A Microeconomic Data Mining Problem: Customer-Oriented Catalog Segmentation

Martin Ester Rong Ge Wen Jin Zengjian Hu
School of Computing Science
Simon Fraser University
Burnaby, B.C.,Canada V5A 1S6
Email: {ester, rge, wjin, zhu}@cs.sfu.ca

ABSTRACT

The microeconomic framework for data mining [7] assumes that an enterprise chooses a decision maximizing the overall utility over all customers where the contribution of a customer is a function of the data available on that customer. In Catalog Segmentation, the enterprise wants to design k product catalogs of size r that maximize the overall number of catalog products purchased. However, there are many applications where a customer, once attracted to an enterprise, would purchase more products beyond the ones contained in the catalog. Therefore, in this paper, we investigate an alternative problem formulation, that we call Customer-Oriented Catalog Segmentation, where the overall utility is measured by the number of customers that have at least a specified minimum interest t in the catalogs. We formally introduce the Customer-Oriented Catalog Segmentation problem and discuss its complexity. Then we investigate two different paradigms to design efficient, approximate algorithms for the Customer-Oriented Catalog Segmentation problem, greedy (deterministic) and randomized algorithms. Since greedy algorithms may be trapped in a local optimum and randomized algorithms crucially depend on a reasonable initial solution, we explore a combination of these two paradigms. Our experimental evaluation on synthetic and real data demonstrates that the new algorithms yield catalogs of significantly higher utility compared to classical Catalog Segmentation algorithms.

Categories and Subject Descriptors: H.2.8 [Database Management]:Database Applications-*data mining*

General Terms: Algorithms

Keywords: microeconomic data mining, catalog segmentation, clustering.

1. INTRODUCTION

So far, only few theoretical frameworks for mining useful knowledge from data have been proposed in the literature. The microeconomic framework for data mining [7] is considered as one of the most promising of these models [9]. This framework considers an enterprise with a set of possible decisions and a set of customers that, depending on the decision chosen, contribute different amounts to the overall utility of a decision from the point of view of the enterprise. It is assumed that the contribution of a customer is a possibly complicated, function of the data available on that customer. The enterprise chooses the decision that maximizes the overall utility over all customers.

The microeconomic framework for data mining has in particular been investigated for segmentation (clustering) problems where the enterprise does not make an optimal decision per individual customer but chooses one optimal decision per customer segment. Catalog Segmentation, a specialized segmentation problem, has received considerable attention [6, 7, 10]: the enterprise wants to design k product catalogs of size r that maximize the overall customer purchases after having sent the best matching catalog to each customer[1].

The Catalog Segmentation problem measures the utility of a customer in terms of catalog products purchased. But there are many applications where a customer, once attracted to an enterprise, would purchase more products beyond the ones contained in the catalog. In the case of traditional brick-and-mortar retailers, for example, a customer typically would purchase additional products if the catalog has attracted him to visit the store. In the case of electronic commerce companies, as another example, there is still a substantial overhead involved in visiting a company's website, and customers that have done so are likely to purchase other products from that website that match their interests. Therefore, we investigate an alternative formulation where we measure the overall utility by the number of customers that have at least a specified minimum interest t in the catalog sent to them. A similar problem has been mentioned as an open problem in [6]. We call the new problem Customer-Oriented Catalog Segmentation problem. The major contributions of this paper are as follows:

- We formally introduce several versions of the Customer-Oriented Catalog Segmentation problem and discuss its complexity. This problem was not analyzed in [6, 7].
- We present efficient algorithms for the Customer-Oriented

[1]The Catalog Segmentation problem can formally be defined as follows: given a ground set U and n subsets $S_1, S_2, ..., S_n$(customers' interests) of U, find k subsets $X_1, X_2, ..., X_k$(catalogs) of U, each of size r, so that $\sum_{i=1}^{n} \max(|S_i \cap X_1|, |S_i \cap X_2|, \ldots, |S_i \cap X_k|)$ is maximized[6].

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

Catalog Segmentation problem, exploring the paradigms of greedy and randomized algorithms.

