Dynamic Micro Targeting: Fitness-Based Approach to Predicting Individual Preferences

Tianyi Jiang, Alexander Tuzhilin
New York Univeristy
tjiang, atuzhili@stern.nyu.edu

Abstract

It is crucial to segment customers intelligently in order to offer more targeted and personalized products and services. Traditionally, customer segmentation is achieved using statistics-based methods that compute a set of statistics from the customer data and group customers into segments by applying clustering algorithms. Recent research proposed a direct grouping-based approach that combines customers into segments by optimally combining transactional data of several customers and building a data mining model of customer behavior for each group. This paper proposes a new micro targeting method that builds predictive models of customer behavior not on the segments of customers but rather on the customer-product groups. This micro-targeting method is more general than the previously considered direct grouping method. We empirically show that it significantly outperforms the direct grouping and statistics-based segmentation methods across multiple experimental conditions and that it generates predominately small-sized segments, thus providing additional support for the micro-targeting approach to personalization.

Index Terms: *Customer segmentation, marketing application, personalization, micro targeting, customer profiles*

1. Introduction

Customer segmentation, such as customer grouping by the level of family income, education, or any other demographic variable, is considered as one of the standard techniques used by marketers for a long time [20]. Its popularity comes from the fact that segmented models usually outperform aggregated models of customer behavior [21]. More recently, there has been much interest in the marketing and data mining communities in learning *individual* models of customer behavior within the context of ***1-to-1*** marketing [18] and personalization [4], when models of customer behavior are learned from the data pertaining only to a particular customer. These learned individualized models of customer behavior are stored as parts of customer profiles and are subsequently used for recommending and delivering personalized products and services to the customers [2].

As was shown in [13], it is a non-trivial problem to compare segmented and individual customer models because of the tradeoff between the sparsity of data for individual customer models and customer heterogeneity in aggregate models: individual models may suffer from sparse data, while aggregate models suffer from high levels of customer heterogeneity.

A typical approach to customer segmentation is based on the *statistics-based* approach that computes the set of statistics from customer's demographic and transactional data [3, 13, 23], such as the maximal and minimal times taken to buy an online product, RFM statistics [16], etc. After such statistics are computed for each customer, the customer base is partitioned into segments by using various clustering methods on the space of the computed statistics [13]. It was shown in [13] that while the best statistics-based approaches can be effective and even outperform the ***1-to-1*** case under certain conditions, the approach can also be very ineffective as different customer statistics calculations result in different n-dimensional spaces and various distance metrics or clustering algorithms would yield very different clusters.

Recent research [12] proposes the *direct grouping* segmentation approach that partitions the customers not based on computed statistics and particular clustering algorithms, but in terms of directly combining *transactional data* of several customers, such as Web browsing and purchasing activities, and building a single model of customer behavior on this combined data. This approach avoids the pitfalls of the statistics based-approach in that it does not require selection of arbitrary statistics and grouping customers based on these statistics. Instead, it provides a more direct approach to customer segmentation by combining customers' data and identifying the groups of customers generating the best models on this data. It was shown in [12] that the *direct grouping* segmentation approach dominates the statistics-based segmentation and the ***1-to-1*** approaches.

In this paper we aim to improve the performance of previous customer *segmentation* approaches [11-13] via the method of *micro targeting*, where predictive models of customer behavior are built not on the segments of customers but rather on the customer-product groups. Table 1 shows a Product Type × Customer matrix describing which of the L possible products N of the customers have purchased. Given this purchasing information, we could build predictive models of customer purchase behavior over specific regions within this Product Type × Customer space. For example, we can build a regression model predicting the purchase volume of jazz CDs that a group of freshman students from University of XYZ majoring in computer science would buy on a monthly basis. Then the research problem is to *identify optimal regions in the*

Product Type × Customer space on which the best predictive models of customer purchasing behavior are built.

TABLE 1. PRODUCT TYPE × CUSTOMER MATRIX OF PURCHASES
(√ stands for a purchase)

	$Customer_1$	$Customer_2$	…	$Customer_N$
$ProductType_1$		√		
$ProductType_2$	√		√	
…	…	…	…	…
$ProductType_L$	√			√

Previous research [11, 12] has considered only vertical partitioning of the Product Type × Customer matrix that grouped customers into segments by combining *all* their purchasing transactions. In this paper, we present a *micro targeting* approach that identifies the most suitable *regions* in the Product Type × Customer space for building the best *local* predictive models of customer behavior rather than grouping all the customer's transactions into segments. For example, we may want to build a model predicting whether a customer is going to purchase a particular product during a visit to a website, and we may want to build this model for a certain segment of customers (e.g. freshman NYU students) *and* a certain category of products (e.g., jazz CDs).

The proposed micro-targeting approach is based on the observation that a customer may possess different underlying utility functions across different types of products, and therefore should be modeled separately for different product types. The advantage of this approach is that the *micro-targeting region* is a smaller and more flexible unit of analysis than a customer segment. Therefore, identification of the best micro-targeting regions is a more general problem than identification of the best customer segments, and thus should produce superior performance. The problem with the micro-targeting approach is that the identified regions can be too small and not have sufficient data to build any meaningful models of customer behavior, and we address this problem in the paper.

In this paper, we compare predictive performance of the micro-targeting approach with the previously studied methods, including direct grouping, and demonstrate that the micro-targeting approach *significantly outperforms* these other methods *by a wide margin*. We also show that computational performance of our method is comparable to previously studied direct grouping approach, thus achieving significant predictive performance improvements without excessive increases in computational costs.

