Eurographic Conference on Visualization 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

*Volume 31 (2012), Number 3*

# Visual Analysis of Multidimensional Categorical Datasets

Bertjan Broeksema[1,2,3]

Alexandru C. Telea[2]

Thomas Baudel[1]

[1]IBM France Center for Advanced Studies
[2]Institute Johann Bernoulli, Univ. of Groningen, The Netherlands [3]INRIA, University of Bordeaux, France

**Abstract**

*We present a set of interactive techniques for the visual analysis of multidimensional categorical data. Our approach is based on Multiple Correspondence Analysis (MCA), which allows one to analyze relationships, patterns, trends and outliers among dependent categorical variables. We use MCA as a dimensionality reduction technique to project both observations and their attributes in the same 2D space. We use a treeview to show attributes and their domains, a histogram of their representativity in the dataset, and as a compact overview of attribute-related facts. A second view shows both attributes and observations. We use a Voronoi diagram whose cells can be interactively merged to discover salient attributes, cluster values, and bin categories. Barchart legends help assigning meaning to the 2D view axes and 2D point clusters. We illustrate our techniques with real-world application data.*

## 1. Introduction

Categorical dimensions are frequent in nowadays business data. Studying, analyzing and visualizing such data is critical for understanding the underlying business processes. Recent work on both multivariate and categorical data [ZSS09, AGMS11, TLLH12] acknowledges the importance for tools that support understanding these kinds of datasets.

Various interactive visualizations for multivariate data have been proposed, *e.g.*, permutation matrices [Ber77], table lenses [RC94], worlds within worlds [FB90], and parallel coordinates [Ins97]. In statistics, multivariate data is studied using Principal Component Analysis (PCA) [Cau29, Pea01, Hot33] and its extensions such as (Multiple) Correspondence Analysis (CA, MCA) [Hir35, BB76]. Such techniques share a common problem: Interpreting the results of an otherwise valuable analysis can be hard. This limits their adoption in business contexts where users quickly need to interpret results and may not have the time or knowledge needed to map abstract multivariate analysis results to their concrete problems.

We present a set of interactive visualization techniques for multivariate data targeted at analysts and business users. Our solution is built around MCA to focus on categorical data analysis. Such data is less covered by MDS or PCA techniques. Our goal is to expose the often complex correlations in categorical data, and answer questions such as: How do values of one attribute (or variable) relate to values of the same, or other, attributes? How to find clusters of similar observations? And how do such clusters relate to a certain value of an attribute? For this, we propose several linked views which blend existing and new visualization techniques. While the complexity of MCA analysis is introduced gently to end users, we still allow refining MCA results to extract additional insight. The main contributions of this paper are as follows:

- A space-filling visualization for the analysis of relationships between the inherent dimensions of categorical data;
- An interactive legend which helps explaining the meaning of dimensions extracted by MCA in terms of dataset attributes;
- An enhanced treeview which integrates raw-data information with the MCA analysis results;
- Interaction techniques that reduce the amount of information shown in the above views and help finding salient data point groups and inherent data dimensions.

In Section 2, we discuss related work. Section 3 presents MCA and its interpretation challenges. Section 4 presents our interactive views. Section 5 evaluates our method on an additional dataset and via a user study. Section 6 discusses our approach. Section 7 concludes the paper.

## 2. Related Work

High-dimensional data visualization involves limiting the number of data dimensions to a number that can be visually accommodated. For categorical data, two approaches exist [FJ11]: *CatVis* methods are specifically designed for categorical data and are more effective for frequency-related tasks. *Quantization* models represent categories by numerical values and is effective for similarity analysis tasks. Our work falls into this second type. Quantization methods reduce the number of dimensions in a meaningful way for visualization: One aims to find a projection of several $N$-dimensional data

points, or observations, in $K < N$ dimensions which keeps relationships (*e.g.*, distances, similarities, or correlations) between data points. The above $N$ dimensions are also called *attributes*. When $K \leq 3$, the projection can be directly shown using scatterplot or point cloud techniques. Several dimensionality reduction techniques exist [Fod02]. From these, we next focus on *multidimensional scaling* (MDS), *principal component analysis* (PCA) and *correspondence analysis* (CA).

MDS projects $N$-dimensional points in $K < N$ dimensions while trying to keep distance ratios between projected point pairs and original point pairs. Distances are typically computed by (weighted) Euclidean metrics. Although MDS has been successfully used in visual analytics [PNML08, BG05], several problems exist. First, while MDS helps finding point groups, it does not explicitly tell *what* the groups mean. To answer this, users resort to iterative brushing, color mapping, and other interaction tools. This requires a non-trivial effort and is hard when the projected $K$ dimensions consist of a *mix* of original $N$ dimensions, *i.e.*, when points are grouped due to similarities in more than one dimension. Secondly, MDS directly works only on numerical, not categorical, datasets.

PCA is one of the widest used multivariate statistical analysis techniques [AW10b]. PCA extracts salient information from a multivariate dataset into a new set of orthogonal attributes called principal components, eigenvectors, or *factors* $e_i$, sorted by variance. Data projections on eigenvectors are called eigenvalues or *factor scores*. Eigenvectors $e_i$ are computed so that they are orthogonal to more important eigenvectors $e_{j,j<i}$ and also capture the largest possible projected data variance. To help interpreting PCA, the *loading*, or correlation between a factor and an attribute can be computed. This estimates how much information a factor and an attribute share.

CA generalizes PCA by using the importances of all observations *and* attributes to discriminate between observations [Gre07]. CA computes two sets of factor scores, one for observations and one for attributes. Since both score sets share the same variance, they can be both shown in the *same* 2D scatterplot, which helps the reading of such plots [AW10b].

Multiple Correspondence Analysis (MCA) extends CA to handle *categorical* data [AV07]. Several MCA variants exist, all leading to the same equations as pointed out by [TY85]. MCA operates by first converting data from categorical to numerical form. Naively assigning a numerical value to each possible categorical value of an attribute can create artificial, arbitrary, distances between two values, which can cause misinterpretations. In contrast, MCA encodes each categorical attribute with a bitmask, one bit for each possible category value. For example, for the attribute $car \in \{Audi, BMW, VW\}$, the value *Audi* is encoded as $[100]$ and the value $VW$ as $[001]$. This effectively adds several new (binary) attributes to the original dataset. These binary attributes, stored in a so-called indicator matrix, are next processed with standard CA. In sociology, MCA has been promoted by Bourdieu [Bou79] to find hidden relationships between various sociological factors.

Many visualizations exist for categorical data, as reviewed by Friendly [Fri00a]. Fore-fold tables [Fri00b] show two-by-two tables. Mosaic plots and mosaic matrices show multi-way tables with tiles proportional to the frequency [Fri00b, Fri94, Fri99]. Parallel sets extend parallel coordinates by replacing individual data points by a frequency-based representation [KBH06]. The contingency wheel shows categories as sectors in a ring chart where sector sizes map the marginal frequency and rows that have a count for that frequency are drawn as nodes in the sectors [AGMS11]. CatTrees [KW01] extend treemaps [JS91] to show hierarchical categorical data. These techniques mainly address frequency related tasks. In contrast, we want to enable data correlation exploration and data classification at a granularity that suits the user. We also want to support exploration of individual observations, which excludes all techniques which are purely based on contingency tables. MCA (and PCA) results can be visualized using scatter plots. CA Maps [Gre07] map each category to a plot point. (CA) biplots [Gab71, Gre07] map both categories and observations to plot points. Our work next extends this bi-plot with interactive visualizations to reduce interpretation effort and help non-scientists answer the questions outlined in Sec. 1.

