Improving Personalization Solutions through Optimal Segmentation of Customer Bases

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

Abstract

On the Web, where the search costs are low and the competition is just a mouse click away, it is crucial to segment the customers intelligently in order to offer more targeted and personalized products and services to them. 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 distance-based clustering algorithms in the space of these statistics. In this paper, we present a direct grouping based approach to computing customer segments that groups customers not based on computed statistics, but in terms of optimally combining transactional data of several customers to build a data mining model of customer behavior for each group. Then building customer segments becomes a combinatorial optimization problem of finding the best partitioning of the customer base into disjoint groups. The paper shows that finding an optimal customer partition is NP-hard, proposes a suboptimal direct grouping segmentation method and empirically compares it against traditional statistics-based segmentation and 1-to-1 methods across multiple experimental conditions. We show that the direct grouping method significantly dominates the statistics-based and 1-to-1 approaches across all the experimental conditions, while still being computationally tractable. We also show that there are very few size-one customer segments generated by the best direct grouping method and that micro-segmentation provides the best approach to personalization.

Index Terms – customer segmentation, marketing application, personalization, 1-to-1 marketing, 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 [25]. Its popularity comes from the fact that segmented models usually outperform aggregated models of customer behavior [26]. 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 [23] and personalization [5], 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 [1].

As was shown in [14], 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. Depending on which effect dominates the other, it is possible that models of individual customers dominate the segmented or aggregated models, and vice versa.

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, 14, 28], such as the average time it takes the customer to browse the Web page describing a product, maximal and minimal times taken to buy an online product, RFM statistics [21], etc. After such 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 [14]. It was shown in [14] that the best statistics-based approaches can be effective in some situations and can even outperform the ***1-to-1*** case under certain conditions. However, it was also shown in [14] that this approach can also be highly ineffective in other cases. This is primarily because computing different customer statistics would results in different n-dimensional spaces, and various distance metrics or clustering algorithms would yield different clusters. Depending on particular customer statistics, distance functions and clustering algorithms, significantly

different customer segments can be generated.

In this paper, we propose 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 collectively resulting in better model for this group of customers.

In this paper we try to partition the customer base into an optimal set of segments using the direct grouping approach, where optimality is defined in terms of a fitness function of a model learned from the customer segment's data. We formulate this optimal partitioning as a combinatorial optimization problem and show that it is NP-hard. Then we propose a suboptimal polynomial-time direct grouping method, called *Iterative Merge (**IM**),* and compare it to the standard statistics-based and ***1-to-1*** approaches. We show that ***IM*** significantly dominates the statistics-based and ***1-to-1*** methods across all the experimental conditions examined in this paper, thus demonstrating the applicability of the direct grouping methods to building personalized models of customers. Therefore, we demonstrate empirically that it is better to segment customer bases by first directly partitioning customer data and then building predictive models from the partitioned data rather than first computing some arbitrary statistics, clustering the resulting *n*-dimensional data points into segments, and then building predictive models on these segments. We also examine the nature of the segments generated by the ***IM*** method and observe that there are very few size-one segments, that the distribution of segment sizes reaches its maximum at a very small segment size, and that the rate of decline in the number of segments after this maximum follows a Zipf's distribution. This observation, along with the dominance of ***IM*** over the ***1-to-1*** method, provides support for the *micro-segmentation* approach to personalization [16], where the customer base is partitioned into a large number of small segments, such as undergraduate students at University of XYZ majoring in computer science and living in the dorms.

In summary, we make the following contributions in this paper:

- Propose the direct grouping method for segmenting customer bases, formulate the optimal segmentation problem, and show that it is NP-hard.
- Propose a suboptimal direct grouping method, ***IM,*** and compare ***IM*** against the statistics-based segmentation and the ***1-to-1*** approaches and demonstrate that ***IM*** significantly dominates them.
- Show that the tail of the cluster size distribution generated by ***IM*** follows a Zipf's distribution and that there are very few size-one clusters. This provides support for the micro-segmentation approach to personalization.

2. Problem formulation

The problem of optimal segmentation of customer base can be formulated as follows. Let C be the customer base consisting of N customers, each customer C_i is defined by the set of m demographic attributes $A = \{A_1, A_2, ..., A_m\}$, k_i transactions $Trans(C_i) = \{TR_{i1}, TR_{i2}, ..., TR_{ik_i}\}$ performed by customer C_i, and h summary statistics $S_i = \{S_{i1}, S_{i2}, ..., S_{ih}\}$, computed from the transactional data $Trans(C_i)$. Moreover, each transaction TR_{ij} is defined by a set of transactional attributes $T = \{T_1, T_2, ..., T_p\}$. The number of transactions k_i per customer C_i varies. Finally, we combine the demographic attributes $\{A_{i1}, A_{i2}, ..., A_{im},\}$ of customer C_i and his/her set of transactions $\{TR_{i1}, TR_{i2}, ..., TR_{ik_i}\}$ into the complete set of customers' data $TA(C_i) = \{A_{i1}, A_{i2}, ..., A_{im}, TR_{i1}, TR_{i2}, ..., TR_{ik_i}\}$ which constitutes a unit of analysis in our work. As an example, assume that customer C_i can be defined by attributes A = {Name, Age, Income, and other demographic attributes}, and by the set of purchasing transactions $Trans(C_i)$ she made at a Web site, where each transaction defined by such transactional attributes T as an item being purchased, when it was purchased, and the price of an item. Finally, a summary statistics vector S_i can be computed for all of C_i's purchasing sessions and can include such statistics as the average amount of purchase per a Web session, the average number of items bought, and the average time spent per online purchase session.

Given the set of n customers $C_1, ..., C_n$, and their respective customer data $p_i = \{TA(C_1),...,TA(C_n)\}$, we want to build a single model M_i of this group of customers p_i and measure its performance using some *fitness function* f mapping the set of customer data p_i into reals, i.e., $f(p_i) \in \Re$. For example, model M_i can be a decision tree built on data p_i of customers $C_1, ..., C_n$, and the fitness function f is its predictive accuracy on the out-of-sample data or obtained using k-fold cross-validation.