2. Problem Formulation

The problem of optimal segmentation of a customer base by customer and product type can be formulated as follows. Assume that $C = \{ C_1,...,C_N,\}$ is the customer base consisting of N customers, and that there are L product types $P = \{P_1, \ldots, P_L\}$. Each customer C_i is defined by the set of m demographic attributes $A = \{A_1, A_2, ..., A_m\}$, and for each product type P_r, customer C_i performed k_{ir} transactions $Trans(C_i, P_r) = \{TR_{ir1}, TR_{ir2}, ..., TR_{irk_{ir}}\}$, where transaction TR_{irj} is defined by its schema $T = \{T_1, T_2, ... T_p\}$ and the set of transaction values $\{t_{irj_1}, t_{irj_2}, ... t_{irj_p}\}$, each value t_{irj_q} corresponding to attribute T_q of schema T. Finally, we combine the demographic data $\{A_{i1}, A_{i2}, ..., A_{im},\}$ of customer C_i and his/her set of transactions for product P_r, $Trans(C_i, P_r)$, into the complete set of customers' product specific data $TA(C_i, P_r) = \{A_{i1}, A_{i2}, ..., A_{im}, TR_{ir1}, TR_{ir2}, ..., TR_{irk_{ir}}\}$ which constitutes a unit of data analysis in our work. For example, a customer C_i can be defined by attributes A= {Name, Age, and other demographic attributes} and by the set of purchasing transactions $Trans(C_i, P_r)$ she made at a Web site for a product type "book", each transaction defined by such transactional attributes T as book title, when it was purchased, and the price of the book. Note that some customers purchase only subsets of products L. Thus $Trans(C_i, P_r)=\{\}$ if customer C_i purchased no products of type P_r.

Given the set of purchasing transactions $s_i = \{ TA_1, TA_2 \ldots, TA_u\}$ performed by customers $C_1,...,C_n$, over product categories $P_1, \ldots, P_r$ from the Product Type × Customer space, we want to build a predictive model M_i on this set s_i and measure its performance using some *fitness function f* mapping transactions s_i into reals, i.e., $f(s_i) \in \Re$. For example, model M_i can be a decision tree built on transactional data s_i of freshmen student customers from University XYZ who bought product categories jazz and hip-hop CDs, and the fitness function f measures predictive accuracy of decision tree model M_i built on s_i using 10-fold cross-validation.

Furthermore, we partition the Product Type × Customer space into a mutually exclusive collectively exhaustive set of regions $S = \{s_1,...,s_k\}$, build models M_i for each region s_i, as described above, and compute their fitness scores $f(s_i)$ using any of the standard methods (e.g. area under ROC curve or predictive accuracy using 10-fold cross-validation). We weight each region s_i, according to its importance α_i and compute the overall fitness score for the partition S as

$$\theta = \sum_{i=1}^{k} \alpha_i * f(s_i) \qquad (1)$$

Finally, we want to find the best partition of the overall set of customer transactions into regions $S = \{s_1,...,s_k\}$ that optimizes the overall fitness score θ over all possible partitions. This optimal partitioning problem is easily reducible to the problem of partitioning C into a set of customers, which was proven in [12] to be NP-hard and, thus, is intractable. Therefore, in this paper, we propose a suboptimal polynomial-time micro-targeting method that has substantially better predictive performance when compared to the previous segmentation methods.

3. Related And Background Work

Our proposed micro-targeting method is related to the reduction-based method for providing multi-dimensional

recommendations [1], where certain segments of ratings are selected from the multi-dimensional cube of ratings and recommendation algorithms build local recommendation models using these and only these segments of ratings. However, unlike [1], we focus on building general local models of customers in this paper, rather than building particular types of local models for solving the multidimensional recommendation problem.

Our work is also related to the work on segmentation by context [9], where product type can be thought of as a type of context which can be used to build different predictive models of purchase behavior for individual customers. However, rather than using product types as an explicit context constraint, our approach starts with a customer and a single product type segment representing purchasing context. Then it works with multiple customer/product type combinations and builds local models maximizing predictive performance, thus removing contextual constraints.

Among various segmentation methods studied in the marketing literature, the ones that are most closely related to our work are various clustering techniques, mixture models, (generalized) mixture regression models and continuous mixture distributions [21]. Our work differs from this previous research in marketing in that we use the *direct grouping* approach [12] to segment customers without deploying statistics-based clustering methods.

Our micro-targeting approach builds on previous work on direct grouping method [11, 12]. Since we compare performance of these two and the statistics-based approach in the paper, we describe the direct grouping and the statistics-based segmentation methods in more detail below.

3.1 Statistics-Based Segmentation Method

The statistics-based approach to customer segmentation first computes the set of summary statistics from customer's demographic and transactional data [3, 13, 23], such as the maximal and minimal times taken to buy an online product, RFM statistics [16], etc. After such summary statistics are computed for each customer, the customer base is partitioned into customer segments by using various clustering methods on the space of the computed statistics [13]. In particular, in this paper, we use hierarchical clustering method to segment the customers and build predictive models of their behavior on these segments. More specifically, we consider a variant of the hierarchical approach below that is described in [6, 10] and deployed in [13, 17], as well as a variant of the previously proposed *affinity propagation (**AP**)* clustering algorithm [8] that is presented in [11].

***Hierarchical Clustering (HC)**:* Using the same hierarchical clustering techniques as in [13], we can learn predictive models of customer behavior of the form

$$Y = \hat{f}(X_1, X_2, \ldots, X_p) \qquad (2)$$

where $X_1, X_2, \ldots, X_p$ are some of the demographic attributes from A and some of the transactional attributes from T (see Section 2), and function $\hat{f}$ is a model that predicts certain characteristics of customer behavior, such as prediction of the product category or the time spent on a Web site purchasing the product. The correctness measure of this prediction is our fitness function f (defined in Section 2). These models $\hat{f}$, defined by expression (2), are built for the groups of customers that are obtained as follows.