## 3. Multiple Correspondence Analysis

Figure 1 outlines our approach. We start with a table of categorical and/or numerical attributes. To use MCA on such data, we first bin numerical (ratio and interval) attributes to convert them to ordinal attributes. For example, an *Age* attribute can be binned to a five-class ordinal attribute $[0 :< 20,\ 1 : 20..30,\ 2 : 30..40,\ 3 : 40..50,\ 4 :\geq 50]$ (years). The number of bins, or categories, and the binning method (constant range or constant area in histogram) is configurable for all numeric attributes. The binning settings are application-specific. For details, we refer to [JJJ08] where an interactive technique is presented for quantification of numerical and categorical attributes.

From this refined table, we construct an indicator matrix (Sec. 3.1). MCA extracts correlation information from this matrix, which we use to create our visualization (Sec. 4). We next briefly overview the MCA technique, to form a basis for understanding our visualization, and to show the MCA interpretation problems that our visualization addresses next. For a thorough understanding of (M)CA, we refer to [AW10a, AV07] on which our implementation is based.

### 3.1. MCA Algorithm

For a table with $I$ tuples, each with $K$ attributes which in turn have $J_k$ levels or distinct values $\{v_k^1, \ldots, v_k^{J_k}\}$, $1 \leq k \leq K$, let $\mathbf{X}$ be the $I \times J$ indicator matrix, where $J = \sum_1^K J_k$. Applying CA on $\mathbf{X}$ gives a row factor score and a column factor score. These factor scores are the projections of observations (rows) and attribute values (columns) on the eigenvectors.

MCA starts by computing the probability matrix $\mathbf{Z} = N^{-1}\mathbf{X}$, where $N$ is the grand total of the matrix $\mathbf{X}$. Let $\mathbf{r}$ and $\mathbf{c}$ be the vectors containing the row, respectively column, totals of $\mathbf{Z}$. Let $\mathbf{D_c} = diag\{\mathbf{c}\}$ and $\mathbf{D_r} = diag\{\mathbf{r}\}$ be matrices with diagonals $\mathbf{c}$ and $\mathbf{r}$ respectively. We compute the factor scores by solving the Singular Value Decomposition (SVD) [Abd10]:

$$\mathbf{D_r^{-1/2}}\left(\mathbf{Z} - \mathbf{r}\mathbf{c}^\top\right)\mathbf{D_c^{-1/2}} = \mathbf{P}\Delta\mathbf{Q}^\top \tag{1}$$

with

$$\mathbf{P}^\top\mathbf{P} = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I} \tag{2}$$

Here, $\mathbf{A}$ is the left side of Eqn. 1; $\Delta$ is the diagonal matrix of eigenvalues of $\mathbf{A}\mathbf{A}^\top$; $\mathbf{P}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$; and $\mathbf{Q}$ are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$. From the SVD we compute the row factor scores $\mathbf{F} = \mathbf{D_r^{-1/2}}\mathbf{P}\Delta$ and column factor scores $\mathbf{G} = \mathbf{D_c^{-1/2}}\mathbf{Q}\Delta$ by projecting attributes on the respective eigenvectors.

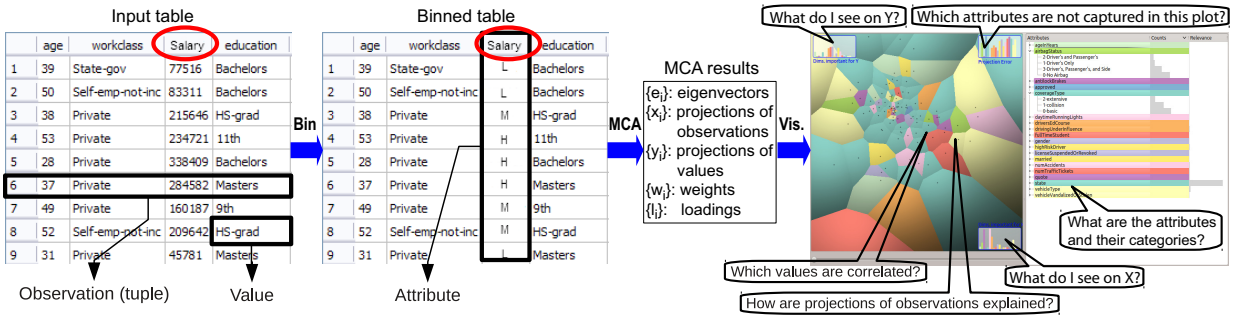


**Figure 1:** *MCA visualization pipeline. Input: multidimensional table with numerical and categorical data. Numerical columns (*e.g.*, salary in three levels: L, M and H) are binned. MCA is done on the binned table. MCA results are used for visualization.*

Visualizing the MCA results now follows the classical scatterplot technique used for MDS: We take the two factors $e_x$ and $e_y$ along which the data has most variance, and plot all data point projections, *i.e.*, factor scores, along $e_x$ and $e_y$. In contrast to MDS, we can also draw the *attributes* in the same plot: These are simply the projections of the $J$ points having one for a particular attribute value and zero for all others.

### 3.2. MCA Interpretation Challenges

As outlined above, MCA creates a plot containing both observations and attributes. Interpreting this plot is based on proximity of points of the *same* kind: Two *observations* plotted close to each other imply that they have similar attribute values or, for categorical data, that they *share* several attribute values (since two categorical values can either be equal or different). *Attributes* plotted close to each other are interpreted differently in CA and MCA. In CA, columns are actual different attributes in the input data. Hence, when two attribute points are close, observations *tend to be similar* with respect to these attributes. In MCA, columns can be either (categorical) values of the same attribute or values of two different attributes from the original data, given the bit structure of the indicator matrix $\mathbf{X}$ (Secs. 2, 3.1). Close plotted points *for values from different attributes* imply that observations tend to select these values together. Close plotted points *for different values of the same attribute* imply that observations select either of these values and are similar with respect to the other attributes.

Although the mathematics of MCA is relatively straightforward, interpreting MCA plots is clearly not. This is firstly due to the abstract nature of the computed quantities, which do not directly map to the user's world (observations and attributes). Secondly, for many observations and/or attributes, 2D scatterplots of observation and attribute factor scores get cluttered. Thirdly, on a technical level, data outliers can influence the factors (eigenvectors): The 2D plot space gives too much space to outliers and too little space to 'interesting' observations.

Without adequate tooling, potential insights delivered by MCA risk being lost. Hence, we want to provide intuitive interactive visualizations of MCA analysis results, to address the following questions:

- How to link the MCA results (factors, factor scores) to the meaning of the original data (observations and attributes)?
- How to show the meaning of the projected dimensions?
- How to explain the grouping of projected observations?
- How to eliminate irrelevant (outlier) dimensions or outlier values of a dimension?
- How to get an overview of values that occur together?

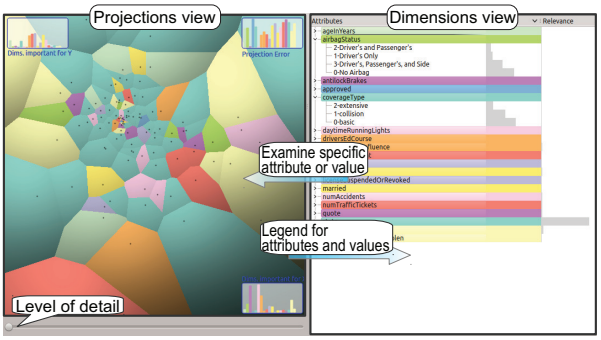


**Figure 2:** *MCA visualization overview.*

## 4. Visualization Overview

To address the above goals, we propose a visualization with two main views: the *dimensions view* (Sec. 4.1) and the *projections view* (Sec. 4.2). These support the steps of observation classification and observation exploration: First, one wants to *classify* data to a granularity level suitable for the task at hand. For example, in a car insurance dataset, finding that students, expensive cars, and many accidents are strongly correlated, leads one to classify such observations as "students causing accidents in expensive cars". Once users have a clear picture of the classes occurring in the data, they next explore the observations to give sense to clusters and understand outliers.

