Journal of Classification 26:155-181 (2009)
DOI: 10.1007/s00357-009-9031-2

Optimization Strategies for Two-Mode Partitioning

Joost van Rosmalen

Erasmus University Rotterdam

Patrick J. F. Groenen

Erasmus University Rotterdam

Javier Trejos

Universidad de Costa Rica

William Castillo

Universidad de Costa Rica

Abstract: Two-mode partitioning is a relatively new form of clustering that clusters both rows and columns of a data matrix. In this paper, we consider deterministic two-mode partitioning methods in which a criterion similar to k-means is optimized. A variety of optimization methods have been proposed for this type of problem. However, it is still unclear which method should be used, as various methods may lead to non-global optima. This paper reviews and compares several optimization methods for two-mode partitioning. Several known methods are discussed, and a new fuzzy steps method is introduced. The fuzzy steps method is based on the fuzzy c-means algorithm of Bezdek (1981) and the fuzzy steps approach of Heiser and Groenen (1997) and Groenen and Jajuga (2001). The performances of all methods are compared in a large simulation study. In our simulations, a two-mode k-means optimization method most often gives the best results. Finally, an empirical data set is used to give a practical example of two-mode partitioning.

Keywords: Two-mode partitioning; Optimization methods; Meta-heuristics.

We would like to thank two anonymous referees whose comments have improved the quality of this paper. We are also grateful to Peter Verhoef for providing the data set used in this paper.

Authors' Addresses: Joost van Rosmalen and Patrick Groenen, Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands, e-mail: vanrosmalen@few.eur.nl; Javier Trejos and William Castillo, CIMPA, Escuela de Matematicá, Universidad de Costa Rica, San José, Costa Rica.

Published online 28 May 2009

1. Introduction

Clustering can be seen as one of the cornerstones of classification. Consider a typical two-way two-mode data set of respondents by variables. Often, clustering algorithms are applied to just one mode of the data matrix, which can be done in a hierarchical or non-hierarchical way. Among the non-hierarchical methods, k-means clustering (Hartigan 1975) is one of the most popular methods and has the advantage of a loss function being optimized. For an overview of clustering methodology for one-mode data, see Mirkin (2005).

A relatively new form of clustering is two-mode clustering. In two-mode clustering, both rows and columns of a two-mode data matrix are assigned to clusters. Each row of a two-mode data matrix is assigned to one or more row clusters, and each column to one or more column clusters. Elements of the data matrix that are in the same row cluster and in the same column cluster should be close. DeSarbo (1982) described such a two-mode clustering method, called the GENNCLUS model. A technique that is related to two-mode clustering is blockmodeling, which is often used in social network analysis (see, for example, Noma and Smith 1985; Doreian, Batagelj, and Ferligoj 2005) and can be used for both one-mode and two-mode data matrices (Doreian, Batagelj, and Ferligoj 2004). Blockmodeling attempts to partition network actors into clusters, so that a block structure can be identified in a data matrix in which the rows and columns have been reordered in a specific way; the elements within each block should typically have similar values. An extensive overview of two-mode clustering methods can be found in Van Mechelen, Bock, and De Boeck (2004).

In this article, we focus on two-mode partitioning, that is, partitioning the sets of rows and columns so that each row and each column is assigned to exactly one cluster. To do so, we use a least-squares criterion that models the elements of the data matrix belonging to the same row and column cluster by their average. Other optimization criteria and methods that do not optimize a criterion can also be used to partition the rows and columns of a data matrix. However, in the remainder of this paper, we focus on deterministic two-mode partitioning using a least-squares criterion and refer to this method simply as two-mode partitioning. Several optimization methods for finding good two-mode partitions based on this criterion are known from the literature. However, these methods are not guaranteed to find the global optimum and often get stuck in local minima. Therefore, we study the local minimum problem of two-mode partitioning for several optimization methods and determine which method tends to give the best local minimum among these methods. In addition, a new optimization method for two-mode partitioning is introduced, based on the fuzzy c-means algorithm of Bezdek

(1981). Using a simulation study, we identify the methods that perform well under most circumstances, within a reasonable computational effort.

The remainder of this article is organized as follows. In the next section, we introduce the notation and give an overview of the optimization problem, including two hill climbing algorithms. Section 3 describes the implementation of two *meta-heuristics* for two-mode partitioning. Section 4 introduces the fuzzy steps two-mode partitioning method. In Section 5, we compare the performances of the methods using a simulation study. Section 6 uses an empirical data set to compare the methods and to give a practical example of two-mode clustering. Finally, we draw conclusions and give recommendations for further research.

2. Overview of Optimization Problem

To define the two-mode partitioning problem, consider the following notation:

$\mathbf{X}_{n\times m} = (x_{ij})_{n\times m}$	Two-mode data matrix of n rows and m columns.
$\mathbf{P}_{n\times K} = (p_{ik})_{n\times K}$	Cluster membership matrix of the rows with K the number of row clusters, $p_{ik} = 1$ if row i belongs to row cluster k, and $p_{ik} = 0$ otherwise.
$\mathbf{Q}_{m\times L} = (q_{jl})_{m\times L}$	Cluster membership matrix of the columns with L the number of column clusters, $q_{jl} = 1$ if column j belongs to column cluster l, and $q_{jl} = 0$ otherwise.
$\mathbf{V}_{K\times L} = (v_{kl})_{K\times L}$	Matrix with cluster centers for row cluster k and column cluster l.
$\mathbf{E}_{n\times m} = (e_{ij})_{n\times m}$	Matrix with errors from cluster centers.

Usually, the rows of $\mathbf{X}$ correspond to objects, and the columns of $\mathbf{X}$ refer to variables. The elements of $\mathbf{X}$ can be associations, confusions, fluctuations, etc., between row and column objects. Applying two-mode partitioning only makes sense if the data matrix is *matrix-conditional*, that is, its values can be compared among each other. Therefore, if one of the modes refers to variables, these variables must be comparable, standardized, or measured on the same scale. In the remainder of this paper, we assume that the data satisfy this condition.

Two-mode partitioning assigns each element of $\mathbf{X}$ to a row cluster and a column cluster. If L equals m, each column can be placed in a cluster by itself, so that two-mode partitioning reduces to one-mode k-means clustering, and the same is true if K equals n. The matrix $\mathbf{V}$ can be interpreted as the combined cluster means. The cluster memberships are given by the

matrices $\mathbf{P}$ and $\mathbf{Q}$. Together, the three matrices $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{V}$ approximate the information in $\mathbf{X}$ by $\mathbf{PVQ}'$. To make this approximation as close to $\mathbf{X}$ as possible, we use the additive model

$$\mathbf{X} = \mathbf{PVQ}' + \mathbf{E}, \tag{1}$$

where $\mathbf{E}$ is the error of the model. Equation (1) can be seen as a special case of the additive box model proposed by Mirkin, Arabie, and Hubert (1995); the additive box model does not partition the rows and columns, but allows for overlapping clusters.

Two-mode partitioning searches for the optimal partition $\mathbf{P}$, $\mathbf{Q}$ and cluster centers $\mathbf{V}$ that minimize the sums of squares of $\mathbf{E}$. This objective amounts to minimizing the squared Euclidean distance of the data points to their respective clusters centers in $\mathbf{V}$. Therefore, the criterion to be minimized can be expressed as

$$f(\mathbf{P}, \mathbf{Q}, \mathbf{V}) = \|\mathbf{X} - \mathbf{PVQ}'\|^2 = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}q_{jl}(x_{ij} - v_{kl})^2. \tag{2}$$

Using the Euclidean metric is not mandatory; other metrics have been used as well, especially in one-mode clustering (see, for example, Bock 1974). However, in this study, we restrict ourselves to the Euclidean metric.

The optimal cluster membership matrices must satisfy the following constraints.

1. The cluster memberships of each row and column object must sum to one, so that $\sum_{k=1}^{K} p_{ik} = 1$ and $\sum_{l=1}^{L} q_{jl} = 1$.
2. All cluster membership values must be either zero or one, so that $p_{ik} \in \{0, 1\}$ and $q_{jl} \in \{0, 1\}$.
3. None of the row or column clusters is empty, that is $\sum_{i=1}^{n} p_{ik} > 0$ and $\sum_{j=1}^{m} q_{jl} > 0$.

