Psychological Methods
2010, Vol. 15, No. 1, 87–95

1082-989X/10/$12.00 DOI: 10.1037/a0018535

The *p*-Median Model as a Tool for Clustering Psychological Data

Hans-Friedrich Köhn and Douglas Steinley
University of Missouri—Columbia

Michael J. Brusco
Florida State University

The *p*-median clustering model represents a combinatorial approach to partition data sets into disjoint, nonhierarchical groups. Object classes are constructed around *exemplars*, that is, manifest objects in the data set, with the remaining instances assigned to their closest cluster centers. Effective, state-of-the-art implementations of *p*-median clustering are virtually unavailable in the popular social and behavioral science statistical software packages. We present *p*-median clustering, including a detailed description of its mechanics and a discussion of available software programs and their capabilities. Application to a complex structured data set on the perception of food items illustrates *p*-median clustering.

Keywords: combinatorial data analysis, cluster analysis, *p*-median problem, Lagrangian relaxation, heuristics

Supplemental materials: http://dx.doi.org/10.1037/a0018535.supp

In general terms, clustering might be characterized as a collection of methods concerned with the identification of homogenous groups of objects based on whatever data are available. Cluster analysis represents an obvious choice for developing a taxonomy for any kind of object domain, be that participants, psychological disorders, test and questionnaire items, experimental stimuli, behavioral patterns, and so forth. Cluster analysis can also serve as a preliminary means to identify potential group differences in a sample, subsequently addressable through hypotheses-driven statistical analysis. However, for many statisticians, the "shady history" of early approaches to clustering, "usually just convenient algorithms devoid of any associated representational model or effort at optimizing a stated criterion" (p. 9) was a long-standing cause of concern and reservation, as Arabie and Hubert (1996) observed more than a decade ago.

The situation today is markedly different: A researcher can choose among a large variety of clustering methods ranging from the "devoid-type" still found in some commercial software packages to state-of-the-art, high-end implementations. Among the latter, we make particular note of Mplus (Muthén & Muthén, 1998–2007) and Latent GOLD (Vermunt & Magidson, 2005), which offer model-based clustering and a diverse collection of discrete latent-class methods that can fairly be subsumed as special instances of clustering. Many noncasual cluster analysis users in psychology have moved to sophisticated statistical computational environments such as MATLAB or R (the latter of which is freely available). Both platforms provide the flexibility to program special purpose routines/toolboxes that are often "open source." Consider, for example, the MCLUST toolbox in R offered by Fraley and Raftery (2006) for model-based clustering (see McLachlan & Basford, 1988, and McLachlan & Peel, 2000, for comprehensive presentations of model-based methods). Similarly, Hubert, Arabie, and Meulman (2006, Chapters 5–8) provide a suite of MATLAB programs for combinatorial (graph–theoretic) clustering through fitting ultrametric and additive tree structures (see also Hubert, Köhn, & Steinley, 2009). The books by Martinez and Martinez (2005, 2007) provide detailed technical descriptions and user instructions for the up-to-date mainstream clustering algorithms implemented in the MATLAB statistics toolbox, as well as for less common clustering routines available in their own toolboxes. MATLAB programs for various combinatorial clustering problems have also been developed by Steinley (2003, 2006b); Brusco and Cradit (2005); and Brusco and Steinley (2006).

Among advanced (multivariate) statistical methods, clustering probably represents the least abstract and intuitively most accessible procedure, as its rationale emulates a major cognitive process: concept-based object categorization. In fact, the classic dichotomy of theories in concept research, prototype and exemplar models (see Murphy, 2002; Ross, Taylor, Middleton, & Nokes, 2008), attribute to classificatory acts distinct structures and principles of operation having direct counterparts in the logic and computational mechanics of certain clustering algorithms.

Prototype models conceptualize object categorization as the result of evaluating and integrating information about all the possible properties of an item in reference to the prototype of a category, an abstraction of the features shared by all its instances: "Mathematically, the prototype is the average or central tendency of all category members" (Love, 2003, p. 648). A novel object is postulated to be assigned to the category centered around the

Hans-Friedrich Köhn and Douglas Steinley, Department of Psychological Sciences, University of Missouri—Columbia; Michael J. Brusco, Department of Marketing, College of Business, Florida State University.

We express our appreciation to Brian Ross, University of Illinois at Champaign–Urbana and Gregory Murphy, New York University for the generous permission to use their data as illustration. In addition, we are indebted to Lawrence Hubert, University of Illinois at Urbana–Champaign, for his insightful comments on a draft of this article. Douglas Steinley was supported by Grant K25AA017456 from the National Institute on Alcohol Abuse and Alcoholism.

Correspondence concerning this article should be addressed to Hans-Friedrich Köhn, Department of Psychological Sciences, 19 McAlester Hall, University of Missouri, Columbia, MO 65211-2500. E-mail: koehnh@missouri.edu

prototype most similar to the item. The observation of additional instances of a category can induce an update of the prototype feature profile. Eventually, in the case of a poor match to all existing prototypes, a newly encountered object might establish a category of its own. Most notably, the prototype of a category can be a virtual object that does not even need to exist. Consequently, the prototype model of object categorization corresponds exactly to the computational logic of the popular *K*-means clustering method (Forgy, 1965; Hartigan & Wong, 1979; Kogan, 2007; MacQueen, 1967; Steinhaus, 1956; for comprehensive reviews, see Bock, 2007; Steinley, 2006a) that produces a partition of a set of objects into exhaustive and disjoint/nonhierarchical groups based on measures on some variables characterizing the objects.