The function f can be very complex in general, as it represents the predictive power of an arbitrary predictive model M_i trained on all the customer data contained in p_i. For example, f could the relative absolute error of a neural network model trained and tested on p_i via ten-fold cross validation. Another example could be the R^2 value

generated from a logistic regression of all the transactional and demographic variables on one dependent purchase variable using all data contained in p_i. This means that, in general, function f does not have "nice" properties, such as additivity or monotonicity. For example, $f(\{TA(C_i)\})$ can be greater than, less than, or equal to $f(\{TA(C_i),TA(C_j)\})$ for any i, j. This lack of nice properties of fitness functions will be a defining issue when we formulate an optimal customer segmentation problem later in this section and will make this problem computationally complex.

Partitioning the customer base C into a mutually exclusive collectively exhaustive set of segments $P = \{p_1,...,p_k\}$, by building models M_i for each segment p_i, as described above, is called *direct grouping* segmentation. Note that in this approach we group customers into segments based on some performance criteria for the segment rather than clustering customers based on intra or inter cluster distance measures. We next formulate an optimal segmentation problem that does this partitioning in the "best possible" manner.

Optimal Customer Segmentation problem. Given the customer base C of N customers, we want to partition it into the disjoint groups $P = \{p_1,...,p_k\}$, such that the models M_i built on each group p_i would collectively produce the best performance for the fitness function $f(p_i)$ taken over $p_1,...,p_k$. Formally, this problem can be formulated as follows. Let α_i be a weighting measure specifying "importance" of segment i. Some examples of α_i include simple average $1/k$ and proportional weights $|TA(p_i)|/|TA(C)|$. Then we want to find partition of the customer base C into the set of mutually exclusive collectively exhaustive segments $P = \{p_1,...,p_k\}$, where segment p_i is defined by its customer data $p_i = \{TA(C_j),...,TA(C_m)\}$, such that the following fitness score

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

is maximized (or minimized if we used error rates as our fitness score) over all possible partitions P.

Note that there can be $B_N = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^N}{k!}$ possible combinations of customer transactions groupings, known as the Bell numbers, which are the number of ways that N distinguishable objects can be grouped into non-empty sets. Since B_N are very large numbers, even for small N's, finding an optimal partition constitutes a complex optimization problem. Due to the arbitrary nature of the fitness function f, the optimal customer segmentation is a *combinatorial partition problem* [13] with very little constraints.

Proposition. *Optimal Customer Segmentation (OCS) problem is NP-hard.*

This result can be obtained by reducing the clustering problem, that is NP-hard [4], to the OCS problem.

Since the OCS problem is NP-hard, we propose the following suboptimal polynomial customer segmentation methods providing reasonable fitness scoring results:

Statistics based – These methods group customers by first computing some statistics from customers' demographic and transactional data, consider these statistics as points in an n-dimensional space, and then group customers into segments by applying various clustering algorithms to these n-dimensional points.

1-to-1 – This approach builds customer segments of size 1 (individual models of customers) by learning them only from the data pertaining to individual customers.

Direct Grouping – We described direct grouping in this section. Instead of looking for an optimal grouping of customers, which is NP-hard, we present a polynomial-time suboptimal direct grouping method ***IM*** in Section 4.

Before describing these methods, we present related work.

3. Related work

The problem of finding the global optimal partition of customers is related to the work on a) combinatorial optimization problems in operations research, b) customer segmentation and clustering in marketing, and c) data mining research on customer segmentation. We examine the relationship of our work to these three areas of research in this section.

Combinatorial optimization models are used across a wide range of applications. The common feature among these problems is that in many practical problems, activities and resources, such as tasks and people are indivisible [13]. Combinatorial optimization problems are in general considered NP-hard, however, depending on the mathematical formulation of a particular problem, exact solutions or close to exact approximations can be achieved. While we cannot use any existing problem formulations in solving our research question due to the lack of additivity, monotonicity and other "nice" characteristics of our fitness function, we could take cues from various approaches used in solving combinatorial optimization problems. These approaches include "branch & bound" enumerative techniques [17], Lagrangian relaxation and decomposition methods [10], and cutting plane algorithms based on polyhedral combinatorics[9].

Despite recent advances in finding solutions to various combinatorial optimizations problems, there are still a large set of problems considered too complex to derive optimal solutions [13]. Therefore, various heuristics were explored in obtaining good solutions that have no guarantees as to their "closeness" to the optimal solution. Our research problem falls in this category. Heuristics used by operation

researchers in solving combinatorial optimizations problems include greedy hill-climbing [18], simulated annealing [11], evolutionary algorithms [20], and neural networks [2]. In our work, we deployed the greedy hill-climbing approach in conjunction with the branch and bound enumerative techniques in developing our fitness function based methods.

Our work is also related to the work on clustering that partitions the customer base and their transactional histories into homogeneous clusters for the purpose of building better models of customer behavior using these clusters [3, 14, 28]. However, any objective measure of intra-cluster similarity and inter-cluster dissimilarity is hard to come by and is shown to be rather erratic [14]. Instead, in this paper we group customer's transactions together and measure performance not in some sense of "intra-cluster similarity", but rather based purely on some performance fitness function having direct implication on the effective utility of any subsequent customer groupings.

Building on top of previous data mining research on customer segmentation [14, 28], this research aims to formulate automated customer segmentation methods that are not influenced by arbitrary selection of summary statistics or population specific factors such as customer heterogeneity and data volume.

4. Predictive Models of Customer Behavior

In this section, we describe the details of the statistics-based, 1-to-1 and direct grouping approaches and propose their representative implementations before empirically comparing them across various experimental settings.

4.1. Statistics-based Segmentation Methods

In terms of the statistics-based segmentation, we consider the following two variants of the hierarchical approach that are described in [7, 12] and deployed in [14, 22]:

Hierarchical Clustering (HC)*:* Using the same hierarchical clustering techniques as in [14], we learn predictive models of customer behavior of the form

$$Y = \hat{f}(X_1, X_2, \ldots, X_p) \qquad (1)$$

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 (1), are built for the groups of customers that are obtained as follows.