The *dimensions view* shows the attributes and their domains. It serves both as an analysis entry point and as a legend for the more complex projections view. The *projections* view shows the factors computed by MCA; it targets questions related to correlations and variances of observations and attributes. The two views are linked via shared colormaps and selection, to support asking questions in one view and using the other view to understand the results (Fig. 2). As an example, we next use a set of 5000 US car insurance quotations, with 19 attributes per observation. Table 1 shows these attributes, and how numerical attributes have been reduced to categories by binning.

### 4.1. Dimensions View

The dimensions view (Fig. 3) shows the attributes present in the raw dataset which is the input of the MCA analysis.

Recall that a $K$-dimensional dataset yields an indicator matrix with $J$ binary attributes, where each binary attribute shows whether an observation selects a given value (Secs. 2, 3.1). We

| Dimension or attribute | Type | Bins | Binning |
|---|---|---|---|
| ageInYears (age) | integer | 4 | $< 35, 35..53, 54..72, > 72$ |
| airbagStatus | category | 4 | none, driver only, front seats, all |
| antilockBrakes | boolean | 2 | true, false |
| approved | boolean | 2 | true, false |
| coverageType | category | 3 | basic, collision, extensive |
| daylightRunningLights (lights) | boolean | 2 | true, false |
| driversEdCourse | boolean | 2 | true, false |
| drivingUnderInfluence | boolean | 2 | true, false |
| fulltimeStudent (student) | boolean | 2 | true, false |
| gender | category | 2 | male, female |
| highRiskDriver | boolean | 2 | true, false |
| licenseSuspendedOrRevoked | boolean | 2 | true, false |
| married | boolean | 2 | true, false |
| numAccidents | integer | 4 | $0, 1, 2, >= 3$ |
| numTrafficTickets | integer | 4 | $0, 1, 2, >= 3$ |
| quote | USD | 4 | $< 310, 310..619, 619..1043, > 1043$ |
| state | category | 50 | AL, AK, AZ, ..., WY |
| vehicleType (vehicle) | category | 9 | compact, sedan, luxury, sport, pickup SUV, sport-luxury, collection, van |
| vehicleVandalizedOrStolen | boolean | 2 | true, false |

**Table 1:** *Data types for the US car insurance dataset. Names between brackets are the labels used in images.*

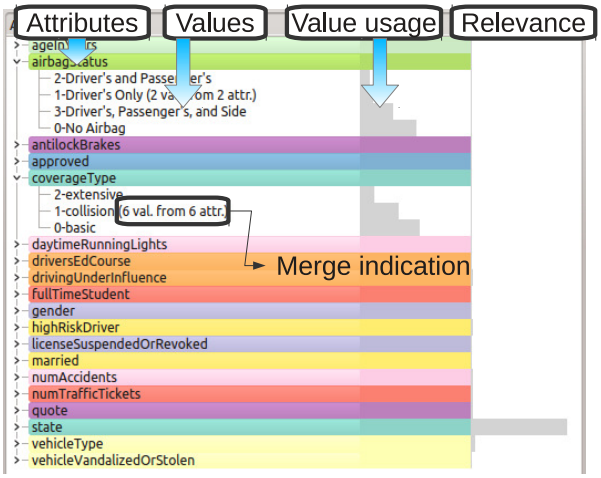


**Figure 3:** *Dimension view for the insurance dataset*

show the raw data using a two-level tree: attributes and values. The first level shows all original attributes. On the second level, each attribute has $J_k$ children, *i.e.*, all its categorical values $\{v_k^i\}$. Attribute nodes are colored as follows. First, we sort attributes based on decreasing relevance, and assign them colors cyclically from a fixed categorical colormap with $C = 10$ hues [BH11]. Next, we set the nodes' color saturations to their attributes' relevance. Given the attribute sorting, even if two nodes have the same hue (for datasets with more than $C$ attributes), their colors will differ in saturation: Most important attributes are bright, and less important ones are dull. We stress that color mapping is not a main contribution of our work: If available, better techniques should be used. Attribute nodes are labeled by their dimension names. Value nodes are labeled by a textual description, see, *e.g.*, the *ageInYears* integer attribute (Fig. 3 top) which is binned in 4 values ($< 35$, 35..53, 54..72, $> 72$ years). Value nodes show three additional properties:

- The percentage of observations with that value, as a bar. This shows which values occur most in the dataset. We later refine this insight to find if such values are indeed discriminative for the correlation of observations or not (Sec. 4.2).
- The attribute value weights $w_j$, or relevances, computed as in [AW10a]. Large weights show attribute values which are important for discriminating between observations.
- Value and attribute merging (see next Sec. 4.3).

Sorting the dimensions view on the value usage column shows the distribution of values for a particular attribute. To find attributes and values which discriminate between observations, the view can be sorted on the relevance column. This relates to value column: Values which are rarely used by the observations may provide more information for discriminating between observations and are therefore more relevant; frequently used values are less interesting [AW10a].

The dimensions view serves as a *legend* for the more complex projections view, which we present next.

### 4.2. Projections View

This view displays projections of both observations and attributes computed by MCA. It helps finding correlations and variances in the input data, *i.e.*, answer questions such as which attributes contribute to a given factor; along which attributes are certain observations most (or least) similar; and what is the meaning of a factor. We use the classical MDS approach: We draw a scatterplot by projecting all observations $\mathbf{x}_i$ and attribute values $v_k^i$ (Sec. 4.1) on the two most important factors computed by MCA (Sec. 3.1). We next add several visual enhancements to this plot, as follows.

Recall that close projections of values of the *same* attribute mean that observations selecting any of these values are similar *vs* their other attributes (Sec. 3.2). Close projections of values of *different* attributes imply that observations tend to have these values for the respective attributes together. In both cases, we want to find (a) the relative distances between projected values and (b) how these values are grouped within categories.

We support this task by drawing a Voronoi partitioning of the 2D plot space, with the projected values as sites. Cell colors show their categorical attribute, as in the dimensions view (Fig. 3). To separate small cells of similar colors, we use parabolic shaded cushions, akin to [TvW01]. Finally, we label cells with their categorical values. Labels are centered and clipped to fit in the inscribed circle in each cell. Tooltips with the full labels are shown when brushing over the cells. Additionally, brushing links the projection and dimensions views.

Fig. 4 shows the projection view for the car insurance dataset. As the *state* attribute (light blue) has many values (50) relative to other attributes, we see many such cells. In the center we see a cluster of small non-state cells (different hues than light blue) surrounded by state cells (light blue). Among these non-state cells are *quote:* $< 310$, *quote:* 310..619, *vehicle type: sport*, *vehicle type: van*, and *#accidents:* $< 1$. Since these cells are small, their projected values are close, so we infer that many observations select these values together. Further, we infer that customers from states surrounding these cells, *e.g.*, *CA*, *FL*, *NJ*, tend to have such a profile, while customers from states that are at the periphery, *e.g.*, *SD*, *AK*, *NV*, have different profiles. Note that the distance metric is important here: the exact locations of the Voronoi cell borders *vs* the observation projections is not decisive; the distance from the attribute projections to the observation projection is.

In Fig. 4 right, we see cells for the values *(daytime running) lights: true*, *airbag status: all seats*, and *vehicle type: SUV, luxury, sport-luxury*. On the left, we see *lights: false*, *airbag status: none*, and *vehicle type: collection, compact, pickup*. This shows that the X axis of this view maps the car class (left = cheap cars with few options, right = expensive cars with many options). This pattern could be related to wealth of the insured

persons. Since wealth is not an attribute in our data, this is an interesting finding.