The exemplar view on object categorization refutes the idea of a representation encompassing an entire concept that summarizes all individual instances of a category. Rather, a person's concept is postulated to consist of the entire set of category members ever encountered and remembered. A novel object is classified in the category where the total sum of its similarities to all recalled exemplars is largest. As Murphy (2002) illustrates:

"The Irish terrier in my yard is extremely similar to some dogs that I have seen, is moderately similar to other dogs, but is mildly similar to long-haired ponies and burros as well. It has the same general shape and size as a goat, though lacking the horns or beard. It is in some respects reminiscent of some wolves in my memory as well. How do I make sense of all these possible categorizations: a bunch of dogs, a few goats, wolves, and the occasional pony or burro?—When you add up all the similarities, there is considerably more evidence for the object's being a dog than for its being anything else" (p. 49).

Does the exemplar model of object categorization translate directly into a statistical clustering method like the prototype model into *K*-means? Not exactly. Yet, techniques known as exemplar-based clustering represent the closest equivalent to object categorization in light of the exemplar view: Given a set of objects, a subset is selected as cluster centers (exemplars), and the remaining objects are allocated to their most similar exemplar such that a given loss criterion is optimized (e.g., maximizing the total sum of similarities between exemplars and "satellites"). For the corresponding clustering method, Kaufman and Rousseeuw (2005) coined the term "partitioning around medoids" (PAM)—in the operations research community, more commonly known as the "*p*-median model"—(see Hanjoul & Peeters, 1985; Kuehn & Hamburger, 1963; Maranzana, 1964; Mladenović, Brimberg, Hansen, & Moreno-Pérez, 2007). Like *K*-means, *p*-median clustering generates a disjoint, nonoverlapping partition of a set of objects. In further exploiting the analogy to the competing prototype and exemplar models in cognitive theory, we mention that quantitative researchers have recurrently advocated *p*-median clustering as a viable alternative for partitioning a data set (Alba & Domínguez, 2006; Brusco & Köhn, 2008a, 2008b; Hansen & Mladenović, 2008; Klastorin, 1985; Mulvey & Crowder, 1979; Rao, 1971; Vinod, 1969). The centers of the *p* groups, called *medians*, represent manifest objects, with clusters built around them, thus, often facilitating substantive interpretation considerably. As an illustration, consider clustering instructional institutions in educational psychology: Groups constructed around existing schools offer an immediate and vivid picture as opposed to the "virtual" centers found when using a prototype-based clustering model. Alternatively, recall Murphy's (2002) example of categorizing dogs: Classes that are characterized by centroids of variables representing weight, height, maximal running speed, costs of keeping, friendliness–aggressiveness ratings, life expectancy, and so forth might be less catching and intuitively accessible than clusters centered around actual dog breeds, such as the German shepherd, poodle, boxer, or pekingese. In addition, *p*-median clustering is applicable to a wide range of data formats, be those square-symmetric/-asymmetric or rectangular proximity matrices, whereas most clustering algorithms are typically constrained either to square-symmetric proximity matrices (in the case of hierarchical clustering) or to a standard, rectangular data matrix (in the case of *K*-means clustering or model-based clustering). Lastly, Kaufman and Rousseeuw (2005, Chapter 2) emphasized the remarkable robustness of the *p*-median approach to outliers.

The unavailability of state-of-the-art implementations of *p*-median clustering through popular social and behavioral science statistical software packages might constitute the main reason for the lack of its awareness among researchers (sole exception: Kaufman and Rousseeuw, 2005, provided the code written in R for *p*-median clustering). In an attempt to make *p*-median clustering more accessible, we offer an introduction to *p*-median clustering that hopefully will help bridge the gap between the theory of clustering based on exemplars and the pragmatic needs of a sophisticated user in psychological research. Additionally, for this purpose, we have developed a suite of user-friendly MATLAB routines for *p*-median clustering, accompanied by an instruction manual; these are available as online supplemental materials to this article. In the next section we briefly review several integral concepts, such as proximity data, loss function, exact/approximate algorithms, and *p*-median clustering, as a combinatorial optimization problem. A detailed description of the rationale underlying the *p*-median model, as well as a brief, nontechnical introduction to computational algorithms is given in the succeeding section. Application to a real-world data set on the perception of a vast collection of food items illustrates *p*-median clustering. We conclude with a discussion of the specific merits of *p*-median clustering and recommendations for practical applications.

Theoretical Preliminaries: Concepts and Terminology

Proximities

Tversky's (1977) seminal paper on similarity gives a most thorough theoretical treatment of the concept, emphasizing its eminent and ubiquitous role in psychological "theories of knowledge and behavior" (p. 327). Similarity and its complement, dissimilarity, are typically subsumed under the notion of proximity. In its broadest sense, the term *proximity* refers to any numerical measure of relationship between the elements of a pair from two (possibly distinct) sets of entities or objects. Proximities are typically collected into a matrix, with rows and columns representing the respective sets of objects; the numerical cell values denote the observed pairwise proximity scores. By assumption, proximities are restricted to be nonnegative and are, henceforth, consistently interpreted as dissimilarities so that larger numerical indices pertain to less similar pairs of objects. Cluster-analytic methods depend on proximity data as key information and basis for identifying maximally homogenous subgroups.

Objective Functions and Algorithms

The least-squares loss function of the simple (linear) regression model can be considered a common example of an "objective function," in which estimation of model parameters is determined by the objective of minimizing the sum of the squared deviations (residuals) between estimated and observed criterion values. In general, an objective function quantifies how well a fitted model approximates the original data. Thus, model estimation governed by an objective function offers the enormous advantage of a solid criterion for evaluating the quality of an obtained solution and is currently considered a mandatory standard for statistical modeling, including the task of clustering a data set.

Loosely speaking, the global optimum of an objective function denotes its absolute minimum/maximum value across the entire set of admissible (feasible) solutions, as opposed to a local optimum pertaining to the minimum/maximum on a subset of the solution space; the associated solutions are said to be globally or locally optimal. The former simply indicates that we cannot "do any better" for a given data-analytic task, whereas the latter, typically, but not necessarily, is inferior to the globally optimal solution.

