

Improving Personalization Solutions through Optimal Segmentation of Customer Bases

Tianyi Jiang and Alexander Tuzhilin

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. This paper shows that finding an optimal customer partition is NP-hard, proposes several suboptimal direct grouping segmentation methods, and empirically compares them among themselves, traditional statistics-based hierarchical and affinity propagation-based segmentation, and one-to-one methods across multiple experimental conditions. It is shown that the best direct grouping method significantly dominates the statistics-based and one-to-one approaches across most of the experimental conditions, while still being computationally tractable. It is also shown that the distribution of the sizes of customer segments generated by the best direct grouping method follows a power law distribution and that microsegmentation provides the best approach to personalization.

Index Terms—Customer segmentation, marketing application, personalization, one-to-one 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 [40]. Its popularity comes from the fact that segmented models usually outperform aggregated models of customer behavior [45]. 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 *one-to-one* marketing [37] and personalization [8], 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 [21], it is a nontrivial 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 a distance-measure-based approach where a certain proximity or similarity measure is used to cluster customers. In this paper, we consider the *statistics-based* approach, a subclass of the distance-measure-based approaches to segmentation that computes the set of summary statistics from customer's demographic and transactional data [6], [21], [47], 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, recency, frequency, and monetary (RFM values) statistics [34], and so forth. These customer summary statistics constitute statistical reductions of the customer transactional data across the transactional variables. They are typically used for clustering in the statistics-based approach instead of the raw transactional data since the unit of analysis in forming customer segments is the customer, not his or her individual transactions. After such statistics are computed for each customer, the customer base is then partitioned into customer segments by using various clustering methods on the space of the computed statistics [21]. It was shown in [21] that the best statistics-based approaches can be effective in some situations and can even outperform the *one-to-one* case under certain conditions. However, it was also shown in [21] that this approach can also be highly ineffective in other cases. This is primarily because computing different customer statistics would result in different h -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.

• T. Jiang is with AvePoint Inc., 3 2nd Street, Suite 803, Jersey City, NJ 07302. E-mail: tjjiang@stern.nyu.edu.

• A. Tuzhilin is with the Department of Information, Operations, and Management Sciences, Stern School of Business, New York University, 44 West 4th Street, Room 8-92, New York, NY 10012. E-mail: atuzhili@stern.nyu.edu.

Manuscript received 22 May 2007; revised 14 Feb. 2008; accepted 15 July 2008; published online 30 July 2008.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-05-0230. Digital Object Identifier no. 10.1109/TKDE.2008.163.

In this paper, we present an alternative *direct grouping* segmentation approach that partitions the customers into a set of mutually exclusive and collectively exhaustive segments 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 is a more direct approach to customer segmentation because it directly groups customer data and builds a single model on this data, thus avoiding the pitfalls of the statistics-based approach, which selects arbitrary statistics and groups customers based on these statistics.

We also discuss how 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 several suboptimal polynomial-time direct grouping methods and compare them in terms of the modeling quality and computational complexity. As a result, we select the best of these methods, called *Iterative Merge (IM)*, and compare it to the standard statistics-based hierarchical segmentation methods [13], [21], to a modified version of the statistics-based *affinity propagation (AP)* clustering algorithm [15], and to the *one-to-one* approach. We show that *IM* significantly dominates all other methods across a broad range of experimental conditions and different data sets considered in our study, thus demonstrating superiority of the direct grouping methods to profiling online customers.

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 power law distribution. This observation, along with the dominance of *IM* over the *one-to-one* method, provides support for the *microsegmentation* approach to personalization [24], where the customer base is partitioned into a large number of small segments.

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 several suboptimal direct grouping methods, empirically compare them, and show that the direct grouping method *IM* significantly dominates others.
- Adapt a previously proposed *AP* clustering algorithm [15] to the statistics-based customer segmentation problem and show that it dominates several existing statistics-based hierarchical clustering (HC) methods.
- Compare *IM* against the statistics-based hierarchical segmentation, *AP*, and the *one-to-one* approaches and demonstrate that *IM* significantly dominates them. This is an important result because it demonstrates

that, in order to segment the customers, it is better to first partition customer data and then build predictive models from the partitioned data rather than first compute some arbitrary statistics, cluster the resulting h -dimensional data points into segments, and only then build predictive models on these segments.

- Plot the frequency distribution of the sizes of customer segments generated by the *IM* method and show that the tail of this plot follows a power law distribution. This means that *IM* generates many small customer segments and only few large segments. This result provides additional support for the microsegmentation approach to personalization.

The rest of this paper is organized as follows: In Section 2, we present our formulation of the optimal direct grouping problem. We review the current literature on related work in Section 3. In Section 4, we discuss our implementations of two statistics-based segmentation methods, our adaptation of *AP* as a statistics-based segmentation method, *one-to-one* segmentation, and three suboptimal direct grouping methods. In Section 5, we lay out our experimental setup and discuss the results in Sections 6. In Section 7, we provide more detailed analysis.

2 PROBLEM FORMULATION

The problem of optimal segmentation of a customer base into a set of mutually exclusive and collectively exhaustive set of customer segments 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 $A = \{A_1, A_2, \dots, A_m\}$, k_i transactions $Trans(C_i) = \{TR_{i1}, TR_{i2}, \dots, TR_{ik_i}\}$ performed by customer C_i , and h summary statistics $S_i = \{S_{i1}, S_{i2}, \dots, S_{ih}\}$ computed from the transactional data $Trans(C_i)$ that are explained below. Each transaction TR_{ij} is defined by a set of transactional attributes $T = \{T_1, T_2, \dots, T_p\}$. The number of transactions k_i per customer C_i varies. Finally, we combine the demographic attributes $\{A_{i1}, A_{i2}, \dots, A_{im}\}$ of customer C_i and his/her set of transactions $\{TR_{i1}, TR_{i2}, \dots, TR_{ik_i}\}$ into the complete set of customers' data $TA(C_i) = \{A_{i1}, A_{i2}, \dots, A_{im}, TR_{i1}, TR_{i2}, \dots, TR_{ik_i}\}$, which constitutes a unit of analysis in our work. As an example, assume that customer C_i can be defined by attributes $A = \{\text{Name, Age, Income, and other demographic attributes}\}$, and by the set of purchasing transactions $Trans(C_i)$ she made at a website, where each transaction is defined by such transactional attributes T as an item being purchased, when it was purchased, and the price of the item.

Summary statistics $S_i = \{S_{i1}, S_{i2}, \dots, S_{ih}\}$ for customer C_i is a vector in an h -dimensional euclidean space, where each statistics S_{ij} is computed from the transactions $Trans(C_i) = \{TR_{i1}, TR_{i2}, \dots, TR_{ik_i}\}$ of customer C_i using various statistical aggregation and moment functions, such as mean, average, maximum, minimum, variance, and other statistical functions. For instance, in our previous example, a summary statistics vector S_i can be computed across all of the customer C_i 's purchasing sessions and can include such statistics as the average amount of purchases per a Web

session, the average number of items bought, and the average time spent per online purchase session. This means, among other things, that each customer C_i has a unique summary statistics vector S_i and that a customer is represented with a unique point in the euclidean space of summary statistics.¹

Given the set of N customers C_1, \dots, C_N , and their respective customer data $p_\alpha = \{TA(C_1), \dots, TA(C_N)\}$, we want to build a single predictive model M_α on this group of customers p_α :

$$Y = \hat{f}(X_1, X_2, \dots, X_p), \quad (1)$$

where dependent variable Y constitutes one of the transactional attributes T_j and independent variables X_1, X_2, \dots, X_p are all the transactional and demographic variables, except variable T_j , i.e., they form the set $T \cup A - T_j$. The performance of model M_α can be measured using some *fitness function* f mapping the data of this group of customers p_α into reals, i.e., $f(p_\alpha) \in \mathbb{R}$.

For example, model M_α can be a decision tree built on data p_α of customers C_1, \dots, C_N , for the purpose of predicting T_j variable “time of purchase” using all the transactional and demographic variables, except variable T_j (i.e., $T \cup A - T_j$) as independent variables. The fitness function f of model M_α can be its predictive accuracy on the out-of-sample data or computed using 10-fold cross validation.

The fitness function f can be quite complex in general, as it represents the predictive power of an arbitrary predictive model M_α trained on all customer data contained in p_α . For example, f could be the relative absolute error (RAE) of a neural network model trained and tested on p_α via 10-fold cross validation for the task of predicting time of the day of a future customer purchase. Given that fitness function f is a performance measure of an arbitrary predictive model, 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 .