We start with a single aggregated grouping of all customers C and compute a set of summary statistics $\{Z_1, \ldots, Z_h\}$, described earlier in this section. Then we use hierarchical clustering methods [6, 10] on the set of these summary statistics to partition the set of N customers (which are viewed as the set of N points in the h-dimensional summary statistics space) by iteratively applying Euclidean distance-based clustering algorithms in the h-dimensional customer summary statistics space. The *Hierarchical Clustering* (***HC)*** method generates new levels of segment hierarchy via progressively smaller groupings of customers until the 1-to-1 level is reached and each segment contains a single customer and his/her transactions. The decision to group certain customers together based on customer demographics attributes and summary statistics $\{A_1, A_2, \ldots, A_m, Z_1, Z_2, \ldots, Z_h\}$ is done using FarthestFirst clustering method [10] that is found to perform well in [13]. We compute progressively smaller customer segments for each level of the clustering hierarchy and build predictive models for these segments at each level. Finally, we compute the overall fitness score (1), where the weights are proportional to the sizes of customer segment, and select the segmentation level with the highest overall fitness score as the best possible segmentation of the customer base.

Affinity Propagation (AP)**:* Starting with n unique customers, ***AP [8] identifies a set of training points, *exemplars*, as cluster centers by recursively propagating "affinity messages" among training points. Similar to greedy K-medoids algorithms, ***AP*** picks exemplars as cluster centers during every iteration, where each exemplar in our study is *a single customer* represented by his/her summary statistics vector. Then ***AP*** forms clusters by assigning an individual exemplar's group membership based on "affinities" that exemplar has with any possible cluster centers. We assume in this paper that affinity is defined as pair-wise Euclidean distance measures an exemplar has with any possible cluster centers. ***AP*** runs in $O(N^2)$. It is a good method for segmenting customers because cluster centers are associated with real customers rather than computed "virtual" customers as in the case of standard clustering algorithms.

3.2 Direct Grouping Method

The *direct grouping* approach, presented in [12], makes decisions on how to group customers into segments by directly combining different customers and their transactions into

groups, building models on the customer data for the group, and measuring the overall fitness score as a linear combination of scores of individual groups. [12] describes the *Iterative Merge (**IM**)* algorithm that works as follows.

Starting from single-customer segments (of size 1), ***IM*** iteratively seeks to merge existing customer segments by combining data from two segments SegA and SegB when (a) the predictive model based on the combined data for segments SegA and SegB performs better than respective models on SegA and SegB and (b) combining SegA with any other existing segments would have resulted in worse performance than the combination of both SegA and SegB. ***IM*** deploys a greedy search strategy since it determines the best pair of customer segments at each iteration and merges them together resulting in the best local solution. The algorithm terminates when there are no more improvements to be made from combining customer segments.

IM runs in $O(N^3)$ time in the worst case, where N is the number of customers, because a single merge of two segments takes $O(N^2)$ time in the worst case, and there can be up to N of such merges. However, in practice, the search space of ***IM*** is not very large because it merges groups, not individual customers, at a time, and the empirical results reported in [12] confirm this observation.

It was shown in [11, 12] that the direct grouping method ***IM*** outperforms the statistics-based approaches, including ***HC, AP*** and some other "traditional" segmentation methods. However, one limitation of the ***IM*** method lies in that it *vertically* partitions the Product Type × Customer matrix presented in Table 1, whereas customer purchasing behavior can vary very significantly across different product categories, such as buying CDs versus diapers. Therefore, we propose the micro-targeting approach in this paper that identifies local regions in the Product Type × Customer space that exhibit homogeneous behavior and builds local predictive models on these regions. In the next section, we present this approach.

4. Building Predictive Models Using Micro-Targeting

Similar to direct grouping methods, such as ***IM***, *micro targeting* method makes locally optimal merging decisions on customer data to improve the overall performance fitness score. Unlike ***IM***, however, this approach goes beyond grouping similar customers. Rather, it tries to identify local regions in the Product Type × Customer space exhibiting truly homogeneous behavior and then builds local predictive models on these regions maximizing predictive performance.

In this paper, we present a specific micro-targeting method, called *Iterative Merge Products (**IM_Prod**)* that differs from ***IM*** in that the unit of analysis is not a single customer but a product category for a customer. The specifics of the ***IM_Prod*** algorithm are presented in Figure 1.

1. Let W = $\{TA(C_1, P_1), TA(C_1, P_2), \ldots, TA(C_1, P_L), TA(C_2, P_1), \ldots TA(C_N, P_L)\}$ // *FIFO queue*
2. CustomerGroupSet S = {} // new set of customer groups
3. While S is changing {
4. While W $\neq 0$ {
5. CustomerGroup $\{CG_i, P_j\}$ = W.pop()
6. CustomerGroup A = new CustomerGroup(TA(CG_i, P_j))
7. $\{CG_s, P_h\}$ = $\{CG_k, P_h\}$ that yields maximum f(A+TA(CG_k, P_h)) $\forall$ $\{CG_k, P_h\} \in$ W;
8. if (f(A+TA(CG_k, P_h)) >= f(A)) {
9. W = {W- $\{CG_s, P_h\}$}; A = {A $\cup$ TA($\{CG_s, P_h\}$)};
10. S={S $\cup$ A};
}
}
11. W= { all $\{CG_i, P_j\}$'s in S}; S={}
}
12. Return S

Figure 1. Iterative Merge Products (*IM_Prod*) Algorithm.

Starting from a single customer and a product type segment, ***IM_Prod*** iteratively seeks to merge existing customer segments by combining data from two segments SegA and SegB similarly to ***IM***. However, for the initial product type and customer specific segments that have very few transactions, where the number of sample points would not be sufficient to build meaningful predictive models, we proceed as follows. We use clustering techniques to group customers' product-specific transactions based on the Euclidian distances between customer's product type and demographic summary statistics vectors so that each cluster results in a set of at least ten purchase transactions for different customers and product types. Having this minimal threshold number of transactions helps to produce more meaningful initial predictive models.