Related to the definitions of optima are the solution methods, or algorithms, themselves. Specifically, for optimization problems, algorithms can be broadly divided into two classes: exact and approximate. Exact algorithms produce guaranteed globally optimal solutions, whereas, approximate algorithms cannot provide this guarantee (in the jargon of operations research, approximate methods are typically referred to as "heuristics," because they employ clever computational shortcut strategies for solving a nontrivial optimization problem such that the obtained solution is at least locally optimal).

p-Median Clustering as a Discrete Optimization Problem

The *p*-median clustering model originated in operations research from attempts to optimize the planning of facility locations (Hanjoul & Peeters, 1985; Kuehn & Hamburger, 1963; Maranzana, 1964; Mladenović et al. 2007). For example, consider the task of rolling out a network of medical emergency wards in a densely populated area with multiple candidate sites. Budget constraints limit the actual number of facilities to be installed; at the same time, the choice of locations should guarantee maximal accessibility within minimal time for the entire population across all communities. The search for the most suitable facility sites requires the evaluation of numerous combinations of community assignments to potential facility locations, known in operations research as the *p*-median facility location problem. Exemplar-based clustering via the *p*-median model is molded from this optimization problem. Specifically, *p*-median clustering of a set of N objects encompasses two tasks: (a) identifying a set of p objects serving as cluster centers and (b) assigning the remaining N–p objects (satellites) to the chosen medians such that the objective function of the total sum of median-to-satellite dissimilarities (e.g., Euclidean distances) is minimized. Obviously, the two tasks are interdependent: The choice of medians determines the set of remaining objects, which, vice versa, affects the final value of the objective function.

A seemingly plausible strategy for finding the best representation for a data set on N objects through p clusters would aim to evaluate all possible choices of p medians and subsequent assignments of each of the N–p remaining objects to one and only one median; the solution providing a global optimum of the objective function is chosen. More succinctly, *p*-median clustering represents a combinatorial optimization problem (*p*-median clustering is often also referred to as a discrete optimization problem because of the exclusive assignment of objects to a single median—the terms *combinatorial* and *discrete optimization* are used interchangeably in the literature). For combinatorial optimization problems, a globally optimal solution always exists (given that, in technical terms, the set of feasible solutions is finite), implying the misrepresentation that these problems are "easy" and solvable through the explicit search of the entire solution space (i.e., "complete enumeration"). Yet, despite the striking simplicity of its rationale, *p*-median clustering poses a very difficult partitioning problem because, for a particular choice of p, the globally optimal solution must be identified among $\binom{N}{p}$ different candidates. Hence, from a certain number of objects onward, complete enumeration does not offer a realistic option for minimizing the objective function, as the number of distinct partitions of N objects into p clusters increases approximately as $p^N/p!$—the number of feasible solutions grows exponentially with problem size. Thus, even for small-scale problems, the computational effort of an exhaustive enumeration of all possible object groupings is not computationally feasible for most practical applications.

Yet how does one solve realistically sized *p*-median clustering tasks? First, we observe, as a further complication, that combinatorial optimization problems are characterized by nonsmooth objective functions involving discrete or integer variables; therefore, familiar calculus-based optimization techniques are not applicable. Second, exact solution methods for the *p*-median problem exist that use sophisticated partial enumeration strategies, such as dynamic programming (see Hubert, Arabie, & Meulman, 2001, Chapter 3), and branch-and-bound methods (Brusco & Stahl, 2005, Chapters 2–5). Such methods can often facilitate globally optimal solutions of larger problems, without the need for explicit enumeration of the entire solution set, but do face serious limitations on the size of problems that can be handled. Therefore, approximate solution algorithms (heuristics) remain necessary for *p*-median clustering tasks of practical size, with no guarantee of identifying a global optimum, but often producing solutions at least within a close neighborhood of the desired global optimum.

Performing *p*-Median Clustering

In its most general form, *p*-median clustering can be used for data represented by a rectangular matrix containing proximities between two distinct sets of entities (such as in the previous example, where cell entries correspond to distances between communities and location candidates). Many data sets in the social sciences, however, focus on documenting the relationship between entities from a single set, often expressed as pairwise interobject dissimilarities and collected into a square-symmetric proximity matrix. As an aside, pairwise dissimilarity scores can either be directly elicited from participants, say, through ratings on a scale, or derived from integrating multiple attribute ratings. Without loss of generality, we develop the logic of p-median clustering in application to square-symmetric proximity matrices.

Concretely, the p-median clustering problem can be formulated as a (linear) integer programming (IP) problem and formalized through an objective function (comparable to a loss function) subject to constraints imposed on the function variables (often referred to as *decision variables*). The specific IP formulation of p-median clustering is given as the minimization problem

$$\text{(IP)} \qquad \min_{\mathbf{X}} \left\{ f(\mathbf{X}) = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij} \right\},$$

subject to constraints:

$$\sum_{j=1}^{N} x_{jj} = p, \tag{1}$$

$$x_{ij} \leq x_{jj} \ \forall \ i, j, \tag{2}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \ \forall \ i, \tag{3}$$