At the top of Fig. 4 we find cells for the age categories 35..53 and 54..72 and married people. At the bottom, we find people below 35 and who are full-time students. Hence, the Y axis maps the phase of life people are in. Finally, outlier cells, such as students (Fig. 4 bottom-left), show that observations that select this value, *i.e* full-time students, share less values with the other observations as compared to observations that select values in the central cells.

In brief, the attribute plot can be interpreted as follows:

- values in central cells are used by the average person type;
- values in periphery cells are used by outlier persons;
- the Y axis reflects life phase, with senior people at top and young people at the bottom;
- the X axis reflects car prices, with more expensive cars at right and cheaper ones at left.

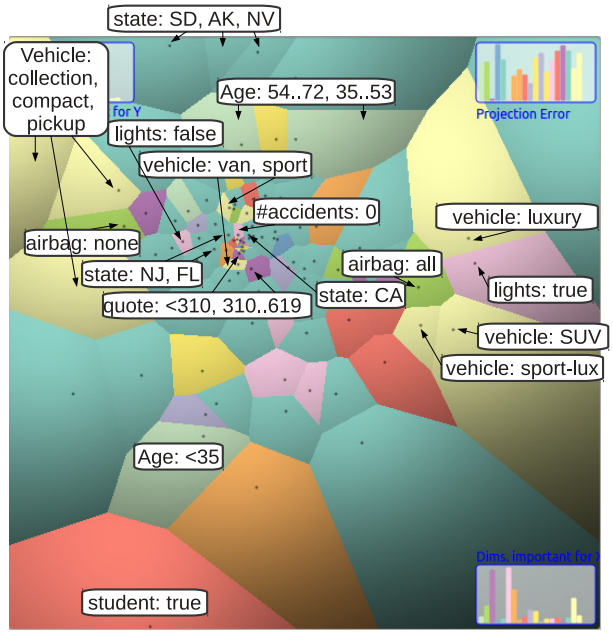


**Figure 4:** *Projections view with attribute values. Labels and arrows are added here manually for illustration purposes.*

### 4.3. Finding meaningful clusters by value cell merging

The projection view (Fig. 4) can easily get crowded, since it shows as many cells as there are different attribute *values* in the input dataset (104, in our case). One task we want to address is classify data in clusters at a level that is meaningful for the analysis goals at hand. To support this, we provide four ways to cluster and filter data:

- Leave out attributes from the analysis;
- Cluster attributes in the attributes view based on distance;
- Cluster values of one user-selected attribute (Sec. 4.4);
- Filter observations based on attribute value (Sec. 4.4).

An attribute can be left out when it is of no importance for the analysis. This creates more space for the remaining values. For instance, when we remove the *state* attribute from Fig. 4, there are 50 cells less in this view. However, this may result in information loss, so users should decide which attributes are relevant for each analysis on a case-by-case basis.

As explained, values who project closely show that observations are very similar with respect to these values. Hence, we are not interested to examine such values separately – instead, we want to find *clusters* of values at various levels of detail, which show us the properties that define a homogeneous subset of our observations.

To find such clusters, we add a level-of-detail option, controlled by the slider shown under the projections view (Fig. 2). The slider maps a distance $\delta$ in 2D (projection) space. When the user changes $\delta$, we iteratively merge pairs of value-projections which are closer than $\delta$ into a new cell whose barycenter is the average of the merged cells' barycenters. Fig. 5 shows the effect of merging: Most small cells at the center of Fig. 4 have now been merged, by grouping attribute values which are selected by the average person. The merged cells can now represent (a) either values of the *same* categorical attribute, or (b) values of *different* categorical attributes. Outlier cells, however, stay roughly unchanged. Hence, we use less cells to show the concept of average person, but keep the cells that show outlier persons.

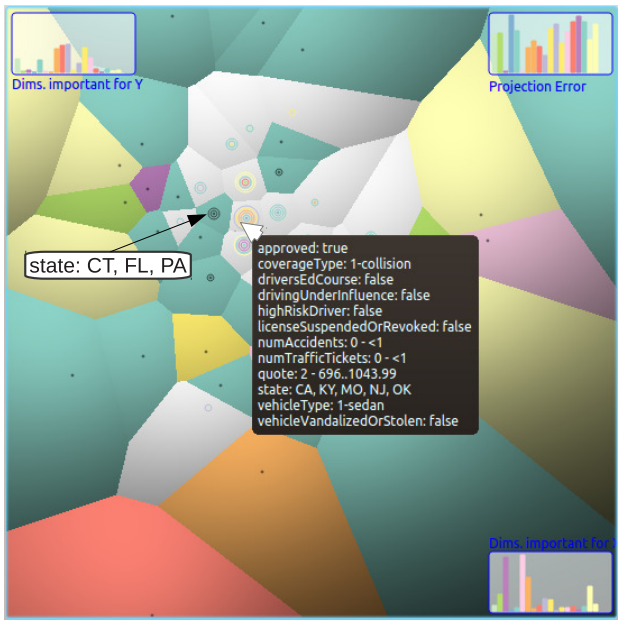


**Figure 5:** *Projection view with merged value cells. White label added manually for illustration purposes.*

To show which values get merged, we draw a set of concentric rings around the merged cells' sites. The number of rings equals the number of merged values within a cell. Cells containing only values of the same attribute are colored using that attribute's hue, as before, and the rings are colored black. For example, in Fig. 5 we see a cell grouping all states *CT*, *FL* and *PA*. Cells containing values of different attributes are colored in light-gray, a reserved color not used in the attribute colormap, to show that they groups different attributes. Rings in such cells are colored by the colors of the merged attributes. We read this visual encoding as follows:

- cells with many rings contain many attribute values;
- non-light gray cells with many rings contain merged values of the same attribute; the cell's color shows the attribute;
- light gray cells contain merged values of different attributes; the rings' colors show the attributes;
- cells with no rings encode individual, non-merged, values.

In Fig. 5, the light gray cell (under the mouse) contains many rings, *i.e.*, many merged values. The rings' colors show that this cell groups values from the attributes *coverageType*, *drivingUnderInfluence*, and *vehicleType*. The tooltip shows details on demand, *i.e.*, the merged values: *coverageType: collision*, *drivingUnderInfluence: false*, and *vehicleType: sedan*. From this data, we infer that this cell groups people who drive safely (no accidents, no traffic tickets, not caught for driving under influence) but who still request a collision-coverage insurance, which is an interesting finding. We also show merging information in the dimensions view. Fig. 3 shows how (at a different merging level) *collision* is merged with 6 values from 6 attributes. When clicking on a cell in the projections view, values merged in this cell get highlighted in the dimensions view.

### 4.4. Value filtering and merging

Merging value cells reduces the cell count while keeping the information encoded by the merged cells in the view. However, the user controls this process only globally, via the projection distance. For finer-grained ways to reduce the cell count, we provide a filter/merge view (Fig. 6). Filtering and merging follows three simple steps: (1) select an attribute; (2) select one or more values thereof; (3) perform filtering or merging.

When an attribute $v_k$ is selected by clicking on its cell in the projection view or tree item in the dimensions view, the filter/merge view shows all values $v_k^i$ of $v$ along with their cell sizes in the projections view. Sorting this list lets one pick the largest cells, which typically appear at the periphery of the Voronoi diagram, and thus take considerable space that could be used to show more detail in the crowded areas. After the desired attribute values are selected, one can filter or merge the data based on this selection.