The first two constraints together require that each row of $\mathbf{P}$ and $\mathbf{Q}$ contains exactly one element with the value 1. Hence, each row and column object is assigned to exactly one cluster. These two constraints are necessary and sufficient for the second equality in (2) to hold. A partition that is optimal according to (2) typically does not have empty clusters, and, in principle, the third constraint is not required during the estimation. However, some algorithms may lead to a partition with empty clusters. Therefore, we adapt these algorithms to correct for potential empty clusters or prevent them from happening.

No known polynomial time algorithm is guaranteed to find the global minimum of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ in every instance. Even for relatively small n and

m (say, $n = 20$ and $m = 20$), the number of possible partitions can become extremely large, and a complete enumeration of the possible solutions is almost always computationally infeasible. However, if two of the three matrices $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{V}$ are known, the optimal value of the third matrix can be computed easily. If both $\mathbf{P}$ and $\mathbf{Q}$ are known, the optimal cluster centers $\mathbf{V}$ can be computed as

$$v_{kl} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}q_{jl}x_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}q_{jl}}, \tag{3}$$

which is the average of the elements of $\mathbf{X}$ belonging to row cluster k and column cluster l. If $\mathbf{V}$ and either $\mathbf{P}$ or $\mathbf{Q}$ are known, the problem of minimizing $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ becomes a linear program that has a closed-form solution. When $\mathbf{V}$ and $\mathbf{Q}$ are known, the optimal value of $\mathbf{P}$ can be computed as follows. Let $c_{ik} = \sum_{j=1}^{m}\sum_{l=1}^{L} q_{jl}(x_{ij} - v_{kl})^2$. Then,

$$p_{ik} = \begin{cases} 1 & \text{if } c_{ik} = \min_{1\le r\le K} c_{ir}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

When $\mathbf{P}$ and $\mathbf{V}$ are known, the optimal matrix $\mathbf{Q}$ can be computed in a similar fashion.

Several methods for finding an optimal two-mode partition, based on the minimization of (2), have been proposed in the literature. Two of these methods are discussed in the remainder of this section. Both methods are hill climbing algorithms and are guaranteed to find a local optimum (which may or may not be the global optimum). Three additional optimization methods for two-mode partitioning are discussed in the next two sections.

2.1 Alternating Exchanges Algorithm

The alternating exchanges algorithm was proposed by Gaul and Schader (1996). This algorithm tries to improve an initial partition by making a transfer of either a row or a column object and immediately recalculating $\mathbf{V}$. Our implementation of the alternating exchanges algorithm performs the following steps:

1. Choose initial $\mathbf{P}$ and $\mathbf{Q}$, and calculate $\mathbf{V}$ according to (3).
2. Repeat the following until there is no improvement of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ in either step.
   (a) For each i and k, transfer row object i to row class k and recalculate $\mathbf{V}$ according to (3). Accept the transfer if it has improved $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$, otherwise return to the old $\mathbf{P}$ and $\mathbf{V}$.
   (b) For each j and l, transfer column object j to column class l and re-calculate $\mathbf{V}$ according to (3). Accept the transfer if it has improved $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$, otherwise return to the old $\mathbf{Q}$ and $\mathbf{V}$.

The alternating exchanges algorithm always converges to a local minimum, as the value of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ decreases in every iteration, the algorithm is defined on a finite set, and $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ is bounded from below by 0.

2.2 Two-Mode *k*-Means Algorithm

The *k*-means algorithm (Hartigan 1975) is one of the simplest and fastest ways to obtain a good partition, which accounts for its popularity in one-mode clustering. This algorithm can easily be extended to handle two-mode partitioning (see Baier, Gaul, and Schader 1997; Vichi 2001). The so-called two-mode *k*-means algorithm aims to improve an initial partition using (3) and (4) and comprises the following steps.

1. Choose initial $\mathbf{P}$ and $\mathbf{Q}$.
2. Repeat the following, until there is no improvement of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ in any step.
   (a) Update $\mathbf{V}$ according to (3).
   (b) Let $c_{ik} = \sum_{j=1}^{m} \sum_{l=1}^{L} q_{jl}(x_{ij} - v_{kl})^2$. Then update $\mathbf{P}$ according to
   $$p_{ik} = \begin{cases} 1 & \text{if } c_{ik} = \min_{1 \le r \le K} c_{ir}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$
   (c) Update $\mathbf{V}$ according to (3).
   (d) Let $d_{jl} = \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ij}(x_{ij} - v_{kl})^2$. Then update $\mathbf{Q}$ according to
   $$q_{jl} = \begin{cases} 1 & \text{if } d_{jl} = \min_{1 \le r \le L} d_{jr}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The two-mode *k*-means algorithm always converges to a local minimum, as the value of the criterion $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ cannot increase in any step. However, one or more clusters may become empty after Step 2b or 2d. This situation is immediately corrected by transferring the row or column object with the highest value of $\sum_{k=1}^{K} p_{ik} c_{ik}$ or $\sum_{l=1}^{L} q_{jl} d_{jl}$ to the empty cluster.

3. Meta-Heuristics

The algorithms discussed in the previous section contain no provisions to avoid finding non-global optima. As no known polynomial time algorithm is capable of finding the global optimum in every instance, we also use optimization methods that are based on *meta-heuristics*. These meta-heuristics aim to increase the likelihood of finding the global optimum, though this cannot be guaranteed. Here, we discuss two optimization methods for two-mode partitioning that are applications of the meta-heuristics simulated annealing and tabu search. For the other three optimization methods in this paper (the two algorithms discussed in the previous section and

the fuzzy steps method that is proposed in the next section), the likelihood of finding a non-global optimum is reduced by performing multiple random starts. That is, these algorithms are performed several times with starting values that are chosen randomly every time an algorithm is run; the best result of all runs is retained as the final solution.

Other optimization methods have also been used for two-mode partitioning. Hansohm (2001) used a genetic algorithm, but found that its performance in two-mode partitioning is not as good as it is in one-mode *k*-means clustering. Gaul and Schader (1996) implemented a penalty algorithm, but found that it does not compare favorably with the alternating exchanges algorithm.

3.1 Simulated Annealing

Simulated annealing (see, for example, Van Laarhoven and Aarts 1987) is a meta-heuristic that simulates the slow cooling of a physical system. Trejos and Castillo (2000) first used simulated annealing for two-mode partitioning. Their implementation of simulated annealing performs a local search that is based on the central idea of the alternating exchanges algorithm. To avoid getting stuck in local minima, transitions that increase $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ are also accepted with a positive probability.

Here, we use the implementation of Trejos and Castillo (2000), except for the values of certain parameters. These parameters are the cooling rate $\gamma < 1$, the number of iterations R in which the temperature remains constant, the initial value of the temperature T, and the maximum number of iterations without accepted transitions, which is denoted by t_{max}. This implementation comprises the following steps.

1. Choose initial $\mathbf{P}$ and $\mathbf{Q}$ and calculate $\mathbf{V}$ according to (3).
2. Choose the parameters R, γ, and an initial value of T.
3. Repeat the following until there is no change in $\mathbf{P}$ and $\mathbf{Q}$ for the last t_{max} values of T.
   (a) Do the following R times:
      i. Choose one of the two modes with equal probability.
      ii. Choose one of the objects of this mode with uniform probability and transfer it to another randomly chosen cluster.
      iii. Update $\mathbf{V}$ according to (3) and calculate Δf as the change in $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ achieved by the transfer and the subsequent updating of $\mathbf{V}$.
      iv. Always accept the transfer if $\Delta f < 0$, otherwise accept it with probability $\exp(-\Delta f/T)$.
   (b) Set $T = \gamma T$.

The partition with the lowest value of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ found during estimation is retained as the final solution.

