Exploring Distributions: Design and Evaluation

Awalin Sopan, Manuel Freire, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck and Ben Shneiderman

Abstract— Visual overviews of tables of numerical and categorical data have been proposed for tables with a single value per cell. In this paper we address the problem of exploring tables including columns consisting of distributions, e.g. the distributions of movie ratings or trust ratings in recommender systems, age distributions in demographic data, usage distributions in logs of telephone calls etc. We propose a novel way of displaying and interacting with distribution data, and present the results of a usability study that demonstrates the benefits of the interface in providing an overview of the data and facilitating the discovery of interesting clusters, patterns, outliers and relationships between columns.

Index Terms—Information visualization, distributions, overview, tabular visualization.

1 Introduction

Many data sets include distributions. For example statistical data often include age, height and weight distributions for large numbers of geographic regions. Activity logs analysis may require exploring the distribution of activity level over the time of day or day of the week for many users, social media data sets may include the distribution of movie ratings or trust ratings from users. The traditional approach is to spread the distribution information over multiple columns - one per possible value or bucket of values. Traditional table sorting, clustering and interaction methods do not consider the fact that these columns are representing distributions. For example there is no way to see correlations between two distributions (e.g. the age and height distribution of children). To our knowledge no tool provides users with the ability to explore the distributions while taking into account the particular characteristics of distributions.

In this paper we present a set of functionalities that can be used to augment table interfaces to better handle distributions. We focus on visual distribution overviews which are customizable and manipulable by users (see Fig1). We present examples taken from multiple application domains, and report on a usability study which highlights the benefits of the approach and suggest improvements to the interface. Our target users are analysts, i.e. expert users conducting fairly complex research tasks, with some knowledge of statistical analysis. An overview of the distributions should allow analysts to "see the big picture", and identify clusters, trends and outliers that may be candidates for detailed inspection. Additionally, visual overviews of the distributions should help users identify relationships between the distribution columns and other columns, or multiple distributions. Rich user controls and interactive features for sorting and filtering the data are important for those analysts.

Our original motivation was to help colleagues analyze trust ratings and movie ratings in a recommender system. Our users wanted to answer questions such as "do raters use the whole rating scale or not?" (i.e. looking at distributions of ratings per user) or "which films receive both very low and very high ratings?" (i.e. looking at distributions of ratings per movie), there was no tool available to browse or manipulate this data beside scrolling through long tables of data. Professional analysts are proficient users of table interfaces and augmenting them with distribution functionalities seemed necessary to investigate such questions. We investigated the design space of distribution column overviews and report on designs we found useful.

Our main contributions are:

1. New methods to present overviews of distributions with features including distribution-aware sorting, clustering and filtering.
2. An interface to produce and manipulate customizable visual overviews of the distributions.
3. Lessons learned from a user study of this interface.

Distributions have specific properties that make their comparison and analysis different from single valued attributes. For example the order of the values is meaningful (e.g. age or height) and should not be changed. The values in the different bins use a similar scale, and while the absolute values in each bin is important the overall shape of the distribution conveys information as well, requiring appropriate pattern matching methods. Reducing the distributions to single-number statistics, such as median or average, is less informative than seeing the distribution itself. For example, Fig 1 (A and B) contains a table of movies and the ratings they received from users. Instead of only giving the average rating received by each movie, the table includes a column showing the distribution of ratings. We call this a distribution column as each cell in this column contains a distribution. This new type of column was first proposed and implemented in ManyNets[9], a tool we obtained from its authors and expanded as described in this paper to address the need of users to analyze those columns of distribution data. At the top of the column an aggregated histogram overview summarizes the entire column. In C, D and E we show a compact row based distribution column overview that shows the entire column without having to scroll through the table. Distribution-specific properties such as skewness or bimodality can be used to sort the overview, and similarity based algorithms can be used as well for clustering or sorting.

