European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

Retrieving a contingency table from a correspondence analysis solution

Michel van de Velden*, Wilco van den Heuvel, Hugo Galy, Patrick J. F. Groenen

Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, Rotterdam, DR 3000, the Netherlands

ARTICLE INFO

Article history:
Received 6 September 2018
Accepted 9 November 2019
Available online xxx

Keywords:
Multivariate statistics
Inverse problems
Integer programming
Correspondence analysis

ABSTRACT

Correspondence analysis (CA) is a dimension reduction technique for categorical data. In particular, CA is typically applied to a contingency matrix in order to visualize the relationships within and between the categories of the two variables as represented by its rows and columns. The CA solution can be obtained by considering a singular value decomposition of the so-called matrix of standardized residuals. Inverse correspondence analysis considers the problem of retrieving the data underlying a given low-dimensional CA solution. Using the specific structure of the CA solutions as well as the characteristics of the original data we formulate the inverse CA problem in an integer linear programming context. Considering various conditions involving the dimensions of the original data matrix, the number of observations, the precision and dimensionality of the CA solution, we show that by solving the integer linear programs, the original data can be retrieved.

1. Introduction

In correspondence analysis (CA; Benzécri, 1973; Greenacre, 2007) the aim is to obtain a low-dimensional representation that optimally depicts associations in a two-way contingency table. Typical output of CA is a plot in which the categories for both variables are depicted in such a way that the associations are optimally represented. For example, Torres and Bijmolt (2009) used CA to study the relationship between brand-attribute assocations; Maiti, Singh, Mandal, and Verma (2014) use CA to study associations of categories of factors contributing to derailements occurring in a steel plant in order to develop meaningful rules to prevent such derailments, and Moya and Jain (2013) study the association between personality traits observed in promotional Facebook messages of popular tourist destinations with the personality traits observed in Facebook posts of those destinations' Facebook friends.

In inverse correspondence analysis (Groenen & Van de Velden, 2004), the problem of retrieving the data underlying a given low-dimensional CA solution was studied. Retrieving a contingency table from a CA solution can be of interest in case only the CA solution is available. Published papers are not always accompanied by the data table underlying the results. If authors can not be reached or are unwilling to share their data, it is not possible to verify results and/or re-analyse the data using alternative methodology. This problem can be resolved using inverse CA. On the other hand, in some cases, researchers are not allowed to share the original data. If inverse CA can successfully retrieve the original contingency table, authors should in such instances be careful when publishing only CA results. This problem is relevant in situations where information disclosure is important as inverse CA allows the reconstruction of the original contingency table out of the CA solution only. In this paper, we consider published CA solutions, that is, the low-dimensional visualization of associations, from a recent academic publication and apply our methodology to retrieve the original, unpublished, contingency tables.

Previous work on inverse CA (i.e., Groenen & Van de Velden, 2004), resulted in two algorithms for obtaining a set of vertices that define a set of matrices for which any convex combination yields the given low dimensional CA solution. For relatively small problems (i.e., data matrices up to 5 by 5, with a two dimensional CA solution known), the first algorithm, using full enumeration, results in the full set of such vertices. For larger problems, full enumeration is no longer feasible and a heuristic approach was proposed. Using the corresponding algorithm, however, only a subset of vertices may be obtained.

One basic assumption in CA is that the values of the original data are nonnegative and that no row or column margins are zero. Therefore, an important aspect in solving the inverse CA

* Corresponding author.
E-mail address: vandevelden@ese.eur.nl (M. van de Velden).

https://doi.org/10.1016/j.ejor.2019.11.014
0377-2217/

problem is that the unknown elements of the data matrix are constrained to be nonnegative. However, as a contingency table consists of counts of co-occurrences, the elements are not only nonnegative but are in fact integer valued. This property, although mentioned in Groenen and Van de Velden (2004), was not explicitly taken into account when solving the inverse problem. In this paper, we add the integrality constraints to the inverse CA problem and reformulate it as an integer programming feasibility problem.

The contribution of this paper is as follows. First of all, we extend the work of Groenen and Van de Velden (2004) by explicitly taking into account integrality constraints. This leads to an integer programming feasibility problem, for which we propose several formulations and an acceleration strategy. Secondly, numerical tests on randomly generated tables show that we can solve fairly large problem instances in a reasonable amount of time. Moreover, we demonstrate the effectiveness of our approach by retrieving the original contingency tables (in case the problem is solved to optimality and the precision of the data is sufficiently high). Finally, the effectiveness is confirmed by a case study on a recent academic publication, where we are able to retrieve the original, unpublished, contingency tables by applying our methodology.

This paper is organized as follows. In Section 2, we briefly summarize the inverse CA problem using similar notation and formulation as the original inverse CA paper. Next, the integrality constraints are added and three integer programming models that can be used to generate solutions are formulated in Section 3. In Section 4, a simulation study is employed to appraise the different models with respect to speed and validity and to establish under which scenarios we are able to retrieve the original contingency table from the solution. In particular, we consider different scenarios with respect to the dimensions of the original tables, as well as different scenarios concerning the rounding of the CA solutions used to retrieve the original data. In Section 5, we apply our methods to a real case where we take published CA solutions from a paper to retrieve the, unpublished, contingency tables underlying these solutions. We conclude the paper by a discussion of our results and we give possible directions for future research.

2. Correspondence analysis

The CA solution is based on a least-squares approximation of the so-called matrix of standardized residuals. This least-squares approximation is based on a singular value decomposition (SVD). Focusing on the most important formulas and equations rather than on interpretation, justification and derivations, we first briefly introduce CA. For a more indepth treatment of CA we refer to Greenacre (2007) and Van de Velden (2000).

