International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

# An Efficient Evaluation of Top-K Competitors using CMiner++

Yandamuri Gayatri & R. Shiva Shankar
M. tech, Department of Computer Science and Technology, SRKR Engineering College, Bhimavaram, West Godavari, Andhra Pradesh, India.

Assistant Professor, Department of Computer Science and Engineering, SRKR Engineering College, Bhimavaram, West Godavari, Andhra Pradesh, India

Abstract – *Data mining is the popular area of the research which facilitates the business improvement process such as mining user preference, mining web information's to get opinion about the product or services and mining the competitors of a specific business. In the current competitive business scenario, there is a need to analyze the competitive features and factors of an item that most affect its competitiveness. The evaluation of competitiveness always uses the customer opinions in terms of reviews, ratings and abundant source of information's from the web and other sources. In this paper, a formal definition of the competitive mining is describes with its related works. Finally the paper provides the challenges and importance in the competitor mining tasks with optimal improvements.*

Index Terms – Cminer++, Information Search and Retrieval, Competitor Mining, Firm analysis, Electronic commerce.

## 1 Introduction

The strategic importance of detecting and observing business competitors is an inevitable research, which motivated by several business challenges. Monitoring and identifying firm's competitors have studied in the earlier work. Data mining is the optimal way of handling such huge information's for mining competitors. Item reviews form online offer rich information about customers' opinions[1] and interest to get a general idea regarding competitors. However, it is generally difficult to understand all reviews in different websites for competitive products and obtain insightful suggestions manually. In the earlier works in the literatures, many i analyzed such big customer data intelligently and efficiently. For example, a lot of studies about online reviews were stated to gather item opinion analysis from online reviews in different levels.

However, most researchers in this field ignore how to make their findings be seamlessly utilized to the competitor mining process[1]. Recently, a limited number of researches were noted to utilize the latest development in artificial intelligence (AI) and data mining in the e-commerce applications. These studies help designers to understand a large amount of customer requirements in online reviews for product improvements. But, these discussions are far from sufficient and some potential problems. These have not been fully investigated such as, with product online reviews, how to conduct a thorough competitor analysis. Actually, in a typical scenario of a customer-driven new product design (NPD), the strengths and weakness are often analyzed exhaustively for probable opportunities to succeed in the fierce market competition.

## 2 Literature Review

This research provides the various methodologies implemented to mine competitors with reference to customer lifetime value, relationship, [1]opinion and behavior using data mining techniques. The web growth has resulted in widespread usage of many applications like e-commerce and other service oriented applications. This varied usage of web applications has provided an enormous amount of data at one's disposal. Data is the input that exists in its raw form resulting in information for further processing. With huge amount

of data, organizations faced the crucial challenge of extracting very useful information from them. This has led to the concept of data mining. Mining[1] competitor's of a given item, the most influenced factor of the item which satisfies the customer need can be extracted from the data that is typically stored in the database. This section gives two types of literatures such as competitor mining and unstructured data management.

A. Unstructured data management:

The data collected from the web are sometimes semi-structured or unstructured. The semi-structured data's are in the format of XML, JSON etc., the unstructured data sources are in a different format, which is not fall under any predefined category. When managing thousands of customers, business will have difficulty sustaining the rising costs created by interactions among people. However, if all customer data is inserted into a database, the resulting records will provide a detailed profile of these customers and their interactions with one another, and will be an important resource for businesses that wish to probe customer data, customer needs, and customer satisfaction levels.

Data mining uses transaction data to gain a better understanding of customers and effectively discover hidden knowledge through the insertion of business intelligence into the process of competitor mining. I 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. Customer data for competitor mining is collected through several methods, which is usually unstructured; however, most data mining technologies can only handle structured data. Therefore, during competitor mining process, unstructured data is not taken into account and much valuable service information is lost. Structured systems are those where the data and the computing activity is predetermined and well-defined. Unstructured systems are those that have no predetermined form or structure and are usually full of textual data.

B. Competitor Mining:

The earlier work on the competitor mining utilized the text data to collect [3][5]comparative evidences between two items. But, the comparative[3][5] evidences are based on the assumptions, which may not always exist. Competitor identification is referred to as a classification[7] process through which competitors of a focal firm are identified based on "relevant similarities". I developed an [10][6] automatic system that discovers competing companies from public information sources. In this system data is crawled from text and it uses transformation oriented learning to obtain appropriate data normalization, combines structured and unstructured information sources, uses probabilistic modeling to represent models of linked data, and succeeds in autonomously discovering competitors. Bayesian network for competitor identification technique is used.

