Mining Competitors from Large Unstructured Datasets

J. Krishna[1], P. Venkata Harshavardhan[2], P. Nirmala[3], N. Pavankumar[4]

1. J. Krishna (Computer Science and Engineering,
Annamacharya Institute of Technology and Sciences, (Rajampeta), India)
2.P. Venkata Harshavardhan (Computer Science and Engineering,
Annamacharya Institute of Technology andSciences, (Rajampeta), India)
3.P. Nirmala (Computer Science and Engineering,
Annamacharya Institute of Technology and Sciences, (Rajampeta), India)
4. N. Pavankumar (Computer Science and Engineering,
Annamacharya Institute of Technology and Sciences, (Rajampeta), 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 paper, 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.

I. INTRODUCTION

A Long line of research has demonstrated the strategic importance of identifying and monitoring a firm's competitors . Motivated by this problem, 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, items have 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, jewelry, 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.

1.1 System architecture:

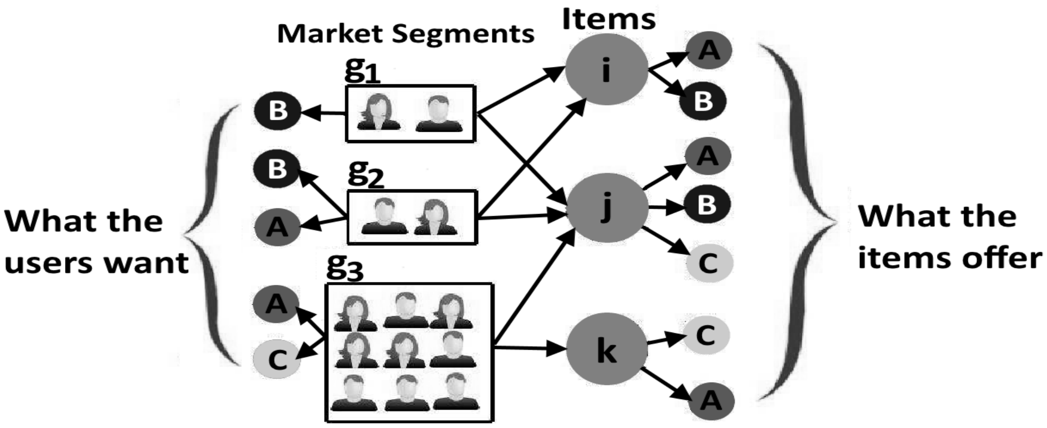

Fig. 1: A (simplified) example of our competitiveness paradigm

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 customers 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 *g*3). Finally, an interesting observation is that *j* competes for 4 users with *i* and for 9 users with *k*. In other words, *k* isa 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.

Our work makes the following contributions:

• 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.

• A highly scalable framework for finding the top-k competitors of a given item in very large datasets.

1.2 System modules

Administrator Module:

In this module, anadmin can upload details about items i.e. Camera, Hotels, Restaurants, and Recipes. After that, admin can check all uploaded items details, customer queries and interests. Finallytop-*k* competitors are identified from given item based on CMiner.

Customer Module:

In the Second module, we develop the Customer based features. In this module, the customer can give queries for anyone item, i.e. Camera, Hotels, Restaurants and recipes.At first creating the data set for cameras, Hotels, restaurant, recipes. Collect the Customer requirement from customer page.

CMiner Algorithm Module:

Next, we present CMiner, an exactalgorithm for finding the top-k competitors of a given item.Our algorithm makes use of the skyline pyramid in orderto reduce the number of items that need to be considered.Given that we only care about the top-k competitors, wecan incrementally compute the score of each candidate andstop when it is guaranteed that the top-k has emerged.

Skyline Operator Module:

In this module, skyline operator is performed. The skyline is a wellstudied concept that represents the subset of points in a population that are not dominated by any other point. We refer to the skyline of a set of items i as Sky(I).

The concept of the skyline leads to the following lemma:

Lemma1. Given the skyline Sky(I) of a set of items I and an item i ∈ I, let Y contain the k items from Sky(I) that are most competitive with i. Then, an item j ∈ I can only be in the top-k competitors of i, if j ∈ Y or if j is dominated by one of the items in Y.