Starting from a single aggregated grouping of all customers, we use hierarchical clustering methods on the set of summary statistics $\{S_1, \ldots, S_h\}$ and partition the set of m customers by iteratively applying Euclidean distance-based clustering algorithms in the n-dimensional customer summary statistics space. The *Hierarchical Clustering* (***HC)*** method generates new levels of segment hierarchy via progressively smaller groupings of customers' transactions until the single customer (1-to-1) level is reached and each segment contains transactions from a single customer. The decision to group certain customers together is done by clustering via FarthestFirst [12], a greedy k-center unsupervised clustering algorithm that is found to perform well in [14] on customer summary statistics and demographics attributes $\{A_1, A_2, \ldots, A_m, S_1, S_2, \ldots, S_h\}$. We compute these segments for each level of the segmentation hierarchy (containing progressively smaller segments), and for each level L, compute the weighted sum of fitness scores. Then the segmentation level with the highest overall fitness score (besides the 1-to-1 level) is selected as the best possible segmentation of the customer base.

Entropy Clustering (EC)*:* Instead of forming different groupings of customer transactions from unsupervised clustering algorithms, as ***HC*** does, ***EC*** forms customer groupings by building a C4.5 decision tree λ on customer summary statistics and demographics $\{A_1, A_2, \ldots, A_m, S_1, S_2, \ldots, S_h\}$, where the class label is the model's dependent variable Y in (1). Unlike ***HC***, this approach is a supervised clustering algorithm, where "similar" customers are grouped in terms of summary statistics and demographics to reduce the entropy of the class label. Once the C4.5 forms the groupings based on the principle of class label entropy minimization, we compute the weighted sum of fitness scores generated by $\hat{f}$ in (1) across these different groupings of customer transaction data. Intuitively, this should be a better approach to clustering customers than ***HC*** because by making grouping decisions based on class label purity, we are in effect measuring similarity in the output space, which reduces the variance of the dependent variable Y classified by our predictive models. In addition, there is no fixed splitting factor as in the case of ***HC,*** as each tree split is based on the number of different values an independent attribute may have. Thus, each split could result in a different number of sub-clusters, which could provide extra flexibility for building more homogeneous and better performing customer segments. However, the formation of customer groups is still based on customer summary statistics which, depending on the types of statistics used, can yield very different decision trees.

4.2. 1-to-1 Method

As explained in Section 2, the 1-to-1 approach builds predictive models of customer behavior only from individual customer's transactional data. In other words, we build a predictive model (1) for *each* customer C_i, $i = 1, ..., N$, using only the demographic and the transactional data of that customer, and we do not have to deal with customer grouping at all in this case. For each model of customer C_i, we compute fitness function $f(C_i)$ (e.g., using 10-fold cross-validation) and obtain the whole distribution of these fitness scores for $i = 1, ..., N$.

4.3. Direct Grouping Methods

The *direct grouping* approach makes decision on how to group customers into segments by directly combining different customers into groups and measuring the overall fitness score as a linear combination of fitness scores of individual segments, as described in Section 2. Since the optimal segmentation problem is NP-hard (see Section 2), we propose the following suboptimal method ***IM***.

Iterative Merge (IM): IM is an iterative segmentation reduction approach which starts from a set of single customer segments and iteratively merges together two segments that result in better performance combined. More specifically, starting with segments containing individual customers, ***IM*** seeks to iteratively merge two existing segments Seg_A, and Seg_B at a time when 1) the predictive model based on the combined data performs better and 2) combining Seg_A with any other existing segments would have resulted in a worse performance than the combination of both Seg_A and Seg_B. ***IM*** is greedy because it attempts to find the best pair of customers groups and merge them together resulting in the best merging combination. The specifics of the ***IM*** algorithm are presented in Figure 1.

```
1. Let W = {C1,C2,...,CN} // FIFO queue
2. CustomerGroupSet P = {} // new set of customer groups
3. While P is changing {
4.         While W ≠ 0 {
5.                 CustomerGroup CGi = W.pop()
6.                 CustomerGroup A = new
CustomerGroup(TA(CGi))
7.                 CGs = CGk that yields maximum f(A+TA(CGk))
∀CGk∈ W;
8.                 if (f(A+TA(CGk)) >= f(A)) {
9.                         W = {W- CGs}; A = {A ∪ TA(CGs)};
10.                        P={P ∪ A};
                   }
          }
11.       W= { all CG's in P}; P={}
    }
12. Return P
```

Figure 1. Iterative Merge (***IM***) Algorithm.

IM runs in $O(n^3)$ in the worst case because a single merge of two groups 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 Section 6 confirm this observation.

In addition, ***IM*** tends to make merging decisions on customer segments of comparable sizes, where each customer segment under merging consideration can significantly affect the performance of the combined segmentation, thus lessen the chance of building large and poorly performing customer segments.

5. Experimental setup

To compare the relative performance of direct grouping, statistics-based, and 1-to-1 approaches, we conduct pair-wise performance comparisons using a variant of the non-parametric Mann-Whitney rank test [19] 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:

Types of datasets. In our study we worked with the following datasets:

(a) Two "real-world" marketing datasets containing panel data[1] of on-line browsing and purchasing activities of Web site visitors and panel data on beverage purchasing activities of "brick-and-mortar" stores. The first dataset contains ComScore data from Media Metrix on Internet browsing and buying behaviors of one hundred thousand users across United States for a period of 6 months (available via Wharton Research Data Services - 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 and is more representative of today's large marketing datasets. We further split these two real world datasets into four datasets of ComScore high- and low-volume customers,

[1] Panel data [16], also called longitudinal or cross-sectional time series data, when used in the context of marketing means that the data about a pre-selected group of consumers on whom a comprehensive set of demographic information is collected is also augmented with the complete set of their purchases. Therefore, this panel data provides a comprehensive view of purchasing activities of a pre-selected panel of consumers.

which represents the top and bottom 1,000 customers in terms of transaction frequencies respectively. Similarly, Nielsen high- and low-volume customer datasets were generated using the top and bottom 500 customers in terms of transaction frequencies respectively.

