A Survey on Feature Based Product Recommendation System

Akshay Giri[1], K. N. Honwadkar[2]

[1]University of Pune, Smt. Kashibai Navale College of Engineering, Vadgaon(Bk), Pune 411041, India

[2]University of Pune, Smt. Kashibai Navale College of Engineering, Vadgaon(Bk), Pune 411041, India

Abstract: ***The Recommender system is an essential part of the data and the e-commerce system. The system provides a highly powerful method for allowing users to filter out large set of information and the product places. The researchers from the same field are still working on how the recommender methodology is implemented in a particular domain. The software product engineering uses Feature Models (FMs) technology in a broad manner for helping and validating each product structure and it also provides support for the domain analysis. The creation of FM is tedious and very time-requiring technology. Now a day, it's popular to buy a product online and online product shopping is a new trade in E-Commerce domain. The customers can choose their preferred products and buy many products online from the Online shopping website or portal. Online shopping websites combine Business-to-Consumers, Business-to-business as well as Consumers-to-Consumers services with each other. Because of the customized, high feature and cost capability products the universality of the software is increases. Model-Driven Software Product Lines (MDSPL) permits the creators to construct a product based on the configuration information and the reusable resources automatically. A product line is constructed from a public set of resources.***

Keywords: Model-Driven software product lines, Feature models, E-commerce, Business-to-Business, Business-to-Consumers and Consumers-to-Consumers, Domain Engineering, Application Engineering.

1. Introduction

The domain analysis is a technique of recognizing, identifying, establishing, exploring, and modeling structures which are common for a specific domain [1], [2]. The input as basic product description is given to the proposed system; the system recognizes this given description and creates a correlated feature recommendation using two recommendation algorithms.

To learn similarity features of a product and to increase and improve the basic profile of the product, the unsupervised association rule mining methodology uses by our first algorithm[5], [6]. The second algorithm is used for recognizing the area of parallel products for finding a new features, it uses k-nearest neighbor technique and also improved the profile of the product [7]. The output of a system is the recommended features of a product, which is subpart of the requirement engineering methodology. This methodology helps project shareholders to define the features of a particular product. The following Figure 1.describe a feature recommendation process of an antivirus product.

A product description is plotted into four different features as:
1) Spam detection
2) Disk scans
3) Virus definitions and
4) Virus databases

This primary profile is then improved for file monitoring, web history, network intrusion and for cookies management. For providing feedback, the user communicates with a system. The file monitoring and history of a web are added to the basic profile of a product.

2. Feature Extraction

The detail description of a product is provided by lots of websites like Google Apps, Softpedia, Softonic and MarketPlace etc. For detail learning of a feature extraction, we had taken an example of Softpedia.com, which having a catalogs of thousands of products. The product in this site mainly contains a summary of product and the detail description of a product. For retrieving the summary and detail description of a product screen-scraper technique is used. The summary information was divided into the sentences. The feature descriptor added these divided sentences for creating the bullet list.

3. Feature Recommendation

We implemented a binary product-by-feature matrix which is created using the IDC clusters. The binary product-by-feature matrix is represented as,

$$M := (mi, j)P*F$$

Where,
P indicates the products quantity,
F indicates the identified features,
and m(i, j)is have the value 1 if the feature which indicates as j having a descriptor initially extracted from the product which indicates as i. The matrix includes the comprehensive set of recommended features which then referred as a feature pool.

4. Related Work

The literature review links a gap between recommender system and automatic feature detection. Hence this literature review provides a detail survey of recommender system and automatic feature detection. For examining, documenting and detecting the unities and inconsistency inside a domain numerous domain analysis approaches are created such as feature-oriented reuse the method (FROM)[2], feature-oriented domain analysis (FODA)[2], feature RSEB, organization domain modeling (ODM) [4] and family-oriented abstraction, specification, and translation (FAST) [5]. These above approaches are depends on human-intensive events. These approaches are suitable for a small scale domain.

For managing the inconsistency within many products the tools are created which maintained these approaches [5], [6], [7], [4]. Hence, the important efforts are needed for performing the domain study and recognize features. The recent research has been considered for the uses of data mining and machine learning models, for creating a domain model via establishing and determining the relations among the system software requirements.

The text processing technique retrieved a domain words from the text and next to that detect basic domain attributes, objects and functions this process is used by the domain analysis and reuse environment.