$$x_{ij} \in \{0, 1\} \ \forall \ i, j, \tag{4}$$

where d_{ij} represents the given input proximities and x_{ij} denotes binary decision variables restricted to take on only values of zero or one (see Constraint 4, hence, the name *integer program*). Let $\mathbf{D}$ represent the collection of interobject dissimilarities, $\mathbf{D}_{N \times N} = \{d_{ij}\}_{N \times N}$. Then the decision variables, x_{ij}, are collected into an analogous matrix, $\mathbf{X}_{N \times N} = \{x_{ij}\}_{N \times N}$. The decision variables in $\mathbf{X}$ split into two sets: The first, $\{x_{jj}\}$, refers to the entries along the main diagonal of $\mathbf{X}$. These variables represent the candidate medians (the subscript j follows from the convention that medians are chosen among the column objects). The second set of decision variables, $\{x_{ij}\}_{i \neq j}$, denotes the off-diagonal entries in $\mathbf{X}$; these indicate whether a remaining (row) object O_i is assigned to a median O_j. Because the decision variables are constrained to values of zero or one, they operate as "switches turning on or off" a specific object, either as a median or a satellite. If a decision variable is set to a value of one, then the dissimilarity value in the corresponding cell in $\mathbf{D}$ enters the objective function. As a technical detail, notice that an object selected as a median itself always contributes a value of $d_{jj} = 0$ to the objective function, with the immediate implication that any p-median clustering problem can be "solved" trivially by simply setting $p = N$. In other words, changing the problem structure from p to $p + 1$ will automatically decrease the loss function.

Constraints 1, 2, and 3 ensure that the choice of the 0–1 values for the decision variables conform to the identification of a feasible solution. First, the sum of the x_{jj} variables along the main diagonal must equal p to guarantee that exactly p objects are selected as medians (see Constraint 1). Second, the value of any decision variable, x_{ij}, can at most equal the value of the median variable, with corresponding index j, x_{jj}. Thus, a remaining object, O_i, can be assigned to median j if and only if object O_j has been selected as a median (see Constraint 2). Third, for each row object, O_i, of $\mathbf{X}$, the sum of the x_{ij} variables across the j columns is limited to equal 1, which translates into the requirement that a remaining object can only be assigned to one median (i.e., multiple assignments are blocked, because the resulting sum across columns then would exceed one; see Constraint 3). Note that for selected medians this condition is automatically fulfilled because in such cases x_{jj} always equals 1.

As an example, consider in Figure 1 the display of a set of $N =$ 9 objects located in a plane. For illustrative purposes, assume that the goal, then, is to assign these nine objects to $p = 3$ medians. We must evaluate $\binom{9}{3} = \frac{9!}{6!3!} = 84$ different choices for selecting cluster centers (note that a candidate solution is well-defined as soon as a particular set of medians has been selected, because the subsequent assignment of remaining objects to medians is fully determined by their row minima across median columns).

The globally optimal solution for $p = 3$ medians (represented as encircled dots) is shown in Figure 1, with the total sum of median-to-satellite Euclidean distances (represented as the lines connecting the satellites to the medians) equaling .77 + .70 + .65 + .89 + .86 + 1.07 = 4.94, which corresponds to the partition $O = \{C_1, C_2, C_3\} = \{\{\mathbf{2}\},\{\mathbf{5}\},\{\mathbf{9}, 1, 3, 4, 6, 7, 8\}\}$.

Figure 2 presents matrices $\mathbf{D}$ and $\mathbf{X}$ side by side to illustrate the underlying mechanics for the globally optimal solution of our small-scale example, {{2},{5},{9, 1, 3, 4, 6, 7, 8}}. Observe that only entries x_{22}, x_{55}, and x_{99} along the main diagonal of $\mathbf{X}$ (corresponding to medians 2, 5, and 9) equal one. The distribution of 0–1 values among the off-diagonal cells indicates that all remaining objects, 1, 3, 4, 6, 7, and 8, have been assigned to median object 9. The value of the objective function is simply computed as the total sum of products of corresponding cells in $\mathbf{D}$ and $\mathbf{X}$: $f(\mathbf{X}) = d_{11}x_{11} + d_{21}x_{21} + \ldots + d_{89}x_{89} + d_{99}x_{99} = 4.94$. More succinctly, solving the p-median IP amounts to searching for a specific 0–1 patterning of $\mathbf{X}$ that will yield a global minimum of the objective function.

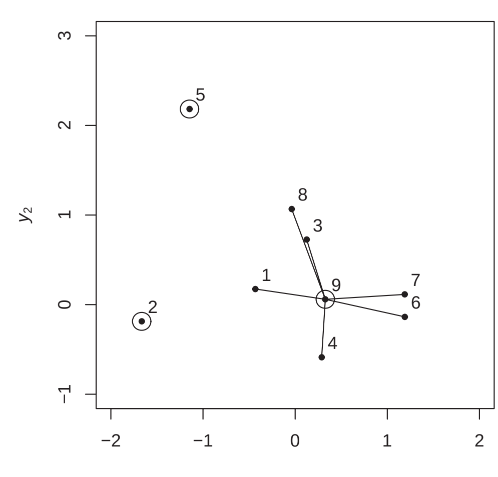

Figure 1. Scatterplot of the globally optimal solution for the nine-object set with $p = 3$ medians.

Object	1	2	3	4	5	6	7	8	9
1	0.00	1.29	0.78	1.05	2.13	1.65	1.62	0.98	0.77
2	1.29	0.00	2.01	1.99	2.43	2.86	2.87	2.06	2.01
3	0.78	2.01	0.00	1.32	1.93	1.37	1.23	0.38	0.70
4	1.05	1.99	1.32	0.00	3.12	1.01	1.14	1.69	0.65
5	2.13	2.43	1.93	3.12	0.00	3.29	3.12	1.57	2.59
6	1.65	2.86	1.37	1.01	3.29	0.00	0.25	1.72	0.89
7	1.62	2.87	1.23	1.14	3.12	0.25	0.00	1.55	0.86
8	0.98	2.06	0.38	1.69	1.57	1.72	1.55	0.00	1.07
9	0.77	2.01	0.77	0.65	2.59	0.89	0.86	1.07	0.00

Object	1	2	3	4	5	6	7	8	9	$\sum_{j}^{p} x_{ij}$
1	0	0	0	0	0	0	0	0	1	1
2	0	1	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	1	1
5	0	0	0	0	1	0	0	0	0	1
6	0	0	0	0	0	0	0	0	1	1
7	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	1	1
9	0	0	0	0	0	0	0	0	1	1

Figure 2. Dissimilarities and matrix of binary decision variables, with entries corresponding to the globally optimal solution marked by circles.