• Our experimental evaluation on synthetic and real data demonstrates that the new algorithms yield catalogs of significantly higher utility compared to classical Catalog Segmentation algorithms.

The rest of this paper is organized as follows. Section 2 reviews related work. In section 3, we formally introduce the Customer-Oriented Catalog Segmentation problem and analyze its complexity. Section 4 presents efficient algorithms for the Customer-Oriented Catalog Segmentation problem. Section 5 reports the results of our experimental evaluation and comparison. The paper concludes with a summary and directions for future research in section 6.

2. RELATED WORK

The microeconomic approach to data mining has been introduced by Kleinberg et al [7] formalizing the optimization problem of enterprises based on data allowing the enterprise to predict the utility of a customer w.r.t. a chosen decision. [7] focuses on a special class of such optimization problems, so-called segmentation problems, and shows that all discussed segmentation problems are NP-complete. The Customer-Oriented Catalog Segmentation problem was not analyzed. [6] also shows how sensitivity analysis of the microeconomic optimization problem can distinguish interesting from uninteresting changes of the decision of the enterprise. In [7], the same authors investigate segmentation problems in more details. As an approximate algorithm for the Catalog Segmentation problem, they outline a sampling-based algorithm (enumerating and measuring all possible partitions of the customers in the sample) and prove probabilistic bounds for its result quality and runtime.

Subsequently, approximate algorithms for the Catalog Segmentation problem have received considerable attention in the algorithms community. Asodi and Safra [2] proved that a polynomial time $(\frac{1}{2}+\epsilon)$-approximation algorithm, for any constant $\epsilon > 0$, would imply $NP = P$. Xu et al [13] developed an approximation algorithm based on semi-definite programming that has a performance guarantee of $1/2$ for general r and of strictly greater than $1/2$ for $r \geq \frac{n}{3}$. In particular, 2-Catalog Segmentation can be approximated by a factor of 0.67 when $r = n/2$.

In the data mining community, the Catalog Segmentation problem has been treated as a clustering problem. Steinbach et al [10] show that the sampling-based enumeration algorithm[6] is infeasible for realistic problem sizes. Instead, they propose two alternative heuristic algorithms and a hybrid algorithm (HCC) combining both of them. The first algorithm, Indirect Catalog Creation (ICC), groups together similar customers using the k-means algorithm and then determines the optimal catalog for each cluster. The second one, Direct Catalog Creation (DCC), iteratively optimizes k catalogs in a manner similar to the EM paradigm. The experimental evaluation demonstrates that DCC and HCC obtain higher overall profit than ICC.

Another research direction that has been inspired by the microeconomic view of data mining is the extension of association rule mining to take into account the indirect profit of products that are frequently purchased together with some other products. Brijs [3] proposes PROFSET to model the cross-selling effects by identifying "purchase intentions" in the transactions. Lin et al [8] introduce a value added model of association rule mining where the value could represent the profit, the privacy or other measures of the utility of a frequent itemset. Wang et al [14] present a method for proposing a target item whenever a customer purchases a non-target item. This method maximizes the total profit of target items for future customers. Wang et al [12] apply the principle of mutual reinforcement of hub/authority web pages in order to rank items taking into account their indirect profits. Addressing a similar problem, Wong et al [11] study the problem of selecting a maximum profit subset of items based on modeling the cross-selling effects with association rules. While all these approaches incorporate the notion of utility into the process of association rule mining, they analyze the relationships between sets of products without considering which customers have purchased these products. These methods aim at suggesting products to individual customers or selecting subsets of (globally) profitable items, but not at Catalog Segmentation or clustering customer databases.

3. PROBLEM FORMULATION

In the microeconomic framework for data mining, there is an enterprise with a set of possible decisions and a set of customers that, depending on the decision chosen, contribute different amounts to the overall utility of a decision from the point of view of the enterprise. It is assumed that the contribution of a customer can be determined based on the data available on that customer. In our case, these data represent the set of products that a customer is interested in. The customer interest can be obtained either from aggregating the history of transactions of that customer or from obtaining explicit customer votes on the set of products. We assume that the (possibly very large) collection of customer data is stored in a Customers Database (Customers DB).