Let $\mathbf{F}$ denote the n_r by n_c contingency table containing the co-occurrences of two categorical variables based on s observations. Furthermore, let $\mathbf{D}_r$ and $\mathbf{D}_c$ denote diagonal matrices of row and column totals $\mathbf{r}$ and $\mathbf{c}$, respectively. Hence, $\mathbf{r} = \mathbf{F}\mathbf{1}_{n_c} = \mathbf{D}_r\mathbf{1}_{n_c}$ and $\mathbf{c} = \mathbf{F}'\mathbf{1}_{n_c} = \mathbf{D}_c\mathbf{1}_{n_r}$, where, generically, $\mathbf{1}_n$ denotes an $n \times 1$ vector of ones. Consider the singular value decomposition

$$\tilde{\mathbf{F}} = \mathbf{D}_r^{-1/2}\mathbf{F}\mathbf{D}_c^{-1/2} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}', \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal and $\mathbf{\Lambda}$ is a diagonal matrix of singular values in non-increasing order.

The largest singular value of $\tilde{\mathbf{F}}$ is 1 and the corresponding columns of $\mathbf{U}$ and $\mathbf{V}$ are $\frac{1}{\sqrt{s}}\mathbf{D}_r^{1/2}\mathbf{1}_{n_r}$ and $\frac{1}{\sqrt{s}}\mathbf{D}_c^{1/2}\mathbf{1}_{n_c}$ respectively, (for a proof of this property see Van de Velden & Neudecker, 2000). This first solution is typically referred to as a trivial solution and discarded. A k-dimensional CA solution can be obtained from the singular value decomposition in Eq. (1), by ignoring the trivial dimension and selecting the next k singular values and corresponding columns of $\mathbf{U}$ and $\mathbf{V}$. More precisely, row and column coordinate matrices in CA are defined as

$$\mathbf{R}_k = \mathbf{D}_r^{-1/2}\mathbf{U}_k\mathbf{\Lambda}_k^{\alpha}, \tag{2}$$

and

$$\mathbf{C}_k = \mathbf{D}_c^{-1/2}\mathbf{V}_k\mathbf{\Lambda}_k^{1-\alpha}, \tag{3}$$

where the subscripted k's indicate the selected number of columns from $\mathbf{U}$ and $\mathbf{V}$, and corresponding rows and columns of $\mathbf{\Lambda}$. The constant α is chosen by the user and is typically either 0, 1/2, or 1. For more details on the meaning and implications of the different choices of α see, for example, Van de Velden and Kiers (2005) or Beh and Lombardo (2014).

2.1. Inverse correspondence analysis

Consider the partitioned matrices

$$\mathbf{U} = \begin{pmatrix}\mathbf{U}_{k+1} & \mathbf{U}_c\end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix}\mathbf{V}_{k+1} & \mathbf{V}_c\end{pmatrix} \text{ and } \mathbf{\Lambda} = \begin{pmatrix}\mathbf{\Lambda}_{k+1} & \\ & \mathbf{\Lambda}_c\end{pmatrix}$$

so that

$$\begin{aligned}\tilde{\mathbf{F}} &= \begin{pmatrix}\mathbf{U}_{k+1} & \mathbf{U}_c\end{pmatrix}\begin{pmatrix}\mathbf{\Lambda}_{k+1} & \\ & \mathbf{\Lambda}_c\end{pmatrix}\begin{pmatrix}\mathbf{V}_{k+1}' \\ \mathbf{V}_c'\end{pmatrix} \\ &= \mathbf{R} + \mathbf{U}_c\mathbf{\Lambda}_c\mathbf{V}_c',\end{aligned} \tag{4}$$

where $\mathbf{R} = \mathbf{U}_{k+1}\mathbf{\Lambda}_{k+1}\mathbf{V}_{k+1}'$ gives the $k+1$ dimensional approximation of $\tilde{\mathbf{F}}$, that is the first $k+1$ singular values and corresponding vectors, including the trivial ones corresponding to singular value 1, and $\mathbf{U}_c$, $\mathbf{V}_c$ and $\mathbf{\Lambda}_c$ are the additional, unknown, singular vectors and values of appropriate dimensionalities from the complete singular value decomposition of $\tilde{\mathbf{F}}$.

As $\mathbf{U}_c$ and $\mathbf{V}_c$ are orthogonal to $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ respectively, we can find $\mathbf{U}_c$ and $\mathbf{V}_c$ by calculating the nullspaces of $\mathbf{U}_{k+1}'$ and $\mathbf{V}_{k+1}'$ That is,

$$\mathbf{U}_{k+1}'\mathbf{U}_0 = \mathbf{0}, \text{ and } \mathbf{V}_{k+1}'\mathbf{V}_0 = \mathbf{0}$$

so that, for any orthonormal matrices $\mathbf{T}$ and $\mathbf{Q}$, we have

$$\mathbf{U}_c = \mathbf{U}_0\mathbf{T}, \text{ and } \mathbf{V}_c = \mathbf{V}_0\mathbf{Q}.$$

Hence, the unknown matrices $\mathbf{U}_c$ and $\mathbf{V}_c$ can be obtained up to rotation. Substituting these expressions in Eq. (4) gives

$$\begin{aligned}\tilde{\mathbf{F}} &= \mathbf{R} + \mathbf{U}_0\mathbf{T}\mathbf{\Lambda}_c\mathbf{Q}'\mathbf{V}_0' \\ &= \mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0',\end{aligned} \tag{5}$$

where $\mathbf{G} = \mathbf{T}\mathbf{\Lambda}_c\mathbf{Q}'$.

For the inverse CA problem, the marginals (i.e., $\mathbf{D}_r$ and $\mathbf{D}_c$) and the k dimensional solution $\mathbf{R}$ are known. Given these marginals, we want to identify the set of CA data matrices for which $\mathbf{R}$ is a solution. Using Eqs. (1) and (5), we see that such reconstituted data matrices $\mathbf{F}_{rec}$ satisfy

$$\mathbf{F}_{rec} = \mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}. \tag{6}$$

In fact, it is not difficult (cf. Lemmas 1 and 2 in Groenen & Van de Velden, 2004) to show that for any choice of $\mathbf{G}$, applying CA to Eq. (6), yields the known correspondence analysis solution $\mathbf{R}$. However, there is no guarantee that the solution shows up as the first k dimensions and that the retrieved matrix is nonnegative containing only integers. In inverse CA, we want the reconstructed matrix to correspond to a CA data matrix, that is, to a contingency table.