2 Design space of Distribution Overviews

We propose two types of distribution overviews (see Fig2): 1) *aggregated single-cell overviews* merge all the distributions into a single distribution (e.g. by summing all the bars and rescaling to fit), and can fit in a single cell of the table; 2) *row-based overviews* draw compact versions of all distributions at once. Aggregated single-cell overviews find their natural place on top of the column-label and have constant height, as they are very small. Row-based overviews benefit from being seen in a large window and find place in a panel on the side of the main table or in a separate coordinated window.

Multiple representations of distributions are available: histograms, heatmaps and boxplots can represent single distributions, while only heatmaps and boxplots can be used for the compact row-based

- *Awalin Sopan is with Human-Computer Interaction Lab at University of Maryland at College Park, E-mail: awalin@cs.umd.edu.*
- *Manuel Freire is with Human-Computer Interaction Lab at University of Maryland, E-mail: manuel.freire@uam.es.*
- *Meirav Taieb-Maimon is with Human-Computer Interaction Lab at University of Maryland and Ben-Gurion University, E-mail:meiravta@bgu.ac.il*
- *Catherine Plaisant is with Human-Computer Interaction Lab at University of Maryland, E-mail: plaisant@cs.umd.edu.*
- *Jennifer Golbeck is with Human-Computer Interaction Lab at University of Maryland, E-mail: golbeck@cs.umd.edu.*
- *Ben Shneiderman is with Human-Computer Interaction Lab at University of Maryland, E-mail: ben@cs.umd.edu.*

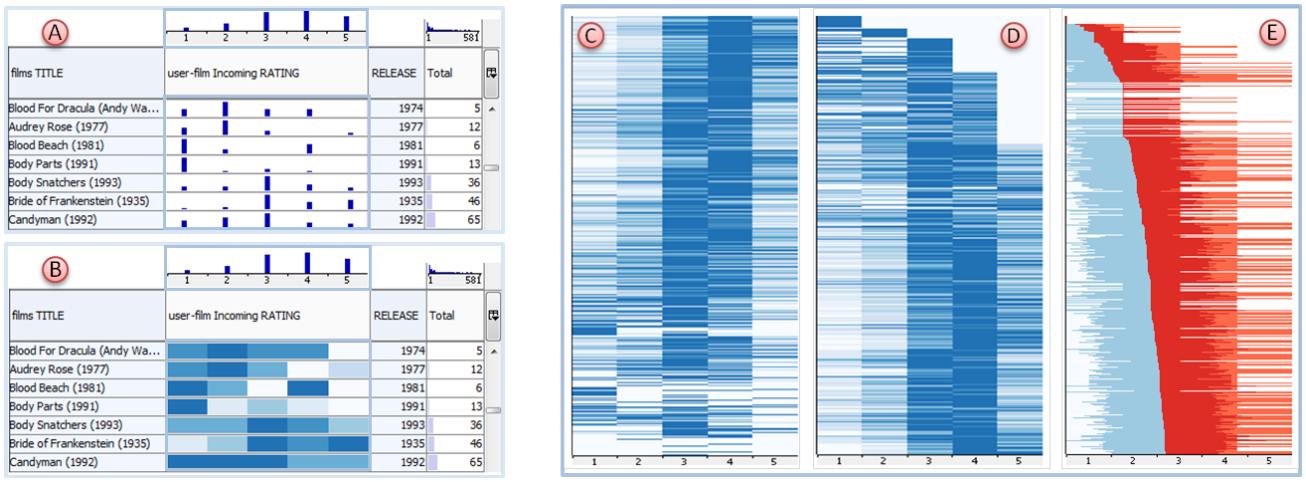

Fig. 1. On the left (i.e. A and B), a table containing distributions of rating received by movies along with total rating and release year. A) Rating distributions are presented as histograms inside table cells. B) same table, now distributions are presented with heatmaps. Only part of the whole column can be shown and scrolling is required. The aggregated single cell histogram overview of all ratings is visible at the top of the column showing the global trend of the rating. On the right (C, D and E) are examples where the the rating distributions are shown in compact "'row based"' overviews showing all the rows without scrolling. Heatmaps can be used in the compact row-based overviews (C, D). All row based overviews can be sorted according to any other column of the dataset (e.g. C, here distributions are sorted by movie Title, but does not show any trend)(D) is sorted by the maximum value(highest rating) of the ratings distributions, and we can see most of the movies received highest possible rating 5 at least once. Heatmap representations can be replaced by other compact views of the distribution e.g. (E) uses stacked boxplots, in this case it is sorted by the average value of the distribution, that is the average rating received by the movie.

