Exploring Items and Features with IF, FI-Tables

Citation for published version (APA):

van der Corput, P. N. A., & van Wijk, J. J. (2016). Exploring Items and Features with I^F, F^I-Tables. *Computer Graphics Forum*, 35(3), 31-40. <https://doi.org/10.1111/cgf.12879>

DOI:

[10.1111/cgf.12879](https://doi.org/10.1111/cgf.12879)

Document status and date:

Published: 01/06/2016

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Exploring Items and Features with I^F , F^I -Tables

Paul van der Corput¹ and Jarke J. van Wijk¹

¹Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Figure 1: Exploration of high-dimensional data with I^F , F^I -Tables: based on selected items, the user can find similar items and relevant features; and based on selected features, the user can find similar features and relevant items.

Abstract

The exploration of high-dimensional data is challenging because humans have difficulty to understand more than three dimensions. We present a new visualization concept that enables users to explore such data and, specifically, to learn about important items and features that are unknown or overlooked, based on the items and features that are already known. The visualization consists of two juxtaposed tables: an I^F -Table, showing all items with a selection of features; and an F^I -Table, showing all features with a selection of items. This enables the user to limit the number of visible items and features to those needed for the exploration. The interaction is kept simple: each selection of items and features results in a complete overview of similar and relevant items and features.

Categories and Subject Descriptors (according to ACM CCS): H5.2 [Information Systems]: User Interfaces—Graphical user interfaces H5.2 [Information Systems]: User Interfaces—Interaction styles

1. Introduction

Multivariate data are maybe the most common type of data. In many cases we are able to gain insight in such data using visualization and interaction techniques such as tables and scatter plots in combination with overview & details, focus + context, and filters. One observation is that these techniques are easy to apply on data with many entries (items), but are no real solution for high-dimensional data (many features). Still, data with many features are ubiquitous. Examples are sensor data, profile data, image metadata, network packages, patient records, etc. Since it is difficult for humans to understand data with more than two or three dimensions, it is a challenge to give them insight in high-dimensional data.

In this paper we focus on the design of a generic method to explore high-dimensional data. For that, we use a generic multivari-

ate data model that consists of items $X = \{x_1, \dots, x_n\}$ (also called observations or events), features $F = \{f_1, \dots, f_m\}$ (also called attributes, variables, or dimensions), and values $v_{i,j} = f_j(x_i)$. Such data could be visualized as a table with features as columns, items as rows, and values as cells, see Figure 2. Although the data model is simple, there are still some challenges for visualization. There is often a mixture of numerical and categorical features, and values can be undefined, causing the data to be very sparse in some real world scenarios. We aim to keep the analysis of such data convenient, even in cases with more than one hundred attributes.

Our main contribution is a simple concept that we call I^F , F^I -Tables, pronounced as *eye-ef*, *ef-eye-tables*, which enables users to deal with high-dimensional data using the well-known table metaphor. It supports important operations from previous work like finding similar items and correlated features.

Multivariate data visualization is an active research field, as we discuss in Section 2. We describe the concept of F^F , F^I -Tables in Sections 3 and 4. Details about the implementation of the tables and possible metrics are given in Section 5. Next, we evaluate the concept in Section 6 with a comparison to related work and an example use case. Finally, in Sections 7 and 8, we discuss the results, limitations, future work, and give our conclusions.

2. Related work

Multivariate data visualization has a long history. Many applications and many solutions to problems in this domain have been proposed. Surveys by Wong and Bergeron [WB97], Hoffman and Grinstein [HG97], Chan [Cha06], and Liu et al. [LMW*15] give a detailed overview of the research field. Below we highlight the four aspects and corresponding publications that are most relevant to our approach.

2.1. Standard methods

Tables, scatter plot matrices (SPLOM), parallel coordinate plots (PCP), and pivot tables are the standard workhorses for the exploration of multivariate data. There exist many variations and combinations of these methods. Polaris [SH00] for example extends pivot tables by using other visualizations in its cells. The Table Lens [RC94] summarizes a table such that the distribution of features and the relations between them become clear. LineUp [GLG*13] ranks table rows and visualizes the contributions of the features. There are techniques to reduce clutter in PCPs [NH06, MM08, HVW10], and find orderings [FR11] that are easier to perceive. Scatter plots and PCPs can be mixed [YGX*09] and made configurable [CvW11] so that the advantages of both methods are combined. Even with optimal orderings and proper clutter reduction, these classic methods do not scale well to a large number of features. The maximum is approximately 10 features in most cases.

2.2. Dimensionality reduction

The question is whether we need to consider all features at the same time. Two common approaches to reduce the number of features shown are feature selection and feature extraction. The first approach is to choose the best subset that describes the desired phenomenon [GE03]. A disadvantage is that possibly relevant features are discarded if too few features are chosen.

The other approach is to derive a small set of new features from the original set such that they describe the phenomenon with the smallest error. Principle Component Analysis (PCA) [Jol02], Multidimensional Scaling (MDS) [BG05], and t-Distributed Stochastic Neighbor Embedding (t-SNE) [vdMH08] are examples of feature extraction that are often used to visualize higher-dimensional data in one, two, or three dimensions. The main issue of these methods is that the meaning of the newly defined axes is hard to relate to the original features and hence difficult to interpret. Kim et al. [KCPE16] managed to overcome this by letting the user define the dimensions based on examples.