We next describe the problem of partitioning the customer base C into a mutually exclusive collectively exhaustive set of segments $P = \{p_1, \dots, p_\beta\}$ and building models M_α of the form (1) for each segment p_α , as described above. Clearly, customers can be partitioned into segments in many different ways, and we are interested in an optimal partitioning that can be defined as follows:

Optimal customer segmentation (OCS) problem. Given the customer base C of N customers, we want to partition it into the disjoint groups $P = \{p_1, \dots, p_\beta\}$, such that the models M_α built on each group p_α would collectively produce the best performance for the fitness function $f(p_\alpha)$ taken over p_1, \dots, p_β . Formally, this problem can be formulated as follows: Let θ_α be a weighting measure specifying “importance” of segment α . Some examples of θ_α include simple average $1/\beta$ and proportional weights $|TA(p_\alpha)|/|TA(C)|$. Then, we want to find a partition of the customer base C into the set of mutually exclusive collectively exhaustive segments $P = \{p_1, \dots, p_\beta\}$, where

segment p_α is defined by its customer data $p_\alpha = \{TA(C_j), \dots, TA(C_m)\}$, such that the following fitness score

$$\varphi = \sum_{\alpha=1}^{\beta} \theta_\alpha * f(p_\alpha)$$

is maximized (or minimized if we used error rates as our fitness score) over all possible partitions $P = \{p_1, \dots, p_\beta\}$, such that $p_\alpha \cap p_\gamma = \emptyset$ and $\cup_\alpha p_\alpha = P$. Note that different predictive tasks (i.e., predicting different transactional variables) result in different optimal partitions of the customer base C .

Clearly, the OCS is a *combinatorial partition problem* [20], and it is computationally hard, as the following proposition shows.

Proposition. *OCS problem is NP-hard.*

Sketch of Proof. This result can be obtained by reducing the clustering problem, which is NP-hard [7], to the OCS problem. \square

Since the OCS problem is NP-hard, we propose some suboptimal polynomial time methods that provide reasonable performance results. More specifically, we consider the following three types of customer segmentation methods as approximations to the solution of the OCS problem:

- *Statistics based*—This traditional customer segmentation approach groups customers by first computing some statistics $S_i = \{S_{i1}, S_{i2}, \dots, S_{ih}\}$ for customer C_i from that customer’s demographic and transactional data, considers these statistics as unique points in an h -dimensional space, groups customers into segments $P = \{p_1, \dots, p_\beta\}$ by applying various clustering algorithms to these h -dimensional points, and then builds models M_α of the form (1) for each segment p_α , as described above. Note that this method constitutes a subset of the distance-measure-based segmentation approach described in Section 3 below.
- *One-to-one*—The basis for one-to-one approach is that each individual customer exhibits idiosyncratic behavior and that the best predictive models of customer behavior are learned from the data pertaining only to a particular customer. Rather than building customer segments of various sizes, the one-to-one approach builds customer segments of size 1, which is just the customer him/herself.
- *Direct grouping*—Unlike the traditional statistics-based approach to segmentation, the direct grouping approach directly groups customer data into segments for a given predictive task via combinatorial partitioning of the customer base (without clustering them into segments using statistics-based or any other distance-based clustering methods). Then, it builds a *single* predictive model (1) on this customer data for each segment and uses the aforementioned fitness score to derive the decisions on how to do the combinatorial partitioning of the customer base. The direct grouping method avoids the pitfalls of the statistics-based approach, which selects and computes an arbitrary set of statistics out

1. This observation will be extensively used later when hierarchical clustering methods are introduced in Section 4 and beyond.

of a potentially infinite set of choices. Since segmentation results critically depend on a good choice of these statistics, the statistics-based approach is sensitive to this somewhat arbitrary and highly nontrivial selection process, which the direct grouping approach avoids entirely. Instead of looking for an optimal grouping of customers, which was shown above to be NP-hard, we will present three polynomial-time suboptimal direct grouping methods in this paper.

Before we describe these methods in more detail, we first present some of the related work.

3 RELATED WORK

The problem of finding the global optimal partition of customers is related to the work on 1) combinatorial optimization problems in operations research, 2) customer segmentation in marketing, and 3) data mining and user-modeling research on customer segmentation. We examine the relationship of our work to these three areas of research in this section.

3.1 Combinatorial Optimization

Combinatorial optimization models are used across a wide range of applications. Combinatorial optimization problems are in general NP-hard; however, depending on the mathematical formulation of a particular problem, near-optimal approximations to exact solutions can be achieved in polynomial time [20]. Despite recent advances in finding solutions to various combinatorial optimization problems, there is still a large set of problems considered too complex to derive optimal or near-optimal solutions [20]. Therefore, various heuristics were explored in obtaining good solutions that have no guarantees as to their “closeness” to the optimal solution, including greedy hill climbing [27], simulated annealing [18], evolutionary algorithms [32], and neural networks [4].

Our OCS problem falls in this category of “hard” problems because we cannot use any existing problem formulations to solve it due to the dependence of the fitness function f on data in p_α in a complicated manner that precludes various useful characterizations of the fitness function, such as having additivity, monotonicity, and other “nice” properties. Therefore, in our work we do not explore any of the near-optimal solutions and deploy the greedy hill-climbing approach instead in conjunction with a branch-and-bound enumerative technique in developing our fitness function-based methods.

3.2 Customer Segmentation in Marketing

Customer segmentation has been extensively studied by marketers since the time Smith introduced the concept of segmentation back in 1956 [40]. Its popularity comes from the fact that segmented models usually outperform aggregated models of customer behavior [45]. In particular, marketers classify various segmentation techniques into a priori versus post hoc and *descriptive* versus *predictive* methods giving rise to a 2×2 classification matrix of these techniques [45]. 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 mixing distributions.

Clustering methods are classified into nonoverlapping methods, when customer can belong to only one segment, overlapping methods, when customer can belong to more than one segment, and fuzzy methods, when customers can belong to different segments with certain probabilities [45]. From this perspective, we assume the nonoverlapping model in our clustering approaches in which customers belong *only to one* segment.

Much segmentation work in marketing pertains to the mixture models where observation is a random variable that follows a probability density function conditional on the observation belonging to segment s . Also, DeSarbo and Cron [11] present a mixture regression model and Wedel and DeSarbo [43] present the generalized mixture regression model that add to these mixture model the fact that the means of the observations in each segment depend in the form of a linear regression on a set of explanatory variables. This means that each segment is defined by its own regression model with the set of parameters unique for that segment.

Furthermore, Allenby and Rossi [3], Wedel et al. [44], and others extended these segmentation-based mixture models to the continuous distribution models where each customer has a set of density distribution parameters *unique* for that customer rather than for the segment to which the customer belongs. Subsequently, Allenby and Rossi [3] used hierarchical Bayesian methods to model these continuous distributions and MCMC simulation methods to estimate unique parameters for each customer. It was shown empirically in [3] that this approach outperformed finite mixture segmentation models, thus providing evidence to the advantages of individual customer modeling.

Our work differs from this previous research in marketing in that we have proposed the direct grouping approach to segmentation that is based on the combinatorial partitioning of the customer base rather than on finite mixture, continuous distribution, or clustering methods used by marketers. Although Spath [41] describes a clusterwise linear regression approach that form a partition of length K and corresponding sets of parameters $b_{k,r}$, such that the sum of the error sums of squares computed over all clusters is minimized, seems similar to our *direct grouping* approach, it is actually very different in that it considers each customer as a single observation point to be clustered with other customers. In contrast, in our *direct grouping* approach, each customer is represented by a group of observation points that is his/her own set of purchase transactions.

In this paper, we compare the direct grouping approach with the previously studied statistics-based and *one-to-one* segmentation methods and demonstrate that the best grouping-based segmentation method outperforms the one-to-one and the traditional clustering methods, and that the segment size distribution generated by this method follows a power law distribution, which is a linear distribution on a log-log plot that are used to describe naturally occurring phenomena such as word frequencies, income, city sizes, and so forth. In our case, our results show that large segment sizes are rare, whereas small segment sizes are very common.

3.3 Customer Segmentation in Data Mining

More recently, there has been much interest in the data mining and user modeling communities in learning *individual* models of customer behavior within the context of personalization [2], [8], when models of customer behavior are learned from the data pertaining only to a particular customer. One stream of research in user modeling pertains to building user profiles [1], [35], [36]. For example, in [36], a user profile is defined as a vector of weights for a set of certain keywords. Moreover, customer profiles can be defined not only as sets of attributes but also as sets of rules defining behavior of the customer [1], sets of sequences such as sequences of Web browsing activities [17], [31], [42], and signatures used to capture the evolving behavior learned from data streams of transactions [10]. There has also been some work done on modeling personalized customer behavior by building appropriate probabilistic models of customers. For example, Cadez et al. [9] build customer profiles using finite mixture models and Manavoglu et al. [29] use maximum entropy and Markov mixture models for generating probabilistic models of customer behavior. Finally, Nasraoui et al. [33] describe how to mine evolving user profiles and use them in recommender systems in order to build relationships with the customers.