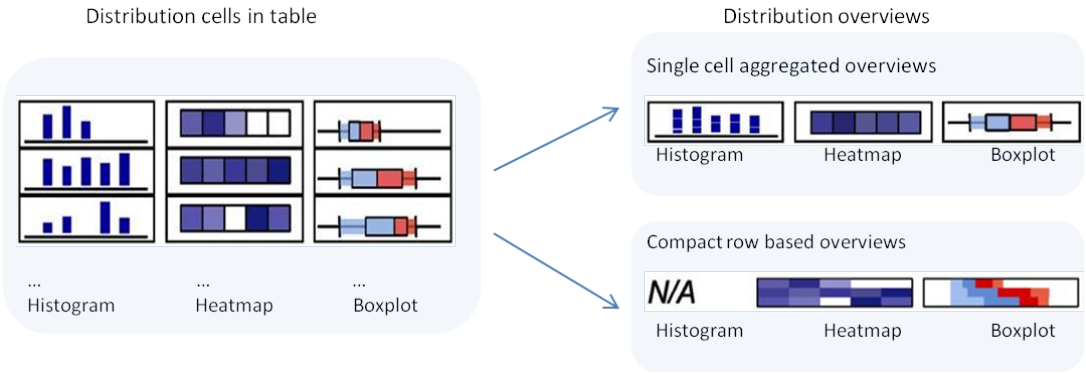

Fig. 2. Aggregated single-cell overviews and row-based overviews for a distribution column. Distributions can be encoded as histograms, heatmaps and box-plots. Multiple distributions can be merged into aggregated one by their union distribution but variability and patterns in the cells of the column can easily be lost in this process. To overcome this, row-based stacked overview can be generated by squeezing visual encodings of each of the distribution cells vertically and that would preserve patterns of trends in the overview.

overviews. In row-based overviews, we can transform the original distribution representation of each cell into a heatmap, reduce each heatmap height to a single pixel, then stack them one on top of another; the result is a heatmap of the entire distribution column.

A key decision for users is to decide how to scale the intensity values in each row, depending on whether they wish to compare the shape of the distributions or the actual bin-counts. When comparing normalized shapes, only the *local* values matter, and intensity values should be scaled according to the highest bin-count in each distribution; otherwise, interval counts can be considered in the *global* context of all distributions in the same column while mapping to color intensity. In all overview types, users can alter the settings on a per-overview basis (e.g. local vs global, or linear vs logarithmic scale).

Stacked boxplots provide overviews of the most important statistics of each distribution; within each boxplot row, the maximum, minimum, average and a standard deviation above and below the average are encoded by color-fields. They can be compacted in a single pixel line and convenient for the row-based overviews. An example is visible in the right most example in Fig 1(E).

To guarantee that all rows are visible without overlaps, there should be at least one vertical overview pixel per row. If there are more rows than available pixels, some aggregation is needed. To present more than one distribution in one vertical pixel, we can take minimum, average or maximum of the intensity values for that pixel.

Users might also want to see overviews of multiple columns at once. In such multi-column overviews, several heatmaps, box-plots overviews are displayed side by side, colored using alternating hues to make the separation between columns more evident. Normal non-distribution columns can be added as well. Although each column of a multi-column overview is separately configurable, they all share the same sorting, so that each row in the overview corresponds to a row in the main table. In general, the order of the rows in the first column of a multi-column overview is carried over to all other columns. Fig 13 shows an example of this multicolumn overview.