In order to formally introduce our problem and for the presentation of our algorithms, we choose to represent the Customers DB as a bipartite graph. We distinguish two sets of vertices, one for the customers and another one for the products, and an edge denotes the fact that the corresponding customer is interested in the corresponding product. In the following, we introduce the graph-based representation and the related notations.

Notations:

- $G(P, C, E)$ denotes a bipartite graph with two vertices sets P, C and the edges set E:
  $P =$ { all products },
  $C =$ { all customers },
  $E =$ { $(p, c) \mid c$ is interested in $p, p \in P, c \in C$ }.
- For $\forall P' \subseteq P$, $\omega(P') =$ {all the vertices of C which have at least a edge connecting to the vertices in P' }.
- For $\forall P' \subseteq P$, $t \geq 1$, $\psi(P', t) =$ { all the vertices of C which have at least t edges connecting to the vertices in P' }.
- For $\forall P' \subseteq P$, $C' \subseteq C$, $\theta(P', C') =$ { all the edges of E with one end in P' and the other in C' }.
- $||$ denotes the cardinality.

The original Catalog Segmentation problem can be defined in a more illustrative graph-based manner (partition version) as follows [13]: given a bipartite graph $G = (P, C, E)$ with $|P| = m$ and $|C| = n$, find a partition of $C =$

$C_1 \cup C_2 \cup \ldots \cup C_k$, and k subsets $P_1, P_2, \ldots, P_k$ of P, such that $|P_i| = r$ and $\sum_{i=1}^{k} |\theta(P_i, C_i)|$ is maximized. Note that for any i,j, $i \neq j$, $|P_i \cap P_j|$ does not have to be empty.

In the following, we formalize our Customer-Oriented Catalog Segmentation model. We first introduce *MEC* (*Maximum Element Cover*), a well known combinatorial problem, whose goal is to find *one* catalog such that the maximum number of customers is interested in at least *one* of its products.

DEFINITION 1. ***(Maximum Element Cover)*** *Given any bipartite graph $G = \{P, C, E\}$ and a positive integer r, find a subset $P' \subseteq P$ with size r such that $|\omega(P')|$ is maximized.*

We generalize the problem for the case of k catalogs and call it k-MEC (k-Maximum Element Cover) problem.

DEFINITION 2. ***(k-Maximum Element Cover)*** *Given any bipartite graph $G = \{P, C, E\}$ and positive integers r, k, find k subsets $P'_1, \ldots, P'_k \subseteq P$, each with size r, such that $|\omega(P'_1) \cup \ldots \cup \omega(P'_k)|$ is maximized.*

Finally, we introduce a threshold t representing the minimum interest in a catalog necessary to attract a customer, and extend k-MEC to k-MECWT (k-Maximum Element Cover With Threshold t).

DEFINITION 3. ***(k-Maximum Element Cover With t)*** *Given any bipartite graph $G = \{P, C, E\}$ and positive integers r, k, t, find k subsets $P'_1, \ldots, P'_k \subseteq P$, each with size r, such that $|\psi(P'_1, t) \cup \ldots \cup \psi(P'_k, t)|$ is maximized.*

The task of the k-MECWT problem is to find k catalogs maximizing the number of distinct customers who have at least t interesting products in the catalog that is sent to them.

MEC is a well known NP-Complete problem and can be easily reduced from *Set Cover* [5]. In [4], Feige proved that the simple greedy algorithm, iteratively selecting the next product that covers the largest number of uncovered customers, approximates MEC by a ratio of at least $1 - 1/e \approx 0.632$. He showed that this ratio cannot be further improved by any constant number unless $P = NP$. More generally, by a simple Turing reduction we can show that the k-MECWT problem is NP-Complete for any $k, t \geq 1$. Thus, k-MECWT is even harder than the classical Catalog Segmentation problem which is NP-complete only for $k \geq 2$. As an example, for $k = 1$, there is an $O(|P| \cdot |C|)$ algorithm solving the Catalog Segmentation problem that simply picks the r products with the largest number of interested customers. But for k-MECWT and $k = 1$, the simple algorithm enumerating and testing all combinations of r products has a runtime complexity of $O(|P|^r \cdot |C|)$.