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 [5], [6], [16], [21], [25], [26], [28], [39], [47]. These clustering methods use distance-measure-based metrics to compute “distances” between individual customers and group customers into clusters based on these distances. These distances between customers can be defined in various ways. For example, Guha et al. [16] describe an HC approach, ROCK, which clusters customers based on the distance between them, where the distance between two customers is defined in terms of how “similar” the two sets of their transactions are. Thus, Guha et al. [16] reduce the customer segmentation problem to finding clusters in weighted graphs and use an HC method for finding the segments. Boztug and Reutterer [5] and Reutterer et al. [39] address the problem of segmenting customers having multiple shopping basket transactions. This segmentation is achieved by, first, creating K shopping basket prototypes from the set of all shopping transactions from all customers using a variant of the K -means clustering method. Then, individual customers are assigned to the closest basket prototype based on a certain distance measure between the set of customer’s transactions and the prototype. This method is highly scalable but assumes a certain fixed number of prototype baskets in defining potential customer segments. Malthouse [28] introduces the concept of subsegmentation, where customers from the same market segment are further partitioned into subsegments using such criteria as RFM values. Then, customers in the same subsegment are targeted with customized offerings. Furthermore, Koyuturk et al. [25] and Leisch [26] present certain clustering approaches based on a variant of a Semidiscrete Decomposition (SDD) and on the k -centroid method, respectively, which are particularly well suited for clustering high-dimensional data sets and can also be used for segmenting customer bases.

Another way to segment the customers using distance-measure-based clustering methods is to compute a set of statistics from the customer’s transactional and demographic data, map each customer into an h -dimensional space, and use standard clustering methods to segment the customers. In Section 1, we called it the statistics-based approach, and it was also deployed in [6], [21], and [47]. In particular, Brijs et al. [6] compute statistics from customer’s transactional and demographic data and use clustering methods to partition the customers into groups. Also, Yang and Padmanabhan [47] use pattern-based and association mining methods to cluster customers’ transactions. Both Brijs et al. [6] and Yang and Padmanabhan [47] cluster customers into groups but do not build customer models on these groups. In contrast to this, Jiang and Tuzhilin [21] cluster customers into groups using HC methods and build predictive models on individual groups. One of the problems with the statistics-based methods from [6], [21], and [47] is that they deploy somewhat arbitrary set of statistics and that it is difficult to construct an objective and truly superior measure of intracluster similarity and intercluster dissimilarity for the segmentation problems considered in this paper. Therefore, it is not surprising that it was shown in [21] that the statistics-based approaches produced mixed performance results.

In summary, the personalization approaches developed in the data mining and user-modeling communities focus on the task of building good profiles and models of customers but do not consider direct grouping methods, do not study optimal or suboptimal customer segmentation strategies, and do not compare direct grouping against one-to-one and statistics-based segmentation methods. In this paper, we focus on the direct grouping methods and empirically compare their performance against the statistics-based and one-to-one approaches.

4 SEGMENTATION APPROACHES TO BUILDING PREDICTIVE MODELS OF CUSTOMER BEHAVIOR

In this section, we describe the details of the statistics-based, one-to-one, and direct grouping approaches before empirically comparing them across various experimental settings. We would like to reiterate that the statistics-based and the direct grouping methods only partition the customers into different segments. Once the segments are formed, we next build predictive models of type (1) on these segments and measure the overall performance of these models across the segments using the fitness function φ .

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 [13] and [19] and deployed in [21] and [35] and an adaptation of a previously proposed statistics-based method AP [15] to our segmentation problem.

HC: Using the same top-down *HC* techniques as in [21], we learn predictive models of customer behavior of the form (1) defined in Section 2, i.e.,

$$Y = \hat{f}(X_1, X_2, \dots, X_p),$$

where X_1, X_2, \dots, 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 website purchasing the product. The correctness measure of this prediction is our fitness function f (as defined in Section 2).

In the context of HC , predictive models M_α of the form (1) are learned on the groups of customers p_α , which are formed as follows. First, each customer C_i is mapped into the space of summary statistics $S_i = \{S_{i1}, S_{i2}, \dots, S_{ih}\}$ that are computed from the transactional data $Trans(C_i)$ of customer C_i as explained in Section 2. Therefore, each customer C_i becomes a single vector point in an h -dimensional euclidean space, and N customers in the customer base C result in N points in this space. Given these N points in the h -dimensional summary statistics space S_i , the customer segments are formed by applying unsupervised clustering methods to that space. The distance between two points (customers) is defined as a euclidean distance between the two vectors. Further, we use distance-based HC method that starts with a single segment containing all the N customers in the customer base C and generates new levels of segment hierarchy via progressively smaller groupings of customers until the individual customer (one-to-one) level is reached containing only single customers. In our particular implementation of HC , we have chosen FarthestFirst [19], a greedy k -center unsupervised clustering algorithm that is found to perform well in [21] on customer summary statistics and demographics attributes $\{A_1, A_2, \dots, A_m, S_1, S_2, \dots, S_h\}$. FarthestFirst generates progressively smaller customer segments for each level of the segmentation hierarchy starting from the single segment C and going down to the one-to-one level. Moreover, for each level L of the segmentation hierarchy, FarthestFirst computes the weighted sum of fitness scores, as described in Section 2.

Once the HC method computes customer segments p_α for each level of the segmentation hierarchy, predictive models M_α of the form (1) are learned on these segments p_α , as described above. More specifically, we build predictive models M_α on each customer segment p_α for a selective set of dependent variables as described in Section 2 and collect the fitness scores of these models. Then, the segmentation level with the highest overall fitness score (besides the one-to-one level) is selected as the best possible segmentation of the customer base. Note that while the segments remain fixed after running the HC algorithm, different models are built for different prediction tasks to make predictions of specific dependent variable, different set of fitness score are computed for these prediction tasks, and different best segmentation levels are selected. As we will discuss in Section 5, we conduct different predictive tasks across different experimental condition in order to ensure the robustness of our findings and that our results are not due to idiosyncratic behavior of one specific transaction variable, prediction task, or data set.

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 $\{A_1, A_2, \dots, A_m, S_1, S_2, \dots, S_h\}$, 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. Thus, for different predictive task, where the dependent variable varies, we generate different decision trees where each end node of the tree constitutes a customer segment. Once the C4.5 forms the groupings based on the principle of class label entropy minimization, we store the fitness scores (e.g., percent correctly classified, RAE, area under ROC curve, and so forth) generated by \hat{f} in (1) across these different groupings of customer transaction data for comparison against other segmentation methods. 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 subclusters, 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. Note that for different predictive tasks, because we use different dependent variables as the class label, different segments are generated from different decision trees.

AP. Starting with n unique customers, AP [15] identifies a set of training points, *exemplars*, as cluster centers by recursively propagating "affinity messages" among training points. Similar to the prototype-generating greedy K -medoids algorithms [23], 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, and forms clusters by assigning an individual exemplar's group membership based on "affinities" that exemplar has with any possible cluster centers. Affinity in our study is defined as pairwise euclidean distance measures an exemplar has with any possible cluster centers. Like other statistics-based methods, we also use summary statistics vectors to represent customers. This reduces the complexity of affinity calculation to that of distances between two customer's summary statistics vectors. AP runs in $O(n^2)$. It is a good method for segmenting customers because cluster centers are associated with real customers rather than "virtual" computed customers as in the case of standard clustering algorithms. Again, similar to EC , AP generates different set of customer segments for different predictive tasks as the summary statistics of the dependent variable are not used for clustering.

4.2 One-to-One Method

As explained in Section 2, the one-to-one approach builds predictive models of customer behavior only from individual customer's transactional data. In other words, for each predictive task (e.g., making prediction on a specific transac-

```

1. Let  $W = \{C_1, C_2, \dots, C_N\}$  // FIFO queue
2. CustomerGroupSet  $P = \{\}$  // new set of customer groups
3. CustomerGroup  $A = \text{new CustomerGroup}()$ 
4. While  $W \neq 0$  {
5.    $C_i = W.\text{pop}()$ 
6.   if A processed  $C_i$  before then
7.      $P = \{P \cup A\}$ ;  $A = \text{new CustomerGroup}()$ ;
8.   else {
9.     if  $f(TA(C_i), A) \geq f(A)$  then {
10.       $A = \{A \cup TA(C_i)\}$ 
11.    } else {
12.       $C_s = C_k$  that yields minimum  $f(\{TA(C_i), A\} - TA(C_k)) \forall C_k \in A$ ;
13.      if  $f(\{TA(C_i), A\} - TA(C_s)) \geq f(A)$  then {
14.         $W.\text{push}(C_s)$ 
15.      } else {
16.         $W.\text{push}(C_i)$ 
17.      }
18.    }
19.  }
20. }
21. Return P

```

Fig. 1. *IG* algorithm.

tional variable), we build a predictive model (1) for *each* customer C_i , $i = 1, \dots, N$, using only the demographic and the transactional data of that customer minus the specific transactional variable used as dependent variable in the predictive task. Since this is individual level modeling, we do not have to deal with customer grouping at all in this case.² For each model of customer C_i , we compute fitness function $f(C_i)$ (e.g., using 10-fold cross validation with a Naïve Bayes classifier on $TA(C_i)$) and obtain the entire distribution of these fitness scores (e.g., RAE or the area under the ROC curve) for $i = 1, \dots, N$. These average predictive performance score distributions are then stored for later distribution comparisons against scores generated by predictive models evaluating segments formed by other segmentation methods.