Sorting and clustering distributions

When building row-based overviews, the sorting of the rows is critical. Not only is a good sorting important in itself to bring patterns into focus; additionally, the use of image interpolation to map a large numbers of rows into small overviews means that relevant detail may be averaged out. Sorting can be done using three different methods:

- using one of the descriptors specific to a distribution (e.g. bimodality, skewness, average, standard deviation, kurtosis, minimum and maximum value)
- using similarity based algorithms
  - sort by similarity with a selected distribution
  - cluster similar distributions together
- using another column in the main table

We illustrate those sorting options provided through our interface using datasets from movie recommendation systems and phone call domains.

3 Examples of Sorting and Clustering

Our movie recommendation data included distribution columns. We use data from two movie rating systems, FilmTrust [11] and MovieLens [13]. Movies receive multiple ratings from reviewers; by analyzing how movies are rated by various groups we can determine their appropriate target audience, for instance, an average-rated movie may be very popular among a specific group of people. This cannot be learnt just by looking at the aggregated single cell overview.

Sorting using Distribution Descriptor

Sorting from distribution descriptors is mostly useful for ordinal distributions, since many of these concepts are not applicable for nominal distributions. Available descriptors include average, median, maximum, minimum, standard deviation, variance, skewness, kurtosis and bimodality. While movies with many ratings have also received more high ratings in FilmTrust, sorting the overview by bimodality reveals a small group of outliers – movies that received highly mixed reviews (see Fig 3). For further analysis, we select the relevant section of the overview. We filter the table to show only those movies, to examine their rating pattern in a separate detached overview (Fig 3, below). This could also have been accomplished by adding movie bimodality as a sort column, and then sorting the whole table by bimodality. This way we can find out that the most controversial movie (highest value of bimodality) in the dataset was "Double Indemnity".

Sorting using Similarity-based Algorithms

These methods rely on the notion of distance (or its complement, similarity) between two distributions. By observing the sorted overview users can understand the reasons of the similarity which is not obvious without visualization. We compare all the distributions to one another to compute their pair-wise distance. The choice of distance metric depends on whether distributions are nominal or ordinal. In the case of ordinal distributions, the use of a cumulative distribution function (CDF) interpretation allows comparisons between distributions with widely varying numbers of elements. This interpretation is not available for nominal distributions, where adjacent values are completely unrelated. We have implemented both nominal and ordinal distance metrics:

- Euclidean - nominal and ordinal; Euclidean distance considering distribution as a vector
- MDPA - ordinal version of the algorithm described in [6]
- Area - normalized, ordinal; the area between two CDFs
- KS - normalized, ordinal; uses Kolmogorov-Smirnov distance, that is, the maximal distance between CDFs

Normalization implies that the distributions will be compared according to their overall shapes, instead of using the actual counts of elements in each interval. All metrics that compare CDFs are therefore