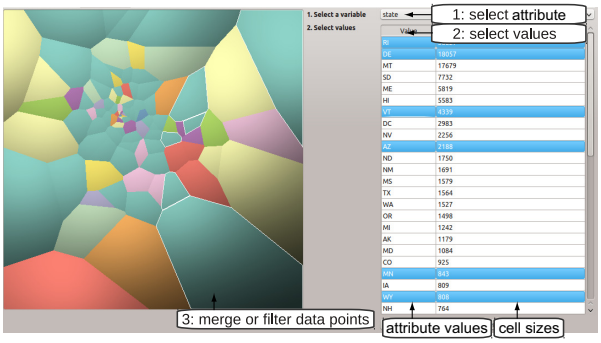


**Figure 6:** *The merge/filter view.*

*Filtering* removes observations which have any of the selected attribute values. For example, to get more insight in student characteristics, we select the *fulltimeStudent: false* cell, filter, and thus remove all non-students from the view. After filtering, MCA is recomputed automatically on the filtered data. This updates the views with a new projection with removed outliers, thus more space for the interesting observations. In our example, our analysis will now only concern students.

*Merging* simplifies the visualization by replacing several values $v_k^i$ of an user-selected attribute $k$ with *one* new value $v_k^{new}$. Like for filtering, MCA is done anew after merging and all views are updated. Unlike filtering, merging $n$ attribute values will remove exactly $n-1$ cells from the projections view, since there is exactly one cell per attribute value. For example, consider the *states* attribute, which has 50 values. Recalling the analysis in Sec. 4.2, we have found cells on the right of Fig. 4 as high-income-related, and cells to the left as low-income-related. If we accept this meaning, we can now merge states on the right of Fig. 4 to a new value *high-income states*, states to the left into a new value *low-income states*, and the remaining (center cells) states *moderate-income states*. Fig. 7 shows the updated view. The view has a similar layout as before merging (modulo a rotation which is an unfortunate side-effect of MCA), but offers now more space to other values than states, since we now have 3 state values instead of 50.

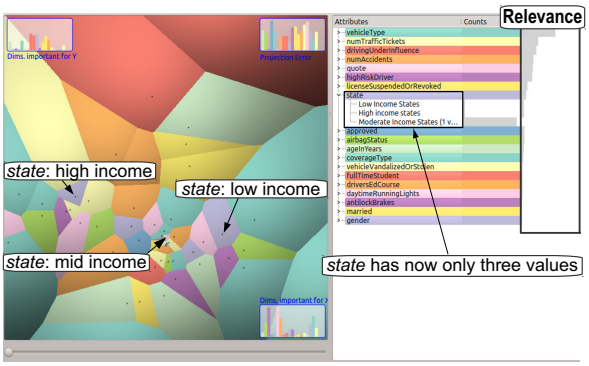


**Figure 7:** *Merging states into three different groups.*

The relevance metric for attribute values, shown in the dimensions view (Fig. 7 right), serves here two purposes. First, we can use it to select which attribute values we want to filter or merge – the less relevant ones. Secondly, this metric tells us how attributes change their relevance (for distinguishing between observations) after a filter or merge was applied. This helps iteratively reducing the dimensionality of the dataset by incrementally merging less relevant values into higher-level concepts, and also helps users focus on the most relevant concepts at a given level of detail.

### 4.5. Projection legends

Dimensionality reduction techniques like MDS or MCA typically project the data along $K \in \{2,3\}$ eigenvectors, and draw projections as $K$-D scatterplots. However, such plots can be hard to read by many business users. One issue is that *axes* have no explicit meaning: These are factors, coming from the SVD in the MCA case. Ideally, we would like to explain the axes in terms of the variance of attributes and attribute values.

For this, we proceed as follows. Given an observation $\mathbf{x}_i = (x_i^1, \ldots x_i^J)$, *i.e.* a row of $\mathbf{X}$, the quantities

$$b_{i,j} = \frac{m_i f_{i,j}^2}{\sum_{k=1}^{I} m_k f_{k,j}^2} \quad \text{where } 1 \le i \le I,\ 1 \le j \le L \qquad (3)$$

give the so-called *contributions* of $\mathbf{x}_i$ to the $j^{th}$ factor, *i.e.*, how important is $\mathbf{x}_i$ for factor $j$. Here, $f_{i,j}$ are the elements of the row factor score matrix $\mathbf{F}$ and $L$ is the number of non-zero singular values, or number of columns of $\mathbf{F}$ (Sec. 3.1), and $m_i = \sum_{k=1}^{J} x_i^k$ is the so-called mass of row $i$ of $\mathbf{X}$. We are, however, interested in the contributions of *attributes*, rather than observations, to the factors. This is easy to compute: We extend the input matrix $\mathbf{X}$ by $J$ supplementary rows $r_i^{sup}$, one row for each attribute value $v_i^j$, containing zeros for all columns except the attribute value's own column, which contains a one. Next, we project these supplementary rows using the SVD already computed by MCA from the observations, and obtain one supplementary factor score row $f_i$ for each $r_i^{sup}$. The values $f_i$, also

called *loading* in the literature, are used to compute the contributions via Eqn. 3. For details, we refer to [AW10a, AV07].

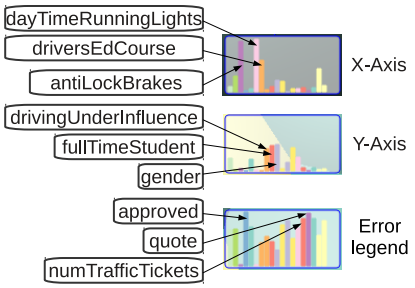


**Figure 8:** *Zoomed-in projection legends from Fig. 4 with attribute contributions.*

We now have two contribution vectors $\mathbf{b}_x = \{b_x^i\}$ and $\mathbf{b}_y = \{b_y^i\}$ for the two factors used to draw our 2D scatterplot. The values $b_x^i$ and $b_y^i$ give the contributions of all values of attribute $i$ attribute to the $x$ and $y$ plot axes. We can now *explain* what the $x$ and $y$ axes mean in terms of a mix of attribute values from the input data. Still, $\mathbf{b}_x$ and $\mathbf{b}_y$ are not in the optimal form for interpretation: Since MCA uses one column for each attribute *value*, our vectors $\mathbf{b}_x$ and $\mathbf{b}_y$ have $J$ elements, one for each attribute value. We simplify the contribution vectors by summing up all values that correspond to the same attribute. The resulting contribution vectors $\tilde{\mathbf{b}}_x$ and $\tilde{\mathbf{b}}_y$ have now $K$ elements, *i.e.* as many as the number of input attributes. Their elements indicate the contribution of each separate attribute (and not attribute value) to the plot axes.

We show these values by barchart legends on the $x$ and $y$ plot axes. This approach is somewhat similar to [ODH*07]. However, we only show the plots for the projected factors and add interaction to the barcharts, as explained next in Sec. 4.6. Each bar is colored by the hue of its corresponding attribute, as in the dimension view (Sec. 4.1) and projections view (Sec. 4.2).

Fig. 8 shows a zoom-in of the projection legends for the insurance dataset in Fig. 4. The $x$ legend has two large bars for the attributes *antilockBrakes* and *daytimeRunningLights*. If we brush the view, we see indeed that these attributes have extreme values at the left and right of the $x$ axis respectively – see *e.g.*, the cell *daytimeRunningLights: true* right in Fig. 4.

MCA shows the input data projected along the two most relevant factors. However, datasets may be inherently of higher dimensions than two [PNML08]. Hence, an MCA (or similar) 2D projection may convey false insights if much of the data variance occurs along the 'discarded' dimensions. We show this by a third barchart: the *error legend* (Fig. 8). This barchart is built similarly to our previous ones, but it shows the sum of contributions of all factors except the two used for the actual projection. We read this chart as follows: Short bars show attributes whose variance is well captured in the 2D projection. Long bars show attributes whose variance is captured mainly by factors *not* used in this projection. Seeing such large bars, users can (a) either continue the analysis, but refrain from making judgments about these attributes; or (b) select one of the $x$ or $y$ dimensions in the current view to use the factor that has the largest variance for the attribute of interest. This can be done by shift-clicking on the respective attribute bar.

