Multiple correspondence analysis for “tall” data sets

Angelos Markos\textsuperscript{a,}\textsuperscript{*}, George Menexes\textsuperscript{b} and Theophilos Papadimitriou\textsuperscript{c}

\textsuperscript{a}Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
\textsuperscript{b}Lab of Agronomy, School of Agriculture, Aristotle University of Thessaloniki, Greece
\textsuperscript{c}Department of Int. Economic Relations and Development, Democritus University of Thrace, Komotini, Greece

Abstract. Correspondence Analysis (CA) is a statistical method aiming at the graphical representation of the contingencies between the rows and the columns of a categorical data set. A critical step of the CA algorithm is the Singular Value Decomposition (SVD) analysis of a coded matrix. The size of this matrix affects drastically the analysis computational cost. As the size of the matrix increases, the method becomes computationally expensive or even impossible. In this paper we propose an alternative scheme that overpasses this limitation, without affecting the results accuracy. A set of Monte Carlo simulations and real data applications showed the efficiency of the proposed approach over the standard one, especially in the case of “tall” data sets.

Keywords: Correspondence analysis, dimensionality reduction, tall data sets, singular value decomposition

1. Introduction

Correspondence Analysis (CA) is mainly considered as a non-linear multidimensional data analytic method, suitable for graphically exploring the association between two or more, non-metric, categorical variables without a priori hypotheses or assumptions. CA was originally used for the analysis and visualization of data arriving typically from fields of social sciences \cite{3,4,21} and biometrics \cite{8,22}. However, the method has recently attracted the attention of the engineering community from a number of disciplines, including machine learning and data mining \cite{13,15}. In image retrieval, CA has been typically performed for feature space reduction followed by a classification algorithm \cite{12,20}. In addition, the ability of CA to visualize many variables simultaneously was utilized to graphically explore the correlation between image features \cite{11,20}. In text mining, CA has been also applied to extract rules from large data sets and to identify meaningful associations between documents or words \cite{16,23}. In other cases, the method has been described as a preprocessing step for pattern recognition \cite{17,18}.

Although CA is oriented towards the analysis of two-way frequency contingency tables, it can also handle several types of data matrices of the form “objects \times variables”. The term “objects” refers to...
people, things, events, instances, or any entity in general for which there are recorded measurements. The “variables” correspond to the available measurements, which pertain to different attributes, features or characteristics for each object. In practice, CA can be performed to analyze almost any type of tabular data after suitable transformation or recoding [7]. The only input requirement of the method is a matrix with non-negative entries, with at least one non-zero entry to each row and each column.

There are various aspects of CA that are underestimated or even unknown. For example, CA can be described as a type of Principal Component Analysis (PCA) of categorical data [5,15]. Similar to PCA, the rows or columns of a data matrix are assumed to be points in a high-dimensional Euclidean space, and the method aims to highlight and graphically represent both visible and hidden relations in the data structure. This is achieved by mapping the original data onto lower-dimensional maps, so that the principal dimensions (usually two or three) capture the most variance possible. These dimensions can be considered as latent constructs or new composite quantitative variables, with metric properties, that summarize the original multi-dimensional information. In that sense, CA can be viewed as a method that quantifies qualitative data, with a simultaneous dimensionality reduction [5,7].

The fundamental mathematical result for CA and related dimension reduction techniques is the Singular Value Decomposition (SVD) of a suitably transformed matrix [7]. In the theory of CA, the SVD provides a straightforward mechanism of approximating a centered and normalized matrix with another matrix of lower rank by weighted least squares. All numerical results of the method are obtained directly from the SVD.

Technically, CA assigns a) numerical values (weights) to the variable categories (or levels) and b) scores to the objects. The object scores, estimated on each dimension through CA, satisfy the following optimization criteria: [5–7,14,19]:

– The maximum discrimination possible is achieved between the objects with regard to the latent construct expressed by each dimension.
– The reliability of dimensions, as regards their internal consistency, is maximized.
– On each dimension the average squared correlation between the object scores and the optimally quantified categorical variables is maximized.
– During the optimization process, the pairwise correlations between the variables are taken into account and the shape of each variable distribution (symmetry, outliers, e.a.) is taken into consideration.

