Les Cahiers du GERAD ISSN: 0711–2440

A model for clustering data from heterogeneous subjects

É. Santi, D. Aloise,
S.J. Blanchard

G–2015–18

March 2015

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A model for clustering data from heterogeneous subjects

Éverton Santi[a]

Daniel Aloise[a,b]

Simon J. Blanchard[c]

[a] *Department of Computer Engineering and Automation, Universidade Federal do Rio Grande do Norte, Natal-RN, Brazil, 59072-970*

[b] *GERAD, HEC Montréal, Montréal (Québec) Canada, H3T 2A7*

[c] *McDonough School of Business, Georgetown University, Washington, DC 20057, USA*

santi.everton@gmail.com
aloise@dca.ufrn.br
sjb247@georgetown.edu

March 2015

Les Cahiers du GERAD

G–2015–18

Abstract: Clustering is a data mining method which consists in partitioning a given set of n objects into p clusters in order to minimize the dissimilarity among objects in the same cluster while dissimilarities regarding objects of other clusters are maximized. Classical clustering methods use only one dissimilarity matrix concerning each pair of objects as input. However, in some settings where data can be collected from different perspectives, multiple dissimilarity matrices are available. In such cases, researchers typically aggregate their data into a single matrix resulting in clustering results that mask the true nature of the data. We propose in this paper a clustering model consisting of a three-way partitioning problem that identifies segments of subjects that cluster objects in a similar way. The model is a nonconvex problem for which we propose a Variable Neighborhood Search heuristic whose local search is based on the solution of mixed-integer problems. Computational experiments show that the heuristic is efficient and that the proposed model is suited for recovering heterogeneous data as well as it is robust to different clustering settings.

Key Words: Data mining, clustering, heterogeneity, optimization, heuristics.

Acknowledgments: The authors would like to thank Wayne DeSarbo for his comments on a previous draft of this manuscript and Leo Liberti for fruitful talks about the model solution and strenghtening. This research has been supported by the National Council for Scientific and Technological Development – CNPq/Brazil grant number 471143/2012-0. Research of the first author was also supported by CAPES/Brazil.

1 Introduction

Clustering models help automatically identify subsets of objects, called clusters, such that objects in the same cluster are similar in some way (Hansen and Jaumard, 1997). Clustering is ubiquitous, with applications in the natural sciences, psychology, medicine, engineering, economics, marketing and other fields (e.g. Frey and Dueck, 2007; Jain et al., 1999; McLachlan and Basford, 1988).

One of the most used types of clustering is *partitioning*, where given a set $O = \{o_1, o_2, \ldots, o_n\}$ of n objects, we look for a partition $P = \{C_1, C_2, \ldots, C_p\}$ of O into p clusters such that:

(i) $C_j \neq \emptyset, \quad \forall j = 1, 2, \ldots, p;$
(ii) $C_i \cap C_j = \emptyset, \quad \forall i, j = 1, 2, \ldots, p$ and $i \neq j$; and
(iii) $\bigcup_{j=1}^{p} C_j = O.$

Typically, clustering is performed over a data matrix X of dimension $n \times s$ obtained by measuring or observing s features of the objects of O. An $n \times n$ matrix of dissimilarities $D = (d_{ij})$ between objects of O is then computed from the matrix X, such that d_{ij} for $i, j = 1, 2, \ldots, n$ (usually) satisfy: (i) $d_{ij} = d_{ji} \geq 0$, and (ii) $d_{ii} = 0$. Such single D dissimilarity matrix data do not need to satisfy triangle inequalities, i.e., to be distances.

There are however many applications in which a single dissimilarity matrix is not appropriate, as the data itself may be subject to measurement error or respondent variation (Daws, 1996). Whereas in the Iris flowers dataset (Fisher, 1936) measurement error is likely to be minimal, there exist settings for which (dis)similarity from individual perceptions of objects are subject to greater variations. In such cases, using a single dissimilarity matrix is likely to lead to erroneous results (Brusco and Cradit, 2005; DeSarbo and Carroll, 1985; Lee, 2001; Steinley et al., 2015; Vichi et al., 2007).

Prior research has shown that judgments of similarity vary because consumers are heterogeneous with respect to the object features that they choose to attend to when forming their similarity judgements (Blanchard, 2011; Blanchard et al., 2012a; DeSarbo et al., 2008). It is thus no surprise that performing analysis on heterogeneous distance matrices would lead to fitting clustering solutions that vary in the classes obtained. As in cluster analysis the number of clusters to be identified is often specified by the user as opposed to data driven (Milligan and Cooper, 1985), identifying the number of clusters when analyzing heterogeneous similarity data can be challenging (e.g. Blanchard et al., 2012b) particularly as there is often no objectively better choice in terms of which clustering model should be used in the first place (Kleinberg, 2003).

The p-median model is certainly one of the most used clustering models. Its objective is to minimize the total distance from each object to the central exemplar (i.e., median) of its cluster, where the total number of clusters is p. The problem may be formulated as an integer linear program. Given n objects to be clustered, let $e_{jj} = 1$, if object j is chosen as the median of a cluster, and 0 otherwise, for $j = 1, \ldots, n$; $e_{ij} = 1$, if object i is assigned to the cluster whose object j is the median, and 0 otherwise, for $i = 1, \ldots, n$ (object j is naturally assigned to itself if it is a median). We obtain the following model:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} e_{ij} \tag{1}$$

subject to

$$\sum_{j=1}^{n} e_{ij} = 1, \qquad \forall i = 1, \ldots, n \tag{2}$$

$$\sum_{j=1}^{n} e_{jj} = p \tag{3}$$

$$e_{ij} \leq e_{jj} \qquad \forall i, j = 1, \ldots, n \tag{4}$$

$$e_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n. \tag{5}$$

The constraints (2) specify that each object must be assigned to one and only one median. Constraint (3) imposes that the number of medians is p. The constraints (4) ensure that object i can only be assigned to object j if it is a median. Finally, constraints (5) give the binary restrictions on the decision variables.

One of the main characteristics of the p-median is its breadth of applicability. It can be applied to cluster metric data as well as to more general similarity/dissimilarity data, even asymmetric or rectangular data structures (Köhn et al., 2010). Mladenović et al. (2007) present an extensive review of exact and heuristic solution methods for this problem. Despite its advantages, including excellent classification rates, robustness to outliers and attractive assumptions, an aggregate p-median formulation may still mask individual heterogeneity.

In this paper, we propose a mathematical programming model to cluster data collected from subjects with heterogeneous clustering structures. The model is conceived in two levels, the first identifies clusters of subjects, thereafter called *segments* for readability, with similar clustering structures and the second partitions the objects in each of these segments. Then, we detail a Variable Neighborhood Search (VNS) (Hansen and Mladenović, 2001) heuristic, showing that it can be gainfully employed to cluster data from heterogeneous subjects. We show that solving the model via our heuristic offers superior recovery for the original data, as illustrated via a Monte Carlo simulation and an empirical application concerning sorts of chocolate snacks.

The mathematical formulation of the model is given in the next section. The VNS heuristic is described in Section 3. Monte Carlo simulation results are shown in Section 4, which demonstrates the necessity for a heuristic in order to tackle large data sets as well as it presents the performance of the algorithm with respect to data recovery under a variety of clustering specifications. Section 5 provides an empirical example from a local US retailer about perceptions of chocolate candies to illustrate how the proposed methodology can help discover insights based on consumer perceptions and uniquely inform marketing decision makers. Concluding remarks are drawn in the last section.