The crucial question remains regarding how to find the optimal values for the decision variables that conform to the stated constraints. For some data sets, general-purpose IP software might suffice; however, specially designed routines are apt to be more effective in most cases. Although considerable progress has been made in the development of exact methods for the *p*-median problem (Avella, Sassano & Vasil'ev, 2007; Beltran, Tadonki, & Vial, 2006; Brusco & Köhn, 2008b; Du Merle & Vial, 2002), their implementation is nontrivial, and computation times can be prohibitively large for problems of practical size and structure. For this reason, we limit our software offerings to approximate methods.

Multistart Fast Interchange Methods

Originally proposed by Teitz and Bart (1968), vertex substitution (VS) represents an efficient, effective, and widely applied approximate algorithm for solving *p*-median problems (the term *vertex* simply refers to an object as a candidate median). VS performs a systematic, but succinct iterative search among all possible median candidates through consecutively replacing selected medians by unselected objects until an exchange step will not yield any further reduction of the objective function (i.e., the total sum of median-to-satellite distances). The resulting solution is locally optimal with respect to all replacement operations, but not necessarily globally optimal. Hence, as *p*-median clustering represents a minimization problem, VS provides at least an upper bound to the globally optimal solution of the *p*-median clustering IP. In drawing from earlier work by Whitaker (1983), Hansen and Mladenović (1997, 2005) developed an efficient implementation of VS termed *fast interchange* that is similar in design to the PAM function found in R (based on Kaufman & Rousseeuw, 2005). Brusco and Köhn (2008a, 2009) evaluated a multiple random restart ("multistart") implementation of the fast interchange procedure that performed well for a large variety of test problems. Each restart uses a different set of *p* randomly selected exemplars.

We provide two MATLAB implementations, MFI.m and MFI_FC.m, of the multistart fast interchange (MFI) algorithm for *p*-median clustering. Both programs allow the user to determine the number of multiple random restarts (we recommend a minimum of 20). MFI.m is designed for standard rectangular (i.e., objects-by-variables) data matrices offering the choice between Euclidean and squared Euclidean distances to create the dissimilarity data. MFI_FC.m accommodates any dissimilarity matrix, with the additional option to specify preferences for objects reflecting their desirability for serving as an exemplar. For the interested reader, a detailed description of MFI.m and MFI_FC.m can be found in the online supplemental materials accompanying this article.

Simulated Annealing Methods

The majority of clustering tasks typically encountered in the social sciences should be well handled by MFI. Yet, as the number of chosen clusters increases beyond 20, the performance of MFI deteriorates (Brusco & Köhn, 2009; Hansen & Mladenović, 1997, 2005). Several metaheuristic algorithms have been proposed for solving large-scale *p*-median problems, including procedures based on simulated annealing (SA; Chiyoshi & Galvão, 2000), tabu search (Rolland, Schilling, & Current, 1996), and genetic algorithms (Alba & Domínguez, 2006). The recent SA algorithm by Brusco and Köhn (2009) proved to be an especially effective alternative to MFI when p is large (i.e., $p > 20$). We offer two MATLAB programs for *p*-median clustering based on the Brusco–Köhn SA-algorithm: SA.m for standard rectangular (i.e., objects-by-variables) input data matrices and SA_FC.m for any dissimilarity matrix (including the option to specify "preferences" for objects). For details, the interested reader is requested to consult the online documentation referenced above.

Lagrangian Relaxation Methods

Cornuejols, Fisher, and Nemhauser (1977; see also, Mulvey & Crowder, 1979) introduced an elegant solution for relatively large *p*-median clustering problems through Lagrangian relaxation. "Relaxation" means to simplify an optimization problem by removing those constraints that are difficult to meet. Thus, in the case of a minimization problem, dropping constraints will automatically lead to a reduction in the value of the objective function. Hence, the solution for the relaxed problem can be considered as a lower bound to the solution of the original optimization problem. In case of LR, the relaxed constraints are not simply discarded from the optimization problem but are attached with a weighting coefficient (Lagrangian multiplier) and incorporated as penalty terms into the objective function. In other words, LR represents a rather conservative form of relaxation. Still, the LR optimization problem is typically easier to solve than the original IP version and provides at least a tight lower bound to the optimal solution of the original IP of *p*-median clustering (see also Christofides & Beasley, 1982; Fisher, 1981; Hanjoul & Peeters, 1985). LR optimization problems can be solved efficiently by iterative subgradient optimization (see Agmon, 1954; Held & Karp, 1970; Motzkin & Schoenberg, 1954) provided that a tight upper bound on the solution for the original IP is available (say, through MFI or SA). Loosely speaking, subgradient optimization iteratively cycles through estimating values for the decision variables in **X**, subsequently updating the penalty coefficients for the relaxed constraints and the value of the objective function of the LR problem, followed by another estimation-update step until the value of the objective function can no longer be reduced. The combination of MFI (or SA) and LR results in at least a narrow interval bracing the globally optimal solution of the original *p*-median clustering IP. Very often, however, a verifiably globally optimal solution is identified for the original IP if the subgradient iterations converge to updated constraint weights that are all equal to zero (which is equivalent to neutralizing the LR relaxation and, therefore, identical to the original *p*-median clustering IP). An optimal solution is also guaranteed if the lower bound from the LR procedure is equal to the upper bound obtained by MFI or SA.