In addition, it is important to outline that:

1. The sets of object scores on each dimension are pair-wise linearly independent.
2. The objects and the variable categories can be simultaneously projected onto the same lower dimensional space.

Consequently, CA can be considered as an optimal scaling technique over the aforementioned properties.

Even though CA was initially applied within a strict exploratory framework, nowadays, it can be used as a statistical method, since it has been enhanced with various inferential aspects [10,19].

Both the theoretical and practical aspects of the method have been covered in depth in a large collection of introductory and advanced textbooks (see for instance [2,5–7,10,19]).

The generalization of CA to the case of more than two variables has been studied under various names such as Multiple Correspondence Analysis, Homogeneity Analysis, Dual Scaling, and Principal Components Analysis for Categorical Data.

Multiple Correspondence Analysis (MCA) is usually applied to either an indicator matrix with dummy coding or to the corresponding generalized contingency table (Burt table). The indicator matrix is a
binary representation of the different categorical values of each variable. Each column in this matrix represents a category generated from the original data matrix during a discretization process, while each row represents an object. Therefore, the indicator matrix represents the total information contained in the original data. On the contrary, the Burt table contains information only for the initial variables, as it cross-tabulates each variable category against itself, and against the categories of all other variables. In other words, the application of MCA to the indicator matrix yields results for both the objects and the variables, while the application of the method on the Burt table yields results only for the variables. Hence, if further treatment of the objects is needed (i.e. a subsequent clustering or classification procedure), the standard MCA algorithm involves the analysis of the indicator matrix. At this point, it is important to note that there are cases where the original data set is very tall (objects $\gg$ variables) and the standard MCA of a large indicator matrix may become computationally expensive or even impossible to implement. Typical examples of tall data sets are census demographic, credit risk prediction and email spam filtering. In this paper, we describe an alternative scheme for the accurate and efficient computation of the MCA results, without involving the analysis of the indicator matrix.

The paper is organized as follows: Section 2 is devoted to the presentation of the classic CA algorithm in the bivariate and multivariate case. The proposed scheme is thoroughly described in Section 3. The efficiency of our approach is demonstrated in Section 4, through a set of experiments on both real and simulation data sets. The paper concludes in Section 5.

2. The correspondence analysis algorithm

In this section we will present in brief the Correspondence Analysis algorithm in the bivariate case. Then we will generalize the theory to the case of Multiple Correspondence Analysis.

2.1. Data coding

Let $D$ be the raw data matrix of $m$ objects and $p$ categorical variables. CA is implemented to the appropriate frequency matrices: a) in the bivariate case ($p = 2$), CA is applied to the variables cross-tabulation and b) in the case that ($p > 2$), the analysis is applied to the corresponding indicator matrix with dummy coding (0–1) or to the generalized contingency table, which is usually referred as Burt table. The relation between the original data set, the indicator and Burt table is depicted in Fig. 1.

The indicator matrix is a binary representation of the original data matrix. Both matrices have the same number of rows: one per object. Each variable is represented by a set of consequent columns (one per category). If we call $n_i$ the number of categories of the $i$-th variable with $i = 1, \ldots, p$, then the indicator matrix $Z$ would have $q = \sum_{i=1}^{p} n_i$ columns. The $Z$ matrix is created as follows: for the $i$-th variable we construct the sub-matrix $Z_i$ of size $m \times n_i$. The value 1 appears once in each row and indicates that this specific object is characterized by the corresponding category of this variable. The rest of the row is filled with 0’s. When the $Z_i$ sub-matrices ($i = 1, \ldots, p$) are horizontally juxtaposed, they form the complete indicator matrix $Z$.