From the point of view of clustering, k-MECWT can be understood as follows. The task of k-MECWT is to find k clusters of customers where each cluster is described by a set of products and each customer is assigned to the cluster with the most similar cluster description. There are two constraints for acceptable clusterings: (1) the cardinality of each cluster description is r and (2) customers can only be assigned to a cluster if they have a minimum similarity of t to the cluster description. The clustering objective is to maximize the number of customers assigned to some cluster.

4. ALGORITHMS

Since the Customer-Oriented Catalog Segmentation problem is NP-complete, in this section, we present several approximate, efficient algorithms. All algorithms are based on the graph representation of the Customers DB. We employ adjacency lists as our major data structure: for each product, the corresponding adjacency list contains all customers interested in that product. The list head records the total number of customers in the list. The Customers DB is read once and transformed into the (main memory) adjacency lists that efficiently support the manipulation of the graph structure from the point of view of products. For each customer, we need a counter denoting the number of additional interesting products that this customer requires to be attracted by the current catalog. This data structure is much smaller than the adjacency lists, and the overall space complexity is $O(|E| + |C|) = O(|E|)$, i.e. proportional to the number of edges in the graph G. In subsection 4.1, we explore different greedy, deterministic algorithms. In particular, the Best-Product-Fit algorithm constructs one catalog at a time by choosing the next product for that catalog based on some heuristic quality criteria. The Best-Product-Fit algorithm is very efficient but, due to its greedy nature, may return a solution which is only locally optimal. Therefore, we also investigate randomized algorithms (subsection 4.2) that iteratively optimize the result of a greedy algorithm, e.g. the Random-Product-Fit algorithm.

4.1 Greedy Algorithms

The basic idea of greedy algorithms for the Customer-Oriented Catalog Segmentation problem is as follows: one catalog is constructed at a time by choosing the "best" next product for the current catalog. The "goodness" of a product is measured by criteria such as the number of customers interested and the products already chosen for the catalog.

Since our objective is to maximize the overall number of customers that have enough interests in at least one of the catalogs, a naive greedy algorithm would always pick the remaining product with the largest number of interested customers. Customers that are already interested in at least t products from the current catalog cannot increase the overall number of customers attracted by that catalog and are not considered by the calculation of this product goodness.

While the naive greedy algorithm is very efficient, it does not take the threshold t into account. This decreases the quality of its resulting solutions whenever the product with the maximum number of interested customers does not cover (a good number of) customers that have already been covered by catalog products.

Since the naive greedy algorithm does not take the threshold t into consideration when choosing the next product, it may choose products interesting for customers whose overall interests in the catalog may never reach the specified threshold. To avoid this waste of resources, we need to increase the priorities of products connected to customers which are already interested in other catalog products. The Best-Product-Fit algorithm, that will be introduced below, assigns a score to each product based on the (remaining) customers interested in that product and the number of additional interesting products that these customers need to be attracted to the current catalog. Before stating our algorithm, we define some notions based on a Customers DB in graph representation $G = (P, C, E)$.

$CustomersCovered$={customers who have already t interests in one of the catalogs};

$\overline{C} := C - CustomersCovered$.

$Counter(c)$ =the counter associated to customer c. Initially, $Counter(c) = t$.

We define the score of product p w.r.t the current catalog cat by the following equation:

$$Score(p) = \sum_{c \in \overline{C}, (p,c) \in E} \frac{1}{Counter(c)} + \left|\{c \in \overline{C} \,|\exists p' \in cat, (p', c) \in E, (p, c) \in E\}\right|.$$

The score depends on two terms. The first term represents the weighted number of all customers interested in product p, where the weight of the customer is the inverse of its counter (the weight is the higher, the more interests the customer already has in the current catalog). The second term focuses only on customers that are already interested in at least one of the current catalog products and measures how many of these customers are also interested in p. As an optimization, for the second term, we do not count the customers who need more than $r - |cat|$ further products to be fully covered. The pseudocode of the Best-Product-Fit algorithm is as follows:

ALGORITHM Best-Product-Fit:
INPUT: (1)Customers DB $G = (P, C, E)$, (2)number k of catalogs, (3)number r of products in each catalog, (4)the t threshold
OUTPUT: k catalogs & k corresponding clusters of customers
METHOD:

```
(1)FOR i = 1 to k DO
(2)   FOR j = 1 to r DO
(3)      FOR each p ∈ P DO
(4)         Calculate Score(p);
(5)      Add the Product p with largest
         Score(p) to Catalog i;
(6)      FOR each c with (p,c) ∈ E DO
(7)         Counter(c) := Counter(c) − 1;
(8)      Remove customers whose counter is 0
         and recalculate Score(p) for all products P
         interesting to those customers.
(9)      FOR each c ∈ C DO
(10)        IF Counter(c) > 0 THEN
(11)           Counter(c) := t
(12)Return k clusters of customers with k catalogs.
```

The runtime complexity of the Best-Product-Fit algorithm is $O(kr|E|)$ where $|E|$ is the total number of edges in the graph, i.e. the total number of interests over all customers.

The algorithm requires only one scan of the database if the memory can hold the necessary data structures. Otherwise, we can adopt the divide-and-conquer approach to scale up the Best-Product-Fit algorithm. First, we partition the Customers DB into several subsets DB_1, DB_2,...,DB_p that each can fit into the memory. Then we apply the Best-Product-Fit algorithm to each subset DB_i to determine k catalogs and combine those $k \cdot p$ catalogs into k final catalogs. This algorithm still requires only one database scan.

4.2 Randomized Algorithms

Greedy algorithms find a local optimum only. They may include products into their catalogs that are interesting for many customers that ultimately may not have enough interest (i.e. t interesting products) in the catalog. This weakness is due to the heuristic nature of the quality criterion for individual products and to the deterministic nature of the algorithm. The proposed greedy algorithm has no means of backtracking from some suboptimal choice of a catalog product. This problem is illustrated by the example in Figure 1 with $k = 2$, $t = 2$ and $r = 2$. The Best-Product-Fit algorithm would pick $Diaper$ first since it has largest number of interested customers and then select $Beer$ as the second product of $Catalog_1$ because it covers two customers who have already been interested in $Diaper$. $Diaper$ is still the product with the largest number of interested customers who have not yet been covered by the $Catalog_1$ and is therefore chosen as the first product of $Catalog_2$. As the second product, VCR is chosen because it is interesting for customer 7 (that is already interested in the first catalog product) and for customer 8 and 9. In this solution, only customer 7 meets the interest threshold, while $Catalog_2 = \{VCR, Coke\}$ covers three customers (7,8,9). Due to the lack of a look-ahead mechanism, the choice of $Diaper$ as the first product of $Catalog_2$ leads into a local optimum that cannot be escaped by the greedy method.

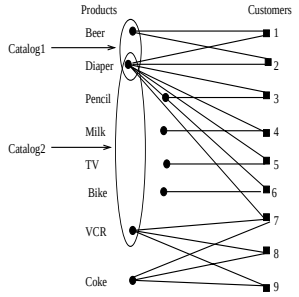


Figure 1: Customers DB in Graph Representation

In order to overcome these limitations, randomized algorithms seem to be promising. Since the performance of randomized algorithms crucially depends on appropriate initial solutions, we propose to combine the greedy deterministic algorithm with a randomized algorithm in a two-step approach (Random-Product-Fit):

- A greedy deterministic algorithm (e.g. Best-Product-Fit) is used to efficiently determine a good solution of the Customer-Oriented Catalog Segmentation problem.
- The resulting catalogs and corresponding clusters are iteratively optimized by randomly replacing one catalog product by a non-catalog product (Random-Product-Switch).

There are different alternatives for the second randomized step with more or less deterministic aspects. A fully randomized algorithm would randomly select one of the catalogs, one of its products and one non-catalog product for replacement. More deterministic versions would select the catalog and the product to be replaced in a deterministic way, e.g. in a round robin fashion. There are two major types of termination conditions for randomized algorithms. They can terminate either after a user-specified number of iterations or as soon as the number of customers covered no longer increases. For simplicity, we propose a fully randomized algorithm with a user-specified number of iterations.