### 4.6. Observation plot

As explained in Sec. 4.2, both observations and attribute values are projected in our 2D plot space. So far, we showed how attribute values are visualized.

Fig. 9 shows the projections view used to explore observations, a view we call the *observation plot*. Typically, one has many more observations than attribute values. To remove clutter, and show observation density, we draw observations using additive alpha blending. Attribute cells are shown in the background, but grayed out, so we can use colors to show the observations' attribute values. The relation between attribute cells and observations is as follows: If an observation $\mathbf{x}_i$ is closer to an attribute value $j$ than to other attribute values $k \neq j$, then $\mathbf{x}_i$ will more likely select value $j$ than select values $k$ *relative* to the other observations. Showing the attribute cells helps assessing such relations without having to visually locate attribute value projections, which is hard given the dense observation plot. To find more information about a cell close to observations of interest, we can switch the view to the attribute plot. The cell layout stays the same, so users keep their mental map.

Observations tend to form clusters (groups of closely packed points) in the observation plot, based on similarity. A standard analysis task is to explain such clusters. We assist this by adding functionality to the barchart legends (Sec. 4.5): When clicking a bar in the $x$, $y$ or error barcharts, observations are colored using a categorical colormap on the values of the bar's attribute. This colormap is different from the hue mapping used in the dimensions and projections views (Secs. 4.1, 4.2), and has a different purpose: The hue map shows the *identity* of an attribute, *i.e.*, links the projections view with the first tree-level in the dimensions view. The value colormap shows the different *values* of an attribute, *i.e.*, links the observations plot with the second tree-level of the dimensions view.

Fig. 9 a shows an example. Two separate clusters are apparent. Both spread along both $x$ and $y$ axes, *i.e.*, along the two factors used to create this projection. Hence, if these clusters are determined by some attribute, this attribute contributes to *both* the $x$ and $y$ factors, otherwise the clusters would be one-dimensional (lines). We use this hint and the barcharts to explain the clusters, as follows. First, the clusters cannot be explained by the two long bars in the $x$ barchart, *antilock brakes* and *daytime running lights*, since these attributes contribute almost fully to the $x$ axis, as shown by their long bars which reach almost 1. The next two longest bars in the $x$ barchart, $A$ and $B$, are about half height, so they contribute only 50% to the $x$ factor. However, they have no contributions to the $y$ factor (very short bars $A'$ and $B'$ in the $y$ barchart), so they cannot explain the spread along $y$. In the $y$ barchart, we see three attributes that contribute almost equally to this axis. The longest one, *gender*, does not explain the clusters, since it is short in the $x$ barchart. We have now two remaining possibilities. Clicking the second-longest bar in the $y$ barchart (*fulltime student*) colors observations based on this attribute, *i.e.*, students=blue and non-students=red (Fig.9 b). The colors match the perceived clusters, so we conclude that the clusters reflect the student status.

In Fig. 9 b, we see that students are mostly present in the lower left area of the plot. If we read the attribute labels for the cells in this area (Fig. 8), we find that students have a

- lower probability of being married;
- higher probability of being under 35;

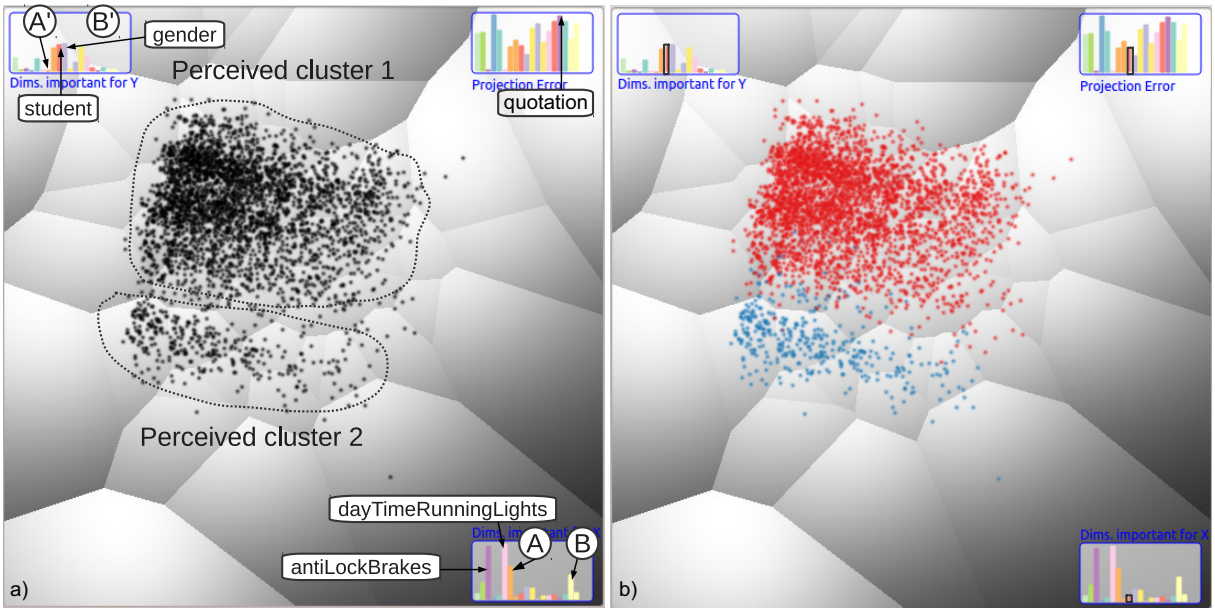


**Figure 9:** *Observation plot: (a) without selection; (b) with 'full-time student' attribute selected (blue = student, red = non-student). Legends help confirming that the clusters reflect the student status attribute.*

- higher probability of being caught driving under influence;
- higher probability of having there license suspended;
- higher probability of causing accidents.

Such findings are evidence of an increased risk for accidents under students. Analysts could use this to adjust insurance quotations. Let us see if this was the case in our data. Looking at the error barchart in Fig. 9, we see that the insurance *quotation* is high, *i.e.*, it contributes very little to the $x$ and $y$ factors, and a lot to the other factors not used in this projection. If quotation and student status were correlated, the quotation attribute should have contributed visibly to the $y$ axis which, as we saw, explains the student status attribute. As this does not happen, it means the quotation is not correlated with student status, even though student status is correlated with accident risk.

## 5. Evaluation

For further evaluation, we analyzed a second example: the *adult* dataset from [FA10]. The data have 15 attributes related to education in the US, including *education level*, *educations*, work *hours/week*, and *classification* (earning below or over 50K USD). After applying cell merging (Sec. 4.3) to find coarse patterns, the attribute plot shows a shape running from left to right and then curving upwards (Fig. 10 a). The $x$ barchart shows that *classification* explains the $x$ axis best: The >50K attribute cell is on the left and the <50K cell is at the right. Another left-to-right trend relates to *hours/week*, which is high on the left and low on the right, *i.e.*, correlates with earnings. A third trend, which also causes the upward curve, follows the number of educations and education level. To the left, we find the most educated people (many *educations*, *education level*=BSc/MSc). Going right and then up, education decreases, with the least educated ($1-4^{th}$ grade) in the purple cell top-right. To confirm this, we use the observation plot (Fig. 10 b), with observations colored by number of educations. We see here too the left-right-upwards trend starting with highly educated people, going through mid-educated people, and ending with a sparse cluster of low-education people.