Figure 2: Multivariate data can be represented by a table where each row corresponds to an item, and each column to a feature.

2.3. Features

Despite all developed methods, understanding a high-dimensional data space is still challenging. In recent years, the research focus is more on visualizing features themselves rather than the individual items. An example is the s-CorrPlot [MMGG15], which reduces each feature to a single point and encodes correlation as the space between them. This way, thousands of features can be visualized simultaneously. The VaR display by Yang et al. [YFH*06] positions features in a two-dimensional space and shows the items with a pixel-based visualization. Qu et al. [QCX*07] and Zhang et al. [ZMM12] show the feature space as a weighted network where the features are the nodes and the edges represent the correlations. Zhang et al. [ZMZM15] additionally combine the network with a scatter plot of the items and introduced a technique to combine numerical and categorical data.

2.4. Dual views

Next to exploring the item and feature spaces, also their interaction can be of interest. Turkay et al. [TFH11] and Yuan et al. [YRWG13] separate the item and feature space into two tightly coupled scatter plots. By selecting items in one plot, the user can inspect the changes in the feature plot and vice versa. Our approach also follows this pattern, but we use tables instead of scatter plots. We describe in Sections 3 and 4 how this allows for fast exploration of the feature space using only a limited set of operations.

2.5. Summary

Standard methods are good at showing relations between a limited number of features, but have difficulties showing more than 10 dimensions simultaneously. There are techniques to reduce the number of dimensions as a pre-processing step for standard methods, but they throw away information and are not intuitive. Another option is to reduce features to dots or other simple representations and treat them as items in standard visualizations. This enables the inspection of relations between features, but the relation to the items is then lost. Dual views show the interaction between the item space and feature space, but it is difficult to spot the most interesting items and features to analyze further.

3. Concept

The table is by far the most popular method to present multivariate data. Items are displayed as rows, features (or attributes) of items as columns, cells contain values of attributes per item. This facilitates quick look-up of exact values. If global patterns, distributions, and correlations between features are important, cells can be filled with graphical items, typically bars, turning columns into vertical histograms. Sorting items by the values of a feature by clicking on a column facilitates fast selection of interesting items, vertical scrolling facilitates linear scanning. Typically, the number of features shown is limited, to prevent the need for horizontal scrolling, and features can be selected separately, often using a somewhat hidden mechanism. We call such a table an I^F -Table. When the values for a large number of features have to be shown, often a transposed table (an F^I -Table) is used, where each row denotes a feature and columns denote items. Typical examples can be found at product selection sites: the user selects a few items that are interesting, next, the values for a long list of features are shown side by side.

Tables do have limitations. For finding clusters and analyzing correlations between features, scatter plots are superior. Also, scalability is an issue, but the classic Table Lens of Rao and Card [RC94] shows how to deal with many items. Hundreds of features are still a challenge though, and if only a subset of these is shown, it is not simple to find for instance additional features that are related to these. Also, if an F^I -Table is used, finding appropriate items with similar properties is typically a separate step, breaking the flow of the exploration process.

We propose a simple design to resolve a number of these issues. We use two juxtaposed tables, an I^F and an F^I -Table that provide complementary views on high-dimensional multivariate data sets. In each table, rows (items or features) can be selected, which are shown as columns in the other table. Furthermore, based on current selections, the similarity of other elements to this selection is calculated and shown. Also, given a selection of items (features), the relevance of features (items) is calculated, shown, and can be used to sort the rows. This design enables users to explore large multivariate data sets, and to find similar and relevant features and items, using simple, well-known displays and interactions.

4. Relevance and similarity

Relevance and similarity are important ingredients for the I^F , F^I -Tables. They are used to relate items and features and hence connect the two tables. We explain the use of these measures by means of an example use case: a United States community data set with socio-economic, crime, and law enforcement properties [Lic13]. Although the data contains only about 2,200 communities, the more than one hundred features make it difficult to spot relations in this large table. Since we are familiar with some of the communities, e.g., famous cities and where we live; and with some of the features, e.g., population and average income; we can use these as a starting point for further exploration. There are obviously some important features and communities that are unknown yet, so the question is how to find them using our current knowledge. We therefore came up with the following pattern, see also Figure 1.

Find similar items. Based on known communities we may want

to find similar ones, for example: ‘Which communities are similar to the ones I like?’ We denote the similarity of an item x to a selected subset of items $X' \subset X$ as $S_{X'}(x)$.

Find relevant features. Given a selection of items, a standard question is what characterizes these items. An example could be: ‘What is special about my neighborhood?’ This requires a selection of this community (and neighboring communities). The answer to this question may give pointers to features that have been overlooked. We denote the relevance of a feature f with respect to a subset of items X' as $R_{X'}(f)$.

Find similar features. Whenever relevant features are known or have been found using the previous step, we may be interested in correlations. An example question could be: ‘What properties are related to the problems in this community?’ This allows us for instance to find crimes that co-occur, or even find possible causes and effects (although we always have to be aware that correlation does not imply causality). We denote the similarity of a feature f to a selected subset of features $F' \subset F$ as $S_{F'}(f)$.

Find relevant items. The opposite of the second step is also possible, which comes down to finding exceptional items given a subset of features. Corresponding questions are in the form: ‘Given these types of crime, which communities are extreme?’ This way we can find communities with exceptionally high or low crime rates. We denote the relevance of an item x with respect to a subset of features F' as $R_{F'}(x)$.