To efficiently support our randomized algorithm, we introduce an additional data structure for each customer c consisting of a customer id (Id) and one list of products for each of the catalogs recording the interesting products (CatalogInterests[k]). This data structure enables us to efficiently calculate the gain (δ) in the number of customers

covered caused by the replacement of catalog product p by p'. The space requirement of this additional data structure is $4k \cdot |C|$ bytes. The pseudocode of the Random-Product-Switch algorithm is as follows:

ALGORITHM Random-Product-Switch:
INPUT: (1)k catalogs & k corresponding clusters of customers (2)number r of products in each catalog, (3)the threshold t and (5)number of iterations s
OUTPUT: k catalogs & k corresponding clusters of customers
METHOD:

```
(1) Calculate the number N_customer of all customers
    covered by the current catalogs;
(2) FOR n = 1 To s DO
(3)     Randomly select a catalog cat;
        Randomly select a product p from cat;
        Randomly select a product p' ∈ {P − cat};
(4)     Calculate the gain δ in N_customer by
        replacing p with p' in cat;
(5)     IF δ ≥ 0 THEN
            Replace p by p' in cat;
            N_customer := N_customer + δ;
(6)Return k clusters of customers with k catalogs.
```

The runtime complexity of the Random-Product-Switch method is $O(sk|C|)$ since in the second step, in each iteration, for each customer we need to access all k elements of CatalogInterests[k] in order to update the number $N_{customer}$ of all covered customers. As the two-step Random-Product-Fit approach consists of (1) Best-Product-Fit (2) Random-Product-Switch methods, the overall runtime complexity of Random-Product-Fit is $O(kr|E|) + O(sk|C|)$.

5. EXPERIMENTAL EVALUATION

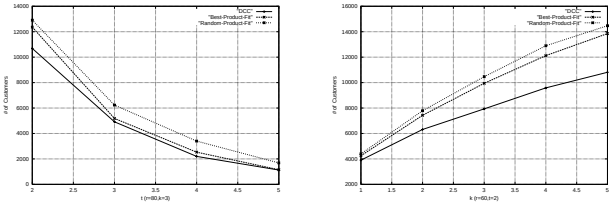
In this section, we report the results of our experimental evaluation using synthetic as well as real datasets. The synthetic datasets were generated using the well-known IBM data generator [1] with different parameter settings. The real dataset records the purchasing transactions of the customers of a large Canadian retailer over a period of several weeks. Since the Customer-Oriented Catalog Segmentation problem has not yet been addressed in the literature, we compare our proposed algorithms with one of the state-of-the-art algorithms [10] for the related Catalog Segmentation problem. We choose DCC as our comparison partner because of the following two reasons. First, the experimental evaluation in [10] showed that DCC, together with HCC, achieved the highest quality results. Second, DCC scales better to large customers databases than HCC because DCC can, different from HCC, use storage efficient adjacency lists instead of an adjacency matrix. Due to the limitation of space, we only report the results of the algorithms w.r.t. utility(quality) and omit the efficiency results.

We evaluate the quality of the catalogs obtained by our algorithms Best-Product-Fit and Random-Product-Fit as well as DCC. Since DCC has been developed for a related, but different problem formulation, we measure the resulting quality w.r.t. both the objective functions of classical Catalog Segmentation (catalog products purchased) and Customer-Oriented Catalog Segmentation (customers covered). To demonstrate the extra profit achievable by Cust omer-Oriented Catalog Segmentation, we also measure the number of non-catalog products that are additionally purchased by customers interested in their corresponding catalogs. We have experimented with several different synthetic datasets, but here we only report results for a dataset with $|C| = 50,000$, $|P| = 7,374$ and $|E| = 376,713$ that seems to be representative for a medium-sized customers database. For the real dataset, $|C| = 45,394$, $|P| = 23,182$ and $|E| = 355,908$.