The Burt table $B$ is a $q \times q$ frequency table, which cross-tabulates all the variable categories against each other. The matrix $B$ has a clearly defined structure: in the case of $p$ categorical variables, it consists
Fig. 1. Recoding of the original data matrix $D$ to the corresponding indicator matrix $Z$ and Burt table $B$ ($B = Z^T Z$).

of $p \times p$ blocks $B_{ij}$ with $i, j = 1 \ldots p$.

$$
B = \begin{bmatrix}
B_{11} & \cdots & B_{1i} & \cdots & B_{1p} \\
\vdots & & \vdots & & \vdots \\
B_{i1} & \cdots & B_{ii} & \cdots & B_{ip} \\
\vdots & & \vdots & & \vdots \\
B_{p1} & \cdots & B_{pi} & \cdots & B_{pp}
\end{bmatrix}
$$

The diagonal blocks ($B_{ii}$) cross-tabulate each variable with itself. Every off-diagonal element is zero, while every diagonal element is the frequency of appearance of a category. Conversely, every off-diagonal block ($B_{ij}$) cross-tabulates two different variables. It is trivial to verify that $B$ is square and symmetric. The Burt matrix $B$ is simply related to the indicator matrix $Z$ as follows:

$$
B = Z^T Z
$$

Let us suppose we have data on 5 objects (e.g. images), for which we have recorded qualitative measurements on 3 variables $X$, $Z$ and $W$, and that variable $X$ has 3 categories, while variables $Z$ and $W$ have two each. The $D$ matrix could then contain the following data:

$$
D = \begin{bmatrix}
2 & 1 & 1 \\
1 & 2 & 1 \\
1 & 2 & 2 \\
3 & 2 & 1 \\
2 & 1 & 2
\end{bmatrix}
$$

It should be noted that the numbers that appear as the elements of the $D$ matrix correspond to the different categories of the three variables and do not represent a measurement. In this case, it is obvious that $m = 5$, $p = 3$ and $q = 3 + 2 + 2 = 7$, while matrices $Z_1$, $Z_2$, $Z_3$, $Z$ and $B$ will be:
2.2. Algorithm analysis

Let us consider two $m$-sized integer vectors, each representing a categorical variable with different number of categories ($n_1$ and $n_2$ respectively), measured within the same sample or population. The information of both vectors is combined in the $n_1 \times n_2$ cross-tabulation frequency matrix $A$ and the relative frequency matrix $F_A$, which is obtained by dividing $A$ with its grand total $m$. The row and column sums of $F_A$ are called mass vectors or marginal relative frequencies and they are calculated as follows:

$$r(i) = \sum_{j=1}^{n_2} F(i, j) = \frac{1}{m} \sum_{j=1}^{n_2} A(i, j) \quad i = 1, \ldots, n_1$$

$$c(j) = \sum_{i=1}^{n_1} F(i, j) = \frac{1}{m} \sum_{i=1}^{n_1} A(i, j) \quad j = 1, \ldots, n_2$$

The so-called standardized residual matrix $R_A$ is a centered and normalized matrix calculated using the following equation:

$$R_A = \text{diag}(r)^{-1/2}(F_A - rc^T)\text{diag}(c)^{-1/2}$$

where $\text{diag}(a)$ is the diagonal matrix corresponding to vector $a$.

The fundamental step of CA corresponds to the SVD analysis of $R_A$

$$R_A = U_A \Sigma_A V_A^T$$

where $U_A$ and $V_A$ are the matrices of the left and right singular vectors, respectively, and $\Sigma_A$ is the diagonal matrix of (positive) singular values in descending order $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_k(A)$. The index $k$ denotes the rank of $R_A$. $U_A$ is a matrix which projects the rows of $A$ into a lower dimensional
space $\mathbb{R}^k$. The same relation is true for the projection matrix $V_A$ and the columns of $A$. Each singular vector in $U_A$ and $V_A$ corresponds to a basis vector of $\mathbb{R}^k$. The squares of the singular values of $R_A$ are denoted by $\lambda_1(A), \lambda_2(A), \ldots, \lambda_k(A)$; they are often called principal inertias in the CA framework.