Also note that after ***IM_Prod*** started to merge existing segments, it does not constrain products of different type from grouping into the same segment, nor does it constrain the same customer from having membership in multiple segments. Therefore, ***IM_Prod*** constitutes a generalization of ***IM*** in the sense that the unit of analysis is more "granular" for ***IM_Prod*** than for ***IM***. Thus, we expect ***IM_Prod*** to perform at least as well as ***IM*** if both methods examine all possible regions in the Product Type × Customer space. However, ***IM_Prod*** does not necessarily outperform ***IM*** in practice since there are possible locally optimal customer combinations that ***IM*** examines and forms which ***IM_Prod*** does not consider due to its greedy nature. For example, ***IM*** may group customers C_1, C_2, and C_3 together, whereas ***IM_Prod*** may not produce such group if ***IM_Prod*** had grouped C_1's product P_1 with C_5's product P_1 and C_7's product P_3 into one segment during a previous iteration. Thus, while ***IM_Prod*** searches over a bigger solution space, it still may not perform better than ***IM*** due to this reason.

As ***IM***, ***IM_Prod*** runs in $O(N^3)$ time in the worst case assuming that the number of product types tend to be significantly less than the number of customers ($L << N$). As

for the case of ***IM***, the search space of ***IM_Prod*** is not very large in practice because it merges segments, not individual customers, at a time. Thus the computational performance of ***IM_Prod*** really depends on the number of segments being processed for possible grouping rather than the number of customers N. Our empirical results reported in Section 6 also confirm this observation.

5. EXPERIMENTAL SETUP

To compare the relative performance of statistics-based, direct grouping, and micro targeting approaches, we conduct pair-wise performance comparisons using a variant of the non-parametric Mann-Whitney rank test [15] to test whether the fitness score distributions of two different methods are statistically different from each other. To ensure robustness of our findings, we set up the pair-wise comparisons across the following four dimensions:

1. Types of datasets. We used the following datasets:

(a) Two "real-world" marketing datasets containing panel data (data about a pre-selected group of consumers on whom a comprehensive set of demographic information is collected along with the complete set of their purchases data) of on-line browsing and purchasing activities of Web site visitors and of beverage purchasing activities of "brick-and-mortar" stores. The first dataset contains ComScore data from Media Metrix on Internet browsing and buying behaviors of 100,000 users across the US over 6 months (available at http://wrds.wharton.upenn.edu/). The second dataset contains Nielsen panelist data on beverage shopping behaviors of 1,566 families for a period of one year.

The ComScore and Nielsen marketing datasets are very different in terms of the type of purchase transactions (Internet vs. physical purchases), variety of product purchases, number of individual families covered, and the variety of demographics. Compared to Nielsen's beverage purchases in local supermarkets, ComScore dataset covers a much wider range of products and demographics. We further split these two datasets into four datasets of ComScore high- and low-volume customers, which represents the top and bottom 2,230 customers in terms of transaction frequencies respectively; similarly, Nielsen high- and low-volume customer data was generated using the top and bottom 156 customers in terms of transaction frequencies respectively.

(b) Two simulated datasets representing high-volume (*Syn-High*) and low-volume customers (*Syn-Low*) who performed many and few transactions respectively, where within each dataset, transactions for customer i where generated from the summary statistics vectors S_i as follows. A unique customer summary statistics vector S_i was generated for each of the 2048 customers by sampling from ComScore customer summary statistics distributions, which is then used to generate the purchase transactions with four transactional variables. The number of transactions per customer is also determined from ComScore customer transaction distributions. Rather than generating artificial datasets with normal data distributions, we feel that synthetic dataset that simulates real world transactional datasets is better suited in testing our approach.

TABLE 2. CUSTOMER TYPES AND TRANSACTION COUNTS

DataSet	Customer Type	% of Total Population	Families	Total Trans.	Average Trans. Per Family
ComScore	High	5%	2,230	137,157	62
ComScore	Low	5%	2,230	24,344	11
Nielsen	High	10%	156	28,985	186
Nielsen	Low	10%	156	5,007	32
Syn-High	High	100%	2,048	204,800	100
Syn-Low	Low	100%	2,048	20,480	10

Since for the ComScore and Nielsen we consider two datasets (each having high- and low-volume customers), this means that we use six datasets in total. Their characteristics are summarized in Table 2. In particular, CustomerType column specifies the transaction frequency of these datasets, *High* meaning that customers perform many transactions on average, while *Low* means only few transactions per customer. The columns "% of Total Population", "Families", and "TotalTransactions" specify the percentage of total data population, the number of families, and the sample family transactions contained in the sample datasets.

2. Types of predictive models. We build predictive models using two types of classifiers in Weka 3.4 [22]: C4.5 decision tree [19] and Naïve Bayes [14]. These were chosen because they represent popular and fast-to-generate classifiers.

3. Dependent variables. We built various models to make predictions of transactional variables, TR_{ij}, and compare discussed approaches across different experimental settings. Examples of some of the dependent variables are day of the week, product price, category of website in ComScore datasets, and category of drinks bought, total price, and day of the week in the Neilson datasets. The data we used to train any one model are independent variables X_1, X_2, ..., X_p, defined in Section 2, except previously chosen variable TR_{ij}.

4. Performance measures. We use the following performance measures: percentage of correctly classified instances (CCI), root mean squared error (RME), and relative absolute error (RAE) [22], all measured on the holdout sample as described in Section 2.

For models α and β, α is considered "better" than β when it provides better classification results and fewer errors: when $(CCI_\alpha > CCI_\beta) \wedge (RME_\alpha < RME_\beta) \wedge (RAE_\alpha < RAE_\beta)$. This is the fitness function which we use in ***IM_Prod*** and ***IM*** to select the best possible merge during every iteration. To determine the best segment level in ***HC***, the CCI, RME, and RAE distributions of different segment levels are compared

separately in choosing the best performing segment level that has the most right-skewed CCI distribution and left skewed RME and RAE distributions.