2 Problem formulation

Let m subjects consider n objects such that a matrix data $D^k = (d^k_{ij})$ is obtained for $k = 1, \ldots, m$, representing the dissimilarities between pairs of objects i and j as perceived by subject k, and c^k, for $k = 1, \ldots, m$ as the number of clusters subject k wants to classify the objects in the dataset.

The clustering problem considered in this work consists in identifying segments of subjects with similar judgements concerning the observed objects at the same time providing a clustering of the objects in each identified segment. The clustering of objects is performed by means of a medians-based model where each clustered object is summarized by the median of its cluster. The problem, thereafter called the *Heterogeneous Clustering Problem* (HCP), is formulated as:

$$\min \sum_{k=1}^{m} \sum_{g=1}^{G} z^{kg} \left[\sum_{i=1}^{n} \sum_{j=1}^{n} d^k_{ij} e^g_{ij} \right] \tag{6}$$

subject to

$$\sum_{j=1}^{n} e^g_{ij} = 1 \qquad \forall g = 1, \ldots, G,\ \forall i = 1, \ldots, n \tag{7}$$

$$e^g_{ij} \leq e^g_{jj} \qquad \forall g = 1, \ldots, G,\ \forall i, j = 1, \ldots, n \tag{8}$$

$$\sum_{g=1}^{G} z^{kg} = 1 \qquad \forall k = 1, \ldots, m \tag{9}$$

$$\sum_{k=1}^{m} z^{kg} \geq 1 \qquad \forall g = 1, \ldots, G \tag{10}$$

$$\sum_{j=1}^{n} e^g_{jj} = \left\lfloor \frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}} \right\rfloor \qquad \forall g = 1, \ldots, G \tag{11}$$

$$e_{ij}^g \in \{0,1\} \qquad \forall g = 1,\ldots,G,\ \forall i,j = 1,\ldots,n \tag{12}$$

$$z^{kg} \in \{0,1\} \qquad \forall g = 1,\ldots,G,\ \forall k = 1,\ldots,m \tag{13}$$

The m subjects must be partitioned into G segments. The decision variables z^{kg} express the assignment of subject k to segment g. Variables e_{ij}^g are equal to 1 if object i is assigned to object j in segment g, 0 otherwise. The objective is to minimize (6), i.e., the sum of dissimilarities between each object and its assigned median, conditional on (subject) segment membership. Constraints (7) impose that each object i must be assigned to exactly one median in each segment g. Constraints (8) ensure that object i can only be assigned to object j in segment g if it is a median of that segment. Constraints (9) ensure that each subject is assigned to exactly one segment whereas constraints (10) guarantee that no empty segment exist. Constraints (11) make the total number of medians for each segment g equal to the floor of the average number of medians expected by the subjects in that segment. This sugestion follows numerous researchers in the behavioral literature who have shown that individuals tend to favor simple rules when forming object perceptions and preferences (Bettman and Park, 1980; Bettman et al., 1998; Simon, 1955; Shugan, 1980).

The optimization process guarantee that $\sum_{j=1}^{J} e_{jj}^g$ is an integer value as big as possible since more medians in a segment imply lower (or equal) objective function values. Consequently, constraints (11) can be replaced by the following inequalities:

$$\sum_{j=1}^{n} e_{jj}^g \leq \frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}}, \qquad \forall g = 1,\ldots,G, \tag{14}$$

without affecting the optimal solution. Moreover, these constraints can be straightforwardly modified if the user prefers that the number of medians in each segment equals the closest integer to $\frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}}$ instead of the floor. For that, it would suffice to add 0.5 to the right-hand side of constraints (14).

The HCP is still nonconvex due to the objective function (6) and constraints (14). Convexification can be achieved by means of Fortet's inequalities (Fortet, 1960), thereby replacing the product of binary variables $e_{ij}^g \times z^{kg}$ by w_{ij}^{kg} ($w_{ij}^{kg} \in [0,1]$) for $i = 1,\ldots,m; j,k = 1,\ldots,n; g = 1,\ldots,G$, along with three additional constraints which together ensure that $\max\{0, e_{ij}^g + z^{kg} - 1\} \leq w_{ij}^{kg}$. The three sets of constraints are:

$$w_{ij}^{kg} \leq e_{ij}^g, \qquad \forall g = 1,\ldots,G,\ \forall k = 1,\ldots,m,\ \forall i,j = 1,\ldots,n, \tag{15}$$

$$w_{ij}^{kg} \leq z^{kg}, \qquad \forall g = 1,\ldots,G,\ \forall k = 1,\ldots,m,\ \forall i,j = 1,\ldots,n, \tag{16}$$

$$w_{ij}^{kg} \geq e_{ij}^g + z^{kg} - 1, \qquad \forall g = 1,\ldots,G,\ \forall k = 1,\ldots,m,\ \forall i,j = 1,\ldots,n. \tag{17}$$

To further accelerate the optimization process of the resulting mixed-integer problem (MIP), we strengthen the formulation by adding constraints (cuts) that do not affect the optimal integer solution. In the spirit of the Reformulation-Linearization Technique (RLT) (Sherali and Alameddine, 1992), we obtain a set of additional cuts by multiplying the $n \times G$ constraints in (7) by z^{kg}, for $k = 1,\ldots,m$, then replacing the products $e_{ij}^g \times z^{kg}$ by w_{ij}^{kg}. This yields the following constraints:

$$\sum_{j=1}^{n} w_{ij}^{kg} = z^{kg}, \qquad \forall g = 1,\ldots,G,\ \forall k = 1,\ldots,m,\ \forall i = 1,\ldots,n. \tag{18}$$

An important consequence of these constraints is that they make constraints (16) and (17) redundant.

As our computational experiments in Section 4 demonstrate, general purpose exact solvers for formulations (6)–(13) and its MIP version are very restrained with respect to the size of the HCP instances they can solve. With that in mind, we introduce in the next section a VNS heuristic for the problem. Finally, it is worthy to say that the problem is already NP-hard for $G = 1$ since, in this case, it is equivalent to the p-median problem (Kariv and Hakimi, 1979).

3 VNS heuristic for the HCP

VNS is a metaheuristic developed to solve combinatorial and global optimization problems by changing neighborhoods in its local descent step for intensification as well as in its shaking step for diversification (see Hansen et al., 2008, for a recent survey).

VNS relies on the following three observations:

Observation 1: *A local minimum with respect to one neighborhood structure is not necessary so for another*;

Observation 2: *A global minimum is a local minimum with respect to all possible neighborhood structures*;

Observation 3: *Local minima with respect to one or several neighborhoods are often relatively close to one another.*

In the VNS framework, the neighborhoods used are defined around types of moves, or perturbations, of the best current solution x – the center of the search. When looking for a better one in a minimization problem, a solution x' is drawn at random in an increasingly wider neighborhood and a local descent is performed from x', leading to another local optimum x''. If x'' is worse than x, then x'' is ignored and one chooses a new neighbor solution x' in a further neighborhood of x. If, otherwise, x'' is better than x, the search is re-centered around x'' and local search restarts in the closest neighborhood of the newly found best current solution. Once all neighborhoods of x have been explored without success, one begins again with the closest one to x, until a stopping condition (e.g. maximum CPU time) is met. As a summary, the steps of a basic VNS are given in Algorithm 1.