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 (as shown in Section 2), we propose three suboptimal grouping methods Iterative Growth (*IG*), Iterative Reduction (*IR*), and *IM* that we describe in this section.

IG. We propose the following greedy *IG* approach that starts from a single customer and grows the segment one customer at a time by adding the “best” and removing the

“weakest” customer, if any, at a time until all the nonsegmented customers have been examined. More specifically, *IG* iteratively grows individual customer groups by randomly picking one customer to start a customer group and, then, tries to add one new customer at a time by examining all customers that have not been assigned a group yet. If augmenting a new customer to a group improves the fitness score of the group, then *IG* examines all existing customers in the group and decides whether or not to exclude one “weakest” customer member to improve the overall group fitness. As a result, only customers not lowering the performance of the group will be added, and the worst performing customer, if any, will be removed from the group, where performance is defined in terms of the fitness function described in Section 2.

Unlike the previous statistics-based methods, where the fitness score distributions stored are from the result of 10-fold cross validations, in direct grouping methods, because the fitness score are used to determine the formation of groupings, we use a separate holdout set to test the predictive models built on the finalized customer segments. This out of sample testing alleviates the problem of overfitting. Also note that as in the case of *EC* and *AP*, for different predictive tasks, where different transaction variables are used as the dependent variable, we generate different set of segments by optimizing different predictive performance objectives specific for that predictive task at hand. The specifics of the *IG* algorithm are presented in Fig. 1.

It is easy to see that *IG* runs in $O(n^3)$ because for the group of i customers it takes $n - i$ choices to add the best new customer to the group and i choices to remove the weakest customer from it, resulting in $(n - i)i$ iterations for each step of the outer loop in Fig. 1.

While this approach does not exhaustively explore all possible combinations of customer partitions, it does focus on meaningful partitions via a branch and bound heuristic.

2. One of the consequences of building predictive models for individual customers is the necessity to have enough transactional data for that customer. For example, if we want to do 10-fold cross validation to compute predictive performance of such a model, a customer should have at least 10 transactions. In case a customer has too few transactions (e.g., less than 10 in our experiments), we group that customer with a couple of other similar customers in order to have a critical mass of transactions (e.g., at least 10 in our experiments).

```

1. Let  $W = \{C_1, C_2, \dots, C_N\}$ 
2. CustomerGroupSet  $P = \{\}$  // new set of customer groups
3. While  $W \neq \emptyset$  {
4.   CustomerGroup  $A = \{TA(W)\}$ 
5.    $W = \{\}$ ;
6.    $\forall C_i \in A$  {
7.     if  $f(\{A\} - TA(C_i)) > f(A)$  then {
8.        $A = \{A\} - TA(C_i)$ ;
9.        $W = \{W \cup C_i\}$ 
8.     }
9.   }
10.   $P = \{P \cup A\}$ ;
11. }
12. Return  $P$ 

```

Fig. 2. *IR* algorithm.

The drawback of *IG* is that any particular customer group's fitness score may not improve until it grows to a certain "critical mass" of a sufficient number of customer purchasing transactions exhibiting "similar" purchase behavior to train a good predictive model. In addition, once a growing customer group passes a certain "mass," any single customer's transaction may not impact the overall performance of the group enough to be considered for rejection or exclusion, which could result in large customer segments having mixed performance results.

IR. To address the "critical mass" problem of *IG*, we propose a top-down *IR* approach where we start with a single group containing all the customers and eliminate the weakest performing customer one at a time until no more performance improvements are possible. Once *IR* finishes this customer elimination process, it will form a segment out of the remaining customers, group all the eliminated customers together into one residual group, and try to reduce it using the same process. The underlying assumption in *IR* is that there exists a fixed number of ideal customer groupings and that we can narrow down to these groupings by removing one member customer at a time from an aggregate group based on the fitness scores. *IR* is greedy in that once it reduces the initial customer group to a

smallest best performing group, it will remove that group from further considerations. The specifics of the *IR* algorithm are presented in Fig. 2.

IR runs in $O(n^2)$ because, in the worst-case scenario, the outer loop runs n times, and each time the inner loop examines each of the remaining i customers for exclusion. As in the case of *IG*, we observed from our empirical results presented in Section 5 that *IR* is also prone to producing large suboptimal customer segments, as the removal of a single customer's transactions may not significantly affect the performance of large customer segments.

IM. Rather than adding or removing a single customer's transactions at a time, *IM* seeks to merge two existing customer groups at a time. Starting with segments containing individual customers, this method combines two customer segments Seg_A and Seg_B , 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 customer groups and merge them together resulting in the best merging combination. The specifics of the *IM* algorithm are presented in Fig. 3.

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. In comparison to *IG*, the search space of *IM* is smaller because it merges groups, not individual customers, at a time, and the results reported in Section 6 confirm this observation.

Finally, unlike *IG* and *IR*, close examination of our empirical results in Section 5 show that *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 merged segments, thus lessen the chance of building large and poorly performing customer segments. This observation provides an intuitive explanation why *IM* outperforms *IG* and *IR* methods, as reported in Section 6.

5 EXPERIMENTAL SETUP

To compare the relative performance of direct grouping, statistics-based, and one-to-one approaches, we conduct

```

1. Let  $W = \{C_1, C_2, \dots, C_N\}$  // FIFO queue
2. CustomerGroupSet  $P = \{\}$  // new set of customer groups
3. While  $P$  is changing {
4.   While  $W \neq \emptyset$  {
5.     CustomerGroup  $CG_i = W.pop()$ 
6.     CustomerGroup  $A = new\ CustomerGroup(TA(CG_i))$ 
7.      $CG_s = CG_k$  that yields maximum  $f(A + TA(CG_k)) \forall CG_k \in W$ ;
8.     if  $(f(A + TA(CG_k)) \geq f(A))$  {
9.        $W = \{W - CG_s\}$ ;  $A = \{A \cup TA(CG_s)\}$ ;
9.        $P = \{P \cup A\}$ ;
8.     }
9.   }
10.   $W = \{\text{all } CG\text{'s in } P\}$ ;  $P = \{\}$ 
11. }
12. Return  $P$ 

```

Fig. 3. *IM* algorithm.

TABLE 1
Customer Types and Transaction Counts

DataSet	Customer Type	% of Total Population	Families	Total Transactions	Average Transactions Per Household
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

pairwise performance comparisons using a variant of the nonparametric Mann-Whitney rank test [30] to test whether the fitness score distributions of two different methods are statistically different from each other. We conduct these performance comparisons across various direct grouping and statistics-based approaches that include *IG*, *IR*, *IM*, *HC*, *EC*, and *AP* methods. To ensure robustness of our findings, we set up the pairwise comparisons across the following four dimensions.

Types of data sets. In our study, we worked with the following data sets.

- a. Two “real-world” marketing data sets containing panel data³ of online browsing and purchasing activities of website visitors and panel data on beverage purchasing activities of “brick-and-mortar” stores. The first data set contains ComScore data from Media Metrix on Internet browsing and buying behaviors of 100,000 users across the US for a period of six months (available via Wharton’s RDS at <http://wrds.wharton.upenn.edu/>). The second data set contains Nielsen panelist data on beverage shopping behaviors of 1,566 families for a period of one year. The ComScore and Nielsen marketing data sets are very different in terms of the type of purchase transactions (Internet versus 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 data set covers a much wider range of products and demographics and is more representative of today’s large marketing data sets.

We further split these two real-world data sets into four data sets of ComScore high- and low-volume customers, which represent the top and bottom 2,230 customers in terms of transaction frequencies, respectively. Similarly, Nielsen high- and low-volume customer data sets were generated using the top and bottom 156 customers in terms of transaction frequencies, respectively.