There are two kinds of coordinates or scores that can be calculated based on the CA theory: the standard and the principal coordinates. The standard coordinates of the row and column categories of $A$, are given by:

$$S^{(r)}_A = \text{diag}(r)^{-1/2} U_A$$

and

$$S^{(c)}_A = \text{diag}(c)^{-1/2} V_A$$

Thereafter, the principal coordinates of the row and column categories are given by:

$$P^{(r)}_A = S^{(r)}_A \Sigma_A$$

and

$$P^{(c)}_A = S^{(c)}_A \Sigma_A$$

respectively. The main difference between the two coordinate systems is that the principal coordinates preserve the chi-square distance between the rows or columns of $F_A$, while the standard coordinates preserve the Mahalanobis distance [9].

There are two important properties connecting the row principal and the column principal coordinates often called the barycentric relationships of CA or transition formulas:

$$P^{(r)}_A = \text{diag}(r)^{-1} F_A S^{(c)}_A = \text{diag}(r)^{-1} F_A P^{(c)}_A \Sigma^{-1}_A$$

and

$$P^{(c)}_A = \text{diag}(c)^{-1} F^T_A S^{(r)}_A = \text{diag}(c)^{-1} F^T_A P^{(r)}_A \Sigma^{-1}_A$$

The CA graphical output is a set of factorial maps representing the columns’ and rows’ relative positions. Equations (10) and (11) allow the distance interpretation between the rows and/or the columns. There is a plethora of referential statistical tools for further treatment of the these maps (e.g. significance, quality and stability indices) (see [5–7,19]).

So far, we have shown the basic steps of the CA algorithm for the bivariate case. The same ideas can be generalized in the case of data sets with more than two variables: the basic calculation process is still valid. The only difference we should take into account is that the CA method is applied to either the $B$ matrix or the $Z$ matrix instead of $A$.

3. The proposed scheme

3.1. Algorithm analysis

Let $R_B$ and $R_Z$ be the standardized residual matrices of $B$ and $Z$, respectively (see Eq. (4)). The MCA algorithm computes either the SVD of $R_Z$ or the SVD of $R_B$. Note that, the analysis of $R_B$ yields
results only for the columns (variables) of the original data matrix. This is due to the loss of information that results from calculating \( B \) (see Fig. 1). Conversely, the analysis of \( R_Z \) yields results for both the rows and columns of the initial matrix \( D \). Unfortunately, the SVD of \( R_Z \) is often computationally intensive and difficult to implement on large data sets. Let us consider the case of a “tall” data set of 500,000 objects and 15 variables with 4 categories each. The corresponding \( R_Z \) and \( R_B \) matrices will be \( 500,000 \times 60 \) and \( 60 \times 60 \), respectively. It is trivial to verify that the computational cost of the SVD analysis of \( R_Z \) is much higher than the one of \( R_B \).

In this section we will describe a scheme that overcomes this drawback: we will present a set of equations that yields the output of the MCA of \( Z \) in a both accurate and efficient way. Our contribution is the calculation of both standard and principal coordinates of \( Z \), bypassing the computationally intensive SVD analysis of \( R_Z \).

The SVD analysis of \( R_Z \) yields:

\[
R_Z = U_Z \Sigma_Z V_Z^T
\]

Similarly, the SVD of \( R_B \) will give:

\[
R_B = U_B \Sigma_B V_B^T
\]

Equations (1), (12) and (13) lead to some interesting properties [6,7]:

Property 1. \( U_B = V_B \)

Property 2. \( \Sigma_B = \Sigma_Z^2 \)

Since \( B \) is a symmetric matrix, the row and column solutions are identical (Property 1). Property 2 indicates that the singular values in the Burt version are the squares of those in the indicator version. This is a direct result of the relationship between \( B \) and \( Z \) (see Eq. (1)).