Two MATLAB programs for *p*-median clustering based on the LR implementation of Brusco and Köhn (2008b) are available: LR.m for standard rectangular (i.e., objects-by-variables) data matrices and LR_GEN.m for analyzing any dissimilarity matrix. For technical details, we again refer the interested reader to the supplemental materials.

Choosing the Number of Medians

As with most clustering procedures, *p*-median clustering requires prespecification of the number of clusters to be extracted—a most excruciating challenge to any user of cluster analysis. Rousseeuw (1987) suggested the widely accepted silhouette index for choosing the (most appropriate) number of groups within the context of *p*-median clustering. The silhouette index is given as

$$SI_p = \frac{1}{N}\sum_{i=1}^{N}\frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where $a(i)$ denotes the average dissimilarity between object i and all objects in its cluster and $b(i)$ represents the minimum dissimilarity of object i to any object assigned to a different cluster. The minimum value for SI_p equals zero, whereas its maximum value is unity. Determining p on the basis of Rousseeuw's (1987) silhouette index requires fitting different numbers of clusters to the data; p is then chosen as the value maximizing SI_p.

Example: Data Set on Perception of Food Items

Our application uses data collected by Ross and Murphy (1999) asking 38 participants to sort 45 foods into as many categories as they wished on the basis of perceived similarity. The data were aggregated into proportions of participants who did not place a particular pair of foods together in a common category (thus, these proportions are keyed as dissimilarities in which larger proportions represent the less pairwise similar foods). The pairwise dissimilarity scores were collected into a 45 × 45 square-symmetric proximity matrix. The ultimate substantive question involves the identification of the natural categories of food that may underlie the participants' classification judgments. The types of categorizations given by the participants can be diverse, implying an intricate, potentially multicriterial evaluation of the stimuli. For example, these might involve the differing situations in which food is consumed or possibly a more basic notion of the type of food the target presents. To illustrate, "egg" might be subsumed among foods that involve breakfast as a common consumption situation or that are dairy products (type). Similarly, "spaghetti" appears related either to those objects that are entrees, and particularly to those that are Italian (consumption situation), or, apparently when relying on a different interpretation for the word, to those foods that are cereal-based (type). We obtained globally optimal solutions for $2 \leq p \leq 14$ clusters. The final number of clusters, p, was determined on the basis of the silhouette index and the percentage reduction in the value of the objective function resulting from an increase in the number of medians from p to $p + 1$ (see Table 1). Although the absolute maximum average silhouette index occurs at $p = 11$, the values increase steeply until $p = 8$ and then begin to become quite flat. The value of .5514 for $p = 8$ also falls within Kaufman and Rousseeuw's (2005) range of "reasonable structure" (p. 88), which is .51–.70.

The solution with eight clusters, is provided in Table 2 (foods representing cluster centers, "medians," are listed in the top line in boldface). In summary, the eight clusters can be characterized as

Table 1
Globally Optimal p-Median Solutions for $2 \leq p \leq 14$, Percentage Reduction in the Value of the Objective Function Realized From Increasing the Number of Medians From p to p + 1, and Corresponding Silhouette Indices

p	Global optimum	Reduction p to $p + 1$ (in %)	Silhouette index (SI_p)
2	2871		.1630
3	2385	16.93	.2227
4	1946	18.41	.3129
5	1619	16.80	.3843
6	1358	16.12	.4330
7	1119	17.60	.4884
8	964	13.85	.5514
9	885	8.20	.5535
10	817	7.68	.5580
11	751	8.08	.5594
12	696	7.32	.5454
13	641	7.90	.5292
14	591	7.80	.5196

Table 2
Globally Optimal Food Clustering Solution, With p = 8 Medians

1	2	3	4	5	6	7	8
Apple (1)	**Broccoli (7)**	**Bagel (14)**	**Pretzels (22)**	**Pie (30)**	**Cheese (35)**	**Water (38)**	**Pork (42)**
Watermelon (2)	Lettuce (6)	Rice (12)	Crackers (20)	Doughnuts (26)	Yogurt (33)	Soda (39)	Hamburger (40)
Orange (3)	Carrots (8)	Bread (13)	Popcorn (23)	Cookies (27)	Butter (34)		Steak (41)
Banana (4)	Corn (9)	Oatmeal (15)	Nuts (24)	Cake (28)	Eggs (36)		Chicken (43)
Pineapple (5)	Onions (10)	Cereal (16)	Potato Chip (25)	Chocolate Bar (29)	Milk (37)		Lobster (44)
	Potato (11)	Muffin (17)		Pizza (31)			Salmon (45)
		Pancake (18)		Ice Cream (32)			
		Spaghetti (19)					
		Granola Bar (21)					

Note. Foods representing cluster centers (medians), are listed in the top line of the body of the table in boldface. The numbers in parentheses refer to the original numerical codes used by Ross and Murphy (1999).

fruit (Cluster 1), vegetables (Cluster 2), grain-based foods (Cluster 3), munchies (Cluster 4), pastries/tarts/desserts (Cluster 5), dairy products (Cluster 6), water (Cluster 7), and animal-based foods (Cluster 8).

Discussion and Conclusion

The choice among the myriad of available clustering methods—model-based or (combinatorial) nonmodel-based, hierarchical or nonhierarchical, and so forth (for a recent documentation, see Gan, Ma, & Wu, 2007)—represents an often difficult decision. Sometimes, theoretical considerations can help; for example, is the continuity assumption underlying most model-based clustering methods justifiable for the given data? Alternatively, were the data generated by a discrete process such that combinatorial clustering or discrete latent class models should be preferred? Many practical applications, however, lack unequivocal theoretical support for such a complex decision; instead, after exploring the results of multiple clustering methods applied to a particular data set, one might simply go with what "works best."