II.EXISTING SYSTEM

- ❖ The management literature is rich with works that focus on how managers can manually identify competitors. Some of these works model competitor identification as a mental categorization process in which managers develop mental representations of competitors and use them to classify candidate firms. Other manual categorization methods are based on market- and resource-based similarities between a firm and candidate competitors.

- Zheng et al. identify key competitive measures (e.g. market share, share of wallet) and showed how a firm can infer the values of these measures for its competitors by mining (i) its own detailed customer transaction data and (ii) aggregate data for each competitor.

Disadvantages

- In this 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, jewelry, hotels, restaurants, and furniture.
- Existing approach is not appropriate for evaluating the competitiveness between any two items or firms in a given market. Instead, the authors assume that the set of competitors is given and, thus, their goal is to compute the value of the chosen measures for each competitor. In addition, the dependency on transactional data is a limitation we do not have.
- The applicability of such approaches is greatly limited

III.PROPOSED SYSTEM

- propose a new formalization of the competitiveness between two items, based on the market segments that they can both cover.
- 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.

Advantages

- To the best of our knowledge, our work is the first to address the evaluation of competitiveness via the analysis of large unstructured datasets, without the need for direct comparative evidence.
- 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.

❖ A highly scalable framework for finding the top-k competitors of a given item in very large datasets.

IV. RELATED WORK

This paper builds on and significantly extends our preliminary work on the evaluation of competitiveness . To the best of our knowledge, our work is the first to address the evaluation of competitiveness via the analysis of large unstructured datasets, without the need for direct comparative evidence. Nonetheless, our work has ties to previous work from various domains.

Managerial Competitor Identification: The management literature is rich with works that focus on how managers can manually identify competitors. Some of these works model competitor identification as a mental categorization process in which managers develop mental representations of competitors and use them to classify candidate firms. Other manual categorization methods are based on market- and resource-based similarities between a firm and candidate competitor.Finally, managerial competitoridentificationhasalsobeenpresentedasasensemaking process in which competitors are identified based on their potential to threaten an organizations identity.

Competitor Mining Algorithms: Zheng et al. Identify key competitive measures (e.g. market share, share of wallet) and showed how a firm can infer the values of these measures for its competitors by mining.

(i) Its own detailed customer transaction data and

(ii) Aggregate data for each competitor.

Contrary to our own methodology, this approach is not appropriate for evaluating the competitiveness between any two items or firms in a given market.

Instead, the authors assume that the set of competitors is given and, thus, their goal is to compute the value of the chosen measures for each competitor. In addition, the dependency on transactional data is a limitation we do not have. Doan et al. explore user visitation data, such as the geo-coded data from location-based social networks, as a potential resource for competitor mining. While they report promising results, the dependence on visitation data limits the set of domains that can benefit from this approach. Pant and Sheng hypothesize and verify that competing firms are likely to have similar web footprints, a phenomenon that they refer to as online isomorphism. Their study considers different types of isomorphism between two firms, such as the overlap between the in-links and out links of their respective websites, as well as the number of times that they appear together online (e.g. in search results or new articles). Similar to our own methodology, their approach is geared toward pairwise competitiveness. However, the need for isomorphism features limits its applicability to firms and make it unsuitable for items and domains where such features are either not available or extremely sparse, as is typically the case with co-occurrence data. In fact, the sparsity of co-occurrence data is a serious limitation of a significant body of work that focuses on mining competitors based on comparative expressions found in web results and other textual corpora. The intuition is that the frequency of expressions like “Item A is better than Item B” “or item A vs. Item B” is indicative of their competitiveness. However, as we have alreadydiscussed in the introduction, such evidence is typically scarce or even non-existent in many mainstream domains. As a result, the applicability of such approaches is greatly limited. We provide empirical evidence on the sparsity of co-occurrence information in our experimental evaluation.

Finding Competitive Products: Recent work has explored competitiveness in the context of product design. The first step in these approaches is the definition of a dominance function that represents the value of a product. The goal is then to use this function to create items that are not dominated by other, or maximize items with the maximum possible dominance value. A similar line of work represents items as points in a multidimensionalspaceandlooksforsubspaceswheretheappealofthe item is maximized. While relevant, the above projects have a completely different focus from our own, and hence the proposed approaches are not applicable in our setting. **Skyline computation:** Our work leverages concepts and techniques from the extensive literature on skyline computation. These include the dominance concept among items, as well as the construction of the skyline pyramid used by our CMiner algorithm. Our work also has ties to the recent publications in reverse skyline queries. Even though the focus of our work is different, we intend to utilize the advances in this field to improve our framework in future work.