I also introduced the iterative graph reconstruction process for inference in relational data, and shown that it leads to improvements in performance. To find the competitors, I used machine learning algorithms and probabilistic approaches. They also validate system results and deploy it on the web as a powerful analytic tool for individual and institutional investors. However, the technique has many problems like finding alliances and market demands using the machine learning approach. In this paper, I presented a formal definition of the competitiveness between two items. I used many domains and handled many shortcomings of previous works. In this paper, I considered the position of the items in the multi-dimensional [2]feature space, and the preferences and opinions[1] of the users.

However, the technique addressed many problems like finding the top-k competitors of a given item and handling structured data. I 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. Several works in the same strategy in literature have discussed the need for accurate identification of competitors and provided theoretical frameworks for that. Given the expected isomorphism between competing firms, the process of competitor identification through pair-wise analysis of similarities between focal and target firms is well founded. The unit of analysis is a pair of firms since competitor relationship is seen as a unique interaction between the pair. We have suggested frameworks for manual identification of competitors. The manual nature of these frameworks makes them very costly for competitor identification over a large number of focal and target firms, and over time.

## 3 Implementation

a) System Architecture:-

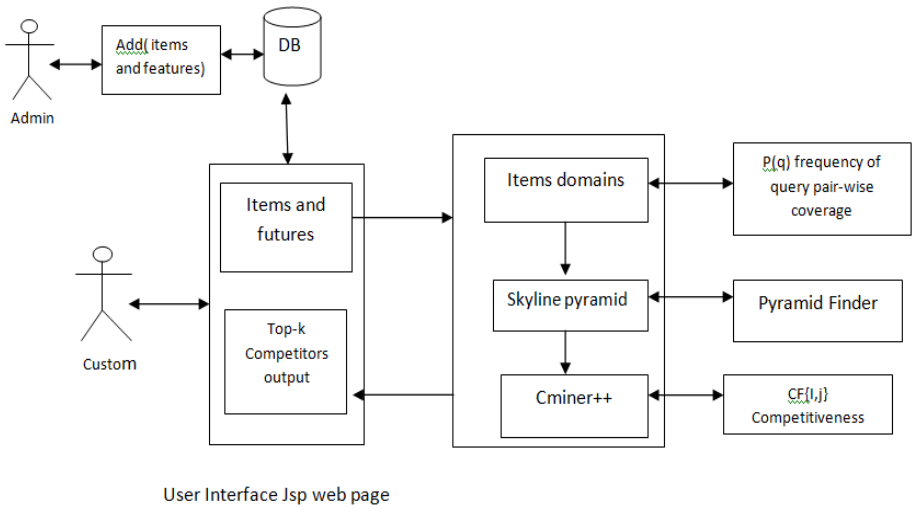


Fig:- System Architecture

An admin 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. Finally top-*k* competitors are identified from given item based on Cminer++. We develop the Customer based [2]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. We present Cminer++, an exact algorithm for finding the top-k competitors of a given item. Our algorithm makes use of the skyline pyramid in order to reduce the number of items that need to be considered. Given that we only care about the top-k competitors, we can incrementally compute the score of each candidate and stop when it is guaranteed that the top-k has emerged. We observe that the enhanced CMiner++ algorithm consistently outperformed all the other approaches, across datasets and values of k. The advantage of CMiner++ is increased for larger values of k, which allow the algorithm to benefit from its improved pruning.

Conceptually, an item dominates another if it has better or equal values across [2]features. We observe that, any item *i* that dominates *j* also achieves the maximum possible competitiveness with *j*, since it can cover the requirements of *any* customer covered by *j*. This motivates us to utilize the *skyline* of the entire set of items *I*. The skyline is a well studied 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*).

In this experiment, we evaluate this assumption on our four datasets. For every pair of items in each dataset, we report (1) the number of reviews that mention both items and (2) the number of reviews that include a direct comparison between the two items. We extract such comparative[3][5] evidence based on the union of "competitive evidence" lexicons used by previous work.

The figure motivates multiple relevant observations. First, we observe that the vast majority of the top-rank competitors proposed by our approach were verified as likely replacements for the seed item. These are thus verified as strong competitors that could deprive the seed item from potential customers and decrease its market share. On the other hand, the top-ranked candidates of NN were often rejected by the users, who did not consider these items to be competitive. Both approaches exhibited their worst results for the RECIPES dataset, even though the "YES" percentage of the top-ranked items by our method was almost twice that of NN. The difficulty of the recipes domain is intuitive, as users are less used to consider recipes in a competitive setting. The middle-ranked candidates of our approach attracted mixed responses from the annotators, indicating that it was not trivial to determine whether the item is indeed competitive or not. An interesting observation is that, for some of our datasets, the middle-ranked candidates of NN were more popular than its top-ranked ones, which implies that this approach fails to emulate the way the users perceive the competitiveness between two items. The bottom-ranked candidates of our approach were consistently rejected, verifying their lack of competitiveness to the seed item. The bottom-ranked items by the NN approach were also frequently rejected, indicating that it is easier to identify items that are not competitive to the target. Finally, to further illustrate the difference between our competitiveness model and the similarity-based approach, we conducted the following quantitative experiment. For each item i in a dataset, we retrieve its 300 top-ranked competitors, as ordered by each of the two methods. We then compute the Kendall τ and overlap of the two lists.