(b) Two simulated datasets representing high-volume customers (*Syn-High*) and low-volume customers (*Syn-Low*) respectively, where within each dataset, customer differences are defined by generating different customer summary statistic vector S_i for each customer i. All subsequent customer purchase data are generated from the set of summary statistic vectors S_i.

The Syn-Low and Syn-High datasets were generated as follows. 2,048 unique customer summary statistics were generated 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. This dataset is used to better simulate real world transactional datasets.

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 in our studies. Some of the main characteristics of these six datasets are presented in Table 1. 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.

TABLE 1. CUSTOMER TYPES AND TRANSACTION COUNTS

DataSet	Customer Type	% of Total Population	Families	Total Transactions	Average Transactions Per Household
ComScore	High	2.2%	1,000	137,157	92
ComScore	Low	2.2%	1,000	24,344	10
Nielsen	High	32%	500	28,985	136
Nielsen	Low	32%	500	5,007	46
Syn-High	High	100%	1,024	102,400	100
Syn-Low	Low	100%	1,024	10,240	10

Types of predictive models. Due to computational expenses of the model-based methods, we build predictive models using two different types of classifiers via Weka 3.4 system [27]: C4.5 decision tree [24] and Naïve Bayes [15]. These are chosen because they represent popular and fast-to-generate classifiers.

Dependent variables. We built various models to make predictions on 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 customer C_i's independent variables $X_1, X_2, ..., X_p$, except TR_{ij}.

Performance measures. We use the following performance measures: percentage of correctly classified instances (CCI), root mean squared error (RME), and relative absolute error (RAE) [27].

Given models α and β, α is considered "better" than β only when it provides better classification results and fewer errors, i.e., 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*** to pick the best possible merge during every iteration. To pick 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 [6]. 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 and Internet browsing durations based on entropy measures via our implementation of Fayyad's [8] recursive minimal entropy partitioning algorithm.

We compared statistics-based segmentation methods ***HC*** and ***EC*** across all these dependent variables, classifiers, and six datasets to select the best one. The results of these comparisons are reported in the next section.

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 considered predictive models across various experimental conditions. Since we make no assumptions about the shape of the generated performance measure distributions, we use a variant of the non-parametric Mann-Whitney rank test [19] to test whether the distribution of performance measures of the one method is statistically different from another method. For example, to compare ***HC*** against the ***1-to-1*** method for the CCI measure, consider the distribution of the CCI measure generated for the best segmentation level of the ***HC*** hierarchical clustering, and compare it against the distribution of the CCI measure obtained for each individual customer. Then we apply the Mann-Whitney

rank test to compare the two distributions.

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 performance 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 different methods, we proceeded as follows. For each dataset, classifier and dependent variable we generate 3 sets of customer groups, CG_1, CG_2 and CG_3, using our three segmentation methods ***IM, HC*** and ***EC***. Let cg_{ij} denote a particular group j of customers belonging to customer group set CG_i generated by method i (***IM, HC*** or ***EC***). For each cg_{ij}, we generate a separate model, m_{ij}, that predicts the dependent variable of the model via ten-fold cross validation and computes three performance measures CCI_{ij}, RME_{ij}, and RAE_{ij}.

Let M_i denote the set of models generated from evaluating all customer groups in CG_i for method i, and let CCI_i, RME_i, and RAE_i be three sets of performance measures evaluated on model set M_i for all customer groups in CG_i. To compare segmentation method i's performance against method h, we would compare whether the distribution of performance measures of CCI_i, RME_i, or RAE_i is statistically different from that of CCI_h, RME_h, or RAE_h respectively via the Mann-Whitney rank tests using hypotheses H_0, H_1+, H_1- specified above.

For example, for the comparisons involving ***HC*** and ***1-to-1***, the above scenario of comparing three measures is repeated across six datasets, three dependent variables per dataset, and two classifiers, resulting in 108 statistical significance tests per method to method comparison pair.

We next compare ***HC*** against ***EC*** and the direct grouping method ***IM*** against the statistics-based and the ***1-to-1*** approaches to determine the best segmentation approach.

6.1. Comparing Statistics Based Methods

We compare the two statistics-based methods ***HC*** and ***EC*** across six datasets, three dependent variables per dataset, two classifiers and three performance measures per model to determine which method is better. This resulted in the total of 108 Mann-Whitney tests for this pair-wise comparison. Table 2 lists the number of statistical tests rejecting the null hypothesis (I) at 95% significance level. As Table 2 shows, only 2 out of 108 produced statistically significant differences between the ***HC*** and ***EC*** methods, in which ***HC*** dominated ***EC***.

From this comparison we can conclude that ***HC*** and ***EC*** methods provide similar performance results with ***HC*** "slightly" dominating ***EC***, i.e., ***EC*** ≲ ***HC***. Since there is a small difference between the two statistics-based segmentation methods, we could have chosen any of the two methods. We decided to choose ***HC*** as a representative statistics-based segmentation method to be compared against ***IM*** and ***1-to-1*** approaches in the following section.

TABLE 2. PERFORMANCE TESTS ACROSS ALL STATISTICS-BASED SEGMENTATION METHODS 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)

Method	***HC***	
	H+	H-
EC	0	2

6.2. Comparing the Direct Grouping, Statistics-based Segmentation and 1-to-1Methods

In this section, we compare the best methods out of the three different modeling approaches to predicting customer behavior. As stated in Sections 6.1, we selected the ***HC*** method to represent statistics-based grouping methods because it outperformed ***EC***. Therefore, we compared ***HC, IM*** and ***1-to-1*** methods across the six datasets, three dependent variables per dataset, two classifiers, and three performance measures. This resulted in the total of 108 Mann-Whitney tests per pair-wise comparison.

TABLE 3. PERFORMANCE TESTS ACROSS ALL *1-TO-1, HC,* AND *IM* 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*	
	H+	H-	H+	H-
1-to-1	76	29	21	86
HC	-	-	3	104

Table 3 summarizes the three pair-wise comparisons by listing the number of statistical tests rejecting the null hypothesis (I) at 95% significance level. As evident from the number of statistically significant test counts, ***IM*** clearly dominates ***1-to-1***, which in turn dominates ***HC***.