3. Panel data [24], also called longitudinal or cross-sectional time series data, when used in the context of marketing means that the data about a preselected group of consumers on whom a comprehensive set of demographic information is collected is also augmented with the complete set of their purchases. Therefore, these panel data provide a comprehensive view of purchasing activities of a preselected panel of consumers.

- b. Two simulated data sets representing high-volume customers (*Syn-High*) and low-volume customers (*Syn-Low*), respectively, where within each data set, customer differences are defined by generating different customer summary statistics vector S_i for each customer i . All subsequent customer purchase data are generated from the set of summary statistics vectors S_i . These two data sets 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 data set is used to better simulate real-world transactional data sets.

Since for the ComScore and Nielsen we consider two data sets (each having high- and low-volume customers), this means that *we use six data sets in total* in our studies. Some of the main characteristics of these six data sets are presented in Table 1. In particular, CustomerType column specifies the transaction frequency of these data sets, High means that customers perform many transactions on average, while Low means only few transactions per customer. The columns “Percentage 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 data sets.

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 [46]: C4.5 decision tree [38] and Naïve Bayes [22]. These are chosen because they represent popular and fast-to-generate classifiers, which is a crucial criterion for us, given that we have built millions of models in making segment merging decisions and stored 105,626 models for the final holdout set evaluation in our study.

Dependent variables. We built various models to make predictions on different transactional variables, TR_{ij} , to ensure the robustness of our findings, and we also compare the 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 data sets, and category of drinks bought, total price, and day of the week in the Nielsen data sets. Note

that, while we did not use all the available transactional variables for testing due to time constraints, we feel that the selected six transactional variables used represent enough of a variation to ensure robust results. Also, for any given predictive task where we use T_{ij} as the dependent variable, the data we used to train any one model are customer C_i 's independent variables X_1, X_2, \dots, X_p , except TR_{ij} .

Note that for every transactional variable for which we want to make predictions using the direct grouping approach, we generate a new set of segments with various segmentation methods, build models on the final segments, and collect the performance measures produced by those predictive models. Also, we create a training set by randomly drawing out 90 percent of our data in building the segments and then evaluate the final quality of the segments generated by the direct grouping methods on the remaining 10 percent of our data as the separate holdout set in order to avoid overfitting.

Performance measures. We use the following performance measures: percentage of correctly classified instances (CCIs), root mean squared error (RME), and RAE [46].

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 that we use only in *direct grouping* methods to make segment merging or splitting decisions during the iterative search for the best performing segments. For example, if we add a new customer to a segment in the *IG* approach, we check if the expanded segment is better than the initial segment, where "better" is defined in terms of the conjunctive expression specified above.

To compare predictive performance of various methods against *HC*, we select the best performing segment level generated by *HC* for CCI, RME, and RAE distributions separately based on the most right skewed CCI distribution and left skewed RME and RAE distributions. To compare the performances of different segmentation methods, such as *HC*, *IM*, and so forth, we take distributions of identical performance measures (e.g., CCI) across all the segments generated by these methods and statistically test the null hypothesis that the two distributions are not equal. The specifics of this comparison process are discussed further in Section 6.

In terms of data preprocessing, we discretized our data sets to improve classification speed and performance [12]. 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 [14] recursive minimal entropy partitioning algorithm.

As was stated before, the goal of this paper is to determine the best performing segmentation method. To reach this goal, we first compared all three methods *IG*, *IR*, and *IM* in order to determine the best *direct grouping* approach. Due to computational expenses of the proposed direct grouping methods, we have done it for all six data sets and all three performance measures, but only for one dependent variable and using

only one classifier (C4.5). We then compared statistics-based segmentation methods *HC*, *EC*, and *AP* across multiple dependent variables, classifiers, and all six data sets to select the best one. After that, we conducted additional experiments comparing the best performing direct grouping-based against the best performing statistics-based and the *one-to-one* methods across multiple classifiers, dependent variables, and all six data sets to identify the best performing method. The results of these comparisons are reported in Section 6.

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 and the number of sample points generated by different segmentation approaches, we use a variant of the nonparametric Mann-Whitney rank test [30] 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 *one-to-one* method for the CCI measure, we select the distribution of the CCI measure generated for the best segmentation level of the *HC* taken across various segments and the distribution of the CCI measure obtained for each individual customer and then 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 given as follows:

- (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 data set, classifier, and dependent variable, we generate six sets of customer groups, CG_1, \dots, CG_6 , from our six segmentation methods *IG*, *IR*, *IM*, *HC*, *EC*, and *AP*, where cg_{ij} denotes a particular group j of customers belonging to customer group set CG_i generated by method i (*IG*, *IR*, *IM*, *HC*, *EC*, or *AP*). For each cg_{ij} , we generate a separate model, m_{ij} , that predicts the dependent variable of the model via 10-fold cross validation and computes three performance measures CCI_{ij} , RME_{ij} , and RAE_{ij} .

Let M_i denote the set of models m_{ij} 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

TABLE 2

Performance Tests across All Statistics-Based Segmentation Methods for Hypothesis Test (I)

Method	HC		AP	
	H ₁₊	H ₁₋	H ₁₊	H ₁₋
HC	-	-	18	66
EC	0	2	11	70

Numbers in columns H₁₊ and H₁₋ indicate the number of statistical tests that reject hypothesis H₀. Total significance tests per method to method comparison pair is 108.

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 via the Mann-Whitney rank tests using hypotheses H₀, H₁₊, H₁₋ specified above. For example, to compare HC and *one-to-one*, the above scenario of comparing three measures is repeated across six data sets, three dependent variables per data set, and two classifiers, resulting in 108 statistical significance tests comparing HC versus *one-to-one*.

In the following sections, due to computational expenses of the direct grouping methods, we first select the best statistics-based segmentation methods comparing HC, EC, and AP across more experimental conditions. We then select the best direct grouping methods by comparing the three proposed methods, IG, IR, and IM, across just one dependent variable and one classifier for each of the six data sets. And last, we compare the best performing methods selected for the direct grouping and the statistics-based approaches along with the *one-to-one* approach to select the overall best segmentation method.

6.1 Comparing Statistics-Based Methods

We compare the three statistics-based methods HC, EC, and AP across six data sets, three dependent variables per data set, 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 each pairwise comparison. Table 2 lists the number of statistical tests rejecting the null hypothesis (I) at 95 percent significance level.⁴ As Table 2 shows, only 2 out of 108 produced statistically significant differences between the HC and EC methods, in which HC dominated EC. However, AP dominates both HC and EC. From this comparison, we observe the following relationship among the statistics-based methods: $EC \leq HC < AP$. Since AP outperforms the other two methods, we have chosen it as a representative statistics-based segmentation method to compare against the best performing direct grouping method and *one-to-one* approaches in Section 6.3.

6.2 Comparing Direct Grouping Methods

We compared the three direct grouping methods IG, IR, and IM across the six data sets, one dependent variable, one classifier, and three performance measures described in Section 5 using Mann-Whitney tests (18 tests per each

4. Methods A (as indicated in the hypotheses H₀, H₁₊, and H₁₋ described above) are listed in the rows, and methods B from the aforementioned hypotheses are listed in the columns of this and other tables presented in this section (Tables 2, 3, 4, 5, and 6).

TABLE 3

Performance Tests across All Direct Grouping Methods for Hypothesis Test (I)

Methods	IR		IM	
	H ₁₊	H ₁₋	H ₁₊	H ₁₋
IG	0	18	0	18
IR	-	-	0	18

Numbers in columns H₁₊ and H₁₋ indicate the number of statistical tests that reject hypothesis H₀. Total significance tests per method to method comparison pair is 18.

comparison). Table 3 lists the number of statistical tests rejecting the null hypothesis (I) at 95 percent significance level for all the pairwise comparisons of IG, IR, and IM methods.

As Table 3 shows, the IM method overwhelmingly dominates the other two methods in all the 18 Mann-Whitney tests. Similarly, IR dominates IG also in all the 18 tests. From this, we can conclude that IM significantly outperforms the other two direct grouping-based segmentation methods, i.e., $IG < IR < IM$.

The IM dominance over the other two methods can be explained as follows: The segment merging decisions are made on the segments of comparable sizes for IM. This would result in more evenly distributed segment sizes in the final partition of the customer base than for the IG and IR methods, which tend to produce segments of larger and less evenly distributed sizes resulting in worse performing groupings of customers.

6.3 Comparing the Best Methods of Direct Grouping, Statistics-Based Segmentation, and One-to-One Methods

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 and 6.2, we selected the AP method to represent statistics-based grouping methods because it outperformed HC and EC, and the IM method to represent direct grouping methods because it outperformed IG and IR. As a result, we compare AP, IM, and *one-to-one* methods across the six data sets, three dependent variables per data set, two classifiers, and three performance measures. This resulted in a total of 108 Mann-Whitney tests per a pairwise comparison.

