Mining Competitors from Large Unstructured Datasets

Ashlesha Ghodpage[1], Shrutika Datir[2], Priti Nishad[3], Rutuja Gulane[4], Vishal Tiwari[5]
[1,2,3,4]*Student, Dept. of Information Technology, St. Vincent College of Engg. and Technology, Nagpur, India*
[5]*Professor, Dept. of Information Technology, St. Vincent College of Engg. and Technology, Nagpur, India*

***Abstract*: In any competitive business, success is based on the ability to make an item more appealing to customers than the competition. A number of questions arise in the context of this task: how do we formalize and quantify the competitiveness between two items? Who are the main competitors of a given item? What are the features of an item that most affect its competitiveness? Despite the impact and relevance of this problem to many domains, only a limited amount of work has been devoted toward an effective solution. In this project, we present a formal definition of the competitiveness between two items, based on the market segments that they can both cover. Our evaluation of competitiveness utilizes customer reviews, an abundant source of information that is available in a wide range of domains. We present efficient methods for evaluating competitiveness in large review datasets and address the natural problem of finding the top-k competitors of a given item. Finally, we evaluate the quality of our results and the scalability of our approach using multiple datasets from different domains. In this project, we propose C-Miner, an algorithm which uses a data mining technique called frequent sequence mining to discover block correlations in storage systems. C-Miner runs reasonably fast with feasible space requirement, indicating that it is a practical tool for dynamically inferring correlations in a storage system.**

***Keywords*: Datasets, C-Miner, Database**

1. Introduction

Nowadays, huge amounts of sequential information are stored in databases (e.g. stock market data, biological data and customer data). Discovering patterns in such databases is important in many domains, as it provides a better understanding of the data. For example, in international trade, one could be interested in discovering temporal relations between the appreciations of currencies to make trade decisions. Various methods have been proposed for mining patterns in sequential databases such as mining repetitive patterns, trends and sequential patterns. Among them, mining sequential patterns is probably the most popular set of techniques. The marketing and management community have focused on empirical methods for competitor identification as well as on methods for analyzing known competitors. Extant research on the former has focused on mining comparative expressions (e.g. “Item A is better than Item B”) from the Web or other textual sources. Even though such expressions can indeed be indicators of competitiveness, they are absent in many domains. For instance, consider the domain of vacation packages (e.g. flight-hotel-car combinations). In this case, item shave no assigned name by which they can be queried or compared with each other. Further, the frequency of textual comparative evidence can vary greatly across domains. For example, when comparing brand names at the firm level (e.g. “Google vs. Yahoo” or “Sony vs. Panasonic”), it is indeed likely that comparative patterns can be found by simply querying the web. However, it is easy to identify mainstream domains where such evidence is extremely scarce, such as shoes, jewellery, hotels, restaurants, and furniture. Motivated by these Shortcomings, we propose a new formalization of the competitiveness between two items, based on the market segments that they can both cover [1].

2. Literature survey

- This paper proposed a new online metrics for competitor relationship predicting. This is based on the content, firm links and website log to measure the presence of online isomorphism, here the Competitive isomorphism, which is a phenomenon of competing firms becoming similar as they mimic each other under common market services.
- Through different analysis they find that predictive models for competitor identification based on online metrics are largely superior to those using offline data. The technique is combined the online and offline metrics to boost the predictive performance. The system also performed the ranking process with the considerations of likelihood.
- In this paper, it is argued that data mining is an approach to assist companies in developing more effective strategies to meet the competitions in the market. Data warehousing is useful and accurate for assembling a business’ dispersed heterogeneous data and providing unified convenient information access technique.
- Data mining technology can be used to transform hidden knowledge into manifest knowledge. A competitor mining from web data system is extremely flexible. Therefore, one of the best competitive

strategies is the successful utilization of web data for timely decision support.