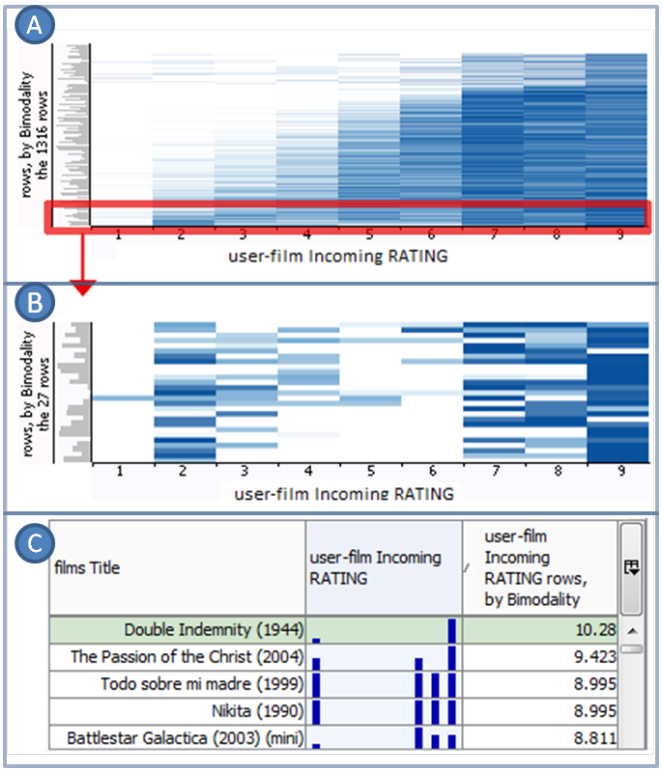

Fig. 3. A: Movie ratings in the FilmTrust dataset, sorted by bimodality. At the very bottom of the overview (see zoomed-in image at B) lie the movies with highest bimodality: users either love them or hate them. These 27 highest bimodally rated movies are selected to create a separate heatmap overview and in C: the portion of the filtered table containing only these movies.

performing normalization. It is important, when displaying the results of clustering or similarity comparison, to make consistent use of normalization. Using Euclidean or MDPA metrics and displaying the results as heatmaps of normalized histograms will generally result in the clusters being undetectable. Therefore, whenever Euclidean or MDPA metrics are chosen, the overview is switched over to global (non-normalized) bin scaling; conversely, when Area or KS metrics are in use, overviews will be displayed using local (normalized) scaling. Computing similarity to a single specific distribution requires $n-1$ distances to be calculated; when clustering, this requires $O(n^2)$ time to build the full distance matrix. We perform cluster-based sorting of the distributions using complete-linkage agglomerative clustering, with a second pass to rearrange the resulting dendrogram using the optimal leaf ordering algorithm described in [4]. The resulting leaf order is then used as the sort order. It is also possible to sort using a nearest-neighbor heuristic TSP, similar to that used in [8]; this approach is faster, but ordering in the last rows tends to suffer.

Sort by similarity to a distribution: To locate distributions that are similar to a particular distribution, we can use "sort by similarity". For the example depicted in Fig 4, contains movies of sci-fi category in MovieLens. We select a popular movie, "Lost World: Jurassic Park" (1997) in the main table and then sort the overview by similarity, using kolmogorov-Smirnov similarity metric, to identify other movies with similar trends. The details and overview both show that these movies have a bell-shaped distribution of ratings.

Sort by clustering: Similar distributions are grouped together in the overview if cluster them using one of the metric discussed before. More examples of these will be discussed in our usability section (e.g. Fig 14).

To compare two communities we can use our clustering technique and compare the overviews from different datasets side by side. Now

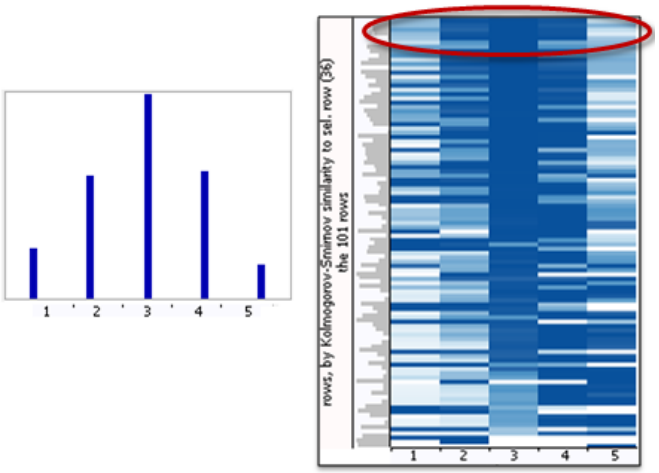

Fig. 4. Movies with rating distributions similar to that of *Jurassic Park* (marked with red oval). The row for Jurassic Park is at the top of the overview. The histogram at the left shows the distribution of rating for this movie which follows a normal distribution pattern.