Table 4 summarizes the three pairwise comparisons by listing the number of statistical tests rejecting the null hypothesis (I) at 95 percent significance level. As is evident

TABLE 4

Performance Tests across *One-to-One*, AP, and IM for Hypothesis Test (I)

Methods	AP		IM	
	H ₁₊	H ₁₋	H ₁₊	H ₁₋
<i>1-to-1</i>	47	31	21	86
AP	-	-	12	54

Numbers in columns H₁₊ and H₁₋ indicate the number of statistical tests that reject hypothesis H₀. Total significance tests per method to method comparison pair is 108.

TABLE 5

Performance Tests across *One-to-One*, *AP*, and *IM* for Hypothesis Test (I) among High-Volume Customers

Methods	<i>AP</i>		<i>IM</i>	
	H ₁ +	H ₁ -	H ₁ +	H ₁ -
<i>I-to-I</i>	30	11	19	34
<i>AP</i>	-	-	4	29

Numbers in columns H₁ + and H₁- indicate the number of statistical tests that reject hypothesis H₀. Total significance tests per method to method comparison pair is 54.

from the number of statistically significant test counts, *IM* clearly dominates *one-to-one*, which in turn slightly dominates *AP*.

As Tables 5 and 6 show, while *one-to-one* dominates *AP* for high-volume customers, *AP* slightly dominates *one-to-one* for low-volume customers, thus reconfirming results demonstrated in [21] where modeling customers in segments using good clustering methods and low-volume transactions tend to perform better than modeling customers individually. We also note that *one-to-one* performs somewhat better against *IM* among high-volume data sets relative to low-volume data sets, which does make sense as the high-volume customers are more likely to have enough transaction data to effectively model individual customers. However, *IM* still shows significant performance dominance over *one-to-one* and *AP* across all the experimental conditions, including high- and low-volume customers.

To get a sense of the magnitude of the dominance that *IM* has over *one-to-one*, we computed the difference between the medians of each distribution. For a particular data set, 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 *one-to-one* methods. Then, we determined the medians of the two distributions⁵ (one for *IM* and one for *one-to-one*) and computed the differences between them. We repeated this process for all the 108 comparisons across the six data sets, three dependent variables per data set, two classifiers, and three performance measures, and plotted out the histograms of the median differences for the CCI, RME, and RAE measures in Figs. 4a, 4b, and 4c, 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 Figs. 4a, 4b, and 4c show that *IM* significantly outperforms the *one-to-one* method across most of the experimental conditions, thus providing additional visual evidence and the quantitative extent of the dominance of *IM* over *one-to-one* that was already statistically demonstrated with the Mann-Whitney tests.

We also did the same type of comparison for the *AP* and the *IM* methods. In Figs. 5a, 5b, and 5c, we show the left

5. 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.

TABLE 6

Performance Tests across *One-to-One*, *AP*, and *IM* for Hypothesis Test (I) among Low-Volume Customers

Methods	<i>AP</i>		<i>IM</i>	
	H ₁ +	H ₁ -	H ₁ +	H ₁ -
<i>I-to-I</i>	17	20	2	52
<i>AP</i>	-	-	8	25

Numbers in columns H₁ + and H₁- indicate the number of statistical tests that reject hypothesis H₀. Total significance tests per method to method comparison pair is 54.

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* versus *one-to-one*, Figs. 5a, 5b, and 5c clearly demonstrate *IM*'s dominance over *AP*.

Last, we did the same type of comparison for the *AP* and the *one-to-one* methods, and the results are reported in Figs. 6a, 6b, and 6c. As Fig. 6a shows, *one-to-one* dominates *AP* 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 *one-to-one* produces approximately the same amount of errors as *AP*. Thus, unlike the case of *IM*'s dominance over *AP*, the *one-to-one* approach does not clearly dominate *AP* across all the experimental conditions.

6.4 Performance Distributions of the One-to-One, AP, 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 Figs. 7a, 7b, and 7c.

Fig. 7a shows the histogram of the CCI performance measure distribution of the Naïve Bayes models generated by the *one-to-one* approach across 2,230 unique customer's data from the high-volume ComScore data set. The *x*-axis indicates the actual CCI score from a specific Naïve Bayes model trained and tested on specific customers, while the *y*-axis indicates the percentage of all the models having the corresponding CCI performance measure. Note that CCI scores vary from 10 percent to 100 percent, and the mean of the distribution is slightly above 50 percent.

Fig. 7b displays the histogram of the CCI performance measure distribution of the Naïve Bayes models generated by the *AP* segmentation method. Note how the CCI scores now have a tighter range, from close to 20 percent to 60 percent correct if we discount some outliers. However, the mean of the CCI distribution is significantly lower, at a little less than 40 percent. This illustrates our findings in the previous section where, compared to *one-to-one*, *AP* 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 *one-to-one* versus *AP*, as reported in Table 4.

Fig. 7c shows CCI distribution generated by the *IM* methods. Note that the distribution is slightly wider than

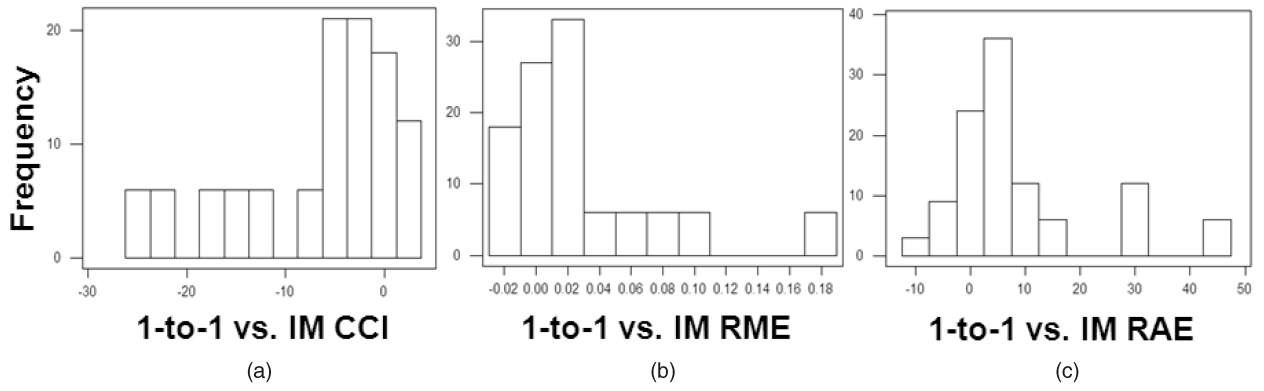


Fig. 4. Median difference distributions of *one-to-one* versus *IM*.

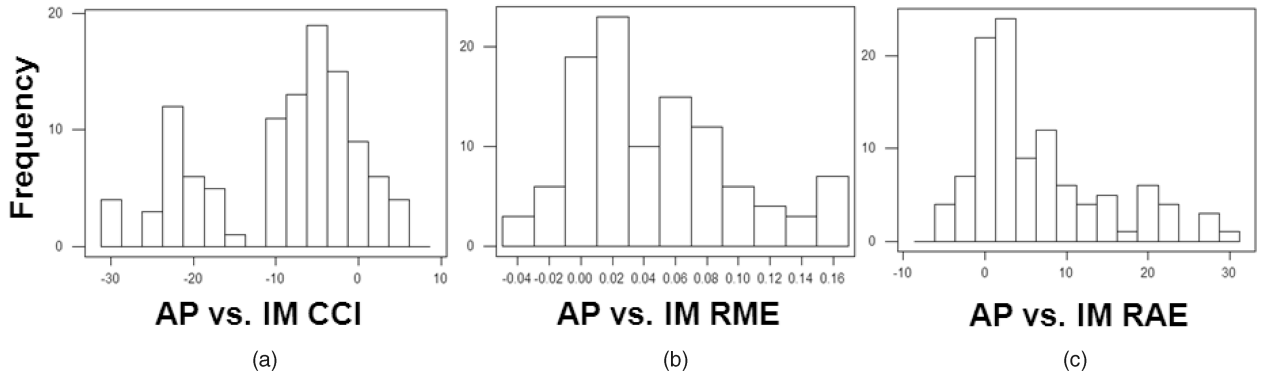


Fig. 5. Median difference distributions of *AP* versus *IM*.