V.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 behavior 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 standard, 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 lengths will not exceed 30 Lines.

Implementation was performed with the following objectives

- Minimize the memory required.
- Maximize output readability or 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.

VI.CONCLUSION

In this, presented a formal definition of competitiveness between two items, which we validated both quantitatively and qualitatively. Our formalization is applicable across domains, overcoming the shortcomings of previous approaches. We consider a number of factors that have been largely overlooked in the past, such as the position of the items in the multi-dimensional feature space and the preferences and opinions of the users. Our work introduces an end-to-end methodology for mining such information from large datasets of customer reviews. Based on our competitiveness definition, we addressed the computationally challenging problem of finding the top-k competitors of a given item. The proposed framework is efficient and applicable to domains with very large populations of items.

VII.SCOPE OF FUTURE ENCHANCEMENT

The efficiency of our methodology was verified via an experimental evaluation on real datasets from different domains. Our experiments also revealed that only a small number of reviews is sufficient to confidently estimate the different types of users in a given market, as well the number of users that belong to each type.

REFERENCES

[1] M.E.Porter,CompetitiveStrategy:TechniquesforAnalyzingIndustries and Competitors. Free Press, 1980. [2] R. Deshpand and H. Gatingon, “Competitive analysis,” Marketing Letters, 1994.

[3] B. H. Clark and D. B. Montgomery, “Managerial Identification of Competitors,” Journal of Marketing, 1999. [4] W. T. Few, “Managerial competitor identification: Integrating the categorization, economic and organizational identity perspectives,” Doctoral Dissertaion, 2007.

[5] M. Bergen and M. A. Peteraf, “Competitor identification and competitor analysis: a broad-based managerial approach,” Managerial and Decision Economics, 2002.

[6] J. F. Porac and H. Thomas, “Taxonomic mental models in competitor definition,” The Academy of Management Review, 2008.

[7] M.-J. Chen, “Competitor analysis and interfirm rivalry: Toward a theoretical integration,” Academy of Management Review, 1996.

[8] R. Li, S. Bao, J. Wang, Y. Yu, and Y. Cao, “Cominer: An effective algorithm for mining competitors from the web,” in ICDM, 2006.

[9] Z. Ma, G. Pant, and O. R. L. Sheng, “Mining competitor relationships from online news: A network-based approach,” Electronic Commerce Research and Applications, 2011.

[10] R. Li, S. Bao, J. Wang, Y. Liu, and Y. Yu, “Web scale competitor discovery using mutual information,” in ADMA, 2006.

[11] S.Bao,R.Li,Y.Yu,andY.Cao,“Competitorminingwiththeweb,” IEEE Trans. Knowl. Data Eng., 2008.

[12] G. Pant and O. R. L. Sheng, “Avoiding the blind spots: Competitor identification using web text and linkage structure,” in ICIS, 2009.

[13] D. Zelenko and O. Semin, “Automatic competitor identification from public information sources,” International Journal of Computational Intelligence and Applications, 2002.

[14] R. Decker and M. Trusov, “Estimating aggregate consumer preferences from online product reviews,” International Journal of Research in Marketing, vol. 27, no. 4, pp. 293–307, 2010.

[15] C. W.-K. Leung, S. C.-F. Chan, F.-L. Chung, and G. Ngai, “A probabilistic rating inference framework for mining user preferences from reviews,” World Wide Web, vol. 14, no. 2, pp. 187–215, 2011.

[16] K. Lerman, S. Blair-Goldensohn, and R. McDonald, “Sentiment summarization: evaluating and learning user preferences,” in ACL, 2009, pp. 514–522.

[17] E. Marrese-Taylor, J. D. Vel´asquez, F. Bravo-Marquez, and Y. Matsuo,“Identifyingcustomerpreferencesabouttourismproductsusing an aspect-based opinion mining approach,” Procedia Computer Science, vol. 22, pp. 182–191, 2013.