In terms of data pre-processing, we discretized our datasets to improve classification speed and performance [5]. Nominal transaction attributes, such as product categories, were discretized to roughly equal representation in sample data to avoid overly optimistic classification due to highly skewed class priors. We also discretized continuous valued attributes such as price via our implementation of Fayyad's [7] recursive minimal entropy partitioning algorithm.

We compared statistical, direct grouping, and micro targeting based methods across all three dependent variables, six datasets, two classifiers and three performance measures to determine the best method. The results of these comparisons are reported in the next section. Since we have compared performance of ***IM*** with other statistical and direct grouping methods in [11, 12] already, and since ***IM_Prod*** dominates the "best-of-breed" statistical and direct grouping methods, as shown in Section 6, it means that ***IM_Prod*** should dominate these other methods considered in [11, 12], and, therefore, we did not perform these additional comparisons in this work.

6. EMPIRICAL RESULTS

In this section, we present our empirical findings. As mentioned in Section 5, we compare the distribution of performance measures generated by the aforementioned predictive models for individual segments across various experimental conditions. Since we make no assumptions about the shape of the generated performance measure distributions, and the number of sample points differ across distributions as a result of different partition schemes, we use a variant of the non-parametric Mann-Whitney rank test [15] to test whether the distribution of performance measures of the one method is statistically different from another method. For example, to compare ***IM*** against the ***IM_Prod*** method for the CCI measure, assume ***IM_Prod*** generated 150 customer segments and ***IM*** generated 50 customer segments for the purpose of grouping customers in a locally optimal way to predict a customer's purchase on a given website. We want to test the null hypothesis that the two distributions of CCI measure generated from the predictive models built on the segments generated by the ***IM_Prod*** and ***IM*** methods are not different. As mentioned before, we apply the Mann-Whitney rank test to statistically compare distributions of the data points generated by ***IM_Prod*** vs. the data points generated by ***IM*** to determine which method produces a better set of customer segments for predicting customers' next purchase.

More generally, the null hypothesis for comparing distributions generated by methods A and B for a performance measure is:

(I) H_0: The distribution of a performance measure generated by method A *is not* different from the distribution of the measure generated by method B.

H_1+: The distribution of a performance measure generated by method A *is* different from the distribution of the performance measure generated by method B in the *positive* direction.

H_1-: The distribution of a performance measure generated by method A *is* different from the distribution of the performance measure generated by method B in the *negative* direction.

To test these null hypotheses across distributions of performance measures generated by ***HC***, ***AP***, ***IM***, and ***IM_Prod*** methods described in Sections 3 and 4, we proceeded as follows. We ran ***HC***, ***AP***, ***IM***, and ***IM_Prod*** on the ComScore, Nielsen, and synthetic data and generated sets of customer segments for each of these methods and various predictive models and fitness functions. Furthermore, we generated sets of CCI, RME, and RAE scores from 36 predictive models (the combination of 6 datasets for the 2 types of customers, 2 classifiers, C4.5 and Naïve Bayes, and 3 dependent variables) for the total of 108 performance distributions. We then compared the sets of CCI, RME, and RAE scores generated from predictive models of the ***IM_Prod*** generated segments vs. that of ***HC***, ***AP***, and ***IM*** generated segments to see if better performance is achieved with ***IM_Prod*** than with other methods. Table 3 lists the number of Mann-Whitney tests rejecting the null hypothesis (I) at 95% significance level for all the pairwise comparisons of ***HC***, ***AP***, and ***IM*** methods across 108 statistical distribution comparisons (methods listed in the leftmost column are compared against the method listed across the top row).

As Table 3 shows, the ***IM_Prod*** method *overwhelmingly* dominates the other three methods in *all* of the 108 tests. Similarly, ***IM*** also dominates ***HC*** and ***AP*** in all the 108 tests. From the results reported in Table 3, we conclude the following performance relationship among the direct grouping segmentation methods: ***HC<AP<IM<IM_Prod***.

TABLE 3. PERFORMANCE TESTS ACROSS AP, HC, IM, IM_PROD FOR HYPOTHESIS TEST (I)
(Numbers in columns H_1+ and H_1- indicate the number of statistical tests that reject hypothesis H_0. Total significance tests per method to method comparison pair is 108)

Methods	***HC***		***IM***		***IM_Prod***	
	H+	H-	H+	H-	H+	H-
AP	66	18	12	57	0	108
HC	-	-	6	90	0	108
IM_Prod	108	0	96	0	-	-

Table 3 demonstrates that performance differences between ***IM_Prod*** and ***IM*** are significant but does not provide quantitative measures of these differences. To show the extent of these differences, we plot some sample CCI distributions of the "day of the week" predictions from the segments generated by ***IM_Prod*** versus that of ***IM*** on the data from both high and

low volume ComScore datasets. These are only representative examples, and predictive models for other variables exhibit similar trends that we cannot present here because of the space limitation. Figures 2 and 3 do it for the high- and Figures 4 and 5 for the low-volume ComScore customers. As Figures 2 – 5 demonstrate, these distributions are significantly skewed to the right for ***IM_Prod*** in comparison to ***IM***, which demonstrates the extent to which ***IM_Prod*** outperforms ***IM*** for the CCI measure across both high- and low-volume customers.

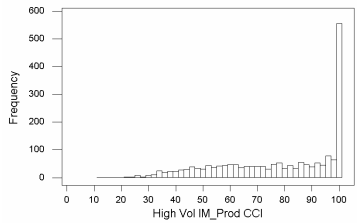

Figure 2. Histogram of CCI measure generated by ***IM_Prod*** customer segment models for the "day of the week" variable in **high volume ComScore data**

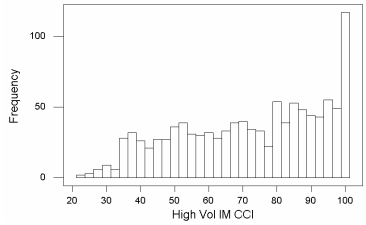

Figure 3. Histogram of CCI measure generated by ***IM*** customer segment models for the "day of the week" variable in **high volume ComScore data**

Besides the clear performance dominance of ***IM_Prod*** versus that of ***IM,*** Figures 2 – 5 also demonstrate that ***IM_Prod*** generates significantly more segments than ***IM***, which is not surprising since ***IM_Prod*** focuses on micro-targeting and partitions the data based not only on populations of customers but also based on product categories. Figures 2 and 4 also demonstrate that the CCI segment distribution among low volume customers generated by ***IM_Prod*** is more positively skewed than for the high volume customers.