To better understand our visualization's strengths and weaknesses, we also conducted an exploratory user evaluation. The users were 14 computer science students (3 BSc, 7 MSc, and 4 PhD), with 1..2 years of experience with general Infovis techniques, but no knowledge of MDS or MCA. They were given a detailed demo of our tool (45 minutes), and next each had to answer three types of questions:

**Q1:** Find a meaning for cell groups to the top, bottom, right, left, and center of the projection view;
**Q2:** Explain the $x$ and $y$ projection axes in terms of attributes;
**Q3:** Find and explain salient clusters in the observation plot in terms of attributes.

The questions followed our own experiments (Sec. 4), so we could use our findings (unknown to the users) to validate results. For each question, users had to rank the usefulness and ease-of-use of the techniques (selection, brushing, color linking, dimensions view, observation plot, projections view, merging/filtering, and barcharts) on a five-point scale: very high (VH), high (H), low (L), very low (VL), and not used (NU). The assignment took under 2 hours. After that, the users could give additional oral feedback on their experience.

Fig. 11 summarizes the study's findings for 13 users (one user dropped out of the study). Overall, most users found the same cell groups, axis explanations, and clusters as ourselves. Color linking and brushing were found useful and easy to use. Barchart legends scored very well for Q2 and Q3 and were not used for Q1, in line with our design intention for this tool. Merging/filtering scored lowest, which can be explained by the relatively short training time put into this feature (5..10 minutes) and the fact that they require more involved choices (which values to merge or filter and merge distance, see Sec. 4.3). Finally, the usefulness and ease-of-use scores for the dimensions view, projections view, and observation plot indicate that most users perceived these (very) positively.

Although this exploratory study is far from a formal user evaluation, the results suggest that our techniques are relatively easy to learn for novice users, and can support the tasks and questions sketched in Sec. 4 up to a good extent.

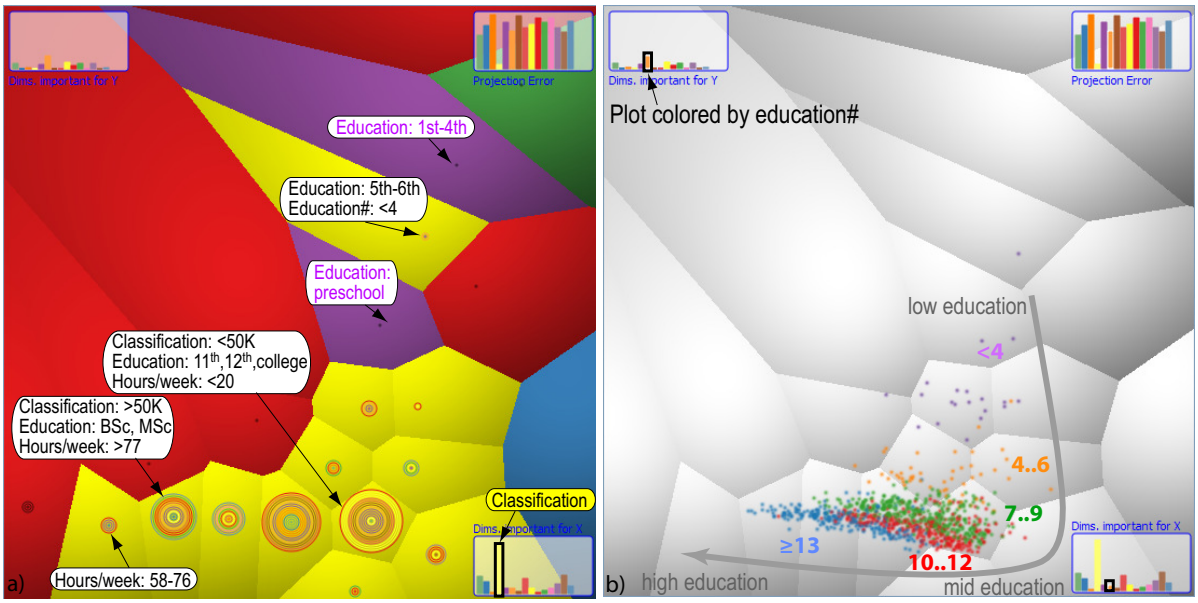


**Figure 10:** *Adult education dataset. MCA arranges data along a curve pattern following education (low..mid..high)*

| Question | | Tools usefulness and ease-of-use | | | | Results | |
|---|---|---|---|---|---|---|---|
| | | color linking | brushing & selection | barchart legends | merging & filtering | | |
| Q1 | VH | 7 | 10 | | | Found right group: | 8 |
| | H | 4 | 3 | | 2 | Found left group: | 7 |
| | L | 2 | | 2 | 6 | Found top group: | 8 |
| | VL | | | 1 | 3 | Found bottom group: | 5 |
| | NU | | | 10 | 2 | Found center group: | 7 |
| Q2 | VH | 6 | 9 | 1 | | Explained x axis: | 9 |
| | H | 5 | 2 | 4 | 3 | Explained y axis: | 8 |
| | L | 2 | 1 | 5 | 6 | Explanation confidence: | |
| | VL | | | 1 | 3 | 8 (sure); 2 (maybe); 3 (none) | |
| | NU | | | 2 | 1 | | |
| Q3 | VH | 11 | 2 | 5 | | Found salient clusters: | 10 |
| | H | 2 | 3 | 5 | | Found other clusters: | 3 |
| | L | | 3 | 3 | | | |
| | VL | | 4 | | 3 | | |
| | NU | | 1 | | 10 | | |

**Figure 11:** *Evaluation results.*

## 6. Discussion

Our visualization, in contrast to MDS techniques, can technically handle both categorical and numerical (binned) data. The main value of MCA is that it enables us to have attributes, attribute values, and observations all in the same projection. In turn, this allows linking attributes with observations, which helps explaining the meaning of projected observations. This addresses one problem of MDS-like plots.

The barchart plots allow seeing which attributes contribute to the $x$ and $y$ projection axes; which are weakly reflected in the projection; and how values of a selected attribute map to projected observations. Understanding the meaning of a scatterplot and/or its clusters requires much less user interaction (clicking a few value bars in the barcharts) than in classical MDS plots where one usually has to cycle through all attributes and color projections based on the selected attribute.

Scalability is covered at several levels: Space-filling Voronoi cells show relative locations and distances of attributes and also which observations most likely sample these. Cell merging, done distance-based or by attribute values, removes understood or uninteresting observations to give more space to project the remaining ones. Computational scalability is good: MCA is $O(J^2I)$ for $J$ distinct attribute values and $I$ observations (Sec. 3.1), under the assumption that $J < I$.

Several limitations exist, though, as follows.

**Colors:** The categorical colormap scheme used for the projections and dimensions views (Sec. 4.1) cannot show more than roughly 10 distinct attributes. Even though the problem is alleviated by using colors to emphasize the most relevant attributes, *i.e.* the ones which are most likely to discriminate between observations, and also by merging cells (Sec. 4.3), the issue still exists. A general solution that can handle datasets having hundreds of attributes, out of which a large subset could be equally relevant, is still required.

**Voronoi cell size:** Voronoi cells partition the 2D plot space to place multivariate information atop projections in a non-overlapping manner. As a by-product, outliers (*e.g.* at the plot periphery) get large cells. Cell area is, thus, a by-product of inter-projection distance, and does not encode data values. Although large cells help locating outlier attribute values, the strong visual salience of area can have undesired effects, *e.g.* users comparing the areas of two cells to draw wrong conclusions about their attribute values. A related issue is the Voronoi cell adjacency: The fact that two cells are adjacent does not carry any additional information besides the fact that they are spatially close, *i.e.* that observations tend to select their respective attribute values together, as explained in Sec. 3.2.