This exploration pattern is schematically depicted in Figure 1. The exact meaning of relevance and similarity is not explicitly specified to keep the pattern generic. Different applications may require different metrics. In Section 5.3 we give examples of metrics that can be used for generic applications.

5. Implementation

We implemented the concept of I^F , F^I -Tables and the pattern of finding similar and relevant items and features in a prototype. A screenshot of this implementation is shown in Figure 3. In the design we aimed at minimal visual complexity and interaction.

5.1. Basic I^F and F^I -Table

The visualization consists of two tables that represent the two main entities: items and features. The left table (I^F) gives an overview of all the items, and the right table (F^I) shows all features. Unlike the classic table concept in Figure 2, initially no values at the intersection between items and features are shown. The idea is to add relevant columns, and hence also the values, to the tables on demand. This is done by selecting rows in the other table, so columns in the F^I -Table are added by selecting the corresponding row in the I^F -Table and vice versa. We assume that users are only interested in a limited number of items or features in each step of the exploration process described in Sections 3 and 4. The display of (temporarily) unnecessary data can be reduced dramatically by adding columns of interest dynamically. The three left-most columns are always present and show respectively the similarity, relevance, and the identifiers or primary keys of the items and features. Names of

Figure 3: Screenshot of the I^F -Table (left) and F^I -Table (right). Colored rows indicate selected items (blue) and features (green). The leftmost columns indicate how similar the rows are to the selection in the same table. The second leftmost columns indicate the relevance of the rows based on selections in the other table. The items and features columns show how many values are present for each row. The +/- column in the F^I -Table shows if the selected items are within a standard deviation (o), above average (+), or under average (-). Selected rows are added as columns in the other table, together with a scented widget showing the distribution within the column. The search box on top of a table can be used to filter rows by their name. The tables visualize a United States community data set [Lic13] and show what is special about the communities with the highest employment rate.

items can be used when available, while features naturally have a name or label themselves.

Since items and features are related in several ways, we got inspired by the Three Table View (3TV) [vdCAvW14]. The 3TV shows one table per entity and encodes relations between rows as glyphs. In this case it is not very interesting to indicate that there exists a relation as most items will have most features and vice versa. Here it is more important to indicate how strong these relations are. We therefore show two bars to indicate (1) the similarity to the selected rows in the same table, and (2) the relevance to the selected rows in the other table. The bars are organized in columns and together form a vertical bar chart. A similar style with bars in cells is used in the other columns, making it easier to see patterns.

The interaction is limited to three actions: (1) selecting rows in a table; (2) sorting rows by one of the columns; and (3) filtering rows. Rows are selected in a similar fashion as in file browsers. Range selects are also supported. Each adjustment to the selection results in an immediate update of the similarity and relevance statistics. Rows can be sorted by selecting the column name. The order of the rows could frequently change when sorted by similarity or relevance because the content of these columns is volatile. We decided to preserve the row order when changing the selection to avoid that users lose rows of interest. As a consequence, users have to reorder the rows manually when needed. Filters can be applied to find specific rows, which can be done by typing (part of) the name of the item

or feature in the search box on top of the table. Additional column filters are explained in Section 5.2. Each filter action immediately hides the rows that do not match the query.

5.2. Column summaries and filters

A major drawback of a classic table is that the distribution of the features is invisible because most of the table is outside the viewport. We could have adopted a Table Lens [RC94] approach, but we opted for histograms in the form of scented widgets [WHA07] on top of the columns. Advantages are that the global distribution of the columns is immediately clear and that users can be enabled to filter quickly. A disadvantage is that correlations between features are not directly visible, but our method provides the feature similarity column as an alternative. Also, sorting a column and looking at the patterns in other columns gives cues for correlation. Nevertheless, if a more detailed view on correlations is needed, an additional scatter plot could be used.

The histograms on top of the columns in the I^F -Table show the distribution of items per feature. The opposite, in the F^I -Table is not trivial, because values from different features have different meaning and are hence not directly comparable. However, it is still possible to compare how features distinguish an item from the other items. Therefore we measure the difference of the value from the mean, normalized by the standard deviation, and plot that distance on the horizontal axis of the histogram. The resulting histogram, on

top of an item column, shows for how many features the item behaves normal and abnormal. A wide spread indicates that the item is special according to many features.

The scented widgets support faceted search [HEE*02], allowing users to quickly find items with certain properties. With the scented widgets in the F^I -Table, something similar can be done to find specific features. The widgets allow for instance queries like: ‘Show the features that are normal for item i and j but extreme for item k .’

5.3. Metrics

Below we give an overview how the four measures $S_{X'}(x)$, $R_{X'}(f)$, $S_{F'}(f)$, and $R_{F'}(x)$ can be defined. Table 1 summarizes the metrics used to display the relations between selected items X' , selected features F' , and the individual rows of the tables.

		given selected rows	
		items X'	features F'
determine for	item x	similarity $S_{X'}(x)$ distance metric	relevance $R_{F'}(x)$ avg. deviation
	feature f	relevance $R_{X'}(f)$ information gain	similarity $S_{F'}(f)$ correlation

Table 1: Metrics used by the system to express relations between a single item or feature and a selection of items and features.