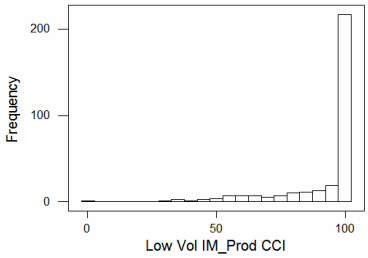

Figure 4. Histogram of CCI measure generated by ***IM_Prod*** customer segment models for the "day of the week" variable in **low volume ComScore data**

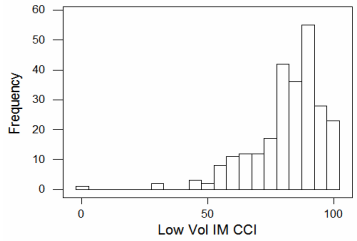

Figure 5. Histogram of CCI measure generated by ***IM*** customer segment models for the "day of the week" variable in **low volume ComScore data**

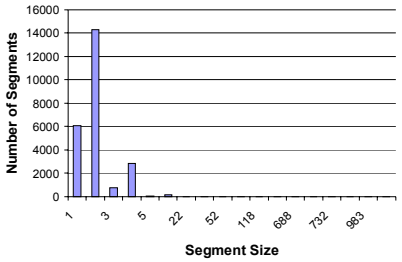

Figure 6. **High volume customer** segment size histogram generated by ***IM_Prod***

To gain further insight into the nature of the low volume CCI distributions, we present the segment size (i.e., number of customers in a segment) distributions in Figures 6 – 7 across the 6 customer type data sets. We note that, unlike ***IM*** generated segment size distributions (Figures 8-9), ***IM_Prod*** generated segments are predominately segments of size 4 and smaller. We also note that unlike what we observed with ***IM*** [12] where high volume customers have proportionally more smaller segment sizes (Figures 8-9), it is the low volume customers that have proportionally more smaller segment sizes in segments generated by ***IM_Prod***. This formation of smaller and thus more homogeneous segments amongst low volume customer datasets help explain why CCI distribution for the low-volume customer segments in Figure 4 is more right skewed than for the high-volume customer segments in Figure 2.

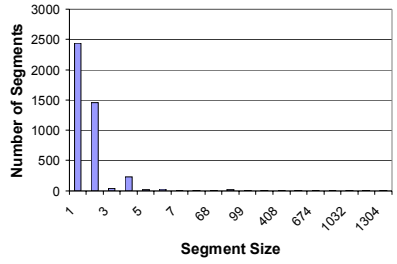

Figure 7. **Low volume customer** segment size histogram generated by ***IM_Prod***

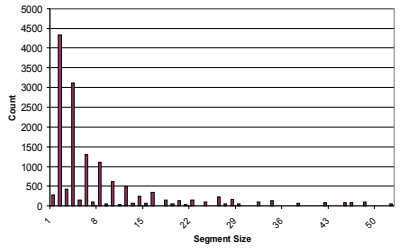

Figure 8. The distribution of segment sizes generated by ***IM*** across **High-volume datasets**

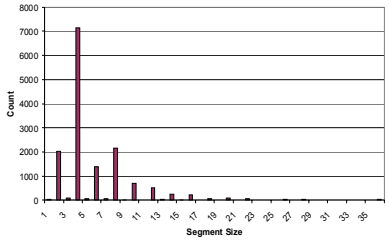

Figure 9. The distribution of segment sizes generated by ***IM*** across **Low-volume datasets**

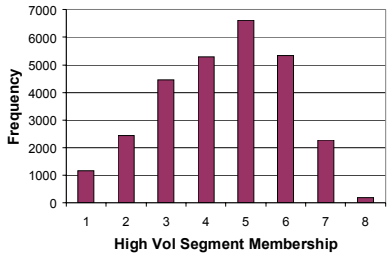

Figure 10. ***IM_Prod*** generated **high volume customer** segment membership count distributions

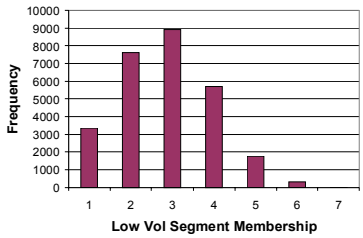

Figure 11. ***IM_Prod*** generated **low volume customer** segment membership count distributions

We noted in Section 1 that ***IM_Prod*** could assign individual customers to multiple customer and product type segments, where the maximum number of segment membership is limited to the total number of product types (in our case, there are eight product types in each of the 6 panel datasets). Therefore, we also studied to how many different segments the customers belong in various datasets, including high- vs. low-volume customers. From frequency distributions of customer segment memberships (Figures 10-11), we observe that high volume customers tend to get assigned to more segments than low volume customers. This makes sense because high volume customers have higher probability to purchase more product types and therefore get assigned to more product type specific segments.

Besides CCI distributions, we also plot sample RME and RAE distributions generated from ***IM*** and ***IM_Prod*** (Figures 12-15) to demonstrate clear dominance of ***IM_Prod*** over ***IM*** as is evident from Figures 12 vs. 13 and 14 vs. 15.