We compare the numbers of the customers covered (i.e., interested in at least t catalog products) w.r.t. t, k and r on the synthetic dataset. The impact of different values of t w.r.t. the numbers of customers covered is depicted in Figure 2(a) in the case of $r = 80$ and $k = 3$, both Best-Product-Fit and Random-Product-Fit yield higher coverages of customers than DCC, while Random-Product-Fit always covers more customers than Best-Product-Fit.

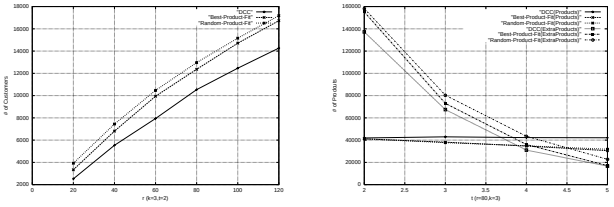
While the effects of different k values on the total number of covered customers in the case of $r = 60$ and $t = 2$ shown in Figure 2(b), reflects the fact that more customers will be covered if more catalogs are created. Random-Product-Fit method always covers the largest number of customers since it has more chances to check and switch more catalog products to cover more customers. Best-Product-Fit still covers more customers than DCC. These results are confirmed by our experiments on the real dataset. The corresponding numbers of covered customers in Figure 5(a) shows the corresponding numbers of covered customers for $r = 30$ and $t = 2$. It also illustrates that the advantage of Random-Product-Fit compared to Best-Product-Fit grows with increasing k values.

The relationship between the size r of the catalog and the number of covered customers with $k = 3$ and $t = 2$ provided in Figure 3(a), has similar results as Figure 2(b). For example, for $r = 100$, the catalog generated by Random-Product-Fit attracts 2,700 (22%) more customers than the DCC catalogs.



(a) Customers covered vs. t (b) Customers covered vs. k

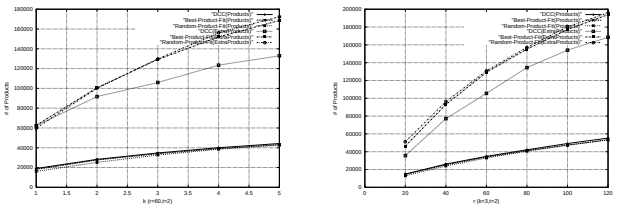
Figure 2: Synthetic Dataset Test 1



(a) Customers covered vs. r (b) Products covered vs. t

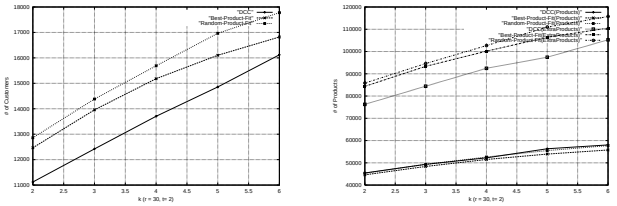
Figure 3: Synthetic Dataset Test 2

The profit in terms of the total numbers of catalog products and the numbers of extra products (beyond the catalogs) w.r.t. t, k and r are also investigated on the synthetic dataset. When measuring the number of catalog products



(a) Products covered vs. k (b) Products covered vs. r

Figure 4: Synthetic Dataset Test 3



(a) Customers covered vs. k (b) Products covered vs. k

Figure 5: Real Data Test 1

covered, these experiments favor DCC due to the different objectives of the comparison partners.

We observe the numbers of products covered w.r.t different values of t in the case of $r = 80$ and $k = 3$ in Figure 3(b). It is expected that DCC covers more products in the catalogs than our methods, but both Best-Product-Fit and Random-Product-Fit have higher extra profits on non-catalog products than DCC. Finally, Random-Product-Fit always covers more extra products than Best-Product-Fit.

It is clear to see the effects of different k values on the same quality measures for $r = 60$ and $t = 2$ in Figure 4(a). All three methods have similar performance w.r.t. the profit on catalog products. However, both of our methods clearly outperform DCC w.r.t. the extra profit from non-catalog products. For example, for $k = 5$, Random-Product-Fit achieves an extra profit of 40,000 products (30%) compared to DCC. We obtain similar results on the real dataset, e.g. with $r = 30$ and $t = 2$ (Figure 5(b)). Figure 4(b) shows how the size r of the catalog affects the number of covered catalog products and extra products with $k = 3$ and $t = 2$. The results are comparable to the results in Figure 4(a).