[18] C.-T. Ho, R. Agrawal, N. Megiddo, and R. Srikant, “Range queries in olap data cubes,” in SIGMOD, 1997, pp. 73–88.

[19] Y.-L.Wu,D.Agrawal,andA.ElAbbadi,“Usingwaveletdecomposition to support progressive and approximate range-sum queries over data cubes,” in CIKM, ser. CIKM ’00, 2000, pp. 414–421.

[20] D. Gunopulos, G. Kollios, V. J. Tsotras, and C. Domeniconi, “Approximating multi-dimensional aggregate range queries over real attributes,” in SIGMOD, 2000, pp. 463–474.

[21] M. Muralikrishna and D. J. DeWitt, “Equi-depth histograms for estimating selectivity factors for multi-dimensional queries,” in SIGMOD, 1988, pp. 28–36.

[22] N. Thaper, S. Guha, P. Indyk, and N. Koudas, “Dynamic multidimensional histograms,” in SIGMOD, 2002, pp. 428–439.

[23] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, “Parallel data processing with mapreduce: a survey,” AcM sIGMoD Record, vol. 40, no. 4, pp. 11–20, 2012.

[24] S.B¨orzs¨onyi,D.Kossmann,andK.Stocker,“Theskylineoperator,” in ICDE, 2001.

[25] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An optimal and progressive algorithm for skyline queries,” ser. SIGMOD ’03.

[26] G. Valkanas, A. N. Papadopoulos, and D. Gunopulos, “Skyline ranking `a la IR,” in ExploreDB, 2014, pp. 182–187.

[27] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson, “On the average number of maxima in a set of vectors and applications,” J. ACM, 1978.

[28] X. Ding, B. Liu, and P. S. Yu, “A holistic lexicon-based approach to opinion mining,” ser. WSDM ’08.

[29] A. Agresti, Analysis of ordinal categorical data. John Wiley & Sons, 2010, vol. 656.

[30] T.Lappas,G.Valkanas,andD.Gunopulos,“Efficientanddomaininvariant competitor mining,” in SIGKDD, 2012, pp. 408–416.

[31] J. F. Porac and H. Thomas, “Taxonomic mental models in competitor definition,” Academy of Management Review, vol. 15, no. 2, pp. 224–240, 1990.

[32] Z. Zheng, P. Fader, and B. Padmanabhan, "From business intelligence to competitive intelligence: Inferring competitive measures using augmented site-centric data," Information Systems Research, vol. 23, no. 3-part-1, pp. 698–720, 2012.

[33] T.-N. Doan, F. C. T. Chua, and E.-P. Lim, "Mining business competitiveness from user visitation data," in International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction. Springer, 2015, pp. 283–289.

[34] 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.

[35] K. Xu, S. S. Liao, J. Li, and Y. Song, "Mining comparative opinions from customer reviews for competitive intelligence," Decis. Support Syst., 2011.

[36] Q. Wan, R. C.-W. Wong, I. F. Ilyas, M. T. ¨Ozsu, and Y. Peng, "Creating competitive products," PVLDB, vol. 2, no. 1, pp. 898– 909, 2009.

[37] Q. Wan, R. C.-W. Wong, and Y. Peng, "Finding top-k profitable products," in ICDE, 2011.

[38] Z. Zhang, L. V. S. Lakshmanan, and A. K. H. Tung, "On domination game analysis for microeconomic data mining," ACM Trans. Knowl. Discov. Data, 2009.

[39] T. Wu, D. Xin, Q. Mei, and J. Han, "Promotion analysis in multidimensional space," PVLDB, 2009.

[40] T. Wu, Y. Sun, C. Li, and J. Han, "Region-based online promotion analysis," in EDBT, 2010.

[41] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: an online algorithm for skyline queries," ser. VLDB, 2002.

[42] A. Vlachou, C. Doulkeridis, Y. Kotidis, and K. Nørv˚ag, "Reverse top-k queries," in ICDE, 2010.

[43] A. Vlachou, C. Doulkeridis, K. Nørv˚ag, and Y. Kotidis, "Identifying the most influential data objects with reverse top-k queries," PVLDB, 2010.

[44] K.HoseandA.Vlachou,"Asurveyofskylineprocessinginhighly distributed environments," The VLDB Journal, vol. 21, no. 3, pp. 359–384, 2012.