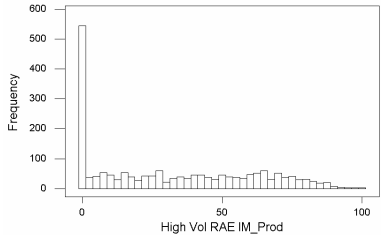

Figure 12. Histogram of RAE measure generated by ***IM_Prod*** customer segment models for the "day of the week" variable **in high volume ComScore data**

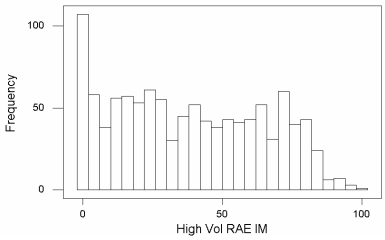

Figure 13. Histogram of RAE measure generated by ***IM*** customer segment models for the "day of the week" variable in **high volume ComScore data**

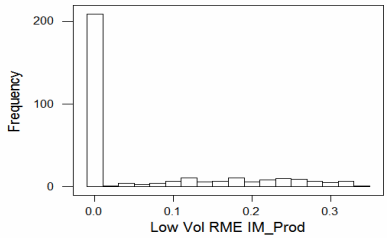

Figure 14. Histogram of RME measure generated by ***IM_Prod*** customer segment models for the "day of the week" variable in **low volume ComScore data**

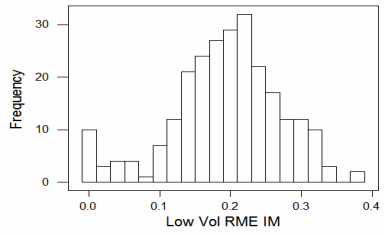

Figure 15. Histogram of RME measure generated by ***IM*** customer segment models for the "day of the week" variable in **low volume ComScore data**

IM_Prod's dominance over ***IM*** is the consequence of our focus on finding more homogeneous regions in the Product Type × Customer space. To get a better sense of the performance impact in that space, we plot average CCI

performance scores across segments of the same size in terms of purchases (as ***IM*** and ***IM_Prod*** generated segments have different meaning in terms of segment size - customers segmented by ***IM_Prod*** do not have the constraint of one customer can only belong to one segment - we cannot compare the segments based on the number of customers assigned to the segments; instead, for the visual analysis of average CCI performance in the Customer × Product space, we group segments together based on the number of purchases within segments) along with the distributions of segment sizes in a three-dimensional space (Figures 16-19).

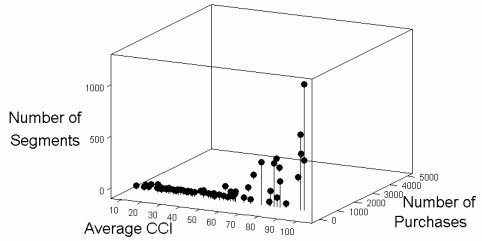

Figure 16. **Low Volume Data**: **IM_Prod** generated clusters in "Segment Count" × "Average CCI per segment" × "Number of Purchases in Segment" space

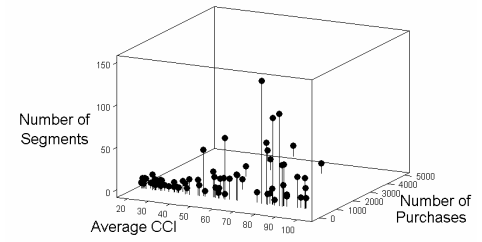

Figure 17. **Low Volume Data**: **IM** generated clusters in "Segment Count" × "Average CCI per segment" × "Number of Purchases in Segment" space

We observe from Figures 16 - 19 that ***IM_Prod*** generated far more segments than ***IM***, the segments tend to be smaller in terms of purchase size, and the predictive models build on these smaller segments perform better in terms of average CCI distributions. This supports our claim that individual customers exhibit different kinds of purchasing behavior across different product types and that it is better to build predictive models for customers purchasing certain product types rather than for the customers across all the product types.

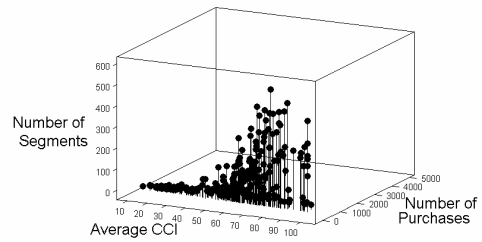

Figure 18. **High Volume Data**: **IM_Prod** generated clusters in "Segment Count" × "Average CCI per segment" × "Number of Purchases in Segment" space

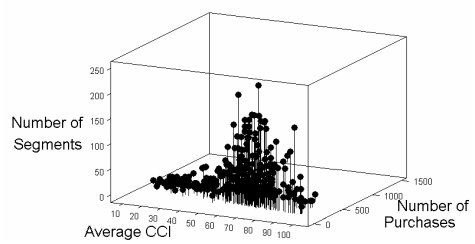

Figure 19. **High Volume Data**: **IM** generated clusters in "Segment Count" × "Average CCI per segment" × "Number of Purchases in Segment" space

Since we build predictive models for Customer × Product combinations with ***IM_Prod***, this raises an important issue of computational performance of the ***IM_Prod*** method because the search space increases significantly in this case in comparison to ***IM***. For example, the search space for ***IM_Prod*** has increased 8-fold in our experiments given the 8 product categories. Although the performance of both ***IM*** and ***IM_Prod*** is $O(N^3)$, as shown in Sections 3 and 4, we compared them experimentally to see how much they differ in practice. The results of this comparison are presented in Figure 20 where the distribution of ratios of run times of ***IM_Prod*** over that of ***IM*** are plotted for all 36 pair-wise comparisons (across 6 datasets, 2 classifiers, and 3 predictive variables per dataset). As Figure 20 demonstrates, the ratio of computational performance of ***IM_Prod*** over ***IM*** is never more than 5.5, and this particular ratio is reached quite infrequently according to Figure 20. In most of the cases, this ratio is 3 or less. These lower than expected ratios indicate that while the combinatorial explosion in the search space alludes to the significantly higher performance ratios, as was shown above and demonstrated in Figures 6 – 9, ***IM_Prod*** tends to generate segments of smaller sizes and converge to the final solution faster than ***IM***. This observation explains why ***IM_Prod*** does not incur significantly more computational expenses than ***IM***, as is empirically shown in Figure 20 and argued in Section 4.