As demonstrated in [14], ***HC*** dominates ***1-to-1*** for the low-volume, highly idiosyncratic customers assuming that a good clustering method is used for ***HC***. Therefore, we decided to do the same type of comparison, as reported in Table 3 but just for the High-Volume customers (Table 4)

and the Low-Volume customers (Table 5), where the high-volume customers constitute customers with at least hundred transactions per household, and the low-volume customers constitute customers with just ten transactions per household for ComScore and simulated datasets and about forty transactions per household for Nielsen datasets (See Table 1).

As Tables 4 and 5 show, ***1-to-1*** still dominates ***HC*** for both high- and low-volume customers, suggesting that the statistics-based clustering is inferior to ***1-to-1*** for both the high- and the low-volume customers. We also note that ***1-to-1*** performs somewhat better against ***IM*** among high-volume datasets relative to low-volume datasets, which does make intuitive sense, as the high-volume customers would be more probable to have enough transaction data to effectively model individual customer behavior. However, ***IM*** still shows significant performance dominance against both ***1-to-1*** and ***HC*** across all the experimental conditions, including high- and low-volume customers.

TABLE 4. PERFORMANCE TESTS ACROSS ALL *1-TO-1, HC,* AND *IM* FOR HYPOTHESIS TEST (I) AMONG HIGH-VOLUME CUSTOMERS (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 54)

Methods	*HC*		*IM*	
	H+	H-	H+	H-
1-to-1	43	11	19	34
HC	-	-	1	53

To get a sense of the magnitude of the dominance that ***IM*** has over ***1-to-1***, we computed the difference between the medians of each distribution. For a particular dataset, dependent variable, classifier and performance measure, we took the two distributions of the performance measures across all the segments for the ***IM*** and all the individual customers for the ***1-to-1*** methods. Then we determined the medians of the two distributions[2] (one for ***IM*** and one for ***1-to-1***), and computed the differences between them. We repeated this process for all the 108 comparisons across the six datasets, 3 dependent variables per dataset, two classifiers, and 3 performance measures, and plotted out the histograms of the median differences for the CCI, RME, and RAE measures in Figure 2-4 respectively. Note that to plot out histograms across real values, we grouped the median differences across the distribution comparisons into bins along the X-axis, while the Y-axis represent the number of tests that falls within the median difference bin.

The negative values for the CCI measure and positive values for the RME and RAE measures in Figures 2 – 4 show that ***IM*** significantly outperforms the ***1-to-1*** method across most of the experimental conditions, thus providing additional visual evidence and the quantitative extent of the dominance of ***IM*** over ***1-to-1*** that was already statistically demonstrated with the Mann-Whitney tests.

We also did the same type of comparison for the ***HC*** and the ***IM*** methods. We show in Figure 5-7 the left skewed median difference distribution for the CCI measure and the right skewed median difference distributions for the RME and RAE measures. As in the case of ***IM*** vs. ***1-to-1***, Figures 5 – 7 clearly demonstrate ***IM***'s dominance over ***HC***.

TABLE 5. PERFORMANCE TESTS ACROSS ALL *1-TO-1, HC,* AND *IM* FOR HYPOTHESIS TEST (I) AMONG LOW-VOLUME CUSTOMERS (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 54)

Methods	***HC***		***IM***	
	H+	H-	H+	H-
1-to-1	33	18	2	52
HC	-	-	2	51

Lastly, we did the same type of comparison for the ***HC*** and the ***1-to-1*** methods, and the results are reported in Figures 8 – 10. As Figure 8 shows, ***1-to-1*** clearly dominates ***HC*** in terms of median CCI difference distributions. However, the small difference in RME error and the relatively evenly distributed RAE median difference distribution indicate that ***1-to-1*** produces approximately the same amount of errors as ***HC***. Thus, unlike the case of ***IM***'s dominance over ***HC***, the ***1-to-1*** approach does not clearly dominate ***HC*** across all the experimental conditions.

6.3. Performance Distributions of the 1-to-1, HC, and IM Methods

We can gain further insight into the issue of performance dominance by plotting percent histograms of CCI distributions across different methods and different experimental conditions. Because of the space limitation, we present only three representative examples of these 108 performance measure histograms in Figures 11 – 13.

Figure 11 shows the histogram of the CCI performance measure distribution of the Naïve Bayes models generated by ***1-to-1*** approach across 1,000 unique customer's data from the High-volume ComScore dataset. The x-axis indicated the actual CCI score from a specific NaiveBayes model trained and tested on a specific segment of customers, while the y-axis indicates the percentage of all the models having the corresponding CCI performance measure. Note how the CCI score varies from 10% to

[2] We selected the medians, rather than the means, of these performance measure distributions because these distributions tend to be highly skewed and the medians are more representative of the performance of the distributions than their averages.

100% correct, and the mean of the distribution is slightly above 50%.

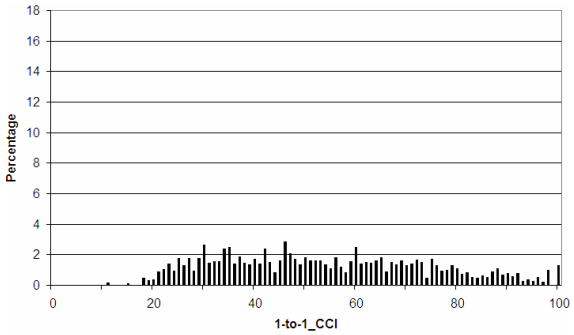


Figure 11. Example Histograms of CCI measures generated by the 1-to-1 method using NaiveBayes on the attribute "Day of the Week" for High-Volume ComScore data

Figure 12 displays the histogram of the CCI performance measure distribution of the best performing segment-level (as explained in Section 4.1) within the set of models trained on segments generated by ***HC***. Note how the CCI scores now have a tighter range, from close to 20% to 60% correct if we discount some outliers. However, the mean of the CCI distribution is significantly lower, at a little less than 30%. This illustrates our findings in the previous section, where compared to ***1-to-1***, ***HC*** has reduced variance and error, but also has a lower CCI measure. This finding is consistent with the results of the Mann-Whitney distribution comparison tests for ***1-to-1*** vs. ***HC***, as reported in Table 4.