These clusters are then utilized by the expert to make a table. Chen et al. [8] manually built necessities relationship charts (RRG) from a few distinctive necessities details, and afterward utilized various leveled clustering approaches to union them into an individual domain. This methodology is not effectively measured the fact that the client is needed to manually make every necessities relationship chart. Although the fact that, these charts are developed from one application and can be reused in different applications within the same area, extensive client mediation is obliged to adjust and receive them. Alves et al. [9] used the vector space model (VSM) and inert semantic investigation (LSA) to focus the similitude in the middle of prerequisites, and afterward produced an affiliation lattice which is grouped utilizing the agglomerative progressive bunching (AHC) calculation.

Recognized clusters are then consolidated together utilizing a profundity first calculation to analyze hubs and make a peculiarity model for the space. Noppen et al. [q5] broadened this work by incorporating fluffy sets in the skeleton to permit singular necessities to be connected with numerous characteristics. Niu and Easterbrook [14], created an on demand bunching structure that gave self-loader help for investigating practical necessities in an item line. They proposed a data recovery (IR) and common dialect preparing (NLP) methodology to recognize paramount substances and capacities in the prerequisites as a useful prerequisites profile (FRP). FRPs re distinguished as indicated by the phonetic characterization of an area and have a tendency to catch the area's activity topics from the prerequisites reports. FRPs are utilized as bunching articles and data recovery systems are utilized to recognize (model) the variability among FRPs. A few different scientists have utilized affiliation data mining to mechanize the development of area gimmick models from a set of officially existing application characteristics.

They utilized a name-based fusing strategy to unite the covering parts of each feature model, include the variability data, and assemble an individual feature model speaking to the family of product. In any case, these methodologies on peculiarity setup expect that the rundown of features for every person application is accessible for blend and developing the last area model. The field of recommender frameworks has likewise been examined broadly, yet generally inside the connection of an e-commerce framework, where various calculations have been created to model the client proclivity and make expectations. These calculations differ significantly, contingent upon the sort of information they use as a data to make the suggestions. Some use content data about the things, or collaborative information of other clients' evaluations [7], or learning principles of the domain, or hybrid methodologies. Considerable measures of work have been carried out in the range of assessing recommender frameworks and on more current exceedingly productive algorithm, for example, those focused around grid factorization. Notwithstanding the multiplication of both of these fields, there has been next to no work consolidating recommender frameworks and prerequisites designing. Work in this range has concentrated on suggesting themes of enthusiasm toward huge scale online prerequisites discussions, and a high-level diagram of conceivable uses and applications of recommender frameworks in this area [5]. All things considered, our utilization of recommender frameworks in this paper expands on the considerable foundation of examination around there.

5. Synthesizing FMs

A few strategies for Synthesizing a FM from a set of designs or demands (e.g., encoded as a propositional recipe) have been proposed [19], [6], [1]. These strategies can't be connected in our connection, since we can't accept the accessibility of formal what's more finish portrayals of setups or stipulations. Consequently we create new extraction and Synthesizing systems to manage the casual and printed nature of item description. An essential constraint of earlier work is the recognizable proof of the peculiarity chain of importance. In [19], [6], the researchers ascertain a diagrammatic representation of all conceivable FMs. However they didn't address the issue of selecting an extraordinary FM with a compelling progressive system. Additionally, the calculation proposed in does not control the way the domain progressive system is incorporated in the ensuing FM. However the heuristics are particular to the focused on frameworks (Linux, Freebsd, ecos both from the area of working framework) and hence scarcely apply to our case. Besides She et al. reported that their endeavors to utilize clustering methodology did not create a solitary and attractive progressive system. They gave a conceivable reason contending that "there is just insufficient data in the information depictions what's more dependency" for the sorts of ancient artefacts they considered.

Acher et al. [1] proposed a method that procedures client pointed out learning for arranging the tree order of features. Our methodology [23], [1], does not require a client involvement. An alternate key distinction is that we coordinate feature retrieving and clustering strategies for building the suggestion diagram. Extraction of FMs. Acher et al. [2] proposed a semi-automated methodology to help the move from product explanation (communicated in an eve configuration) to FMs. Ryssel et al. created systems focused around Formal Concept Analysis having similar occurrence containing equivalent relations. A typical presumption is the accessibility of formal and complete details of product, which is most certainly not the case in our setting. The two works additionally accept a certain structure in the product details or other information that is enhancing to the tree based managed structure.