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

2.2. Inverse Correspondence Analysis with integrality constraints

Groenen and Van de Velden (2004) studied the inverse CA problem described in the previous subsection. An algorithm was proposed to find a set of reconstructed (not necessarily integral) **F** matrices corresponding to the vertices (also known as extreme points) of the (polyhedral) set of matrices **G** satisfying Eq. (6) and nonnegativity of $\mathbf{F}_{rec}$. Moreover, they showed that any convex combination of the obtained set of vertices also provides a matrix satisfying the constraints. However, the proposed algorithm, which was based on complete enumeration, was only feasible for relatively small problems. For larger problems, a heuristic approach was proposed that yielded a subset of vertices.

From Eq. (6) it follows that the inverse correspondence analysis problem amounts to finding a matrix **G** in such a way that a set of constraints is satisfied. In particular, Groenen and Van de Velden (2004) considered the problem of finding **G** resulting in a nonnegative $\mathbf{F}_{rec}$. However, if the underlying data matrix is in fact a contingency table, the correct constraint would be that all entries of $\mathbf{F}_{rec}$ are nonnegative integers.

Note that this formulation of the problem constitutes a feasibility problem. That is, we only have constraints but no objective function to optimize. For the heuristic inverse CA approach of Groenen and Van de Velden (2004), a linear combination of the elements of the unknown matrix **G** was chosen as the objective function. This objective was then optimized subject to a set of inverse CA constraints (resulting from the nonnegativity constraint of the data). By considering several (random) linear combinations, a set of solutions (i.e., vertices), can be obtained. However, whether the obtained set is the full set, or only a subset, cannot be determined in this way.

3. Integer programming approach

3.1. The basis models

Instead of using a heuristic approach, we now take an exact approach by reformulating the problem as a mixed integer linear programming problem (MIP). This is useful, as the theory of solving MIPs is well-understood (e.g., see Wolsey & Nemhauser, 2014) and they can be solved by off-the-shelf commercial solvers like Cplex or Gurobi, in which state-of-the-art branch-and-bound techniques are used. Compared to Groenen and Van de Velden (2004), we add integrality constraints in addition to the nonnegativity constraints. Furthermore, instead of considering random linear combinations of **G**, we propose an objective function in which we minimize the amount of violation of the constraints, which should be zero if a feasible solution exists. Formally, we are looking for a solution $\mathbf{F}_{rec}$ from the set

$$\mathcal{F} = \{\mathbf{F}_{rec} \in \mathbb{N}^{n_r \times n_c} : \mathbf{F}_{rec} = \mathbf{D}_r^{1/2}(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0')\mathbf{D}_c^{1/2}, \mathbf{G} \in \mathbb{R}^{(n_r-(k+1))\times(n_c-(k+1))}\}. \quad (7)$$

We propose several mixed integer linear programming (MIP) formulations to solve this feasibility problem. In the first formulation we keep the original equalities and introduce a variable $z_{ij} \in \mathbb{R}$ for each equality, representing the amount by which the equality is violated. The objective function to be minimized is the sum of the absolute values of these violations $\sum_{i=1}^{n_r}\sum_{j=1}^{n_c}|z_{ij}|$. If the set as defined in Eq. (7) is non-empty, this sum should be zero. Since absolute values lead to a non-linear objective function, we apply a common approach and replace the real variable z_{ij} by nonnegative variables z_{ij}^+ and z_{ij}^- which represent the positive and negative part of z_{ij}, respectively. After computing the null spaces $\mathbf{U}_0$ and $\mathbf{V}_0$, and applying the substitution $z_{ij} = z_{ij}^+ - z_{ij}^-$, we obtain our first model, MIP1:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n_r}\sum_{j=1}^{n_c}(z_{ij}^+ + z_{ij}^-) \\
\text{s.t.} \quad & f_{ij} = \left(\mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}\right)_{ij} + (z_{ij}^+ - z_{ij}^-) \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& f_{ij} \in \mathbb{N} \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& z_{ij}^+, z_{ij}^- \geq 0 \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& g_{ij} \in \mathbb{R} \\
& \quad \text{for } i = 1,\ldots,n_r-(k+1);\ j = 1,\ldots,n_c-(k+1).
\end{aligned}$$

Note that in an optimal solution, either z_{ij}^+ or z_{ij}^- will be zero so that they correctly represent the absolute value of z_{ij} by $|z_{ij}| = z_{ij}^+ + z_{ij}^-$.

An alternative way of modelling the problem, requiring fewer variables but more constraints, is by changing the equality in equation (6) into two inequalities (a less-than-or-equal and greater-than-or-equal inequality) and by introducing a variable z_{ij} representing the (nonnegative) violation of these new inequality constraints corresponding to element $f_{ij} = (\mathbf{F}_{rec})_{ij}$. This leads to our second integer linear programming problem, referred to as MIP2:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n_r}\sum_{j=1}^{n_c} z_{ij} \\
\text{s.t.} \quad & f_{ij} \leq \left(\mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}\right)_{ij} + z_{ij} \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& f_{ij} \geq \left(\mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}\right)_{ij} - z_{ij} \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& f_{ij} \in \mathbb{N} \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& z_{ij} \geq 0 \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& g_{ij} \in \mathbb{R} \\
& \quad \text{for } i = 1,\ldots,n_r-(k+1);\ j = 1,\ldots,n_c-(k+1).
\end{aligned}$$

Finally, instead of minimizing the violation over all constraints, we can minimize the maximum violation z over the constraints. The advantage of this approach compared to the previous formulation is that we need fewer variables. This results in model MIP3:

$$\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad & f_{ij} \leq \left(\mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}\right)_{ij} + z \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& f_{ij} \geq \left(\mathbf{D}_r^{1/2}\left(\mathbf{R} + \mathbf{U}_0\mathbf{G}\mathbf{V}_0'\right)\mathbf{D}_c^{1/2}\right)_{ij} - z \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& f_{ij} \in \mathbb{N} \\
& \quad \text{for } i = 1,\ldots,n_r;\ j = 1,\ldots,n_c, \\
& g_{ij} \in \mathbb{R} \\
& \quad \text{for } i = 1,\ldots,n_r-(k+1);\ j = 1,\ldots,n_c-(k+1), \\
& z \geq 0.
\end{aligned}$$