6. CONCLUSIONS

The microeconomic view of data mining is one of the most promising theoretical frameworks capturing the notion of utility of the discovered knowledge. This data mining framework has in particular been investigated for segmentation problems such as the Catalog Segmentation problem. In this paper, we have investigated an alternative problem formulation measuring the overall utility by the number of customers that have at least a specified minimum interest t in the catalog. We have formally introduced several versions of the Customer-Oriented Catalog Segmentation problem and analyzed its complexity. We have presented efficient, approximate algorithms adopting the paradigms of greedy and randomized algorithms. Our experimental evaluation on synthetic and real data showed that the new algorithms yield catalogs of significantly higher utility compared to classical Catalog Segmentation algorithms. Our best algorithm, Random-Product-Fit, achieves an excellent tradeoff between quality and runtime by optimizing a greedily determined initial solution in a randomized manner.

We believe that this research does not only have many promising applications, but also indicates several interesting directions that deserve further investigation. In order to better judge the relative utility values obtained by different algorithms, it is necessary to develop methods to estimate the utility of the optimal solution. The optimum can only be approximated since k-MECWT is NP-complete even for $k = 1, t = 1$. To make the k-MECWT model even more realistic, it could be generalized by replacing the crisp threshold by a probabilistic threshold, i.e., a customer would be attracted to a catalog with some probability. The Customer-Oriented Catalog Segmentation model should also be studied in the case that the number of catalogs is not set in advance, but there is a fixed cost for each catalog. Finally, Customer-Oriented Catalog Segmentation could be combined with association rule mining techniques to find novel types of customer purchase patterns.

Acknowledgements
We thank Dr. Pavol Hell for valuable comments and suggestions on this study.

7. REFERENCES

[1] R.Agrawal. IBM synthetic data generator. 1994.

[2] V.Asodi and S.Safra. On the complexity of the catalog segmentation problem. Unpublished manuscript

[3] T.Brijs, B.Goethals, G.Swinnen, K.Vanhoof and G.Wets. A Data Mining Framework for Optimal Product Selection in Retail Supermarket Data: The Generalized PROFSET Model. In *Proc. of SIGKDD 2000.*

[4] U.Feige. A threshold of ln n for approximating set cover, *J. ACM 45(4)* pages 634 - 652 1998.

[5] M.R.Garey and D.S.Johnson. *Computers and Intractability, a guide to the Theory of NP-completeness.* W.H. Freeman and company, 1979.

[6] J.Kleinberg, C.Papadimitriou, and P.Raghavan. Segmentation problems. In *Proc. of 13th Symposium on Theory of Computation*, 1998.

[7] J.Kleinberg, C.Papadimitriou, and P.Raghavan. A Microeconomic View of Data Mining. In *Journal of Data Mining and Knowledge Discovery*, 1998.

[8] T.Y.Lin, Y.Y.Yao and E.Louie. Mining values added association rules. In *Proc. of PAKDD 2002.*

[9] H.Mannila. Theoretical Framework for Data Mining. In *SIGKDD Explorations*, Jan. 2000.

[10] M.Steinbach, G.Karypis and V.Kumar. Efficient Algorithms for Creating Product Catalogs. In *Proc. of SDM 2001.*

[11] R.C.W.Wong, A.W.C.Fu and K.Wang. MPIS: Maximal-profit item selection with cross-selling considerations. In *Proc. of ICDM 2003.*

[12] K.Wang and M.Y.Su. Item selection by "hub-authority" profit ranking. In *Proc. of SIGKDD 2002.*

[13] D.Xu, Y.Ye, and J.Zhang. Approximate the 2-Catalog Segmentation Problem Using Semidefinite Programming Relaxations. In *Optimization Methods and Software.*

[14] K.Wang, S.Q.Zhou and J.W.Han. Profit Mining: From Patterns to Actions. In *Proc. of EBDT 2002.*