The similarity of an item x to subset X' . The similarity between two items is usually determined by some distance metric. Here we want to determine how similar one item is to a set of items; for this we take the average linkage between x and X' : $S_{X'}(x) = \sum_{x' \in X'} S_{x'}(x) / |X'|$. We use the Gower distance [Gow71] for this comparison, since items contain a mix of numerical and categorical features and there are possibly missing values, so

$$S_{X'}(x) = \frac{\sum_{f \in F'} \delta s(f(x), f(x'), f)}{\sum_{f \in F'} \delta}, \quad (1)$$

where δ is 0 if the comparison $s()$ is undefined (in case of missing values), and 1 otherwise. The comparison depends on the feature type: the normalized distance $s(v_1, v_2, f) = 1 - |v_1 - v_2| / (\max f - \min f)$ is used for numerical features; while for categorical features the distance is 1 if the values are equal and 0 otherwise. Note that we only use the selected features for the computation instead of all features. This eliminates noise from irrelevant features and gives the user more control over the criteria used to compute the similarity.

The relevance of a feature f with respect to subset X' . We consider a feature relevant if it can be used to distinguish the selected items from the others. This means that relevant features contain information that can reveal what is special about the selection. This is often expressed as the feature’s information gain [Qui86]

$$R_{X'}(f) = I(p, n) - \sum_{v \in \hat{f}} \frac{p_v + n_v}{p + n} I(p_v, n_v), \quad \text{where} \quad (2)$$

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}, \quad (3)$$

$p_v = |\{x \in X' | v = \hat{f}(x)\}|$, and $n_v = |\{x \notin X' | v = \hat{f}(x)\}|$, $p = |X'|$, $n = |X - X'|$, and $\hat{f}(x)$ is a discretized version of the numerical value x . The information gain metric is based on the ratio of p_v and n_v for each of the unique values in the feature, so this does not work for real values. To apply this metric generically to all features, we convert numerical features into three discrete ranges. A value is considered *normal* when it is within one standard deviation from the mean and considered *low* or *high* when further away. As an alternative to normalizing by standard deviation, percentiles could be used to make the discretization more robust to outliers. More formally, the following scheme with three ranges is used, although a more fine-grained scheme is also possible:

$$\hat{f}(x) = \begin{cases} \text{low}, & f(x) < \mu_f - \sigma_f; \\ \text{high}, & \mu_f + \sigma_f < f(x); \\ \text{normal}, & \text{otherwise.} \end{cases} \quad (4)$$

The relevance of an item x with respect to subset F' . An item’s relevance is its uniqueness with respect to the selected features, and can be defined in many ways. We define uniqueness as the deviation from the feature’s mean, normalized by standard deviation. Categorical features are converted by taking frequencies of the discrete values. We define the relevance of an item as the average distance over the selected features:

$$R_{F'}(x) = \frac{1}{|F'|} \sum_{f \in F'} \frac{|f(x) - \mu_f|}{\sigma_f}. \quad (5)$$

A possible alternative for (5) could be to remove the absolute sign so that items that maximize or minimize all features are found. Furthermore, instead of an average distance, also the maximum distance can be used to find extreme values for a single feature.

The similarity of a feature f to subset F' . The similarity of two features is often expressed via correlation. For the computation of the similarity, we average the correlation between f and each $f' \in F'$: $S_{F'}(f) = \sum_{f' \in F'} S_{f'}(f) / |F'|$. For numerical features we use the absolute value of Pearson’s correlation coefficient

$$S_{F'}(f) = \frac{\sum_{x \in X} |(f(x) - \mu_f)(f'(x) - \mu_{f'})|}{\sqrt{\sum_{x \in X} (f(x) - \mu_f)^2} \sqrt{\sum_{x \in X} (f'(x) - \mu_{f'})^2}}, \quad (6)$$

and in case of categorical features we use Cramér’s V

$$S_{F'}(f) = \sqrt{\frac{\chi^2}{|X| \min(|f| - 1, |f'| - 1)}}, \quad (7)$$

where χ^2 is derived from Pearson’s χ^2 test

method	encoding			readability				complexity
	feature	item	value	$S_{X'}(x)$	$R_{X'}(f)$	$S_{F'}(f)$	$R_{F'}(x)$	
Table	column	row	cell	T1	T1	T1	T1	$ F \times X $ cells
SPLOM	row/column	$ F ^2$ dots	position	T1	T1	T1	T1	$ F ^2 \times X $ dots
PCP	axis	polyline	position	T1	T1	T1	T1	$ F \times X $ segments
DV/DR	dot	dot	position	T1, 2, 3	T1	T1, 2, 3	T1	$ F + X $ dots
I^F , F^I -Tables	column (row)	column (row)	cell	T1, 2	T1, 2	T1, 2	T1, 2	$ F \times X' + X \times F' $ cells

Table 2: Summary of the various methods with their visual encodings, readability according to the three tasks described in Section 6.1, and the complexity in number of visual elements. The columns of the I^F , F^I -Tables show complete items and columns, whereas the rows show only the selected part. ‘Position’ means the position of the value projected onto the corresponding axis.

$$\chi^2 = \sum_{v \in f, v' \in f'} \frac{(n_{vv'} - \frac{n_v n_{v'}}{n})^2}{\frac{n_v n_{v'}}{n}}, \quad (8)$$

and $n_{vv'}$ is the number of times the values v and v' co-occur. Both versions of $S_{f'}(f)$ give values between 0 and 1, where 0 means no correlation and 1 means that the features are identical. It is not trivial to compare a numerical feature with a categorical feature. For that, we use the method by Zhang et al. [ZMZM15] to convert categorical values to numerical ones by replacing them by the mean of the corresponding values of the numerical feature.