3.2 Tabu Search

The tabu search meta-heuristic (see, for example, Glover 1986) also performs a local search, but tries to avoid local optima by maintaining a tabu list. The tabu list is a list of solutions that are temporarily not accepted. We use the following implementation of tabu search for two-mode partitioning, which is based on the alternating exchanges algorithm. See Castillo and Trejos (2002) for a more detailed description of this implementation.

Define $Z(\mathbf{P}, \mathbf{Q}) = \min_{\mathbf{V}} f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$. The algorithm performs the following steps.

1. Start with an initial partition $(\mathbf{P}, \mathbf{Q})$ and an empty tabu list. Set $(\mathbf{P}, \mathbf{Q})_{opt} = (\mathbf{P}, \mathbf{Q})$.
2. Choose the number of iterations S and a maximum length of the tabu list.
3. Perform the following steps S times.
   (a) Generate a neighborhood of partitions N, consisting of the partitions that can be constructed by transferring one row or column object from (**P**,**Q**) to another cluster.
   (b) Choose the partition $(\mathbf{P}, \mathbf{Q})_{cand}$ as the partition in N with the lowest value of $Z(\mathbf{P}, \mathbf{Q})$ that is not on the tabu list.
   (c) Set $(\mathbf{P}, \mathbf{Q}) = (\mathbf{P}, \mathbf{Q})_{cand}$. If $Z((\mathbf{P}, \mathbf{Q})_{cand}) < Z((\mathbf{P}, \mathbf{Q})_{opt})$, then $(\mathbf{P}, \mathbf{Q})_{opt} = (\mathbf{P}, \mathbf{Q})_{cand}$.
   (d) Add (**P**,**Q**) to the tabu list. Remove the oldest item from the tabu list, if the list exceeds its maximum length.

The final solution of the algorithm is given by $(\mathbf{P}, \mathbf{Q})_{opt}$.

4. Fuzzy Two-Mode Clustering

Fuzzy methods relax the requirement that an object belongs to a single cluster, so that the cluster membership can be distributed over the clusters. For one-mode clustering, the best known method is fuzzy c-means (Bezdek 1981); for adaptations of this method see Tsao, Bezdek, and Pal (1994) and Groenen and Jajuga (2001). These methods try to make the optimization task easier by allowing for cluster membership values between 0 and 1. In this section, we extend one-mode fuzzy optimization methods to the two-mode case. First, we introduce a fuzzy two-mode clustering criterion. We then discuss an algorithm for finding an optimal fuzzy partition, based on

Bezdek (1981). Finally, we describe the fuzzy steps method, which reduces the fuzziness of the solution in steps until a crisp partition (that is, a partition in which all cluster membership values are either 0 or 1) is found.

Simply relaxing the constraint that the cluster membership values in (2) must be 0 or 1 by allowing for values between 0 and 1 does not guarantee an optimal partition that is fuzzy. A crisp partition will still minimize $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ in that case, though a fuzzy partition might be equally good. Fuzzy optimal partitions can be obtained if the criterion $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ is altered by raising the cluster membership values to a power s, with $s \geq 1$. Thus, the fuzzy two-mode clustering criterion is defined as

$$f_s(\mathbf{P}, \mathbf{Q}, \mathbf{V}) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ik}^s q_{jl}^s (x_{ij} - v_{kl})^2, \tag{7}$$

subject to the constraints

$$\sum_{k=1}^{K} p_{ik} = 1, \sum_{l=1}^{L} q_{jl} = 1, p_{ik} \geq 0, q_{jl} \geq 0, \sum_{i=1}^{n} p_{ik} > 0, \text{and} \sum_{j=1}^{m} q_{jl} > 0. \tag{8}$$

This criterion yields fuzzy optimal partitions and the fuzziness parameter s determines how fuzzy the optimal partition is. For $s = 1$, the fuzzy criterion coincides with the crisp criterion.

4.1 The Two-Mode Fuzzy *c*-Means Algorithm

The algorithm for the optimization of $f_s(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ is based on iteratively updating each set of parameters while keeping the other two sets fixed. Given optimal $\mathbf{Q}$ and $\mathbf{V}$, the optimal $\mathbf{P}$ can be found using the Lagrange method for each row i of $\mathbf{P}$. The Lagrangian is given by

$$L_i(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \lambda) = \sum_{k=1}^{K} p_{ik}^s \sum_{l=1}^{L} \sum_{j=1}^{m} q_{jl}^s (x_{ij} - v_{kl})^2 - \lambda \left(\sum_{k=1}^{K} p_{ik} - 1 \right). \tag{9}$$

Defining $c_{ik} = \sum_{l=1}^{L} \sum_{j=1}^{m} q_{jl}^s (x_{ij} - v_{kl})^2$ and taking partial derivatives of L_i gives

$$\frac{\partial L_i}{\partial p_{ik}} = s p_{ik}^{s-1} c_{ik} - \lambda \quad \text{and} \quad \frac{\partial L_i}{\partial \lambda} = \sum_{k=1}^{K} p_{ik} - 1. \tag{10}$$

Setting these derivatives to zero and solving for p_{ik} yields

$$p_{ik} = \frac{c_{ik}^{1/(1-s)}}{\sum_{k=1}^{K} c_{ik}^{1/(1-s)}}. \tag{11}$$

However, (11) does not apply if any c_{ik} in row i is zero. In that case, any partition with $p_{ik} = 0$ whenever $c_{ik} > 0$ and $\sum_{k=1}^{K} p_{ik} = 1$ is optimal. Finding the optimal $\mathbf{Q}$ given $\mathbf{P}$ and $\mathbf{V}$ can be done in a similar fashion. When s is large enough, the optimal values of the cluster memberships become $p_{ik} \approx 1/K$ and $q_{jl} \approx 1/L$, which can easily be derived from (11). In practice, the cluster membership values approach these values quite rapidly for reasonably large s. For $s = 3$, the cluster membership values often differ only slightly and for $s > 10$, they are usually equal to each other within the numerical accuracy of current computers. As s approaches 1 from above, $1/(1 - s)$ approaches minus infinity, and the fuzzy optimization formula (11) becomes its crisp counterpart (4). In that case, the optimal partition based on (7) also approaches the optimal crisp partition. Therefore, higher values of s correspond to fuzzier optimal partitions, and s close to 1 to crisp partitions. The optimal $\mathbf{V}$ can be obtained by setting the partial derivatives of (7) with respect to v_{kl} to zero and solving for v_{kl}, which yields

$$v_{kl} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}^{s} q_{jl}^{s} x_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}^{s} q_{jl}^{s}}. \tag{12}$$

For a given value of s, the *two-mode fuzzy* c*-means algorithm* comprises the following steps.

1. Choose initial $\mathbf{P}$ and $\mathbf{Q}$, which can be either crisp or fuzzy, and calculate $\mathbf{V}$ according to (12).
2. Repeat the following, until the decrease in $f_s(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ is small.
   (a) Calculate $c_{ik} = \sum_{l=1}^{L}\sum_{j=1}^{m} q_{jl}^{s}(x_{ij} - v_{kl})^2$ and update $\mathbf{P}$ as $p_{ik} = c_{ik}^{1/(1-s)} / \sum_{k=1}^{K} c_{ik}^{1/(1-s)}$.
   (b) Calculate $d_{jl} = \sum_{k=1}^{K}\sum_{i=1}^{n} p_{ik}^{s}(x_{ij} - v_{kl})^2$ and update $\mathbf{Q}$ as $q_{ik} = d_{jl}^{1/(1-s)} / \sum_{k=1}^{K} d_{jl}^{1/(1-s)}$.
   (c) Update $\mathbf{V}$ according to (12).

This algorithm lowers the value of $f_s(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ in each iteration, until convergence has been achieved. Therefore, the algorithm always converges to a saddle point or a local minimum, which may or may not be a global minimum.

4.2 Fuzzy Steps

The two-mode fuzzy *c*-means algorithm generally converges to a fuzzy partition. To ensure that our fuzzy optimization method converges to a crisp partition, we use the idea of fuzzy steps, which was proposed by Heiser and Groenen (1997). Our fuzzy steps method for two-mode partitioning starts

with an initial value of s that is greater than 1. This method uses the two-mode fuzzy c-means algorithm to minimize $f_s(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ for a given value of s and gradually lowers s to avoid local minima and obtain a good crisp partition. Our fuzzy steps method performs the following steps.