Furthermore, the column standard coordinates are identical in the two versions of MCA [6]:

\[
S^{(c)}_Z = S^{(c)}_B
\]

Equation (9) in terms of \( Z \) will give:

\[
P^{(c)}_Z = S^{(c)}_Z \Sigma_Z
\]

Considering (14) and Property 2 we can now write:

\[
P^{(c)}_Z = S^{(c)}_B \Sigma^{1/2}_B
\]

Equations (14) and (16) give a direct calculation of the column standard and principal coordinates of the indicator matrix \( Z \). So far, the two alternative versions of MCA, applying CA to the indicator matrix or to the Burt matrix, are almost equivalent.

Recall that Eq. (10) connects the row principal coordinates with the column principal coordinates of a matrix. We rewrite Eq. (10) in terms of \( Z \) as follows:

\[
P^{(r)}_Z = \text{diag}(r_Z)^{-1} F_Z P^{(c)}_Z \Sigma^{-1}_Z
\]
In a similar manner to the classic CA algorithm described in Section 2, $F_Z$ denotes a relative frequency matrix, which is obtained by dividing $Z$ with its grand total. Since $mp$ is the grand total of $Z$ we have:

$$F_Z = \frac{1}{mp} Z$$

(18)

Note that, $Z$ has constant row sums (equal to the total number of variables $p$). Hence, the row mass vector of $Z$ is given by $r_Z = p/mp$ and,

$$\text{diag}(r_Z) = \frac{p}{mp} I$$

$$= \frac{1}{m} I$$

(19)

where $I$ is the identity matrix. If we apply (16), (18), (20) and Property 2 on (17) then we have the row principal coordinates of $Z$:

$$P_r(Z) = \text{diag}(r_Z)^{-1} F_Z P_c(Z)^{-1}$$

$$= \frac{m}{mp} Z P_c(Z)^{-1/2}$$

$$= \frac{1}{p} Z P_c(Z)^{-1/2}$$

$$= \frac{1}{p} Z S_c(B)$$

(20)

Thereafter, the row standard coordinates are given by:

$$S_r(Z) = P_r(Z)^{-1}$$

$$= \frac{1}{p} Z S_c(B)^{-1/2}$$

(21)

The proposed MCA algorithm could be summarized as follows:

**Step 1** - Calculate the matrix $R_B$ of standardized residuals

$$R_B = \text{diag}(r)^{-1/2}(F_B - r c^T)\text{diag}(r)^{-1/2}$$

**Step 2** - Calculate the SVD of $R_B$

$$R_B = U_B \Sigma_B V_B^T$$

**Step 3** - Standard coordinates of rows and columns of $Z$

$$S_r(Z) = p^{-1} Z S_c(B)^{-1/2}$$

$$S_c(Z) = \text{diag}(r)^{-1/2} U_B$$

**Step 4** - Principal coordinates of rows and columns of $Z$

$$P_r(Z) = p^{-1} Z S_c(B)$$

$$P_c(Z) = S_c(B)^{1/2}$$
3.2. Discussion

It is important to note the differences between the standard MCA of the indicator matrix $Z$ and the proposed approach. With the scheme presented above, the analysis of $Z$ is described in terms of the analysis of $B$. The crucial step of the standard MCA algorithm is the SVD of the $m \times q$ matrix $R_Z$, while in the proposed algorithm this step is replaced by the SVD analysis of the $q \times q$ matrix $R_B$ (see Step 2 above). The standard and principal coordinates are then obtained through a mathematically equivalent approach, based on the barycentric relations of CA. Consider that the numerical results are the same in the two approaches.

Let us consider the case of a “tall” categorical data set, when the number of objects is much higher than the number of variables ($m \gg q$). The standard MCA algorithm may become computationally inefficient, due to the large size of the SVD input matrix. In the following section, we will demonstrate that the effectiveness of the presented approach is almost independent of the data matrix row size. Finally, it is important to outline, that during the experiments, we have found extreme cases (very tall indicator matrices) where standard MCA crashed, while the proposed version gave accurate results.