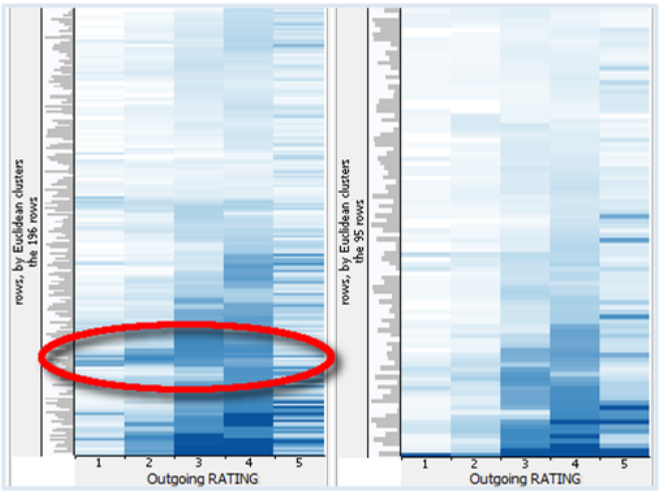

Fig. 5. Left: ratings from students; Right: ratings from educators. In both cases the rating distributions are clustered using Euclidean distance as a distance metric. Group of student critics are marked with red oval.

from movie, we move to the users and look at the distributions of ratings given by the users. We decided to check for differences in rating patterns between users with different occupations (MovieLens includes self-reported occupations for all users). We separated rating-distributions from students and educators (Fig 5). On the left are the students, and we see a small cluster of "'highly critical"' raters (i.e. they have many low ratings of 1 or 2 out of 5). On the right side are the educators and we see no similar cluster, revealing a difference between the two groups.

Multicolumn overview

Multicolumn overviews can help reveal correlation between columns. Her our example uses FilmTrust, a system that is both a movie recommender and a social network of movie raters. Besides rating movies, users can give other users a "trust rating", ranging from one to ten. From the point of view of users, the distribution of all their outgoing movie ratings can be analyzed as a distribution column, and so can the distributions of trust ratings that they have received – and sent. The analyst who had been working with this dataset, hypothesized that users

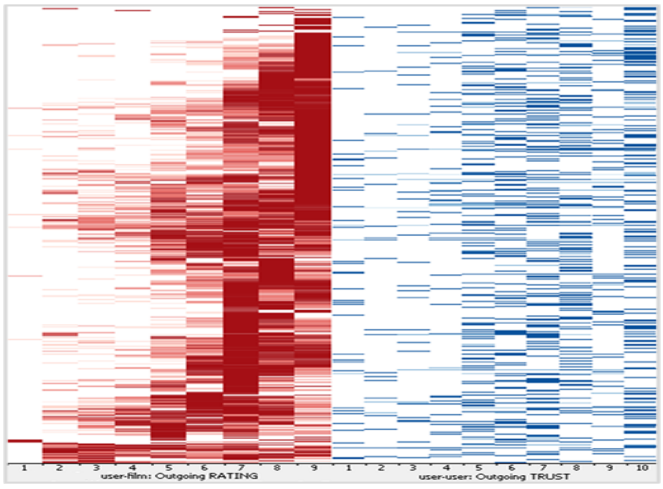

Fig. 6. Multicolumn overview: the left column is the distribution of ratings given by users to movies, the right column is the distribuion of trust rating given by users to other users.

that rate movies very highly would also assign higher trust ratings to their colleagues. Using our system, we have not been able to find any such correlation (see Fig 6). Although users are more generous giving high rating to movies, trust ratings tend to be more moderate. We also allow sorting by simultaneous similarity of more than a single column. For instance, in multicolumn overview clustering everything at once and then looking at the cluster is also possible in addition to order any of the two columns independently of the other.

Sorting using another column in main table

When an overview has been set to sort itself by an external source (another column in the table), it will mirror the order of the rows in that table, and update its rendering whenever the main table sorting changes.