It is not difficult to see that for each of the three formulations there exists a solution $\mathbf{F}_{rec} \in \mathcal{F}$ if and only if the optimal objective value is zero. Moreover, the formulations have a nice interpretation when the input data is imprecise. Such imprecisions happen naturally when (i) the CA solution (i.e., the coordinate matrices) is given in a small number of digits, or (ii) when it is estimated from a pictorial representation (for example, by using a ruler). Clearly, in the case of imprecise input there may not exist a solution satisfying Eq. (7), that is, the set $\mathcal{F}$ may be empty. However, our models (MIP1–3) always yield a feasible solution, although the solutions do not necessarily reach the optimal objective value of zero. Since we minimize the sum of the violations

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

over all constraints, an optimal solution can be interpreted as a contingency table $\mathbf{F}_{rec}$ that satisfies the constraints of Eq. (7) as 'close' as possible, i.e., with minimal violation of the constraints. Note that different measures for 'close' can be taken. The objective functions of MIP1–3 represent three natural choices.

3.2. Checking uniqueness

An interesting question is whether an optimal solution of MIP1–3 is unique. An affirmative answer would mean that the application of CA has not led to a loss of information, as the original contingency table can still be retrieved. Checking uniqueness is equivalent to checking whether a second solution exists. Suppose that $\mathbf{F}^1_{rec}$ is an optimal solution of MIP1–3. To check uniqueness of this solution we can adjust the formulations MIP1–3 by introducing a binary variable y_{ij}, which equals 1 if $\mathbf{F}_{rec}$ is different from $\mathbf{F}^1_{rec}$ in element (i, j). We add the constraints

$$\sum_{i=1}^{n^r}\sum_{j=1}^{n^c} y_{ij} \geq 1 \tag{8}$$

$$(\mathbf{F}_{rec})_{ij} \geq (\mathbf{F}^1_{rec})_{ij} + (M+1)y_{ij} - M \quad \text{for} \quad i=1,\ldots,n_r;\ j=1,\ldots,n_c, \tag{9}$$

$$y_{ij} \in \mathbb{B} \quad \text{for} \quad i=1,\ldots,n_r;\ j=1,\ldots,n_c, \tag{10}$$

where M is a number at least equal to the largest element of $\mathbf{F}^1_{rec}$ in any optimal solution. Constraint (8) forces at least one of the binary variables y_{ij} to be one. In turn, this implies by Eq. (9) that at least one element of $\mathbf{F}_{rec}$ will be larger than the corresponding element in $\mathbf{F}^1_{rec}$, which should be the case in an alternative solution as the sum over all elements equals s. It is not difficult to see that the optimal objective value of the problem with the additional constraints is zero if and only if the original problem MIP1–3 has a second solution (i.e., $\mathcal{F}$ has at least two elements).

We can repeatedly apply this approach to find all optimal solutions. Each time we find an alternative optimal solution, we add constraints of type (9) and (10) corresponding to the alternative solution to the problem formulation until the optimal objective value becomes positive.

4. Computational results

In this section we perform some computational experiments in order to test the effectiveness of our models. In particular, the main questions we are interested in are:

- Given the best formulation and setting, what problem sizes, that is the numbers of rows and columns as well as the number of observations underlying the contingency tables, can we solve in a reasonable amount of time?
- What is the effect of various degrees of imprecision (i.e., rounding) of the input?
- How often are we able to retrieve the original data?

In the following subsections we consider these questions by applying the integer programming approach of Section 3 on randomly generated tables. All tests are performed on an Intel Xeon @ 3.50 gigahertz with 4 Cores and 8 threads, and 16.0 gigabytes RAM. The MIP formulations are solved with Cplex 12.6.2, where we have set the parameter for numerical emphasis to true to avoid loss of numerical accuracy.

4.1. Comparing the MIP formulations

As a preliminary experiment, we tested the three different MIP formulations for contingency tables of different dimensions $n_r \times n_c$ and different numbers of observations s. It turned out that it is useful to add an additional constraint bounding the variables z_{ij} by some small ε. That is, by adding constraints of type $z_{ij} \leq \varepsilon$ to the models. We can do this because we know that at least one feasible solution exists. Apparently, setting this value of ε helps the branch-and-bound method to quickly cut off infeasible solutions, improving the solution speed. However, setting ε too small may lead to numerical problems. Therefore, we activated the parameter "Numerical Emphasis" in Cplex which should avoid loss of numerical accuracy as much as possible.

To test our procedures we randomly generated 20 tables per setting. A single table is constructed by making a contingency table from s uniformly distributed random drawings from two categorical variables having n_r and n_c categories. From this contingency table a $k+1=3$-dimensional CA solution is computed which is used as input for our models. Furthermore, the MIP solver is terminated after 15 minutes. The average (possibly truncated) runtimes and standard deviations, in seconds, as well as the number of times the problem is solved to optimality, are reported in Table 1 for models MIP1–MIP3.

In general, we observe that runtimes go down as ε decreases, which validates the addition of the constraints $z_{ij} \leq \varepsilon$. Furthermore, we see that runtimes significantly increase with the number of observations s: we can retrieve all generated tables with $s = 250$ for at least one setting, while this does not hold for the 8×10 and 10×10 cases and $s = 500$. It appears that there is no clear formulation that performs best over all tables. MIP1 performs well for small s and small ε, while MIP3 performs better for larger s and ε. Furthermore, for nearly all settings MIP2 is outperformed, with respect to time as well as the number of problems solved by MIP1 and MIP3 for $s = 500$.

4.2. Results without rounding

Based on the preliminary tests, we ran experiments for models MIP1 and MIP3 with $k+1=3$ and $\varepsilon = 0.0001$ where we generated 20 tables per setting and terminated the MIP after 15 minutes. Table 2 shows the results, where the columns contain (i) the problem dimensions, (ii) number of observations, (iii) average and (iv) standard deviation of the runtime in seconds, (v) the number of times the original data was found, (vi) the number of times an optimal solution is found, and (vii) the number of times the MIP solver hit the time limit. As the table shows, when an optimal solution was found, this solution was always equal to the original contingency table. Furthermore, using these formulations, we are often able to solve even for tables of fairly large dimensionality within 15 minutes of runtime.