We report the average of these two quantities over all items in the dataset. The results demonstrate that the rankings of the two techniques are significantly different both in their ordering[9] and in the items that they contain. The second observation is that, for all datasets except recipes, Cminer++ achieves near-perfect results even for larger values of T. This is based on the observed values of the Kendall τ coefficient, which was consistently above 0.9 for all evaluated combinations of the k and T parameters. This is an encouraging finding, since it reveals a highly appealing and practical tradeoff between the computational efficiency and quality of Cminer++. In addition, it is important to note that the practice of reducing the size of number of considered queries does not require any modifications to the algorithm itself and can thus be applied with minimum effort. A careful examination of the recipes dataset reveals that the low correlation values can be attributed to the fact that most queries have a low frequency and, in fact, their frequency distribution is nearly uniform. As a result, even a low value for the T threshold eliminates a large number of queries and prevents Cminer++ from computing the exact solution to the top-k problem.

b) CMiner++ Algorithm :-

**Input:** Set of items I, Item of interest i ∈ I, feature space F, Collection Q ∈ 2F of queries with non-zero weights, skyline pyramid DI, int k.

**Output:** Set of top-k competitors for i

1: TopK ← masters(i)
2: if ( k ≤ |TopK| ) then
3: return TopK

4: end if
5: k ← k − |TopK|
6: LB ← −1
7: X ←GETSLAVES(TopK;DI) ∪ DI[0]
8: while ( |X | != 0 ) do
9: X GETSLAVES(X;DI)
10: end if
11: end while
12: return TopK
13: Routine UPDATETOPK(k, LB, X)
14: localTopK ← ∅
15: min(j) ← 0; ∀j ∈ X.
16: up(j) ←Σq2Qp(q) × V q j;j ; ∀j ∈ X.
17: for every q ∈ Q do
18: maxV ← p(q) × V qi;i
19: for every item j ∈ X do
20: up(j) ← up(j) − maxV + p(q) × V qi;j
21: if ( up(j) < LB ) then
22: X ← X \ {j}
23: else
24: min(j) ← min(j) + p(q) × V qi;j
25: localTopK:update(j; min(j))
26: if ( |localTopK| ≥ k ) then
27: LB ←WORSTIN(localTopK)
28: end if
29: end if
30: end for
31: if (|X | ≤ k ) then
32: break
33: end if
34: end for
35: for every item j ∈ X do
36: for every remaining q ∈ Q do
37: min (j) ← min (j) + p(q) × V q i;j
38: end for
39: localTopK:update(j; min (j))
40: end for
41: return TOPK(localTopK)

The Cminer++ Algorithm: Next, we present Cminer++, an exact algorithm for finding the top-k competitors of a given item. Our algorithm makes use of the skyline pyramid in order to reduce the number of items that need to be considered. Given that we only care about the top-k competitors, we can incrementally compute the score of each candidate and stop when it is guaranteed that the top-k have emerged.

Discussion of Cminer++: The input includes the set of items I, the set of features F, the item of interest i, the number k of top competitors to retrieve, the set Q of queries and their probabilities, and the skyline pyramid DI. The algorithm first retrieves the items that dominate i, via masters(i) (line 1). These items have the maximum possible competitiveness with i. If at least k such items exist, we report those and conclude (lines 2-4). Otherwise, we add them to Top-K and decrement our budget of k accordingly (line 5). The variable LB maintains the lowest lower bound from the current top-k set (line 6) and is used to prune candidates. In line 7, we initialize the set of candidates X as the union of items in the first layer of the pyramid and the set of items dominated by those already in the Top-K. This is achieved via calling GETSLAVES(Top-K,DI). In every iteration of lines 8-17, Cminer++ feeds the set of candidates X to the UPDATETOPK() routine, which prunes items based on the LB threshold. It then updates the Top-K set via the MERGE() function, which identifies the items with the highest competitiveness from Top-K [ X]. This can be achieved in linear time, since both X and Top-K are sorted. In line 13, the pruning threshold LB is set to the worst (lowest) score among the new Top-K. Finally, GETSLAVES() is used to expand the set of candidates by including items that are dominated by those in X.