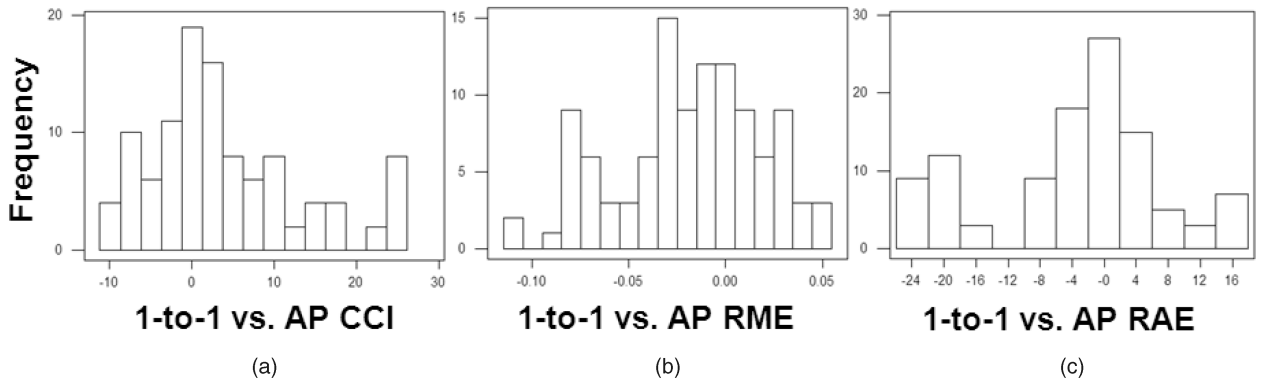


Fig. 6. Median difference distributions of *one-to-one* versus *AP*.

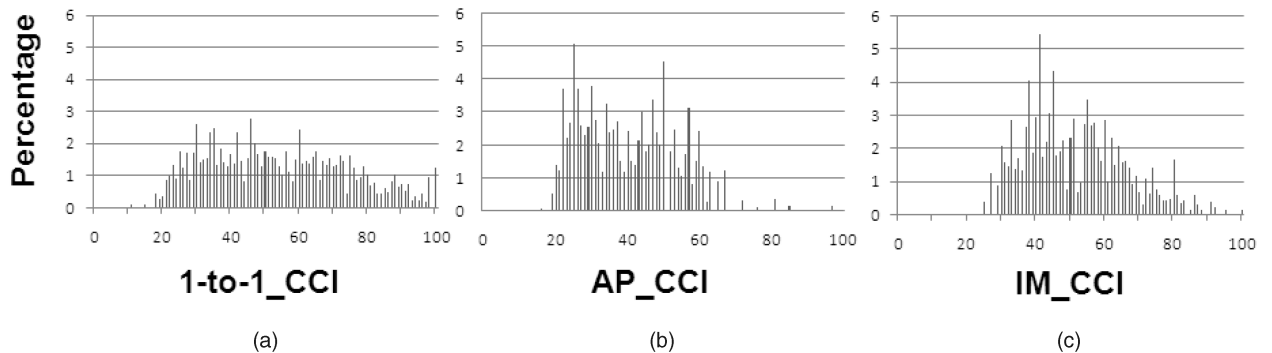


Fig. 7. Sample histograms of CCI measures generated by the *one-to-one*, *AP*, and *IM* methods using Naïve Bayes on the attribute “day of the week” for high-volume ComScore data.

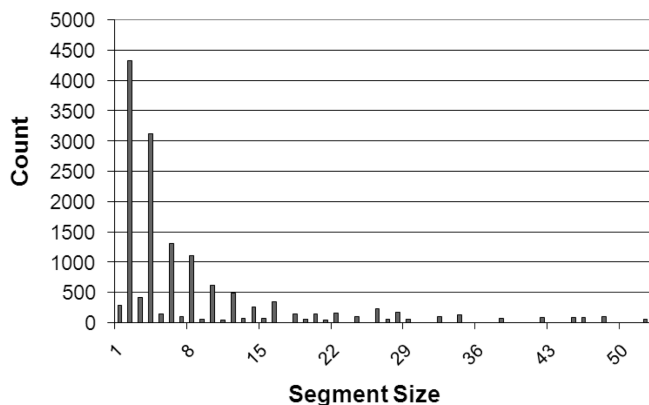


Fig. 8. The distribution of segment sizes generated by *IM* across high-volume data sets.

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

Again, Figs. 7a, 7b, and 7c 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*, *AP*, and *one-to-one* methods. Therefore, these selected CCI histograms provide additional insights into the nature of the *IM* dominance over the *one-to-one* and *AP* methods, as demonstrated in Tables 4, 5, and 6 and Figs. 4a, 4b, 4c, 5a, 5b, and 5c.

In summary, our empirical analysis clearly shows that, contrary to the popular belief [37], the *one-to-one* approach is definitely not the best solution for predicting customer behaviors. On the other hand, *IM*, which is essentially a microsegmentation 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 *one-to-one* CCI in Fig. 7a, which did not get picked by the *IM* method as presented in Fig. 7c. This shows that, while the *IM* method is statistically dominant over *one-to-one* and *AP*, *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 reasonable approach toward 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*.

Fig. 8 shows the distribution of segment sizes generated by *IM* for the high-volume customer data sets aggregated

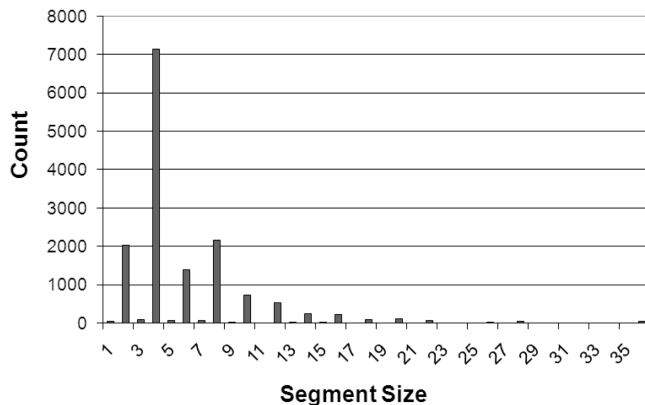


Fig. 9. The distribution of segment sizes generated by *IM* across low-volume data sets.

over all the experimental conditions. We note that the overall counts of segments peak 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. 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 *one-to-one* 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 Fig. 8.

Fig. 9 shows the distribution of segment sizes generated by *IM* across low-volume customer data sets. 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 Figs. 8 and 9 suggest that *IM*'s dominance over *one-to-one* and *AP* is largely due to the formation of large numbers of small customer segments, thus adding support to the use of microsegmentation in forming robust and effective models of customer behavior.

The distribution of segment sizes generated by *IM* (as shown in Figs. 8 and 9) clearly indicates that *IM* was able to find better performing groupings than just simple size-one segments. The peak at segment sizes of two and four for *IM* implies that segments of small sizes are better performing segments that significantly outperform their respective individual segments of size one, as demonstrated from *IM*'s dominance over *one-to-one*.

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 *one-to-one* solution, and we conjecture that it will contain predominantly small-sized segments, resulting in a microsegmented solution, as in the case of *IM*. We emphasize that 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 Figs. 8 and 9, we observe that the rate of decline in segment counts follows the power law distribution [48], and we tested and proven this conjecture as follows: Power law states that

$$\log(P_n) \approx -a \log(n), \quad (2)$$

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 power law hypothesis, and it turned out that the regression model indeed fitted the data. In particular, for coefficient a in (2) for the high-volume customers, the segmentation size distribution starting from segment size two has a value of $a = 0.828$, with p -value less than 0.001. As for low-volume customers, segmentation size distributions starting from segment size four has $a = 1.67$, with p -value less than 0.01. As with many natural phenomenon that has a power law 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 power law distribution for the *optimal* solution. However, formal analysis is required to prove this conjecture.

8 CONCLUSIONS

In this paper, we have examined the problem of optimal partitioning of customer bases into homogeneous segments for building better customer profiles and have presented 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 three suboptimal polynomial-time direct grouping methods *IM*, *IG*, and *IR* and showed that the *IM* method provides the best performance among them. We also adopted a previously proposed *AP* clustering algorithm to the statistics-based customer segmentation problem and showed that *AP* dominates two other standard statistics-based *HC* methods. We then compared *IM* against *AP* and showed that *IM* significantly dominates *AP* and, therefore, other statistics-based approaches considered in this paper across all the experimental conditions examined in this paper. We also showed that, contrary to the popular beliefs, *one-to-one* turned out to be significantly inferior to *IM* across all the experimental conditions. We then examined the distribution of the segments generated by *IM* and observed that there were very few size-one segments, that the distribution of segment sizes reached the maximum at the very small segment sizes, and that the rate of decline in the number of segments after this maximum followed a power law distribution. This

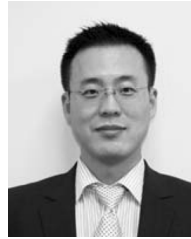
observation, along with the dominance of *IM* over *one-to-one*, provides strong support for the *microsegmentation* approach to personalization, where the customer base is partitioned into a large number of small segments.