Algorithm 1 VNS Framework

Initialization: Select the set of neighborhoods structures $\mathcal{N}_t$, for $t = 1, \ldots, t_{max}$, that will be used in the search; find an initial solution x.
repeat
  $t \leftarrow t_{min}$
  repeat
    Shaking: Generate a point x' at random from the t^{th} neighborhood of x (i.e., $x' \in \mathcal{N}_t(x)$);
    Local search: Apply some local search method with x' as the initial solution; denote with x'' the so obtained local optimum;
    Move or not: If the local optimum x'' is better than the incumbent x, move there ($x \leftarrow x''$), and continue the search with $\mathcal{N}_{t_{min}}(t \leftarrow t_{min})$; otherwise, set $t \leftarrow t + t_{step}$.
  until $t = t_{max}$
until a stopping criterion is met

As the size of neighborhoods tends to increase with their distance from the current best solution x, close-by neighborhoods are explored more thoroughly than far away ones. This strategy takes advantage of the three observations 1–3 mentioned above, and yet can ensure with sufficient computational time that the algorithm is not stuck in a poor local optima if the response surface is very ill shaped.

3.1 Initialization

VNS requires an initial solution which can be either provided or constructed by the user. Algorithm 2 presents the pseudocode of our approach to construct an initial solution.

Algorithm 2 first solves the problem of assigning subjects to segments, and does so by using a distance matrix of $m \times m$ subjects based on the Frobenius norm of each subject's distance matrix between objects. Once the p-median is applied to this distance matrix between subjects, they are assigned to segments according to the partition obtained, i.e., if a pair of subjects have their distance matrices assigned to the same cluster in the p-median model then these subjects are assigned to the same segment in the initial solution. Then, for each initial segment, solving subproblems $M_g(z)$, for $g = 1, \ldots, G$ provides a complete initial solution for HCP:

Algorithm 2 VNS: Constructive Heuristic (CH)

Compute a matrix $\mathcal{F} = (f^{ab})$ with dimension $m \times m$ such that f^{ab} is the Frobenius norm of $D^a - D^b$;
Solve a p-median problem with $p = G$ medians using matrix $\mathcal{F}$ as input;
for $k = 1, \ldots, m$ **do**
  set $z^{kg} = 1$ if the dissimilarity matrix of subject k is assigned to the g-th median, $z^{kg} = 0$ otherwise;
end for
for $g = 1, \ldots, G$ **do**
  solve subproblems $M_g(z)$;
end for

$$M_g(z) = \min \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{d}_{ij}^{g} e_{ij}^{g} \tag{19}$$

subject to

$$\sum_{j=1}^{n} e_{ij}^{g} = 1 \qquad g = 1, \ldots, G,\ j = 1, \ldots, n \tag{20}$$

$$e_{ij}^{g} \leq e_{jj}^{g} \qquad \forall g = 1, \ldots, G,\ i, j = 1, \ldots, n \tag{21}$$

$$\sum_{j=1}^{n} e_{jj}^{g} = \lfloor \Omega^{g} \rfloor \tag{22}$$

$$e_{ij}^{g} \in \{0, 1\} \qquad \forall g = 1, \ldots, G,\ i, j = 1, \ldots, n, \tag{23}$$

where $\overline{d}_{ij}^{g} = \sum_{k=1}^{m} d_{ij}^{k} z^{kg}$, and $\Omega^{g} = \frac{\sum_{k=1}^{m} c^{k} z^{kg}}{\sum_{k=1}^{m} z^{kg}}$. We note that problem (19-23) corresponds to the p-median problem (1-5).

3.2 Shaking

The shaking component of our VNS is implemented by means of random moves in the *swap* neighborhood which encompasses all the possible ways of removing a subject from a segment and adding it to a different one. Thus, if the parameter $t = 2$ for shaking, then two random swap moves are performed for two subjects; if $t = 3$, then three swap moves are performed for three subjects, and so on.

3.3 Local search

Given an existing solution, we need to search the neighborhood to reach a local optima. We developed our local search following the Variable Neighborhood Descent (VND) framework, which generalizes the observations 1–3 to descent methods. Algorithm 3 presents a general VND's algorithmic steps.

Algorithm 3 Local Search: VND Framework

Input: a solution x and a set of descent methods $\mathsf{descent}_s$, for $s = 1, \ldots, s_{max}$
$s \leftarrow s_{min}$
repeat
  $x' \leftarrow \mathsf{descent}_s(x)$;
  If $x \neq x'$ make $x \leftarrow x'$ and $s \leftarrow s_{min}$; otherwise $s \leftarrow s + 1$;
until $s > s_{max}$

Applied to HCP, VND involves the alternating optimization of the objective function via improvements based on three descent methods: 1) descent on objects clusterings (conditional on segment memberships and number of medians), 2) descent on segment memberships (conditional on objects clusterings and number of medians), and 3) descent by augmenting the number of medians. VND (the local search) ends when all

descent methods are explored from a solution x without improvement in the objective function. Whenever an improvement occurs, the algorithm resets s to s_{min}. In the present section, we present each one of our descent procedures.

3.3.1 First descent

The descent method $\mathsf{descent}_1$ for Algorithm 3 solves subproblem (19-23) for each segment affected by the shaking procedure. Namely, it identifies for each segment the best clustering for the objects while temporarily assuming that both a) the number of medians, and b) the segment memberships, are known. Given that this descent is performed several times by the algorithm, we use the state-of-the-art POPSTAR heuristic (Resende and Werneck, 2004) for the p-median problem in order to accelerate its estimation as opposed to solving it exactly.

3.3.2 Second descent

The second descent method $\mathsf{descent}_2$ temporarily considers the clustering of objects as known in all segments (i.e. variables e) and improves on the objective function by conditionally reassigning subjects to the segments that provide the best values for z. To do so, the following binary program is solved:

$$W(e) = \min \sum_{k=1}^{m} \sum_{g=1}^{G} z^{kg} \tilde{d}^{kg} \tag{24}$$

subject to

$$\frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}} \geq \omega^g \qquad \forall\ g = 1, \ldots, G \tag{25}$$

$$\sum_{g=1}^{G} z^{kg} = 1 \qquad \forall\ k = 1, \ldots, m \tag{26}$$

$$z^{kg} \in \{0, 1\} \qquad \forall\ g = 1, \ldots, G,\ k = 1, \ldots, n \tag{27}$$

where $\tilde{d}^{kg} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^k e_{ij}^g$, and $\omega^g = \sum_{j=1}^{n} e_{jj}^g$. Problem (24-27) is a binary program which usually require a few branch-and-bound nodes to be solved by CPLEX according to our limited computational experiments. As solving it exactly would involve extensive computational times, we chose to impose a time limit of one second to the exploration of the branch-and-bound tree - at which point the best feasible solution found is returned (if it exists). Otherwise, the search proceeds with current solution x. Whereas increasing the search time could influence the solution found in this step, our experiments suggest that allowing for additional computational time has no significant effect.

3.3.3 Third descent

The clustering of objects in each segment of the HCP, represented by variables e, has indeed smaller cost as more medians are used. However, the HCP restrains the number of medians in each segment g, for $g = 1, \ldots, G$, by means of constraints (14), thus pushing that number to the largest integer smaller or equal to $\frac{\sum_{k=1}^{m} c^k z^{kg}}{\sum_{k=1}^{m} z^{kg}}$.

The idea of the third descent is to verify if the number of medians in a segment g^* can be augmented by reallocating subjects to it, thereby satisfying constraints (14) to the new number of medians.