1. Choose an initial value of s, a fuzzy step size $\gamma < 1$, and a threshold value s_{min}.
2. Choose initial $\mathbf{P}_0$ and $\mathbf{Q}_0$ and calculate $\mathbf{V}_0$ according to (12). The initial $\mathbf{P}_0$ and $\mathbf{Q}_0$ can be either crisp or fuzzy.
3. Repeat the following while $s > s_{min}$.
   (a) Perform the two-mode fuzzy c-means algorithm starting with $\mathbf{P}_0$, $\mathbf{Q}_0$, and $\mathbf{V}_0$. The results are in $\mathbf{P}_1$, $\mathbf{Q}_1$, and $\mathbf{V}_1$.
   (b) Set $s = 1 + \gamma(s-1)$ and set $\mathbf{P}_0 = \mathbf{P}_1$, $\mathbf{Q}_0 = \mathbf{Q}_1$, and $\mathbf{V}_0 = \mathbf{V}_1$.
4. Apply the two-mode k-means algorithm starting from $\mathbf{P}_0$, $\mathbf{Q}_0$, and $\mathbf{V}_0$.

The formula in Step 3b for decreasing s gives an exponential decay of $(s-1)$. The value of s_{min} should generally be set to a value slightly higher than 1, for example, 1.001. The two-mode k-means algorithm is performed at the end of the fuzzy steps method to ensure that a crisp solution is found. Although the two-mode k-means algorithm was defined for crisp partitions in Section 2.2, it can also be used in combination with fuzzy initial partitions.

The fuzzy steps optimization method may sometimes get stuck in a saddle point, if two or more row or column clusters become equal. In that case, these clusters and their corresponding cluster membership values will remain equal for any value of s. Preliminary tests with this method suggest that it often reaches a saddle point, if the starting value of s is too high. Therefore, one should not set the starting value of s too high, for example, $s \leq 1.2$.

5. Simulation Study

To compare the performances of the optimization methods described in the previous sections, we conduct a simulation study. With this simulation study, we also aim to determine which methods perform well under most circumstances and how well the optimization methods can retrieve a clustering structure. First, we describe the setup of the simulation study and how the results are represented. We then give the results of the simulation study and interpret them.

5.1 Setup of the Simulation Study

We generate the data matrix $\mathbf{X}$ in each problem instance by simulating $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{V}$, and $\mathbf{E}$, and then using (1) to construct $\mathbf{X}$. Generating simu-

lated data this way comes natural and has the advantage that some clustering structure exists in the data. Also, the $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{V}$ used in generating $\mathbf{X}$ can give a useful upper bound on the optimal value of $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$.

A large number of factors can be varied in a simulation study for two-mode clustering, such as the values of n, m, K, and L, the size and the distribution of the errors, the numbers of elements in the clusters, and the locations of the cluster centers. As a full-factorial design with several levels for each of these factors would require a prohibitively large number of simulations, we limit the number of factors and the number of levels for each factor. The simulation study is set up using a four-factor design and is loosely based on the approach of Milligan (1980). The first factor is the size of the data matrix $\mathbf{X}$. The numbers of rows and columns are given by $n = m = 60$, $n = 150$ and $m = 30$, and $n = m = 120$ for the three levels of this factor. The second factor is the numbers of clusters; the three levels of this factor are $K = L = 3$, $K = L = 5$, and $K = L = 7$. The third factor is the size of the error perturbations. All elements of $\mathbf{E}$ are independently normally distributed with mean 0 and standard deviation equal to 0.5, 1, or 2 for the three levels of this factor. The final factor is the distribution of the objects over the clusters. For the first level of this factor, all objects are divided over the clusters with equal probability. For the second and third levels, one cluster contains exactly 10% and 60% of the objects, respectively. The remaining objects are then divided over the remaining clusters with uniform probability. Constructing one cluster with 10% of the objects represents a small deviation from a uniform distribution, whereas a cluster with 60% constitutes a large deviation. The size of this deviation also depends on the numbers of clusters.

Empty clusters are not allowed in the generated data sets. If a simulated data set contains an empty cluster, this data set is discarded and another data set is simulated. The locations of the cluster centers v_{kl} are chosen by randomly assigning the numbers $\Phi(\frac{i}{K \times L + 1}), i = 1, \ldots, K \times L$ to the elements of $\mathbf{V}$, where $\Phi(.)$ is the inverse standard normal cumulative distribution function. As a result, the elements of $\mathbf{V}$ appear standard normally distributed, and a fixed minimum distance between the cluster centers is ensured.

We show the effects of some of these choices in Figures 1 through 3. These figures give a visual representation of a simulated data matrix for various levels of the factors. In these figures, the rows and columns are ordered according to their cluster. The elements of the data matrix are represented by different shades of gray. The bars at the right-hand side show what values the shades of gray represent. Figure 1 represents a data set for which the levels of the factors ensure that the original clusters can easily be recognized. In Figure 2, the original clusters are more difficult to recognize,

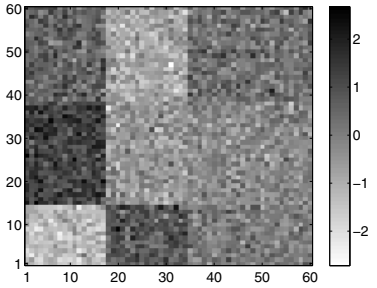

Figure 1: Graphical representation of simulated data set with sizes $n = m = 60$, $K = L = 3$, error standard deviation 0.5, and a uniform distribution of the objects over the clusters.

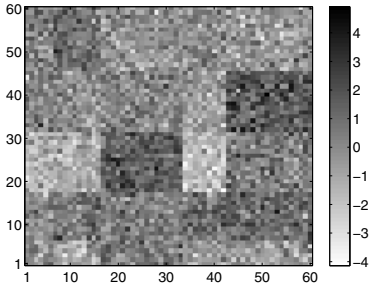

Figure 2: Graphical representation of simulated data set with sizes $n = m = 60$, $K = L = 5$, error standard deviation 1, and 10% of the objects in one cluster.

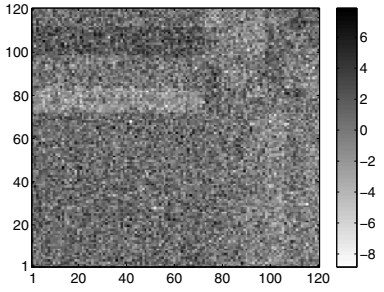

Figure 3: Graphical representation of simulated data set with sizes $n = m = 120$, $K = L = 7$, error standard deviation 2, and 60% of the objects in one cluster.

and in Figure 3, this is almost impossible. The optimal partition should generally be easier to find if the number of clusters is low, the error standard deviation is small, and the objects are evenly divided over the clusters.

The four factors in the simulation design give a total of $3^4 = 81$ possible combinations. To avoid drawing spurious conclusions based on a single data set, we simulate 50 data sets for each combination, and all five methods are performed for each data set.

All optimization methods require an initial choice for $\mathbf{P}$ and $\mathbf{Q}$. For the alternating exchanges and two-mode k-means methods, $\mathbf{P}$ and $\mathbf{Q}$ are chosen randomly by assigning each row and column to a cluster with uniform probability. However, some methods may perform better if the initial partitions are relatively good. Therefore, the initial partitions of the simulated annealing, tabu search, and fuzzy steps methods are chosen by applying the two-mode k-means algorithm to a random partition.