**Observations *vs* cells:** A separate challenge relates to interpreting observations *vs* Voronoi cells in the observation plot. As mentioned in Sec. 4.6, if an observation $\mathbf{x}$ is closer to an attribute value $j$ than to other attribute values $k \neq j$, then $\mathbf{x}$ will more likely have value $j$ than values $k$ relative to the other observations. Thus, if $\mathbf{x}$ falls within the Voronoi cell of some attribute value $j$, it only means that $\mathbf{x}$ will *more likely* have value $j$ than other attribute values. Cell borders are thus only indicators of a change in attribute-value likelihood for observations, and not a precise indication of actual attribute values for observations. Hence, observations that fall within a large cell and are far from the cell borders are very likely to actually *have* the attribute value of that cell. In contrast, for observations that fall close to cell borders, or are located in small, densely packed, cells, we can only say that they more likely take *one* of the attribute values of the respective cells than values of far-away

cells. This interpretation challenge is clearly not trivial, and a recognized limitation of our visual encoding.

**Usability:** Although linking our views to concepts and questions from the application domain is arguably easier than for existing MDS plots, there is still some effort and learning curve required. Making the mapping between questions and views even simpler and more explicit is a main point for future work. Also, investigating the use of MDS techniques, *e.g.* [PNML08] instead of our current MCA technique, would extend the scope of our explanatory visualizations to a larger area.

## 7. Conclusions

We have presented a set of visual analysis techniques for multivariate categorical data. In contrast to classical numerical MDS, we use MCA to create 2D projections which display attributes, attribute values, and observations. We introduce several visual encodings which help correlating values, observations, and observations with values. We showed how our techniques can be used to find non-trivial insights with limited effort in a dataset from the insurance industry.

A standard tool in sociology, MCA is rarely used for information visualization of multivariate data. Yet, categorical data is very common in datasets concerning business processes. To our knowledge, our work is the first application of MCA in visual analytics, and demonstrates the usefulness of this technique in understanding categorical data. We wish that our work, leveraging modern interactive visualization practices on this particular technique, can contribute to making MCA more widespread, bringing its power of explanation to analysts and casual users outside the domain of social sciences. A further direction of work is to leverage the presented visualizations to facilitate the explanation of other dimensionality reduction techniques, such as 2D and 3D MDS scatterplots.

## References


[Abd10] ABDI H.: *Encyclopedia of Measurement and Statistics*. Thousand Oaks, 2010, ch. Singular Value Decomposition and Generalized Singular Value Decomposition, pp. 907–912. 2

[AGMS11] ALSALLAKH B., GRÖLLER E., MIKSCH S., SUNTINGER M.: Contingency wheel: Visual analysis of large contingency tables. In *Proc. EuroVA* (2011), pp. 53–56. 1, 2

[AV07] ABDI H., VALENTIN D.: *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA), 2007, ch. Multiple Correspondence Analysis, pp. 651–657. 2, 7

[AW10a] ABDI H., WILLIAMS L.: *Encyclopedia of Research Design*. Thousand Oaks, 2010, ch. Correspondence Analysis, pp. 267–278. 2, 4, 7

[AW10b] ABDI H., WILLIAMS L. J.: Prinicpal component analysis. *Wiley Interdisciplinary Reviews:Comp. Stat 2* (2010), 433–459. 2

[BB76] BENZECRI J. P., BELLIER L.: *L'analyse des données*, 2 ed. Dunod, 1976. 1

[Ber77] BERTIN J.: *La graphique et le traitement graphique de l'information*. Flammarion, 1977. 1

[BG05] BORG I., GROENEN P.: *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005. 2

[BH11] BREWER C., HARROWER M.: Color Brewer 2.0. http://colorbrewer2.org, Dec. 2011. 4

[Bou79] BOURDIEU P.: *La distinction, critique sociale du jugement*. Editions de Minuit, 1979. 2

[Cau29] CAUCHY A.: Sur l'equation à l'aide de laquelle on determine les inegalités seculaires des mouvements des planètes. *Oeuvres Completes ($2^{eme}$ serie)* (1829). 1

[FA10] FRANK A., ASUNCION A.: UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010. 8

[FB90] FEINER S., BESHERS C.: Worlds within worlds : Metaphors for exploring n-dimensional virtual worlds. In *Proc. UIST* (1990), ACM, pp. 76–83. 1

[FJ11] FERNSTAD S. J., JOHANSSON J.: A task based performance evaluation of visualization approaches for categorical data analysis. In *Proc. Information Visualisation* (2011), IEEE, pp. 80–89. 1

[Fod02] FODOR I.: *A Survey of Dimension Reduction Techniques*. Tech. rep., Center for Appl. Sci. Comp., LLNL, 2002. 2

[Fri94] FRIENDLY M.: Mosaic displays for multi-way contingency tables. *JSTOR 89*, 425 (1994), 190–200. 2

[Fri99] FRIENDLY M.: Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *J. Comput. Graph. Stat. 8* (1999), 373–395. 2

[Fri00a] FRIENDLY M.: *Visualizing Categorical Data*. Statistics and Applications Series. SAS Institute, 2000. 2

[Fri00b] FRIENDLY M.: Visualizing categorical data: Data, stories, and pictures. In *Proc. SAS User Group Conf.* (2000). 2

[Gab71] GABRIEL K.: The biplot graphic display of matrices with application to principal component analysis. *Biometrika 58*, 3 (1971), 453–467. 2

[Gre07] GREENACRE M.: *Correspondence analysis in practice*. CRC Press, 2007. 2

[Hir35] HIRSCHFELD H. O.: A connection between correlation and contingency. *Proc. Cambridge Phil. Soc. 31* (1935), 520–524. 1

[Hot33] HOTTELING H.: Analysis of a complex of statistical variables into principal components. *J. Psych. 25* (1933), 417–441. 1

[Ins97] INSELBERG A.: Multidimensional detective. In *Proc. Infovis* (1997), IEEE, pp. 100–107. 1

[JJJ08] JOHANSSON S., JERN M., JOHANSSON J.: Interactive quantification of categorical variables in mixed data sets. In *Proc. Information Visualisation* (2008), IEEE, pp. 3–10. 2

[JS91] JOHNSON B., SHNEIDERMAN B.: Treemaps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. Infovis* (1991), IEEE, pp. 284–291. 2

[KBH06] KOSARA R., BENDIX F., HAUSER H.: Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE TVCG 12*, 4 (2006), 558–568. 2

[KW01] KOLATCH E., WEINSTEIN B.: Cattrees: Dynamic visualization of categorical data using treemaps. http://www.cs.umd.edu/class/spring2001/cmsc838b/Project/Kolatch_Weinstein/index.html, 2001. 2

[ODH*07] OELTZE S., DOLEISCH H., HAUSER H., MUIGG P., PREIM B.: Interactive visual analysis of perfusion data. *IEEE TVCG 13*, 6 (2007), 1392–1399. 7

[Pea01] PEARSON K.: On lines and planes of closest fit to systems of points in space. *Philos. Mag. 6* (1901), 559–572. 1

[PNML08] PAULOVICH F., NONATO G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE TVCG 14*, 3 (2008), 564–575. 2, 7, 10

[RC94] RAO R., CARD S. K.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proc. ACM CHI* (1994), pp. 318–322. 1

[TLLH12] TURKAY C., LUNDERVOLD A., LUNDERVOLD A., HAUSER H.: Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE TVCG 18*, 12 (2012), 2621–2630. 1

[TvW01] TELEA A., VAN WIJK J. J.: Visualization of generalized Voronoi diagrams. In *Proc. VisSym* (2001), pp. 324–332. 4

[TY85] TENENHAUS M., YOUNG F.: An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data. *Psychometrika 50*, 1 (1985), 91–119. 2

[ZSS09] ZHICHENG L., STASKO J., SULLIVAN T.: Selltrend: Inter-attribute visual analysis of temporal transaction data. *IEEE TVCG 15*, 6 (nov.-dec. 2009), 1025 –1032. 1