K-means clustering and model-based clustering provide a very reasonable choice if object categorization is to be analyzed in light of a prototype theory. Contrastingly, *p*-median classification incorporates a clustering rationale, most closely related to a perspective on object categorization elaborated by the exemplar model in cognitive theory (i.e., object categories are constructed around manifest objects). Therefore, researchers studying exemplar-driven object classification might, indeed, consider *p*-median clustering as a viable clustering technique. In addition, Kaufman and Rousseeuw (2005, Chapter 2) indicate that the *p*-median approach is more robust and less sensitive to outliers than *K*-means, which suggests that the former method is particularly suitable for clustering data potentially contaminated by such distorting observations.

As a notable benefit, the *p*-median algorithm can flexibly handle a large variety of input formats of proximity matrices, be they square-symmetric, square-asymmetric, or rectangular, containing interval-scale or categorical data (Brusco & Köhn, 2008a). Contrastingly, most model-based clustering procedures (and nonmodel based clustering procedures as well) typically only accommodate rectangular input matrices, with rows referring to objects and columns, to variables having at least interval-scale level.

Because *p*-median clustering represents a combinatorial optimization problem, it suffers from an explosive growth of feasible solutions as the number of objects increases. Hence, for realistically sized data sets, analysts typically rely on approximate procedures that do not guarantee globally optimal solutions, for example, the previously mentioned R function, PAM, based on Kaufman and Rousseeuw's (2005) work. Our MFI procedures are similar in design to PAM; however, MFI_FC.m permits differential preference weights for exemplars. The SA procedures in our program library offer a more powerful heuristic procedure for *p*-median clustering than either PAM or MFI and are well suited for challenging problems where p is large (e.g., $p > 20$). For most problems of the size and complexity observed in the social sciences literature, in our experience, MFI and SA often do obtain globally optimal partitions and, at worst, provide locally optimal solutions, with loss function values very close to the global optimum. However, the problem of multiple optima in the *p*-median context is not typically severe, as they usually involve only minor re-allocation of a few objects that do not promote an alternative interpretation of the obtained cluster structure. The LR routines we offer frequently enable a researcher to verify global optimality for a given partition. If LR fails to converge to the global optimum, then the user is provided with a margin indicating the proximity of the actual solution to global optimality.

It is our sincerest hope that researchers in psychology will appreciate the importance of exemplar-based clustering vis-à-vis the more common prototype approaches. Although exemplar-based clustering need not always be the method of choice, reverting to prototype methods solely for reasons of software (in)accessibility should present an undesirable limitation to any researcher. The *p*-median clustering programs supplementing this article afford the opportunity to analyze data with exemplar-based methods when deemed appropriate.

References

Agmon, S. (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics, 6,* 382–392.

Alba, E., & Domínguez, E. (2006). Comparative analysis of modern optimization tools for the *p*-median problem. *Statistics and Computing, 16,* 251–260.

Arabie, P., & Hubert, L. J. (1996). An overview of combinatorial data analysis. In P. Arabie, L. J. Hubert & G. de Soete (Eds.), *Clustering and classification* (pp. 5–63). River Edge, NJ: World Scientific.

Avella, P., Sassano, A., & Vasil'ev, I. (2007). Computational study of large-scale *p*-median problems. *Mathematical Programming A, 109,* 89–114.

Beltran, C., Tadonki, C., & Vial, J.-P. (2006). Solving the *p*-median problem with a semi-Lagrangian relaxation. *Computational Optimization and Applications, 35,* 239–260.

Bock, H. H. (2007). Clustering methods: A history of *K*-means algorithms. In P. Brito, P. Bertrand, G. Cucumel, & F. De Carvalho (Eds.), *Selected contributions in data analysis and classification* (pp. 161–172). Berlin, Germany: Springer.

Brusco, M. J., & Cradit, J. D. (2005). Bicriterion methods for partitioning dissimilarity matrices. *British Journal of Mathematical and Statistical Psychology, 58,* 319–332.

Brusco, M. J., & Köhn, H.-F. (2008a). Comment on "Clustering by passing messages between data points." *Science, 319,* 726c.

Brusco, M. J., & Köhn, H.-F. (2008b). Optimal partitioning of a data set based on the *p*-median model. *Psychometrika, 73,* 89–105.

Brusco, M. J., & Köhn, H.-F. (2009). Exemplar-based clustering via simulated annealing. *Psychometrika, 74,* 457–475.

Brusco, M. J., & Stahl, S. (2005). *Branch-and-bound applications in combinatorial data analysis.* New York: Springer.

Brusco, M. J., & Steinley, D. (2006). Clustering, seriation, and subset extraction of confusion data. *Psychological Methods, 11,* 271–286.

Chiyoshi, F., & Galvão, D. (2000). A statistical analysis of simulated annealing applied to the *p*-median problem. *Annals of Operations Research, 96,* 61–74.

Christofides, N., & Beasley, J. E. (1982). A tree search algorithm for the *p*-median problem. *European Journal of Operational Research, 10,* 196–204.

Cornuejols, G., Fisher, M. L., & Nemhauser, G. L. (1977). Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science, 23,* 789–810.

Du Merle, O., & Vial, J.-P. (2002). *Proximal-ACCPM, a cutting plane method for column generation and Lagrangian relaxation: Application to the p-median problem* (Technical Report 2002.23). Geneva, Switzerland: University of Geneva.

Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science, 27,* 1–18.

Forgy, E. W. (1965). Cluster analyses of multivariate data: Efficiency versus interpretability of classifications. *Biometrika, 61,* 621–626.

Fraley, C., & Raftery, A. (2006). *MCLUST Version 3 for R: Normal mixture modeling and model-based clustering* (Technical Report 504). Seattle, WA: Department of Statistics, University of Washington.