The optimization methods also require choosing additional parameters. For the optimization methods that use multiple random starts (that is, the alternating exchanges, two-mode k-means, and fuzzy steps methods), the numbers of random starts have been set so that the total computation time of these methods is approximately equal. This is also true for the numbers of iterations R and S that are respectively used in the simulated annealing and tabu search methods. We also wish to ensure a good balance between the performance and the computation time of the simulated annealing and tabu search methods for varying problem sizes. Therefore, the parameters R and S are chosen as a function of the size of the neighborhood used in these two methods, which equals $n(K-1)+m(L-1)$; as a result, the computation time should increase modestly with the size of the problem. The initial temperature in the simulated annealing method is chosen in such a way that the initial acceptance rate for the transfer of a row or a column to another cluster is at least 85%. As a result, the transfers of rows and columns should occur almost randomly in the initial iterations of this method. The cooling rate γ in this method should be chosen slightly lower than 1. We stop the simulated annealing method if there are no changes for $t_{max} = 10$ values of the temperature parameter, as further improvements are unlikely to occur in that case. Preliminary experimentation with the tabu search method shows that increasing the tabu list length typically has a positive effect on the performance of this method. We set the length of the tabu list to one third of the number of iterations S, as increasing the tabu list length any further would increase computation time without meaningfully improving the performance. In the fuzzy steps method, the fuzzy step size γ is set to a value somewhat lower than 1. The threshold value of s (that is, the parameter s_{min}) is set to 1.001, as experimentation shows that the solution is almost always close to a crisp partition at that value of s. The initial value of s was optimized on a separate set of data sets, which were simulated in the same way as the data sets in the simulation study. Initial values of $s = 1.025, 1.05, \ldots, 1.2$ were tested, and an optimal recovery was achieved with $s = 1.05$.

The parameters of the optimization methods are thus chosen as follows:

- Alternating exchanges: The alternating exchanges algorithm is run 30 times for every simulated data set.
- Two-mode *k*-means: The two-mode *k*-means algorithm is run 500 times for every simulated data set.
- Simulated annealing: Initial temperature $T = 1$, $\gamma = 0.95$, $t_{max} = 10$, and $R = 1/4(n(K-1) + m(L-1))$.
- Tabu search: We perform $S = 3(n(K-1) + m(L-1))^{1/2}$ iterations, and the length of the tabu list is $1/3S$.
- Fuzzy steps: The initial value of s is 1.05, the fuzzy step size γ is 0.85, and the threshold value s_{min} is 1.001. The fuzzy steps method is run 20 times for every simulated data set.

The values of the parameters that are a function of the neighborhood size $n(K-1) + m(L-1)$ are rounded to integer values.

We use three criteria to evaluate the results of the simulation study. The first criterion is the variance accounted for (VAF) criterion, which is comparable to the R^2-measure used in regression analysis. The VAF criterion is defined as

$$\text{VAF} = 1 - \frac{\sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ik}q_{jl}(x_{ij} - v_{kl})^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \bar{x})^2}, \qquad (13)$$

where $\bar{x} = 1/(nm)\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}$. The optimal value of VAF ranges from 0 to 1, and maximizing VAF corresponds to minimizing $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$.

Second, we report the average adjusted Rand index (ARI), which was introduced by Hubert and Arabie (1985). This metric shows how well the partitions found by the optimization methods approximate the original partition. The ARI is invariant with respect to the ordering of the clusters, which is arbitrary in two-mode partitioning. The ARI is based on the original Rand index, which is defined as the fraction of the pairs of elements on which the two partitions agree. For two random partitions, the expected value of the Rand index depends on the parameters of the clustering problem. To ensure a constant expectation, the ARI is a linear function of the Rand index, so that its expectation for randomly chosen partitions is 0, and its maximum value is 1. The ARI is defined as

$$\text{ARI} = \frac{\sum_{i=1}^{R}\sum_{j=1}^{R}\binom{a_{ij}}{2} - \sum_{i=1}^{R}\binom{a_{i.}}{2}\sum_{j=1}^{R}\binom{a_{.j}}{2}/\binom{a}{2}}{[\sum_{i=1}^{R}\binom{a_{i.}}{2} + \sum_{j=1}^{R}\binom{a_{.j}}{2}]/2 - \sum_{i=1}^{R}\binom{a_{i.}}{2}\sum_{j=1}^{R}\binom{a_{.j}}{2}/\binom{a}{2}}, \qquad (14)$$

where a_{ij} is the number of elements of $\mathbf{X}$ that simultaneously belong to cluster i of the original partition and cluster j of the retrieved partition, R

is the total number of clusters, $a_{i.} = \sum_{j=1}^{R} a_{ij}$, $a_{.j} = \sum_{i=1}^{R} a_{ij}$, and $a = \sum_{i=1}^{R} \sum_{j=1}^{R} a_{ij} = nm$. Here we consider a pair of elements to be in the same cluster only if they belong to same row cluster and to the same column cluster, so that $R = K \times L$.

The final criterion is the average CPU time, which is of practical interest. In this study, the CPU time is also used as a control variable, as the parameters of the optimization methods have been chosen in such a way, that the average computation time is approximately equal for all methods. To provide a fair comparison, we made sure that the optimization methods were programmed in an efficient way. Efficient updating formulas for computing the change in $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$ when transferring an object from one cluster to another are given in Castillo and Trejos (2002). These updating formulas were used for the alternating exchanges, simulated annealing, and tabu search methods. For the two-mode *k*-means algorithm, an efficient implementation is given by Rocci and Vichi (2008). All computer programs are written in the matrix programming language MATLAB 7.2 and are executed on a Pentium IV 2.8 GHz computer. These programs are available from the authors upon request.

Dolan and Moré (2002) discuss a convenient tool, called *performance profiles*, for graphically representing the distribution of the results of a simulation study. Performance profiles are especially useful for determining which optimization methods perform reasonably well in almost every instance. They can be constructed as follows. First, one has to identify a performance measure. We use the VAF criterion for this and define $\text{VAF}^{(p,s)}$ as the VAF achieved by method s in problem instance p. We include the partition used to generate the data as one of the methods in the performance profiles. Then, the *performance ratio* $\rho^{(p,s)}$ is defined as

$$\rho^{(p,s)} = \frac{\max_s \text{VAF}^{(p,s)}}{\text{VAF}^{(p,s)}}. \tag{15}$$

Finally, the cumulative distribution function of the performance ratio can be computed as

$$\Psi^{(s)}(\tau) = \frac{1}{50} \sum_{p=1}^{50} I(\rho^{(p,s)} \leq \tau), \tag{16}$$

where $I()$ denotes the indicator function. By drawing $\Psi^{(s)}(\tau)$ in one figure for all optimization methods, their performances can be compared quite easily.

5.2 Simulation Results

We now present the results of the simulation study. Tables 1 through 3 show the average VAF values of all optimization methods and the original partitions for various combinations of the factors. Tables 1, 2, and 3 respectively show the effects of the size of the data set, the numbers of clusters, and the distribution of the objects over the clusters. As the error standard deviation strongly affects the magnitude of the VAF values, we do not report average VAF values over data sets with different error standard deviations; the effects of the error standard deviation are shown within each of these three tables. The results of the best performing optimization method are shown in italics, for each column in these tables.

Based on the VAF values, the two-mode *k*-means method appears to perform relatively well. This method always has the best average performance if the error standard deviation is 0.5 or 1. In addition, the partition found by this method is almost always as least as good as the original partition. For data sets with an error standard deviation of 2, the best performance is most often achieved by the fuzzy steps method, though the alternating exchanges and the two-mode *k*-means methods also perform well. The simulated annealing method and especially the tabu search method do not perform as well as the other optimization methods in the simulation study.

Table 4 gives the average values of the adjusted Rand Index, for each level of each factor, averaged over the levels of the three other factors; the best results are shown in italics. The two-mode *k*-means method again has the best average performance. This method almost always exactly retrieves the original partition for data sets with a small error standard deviation and a uniform distribution of the rows and columns over the clusters. The best performing optimization method usually has an average ARI above 90% or even 99% if the characteristics of the simulated data sets are favorable. However, this value rapidly decreases, when the problem instances become harder. The original partitions are especially hard to retrieve if the error standard deviation is 2 or if 60% of the objects are located in one cluster. The ARI never becomes negative or close to 0 for any optimization method, indicating that at least some structure can always be found in the data set. Another important conclusion is that the differences in the average adjusted Rand indices between the methods can be large. The difference in ARI can sometimes be as large as 20%, when the corresponding difference in VAF is just a few percentage points. Therefore, the choice of the optimization method can be quite important, if one wants to find the 'true' clustering.