Discussion of UPDATETOPK(): This routine processes the candidates in X and finds at most k candidates with the highest competitiveness with i. The routine utilizes a data structure local Top-K, implemented as an associative array: the score of each candidate serves as the key, while its id serves as the value. The array is key-sorted, to facilitate the computation of the k best items. The structure is automatically[10][6] truncated so that it always contains at most k items. In lines 21-22 we initialize the lower and upper bounds. For every item j 2 X, low(j) maintains the current competitiveness score of j as new queries are considered, and serves as a lower bound to the candidate's actual score. Each lower

bound low(j) starts from 0, and after the completion of UPDATETOPK(), it includes the true competitiveness score CF(i, j) of candidate j with the focal item i. On the other hand, up(j) is an optimistic upper bound on j's competitiveness score. Initially, up(j) is set to the maximum possible score (line 22). This is equal to Σ q∈Q p(q) _ V qi,i, where V q i,i is simply the coverage provided exclusively by i to q. It is then incrementally reduced toward the true CF(i, j) value as follows. For every query q 2 Q, maxV

4 Result analysis

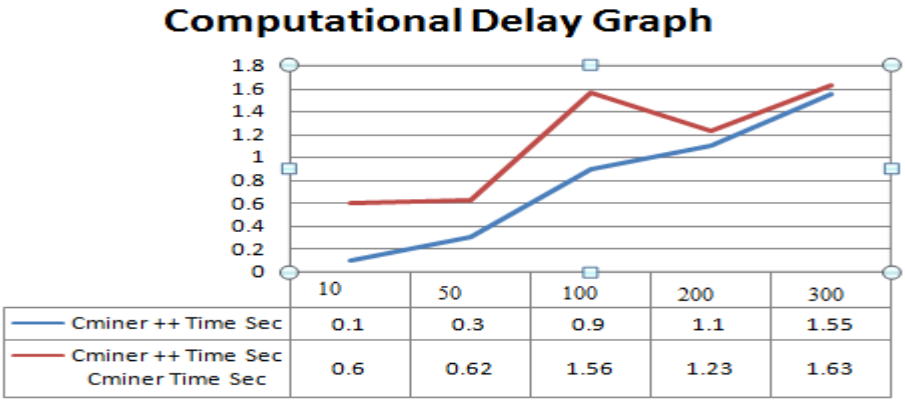


Fig:- Computational Delay Graph

| no. of Competitors | Cminer ++ Time Sec | Cminer Time Sec |
|---|---|---|
| 10 | 0.1 | 0.6 |
| 50 | 0.3 | 0.62 |
| 100 | 0.9 | 1.56 |
| 200 | 1.2 | 1.23 |
| 300 | 1.6 | 1.63 |

Fig:- Computational Delay analysis

The existing competitor mining algorithms such as CMiner and Cminer++ has been evaluated and compared with the time complexity. The below graph shows the computational time taken for the individual algorithm is plotted. Below graph Computational efficiency analysis chart.

5 Conclusion

Data mining has importance regarding finding the patterns, forecasting, discovery of knowledge etc., in different business domains. Machine learning algorithms are widely used in various applications. Every business related application uses data mining techniques. To improve such business or providing appropriate competitors for the business to the user need the support of web mining techniques. The competitor mining is one such a way to analyze competitors for the selected items. In this paper, we gave a comprehensive analysis of the competitor mining algorithms with its advantages and drawbacks. Finally, the CMiner++ yielded least computation time when comparing others. The most important [2]features and process

are not considered in the all baseline algorithms. This can be improved in the further researches.

## 6 References

[1] Ding, X., Liu, B., Yu, P.S., 2008. A holistic lexicon-based approach to opinion mining. In: Proceedings of the WSDM'08.

[2] Abbasi, A., Chen, H., Salem, A., 2008. Sentiment analysis in multiple languages: feature selection for opinion classification in web forums. ACM Trans. Inf. Syst. 26 (3), 12:1–12:34

[3] Chen, L., Qi, L., Wang, F., 2012. Comparison of feature-level learning methods for mining online consumer reviews. Expert Syst. Appl. 39 (10), 9588–9601.

[4] Zhan, J., Loh, H.T., Liu, Y., 2009. Gather customer concerns from online product reviews – a text summarization approach. Expert Syst. Appl. 36 (2 Part 1), 2107–2115

[5] 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): 61-73.

[6] Saxena, Prateek, David Molnar, and Benjamin Livshits. "SCRIPTGARD: automatic context-sensitive sanitization for largescale legacy web applications." Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011.

[7] 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 Geoscience and Remote Sensing 52.9 (2014): 5771-5782.

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

[9] Gentile, Anna Lisa, Ziqi Zhang, Isabelle Augenstein, and Fabio Ciravegna. "Unsupervised wrapper induction using linked data." In Proceedings of the seventh international conference on Knowledge capture, pp. 41-48. ACM, 2013.

[10] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," Proc. 14th ACM Int'l Conf. Information and Knowledge Management, pp. 381-388, 2005.