Algorithm 4 details how this procedure works. To explain the procedure presented in Algorithm 4, consider that the following actions are applied sequentially to each segment $g^* (g^* = 1, \ldots, G)$. Firstly, a solution (z^{best}, e^{best}) is initialized with the values of the best current solution for the HCP, i.e., $(z^{best}, e^{best}) \leftarrow (z, e)$. From that, the problem $M_{g^*}(z)$ is solved with Ω^{g^*} replaced by $\Omega^{g^*} + 1$, thereby producing a new partition for the objects in g^* with the number of medians augmented by one unit. In the sequel, the problem $W(e)$ is solved in order to verify if this modification in segment g^*, i.e., Ω^{g^*} replaced by $\Omega^{g^*} + 1$, can be accomodated

Algorithm 4 descent$_3$

Input: a solution (z, e) for HCP
$(z^{best}, e^{best}) \leftarrow (z, e)$
for $g^* = 1, \ldots, G$ **do**
  Solve $M_{g^*}(z)$ with Ω^{g^*} replaced by $\Omega^{g^*} + 1$;
  Solve $W(e)$;
  if $W(e)$ is infeasible **or** cost of (z, e) greater than the cost of (z^{best}, e^{best}) **then**
    $(z, e) \leftarrow (z^{best}, e^{best})$;
  else
    $(z^{best}, e^{best}) \leftarrow (z, e)$;
  end if
end for
return (z^{best}, e^{best})

by reassigning the subjects among the segments. If $W(e)$ is infeasible or the cost of the new yielded solution is larger than the best solution found so far, the latter is restored. Otherwise, it becomes the new best incumbent solution.

A critical design decision in VND heuristics is how to define an order for the descent methods in Algorithm 3. This decision was straightforward here since: (i) the third descent should be clearly the last one since it is much more computationally expensive than the other two; and (ii) if the second descent was selected as the first in Algorithm 3, it would revert the shaking procedure previously applied.

4 Monte Carlo simulation: Testing the performance and robustness

In the present section, we seek to illustrate the perfomance of approaching the HCP directly via: (a) the state-of-the-art nonconvex Mixed-Integer Nonlinear Program (MINLP) solver Couenne version 0.4 (Belotti et al., 2009) on its nonconvex formulation (6)–(13), (b) the state-of-the-art MIP solver CPLEX version 12.5 (ILOG, 2012) on its MIP formulation, and (c) the VNS heuristic presented in the last section. Moreover, we wish to compare the VNS performance in recovering the ground-truth partitions of synthetic data.

To obtain a set of known data structures for D^k, c^k and z^{kg}, for $k = 1, \ldots, m; g = 1, \ldots, G$ so that performance can be compared, we generated data following the fractional factorial design used by Blanchard et al. (2012b). The design involves 27 synthetic trials that can be used to study the impact of dataset characteristics on the ability of the competing models to recover the original data. The factorial design appears in Table 1. The experimental factors included the total number of subjects ($m = 150, 300, 450$), the number of segments ($G = 2, 6, 10$), the number of objects ($n = 18, 30$), variance in the number of medians across segments, error added to the dissimilarity matrix of each subject (using $N(0, 0.05)$ or $N(0, 0.1)$ before rounding), error added to the number of medians sought by each subject (using $N(0, 0.5)$ or $N(0, 1)$ before rounding). Following the works of Blanchard et al. (2012b), Blanchard and DeSarbo (2013), Brusco and Cradit (2001) and others, we also assume that the true number of segments G is known - but not their composition.[1]

4.1 Optimization results

We first turn to illustrating how Couenne, CPLEX and the VNS heuristic compare with respect to solving the 27 synthetic instances from Table 1. Computational experiments were performed on a Xeon(R) CPU X5650 2.67GHz and 62GB of RAM memory with 12 processors. CPLEX was allowed to run for 24 hours using all 12 processors in parallel. The VNS heuristic was allowed to run for 600 seconds in a single processor. The algorithm was implemented in C++ and compiled by gcc 4.4.

[1] The instances can be found at `http://www.gerad.ca/~aloise/publications.html`

Table 1: Montecarlo simulation: Experimental design

Instance	Subjects m	Segments G	Objects n	Medians	Perturbation Dissimilarities	Perturbation Medians
1	150	10	30	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)
2	300	2	18	All 6	N(0, 0.1)	0
3	450	2	18	50 % 3, 50 % 6	N(0, 0.05)	0
4	150	2	18	All 3	N(0, 0.05)	N(0, 0.5)
5	450	10	18	All 6	N(0, 0.05)	N(0, 1)
6	150	10	18	50 % 3, 50 % 6	N(0, 0.05)	0
7	300	2	18	All 6	0	N(0, 0.5)
8	150	10	18	50 % 3, 50 % 6	0	N(0, 1)
9	300	10	30	All 3	N(0, 0.05)	N(0, 0.5)
10	450	6	18	All 3	N(0, 0.1)	N(0, 1)
11	150	6	30	All 6	N(0, 0.1)	0
12	300	10	18	All 3	0	0
13	450	10	18	All 6	N(0, 0.1)	0
14	300	6	18	50 % 3, 50 % 6	0	N(0, 1)
15	300	2	30	All 6	N(0, 0.05)	N(0, 1)
16	450	2	30	50 % 3, 50 % 6	0	N(0, 1)
17	300	6	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)
18	300	6	30	50 % 3, 50 % 6	N(0, 0.05)	0
19	150	6	18	All 6	0	N(0, 0.5)
20	450	6	30	All 3	0	0
21	150	2	30	All 3	N(0, 0.1)	N(0, 1)
22	450	2	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)
23	450	6	18	All 3	N(0, 0.05)	N(0, 0.5)
24	300	10	18	All 3	N(0, 0.1)	N(0, 1)
25	150	6	18	All 6	N(0, 0.05)	N(0, 1)
26	150	2	18	All 3	0	0
27	450	10	30	All 6	0	N(0, 0.5)

Table 2 presents lower and upper bounds (feasible solutions) obtained by CPLEX and the VNS heuristic for the HCP. For each instance, the second and third columns report the best lower and upper bounds obtained by CPLEX within 24 hours of CPU time. The fourth column reports the number of branch-and-bound nodes explored by CPLEX in the associated execution whereas the fifth column refers to the relative difference (in %) between the upper and lower bounds. Regarding the tailored VNS heuristic, the sixth column reports the cost of the initial solution provided by the constructive heuristic. The seventh column presents the average upper bound solutions obtained in 10 distinct executions of the heuristic, whereas the eigth column shows their associated standard deviation. Finally, the ninth column refers to the relative difference (in %) between the average VNS upper bound and the lower bound obtained by CPLEX. The table does not present results for Couenne since it was never able to obtain a lower bound different from the trivial one (i.e., zero) within 24 hours of CPU time.

The results in Table 2 reveal that:

- The VNS outperforms CPLEX in terms of upper bound solutions in all instances, except for instances #7 and #26 whose results obtained by both algorithms are equal. The superior performance of VNS attains its maximum for instance #13 with a difference of 44.22% in solution quality.
- The VNS algorithm is stable as demonstrated by the very small standard deviation results. The worst values are, however, observed for larger instances. For instance, the worst standard deviation for the 10 distinct VNS executions is found for instance #8 which is varied with respect to the number of clusters across segments and is solved for $G = 10$.
- The constructive heuristic provides in 9 out of 27 instances (i.e. $\approx 33\%$) an initial solution which is not improved by the VNS heuristic. Moreover, it already starts the VNS with a upper bound solution better than that obtained by CPLEX in 23 out of 27 instances, i.e., $\approx 85\%$.
- CPLEX and VNS results are optimal for instance #26, which has the easiest configuration of all our experiments, with $m = 150$, $n = 18$ and $G = 2$ without any kind of perturbation. Instances #12 and #20 are proved to be optimally solved too by means of the upper bound solutions obtained by the

Table 2: Results obtained by CPLEX and the VNS heuristic for the 27 instances generated for Montecarlo simulation