4.3. Results with rounding

In practical situations, in particular, in published results, it is unlikely that one obtains the exact output from a CA. It is more likely that the CA output is available after being rounded to a certain number of digits. Therefore, we perform some computational tests with rounded data and report the results in this section. Note that we need $\mathbf{R}$, $\mathbf{U}_0$ and $\mathbf{V}_0$ as input for the inverse CA, cf. equation (6). These matrices, however, are typically not reported and first need to be calculated from the CA output.

Suppose that CA has been performed on some $\mathbf{F}$, with as output $\mathbf{R}_{k+1}$ and $\mathbf{C}_{k+1}$, which is reported as $\hat{\mathbf{R}}_{k+1}$ and $\hat{\mathbf{C}}_{k+1}$ after rounding (we use a hat to denote that rounding has occurred). Using the analysis of Section 2.1 (assuming $\alpha = 1$), we can obtain the rounded inverse CA input by executing the following steps:

1. compute $\hat{\mathbf{R}} = \mathbf{D}_r^{1/2}\hat{\mathbf{R}}_{k+1}\hat{\mathbf{C}}'_{k+1}\mathbf{D}_c^{1/2}$
2. compute $\hat{\mathbf{\Lambda}} = \hat{\mathbf{R}}'_{k+1}\mathbf{D}_r\hat{\mathbf{R}}_{k+1}$

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

Table 1
Running times, in seconds, and number of problems solved by models MIP1–MIP3.

$n_r \times n_c$	ε	$s = 250$									$s = 500$								
		(MIP1)			(MIP2)			(MIP3)			(MIP1)			(MIP2)			(MIP3)		
		avg	stdv	opt	avg	stdv	opt	avg	stdv	opt	avg	stdv	opt	avg	stdv	opt	avg	stdv	opt
6 × 6	0.0001	0.2	0.1	20	0.2	0.1	20	0.3	0.1	20	0.5	0.4	20	0.6	0.5	20	0.5	0.5	20
6 × 6	0.001	0.2	0.1	20	0.2	0.1	20	0.3	0.1	20	0.5	0.5	20	0.5	0.6	20	0.7	0.5	20
6 × 6	0.01	0.2	0.1	20	0.3	0.1	20	0.3	0.1	20	0.4	0.4	20	0.6	0.5	20	0.7	0.6	20
6 × 8	0.0001	0.4	0.3	20	0.5	0.5	20	0.3	0.1	20	7.5	9.3	20	13.1	27	20	5.5	6.7	20
6 × 8	0.001	0.3	0.3	20	0.6	0.7	20	0.4	0.4	20	7.2	11.1	20	9	14.2	20	12.6	27.2	20
6 × 8	0.01	0.4	0.3	20	0.6	0.6	20	0.5	0.5	20	30.2	86.7	20	42.3	132	20	13.9	19.4	20
6 × 10	0.0001	1.2	1.7	20	1.6	1.5	20	1.8	2.2	20	201	255.2	20	220.3	315.3	18	169	269.2	18
6 × 10	0.001	1.4	1.3	20	1.4	1.6	20	1.5	1.5	20	102.4	147	20	189.6	285.9	18	158.1	195.7	20
6 × 10	0.01	1.5	1.5	20	1.9	1.9	20	1.5	1.8	20	88.2	180.6	20	230.1	334.6	18	137.3	269.8	18
8 × 8	0.0001	1.4	0.8	20	1.5	0.9	20	2	1.6	20	77.3	173.7	20	186.6	285.3	18	123.7	231.1	19
8 × 8	0.001	3.4	5	20	2.2	2.2	20	1.6	0.8	20	48.2	119.8	20	142.1	253.1	20	135.9	289.5	18
8 × 8	0.01	2.3	1.2	20	2.6	1.8	20	1.8	0.6	20	190.3	275.3	19	219.9	323.7	17	231.1	341.1	17
8 × 10	0.0001	4.8	5.4	20	5.7	6.2	20	6.4	6.6	20	603.7	366.1	10	679.6	317.8	8	451.5	371	15
8 × 10	0.001	5.6	4.5	20	4.9	4.3	20	7.6	6.5	20	425.7	363.1	15	581.7	349	12	332.7	342.5	17
8 × 10	0.01	16	32.4	20	8.2	12.2	20	5.2	4.4	20	625	354.7	10	786.5	257.9	4	806.5	214.7	6
10 × 10	0.0001	72.1	195.7	20	60.6	198.5	19	68.3	105.5	20	733.5	309.1	5	773.9	273.4	4	736.6	269.8	7
10 × 10	0.001	79.1	137.1	20	58.2	136.3	20	17.5	19	20	669.1	318.5	8	679.2	337.8	7	610.2	360.8	9
10 × 10	0.01	245.8	343.8	16	220.8	354.5	16	202.6	316.7	17	845.1	194.2	2	870.1	90.2	3	844.2	183.8	2

Table 2
Running times, in seconds, and solution status at time limit for model MIP1 and MIP3 when starting with accurate CA output as input.

$n_r \times n_c$	s	model (MIP1)					model (MIP3)				
		avg	stdv	#orig	#opt	#limit	avg	stdv	#orig	#opt	#limit
6 × 6	250	0.24	0.07	20	20	0	0.25	0.08	20	20	0
6 × 6	500	0.51	0.44	20	20	0	0.46	0.45	20	20	0
6 × 6	1000	1.54	1.97	20	20	0	1.98	1.93	20	20	0
6 × 8	250	0.39	0.35	20	20	0	0.27	0.08	20	20	0
6 × 8	500	7.48	9.34	20	20	0	5.5	6.69	20	20	0
6 × 8	1000	157.46	239.41	19	19	1	267.89	379.61	15	15	5
6 × 10	250	1.23	1.69	20	20	0	1.79	2.25	20	20	0
6 × 10	500	201.03	255.15	20	20	0	169.02	269.18	18	18	2
6 × 10	1000	818.38	256.69	3	3	17	601.14	419.85	7	7	13
8 × 8	250	1.36	0.77	20	20	0	2	1.56	20	20	0
8 × 8	500	77.33	173.71	20	20	0	123.75	231.09	19	19	1
8 × 8	1000	597.68	303.77	13	13	7	627.48	341.7	9	9	11
8 × 10	250	4.84	5.39	20	20	0	6.39	6.61	20	20	0
8 × 10	500	603.71	366.14	10	10	10	451.52	370.99	15	15	5
8 × 10	1000	901.7	0.64	0	0	20	900.28	0.14	0	0	20
10 × 10	250	72.13	195.73	20	20	0	68.28	105.51	20	20	0
10 × 10	500	733.54	309.13	5	5	15	736.62	269.76	7	7	13
10 × 10	1000	901.33	0.55	0	0	20	900.2	0.11	0	0	20