Finally, we give the average CPU times in seconds used by the methods in Table 5, for each size of the data set and each number of clusters. Note that the average CPU times strongly depend on the type of computer

Table 1: Average VAF for data sets with different error variances and numbers of rows and columns.

Size of data set	$n = 60, m = 60$			$n = 150, m = 30$			$n = 120, m = 120$		
Error st. dev.	.5	1	2	.5	1	2	.5	1	2
Alternating exchanges	.7260	.4128	.1685	.7288	.4140	.1750	.7251	.4058	.1532
Two-mode *k*-means	*.7283*	*.4134*	.1667	*.7322*	*.4146*	.1733	*.7293*	*.4082*	*.1535*
Simulated annealing	.7092	.4068	.1668	.7166	.4078	.1739	.7108	.3987	.1518
Tabu search	.6823	.4003	.1642	.6904	.4015	.1660	.6703	.3853	.1496
Fuzzy steps	.7248	.4129	*.1688*	.7299	.4143	*.1755*	.7199	.4064	.1531
Original partition	.7292	.4126	.1540	.7326	.4122	.1531	.7308	.4088	.1515

Table 2: Average VAF for data sets with different error variances and numbers of clusters.

Numbers of clusters	$K = 3, L = 3$			$K = 5, L = 5$			$K = 7, L = 7$		
Error st. dev.	.5	1	2	.5	1	2	.5	1	2
Alternating exchanges	.6843	.3558	.1294	.7399	.4260	*.1725*	.7557	.4508	.1948
Two-mode *k*-means	*.6843*	*.3558*	*.1296*	*.7440*	*.4276*	.1724	*.7614*	*.4529*	.1915
Simulated annealing	.6804	.3525	.1284	.7221	.4186	.1706	.7340	.4422	.1935
Tabu search	.6584	.3476	.1259	.6868	.4066	.1663	.6980	.4329	.1876
Fuzzy steps	.6838	.3558	.1288	.7394	.4264	.1725	.7515	.4515	*.1960*
Original partition	.6843	.3549	.1214	.7442	.4265	.1601	.7640	.4522	.1771

Table 3: Average VAF for data sets with different error variances and distributions of the objects over the clusters.

Object distribution	uniform distribution			10% in one cluster			60% in one cluster		
Error st. dev.	.5	1	2	.5	1	2	.5	1	2
Alternating exchanges	.7460	.4293	.1695	.7330	.4171	.1666	.7009	.3862	.1606
Two-mode *k*-means	*.7467*	*.4294*	.1683	*.7336*	*.4174*	.1655	*.7094*	*.3894*	.1596
Simulated annealing	.7352	.4255	.1684	.7199	.4105	.1650	.6814	.3774	.1593
Tabu search	.6858	.4127	.1641	.6818	.3993	.1606	.6755	.3751	.1552
Fuzzy steps	.7419	.4291	*.1698*	.7299	.4170	*.1668*	.7029	.3875	*.1608*
Original partition	.7467	.4288	.1602	.7337	.4167	.1566	.7121	.3881	.1418

used and how the methods have been implemented; they should only serve as a general indication of the amount of computation that optimization methods require. The effects of the distribution of the objects over the clusters and the error standard deviation on the computation time are not reported here. The computation time does not increase with m, n, K, and L in the same manner for all optimization methods. The computation time of the two-mode *k*-means method increases relatively slowly with the size of the data set and the numbers of clusters; the increase of the computation time is more rapid for the tabu search and fuzzy steps methods.

The average VAF values in Tables 1 through 3 do not show the distribution of the VAF values. For example, a method may usually perform well, but occasionally give very poor results. Here, we show performance profiles of the VAF values for all methods and the original partition, for various

Table 4: Average adjusted Rand indices for each level of each factor.

Size of data set	$n = 60, m = 60$	$n = 150, m = 30$	$n = 120, m = 120$
Alternating exchanges	.739	.721	.846
Two-mode k-means	*.759*	*.748*	*.893*
Simulated annealing	.678	.673	.793
Tabu search	.636	.613	.729
Fuzzy steps	.745	.737	.851
Numbers of clusters	$K = 3, L = 3$	$K = 5, L = 5$	$K = 7, L = 7$
Alternating exchanges	.865	.772	.669
Two-mode k-means	*.868*	*.815*	*.718*
Simulated annealing	.830	.702	.612
Tabu search	.775	.637	.567
Fuzzy steps	.858	.783	.692
Error standard deviation	.5	1	2
Alternating exchanges	.915	.853	*.538*
Two-mode k-means	*.974*	*.889*	.537
Simulated annealing	.852	.788	.503
Tabu search	.782	.742	.454
Fuzzy steps	.934	.863	.537
Distribution of the objects	uniform distribution	10% in one cluster	60% in one cluster
Alternating exchanges	*.874*	*.861*	.571
Two-mode k-means	.873	.860	*.668*
Simulated annealing	.837	.805	.501
Tabu search	.741	.727	.510
Fuzzy steps	.868	.856	.610

Table 5: Average CPU times of all optimization methods in seconds.

Size of data sets	$n = 60, m = 60$			$n = 150, m = 30$			$n = 120, m = 120$		
Numbers of clusters (K, L)	3	5	7	3	5	7	3	5	7
Alternating exchanges	1.21	2.54	3.96	1.92	4.22	6.43	2.39	5.59	8.72
Two-mode k-means	1.54	2.36	3.09	1.96	3.58	4.57	2.50	5.31	7.66
Simulated annealing	1.08	2.30	3.54	1.75	3.82	5.78	2.42	5.47	8.49
Tabu search	0.82	2.16	3.74	1.47	3.82	6.77	2.34	6.04	10.75
Fuzzy steps	0.99	1.91	2.66	1.05	2.48	3.47	2.22	7.92	11.28

combinations of the levels of the four factors. As drawing these profiles for each of the 81 combinations of the factors requires too much space, we give three examples in Figures 4 through 6.

Figure 4 shows a performance profile of 50 data sets with a low error standard deviation. The value of the graph at the y-axis gives the fraction of times that a method achieved the best VAF value of all methods (including the original partition); this value is an upper bound on the fraction of times that a method managed to find the global optimum. The alternating exchanges, two-mode k-means method, and fuzzy steps methods found the best partition, which usually was the original partition, in every problem instance. The simulated annealing and tabu search methods usually also found

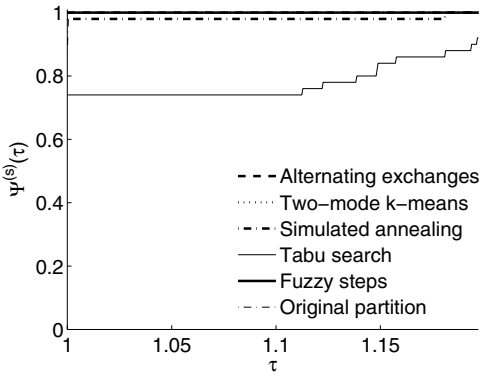

Figure 4: Performance profile of data sets with sizes $n = 150$, $m = 30$, $K = 3$, $L = 3$, error standard deviation 0.5, and a uniform distribution of the objects.