Instance	CPLEX				VNS			
	Best lower bound	Best upper bound	#bbnodes	GAP	CH	Average upper bound	std dev	GAP
1	2705.93	-	1	-	3257.67	3208.26	0.97	15.66%
2	2336.58	2389.32	8310	2.21%	2388.25	2388.25	0.00	2.16%
3	4398.94	4607.07	1097	4.52%	4604.21	4604.21	0.00	4.46%
4	1823.61	1870.07	41186	2.48%	1996.80	1870.06	0.02	2.48%
5	3355.22	7353.26	1	54.37%	3936.21	3786.79	35.37	11.40%
6	1382.72	2396.45	276	42.30%	1525.20	1525.20	0.00	9.34%
7	2413.33	2651.01	3432	8.97%	2900.01	2651.01	0.00	8.97%
8	1470.84	1824.17	8	19.37%	1785.67	1583.05	21.87	7.09%
9	6654.00	8425.28	1	21.02%	7604.90	7383.52	19.08	9.88%
10	4995.30	7267.40	3	31.26%	6049.09	5681.36	2.79	12.08%
11	2500.14	4235.07	1	40.97%	2835.41	2835.41	0.00	11.82%
12	3749.99	4850.00	1	22.68%	3749.99	3749.99	0.00	0.00%
13	3064.39	7330.24	1	58.20%	3562.48	3562.48	0.00	13.98%
14	2905.84	4648.16	15	37.48%	3250.00	3083.87	3.53	5.77%
15	5644.12	6087.38	677	7.28%	6000.37	5761.11	0.06	2.03%
16	9670.02	10752.50	29	10.07%	10192.50	9870.57	0.41	2.03%
17	2671.69	4127.72	1601	35.27%	3441.24	3139.15	11.96	14.89%
18	5958.80	8448.21	1	29.47%	6497.31	6497.31	0.00	8.29%
19	1190.01	1269.01	342	6.23%	1241.67	1206.01	0.00	1.33%
20	10935.00	12645.00	1	13.52%	10935.00	10935.00	0.00	0.00%
21	3405.44	3726.86	111	8.62%	3709.92	3594.90	0.52	5.27%
22	4233.50	4710.93	1871	10.13%	4929.88	4611.65	0.11	8.20%
23	5312.94	7262.72	5	26.85%	5987.09	5675.08	0.83	6.38%
24	3212.93	4834.31	3	33.54%	3945.64	3778.96	15.38	14.98%
25	1141.32	1460.81	1752	21.87%	1389.30	1247.82	0.23	8.53%
26	1874.99	1874.99	1	0.00%	1874.99	1874.99	0.00	0.00%
27	8657.60	12690.00	1	31.78%	9207.00	8886.92	50.15	2.58%

constructive heuristic. These three instances are particularly easy since their original configurations are not perturbed. Despite that, CPLEX was able to close the integrality gap for only one of them in the given time limit.

- The MIP formulation of Section 2 is quite difficult to be solved as revealed by the number of branch-and-bound nodes solved by CPLEX within 24 hours. In 10 of the 27 instances ($\approx 37\%$), CPLEX was able to solve only the root node. This fact affects directly the quality of the lower bounds reported in the table, making the gap for the upper bound solutions obtained by the VNS heuristic quite large in some instances, e.g. #17 and #24.

4.2 Data recovery results

We compare the classification robusteness of the HCP with the so-called *heterogeneous p-median problem* (HPM) (Blanchard et al., 2012b). Both models aim to find a partition of subjects in segments with common clustering structures. However, the models are conceptually distinct since: (i) whereas the number of medians in a segment is conditioned to subjects membership in HCP, it is a variable in HPM; (ii) HPM is a multi-objective model converted to a mono-objective model in its essence; it weights the sum of dissimilarities between each object and its assigned cluster, conditional on segment membership, and the difference between the number of medians of each subject and the estimated number of medians. Fact (ii) is critical in HPM because it requires the user of the model to select the "appropriate" value for the weight parameter which effectively creates the scale difference between its two objectives, which consequently biases the attention of any algorithm to solve the model. Our comparison contrasts the results obtained by the VNS heuristic of Section 3 and the VNS heuristic of Blanchard et al. (2012b) for the HPM using the same aforementioned computational platform. The weight used in HPM was set to the average of all dissimilarity values, roughly weighting both objectives equally.

To examine the models' ability to recover the original data of Table 1, we need metrics that would be independent of both the specification of the objective function and the number of decision variables. We determine that using the Adjusted Rand Index (ARI; Hubert and Arabie, 1985) that allow us to compare the ground-truth objects partitions from the Montecarlo design and the predicted clustering variables e, as well the subjects true partition in comparison with the predicted assignment variables z.

The last four columns of Table 3 indicate, respectively, the ARI index with respect to the recovery of the objects clustering (column $\overline{e}$) and the subjects segmentation (column $\overline{z}$) for both models in each of the instances of the Montecarlo simulation. The ARI is calculated for each subject, and the numbers presented are the averages obtained across the data. For both heuristics, we fixed the allowed computational time to 600 seconds on a single computer thread.

Table 3: Monte Carlo simulation: Recovery results

							HCP		**HPM**	
				Perturbation			ARI		ARI	
Instance	Subjects	Segments	Objects	Medians	Dissimilarities	Medians	$\overline{z}$	$\overline{e}$	$\overline{z}$	$\overline{e}$
1	150	10	30	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.886	0.903	0.958	0.713
2	300	2	18	All 6	N(0, 0.1)	0	1.000	1.000	1.000	1.000
3	450	2	18	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000	1.000	1.000
4	150	2	18	All 3	N(0, 0.05)	N(0, 0.5)	0.947	0.988	0.947	0.616
5	450	10	18	All 6	N(0, 0.05)	N(0, 1)	0.928	0.911	0.985	0.892
6	150	10	18	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000	1.000	1.000
7	300	2	18	All 6	0	N(0, 0.5)	0.987	0.881	0.987	1.000
8	150	10	18	50 % 3, 50 % 6	0	N(0, 1)	0.166	0.979	0.167	0.182
9	300	10	30	All 3	N(0, 0.05)	N(0, 0.5)	0.700	0.787	0.852	0.727
10	450	6	18	All 3	N(0, 0.1)	N(0, 1)	0.771	0.844	0.783	0.267
11	150	6	30	All 6	N(0, 0.1)	0	1.000	1.000	1.000	1.000
12	300	10	18	All 3	0	0	1.000	1.000	1.000	1.000
13	450	10	18	All 6	N(0, 0.1)	0	1.000	1.000	1.000	1.000
14	300	6	18	50 % 3, 50 % 6	0	N(0, 1)	0.960	0.985	0.878	0.657
15	300	2	30	All 6	N(0, 0.05)	N(0, 1)	0.934	0.985	0.871	0.716
16	450	2	30	50 % 3, 50 % 6	0	N(0, 1)	0.879	0.969	0.293	0.846
17	300	6	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.900	0.958	0.711	0.818
18	300	6	30	50 % 3, 50 % 6	N(0, 0.05)	0	1.000	1.000	1.000	1.000
19	150	6	18	All 6	0	N(0, 0.5)	0.968	0.987	0.947	1.000
20	450	6	30	All 3	0	0	1.000	1.000	1.000	1.000
21	150	2	30	All 3	N(0, 0.1)	N(0, 1)	0.821	0.955	0.797	0.200
22	450	2	18	50 % 3, 50 % 6	N(0, 0.1)	N(0, 0.5)	0.956	0.990	0.247	1.000
23	450	6	18	All 3	N(0, 0.05)	N(0, 0.5)	0.783	0.882	0.860	0.668
24	300	10	18	All 3	N(0, 0.1)	N(0, 1)	0.751	0.862	0.879	0.417
25	150	6	18	All 6	N(0, 0.05)	N(0, 1)	0.890	0.913	0.868	0.897
26	150	2	18	All 3	0	0	1.000	1.000	1.000	1.000
27	450	10	30	All 6	0	N(0, 0.5)	0.889	0.926	0.946	1.000

With respect to the recovery of the objects clusterings, both models performed very well with an average ARI of .952 for HCP and .801 for HPM. However, a paired sample t-tests that the difference betwen HCP ($M = .952$, $SD = .059$) and HPM ($M = .801$, $SD = .263$) is significant ($t(26) = 3.20, p < .01$). In fact, excluding the 9 out of 27 easier trials where both models perfectly recovered the original data, HCP outperformed HPM in 14 out of 18 trials. HCP was thus clearly better at recovering the original clusters, though both were good. With respect to subjects segmentation, both models also perform very well. Namely whereas HCP obtains an average ARI of .893 ($M = .893$, $SD = .169$), HPM obtains .851 ($M = .851$, $SD = .236$). The difference between the two is not significant ($t(26) = 1.18, p = .25$).