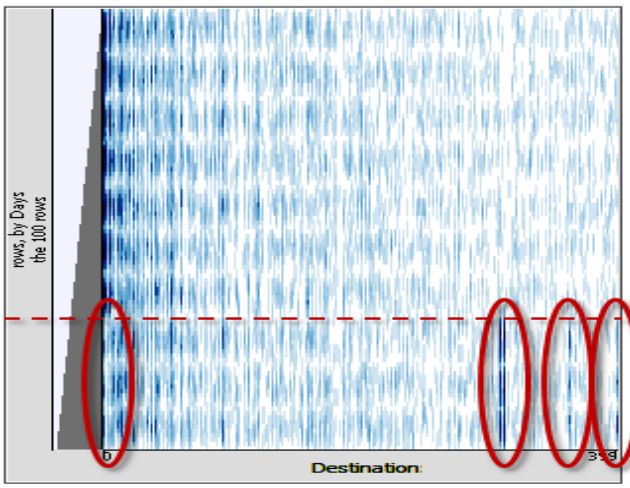

Fig. 7. Overview of destination column of VAST 2008 Mini Challenge 3 timesliced telephone call data; after the first 7 days of calls (red dashed line), represented by 70 rows, several trends change (marked by red ovals). Highly active low-id destinations (leftmost oval) stop receiving calls, while previously inactive high-id destinations (rightmost 3 ovals) start taking in calls.

The VAST 2008 Mini Challenge 3 dataset [12] consisted of simulated telephone calls over 10 consecutive days. We have aggregated the calls into 100 partially overlapping time-slices, 10 per day; each

of these 100 slices is displayed as a row in the table, and contains distributions of, for example, the IDs of the speakers, ranging from 0 to 399 (see Fig 7). If we sort the distribution of call destination IDs by start time, we can see two different regions (see heatmap overview in Fig 7). Up to row 70 (the first 7 days) show a daily occurring pattern of calls, where call destinations with small ID received many calls as compared to other areas. However, from day eight to ten, represented by rows 71 to 99, a different pattern can be seen. Suddenly several destinations with higher IDs, including persons 308 and 397, who were not specially active before, started to receive great numbers of calls. At the same time, the previously popular destinations became silent. This abrupt shift is not easy to find without a complete overview. A separate closer look at the ego network of the newly active IDs would still be needed to reveal the cause of the change (i.e. the suspects had switched to different phone numbers to escape monitoring, but the structures of their ego networks remained unchanged).

4 Interface description

In this section we will describe our tool along with its interaction techniques that lets the users explore the overviews. We use overviews in four roles within the interface. First, each column header is extended with a single-cell overview of the column's content (see Fig 1 and Fig 8). Second, whenever a column is selected, a larger version of the corresponding column overview is displayed in a details-on-demand sidepane (see *A* in Fig 8). Additionally, the distribution formed by merging together all currently-selected cells in the active column is also shown within this sidepane (*B* in the same figure). Finally, it is possible to detach any of the sidepane's overviews and display it in a separate window; detached overviews are no longer linked to table selections, and can be moved and resized freely.

Hovering the mouse pointer for a few seconds over any part of an overview will display a small tooltip, describing the value or values under the pointer. Users can select portion of overviews by selection-drag, and the corresponding rows of the table will become selected and highlighted. Once selected, the values in the selection can be visually compared to those of non-selected rows by looking at the details-on-demand sidepane. The converse is true if a selection is made within the original table: all column summaries will update themselves to highlight the contribution of the selected values within the overviews. Therefore, selection on the overviews can be used to answer the question "what rows contribute to these values", while selection on table rows answers the reciprocal question: "what values are contributed by these rows". Overviews can be configured in a settings panel in the details pane; the settings themselves are hidden unless requested. Fig 8 shows available settings for a heatmap overview. In this case, controls for sorting options, bin height mapping (local or global) and intensity mapping are visible.

For the analysis of distribution columns, we can choose the type of overview, the sort order for row-based overviews and also the distance metric in case of similarity-based sorting. For histogram overview, we can choose appropriate scaling including linear, logarithmic and square root. Fig 9 summarizes the possibilities.