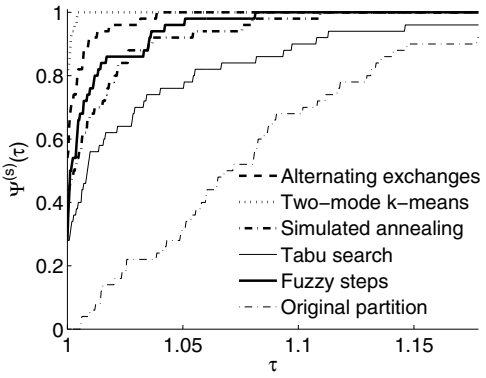

Figure 5: Performance profile of data sets with sizes $n = 60$, $m = 60$, $K = 5$, $L = 5$, error standard deviation 2, and one cluster containing 10% of the objects.

the best partition, but sometimes got stuck in an inferior solution. Figures 5 and 6 show performance profiles of data sets with a large error standard deviation, in which the optimal partition was hard to find. In these cases, no method performed at least as good as all others in every problem instance. In Figure 5, the two-mode *k*-means method performed best, followed by the alternating exchanges and fuzzy steps methods. This performance profile shows that each of these three methods almost always found a solution within 5 to 10 percent of the best solution found, in terms of the optimization criterion $f(\mathbf{P}, \mathbf{Q}, \mathbf{V})$. In Figure 6, each optimization method ended up in a non-global optimum for at least 46% of the problem instances. In this figure, the fuzzy steps and two-mode *k*-means methods had the best relative

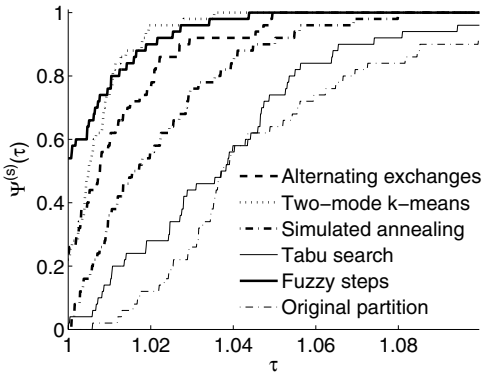

Figure 6: Performance profile of data sets with sizes $n = 120$, $m = 120$, $K = 7$, $L = 7$, error standard deviation 2, and one cluster containing 60% of the objects.

performances, followed by the alternating exchanges method. These three methods seldom find a solution that is more than 5 percent worse than the best solution found. The two implementations of meta-heuristics (simulated annealing and tabu search) do not perform as well as the other methods, but can usually find partitions that are better than the original partition.

As all optimization methods do not find a global optimum for a large number of problem instances represented in Figure 6, their performances may be considered unsatisfactory. If finding the global optimum is required, one may consider increasing the numbers of random starts in the optimization methods. To test the effectiveness of increasing the number of random starts, we applied the two-mode *k*-means and fuzzy steps methods to data sets similar to the ones used for Figure 6 with 50,000 and 2,000 random starts, respectively. The results show that increasing the numbers of random starts in this way (by a factor of 100) improves the performances of the methods only slightly; the average VAF values for the methods with increased numbers of random starts were approximately 0.2 percentage points higher than for the methods with the original numbers of random starts. However, even with these large numbers of random starts, no optimization method performed as well as all others for a majority of the data sets. These results show that all optimization methods have great difficulty in finding the global optimum in conditions such as represented in Figure 6. This problem is much less severe for smaller data sets and data sets with lower numbers of clusters and lower error standard deviations. As strongly increasing the number of random starts tends to improve the VAF value only slightly, we find it unlikely that the solution found by the optimization methods are much worse than the global optimum, in terms of the VAF criterion.

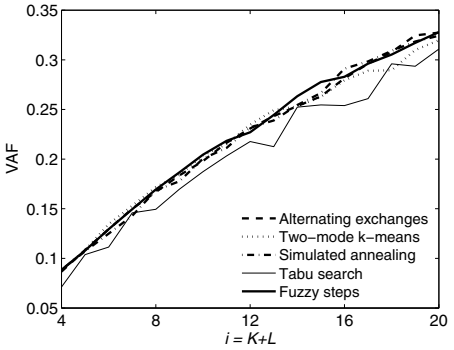

Figure 7: Best VAF values found in internet data set, for $K + L = 4 \ldots 20$.

6. Empirical Application

We use an empirical data set to illustrate two-mode partitioning and determine whether the conclusions of the simulation study are valid for empirical data. The data set is based on a questionnaire about the internet and comprises evaluations of 22 statements by 193 respondents. The statements were evaluated using a seven-point Likert scale, ranging from 1 (completely disagree) to 7 (completely agree). The average scores in the data set can differ significantly per individual and per statement. A sample run of the optimization methods shows that, if the raw data set is used, the individuals and the statements are mostly clustered based on their average scores. We correct for this problem by double centering the data matrix $\mathbf{X}$, that is, by replacing each x_{ij} with $\tilde{x}_{ij}$, where

$$\tilde{x}_{ij} = x_{ij} - \frac{1}{n}\sum_{i'=1}^{n} x_{i'j} - \frac{1}{m}\sum_{j'=1}^{m} x_{ij'} + \frac{1}{nm}\sum_{i'=1}^{n}\sum_{j'=1}^{m} x_{i'j'}, \qquad (17)$$

so that the row and column averages of $\tilde{\mathbf{X}}$ are zero; double centering accounts for 34% of the variance in $\mathbf{X}$.

As good values for the numbers of clusters are not known in advance, we use the following procedure for choosing K and L. First, we perform the optimization methods on $\tilde{\mathbf{X}}$, for all K and L such that $K + L = i$, for $i = 4, \ldots, 20$. For each value i and for each optimization method, we determine the values of K and L that yield the best VAF. The resulting VAF values are shown in Figure 7. The optimization methods were given the same parameters as in the simulation study. For this data set, the fuzzy steps method performed best on average, followed by the alternating exchanges, simulated annealing, and two-mode k-means methods. The tabu search

Table 6: Average evaluations per cluster, cluster sizes, and interpretations.

Respondent clusters	Statement clusters 1	2	3	4	5	Cluster size	Interpretation
1	-2.03	0.19	0.42	0.40	-0.89	36	enthusiasts
2	1.86	0.35	0.15	-0.34	-0.41	54	safety-conscious
3	-1.42	-0.02	-1.42	0.81	-0.16	32	experts
4	1.61	-0.37	-0.46	-0.28	1.06	37	skeptics
5	-1.21	-0.33	1.15	-0.33	0.59	34	price-conscious
Cluster size	1	6	3	8	4		
Interpretation	regulation	enthusiastic	expensive	experience	unreliable		

Table 7: Statement clusters in internet data set.

Cluster 1 (regulation):
The content of web sites should be regulated
Cluster 2 (enthusiastic):
Internet is easy to use
Internet is fast
Internet is the future's means of communication
Internet is user-friendly
Internet offers unbounded opportunities
Surfing the internet is easy
Cluster 3 (expensive):
Internet phone costs are high
The costs of surfing are high
The prices of internet subscriptions are high
Cluster 4 (experience):
I always attempt new things on the internet first
I know much about the internet
I like surfing
I like to be informed of important new things
I often speak with friends about the internet
I regularly visit web sites recommended by others
Internet is addictive
Paying using the internet is safe
Cluster 5 (unreliable):
Internet is slow
Internet is unreliable
Internet offers many possibilities for abuse
Transmitting personal data using the internet is unsafe

method performed poorly. These results appear to support the conclusions of the simulation study for data sets in which the optimal partition is hard to find.

The maximum of the VAF values shown in Figure 7 increases smoothly with the numbers of clusters, so that it is not clear which values of K and L should be chosen. To determine the numbers of clusters for two-mode partitioning, several selection criteria are discussed by Schepers, Ceulemans, and Mechelen (2008); they found the best performance using the numerical convex hull method. This criterion prefers $K = 2$ and $L = 2$,

with a VAF value of 8.8%. In this solution, both the statements and the respondents are divided into a cluster with a positive attitude and a cluster with a negative attitude. We find that this solution lacks a useful interpretation. As we seek to obtain a more fine-grained clustering, we select the solution with $K = 5$ and $L = 5$, based on the interpretability of the findings. The results of this solution are shown in Table 6. The associated VAF value is 20.4%, so that two-mode partitioning in combination with double centering accounts for 48% of the variance in the original data.