4.3 Data sensitivity results

The sensitivity of the HCP and HPM models was evaluated by predicting each model's ARI using dummy-coded factors for each of the data characteristics. The results are displayed in Table 4 for the objects clustering variables and in Table 5 for the subjects segmentation variables. In both tables, the rows indicate the data structures characteristics that were manipulated in the 27 synthetic instances. Each row contains the main

Table 4: Monte Carlo simulation: Factors influencing the clustering of objects by the subjects

		HCP				HPM			
Factor		**Beta**	**t-value**	**p-value**		**Beta**	**t-value**	**p-value**	
Intercept		1.084	37.521	.000	***	1.081	37.826	.000	***
Subjects	300	-.030	-1.426	.174		-.023	-1.134	.275	
(default: 150)	450	-.023	-1.085	.295		.000	.005	.996	
Segments	6	-.022	-1.067	.303		-.028	-1.378	.188	
(default: 2)	10	-.045	-2.136	.050	**	-.008	-.387	.704	
Number of Objects	30	-.007	-.404	.692		.001	.051	.960	
(default: 18)									
Median Spread	All 3	-.052	-2.479	.026	**	-.091	-4.383	.001	***
(default: 50% 3, 50% 6)	All 6	-.020	-.964	.350		.041	1.971	.067	*
Added Error on Dissimilarities	Small	-.029	-1.397	.183		-.054	-2.598	.020	**
(default: none)	Large	-.024	-1.143	.271		-.105	-5.086	.000	***
Added Error on Number of Medians	Small	-.078	-3.718	.002	***	-.078	-3.800	.002	***
(default: none)	Large	-.067	-3.190	.006	***	-.159	-7.709	.000	***
R^2	0.678					0.893			
Adjusted R^2	0.448					0.882			
Mean ARI	0.952					0.913			
Std ARI	0.059					0.104			

* indicates p < .10, ** indicates p < .05, *** indicates p < .01.

Table 5: Monte Carlo simulation: Factors influencing segment membership

		PFVNS				VNS-HPM			
Factor		**Beta**	**t-value**	**p-value**		**Beta**	**t-value**	**p-value**	
Intercept		.956	9.216	.000	***	.985	9.278	.000	***
Individuals	300	.062	.822	.424		.079	1.031	.319	
(default: 150)	450	.059	.783	.446		.111	1.450	.168	
Segments	6	-.028	-.374	.714		.009	0.122	.905	
(default: 2)	10	-.134	-1.784	.095	*	-.031	-0.405	.691	
Number of Objects	30	.012	.181	.858		.018	0.278	.785	
(default: 18)									
Median Spread	All 3	.003	.038	.970		-.142	-1.855	.083	*
(default: 50% 3, 50% 6)	All 6	.094	1.259	.227		.116	1.507	.152	
Added Error on Dissimilarities	Small	.037	.492	.630		-.018	-0.239	.814	
(default: none)	Large	.026	.348	.733		-.095	-1.245	.232	
Added Error on Number of Medians	Small	-.109	-1.460	.165		-.134	-1.747	.101	**
(default: none)	Large	-.211	-2.816	.013	**	-.428	-5.589	.000	***
R^2	0.494					0.763			
Adjusted R^2	0.124					0.590			
Mean ARI	0.893					0.813			
Stdev ARI	0.170					0.254			

* indicates p <.10, ** indicates p < .05, *** indicates p < .01.

effects (beta coefficients) for the factors used as independent variables, along with the significance of the factor. Significant effects suggest that the model (HCP or HPM) is sensitive to the data characteristic. As few significant cofficients as possible is desired for a model to be robust.

First, we examine whether the models can recover the ground-truth subjects clusterings of the objects. With respect to HPM, we find that the model has some sensitivity to changes in data structures. Specifically whereas the model is unaffected by the number of subjects or the number of clusters, partitions with numerous clusters of equal sizes are better recovered than fewer clusters or those with uneven sizes. We also find that noise (even in small amounts) to both dissimilarities and to number of medians significantly affects ARI. The HCP, in contrast, is mostly unaffected by data structures. Of note, it is particularly insensitive to errors added on the distances. It is also better able to recover clustering structures with a larger number of objects,

and datasets when the number of segments is smaller than 10. That said, the impact of these factors is minimal as the mean ARI is .95 – a near perfect recovery of the original pairwise data.

Second, we examine whether the models can recover the segment memberships. We find that HPM has some difficulty in recovery original segment memberships when some (even small) error was added to the number of medians and when the clustering structures used in the data generating process were made of a few large clusters. The HCP is mostly unaffected when it comes to recovering segment memberships. It has marginally more difficulty recovering large segment memberships (when $G = 10$) and is only impacted by large error added to the number of medians.

It is important to remark that our analysis depends on the heuristics used on solving both models. In our case, we believe that the model comparison is fair enough since: (i) the same framework, i.e., VNS, was used; (ii) both heuristics were demonstrated to have good perfomance regarding the optimization of their respective models; and (iii) the results were collected using the same computational platform.

5 Empirical illustration: Understanding differences in perceptions of assortments of chocolate candy

To further demonstrate the usefulness and performance of the proposed procedure, we collected sorting task data for a real-world application and used the proposed VNS heuristic to illustrate heterogeneity in the clustering perfomed by different individuals. The sorting task (also known as card sorting) asks participants to allocate a set of objects into piles according to their own perception. It is common to instruct participants to 1) put objects into the same pile if they are similar in some way (there are no pre-determined labels), and 2) use as many piles as they desire. The pairwise similarity data provided by the sorting task is, in the most simplified way, $y_{ij}^k = 1$ if consumer k ($k = 1, ..., m$) places objects i and j ($i, j = 1, ..., n$) in the same pile (high similarity), 0 otherwise (low similarity). The task is particularly suited for the generation of such pairwise similarity data because the task mirrors closely the cognitive activities involved in the categorization process consumers follow as they form similarity judgments (Coxon, 1999), and it leads to as high quality data with less fatigue and boredom from participants as compared to pairwise similarity tasks (Bijmolt and Wedel, 1995; Rao and Katz, 1971).

Data was collected from an online sorting study featuring $m = 189$ undergraduate students from a large northeastern U.S. university who answered about their perceptions about $n = 20$ chocolate candies disposed at the Corp's Vital Vittles, the first storefront for Students of Georgetown Inc. opened in 1973. Today, Vital Vittles is a full-service grocery store which sells frozen foods, meals on the go, and a variety of home supplies. Because the Georgetown Campus housing is fairly isolated, it is considered a "one-stop shop": Vittles faces little competition from local grocery stores. It is also very healthy financially, with gross sales averaging over 2 million dollars a year and a 23% gross profit margin.