6. Conclusion

This paper has displayed a new feature recommender framework to backing the domain analysis process. This is a discriminating early-stage some piece of the product improvement lifecycle also is fundamental in both application improvement and product offering improvement. Our framework extract the feature for several products from freely accessible storehouses of product details and utilizes this information to find characteristics and their affiliations. For feature disclosure, we proposed a novel incremental diffusive clustering calculation. The assessment results introduced in this paper uncover that our clustering strategy is more suitable for this assignment than some other well-known clustering strategies. We expect that the results from our clustering examination can be connected to other clustering issues in the necessities engineering domain. An alternate novel commitment of this work is the mixture approach which utilizes affiliation standard mining to increase a starting profile, and after that uses the standard kNN methodology to make extra proposals. This has the point of interest of enlarging an at first parse product details in the recent past making a more far reaching set of proposals. The execution of the recommender framework was assessed both through quantitative examinations and subjective investigation. The results demonstrated that, our recommender framework is equipped for making suitable feature suggestions to help area examiners assemble space models for a mixture of programming applications. Moreover, the illustrations demonstrated all through the paper have represented methodologies, its advantages and its limitations.

In this paper we have displayed a novel calculation for automating the era of a FM from a set of casual furthermore inadequate product details. The results from the reported assessment demonstrate, that on account of the antivirus programming, using clustering systems to increase association rule mining prompted stamped upgrades in the quality of the created FM. As being what is indicated, the discoveries reported in this paper make a noteworthy commitment to the progressing research objective to mechanize the era of FMs.

References

[1] G. Arango and R. Prieto-Diaz, Domain Analysis: Acquisition of Reusable Information for Software Construction. IEEE CS Press, May 1989.
[2] 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., 1990.
[3] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases,1994.
[4] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases,1994.
[5] R. Agrwal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases (VLDB ’94), pp. 487-499, Sept. 1994.
[6] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” The Adaptive Web, pp. 291-324, Springer, 2007.
[7] 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.
[8] J. Sandvig, B. Mobasher, and R. Burke, “Robustness of Collaborative Recommendation Based on Association Rule Mining,” Proc.ACM Conf. Recommender Systems, 2007.
[9] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” ACM Trans.Information Systems, vol. 22, pp. 5-23, 2004.
[10] D. Weiss and C. Lai. *Software product-line engineering: a family-based software development process*. Addison-Wesley, 1999.
[11] S. Apel and C. K¨astner. An overview of feature-oriented software development. *Journal of Object Technology (JOT)*, 8(5):49–84, July/August 2009.
[12] D. Benavides, S. Segura, and A. R. Cort´es. Automated analysis of feature models 20 years later: A literature review. *Inf. Syst.*, 35(6):615–636, 2010.
[13] T. Berger, R. Rublack, D. Nair, J. M. Atlee, M. Becker, K. Czarnecki, and A. Wą̈sowski. A survey of variability modeling in industrial practice. In *Proceedings of VaMoS’13*. ACM, 2013.
[14] P. Borba, L. Teixeira, and R. Gheyi. A theory of software product line refinement. *Theor. Comput. Sci.*,455:2–30, 2012.
[15] K. Chen, W. Zhang, H. Zhao, and H. Mei. An approach to constructing feature models based on requirements clustering. In *Proceedings of RE’05*, pages 31–40, 2005.
[16] A. Classen, Q. Boucher, and P. Heymans. A text-based approach to feature modelling: Syntax and semantics of tvl. *Sci. Comput. Program.*, 76(12):1130–1143, 2011.
[17] A. Classen, P. Heymans, and P.-Y. Schobbens. What’s in a feature: A requirements engineering perspective. In *Proceedings of FASE’08*, volume 4961 of *LNCS*, pages 16–30, 2008.
[18] P. C. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley, 2001.
[19] K. Czarnecki, P. Gr¨unbacher, R. Rabiser, K. Schmid, and A. Wą̈sowski. Cool features and tough decisions: a comparison of variability modeling approaches. In *Proceedings of VaMoS’12*, pages 173–182, New York, NY, USA, 2012. ACM.

[20] K. Czarnecki, S. She, and A. Wasowski. Sample Spaces and Feature Models: There and Back Again. In *Proceedings of SPLC'08*, pages 22–31. IEEE, 2008.
[21] K. Czarnecki and A. Wasowski. Feature Diagrams and Logics: There and Back Again. In *Proceedings of SPLC'07*, pages 23–34. IEEE, 2007.
[22] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Journal of Machine Learning*, 42:143–175, january 2001.
[23] D. Dhungana, P. Gr¨unbacher, and R. Rabiser. The dopler meta-tool for decision-oriented variability modeling: a multiple case study. *Autom. Softw. Eng.*,18(1):77–114, 2011.