6. Evaluation

We evaluate I^F , F^I -Tables by means of a comparison with alternative approaches, and an example use case.

6.1. Comparison

We first compare I^F , F^I -Tables with standard methods for exploring multivariate data: the TableLens [RC94] (Table), a standard scatter plot matrix (SPLOM), a standard parallel coordinate plot (PCP), and a dual analysis model [TFH11] based on dimensionality reduction (DV/DR). Figure 4 gives a schematic overview of these.

Visual encodings and complexity. These five methods look different because they encode the features, items, and values differently, see Table 2. The visual complexity can be expressed in the number of graphical items that are needed to convey all information. For a table, $|F| \times |X|$ cells are needed. A SPLOM shows $|X| \times |F|^2$ dots for the items, but this can also be interpreted as $|F|^2$ shapes. A PCP only shows $|F|$ axes, but does require $|F| \times |X|$ line segments. In contrast to scatter plots, a large number of items quickly affects the readability because of intersecting line segments. Using dimensionality reduction and dual views, items and features can be represented by a single dot, making this visually simpler than the others. The number of cells used by the I^F , F^I -Tables is dynamic but reduced compared to a regular table under the assumption that not all items and features are of interest at the same time. The number of cells is $|X| \times |F'| + |F| \times |X'|$.

Similarity and relevance. Similarity and relevance are important measures in I^F , F^I -Tables. They are explicitly visualized using bars, and the most interesting items and features according to

these measures are found by sorting the rows. Figure 4 sketches how these four measures are visually encoded by the five methods. It seems plausible that questions about similarities and relevances are easy to read provided the sketches. We can however distinguish different types of questions, some of which are harder to answer. A list of these types is worked out below for items. The examples are symmetric, so items can be replaced by features and vice versa.

1. *Verify whether x is similar to X' or relevant to F' :* This is the simplest type because x , X' , and F' are all known and therefore can be highlighted. Figure 4 shows how similarity and relevance in this case can be read.
2. *Find the item that is most similar to X' or most relevant to F' :* This is potentially more difficult because a question of the first type needs to be answered for each $x \in X$ in the worst case. I^F , F^I -Tables are ideal for these questions because they are designed to sort by similarity and relevance. Dimensionality reduction techniques are ideal to find similarities, but for relevances however, the displacement needs to be tracked and could be problematic if there are many items.
3. *Find a set of items that are similar or items and features that are related:* These are typical questions when no interesting items and features are known and hence cannot be selected. Clusters need to be spotted in a high-dimensional space and I^F , F^I -Tables are not suited for this because all combinations of row selections need to be tried. Dimensionality reduction would be the best solution here because clusters in two-dimensional scatter plots can easily be spotted.

Our conclusion is that comparisons are easy to do with all these methods, provided sufficient highlighting of the items and features of interest. The DV/DR method excels at detecting clusters, whereas the I^F , F^I -Tables is most suited for ranking items and features. A complete overview of which methods can answer which types of questions is included in Table 2.

6.2. Use case

To illustrate the strengths and weaknesses of I^F , F^I -Tables, we show an example use case based on the European Urban Audit data set provided by Eurostat [Eur15]. This data set contains more than 200 features about one thousand cities and countries within the European Union. The statistics include population, living conditions, education, economy, culture, tourism, and labor market. EuroStat provides yearly statistics, and we selected the statistics of the year

(a) Table lens. The similarity of two items, with respect to one feature, can be found by sorting on the column and inspect the proximity of the items. Features are similar if their columns have a similar shape. An item is relevant if it is extremely large or small for the selected features. A feature is relevant if the selection as a whole is extreme for the feature.

(b) Scatter plot matrix. The similarity of items, with respect to two features, is encoded as a distance. Similar features can be found by looking at the shape of the point cloud. An item is relevant if the distance to the other points is large in the corresponding plots. A feature is relevant if it separates the selection from the other points.

(c) Parallel coordinate plot. The similarity of items is encoded as the distance between two polylines. Two features are similar if lines between the axes are parallel. An item is relevant if its position is an outlier on the selected axes. A feature is relevant if the selected items are outliers on its axis.

(d) Dual view / dimensionality reduction. The similarity of items and features is encoded as a distance between points. An item is relevant if its position changes significantly when the set of selected features is changed. A feature is relevant if its position changes significantly after selecting items.

(e) I^F , F^I -Tables. Both similarity and relevance are visualized by the two leftmost columns of the tables. They can be found by sorting on these columns.

Figure 4: Comparison of visualizations for multivariate data showing how similarity and relevance are visually encoded. The blue objects are selected items (X'), the green objects are selected features (F'), and the gray areas denote the bulk of the items. The cyan blob in (d) shows the new position of the items when using only the selected features for PCA. The similarities $S_{X'}(x)$ and $S_{F'}(f)$, and relevances $R_{X'}(f)$ and $R_{F'}(x)$ are displayed for the single item x and single feature f .