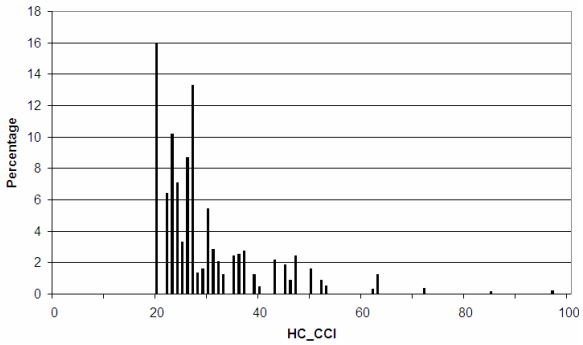


Figure 12. Example Histograms of CCI measures generated by the HC method using NaiveBayes on the attribute "Day of the Week" for High-Volume ComScore data

Figure 13 shows CCI distribution generated by the ***IM*** methods. Note that the distribution is slightly wider than that of ***HC***, ranging from 30% to 90%. However, ***IM*** has tighter variance than ***1-to-1*** and does not drop in CCI mean relative to ***1-to-1*** and definitely has a higher mean compared to ***HC***. This CCI distribution generated by ***IM*** clearly shows improved performance over the ***HC*** and ***1-to-1*** methods for the reasons demonstrated above, which is consistent with the results of the Mann-Whitney comparison tests as also reported in Table 4.

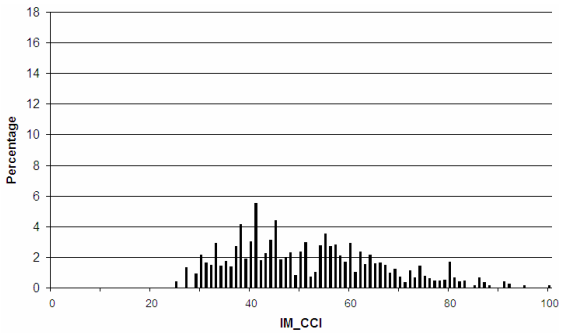


Figure 13. Example Histograms of CCI measures generated by the IM method using NaiveBayes on the attribute "Day of the Week" for High-Volume ComScore data

Again, Figures 11 – 13 provide only three examples of distributions of the CCI measure out of the total of 108 histograms. However, these examples are very typical and clearly delineate the differences between the ***IM, HC*** and ***1-to-1*** methods. Therefore, these selected CCI histograms provide additional insights into the nature of the ***IM*** dominance over the ***1-to-1*** and ***HC*** methods, as demonstrated in Tables 3 – 5 and Figures 2-7.

In summary, our empirical analysis clearly shows that, contrary to the popular belief [23], the ***1-to-1*** approach is definitely not the best solution for predicting customer behaviors. On the other hand, ***IM***, which is essentially a micro-segmentation approach to segmentation, shows clear dominance over all methods tried in our experimental settings.

However, we noted that there are some high performing size-one segments that were present in the distribution of ***1-to-1*** CCI in Figure 11, which did not get picked by the ***IM*** method as presented in Figure 13. This shows that, while the ***IM*** method is statistically dominant over ***1-to-1*** and ***HC, IM*** is still not the optimal segmentation solution described in Section 2. Nevertheless, ***IM***'s dominance over other popular segmentation methods across all the experimental settings indicates that it constitutes a sound initial approach towards reaching the final goal of generating best computationally tractable approximate solutions of the intractable optimal segmentation problem.

7. In depth analysis of IM

In this section we make a closer examination of the segments created by the ***IM*** method. Specifically, we want to study the distribution of segment sizes generated by ***IM*** and investigate ways to improve ***IM***.

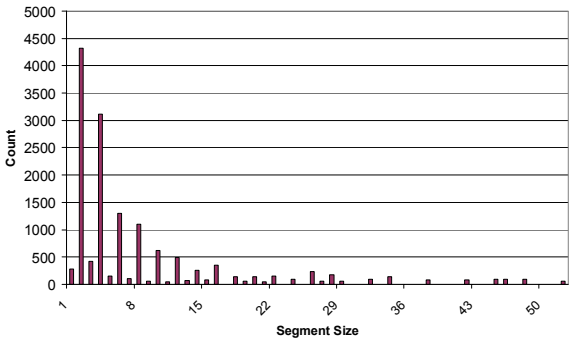


Figure 14. The distribution of segment sizes generated by IM across High-volume datasets

Figure 14 shows the distribution of segment sizes generated by ***IM*** for the high-volume customer datasets aggregated over all the experimental conditions. We note that the overall counts of segments peaks at segments of size two and then decrease steadily as the segment sizes increase. We also observe small counts among segments of odd sizes, which is an expected artifact of the ***IM*** algorithm where segment groups of roughly equal sizes were iteratively merged to improve performance. However, IM does not inherently discriminate against segments of size one's. Rather, segments will remain as size one if there are no other segment, once combined, that could improve the new combinations' overall fitness. Thus, these observations provide evidence against the ***1-to-1*** approach to personalization, as most of size-one segments do find at least one other size-one segment to merge and improve the overall performance, as evident from the spike in size-two segments in Figure 14.

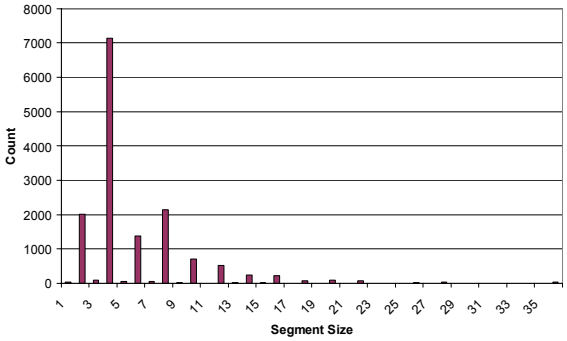


Figure 15. The distribution of segment sizes generated by IM across Low-volume datasets

Figure 15 shows the distribution of segment sizes generated by ***IM*** across low-volume customer datasets. Interestingly, the spike of the segment size distribution occurs at segments of size four. This does make intuitive sense, as low-volume customers need to form bigger groups in order to reach the "critical mass" in terms of the data necessary for building good predictive models. Taken together, both Figure 14 and 15 suggest that ***IM***'s dominance over ***1-to-1*** and ***HC*** is largely due to the formation of large numbers of small customer segments, thus adding support to the use of micro-segmentations in forming robust and effective customer behavior models.