Gan, G., Ma, C., & Wu, J. (2007). *Data clustering: Theory, algorithms, and applications.* Philadelphia: Society for Industrial and Applied Mathematics.

Hanjoul, P., & Peeters, D. (1985). A comparison of two dual-based procedures for solving the *p*-median problem. *European Journal of Operational Research, 20,* 387–396.

Hansen, P., & Mladenović, N. (1997). Variable neighborhood search for the *p*-median. *Location Science, 5,* 207–226.

Hansen, P., & Mladenović, N. (2005). Variable neighborhood search. In E. K. Burke & G. Kendall (Eds.), *Search methodologies* (pp. 211–238). New York: Springer.

Hansen, P., & Mladenović, N. (2008). Complement to a comparative analysis for the *p*-median problem. *Statistics and Computing, 18,* 41–46.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A *K*-means clustering algorithm. *Applied Statistics, 28,* 100–108.

Held, M., & Karp, R. M. (1970). The traveling salesman problem and minimum spanning trees. *Operations Research, 18,* 1138–1162.

Hubert, L. J., Arabie, P., & Meulman, J. (2001). *Combinatorial data analysis: Optimization by dynamic programming.* Philadelphia, PA: Society for Industrial and Applied Mathematics.

Hubert, L. J., Arabie, P., & Meulman, J. (2006). *The structural representation of proximity matrices with MATLAB.* Philadelphia, PA: Society for Industrial and Applied Mathematics.

Hubert, L. J., Köhn, H.-F., & Steinley, D. (2009). Cluster analysis. In R. Millsap & A. Maydeu-Olivares (Eds.), *The SAGE handbook of quantitative methods in psychology* (pp. 444–512). Los Angeles, CA: Sage.

Kaufman, L., & Rousseeuw, P. J. (2005). *Finding groups in data: An introduction to cluster analysis* (9th ed.). New York: Wiley.

Klastorin, T. (1985). The *p*-median problem for cluster analysis: A comparative test using the mixture model approach. *Management Science, 31,* 84–95.

Kogan, J. (2007). *Introduction to clustering large and high-dimensional data.* New York: Cambridge University Press.

Kuehn, A. A., & Hamburger, M. J. (1963). A heuristic program for locating warehouses. *Management Science, 9,* 643–666.

Love, B. C. (2003). Concept learning. In L. Nadel (Ed.), *The encyclopedia of cognitive science* (Vol. 1, pp. 646–652), London: Nature Publishing Group.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297), Berkeley, CA: University of California Press.

Maranzana, F. E. (1964). On the location of supply points to minimize transportation costs. *Operations Research Quarterly, 12,* 138–139.

Martinez, W. L., & Martinez, A. R. (2005). *Exploratory data analysis with MATLAB.* Boca Raton, FL: Chapman & Hall/CRC.

Martinez, W. L., & Martinez, A. R. (2007). *Computational statistics handbook with MATLAB* (2nd ed.). Boca Raton, FL: Chapman & Hall/CRC.

McLachlan, G., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering.* New York: Marcel Dekker.

McLachlan, G., & Peel, D. (2000). *Finite mixture models.* New York: Wiley.

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

Motzkin, T., & Schoenberg, I. J. (1954). The relaxation method for linear inequalities. *Canadian Journal of Mathematics, 6,* 393–404.

Mulvey, J. M., & Crowder, H. P. (1979). Cluster analysis: An application of Lagrangian relaxation. *Management Science, 25,* 329–340.

Murphy, G. L. (2002). *The big book of concepts.* Cambridge, MA: MIT Press.

Muthén, L. K., & Muthén, B. O. (1998–2007). *Mplus user's guide* (5th ed.). Los Angeles, CA: Muthén & Muthén.

Rao, M. R. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association, 66,* 622–626.

Rolland, E., Schilling, D. A., & Current, J. R. (1996). An efficient tabu search procedure for the *p*-median problem. *European Journal of Operational Research, 96,* 329–342.

Ross, B. H., & Murphy, G. L. (1999). Food for thought: Cross-classification and category organization in a complex real-world domain. *Cognitive Psychology, 38,* 495–553.

Ross, B. H., Taylor, E. G., Middleton, E. L., & Nokes, T. J. (2008). Concept and category learning in humans. In H. L. Roediger III & J. H. Byrne (Eds.), *Cognitive psychology of memory* (Vol. 2, pp. 535–556), Oxford, United Kingdom: Elsevier.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics, 20,* 53–65.

Steinhaus, H. (1956). Sur la division des corps matériels en parties [On the division of a collection of objects into parts]. *Bulletin de l'Académie Polonaise des Sciences, Classe III, 4,* 801–804.

Steinley, D. (2003). *K*-means clustering: What you don't know may hurt you. *Psychological Methods, 8,* 294–304.

Steinley, D. (2006a). *K*-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology, 59,* 1–34.

Steinley, D. (2006b). Profiling local optima in *K*-means clustering: Developing a diagnostic technique. *Psychological Methods, 11,* 178–192.

Teitz, M. B., & Bart, P. (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research, 16,* 955–961.

Tversky, A. (1977). Features of similarity. *Psychological Review, 84,* 327–352.

Vermunt, J., & Magidson, J. (2005). *Latent GOLD 4.0 user's guide.* Belmont, MA: Statistical Innovations.

Vinod, H. (1969). Integer programming and the theory of grouping. *Journal of the American Statistical Association, 64,* 507–517.

Whitaker, R. (1983). A fast algorithm for greedy interchange of large-scale clustering and median location problems. *INFOR, 21,* 95–108.

Received September 20, 2008
Revision received October 5, 2009
Accepted November 4, 2009 ■