As part of its most prominent checkout counter shelves, Vital Vittles offers a large selection of chocolate candy. The shelf section dedicated to chocolate snacks includes the following options: Almond Joy, Baby Ruth, Butterfinger, Hershey (Almond), Hershey (Plain), Junior Mints, Kit Kat, M & M (Peanut), M & M (Plain), Mars Bar, Milky Way, Mounds Bar, Nestle's Crunch, Oh Henry!, Payday, Reece's Cups, Snickers, Three musketeers, Twix, and York Mint. As the selection changes frequently, so does the way the assortment is organized.

In order to be used by the HCP model, the sort done by each individual k, for $k = 1, \ldots, m$, is converted to a dissimilarity matrix D^k following the procedure proposed by Takane (1980), and the number of medians c^k is made equal to the number of piles made by that individual in the sorting task.

5.1 Model selection & performance

For both VNS heuristics, we performed 10 executions of the procedures. All executions were terminated after 600 seconds of CPU time and Table 6 shows the best of the objective functions as the number of segments (G) increases. To facilitate model selection, the table also shows the percentage improvement obtained when

Table 6: Chocolate snacks application: Model selection (G)

G	HCP	% Improvement	HPM	% Improvement
1	2493.59	-	2929.25	-
2	2317.92	7.05%	2542.14	13.22%
3	2300.35	0.76%	2417.27	4.91%
4	2283.16	0.75%	2365.69	2.13%
5	2276.37	0.30%	2333.86	1.35%
6	2266.53	0.43%	2316.06	0.76%
7	2262.32	0.19%	2306.09	0.43%
8	2256.51	0.26%	2292.62	0.58%

an additional segment is added. Both models seem to identify a solution with $G = 2$ segments, following the traditional "elbow in the curve" approach. The selection of $G = 2$ is particularly evident for HCP which produces minimal improvements when $G > 2$ (less than 1%).

Although the ARI measuring the agreement between the unknown ground-truth partition of subjects and the predicted subjects segmentation cannot be calculated, the sorting task allows us to obtain the ARI regarding the piles composition made by each individual and its predicted clustering of objects according to the HCP. In our application, an agreement exists if $y_{ij}^k = 1$ whereas the HCP predicts $z^{kg} = 1$ and $e_{ij}^g = 1$.

Among the two $G = 2$ solutions, we can state that HCP performed slightly better. Namely whereas HPM obtained $ARI = .2931$, HCP obtained $ARI = .3143$ - an improvement of 7.23%. This difference in fit comes primarily from a difference in assignments of subjects to segments. When comparing variables z predicted by both models, we find significant differences ($ARI = .3413$).

5.2 Interpreting the segment-level clustering structures

The solution for HCP is presented in Table 7. The solution is composed of two segments, *Novices* (Segment 1; #individuals = 69) and *Experts* (Segment 2; #individuals = 120). As compared to members of the expert segment, members of the novice segment reported (1-strongly disagree, 5-strongly agree) being less confident of their piles ($M_{Novices} = 4.55; M_{Experts} = 4.92; t(184) = 1.72, p = .09$) and more likely to enjoy salty foods more than sweets ($M_{Novices} = 1.79; M_{Experts} = 1.42; t(184) = 1.68, p = .09$) than experts. Furthermore, when asked about their knowledge of such snacks compared to their peers, they were less likely to agree when compared to others, that they know more about chocolate ($M_{Novices} = 2.94; M_{Experts} = 3.19; t(184) = 1.72, p = .09$).

Table 7: Chocolate snacks application: Segment-level partition structure

Segment 1 - Novices

Cluster Label	Cluster Median	Members				
Mint	Junior Mints	York Mint				
Tablets	Hershey (Plain)	Kit Kat	Hershey (Almond)	Nestle's Crunch		
Small Candy	M & M (Peanut)	M & M (Plain)	Reese's Cups			
More Popular Bars	Snickers	Butterfinger	Milky Way	Three Musketeers	Twix	
Less Popular Bars	Mounds Bar	Almond Joy	Baby Ruth	Mars	Oh Henry!	Payday

Segment 2 - Experts

Cluster Label	Cluster Median	Members		
Mint	Junior Mints	York Mint		
Nougat Chocolate Bars	Milky Way	Mars	Snickers	Three Musketeers
Almond/Coconut	Almond Joy	Hershey (Almond)	Mounds Bar	
Cripsy	Kit Kat	Nestle's Crunch	Twix	
Peanut Butter	Reese's Cups	Butterfinger		
Peanuts & Caramel	Payday	Baby Ruth	Oh Henry!	
Coated Chocolate	M & M (Plain)	Hershey (Plain)	M & M (Peanut)	

5.2.1 Segment 1: Novices

The partition structure of the *Novices* segment contains 5 clusters. Because members of this segment are less frequent consumers of chocolates, their clusters are largely structured around either attributes (main ingredient) or awareness. For this segment, Cluster 1 include the *mint based candy* of Junior Mints and York Mint - an attribute salient in the product names. Cluster 2 includes *tablets* (thinner and wider) Hershey plain (median), Kit Kat, Hershey (Almond) and Nestle's Crunch. Cluster 3 contains the *small candy* of M&M Peanuts (median) M&M Regular and Reese's cups.

Both Clusters 4 and 5 involve chocolate covered candy, yet they differ based on their perceived popularity among the participants. Cluster 4 contains the *more popular bars*, a cluster for which all the members of segment knew all the items. Snickers, the most sold chocolate candy bar in the world, is most representative of the cluster. Other members include Butterfinger, Milky Way, 3 Musketeers, and Twix. lesser known chocolate bars such as Mounds (median), Almond Joy, Baby Ruth, Mars, Oh Henry! and Payday. On average, 99.13% of cases were each of these bars known to members of this novice segment. This is further confirmed by the results of our survey that followed the sorting task. Namely, on average each candy in this cluster had a 16.23% chance of being consumed once a month or more. Cluster 5, in contrast, includes *less popular bars* such as Mounds (median), Almond Joy, Baby Ruth, Mars, Oh Henry! and Payday. The results of the survey suggest that, on average, 28% of these bars were unknown to members of this novice segment. Further, in only 3% of the cases were candy in this cluster consumed once a month or more.

5.2.2 Segment 2: Experts

Just as novices, members of the *Experts* segment also have a cluster composed of *mint based candies* which includes Junior Mints and York Mint. However, their partition structure is generally more complex starting with the presence of 2 additional clusters. For the *Experts*, cluster 2 is composed of *nougat chocolate bars* including Milky Way (median), Mars, Snickers and Three Musketeers. Cluster 3 focuses on almond & coconut based chocolates, with Almond Joy (median), Hershey (Almond) and Mounds Bar as members of the cluster. Cluster 4 includes *crispy candy* of Kit Kat (median), Nestle's Crunch, and Twix, all candy known for their cripsiness. Cluster 5 includes Reese's cups (median) and butterfinger, two chocolates known for their *peanut butter* flavors. Cluster 6 includes Payday (median) and Baby Ruth, and Oh Henry, two candy bars heavily focused on *peanuts and caramel*. Finally, their Cluster 7 includes popular *coated chocolates* M&M, Hershey Plain, and M&M Peanuts.

5.3 Illustrative implications

Comparing the partition structures of the two segments brings important implications for the retailer. First, it seems that infrequent chocolate buyers pay more attention to physical cues such as the size (small, tablet, bar) than more frequent buyers. They also tend to know fewer brands and thus distinguish between the popular and less popular candy in the first stage. As previous research has shown, consumers with low store knowledge and with significant time pressure tend to be the most likely to be lead to unplanned purchases (Park et al., 1989) and that the assortment structure chosen can influence similarity perceptions and willingness to pay (Lamberton and Diehl, 2013), we would first recommend that Vital Vittles organizes a section of its checkout counter candy display along the lines of the clusters encountered by novices (a large sample of busy students). Second, frequent buyers (experts) tend to think of chocolate as a function of their key ingredients, and the salient ones seem to be more about the filling rather than the coatings or shape. An organization that helps label these chocolates for quick identification (e.g., stickers that indicate that the chocolate contains nougat) could prove to be especially helpful in directing this audience.