≈	←	name	items	+/-	Veliko Tarnovo	Marburg	Tübingen
		CITIES(L)/INDIC_UR	1007		Veliko Tarnovo	Marburg	Tübingen
		CITIES	1007		BG017C1	DE053C1	DE050C1
		Students high. edu., male...	593	+	151.03	122.69	116.45
		Students high. edu., tota...	682	+	378.45	274.49	274.62
		Students high. edu., fem a...	584	+	227.42	151.79	158.17
		Avg. temperature of colde...	211	o		1.7	1.7
		Avg. temperature of warme...	212	o		18.1	19.8
		Rainfall - litre/m²	214	o		559	688.5
		Total number of hours of ...	207	o		4.9	5.6
		Country code	1007		BG	DE	DE
		Cost of a taxi ride of 5 ...	351	o	1.65	10.3	12
		Num. people in accommodat...	234	o	0		25
		Num. days particulate mat...	415	+	108	7	8
		Cost of a combined monthl...	338	o	15	38.1	45.5
		Length of bicycle network...	236	o	5		89
		Cinema attendance indivi...	240	o	96909		

Figure 5: Features sorted by their relevance to the top 25 cities with high student density.

2011, and compiled them into one big table. Some features were added to present statistics per 1000 residents to enable comparisons of larger and smaller cities.

The remainder of this section describes how I^F , F^I -Tables can be used to find (the characteristics of) student cities. We define a student city as a city with many students compared to its population. In each step of the exploration process we show which of the three interactions, <select>, <sort>, or <filter>, is applied. A detailed view on the steps is given in the accompanying video.

Step 1: Find a good starting point. There are two possible starting points for exploration: either use good examples of student cities, or use features that are good indicators. We try the second option because we can easily <filter> the F^I -Table on the word ‘student’. There we find a feature called ‘Number of students in higher education per 1000 residents’.

Step 2: Find typical properties of student cities. There may be other important features without ‘student’ in their name, and our goal is to find these. One way would be to check for correlation, and therefore we <select> the feature we just found and <sort> the F^I -Table by similarity ($S_{F'}(f)$). As a result we find that ‘the proportion of the population aged 20-24 years’, ‘jobs per 1000 residents in public administration, defence, etc.’, ‘proportion of the households that are 1-person households’, and ‘people commuting into the city per 1000 residents’ are correlated to student cities.

Step 3: Find student cities. Extreme items can be found for each (set of) selected features, therefore we <sort> the I^F -Table by relevance ($R_{F'}(x)$) to find cities with extreme high and low rate of students. Pisa, Milton Keynes, Leuven, and Gießen appear to be outliers with over 400 students per 1000 residents.

Step 4: Find global patterns. Now we want to find out what distinguishes typical student cities from other cities. Therefore, we <select> the top 25 cities found in Step 3 and <sort> the features by relevance ($R_X(f)$). Interesting features that pop up are: ‘Country code’ (7 out of the top 25 student cities are Polish); features related to the climate such as average temperature of the coldest and warmest month, rainfall, and hours of sunshine (selected cities do not experience extreme weather); and finally we see that the

≈	⇒	name	features	Jobs p/1000 res. in ...	Prop. pop. aged 20-2...	Students high. edu.,...
		Groningen	113	293.67	16.5	181.67
		Cambridge	207	304.36	14.6	161.99
		Poitiers	234	251.6	13.7	187.65
		Oxford	204	352.19	14.8	229.02
		Nottingham	221	205.44	14	184.72
		Newcastle upon Tyne	215	235.92	12.9	159.92
		Exeter	205	254.73	12	146.43
		Marburg	225	304.96	12.6	274.49
		Nancy	234	220.15	12	173.41
		Greifswald	224	237.81	12.2	227.98
		Bayreuth	234	303.28	10.3	151.27
		Trier	232	255.19	10.5	189.88
		Nijmegen	115	288.52	11.2	109.59
		Würzburg	238	323.87	11	221.52
		Manchester	220	184.05	13.3	142.51
		Utrecht	115	271.56	11.4	103.38

Figure 6: Groningen compared to other cities based on three criteria.

‘number of days that particulate matter PM10 concentrations exceed $50\mu\text{g}/\text{m}^3$ ’ is high, see also Figure 5.

Step 5: Inspect properties. The particulate matter concentration is a strange observation because we cannot link this to student population. We <select> the relevant feature and <filter> out cities with infrequent air pollution. Out of the seven selected cities that remain, still five are Polish. This indicates that particulate matter is probably not directly related to students, but rather related to certain countries, of which one happens to have many student cities.

To get clues about possible causes or effects of particulate matter concentration, we can <select> this feature and sort the F^I -Table by similarity ($S_{F'}(f)$). In the top 20 features, we find: ‘annual average concentration of PM10’, ‘total deaths per year per 1000 residents’, ‘total deaths under 65 per year per 1000 residents’, ‘crude death rate’, ‘annual average concentration of NO_2 ’, and ‘number of registered cars per 1000 residents’. This list could be a reason for further studies on the effects and causes of particulate matter.

Step 6: Find cities based on criteria. We go back to the features found in Step 2 and use these to find interesting student cities according to these features. We <select> the features ‘jobs in public administration’, ‘proportion of population 20-24’, ‘students in higher education’, and <sort> the cities by relevance ($R_{F'}(x)$). Looking at the top 25 interesting cities, we see that famous student cities, such as Cambridge, Poitiers, and Oxford are amongst them, and also Groningen, host of EuroVis 2016.