3. compute $\hat{\mathbf{U}}_{k+1} = \mathbf{D}_r^{1/2}\hat{\mathbf{R}}_{k+1}(\hat{\mathbf{\Lambda}}_{k+1}^{\alpha})^{-1}$ and $\hat{\mathbf{V}}_{k+1} = \mathbf{D}_c^{1/2}\hat{\mathbf{C}}_{k+1}(\hat{\mathbf{\Lambda}}_{k+1}^{1-\alpha})^{-1}$
4. compute $\hat{\mathbf{U}}_0 = \text{null}(\hat{\mathbf{U}}_{k+1})$ and $\hat{\mathbf{V}}_0 = \text{null}(\hat{\mathbf{V}}_{k+1})$

In summary, the rounded input for the inverse CA consists of $\hat{\mathbf{R}}$, and $\hat{\mathbf{U}}_0$ and $\hat{\mathbf{V}}_0$ obtained from Steps 1 and 4.

We repeated the experiments of the previous section but now using rounded input. Tables 3 and 4 show the results when rounding was done to 3 or 4 digits, respectively. We used model MIP3 and set $\varepsilon = 1$ relatively high in order to obtain a feasible solution. Furthermore, we added the column '#feas' indicating the number of times a feasible but not provably optimal solution was found after reaching the time limit (note that there were no infeasible solutions when the time limit was reached). These results suggest that we can solve a much smaller number of generated tables to optimality due to the choice of ε, which slows down the MIP solver. Furthermore, the tables show that when the input is rounded to 3 digits, it becomes very hard to retrieve the original data, while rounding to 4 digits gives much better results. Note also that it sometimes happens that we retrieve the original data, although the MIP was terminated because of the set time limit (see Table 4, the row corresponding to a 6×8 table with $s = 500$).

Table 3
Running times, in seconds, and solution status at time limit for model (MIP1) and (MIP3) when starting with CA output rounded to 3 digits.

$n_r \times n_c$	s	avg	stdv	#orig	#opt	#feas	#limit
6 × 6	250	16.49	25.83	4	20	0	0
6 × 6	500	512.21	387.91	0	12	8	8
6 × 8	250	205.45	276.97	2	18	2	2
6 × 8	500	906.43	3.1	0	0	20	20
6 × 10	250	662.30	314.36	2	9	11	11
6 × 10	500	906.52	2.08	0	0	20	20
8 × 8	250	905.44	2.68	0	0	20	20
8 × 8	500	908.53	2.23	0	0	20	20
8 × 10	250	905.88	2.07	0	0	20	20
8 × 10	500	908.00	2.24	0	0	20	20
10 × 10	250	906.86	1.49	0	0	20	20
10 × 10	500	908.18	0.49	0	0	20	20

4.4. Uniqueness

In the (non-rounded) cases run so far, it turns out that we always find back the original data if the problem was solved to optimality within the given time limit. To see whether this holds

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

Table 4
Running times, in seconds, and solution status at time limit for model (MIP1) and (MIP3) when starting with CA output rounded to 4 digits.

$n_r \times n_c$	s	avg	stdv	#orig	#opt	#feas	#limit
6×6	250	1.59	1.12	20	20	0	0
6×6	500	43.38	57.77	20	20	0	0
6×8	250	6.97	15.2	20	20	0	0
6×8	500	553.93	357.8	14	11	9	9
6×10	250	9.51	10.02	20	20	0	0
6×10	500	796.93	270.04	4	3	17	17
8×8	250	468.08	364.58	15	14	6	6
8×8	500	908.50	1.51	0	0	20	20
8×10	250	799.47	235.37	4	4	16	16
8×10	500	907.64	1.36	0	0	20	20
10×10	250	906.01	1.24	0	0	20	20
10×10	500	907.14	1.07	0	0	20	20

even more generally, we performed a more extensive test by generating 10,000 contingency matrices of dimensions 6×6, each containing $s = 250$ observations, and applying CA to each table using $k + 1 = 3$ dimensions. Surprisingly, we retrieve the original data in all cases. Moreover, the same happens when setting $k + 1 = 2$, which is even more surprising, as the dimensionality of the data is even further reduced.

It could be a coincidence that we always retrieve the original data, while alternative solutions exist (note that the MIP solver terminates once a single optimal solution is found). Therefore, we ran the MIP model of Section 3.2 using formulation MIP1 with $\varepsilon = 0.0001$ to test uniqueness of the solutions. It turns out that for $k + 1 = 3$ the 10,000 retrieved tables are infeasible. There appears to be no other contingency table yielding the same CA solution. That is, in all 10,000 cases we retrieve the original contingency table.