- Information extraction from web pages is an active research area. Researchers have been developing various solutions from all kinds of perspectives to provide the comparative report. Many web information extraction systems rely on human users to provide marked samples so that the data extraction rules could be learned.
- Because of the supervised learning process, semi-automatic systems usually have higher accuracy than fully automatic systems that have no human intervention. Semi-automatic methods are not suitable for large-scale web applications that need to extract data from thousands of web sites.
- Also web sites tend to change their web page formats frequently, which will make the previous generated extraction rules invalid, further limiting the usability of semi-automatic methods. That's why many more recent work focus on fully or nearly fully automatic solutions.
- In the paper, presented a formal definition of the competitiveness between two items. Authors used many domains and handled many shortcomings of previous works. In this paper, the author considered the position of the items in the multi-dimensional feature space, and the preferences and opinions of the users. However, the technique addressed many problems like finding the top-k competitors of a given item and handling structured data.
- Web information extraction can be at the record level or data unit level. The former treat each data record as a single data unit while the latter go one step further to extract detailed data units within the data records [10]. Record level extraction method generally involves identifying the data regions that contain all the records, and then partitioning the data regions into individual records. Structured data extraction from Web pages has been studied extensively. Early works on manually constructed wrappers were found difficult to maintain and be applied to different Websites, because they are very labour intensive.

3. Significance of this work

The significance of this project is to help the customer to view the products as their convenience. In this project customer can write their views and also can check reviews whether it's good or bad. A formal definition of the competitiveness between two items, based on their appeal to the various customer segments in their market. Our approach overcomes the reliance of previous work on scarce comparative evidence mined from text. A formal methodology for the identification of the different types of customers in a given market, as well as for the estimation of the percentage of customers that belong to each type.

4. System design

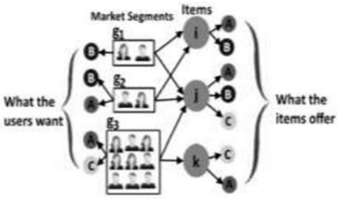

Fig. 1. An example of competitiveness paradigm

The figure illustrates the competitiveness between three items i, j and k. Each item is mapped to the set of features that it can offer to a customer. Three features are considered in this example: A, B and C. Even though this simple example considers only binary features (i.e. available/not available), our actual formalization accounts for a much richer space including binary, categorical and numerical features. The left side of the figure shows three groups of customer's g1, g2, and g3. Each group represents a different market segment. Users are grouped based on their preferences with respect to the features. For example, the customers in g2 are only interested in features A and B. We observe that items i and k are not competitive, since they simply do not appeal to the same groups of customers. On the other hand, j competes with both i (for groups g1 and g2) and k (for g3). Finally, an interesting observation is that j competes for 4 users with i and for 9 users with k. In other words, k is a stronger competitor for j, since it claims a much larger portion of its market share than i. This example illustrates the ideal scenario, in which we have access to the complete set of customers in a given market, as well as to specific market segments and their requirements. In practice, however, such information is not available. In order to overcome this, we describe a method for computing all the segments in a given market based on mining large review datasets. This method allows us to operationalize our definition of competitiveness and address the problem of finding the top-k competitors of an item in any given market. As we show in our work, this problem presents significant computational challenges, especially in the presence of large datasets with hundreds or thousands of items, such as those that are often found in mainstream domains. We address these challenges via a highly scalable framework for top-k computation, including an efficient evaluation algorithm and an appropriate index.

5. Implementation

In the implementation phase software development is concerned with translating design specifications into source code. The primary goal of implementation is to write the source code for internal documentation so that conformance of the code to its specification can be easily verified, and so that debugging, testing and modifications are erased. This goal is achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmarks of good programs. Obscurity, cleverness and complexity are indications of inadequate design and misdirected thinking. Source code clarity is enhanced by strutted techniques, good coding style, appropriate documents, go internal comments, and the features provided in the modern programming languages. The main aim of structured coding is adhere to single entry, single exit constructs in the majority of situations since it allows one to understand program behaviour by reading the code from beginning to end. Bust strict adherence to this construct may cause problems it raises concerns for the time and space efficiency of the code.

