.

[10] Francesco Ricci and Hannes Werthner, Guest Editors *International Journal of Electronic Commerce*, Volume 11, Number 2, Winter 2006-07, pp. 5.

[11] Negar Hariri, Carlos Castro-Herrera, Member, IEEE, Mehdi Mirakhorli, Student Member, “Supporting Domain Analysis through Mining and Recommending Features from Online Product Listings” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 12, DECEMBER 2013.

[12] Negar Hariri, Carlos Castro-Herrera, Jane Cleland-Huang, and Bamshad Mobasher “Recommendation Systems in Requirements Discovery.”

[13] H.Dumitru, A.Czauderna, and J.Cleland-Huang. Automated mining of cross-cutting concerns from problem reports and requirements specifications. Argonne Undergraduate Symp., 2009.

[14] J. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” Computer, vol. 42, no. 8, pp. 30-37, Aug. 2009.

[15] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, “Effective Personalization Based on Association Rule Discovery from WebUsage Data,” Proc. Third Int’l Workshop Web Information and Data Management (WIDM ’01), Nov. 2001

[16] J. Sandvig, B. Mobasher, and R. Burke, “Robustness of Collaborative Recommendation Based on Association Rule Mining,” Proc. ACM Conf. Recommender Systems, 2007.

[17] M.P. Robillard et al. (eds.), *Recommendation Systems in Software Engineering*, DOI 10.1007/978-3-642-45135-5__17, © Springer-Verlag Berlin Heidelberg 2014.

[18] Manning, C.D., Raghavan, P, Schutze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008).

[19] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003).

[20] Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Mach. Learn. **42**(1–2), 143–175 (2001). doi:10.1023/A:1007612920971

[21] B.M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of Recommender Algorithms for E-Commerce,” Proc. Second ACM e-Commerce Conf. (EC ’00), pp. 158-167, Oct. 2000.

[22] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer, New York (2011). doi:10.1007/978-0-387-85820-3.

[23] Felfernig, A., Schubert, M., Mandl, M., Ricci, F., Maalej, W.: Recommendation and decision technologies for requirements engineering. In: Proceedings of the International Workshop on Recommendation Systems for Software Engineering, pp. 11–15 (2010) doi:10.1145/1808920.1808923.

[24] Maalej, W., Thurimella, A.: Towards a research agenda for recommendation systems in requirements engineering. In: Proceedings of the International Workshop on Managing Requirements Knowledge, pp. 32–39 (2009). oi:10.1109/MARK.2009.12

[25] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,”Proc. 20th Int’l Conf. Very Large Data Bases, 1994.

[26] J. Coplien, D. Hoffman, and D. Weiss, “Commonality and Variability in Software Engineering,”IEEE Software, vol. 15, no. 6, pp. 37-45, Nov./Dec. 1998

[27] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th Intn’l Conf. on Very Large Data Bases

(VLDB’94), pages 487–499, Santiago, Chile, September 1994.

[28] M. Simos, D. Creps, C. Klingler, L. Levine, and D. Allemang, “Software Technology for Adaptable Reliable Systems (STARS) Organization Domain Modeling (ODM) Guidebook Version 2.0,”

 ISSN: 2321-8169
1662 - 1667

Technical report STARS-VC- A025/001/00 , Lockheed Martin Tactical Defense Systems, 1996.

[29] W. Lin, S.A. Alvarez, and C. Ruiz, "Efficient Adaptive-Support Association Rule Mining for Recommender Systems," Data Mining and Knowledge Discovery, vol. 6, pp. 83-105, 2002.

[30] Jiawei Han, Jian Pei, Yiwen Yin" Mining Frequent Patterns without Candidate Generation" Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.

[31] S.ayse ozel ,H.Altay Guvenir "An Algorithm for Mining Association Rules Using Perfect Hashing and Database Pruning"

Bilkent University, Department of Computer Engineering,Ankara, Turkey.

[32] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective Personalization Based on Association Rule Discovery from Web Usage Data," Proc. Third Int'l Workshop Web Information and Data Management (WIDM '01), Nov. 2001.

[33] Ramesh C. Agarwal, Charu C. Aggarwal, V.V.V. Prasad "A tree projection algorithm for generation of frequent itemsets" IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

[34] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009).

[35] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00), June 2000.

[36] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective Personalization Based on Association Rule Discovery from Web Usage Data," Proc. Third Int'l Workshop Web Information and Data Management (WIDM '01), Nov. 2001.

[37] J. Sandvig, B. Mobasher, and R. Burke, "Robustness of Collaborative Recommendation Based on Association Rule Mining," Proc. ACM Conf. Recommender Systems, 2007.

[38] C. Castro-Herrera, J. Cleland-Huang, and B. Mobasher. Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. In Requirements Eng., IEEE Intn'l Conf. on, pages 37–46, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[39] E. Spertus, M. Sahami, and O. Buyukkokten.Evaluating similarity measures: a large-scale study in the orkut social network. pages 678–684, Chicago, Illinois, USA, 2005. ACM.

[40] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Published by Addison Wesley.

[41] C. Castro-Herrera, C. Duan, J. Cleland-Huang, and B. Mobasher. A recommender system for requirements elicitation in large-scale software projects. Proc. of the 2009 ACM Symp on Applied Computing, pages 1419–1426, 2009.

[42] http://en.wikipedia.org/wiki/Euclidean_distance

[43] Programming Collective Intelligence, Toby Segaran, First Edition, Published by O' Reilly Media, Inc.

[44] http://en.wikipedia.org/wiki/Jaccard_index

[45] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms,"Proc. 10th Int'l Conf. World Wide Web,pp. 285-295, 2001

[46] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-thieme, "L.S.: Bpr: Bayesian Personalized Ranking from Implicit Feed back,"Proc. 25th Conf. Uncertainty in Artificial Intelligence (UAI), 2009

.