Step 7: Compare cities. To confirm that Groningen is similar to Cambridge, Poitiers, and Oxford, we <select> Groningen and <sort> the I^F -Table by similarity ($S_X(x)$). Figure 6 shows that the three famous cities appear to be more similar (based on the three selected features) to Groningen than any other city in Europe. To find out what Groningen and Oxford for instance have in common, we <select> both, and inspect their column histograms in the F^I -Table. By placing a <filter> on the upper range of both selected widgets, we see that besides the selected features, the cities also have many jobs in information and communication and are growing in population, see Figure 7. Similarly, by selecting the lower range of the widgets, we observe that both cities have a low proportion of people aged 65 years and older and that there are relatively few cars.

	name	items	+/-	Groningen	Oxford
	Jobs p/1000 res. in infor...	733	+	38.15	39.5
	Jobs p/1000 res. in publi...	706	+	293.67	352.19
	Pop. change over 1 year	840	+	1.4	3.5
	Prop. pop. aged 20-24 yea...	920	+	16.5	19.8
	Students high. edu., tota...	682	+	181.67	229.02

Figure 7: Features for which Groningen and Oxford both score high are found by using the histograms in the F^I -Table.

7. Discussion

The evaluation showed two things: (1) I^F , F^I -Tables enable users to spot relations in high-dimensional data that are difficult to find with conventional methods, and (2) I^F , F^I -Tables enable users to make quite some observations from a data set with 1,000 items and almost 300 features, with very limited interaction. Below are some notes on scalability, limitations, and wishes for the future.

7.1. Scalability

We tested our implementation of I^F , F^I -Tables on a data set with about 1,000 items and approximately 300 features. The question is what the upper limit is on the number of instances. The I^F -Table and F^I -Table are on their own classic tables (except that they have a variable number of columns). Therefore, we can analyze the scalability in terms of a classic table visualization.

We assumed that users are interested in a small subset of items and features in each step of the exploration. If this is not the case, and the user selects all items/features, then the result is a conventional table as described in Figure 2, which is too large to handle.

The computational complexity depends on the metrics used and the number of (selected) items and features. Finding the similarity of one item to a set of selected items based on selected features takes $O(|F'| \times |X'|)$ computations. Finding the relevance of one item based on a set of selected features takes $O(|F'|)$ computations, given pre-computed μ and σ . This comes down to $O(|F'| \times |X'|)$ computations per item, and $O(|X| \times |F'| \times |X'|)$ for the I^F -Table. For finding the relevance of one feature, we must count the distribution of selected items over the bins, which requires $O(|X|)$ computations per feature. Finding the similarity of two features requires $O(|X|)$ computations for Pearson's correlation coefficient or $O(|X|^2)$ for the χ^2 test, but this can be pre-computed. These values need to be averaged for the selected features, so this comes down to $O(|F'|)$ per feature and $O(|F| \times |F'|)$ for the F^I -Table.

7.2. Limitations

I^F , F^I -Tables always require a selection of rows to compute statistics that help users to explore the data. This implies that users must know some items or features in advance to select as a starting point for exploration with I^F , F^I -Tables.

The use case in Section 6.2 shows that exploration can involve many steps. Some require the use of information found in previous steps, such as interesting items or features. This implementation with minimal interaction does not enable easy access to previous

findings, so users need to remember this as shown in Step 6 of the use case. Another problem related to the minimalist interaction is that users sometimes have to select items for computing statistics, but do not need to inspect them as columns in the F^I -Table. Step 4 of the use case describes a situation where the F^I -Table is unnecessarily cluttered with cities, only to find the relevant features.

The system and dataset we demonstrated only use numerical and categorical features. Time series data is challenging to show this way, because each cell then represents multiple data points. The definitions of similarity and relevance will become more complex, and actions like sorting and filtering are undefined.

7.3. Future work

Because a good starting point is important for the I^F , F^I -Tables to be efficient and effective, a pre-analysis step could be useful to learn about the data upfront. One possibility would be to use a dimensionality reduction technique like PCA to visualize clusters or outliers that are potentially interesting to inspect in the I^F , F^I -Tables.

We believe that the combination of I^F , F^I -Tables with other visualizations into one system will be an effective way to get insight in phenomena that are not easily visualized by tables. Scatter plots could for instance be used to visualize the correlation of interesting features that are found in the tables; a timeline can be used to gain insight in the temporal features, etc.

The metrics that are used in this paper are simple and generic, and optimizing these for certain data types and distributions may pay off. Ideally, the metrics would be adjusted automatically to the data, without intervention of the users so they only have to be aware of the concept of relevance and similarity.

8. Conclusions

We presented a new method for the exploration of high-dimensional multivariate data using two juxtaposed tables with simple, well-known interaction techniques. Given some initial knowledge about some properties of items and features, similar and relevant items and features can be found quickly with minimal interaction. The method is generic and has successfully been tested with a mix of numerical and categorical multivariate data. The minimalist approach has its limitations, but we argue that it would be a useful addition to existing multivariate data visualizations. Other methods can either be used to find possible starting points, or to present the properties found using our method. These should be easy to integrate, because we made only few assumptions about the data. Future systems based on our method can be focused on specific applications by replacing the generic metrics by domain specific definitions.

Acknowledgments

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs.