6 Concluding remarks

We presented a new model for clustering data collected from heterogeneous subjects. It simultaneously selects medians among the available objects assigning objects to them at the same time identifying segments of sub-

jects with similar clustering structures. The model incorporates key data from the business/marketing/data mining context for which classical clustering fails to deal with, thus leading to poor or even erroneous analyses.

The model is nonconvex and difficult to be tackled by state-of-the-art nonconvex solvers as well as by MIP solvers after a convexification of the model is achieved via typical Fortet's reformulation strenghtened by RLT-cuts. Consequently, a Variable Neighborhood Search heuristic is designed in order to tackle practical instances of the problem. Its local search relies on the solution of different MIP models consisting of three local descents which are embedded into the Variable Neighborhood Descent framework. Our computational experiments show that the VNS heuristic outperforms general purpose exact solvers widely used, providing better (or equivalent) solutions in all tested cases in much less computing time.

Through a Monte Carlo simulation, we demonstrate the model's capacity to recover heterogeneous data, confirmed by average Adjusted Rand Indices always superior to 0.89, and its (in)sensitivity to various latent clustering settings. Finally, through an empirical illustration that involves consumer sorts of chocolate candy, we illustrate how the model can be used to gain insights into perceptions of a set of brands for segments of consumers who vary in their expertise with the product category. Thus, the present paper contributes to a growing literature that focuses on the role of unobserved clusters in the consumer decision making processes (e.g. Noseworthy et al., 2012; Swait et al., 2014).

References

Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A., 2009. Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods and Software 24, 597–634.

Bettman, J.R., Luce, M.F., Payne, J.W., 1998. Constructive consumer choice processes. Journal of Consumer Research 25(3), 187–217.

Bettman, J.R., Park, C.W., 1980. Effects of prior knowledge and experience and phase of the choice process on consumer decision processes: A protocol analysis. Journal of Consumer Research 7 (Dec.), 234–248.

Bijmolt, T.H., Wedel, M., 1995. The effects of alternative methods of collecting similarity data for multidimensional scaling. International Journal of Research in Marketing 12(4), 363–371.

Blanchard, S., 2011. A methodology for identifying unobserved categories when consumers assign brands to multiple categories. Ph.D. thesis, The Pennsylvania State University.

Blanchard, S., DeSarbo, W., Atalay, A., Harmancioglu, N., 2012a. Identifying consumer heterogeneity in unobserved categories. Marketing Letters 1, 177–194.

Blanchard, S.J., Aloise, D., DeSarbo, W.S., 2012b. The heterogeneous p-median problem for categorization based clustering. Psychometrika 77(4), 741–762.

Blanchard, S.J., DeSarbo, W.S., 2013. A new zero-inflated negative binomial methodology for latent category identification. Psychometrika 78(2), 322–340.

Brusco, M.J., Cradit, J.D., 2001. A variable-selection heuristic for k-means clustering. Psychometrika 66(2), 249–270.

Brusco, M.J., Cradit, J.D., 2005. Conpar: a method for identifying groups of concordant subject proximity matrices for subsequent multidimensional scaling analyses. Journal of Mathematical Psychology 49(2), 142–154.

Coxon, A.P.M., 1999. Sorting data: Collection and analysis. Vol. 127. Sage.

Daws, J., 1996. The analysis of free-sorting data: Beyond pairwise co-occurrence. Journal of Classification 13, 57–80.

DeSarbo, W., Atalay, A., LeBaron, D., Blanchard, S., 2008. Estimating multiple consumer segment ideal points from context dependent survey data. Journal of Consumer Research 35(1), 142–153.

DeSarbo, W., Carroll, J., 1985. Three-way metric unfolding via alternating weighted least squares. Psychometrika 50, 275–300.

Fisher, R., 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics VII, 179–188.

Fortet, R., 1960. Applications de l'algèbre de Boole en recherche opérationelle. Revue Française de Recherche Opérationelle 4(14), 17–26.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315, 972–976.

Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. Mathematical Programming 79, 191–215.

Hansen, P., Mladenović, N., 2001. Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3), 449–467.

Hansen, P., Mladenović, N., J.A.M., P., 2008. Variable neighborhood search: Methods and applications. 4OR 6, 319–360.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2(1), 193–218.

ILOG, I., 2012. Inc. CPLEX 12.5 user manual.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: A review. ACM Computing Surveys (CSUR) 31(3), 264–323.

Kariv, O., Hakimi, S.L., 1979. An algorithmic approach to network location problems. i: The p-centers. SIAM Journal on Applied Mathematics 37(3), 513–538.

Kleinberg, J., 2003. An impossibility theorem for clustering. Advances in Neural Information Processing Systems, 463–470.

Köhn, H.-F., Steinley, D., Brusco, M.J., 2010. The p-median model as a tool for clustering psychological data. Psychological Methods 15(1), 87–95.

Lamberton, C.P., Diehl, K., 2013. Retail choice architecture: The effects of benefit-and attribute-based assortment organization on consumer perceptions and choice. Journal of Consumer Research 40(3), 393–411.

Lee, M., 2001. Determining the dimensionality of multidimensional scaling representations for cognitive modeling. Journal of Mathematical Psychology 45, 149–166.

McLachlan, G.J., Basford, K.E., 1988. Mixture models: Inference and applications to clustering. M. Dekker, New York, NY.

Milligan, G., Cooper, M., 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika 50, 159–179.

Mladenović, N., Brimberg, J., Hansen, P., Moreno-Prez, J.A., 2007. The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research 179(3), 927–939.

Noseworthy, T.J., Wang, J., Islam, T., 2012. How context shapes category inferences and attribute preference for new ambiguous products. Journal of Consumer Psychology 22(4), 529–544.

Park, C.W., Iyer, E.S., Smith, D.C., 1989. The effects of situational factors on in-store grocery shopping behavior: the role of store environment and time available for shopping. Journal of Consumer Research 15(4), 422–433.

Rao, V.R., Katz, R., 1971. Alternative multidimensional scaling methods for large stimulus sets. Journal of Marketing Research 8(4), 488–494.

Resende, M.G., Werneck, R.F., 2004. A hybrid heuristic for the p-median problem. Journal of Heuristics 10(1), 59–88.

Sherali, H.D., Alameddine, A., 1992. A new reformulation-linearization technique for bilinear programming problems. Journal of Global Optimization 2(4), 379–410.

Shugan, S.M., Sep. 1980. The cost of thinking. Journal of Consumer Research 7, 99–111.

Simon, H.A., 1955. A behavioral model of rational choice. the Quarterly Journal of Economics 69(1), 99–118.

Steinley, D., Hendrickson, G., Brusco, M., 2015. A note on maximizing the agreement between partitions: A step-wise optimal algorithm and some properties. Journal of Classification, in press.

Swait, J., Brigden, N., Johnson, R.D., 2014. Categories shape preferences: A model of taste heterogeneity arising from categorization of alternatives. Journal of Choice Modelling, 13, 3–23.

Takane, Y., 1980. Analysis of categorizing behavior using a quantification method. Behaviormetrika 7, 75–86.

Vichi, M., Rocci, R., Kiers, H., 2007. Simultaneous component and clustering models for three-way data: Within and between approaches. Journal of Classification 24, 71–98.