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 (i.e., we want to determine if it follows the power law distribution similarly to *IM*). We would also like to develop additional efficient polynomial-time direct grouping methods that approach this optimal solution within some bounding limits and thus outperform *IM* and, hence, the *one-to-one* method. Further, we would like to develop more efficient versions of the suboptimal direct grouping methods having computational complexity better than $O(n^2)$ and $O(n^3)$. This would allow direct grouping segmentation of very large customer bases. 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- and economics-oriented performance measures such as customer value, profitability, and other economics-based performance measures.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin, "Expert-Driven Validation of Rule-Based User Models in Personalization Applications," *Data Mining and Knowledge Discovery*, vol. 5, nos. 1/2, pp. 33-58, 2001.
- [2] G. Adomavicius and A. Tuzhilin, "Personalization Technologies: A Process-Oriented Perspective," *Comm. ACM*, 2005.
- [3] G.M. Allenby and P.E. Rossi, "Marketing Models of Consumer Heterogeneity," *J. Econometrics*, vol. 89, 1999.
- [4] D. Beyer and R. Ogier, "Tabu Learning: A Neural Network Search Method for Solving Nonconvex Optimization Problems," *Proc. Int'l Joint Conf. Neural Networks (IJCNN)*, 1991.
- [5] Y. Boztug and T. Reutterer, "A Combined Approach for Segment-Specific Analysis of Market Basket Data," *European J. Operational Research*, 2007.
- [6] T. Brijs, T. Swinnen, K. Vanhoof, and G. Wets, "Using Shopping Baskets to Cluster Supermarket Shoppers," *AARTF*, Amelia Island Plantation, FL, 2001.
- [7] P. Brucker, "On the Complexity of Clustering Problems," *Optimization and Operations Research*, R. Henn, B. Korte, and W. Oettli, eds., pp. 45-54, Springer Verlag, 1977.
- [8] *Comm. ACM*, special issue on personalization, 2000.
- [9] I.V. Cadez, P. Smyth, and H. Mannila, "Predictive Profiles for Transaction Data Using Finite Mixture Models," Technical Report No. 01-67, UC Irvine, 2001.
- [10] C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith, "Hancock: A Language for Extracting Signatures from Data Streams," *Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, 2000.
- [11] W. DeSarbo and W.L. Cron, "A Maximum Likelihood Methodology for Clusterwise Linear Regression," *J. Classification*, vol. 5, pp. 249-282, 1988.
- [12] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," *Proc. 12th Int'l Conf. Machine Learning (ICML)*, 1995.
- [13] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, second ed. John Wiley & Sons, 2001.
- [14] U.M. Fayyad and K.B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI)*, 1993.
- [15] B. Frey and D. Dueck, "Mixture Modeling by Affinity Propagation," *Advances in Neural Information Processing Systems*, vol. 18, Y. Weiss, B. Scholkopf, and J. Platt, eds., MIT Press, 2006.
- [16] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," *Information Systems*, vol. 25, no. 5, pp. 345-366, 2000.

- [17] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, Sec. 6.3.2-6.3.3, MIT Press, 2001.
- [18] P. Hansen, "The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming," *Congress on Numerical Methods in Combinatorial Optimization*, 1986.
- [19] S.D. Hochbaum and B.D. Shmoys, "A Best Possible Heuristic for the K -Center Problem," *Math. Operational Research*, vol. 10, no. 2, pp. 180-184, 1985.
- [20] K. Hoffman, "Combinatorial Optimization: Current Successes and Directions for the Future," *J. Computational and Applied Math.*, vol. 124, pp. 341-360, 2000.
- [21] T. Jiang and A. Tuzhilin, "Segmenting Customers from Population to Individual: Does 1-to-1 Keep Your Customers Forever?" *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 10, Oct. 2006.
- [22] G.H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," *Proc. 11th Ann. Conf. Uncertainty in Artificial Intelligence (UAI)*, 1995.
- [23] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
- [24] P. Kotler, *Marketing Management*, 11th ed. Prentice Hall, 2003.
- [25] M. Koyuturk, A. Grama, and N. Ramakrishnan, "Compression, Clustering and Pattern Discovery in Very High Dimensional Discrete-Attribute Datasets," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 4, pp. 447-461, Apr. 2005.
- [26] F. Leisch, "A Toolbox for K -Centroids Cluster Analysis," *Computational Statistics and Data Analysis*, vol. 51, no. 2, pp. 526-544, 2006.
- [27] S. Lin and B.W. Kernighan, "An Effective Implementation for the Traveling Salesman Problem," *Operations Research*, vol. 21, pp. 498-516, 1973.
- [28] E. Malthouse, "Database Sub-Segmentation," *Kellogg on Integrated Marketing*, D. Iacobucci and B. Calder, eds., pp. 162-188, 2003.
- [29] E. Manavoglu, D. Pavlov, and C.L. Giles, "Probabilistic User Behavior Models," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM)*, 2003.
- [30] W. Mendenhall and R.J. Beaver, *Introduction to Probability and Statistics*. Thomson, 1994.
- [31] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Using Sequential and Non-Sequential Patterns for Predictive Web Usage Mining Tasks," *Proc. IEEE Int'l Conf. Data Mining (ICDM)*, 2002.
- [32] H. Mühlenbein, "Parallel Genetic Algorithms in Combinatorial Optimization," *Computer Science and Operations Research*, O. Blaci, ed., Pergamon Press, 1992.
- [33] O. Nasraoui, M. Soliman, E. Saka, A. Badia, and R. Germain, "A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 2, Feb. 2008.
- [34] J. Novo, *Drilling Down: Turning Customer Data into Profits with a Spreadsheet*, Booklocker, 2004.
- [35] M. Ozdal and C. Aykanat, "Clustering Based on Data Patterns Using Hypergraph Models," *Data Mining and Knowledge Discovery*, vol. 9, pp. 29-57, 2004.
- [36] M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning*, vol. 27, no. 3, pp. 313-331, 1997.
- [37] D. Peppers and M. Rogers, *Enterprise One to One*. Bantam, 1997.
- [38] R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [39] T. Reutterer, A. Mild, M. Natter, and A. Taudes, "A Dynamic Segmentation Approach for Targeting and Customizing Direct Marketing Campaigns," *Interactive Marketing*, vol. 20, no. 3/4, pp. 43-57, 2006.
- [40] W. Smith, "Product Differentiation and Market Segmentation as Alternative Marketing Strategies," *J. Marketing*, vol. 21, 1956.
- [41] Spath, "Algorithm 39: Clusterwise Linear Regression," *Computing*, vol. 22, pp. 363-373, 1979.
- [42] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa, "A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis," *INFORMS J. Computing*, no. 2, p. 15, 2003.
- [43] M. Wedel and W.S. DeSarbo, "A Mixture Likelihood Approach for Generalized Linear Models," *J. Classification*, vol. 12, 1995.
- [44] M. Wedel, W. Kamakura, N. Arora, A. Bemmaor, J. Chiang, T. Elrod, R. Johnson, P. Lenk, S. Neslin, and C.S. Poulsen, "Discrete and Continuous Representations of Unobserved Heterogeneity in Choice Modeling," *Marketing Letters*, vol. 10, no. 3, pp. 219-232, 1999.
- [45] M. Wedel and W. Kamakura, *Market Segmentation: Conceptual and Methodological Foundations*. Kluwer, 2000.
- [46] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.
- [47] Y. Yang and B. Padmanabhan, "Segmenting Customer Transactions Using a Pattern-Based Clustering Approach," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM)*, 2003.
- [48] G.K. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.



Tianyi Jiang received the BS and the MEng degrees from Cornell University and the PhD degree in data mining from Stern School of Business, New York University. His current research interests include personalization, customer segmentation, consumer profiling, and credit risk modeling. In addition to publishing in leading CS and IS conference proceedings on these topics, he is also COO and cofounder of AvePoint, a global enterprise data management software company.



Alexander Tuzhilin received the PhD degree in computer science from the Courant Institute of Mathematical Sciences, New York University (NYU). He is a professor of information systems and the NEC faculty fellow in the Stern School of Business, NYU. His current research interests include data mining, personalization, recommender systems, and CRM. He has published widely in the leading CS and IS journals and conference proceedings on these and other research topics. He served on the organizing and program committees of numerous CS and IS conferences, including as a program cochair of the Third IEEE International Conference on Data Mining and as a conference cochair of the Third ACM Conference on Recommender Systems. He has also served on the editorial boards of the *IEEE Transactions on Knowledge and Data Engineering*, the *Data Mining and Knowledge Discovery Journal*, the *INFORMS Journal on Computing* (as an area editor), the *Electronic Commerce Research Journal*, and the *Journal of the Association of Information Systems*.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.