In some cases, single entry and single exit programs will require repeated code segments or repeated subroutines calls. In such cases, the usage of this construct would prevent premature loop exits and branching to exception handling code. So, in certain situations we violate this construct to acknowledge the realities of implementation although our intent is not encouraging poor coding style. In computer programming, coding style is manifest in the patterns used by programmers to express a desired action or outcome good coding style can overcome the deficiencies of primitive programming languages, while poor style can defeat the intent of an excellent language. The goal of good coding style is to provide easily understood straightforward, elegant code. Every good coding style performs the following Do's,

- Introduce user defined data types to model entities in the problem domain.
- Use a few standards agreed upon control statements.
- Hide data structures behind access functions.
- Use goto's in a disciplined way.
- Isolate machine dependencies in a few routines.
- Use indentation, parenthesis, blank lines and borders around comment blocks to enhance readability.
- Carefully examine the routines having fewer than 5 or more than 25 executable statements.

The following are the Don'ts of good coding style,

- Avoid null then statements
- Don't put nested loops very deeply.
- Carefully examine routines having more than five parameters.
- Don't use an identifier for multiple purposes.

Adherence implementation standards and guidelines by all programmers on a project results in a product of uniform quality. Standards were defined as those that can be checked by an automated tool. While determining adherence to a guideline requires human interpretation. A programming standard might specify items such as:

- The nested depth of the program constructs will not executed five levels.
- The goto statements will not be used.
- Subroutines length will not exceed 30 lines.

Implementation was performed with the following objectives,

- Minimize the memory required
- Maximize output readability and clarity
- Maximize source text readability
- Minimize the number of source statements
- Minimize the development time
- To ease the understanding of the source code
- To ease debugging
- To ease testing
- To ease documentation
- To ease modification of the program
- To facilitate formal verification of the program
- To put the tested system into operation while holding costs, risks and user irritation to minimum.
- Supporting documents for the implementation phase include all base-lined work products of the analysis and design phase.

6. Future work

The system will allow the users to view products of their choices and give the ratings to each of the product. More security measures can be added such as the website confidentiality, user identity and confidentiality. Product specific surveys can be done in future.

7. Conclusion

This paper presented an overview on mining competitors from large unstructured datasets.

References

[1] J. Krishna, P. Venkata Harsha Vardhan, P. Nirmala, N. Pavan Kumar - Mining Competitors from Large Unstructured Datasets, 2018.

[2] Valkanas, George, Theodoros Lappas, and Dimitrios Gunopulos - “Mining Competitors from Large Unstructured Datasets” IEEE Transactions on Knowledge and Data Engineering, 2017.

[3] Jin, Jian, Ping Ji, and Rui Gu, "Identifying comparative customer requirements from product online reviews for competitor analysis," Engineering Applications of Artificial Intelligence, 2016.

[4] Ghamisi, Pedram, Jon Atli Benediktsson, and Johannes R. Sveinsson, "Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction," IEEE Transactions on Geo science and Remote Sensing, 2016.

[5] Pant, Gautam, and Olivia RL Sheng - "Web footprints of firms: Using online isomorphism for competitor identification." Information Systems Research, 2015.

[6] Petrucci, Giulio "Information extraction for learning expressive ontologies." In European Semantic Web Conference, pp.740-750, Springer, Cham, 2015.

[7] Petrucci, Giulio. "Information extraction for learning expressive ontologies," in European Semantic Web Conference, pp. 740-750. Springer, Cham, 2015.

[8] Jin, Jian, Ping Ji, and Rui Gu, "Identifying comparative customer requirements from product online reviews for competitor analysis," Engineering Applications of Artificial Intelligence, 49, 2016.

[9] Valkanas, George, Theodoros Lappas, and Dimitrios Gunopulos, "Mining Competitors from Large Unstructured Datasets," IEEE Transactions on Knowledge and Data Engineering, 2017.

[10] G. Pant and O. R. Sheng, “Web footprints of firms: Using online isomorphism for competitor identification,” Information Systems Research, vol. 26, no. 1, pp. 188– 209, 2015.