References

- [BG05] BORG I., GROENEN P. J.: *Modern Multidimensional Scaling*, 2nd ed. Springer, 2005. 2
- [Cha06] CHAN W. W.-Y.: A Survey on Multivariate Data Visualization. June 2006. 2
- [CvW11] CLAESSEN J. H., VAN WIJK J. J.: Flexible linked axes for multivariate data visualization. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (Dec 2011), 2310–2316. 2
- [Eur15] EUROSTAT: Cities Urban Audit, 2015. URL: <http://ec.europa.eu/eurostat>. 6
- [FR11] FERDOSI B. J., ROERDINK J. B.: Visualizing High-Dimensional Structures by Dimension Ordering and Filtering using Subspace Analysis. *Computer Graphics Forum* 30, 3 (2011), 1121–1130. 2
- [GE03] GUYON I., ELISSEEFF A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3 (Mar. 2003), 1157–1182. 2
- [GLG*13] GRATZL S., LEX A., GEHLENBORG N., PFISTER H., STREIT M.: LineUp: Visual Analysis of Multi-Attribute Rankings. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (Dec. 2013), 2277–2286. 2
- [Gow71] GOWER J. C.: A General Coefficient of Similarity and Some of Its Properties. *Biometrics* 27, 4 (Dec. 1971), 857–871. 5
- [HEE*02] HEARST M., ELLIOTT A., ENGLISH J., SINHA R., SWEARINGEN K., YEE K.-P.: Finding the Flow in Web Site Search. *Commun. ACM* 45, 9 (Sept. 2002), 42–49. 5
- [HG97] HOFFMAN P., GRINSTEIN G.: Visualizations for high dimensional data mining - table visualizations, 1997. 2
- [HVW10] HOLTEN D., VAN WIJK J. J.: Evaluation of Cluster Identification Performance for Different PCP Variants. *Computer Graphics Forum* 29, 3 (2010), 793–802. 2
- [Jol02] JOLLIFFE I. T.: *Principal Component Analysis*, 2nd ed. Springer, 2002. 2
- [KCPE16] KIM H., CHOO J., PARK H., ENDERT A.: InterAxis: Steering Scatterplot Axes via Observation-Level Interaction. *Visualization and Computer Graphics, IEEE Transactions on* 22, 1 (Jan. 2016), 131–140. 2
- [Lic13] LICHMAN M.: UCI Machine Learning Repository, 2013. URL: <http://archive.ics.uci.edu/ml>. 3, 4
- [LMW*15] LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing High-Dimensional Data: Advances in the Past Decade. In *Eurographics Conference on Visualization (EuroVis) - STARs* (2015). 2
- [MM08] McDONNELL K. T., MUELLER K.: Illustrative Parallel Coordinates. *Computer Graphics Forum* 27, 3 (2008), 1031–1038. 2
- [MMGG15] MCKENNA S., MEYER M., GREGG C., GERBER S.: s-corplot: An interactive scatterplot for exploring correlation. *Journal of Computational and Graphical Statistics* 0, ja (Apr. 2015), 0–0. 2
- [NH06] NOVOTNÝ M., HAUSER H.: Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *Visualization and Computer Graphics, IEEE Transactions on* 12, 5 (Sept. 2006), 893–900. 2
- [QCX*07] QU H., CHAN W.-Y., XU A., CHUNG K.-L., LAU K.-H., GUO P.: Visual Analysis of the Air Pollution Problem in Hong Kong. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6 (Nov. 2007), 1408–1415. 2
- [Qui86] QUINLAN J.: Induction of Decision Trees. *Machine Learning* 1, 1 (1986), 81–106. 5
- [RC94] RAO R., CARD S. K.: The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Apr. 1994), pp. 318–322. 2, 3, 4, 6
- [SH00] STOLTE C., HANRAHAN P.: Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *Proceedings of the IEEE Symposium on Information Visualization* (2000), pp. 5–14. 2
- [TFH11] TURKAY C., FILZMOSER P., HAUSER H.: Brushing Dimensions - A Dual Visual Analysis Model for High-Dimensional Data. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (Dec. 2011), 2591–2599. 2, 6
- [vdCAvW14] VAN DER CORPUT P., ARENDS J., VAN WIJK J. J.: Visualization of Medicine Prescription Behavior. *Computer Graphics Forum* 33, 3 (2014), 161–170. 4
- [vdMH08] VAN DER MAATEN L., HINTON G. E.: Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. 2
- [WB97] WONG P. C., BERGERON R. D.: 30 Years of Multidimensional Multivariate Visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques* (1997), pp. 3–33. 2
- [WHA07] WILLETT W., HEER J., AGRAWALA M.: Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1129–1136. 4
- [YFH*06] YANG J., FAN J., HUBBALL D., GAO Y., LUO H., RIBARSKY W., WARD M. O.: Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis. In *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2006), pp. 191–198. 2
- [YGX*09] YUAN X., GUO P., XIAO H., ZHOU H., QU H.: Scattering Points in Parallel Coordinates. *Visualization and Computer Graphics, IEEE Transactions on* 15, 6 (Nov 2009), 1001–1008. 2
- [YRWG13] YUAN X., REN D., WANG Z., GUO C.: Dimension Projection Matrix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data. *Visualization and Computer Graphics, IEEE Transactions on* 19, 12 (Dec. 2013), 2625–2633. 2
- [ZMM12] ZHANG Z., McDONNELL K. T., MUELLER K.: A Network-Based Interface for the Exploration of High-Dimensional Data Spaces. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE* (Feb. 2012), pp. 17–24. 2
- [ZMZM15] ZHANG Z., McDONNELL K. T., ZADOK E., MUELLER K.: Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map. *Visualization and Computer Graphics, IEEE Transactions on* 21, 2 (Feb. 2015), 289–303. 2, 6