For $k + 1 = 2$ we do find alternative solutions, but with an interesting property. Let λ_i^O (resp. λ_i^A) be the ith singular value of the original data (resp. alternative solution). Then for the alternative solution we have $\lambda_3^A = \lambda_2^O$. As an example of such a case, let the original data be

$$\mathbf{F} = \begin{bmatrix} 13 & 9 & 8 & 6 & 8 & 8 \\ 9 & 7 & 8 & 8 & 1 & 4 \\ 6 & 6 & 6 & 7 & 5 & 9 \\ 10 & 6 & 9 & 3 & 11 & 9 \\ 9 & 12 & 8 & 5 & 5 & 4 \\ 6 & 5 & 3 & 3 & 5 & 9 \end{bmatrix}$$

with singular values $(\lambda_1^O, \lambda_2^O, \lambda_3^O) = (1, 0.239471, 0.154923)$. After running the uniqueness test we find as a second solution the reconstructed data

$$\mathbf{F}_{rec} = \begin{bmatrix} 11 & 7 & 2 & 13 & 9 & 10 \\ 9 & 7 & 8 & 8 & 1 & 4 \\ 8 & 8 & 12 & 0 & 4 & 7 \\ 10 & 6 & 9 & 3 & 11 & 9 \\ 9 & 12 & 8 & 5 & 5 & 4 \\ 6 & 5 & 3 & 3 & 5 & 9 \end{bmatrix}.$$

with singular values $(\lambda_1^A, \lambda_2^A, \lambda_3^A) = (1, 0.307297, 0.239471)$. After finding such a solution, we should disregard it as an alternative solution, because the first $k + 1 = 3$ singular values do not correspond exactly with the original ones. In particular, the second singular value is different whereas the third singular value corresponds to the second singular value of the original original table. In conclusion, our tests show that we are able to retrieve the original contingency table for our testbed of randomly generated tables.

5. Empirical application

For a realistic validation of our method we use the CA results from Maiti et al. (2014) on the relationship between categories contributing to derailments in a steel plant. In their paper, two dimensional CA plots are provided together with tables with the corresponding coordinates rounded to three decimals. In addition, separate pie charts are provided that report the marginal frequencies for all variables. The contingency matrices are not reported in the paper.

Table 5
Published marginal frequencies and coordinates for CA Location versus Cause.

Location	Marginals	Dim 1	Dim 2
Empty/Cooling	22	0.660	0.123
Finished	73	0.351	0.078
Hot	73	−0.325	0.207
Maintenance	6	−0.416	0.375
Raw	174	−0.080	−0.148
Cause			
Jam	55	−0.336	−0.299
Manual	209	0.225	−0.003
Mechanical	38	−0.176	0.244
Track	46	−0.477	0.171

The published coordinate matrices are all in so-called principal coordinates. That is, for the row coordinates, the standardization using $\alpha = 1$ is used whereas the column coordinates are calculated using $\alpha = 0$. Such a solution is not uncommon in CA and it is sometimes referred to as the French plot (e.g., Carroll, Green, & Schaffer, 1989), a symmetrical plot (Greenacre, 2007) or a correspondence plot (Beh & Lombardo, 2014). Hence, in order to reconstruct the contingency table using Eq. (6), we must transform one of the two sets of coordinates. In addition, the published coordinates are standardized in such a way that, using our definitions for $\mathbf{D}_r$ and $\mathbf{D}_c$, $\mathbf{R}'_{k+1}\mathbf{D}_r\mathbf{R}_{k+1} = \mathbf{C}'_{k+1}\mathbf{D}_c\mathbf{C}_{k+1} = s\mathbf{\Lambda}^2$, where s is the number of observations. This difference in standardization follows from a slightly different formulation of CA where the table of relative frequencies $\mathbf{P} = \frac{1}{s}\mathbf{F}$ is used as the starting point. Note that, for our purposes, it is essential to use the CA formulation based on the contingency table as this allows us to impose the integrality constraints. Hence, we need to account for this difference by dividing the published coordinates by $\sqrt{s}$.

In Maiti et al. (2014), four categorical variables are involved in the study and the authors consider separate analyses for each combination of two variables. Hence, six separate CA analyses have been performed. The first variable concerns the shift in which a derailment occurred. For this variable only three categories are distinguished. Consequently, the three CA solutions corresponding to the contingency matrices involving this variable, can be described perfectly in two dimensions. Hence, the real data can be retrieved, up to rounding error, by simply inserting the coordinates into Eq. (6) where $\mathbf{U}_0\mathbf{G}\mathbf{V}'_0$ is zero.

For convenience, the marginal frequencies and the solutions for the other (three) correspondence analyses described in Maiti et al. (2014), are summarized in Tables 5 through 7. Note that the dimensionalities of the contingency matrices to be retrieved are 5×4, 5×7 and 4×7 respectively. The number of observations equals 384. In our simulation study, it proved to be hard to retrieve the original table based on coordinates rounded to three decimals. However, the tables here are slightly smaller whereas the number of observations is comparable.

We use model MIP1 with $\varepsilon = 0.0001$ to find the contingency matrices corresponding to the published results. For the row and column coordinates $\mathbf{R}_{k+1}$ and $\mathbf{C}_{k+1}$ we use the published coordinates divided by $\sqrt{s}$. The (non trivial) eigenvalues can then be calculated by taking the square root of the diagonal elements of $\frac{1}{2}(\mathbf{R}'_k\mathbf{D}_r\mathbf{R}_k + \mathbf{C}'_k\mathbf{D}_c\mathbf{C}_k) = \mathbf{\Lambda}^2$. We used this to transform the published column coordinates to so-called standard coordinates so that Eq. (6) can be used to retrieve the contingency table.

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

Table 6
Published marginal frequencies and coordinates for CA Location versus Department.

Location	Marginals	Dim 1	Dim 2
Empty/Cooling	22	0.287	0.271
Finished	73	−1.283	0.022
Hot	73	0.360	−0.761
Maintenance	6	−0.180	−0.991
Raw	174	0.357	0.310
Department			
Auxiliary	6	0.127	0.191
Logistics (IN)	168	0.351	0.201
Logistics (OUT)	45	−1.331	0.099
Maintenance	49	0.152	−0.527
Production (crude)	30	0.290	−1.074
Production (finished)	20	−1.477	0.033
Production (raw)	30	0.452	0.602

Table 7
Published marginal frequencies and coordinates for CA Cause versus Department.

Cause	Marginals	Dim 1	Dim 2
Jam	55	0.245	0.442
Manual	209	0.283	−0.053
Mechanical	38	−0.206	−0.466
Track	46	−1.410	0.099
Department			
Auxiliary	6	0.486	0.121
Logistics (IN)	30	0.230	0.198
Logistics (OUT)	20	0.249	−0.378
Maintenance	30	−1.405	−0.001
Production (crude)	49	0.171	0.003
Production (finished)	45	0.200	−0.159
Production (raw)	168	0.231	−0.463

Table 8
Retrieved contingency table Location versus Cause.