4. Experiments

4.1. Simulation data

The efficiency of the proposed scheme was tested against the classic MCA method in a Monte Carlo framework. We tested both methods under two different scenarios: a) a stable number of columns and a various number of rows, and b) a stable number of rows and a various number of columns. The efficiency
was measured by the CPU runtime in seconds. We used a standard single 3.4 GHz processor computer with 2 GB of RAM to conduct the experiments running MATLAB®.

For the first set of tests we created a set of 20 variables. Each variable could have 2 to 8 categories, though their total number remained stable for the whole Monte Carlo procedure. The corresponding indicator matrix had 117 categories. We let the objects vary from 100,000 to 600,000. For each case we created 100 different matrices and performed the classic MCA and the proposed MCA. In Fig. 3 we present the mean processing time in each case.

The proposed version is always faster than the classic MCA. The CPU time gain is from 76.9% to 79.8%: almost stable. The reason we conducted only 100 Monte Carlo experiments for each case was that the standard deviation of the processing time was quite small (std(t) \(\leq 0.02 \times \text{mean}(t)\)). It is worthwhile to note that the modified MCA algorithm was able to treat the case of data sets with up to 700,000 rows, while the standard one crashed during the SVD analysis of the indicator matrix, given the memory limitations of the computer system on which these simulations were run and the particular SVD implementation we used.

In the second set of tests we let the number of variables vary from 4 to 30, and fixed the number of objects to 50,000. Each variable had exactly 4 categories. Both the classic and the proposed versions of MCA were performed in a Monte Carlo framework, using 100 matrices. Figure 4 shows the corresponding mean processing time.

The simulation results verify that the proposed algorithm always gives better results in terms of execution time than the standard algorithm. As the number of variables increases, so is the processing time gain (from 66% to almost 80%).

4.2. Application to real data

The purpose of this set of experiments was to test the efficiency of the proposed scheme against the standard MCA method, when applied on real data. Two real world data sets, obtained from the UCI
machine learning repository [1], were used. In both cases MCA could serve as a preprocessing step for dimensionality reduction, followed by a clustering or classification procedure.

For our first experiment, we consider a data set from a census bureau database containing 14 demographic characteristics of 48,842 adults. The final data set had 45,222 objects, 6 continuous and 8 discrete variables (objects with missing values were discarded from the analysis). The values of continuous variables were then categorized into three classes or intervals, with ranges determined on the basis of the values of the three quartiles $Q_{25}, Q_{50}, Q_{75}$. The corresponding indicator matrix had 45,222 rows and 133 columns. The proposed version is faster than the standard one. The absolute processing times are 5.51 and 1.32 seconds for the classic and the proposed MCA scheme, respectively. The CPU time gain is 76%.

For our second experiment, we utilize a data set from a geological survey involving 581,012 observations (30 × 30 meter ground cells) and 54 cartographic variables (10 quantitative, 4 binary wilderness areas and 40 binary soil type). The values of quantitative variables were grouped into three categories by quartiles. The corresponding indicator matrix had 581,012 rows and 74 columns. The proposed MCA scheme run in 6.2 seconds, while the classic version crashed during the SVD analysis of the corresponding indicator matrix.

5. Conclusions

In this paper we presented an alternative scheme to perform MCA. The application of the method often involves the SVD analysis of a matrix that can be very tall. Consequently, this fundamental step of the method is often computationally expensive and in some cases impossible to implement. It was demonstrated that the proposed scheme overpasses this limitation without affecting the calculation accuracy. The presented applications on both simulated and real data sets indicated important gain on the processing time of the proposed approach over the classic MCA version.
The alternative MCA scheme can be used as the preprocessing step in a two-stage sequential approach for clustering large data sets. The second step could involve a large-scale clustering of the object scores on the low dimensional space. As a final point, the modified algorithm could be adopted by the software packages which implement the MCA method. A MATLAB® and R implementation of both MCA versions is publicly available.2

Aknowledgements

We would like to thank the anonymous reviewers whose valuable comments have significantly improved the quality of this paper.

References


2http://www.amarkos.gr/research/tallmca.