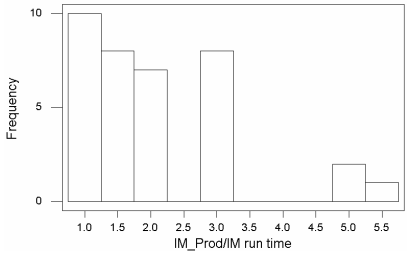

Figure 20. Histogram of run time multipliers of ***IM_Prod*** over ***IM*** across all 36 pair-wise computational expense comparisons

7. Conclusions

In this paper, we proposed a *micro-targeting* approach to address the problem of optimal partitioning of customer bases into homogeneous segments for building better predictive models of customer behavior. This approach extends previously proposed direct grouping method by directly combining *product-specific* transactional data of one or several customers

and building a single local model of customer behavior on this combined data. We presented a polynomial-time direct grouping method, ***IM_Prod*** that identifies segments of customers and product types and builds predictive models on these local regions in the Product-Type × Customer space. We compared performance of ***IM_Prod*** against the traditional statistics-based hierarchical, affinity propagation clustering and the direct grouping method ***IM***. We showed that ***IM_Prod*** significantly dominated by a wide margin all other methods across all the experimental conditions and did not incur additional substantial computational expenses in comparison to ***IM***. We then examined the segments generated by ***IM_Prod*** and observed that there were many small size segments, and that customers were often assigned to multiple segments. Since most of the generated segments are small, this provides strong support for the *micro-targeting* approach to personalization.

By identifying local regions in the Product Type × Customer space that exhibit truly homogeneous behavior and building local predictive models on these regions, the micro-targeting approach increases flexibility of the predictive models describing customer behavior, improves the overall predictive performance, and keeps computational costs at the same level as other direct grouping methods.

As a future research, we would like to test the effectiveness of our segmentation strategies not only in terms of predictive performance but also in terms of the standard marketing oriented performance measures such as customer value, profitability and other economics-based performance measures.

8. References

[1] Adomavicius, G., R. Sankaranarayanan, S. Sen, and A. Tuzhilin, *Incorporating contextual information in recommender systems using a multidimensional approach.* ACM TOIS, 2005. **23**(1): p. 103-145.

[2] Adomavicius, G. and A. Tuzhilin, *Personalization technologies: A process-oriented perspective*, in *CACM*. 2005.

[3] Brijs, T., T. Swinnen, K. Vanhoof, and G. Wets. *Using shopping baskets to cluster supermarket shoppers*. in *AARTF*. 2001. Amelia Island Plantation, FL.

[4] CACM, *Communications of ACM*, in *Special Issue on Personalization*. 2000.

[5] Dougherty, J., R. Kohavi, and M. Sahami. *Supervised and Unsupervised Discretization of Continuous Features*. in *12th ICML*. 1995. San Francisco, CA: Morgan Kaufmann.

[6] Duda, R., P. Hart, and D. Stork, *Pattern Classification*. 2 ed. 2001, New York, NY: John Wiley & Sons, Inc.

[7] Fayyad, U.M. and K.B. Irani. *Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning*. in *IJCAI*. 1993.

[8] Frey, B. and D. Dueck, *Mixture Modeling by Affinity Propagation*. Advances in Neural Information Processing Systems 18, ed. Y. Weiss, B. Scholkopf, and J. Platt. 2006: MIT Press.

[9] Gorgoglione, M., C. Palmisano, and A. Tuzhilin. *Personalization in Context: Does Context Matter When Building Personalized Customer Models?* in *ICDM*. 2006.

[10] Hochbaum, S.D. and B.D. Shmoys, *A Best Possible Heuristic for the K-Center Problem.* Math. of Operational Research, 1985.

[11] Jiang, T. and A. Tuzhilin. *Forming Segments From Individuals Using Direct Grouping Methods*. in *WITS*. 2006.

[12] Jiang, T. and A. Tuzhilin. *Improving Personalization Solutions through Optimal Segmentation of Customer Bases*. in *ICDM*. 2006.

[13] Jiang, T. and A. Tuzhilin, *Segmenting Customers from Population to Individual: Does 1-to-1 Keep Your Customers Forever?* IEEE TKDE, 2006. **18**(10).

[14] John, G.H. and P. Langley. *Estimating Continuous Distributions in Bayesian Classifiers*. in *UAI*. 1995.

[15] Mendenhall, W. and R.J. Beaver, *Introduction to probability and statistics*. 1994: Thomson Pub.

[16] Novo, J., *Drilling Down: Turning Customer Data Into Profits with a Spreadsheet*. 2004: Booklocker.

[17] Ozdal, M. and C. Aykanat, *Clustering Based on Data Patterns using Hypergraph Models.* Data Mining and Knowledge Discovery, 2004. **9**: p. 29-57.

[18] Peppers, D. and M. Rogers, *Enterprise One to One*. 1997, New York: Bantam Pub. Group Inc.

[19] Quinlan, R., *C4.5: Programs for Machine Learning*. 1993.

[20] Smith, W., *Product Differentiation and Market Segmentation as Alternative Marketing Strategies.* Journal of Marketing, 1956. **21**.

[21] Wedel, M. and W. Kamakura, *Market Segmentation: Conceptual and Methodological Foundations*. 2000: Kluwer Pub.

[22] Witten, I.H. and E. Frank, *Data Mining: practical machine learning tools and techniques with Java implementations*. 2000.

[23] Yang, Y. and B. Padmanabhan. *Segmenting Customer Trans. Using a Pattern-Based Clustering Approach*. in *ICDM*. 2003.