The statements in each statement cluster are shown in Table 7. Although the proportion of the variance of $\tilde{\mathbf{X}}$ explained by the clustering is relatively small (20.4%), it seems possible to interpret the clusters in a meaningful way. The statement clusters can be interpreted as follows. Cluster 1 comprises only one statement, which argues that the content of web sites should be regulated. Cluster 2 consists of statements that show an enthusiastic attitude towards the internet. The statements in cluster 3 state that using the internet is expensive. The statements in cluster 4 are associated with experienced users of the internet. Finally, the statements in cluster 5 consider the internet unreliable. To interpret the respondent clusters, we use the interpretations of the statement clusters and the values of the cluster centers in Table 6. The respondents in cluster 1 are ordinary internet users with a positive and enthusiastic attitude. Cluster 2 consists of people who mainly want regulation of the content of web sites. Cluster 3 consists of experienced internet users, who find using the internet quite cheap, but apparently are not enthusiastic about it. The respondents in cluster 4 seem to dislike using the internet and find it unreliable. The respondents in the final cluster seem to dislike using the internet because of the high costs.

7. Conclusions

Two-mode partitioning seems a powerful statistical technique, and a variety of optimization methods exist for finding good partitions. However, it remains unclear which optimization method has the best relative performance and is thus preferable in practice. We have tried to alleviate this problem by giving an overview of five optimization methods, one of which has been introduced in this paper.

Using a simulation study, the effects of various characteristics of the clustering problem on the performances of the methods were evaluated. We found that both the error variance and the relative sizes of the clusters strongly affect how well a clustering structure can be retrieved and can also affect the relative performances of the optimization methods. The two-mode *k*-means method most often had the best performance in the simulated data sets, especially if the characteristics of the problem were favorable.

When the optimal partition was hard to find, the alternating exchanges and fuzzy steps methods often performed best. The simulated annealing method also performed fairly well, but not as good as the three methods mentioned above. The performance of the tabu search method generally is inferior.

In sum, the best average performance is obtained using the two-mode k-means method, followed by the alternating exchanges and fuzzy steps methods. Which of these three methods gives the best performance depends on characteristics of the clustering problem. The performance profiles show that no optimization method can find the global optimum in a majority of the cases for every type of simulated data set. However, the alternating exchanges, two-mode k-means, and fuzzy steps methods almost always give a solution within 10% of the best solution found, in terms of the optimization criterion. As the two-mode k-means method requires relatively little CPU time and is easy to implement, we believe it is the best choice for data sets similar to the ones in this study. Note that performing a fairly large number (preferably more than 100) of random starts is necessary to ensure a good performance for this method. If finding a globally optimal partition is important, and computation time is not really an issue, the number of random starts should be set much higher. However, for data sets in which the optimal partition is hard to find, even a very large number of random starts may not ensure that the global optimum is found.

The results of the methods for the empirical data set are similar to the results of the simulation study. This data set also gives a useful example of the potential applications of two-mode partitioning. We believe that the clustering found in this data set is meaningful and provides relevant insight.

There are a few limitations associated with this study. First, we can only evaluate the relative performances of the methods for situations similar to our simulation study. It is possible that the relative performances of the methods are different for data sets with different characteristics. However, we have tried to make the simulated data sets similar to data sets that are typically found in practice. Therefore, the results should be reasonably similar for many empirical data sets. Second, the results of such a comparison of optimization methods depend on the values of certain parameters of the methods and the way these methods have been implemented. Here, we have tried to choose the parameters of the optimization methods in a sensible way that gives good performance. In addition, we have tried to make our computer programs as fast as possible. To do so, the efficient updating formulas that were given by Castillo and Trejos (2002) and Rocci and Vichi (2008) turn out to be quite important.

Some avenues for further research exist. For example, we cannot exclude the possibility that some optimization methods can be improved by choosing their parameters differently or implementing them in a different

way. Besides further theoretical research, we believe that further practical experience with two-mode partitioning is also required. Practice can show the real merits and drawbacks of using two-mode partitioning.

References

BAIER, D., GAUL, W., and SCHADER, M. (1997), "Two-Mode Overlapping Clustering with Applications in Simultaneous Benefit Segmentation and Market Structuring," in *Classification and Knowledge Organization*, eds. R. Klar and O. Opitz, Heidelberg: Springer, pp. 557-566.

BEZDEK, J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press.

BOCK, H. H. (1974), *Automatische Klassifikation*, Göttingen: Vandenhoeck & Ruprecht.

CASTILLO, W. and TREJOS, J. (2002), "Two-Mode Partitioning: Review of Methods and Application of Tabu Search," in *Classification, Clustering and Data Analysis*, eds. K. Jajuga, A. Sololowksi, and H. Bock, Berlin: Springer, pp. 43-51.

DESARBO, W. S. (1982), "GENNCLUS: New Models for General Nonhierarchical Clustering Analysis," *Psychometrika*, *47 (4)*, 449-475.

DOLAN, E. D. and MORÉ, J. J. (2002), "Benchmarking Optimization Software with Performance Profiles," *Mathematical Programming*, *91*, 201-213.

DOREIAN, P., BATAGELJ, V., and FERLIGOJ, A. (2004), "Generalized Blockmodeling of Two-Mode Network Data," *Social Networks*, *26*, 29-53.

DOREIAN, P., BATAGELJ, V., and FERLIGOJ, A. (2005), *Generalized Blockmodeling*, Cambridge University Press.

GAUL, W. and SCHADER, M. (1996), "A New Algorithm for Two-Mode Clustering," in *Data Analysis and Information Systems*, eds. H. Bock and W. Polasek, Heidelberg: Springer, pp. 15-23.

GLOVER, F. (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, *13*, 533-549.

GROENEN, P. J. F., and JAJUGA, K. (2001), "Fuzzy Clustering with Squared Minkowski Distances," *Fuzzy Sets and Systems*, *120*, 227-237.

HANSOHM, J. (2001), "Two-Mode Clustering with Genetic Algorithms," in *Classification, Automation, and New Media*, Berlin: Springer, pp. 87-93.

HARTIGAN, J. A. (1975), *Clustering Algorithms*, New York: John Wiley and Sons.

HEISER, W. J. and GROENEN, J. (1997), "Cluster Differences Scaling with a Within-Clusters Loss Component and a Fuzzy Successive Approximation Strategy to Avoid Local Minima," *Psychometrika*, *62(1)*, 63-83.

HUBERT, L. and ARABIE, P. (1985), "Comparing Partitions," *Journal of Classification*, *2*, 193-218.

MILLIGAN, G. W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, *45(3)*, 325-342.

MIRKIN, B. (2005), *Clustering for Data Mining: A Data Recovery Approach*, Boca Raton FL: Chapman & Hall.

MIRKIN, B., ARABIE, P., and HUBERT, L. J. (1995), "Additive Two-Mode Clustering: The Error-Variance Approach Revisited," *Journal of Classification*, *12*, 243-263.

NOMA, E. and SMITH, D. R. (1985), "Benchmark for the Blocking of Sociometric Data," *Psychological Bulletin*, *97*(3), 583-591.

ROCCI, R. and VICHI, M. (2008), "Two-Mode Multi-Partitioning," *Computational Statistics & Data Analysis*, *52*, 1984-2003.

SCHEPERS, J., CEULEMANS, E., and VAN MECHELEN, I. (2008), "Selecting among Multi-Mode Partitioning Models of Different Complexities: A Comparison of Four Model Selection Criteria," *Journal of Classification*, *25*, 67-85.

TREJOS, J. and CASTILLO, W. (2000), "Simulated Annealing Optimization for Two-Mode Partitioning," in *Classification and Information at the Turn of the Millenium*, eds. W. Gaul and R. Decker, Heidelberg: Springer, pp. 135-142.

TSAO, E. C. K., BEZDEK, J. C., and PAL, N. R. (1994), "Fuzzy Kohonen Clustering Networks," *Pattern Recognition*, *27(5)*, 757-764.

VAN LAARHOVEN, P. J. M. and AARTS, E. H. L. (1987), *Simulated Annealing: Theory and Applications*, Eindhoven: Kluwer Academic Publishers.

VAN MECHELEN, I., BOCK, H. H., and DE BOECK, P. (2004), "Two-Mode Clustering Methods: A Structured Overview," *Statistical Methods in Medical Research*, *13*, 363-394.

VICHI, M. (2001), "Double *k*-Means Clustering for Simultaneous Classification of Objects and Variables," in *Advances in Classification and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*, eds, S. Borra, R. Rocchi, and M. Schader, Heidelberg: Springer, pp. 43-52.