The distribution of segment sizes generated by ***IM*** (as shown in Figures 14 and 15), clearly indicates that ***IM*** was able to find better performing groupings than just simple size-one segments. The peak at segment size of two and four implies that segments of small sizes, but sizes greater than one, are better performing segments for ***IM***. And by the definition of ***IM***, these small multi-customer segments, when modeled together, significantly outperform their respective individual segments of size one, as demonstrated from ***IM***'s dominance over ***1-to-1***.

While the ***IM*** direct grouping approach does not constitute an optimal grouping, the lack of size-one segments after many rounds of attempted segment merges implies that there will not be many size-one segments in the optimal solution. In addition, the optimal solution will definitely dominate the ***1-to-1*** solution, and we conjecture that it will contain predominantly small sized segments, resulting in a micro-segmented solution, as in the case of ***IM***. We emphasize this lack of size-one segments in the optimal solution is only a conjecture and need to be proven, as we investigate better methods to approach the optimal partition solution.

As one additional step in the analysis, we characterize the rate of decline in segment counts as segment sizes increase past the initial peaks in the distribution of segment sizes. From Figures 14 and 15, we observe that the rate of decline in segment counts follows the Zipf's distribution [29], and we formally tested and proven this conjecture as follows. Zipf's law states that

$$\log(P_n) \approx -a\log(n) \qquad (2)$$

where P_n is the frequency of occurrence of a segment of size n.

We fitted the regression model (2) against the high-volume and low-volume data to test the Zipf's law hypothesis, and it turned out that the regression model indeed fitted the data. In particular, the coefficient $\boldsymbol{a}$ in Equation (2) for the high-volume customers, the segmentation size distribution starting from segment size two has a value of $\boldsymbol{a}$=0.828, with *p-value* less than 0.001. As for low-volume customers, segmentation size distributions starting from segment size four has $\boldsymbol{a}$=1.67, with *p-value* less than 0.01. As with many natural phenomena that has a Zipf's distribution, our result suggest that the decline rate in terms of segment counts per segment size, starting from the peak of the segment size distribution, would also follow a Zipf's distribution in the optimal solution. However, formal analysis is required to prove this

conjecture.

8. Conclusions

In this paper, we examined the problem of optimal partitioning of customer bases into homogeneous segments for building better customer profiles and proposed the *direct grouping* approach as a solution. This approach partitions the customers not based on computed statistics and particular clustering algorithms, but in terms of directly combining transactional data of several customers and building a single model of customer behavior on this combined data. We formulated the optimal partitioning problem as a combinatorial optimization problem and showed that it is NP-hard. Then we proposed a suboptimal polynomial-time direct grouping method, ***IM***, and compared ***IM*** against the traditional statistics-based and ***1-to-1*** clustering approaches. We showed that ***IM*** significantly dominates the statistics-based approaches deploying standard clustering methods across all the experimental conditions examined in this paper. We also showed that, contrary to the popular beliefs, ***1-to-1*** turned out to be significantly inferior to ***IM*** across all the experimental conditions. We then examined the nature of the segments generated by ***IM*** and observed that there were very few size-one segments, that the distribution of segment sizes reached a maximum at the very small segment sizes, and that the rate of decline in the number of segments after this maximum followed a Zipf's distribution. This observation, along with the dominance of ***IM*** over ***1-to-1***, provides strong support for the *micro-segmentation* approach to personalization, where the customer base is partitioned into a large number of small segments.

As a future research, we would like to gain additional insights into the optimal customer partitioning problem, including the distribution of the segment sizes for this *optimal* partitioning (e.g., does it form the Zipf distribution?). We would also like to develop additional polynomial-time direct grouping methods that approach this optimal solution within some bounding limits and thus outperform ***IM*** and, hence, the ***1-to-1*** method. Finally, 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.

9. References

[1] Adomavicius, G. and A. Tuzhilin, *Personalization technologies: A process-oriented perspective.* Communication of the CAM, 2005.

[2] Beyer, D. and R. Ogier. *Tabu learning: a neural network search method for solving nonconvex optimization problems*. in *International Joint Conference on Neural Networks*. 1991. Singapore: IEEE and INNS.

[3] Brijs, T., et al. *Using shopping baskets to cluster supermarket shoppers*. in *AARTF*. 2001. Amelia Island Plantation, FL.

[4] Brucker, P., *On the complexity of clustering problems*. Optimization and Operations Research, ed. R. Henn, B. Korte, and W. Oettli. 1977: Springer Verlag. 45-54.

[5] CACM, *Comm of ACM*, in *Special Issue on Personalization.* 2000.

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

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

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

[9] Gomory, R.E., *Outline of an algorithm for integer solutions to linear programs.* Bulletin American Mathematical Society, 1958. **64**: p. 275-278.

[10] Guignard, M. and S. Kim, *Lagrangian decomposition: a model yielding stronger Lagrangian bounds*. Mathematical Programming, 1987. **39**: p. 215-228.

[11] Hansen, P. *The steepest ascent mildest descent heuristic for combinatorial programming*. in *congress on Numerical Methods in Combinatorial Optimization*. 1986. Capri, Italy.

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

[13] Hoffman, K., *Combinatorial Optimization: Current Successes and Directions for the Future.* Journal of Computational and Applied Mathematics, 2000. **124**: p. 341-360.

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

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

[16] Kotler, P., *Marketing Management*. 11 ed. 2003: Prentice Hall.

[17] Land, A.H. and A.G. Doig, *An automatic method for solving discrete programming problems*. Econometrica, 1960. **28**(97).

[18] Lin, S. and B.W. Kernigham, *An effective implementation for the traveling salesman problem.* Operations Research, 1973. **21**(498-516).