	Jam	Manual	Mechanical	Track
Empty/Cooling	0	20	2	0
Finished	5	56	7	5
Hot	11	33	11	18
Maintenance	1	2	2	1
Raw	38	98	16	22

Table 9
Retrieved contingency table Department versus Location.

	Empty/Cooling	Finished	Hot	Maintenance	Raw
Auxiliary	0	1	1	0	4
Logistics (IN)	17	12	31	0	108
Logistics (OUT)	2	34	2	0	7
Maintenance	1	6	17	5	20
Production (crude)	1	3	21	0	5
Production (finished)	0	16	0	1	3
Production (raw)	1	1	1	0	27

For all problems, a solution was obtained within a second. The value of the objective function for all problems was close to zero indicating small differences between the rounded solutions and the solution obtained using the retrieved contingency table. We verified that the CA solutions (rounded to three decimals) obtained from the retrieved contingency matrices were equal to the coordinates reported in the paper. The retrieved contingency matrices are given in Tables 8 through 10. As in the tests on the randomly generated tables, we conclude that we are able to find the original contingency tables from output resulting from a dimension reduction technique.

Table 10
Retrieved contingency table Department versus Cause.

	Jam	Manual	Mechanical	Track
Auxiliary	1	5	0	0
Logistics (IN)	40	106	11	11
Logistics (OUT)	1	36	6	2
Maintenance	3	9	8	29
Production (crude)	6	18	4	2
Production (finished)	0	18	0	2
Production (raw)	4	17	9	0

6. Conclusion

In this paper, we reconsidered the inverse correspondence analysis (CA) problem described in Groenen and Van de Velden (2004). By adding integrality constraints and formulating appropriate integer programming models we were able to improve considerably upon the original results.

Using a simulation study, we showed that even for tables of fairly large dimensionalities, it is possible to retrieve the original contingency table. This can be considered surprising as CA is a dimension reduction technique and hence one would expect that information is lost. Important factors concerning the success of our procedures is the dimensionality of the original contingency matrix (larger is harder) as well as the number of observations (more is harder) and the precision of the coordinates (more precision is easier; i.e., rounding the coordinates makes it harder to retrieve the original matrix).

To illustrate our method we applied it to CA results published in Maiti et al. (2014). In this publication, the results of several correspondence analyses were reported. The original tables, however, were not provided. We showed that based on the published marginals and CA coordinates, rounded to three decimals, it is possible to retrieve the contingency tables.

Retrieving a contingency table based on its CA solution is one application of our inverse approach. An alternative potential application of inverse CA, concerns the generation of contingency tables corresponding to an a-priori determined CA configuration. That is, rather than starting with a real CA solution, we start with a configuration and use inverse CA to find a contingency table that (approximately) yields the given configuration. This application of inverse CA could be of interest, for example, in simulation studies.

There are several directions for future research. Firstly, in this paper we assume that the marginals of the contingency table are known, which is not always the case. The absence of marginals complicates the inverse CA problem considerably as they then become decision variables instead of known parameters, resulting in a nonlinear optimization problem. Secondly, if on the other hand the marginals are known, bounds can be derived on the individual cell counts (see Duncan & Davis, and Dobra & Fienberg (2000)). If these bounds can be precomputed, we can add them as additional constraints to the MIP models to speed up computations. Thirdly, from our experiments and case study, it follows that the optimal solution (when found) always corresponds to the original contingency table (in case of a 3-dimensional CA approximation). It would be interesting to prove this formally. Finally, it would be interesting to extend our results in such a way that we can apply it more generally. An obvious extension concerns inverse multiple correspondence analysis, which is the multivariate extension of CA that allows the analysis of more than two categorical variables. However, although we tackled some relatively large problems in the simulation study, in multiple correspondence analysis the dimensionality of the original data matrix is typically much larger and our algorithms cannot be applied directly.

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014

References

Beh, E., & Lombardo, R. (2014). *Correspondence analysis: theory, practice and new strategies*. John Wiley & Sons. doi:10.1002/9781118762875.

Benzécri, J. P. (1973). *L'Analyse des Données. Tome 1: La Taxinomie. Tome 2: L'Analyse des correspondances. (2nd ed. 1976)*. Dunod Paris.

Carroll, J. D., Green, P. E., & Schaffer, C. M. (1989). Reply to Greenacre's commentary on the Carroll-Green-Schaffer scaling of two-way correspondence analysis solutions. *Journal of Marketing Research, 26*(3), 366–368.

Greenacre, M. J. (2007). *Correspondence analysis in practice*. CRC Press.

Groenen, P. J. F., & Van de Velden, M. (2004). Inverse correspondence analysis. *Linear Algebra and its Applications, 388*, 221–238.

Maiti, J., Singh, A. K., Mandal, S., & Verma, A. (2014). Mining safety rules for derailments in a steel plant using correspondence analysis. *Safety Science, 68*, 24–33. doi:10.1016/j.ssci.2014.02.011.

Moya, M. D., & Jain, R. (2013). When tourists are your "friends": Exploring the brand personality of Mexico and Brazil on Facebook. *Public Relations Review, 39*(1), 23–29. doi:10.1016/j.pubrev.2012.09.004.

Torres, A., & Bijmolt, T. H. (2009). Assessing brand image through communalities and asymmetries in brand-to-attribute and attribute-to-brand associations. *European Journal of Operational Research, 195*(2), 628–640. doi:10.1016/j.ejor.2008.02.020.

Van de Velden, M. (2000). *Topics in correspondence analysis*. Amsterdam: Tinbergen Institute, University of Amsterdam.

Van de Velden, M., & Kiers, H. A. L. (2005). Rotation in correspondence analysis. *Journal of Classification, 22*, 251–271.

Van de Velden, M., & Neudecker, H. (2000). On an eigenvalue property relevant in correspondence analysis and related methods.. *Linear Algebra and its Applications, 321*, 347–364.

Wolsey, L. A., & Nemhauser, G. L. (2014). *Integer and combinatorial optimization*. John Wiley & Sons.

Please cite this article as: M. van de Velden, W. van den Heuvel and H. Galy et al., Retrieving a contingency table from a correspondence analysis solution, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.11.014