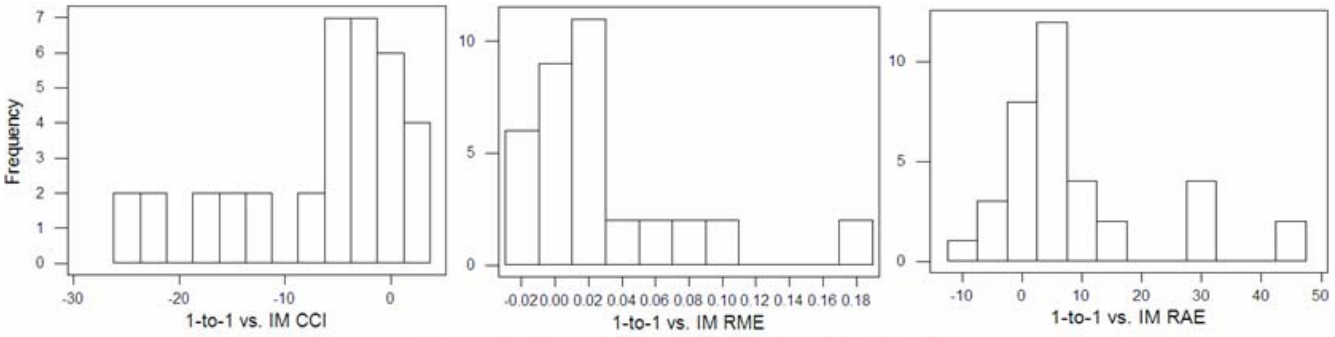
[19] Mendenhall, W. and R.J. Beaver, *Introduction to probability and statistics*. 1994: Thomson Pub.

[20] Mühlenbein, H., *Parallel genetic algorithms in combinatorial optimization*. Computer Science and Operations Research, ed. O. Blaci. 1992, New York: Pergamon Press.

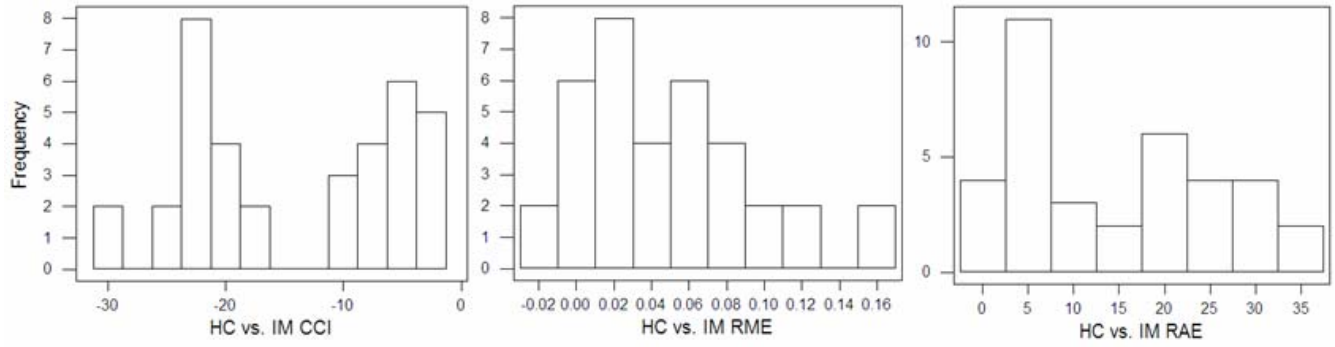
[21] Novo, J., *Drilling Down: Turning Customer Data Into Profits with a Spreadsheet*. 2004: Booklocker.com.

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

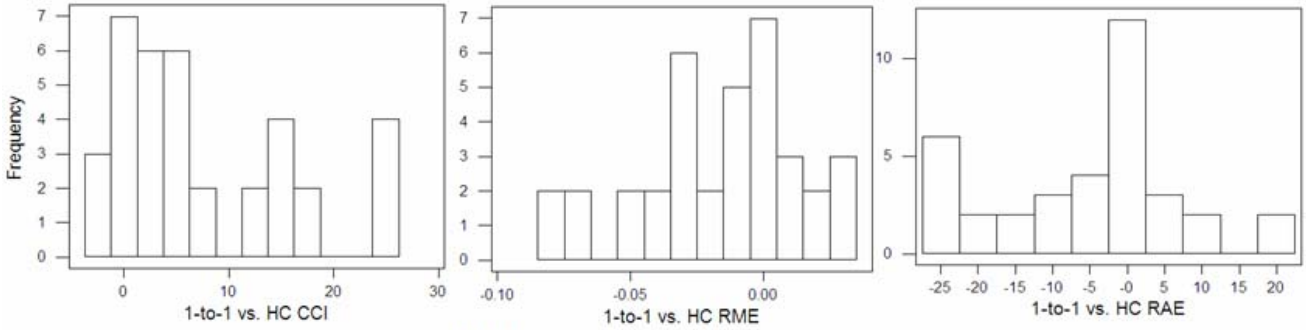
[23] Peppers, D. and M. Rogers, *Enterprise One to One*. 1997, New York: Bantam Doubleday Dell Pub. Group Inc.
[24] Quinlan, R., *C4.5: Programs for Machine Learning*. 1993.
[25] Smith, W., *Product Differentiation and Market Segmentation as Alternative Marketing Strategies*. Journal of Marketing, 1956. **21**: p. 3-8.
[26] Wedel, M. and W. Kamakura, *Market Segmentation: Conceptual and Methodological Foundations*. 2nd ed. 2000: Kluwer Publishers.
[27] Witten, I.H. and E. Frank, *Data Mining: practical machine learning tools and techniques with Java implementations*. 2000.
[28] Yang, Y. and B. Padmanabhan. *Segmenting Customer Trans. Using a Pattern-Based Clustering Approach*. in *ICDM*. 2003.
[29] Zipf, G.K., *Human Behavior and the Principle of Least Effort*. 1949, New York: Addison-Wesley.



Figures 2-4. Median Difference Distributions of 1-to-1 vs. IM



Figures 5-7. Median Difference Distributions of HC vs. IM



Figures 8-10: Median Difference Distributions of 1-to-1 vs. HC