Row-based overviews can be sorted in more ways than the table columns themselves: clicking on any of the table's headers can sort the corresponding column according only to the values (if it is not a distribution) or average values (if it is). Therefore, we allow users to apply complex overview-driven sortings to the main table by adding "sorting columns". For instance, it is possible to add a "skewness" column by sorting an overview by skewness, and then pressing the corresponding button (labeled *C* in Fig 8). This will result in a new skewness column in the main table, showing the skewness values for all the distributions in the overview. If the overview is sorted by clustering, the generated sorting columns will contain the sequential order of rows in the overview. This way we can get the same ordering of rows in the main table as is in the overview. So we can sort the pixel rows in the overview by the table and vice versa.

Since the ordering of row-based overviews can be set independently from that of the main table, we have added a "graphical label" (see callout in Fig 8) to encode, in the length of miniature horizontal bars, the current position of each row of the overview in the main table. Within these labels, currently-selected rows are assigned a dark-green/bright-green color scheme, to distinguish them from unselected rows.

Users can select portions of the overview by mouse-drag and the corresponding rows in the main table get selected. Then we can filter out those entries from the table or keep only those entries in the table. This way we can explore interesting subsets of the data.

The parameter space for these settings can be overwhelming for first-time users; therefore, we have added a "show-me" option (marked as *E* in Fig 8). This brings up a dialog, displayed in Fig 10, with a set of alternative fully-configured overviews appropriate for the current data-type; clicking on any of these options selects the corresponding settings. Since generating cluster-sorted overviews of large amounts of data can require a significant amount of time, small (100-row) random samples are used to render the corresponding thumbnail overviews.

We have used sequential color schemes provided by ColorBrewer [5] to map numerical counts to intensities. There are several options for the value-to-intensity mapping, roughly equivalent to the Y axis scaling for usual bar histograms. The mapping of a normalized value $v \in [0,1[$ to an intensity i is adjustable using a single parameter m: $i = 1-(1-v)^m$. An alternative mapping, used for instance in [16], is based on the sigmoid function: $i = 2/(1+e^{-mv})-1$. Users can choose from available color schemes such as white-to-blue, red-black-green, yellow-to-green, white-to-red etc.

5 User study

Our goal in this study was to investigate if the distribution overviews were easy to interpret, and if users could use the interface to answer representative questions effectively and efficiently. We also wanted to observe what strategies users chose and what problems they would encounter, and gather feedback and suggestions for further improvement.

Procedure

We used a dataset that included 3018 records of the census data of population age distributions in US counties (see Fig 11). Because analysts have very little availability, are hard to recruit for a user study, and the data used in the study is simple enough to be understood by students, the participants were 10 graduate students from various departments. None of them was a member of the development team.

Training consisted of reading a printed manual, seeing a demonstration and interacting with the tool according to a predefined script (including examples of analyzing movie ratings distributions from the MovieLens dataset) and finally answering two training questions. When the participants could answer the training questions correctly, they were considered ready to perform the 1st three study tasks. After the third task additional training was provided on similarity based sorting, and the remaining tasks where completed. Overall each test took about 1.5 hours, of which about 1/3 consisted of training. Participants were encouraged to think aloud while performing the tasks. Observers recorded tasks' completion time and errors, if any. Because the participants needed time to understand the tasks, we gave them time to read the task description before starting the timer. Upon completion of the tasks we debriefed the participants, to learn about their feedback regarding the effectiveness of the tool, ease of using it, whether they found any task to be particularly difficult and their suggestions for improving the tool.

Each task required completion of two stages:

- Interacting properly with the interface to obtain the appropriate visualization of the age-distribution column.
- Once the appropriate visualization was obtained, the participants had to interpret the information presented in the visualization to draw the correct conclusions and properly answer the questions.

For each task the observers recorded task performance and time for each of the two stages. If the participants obtained the wrong visualization, they were given hints (for example see task 6 below) and were encouraged to try again.

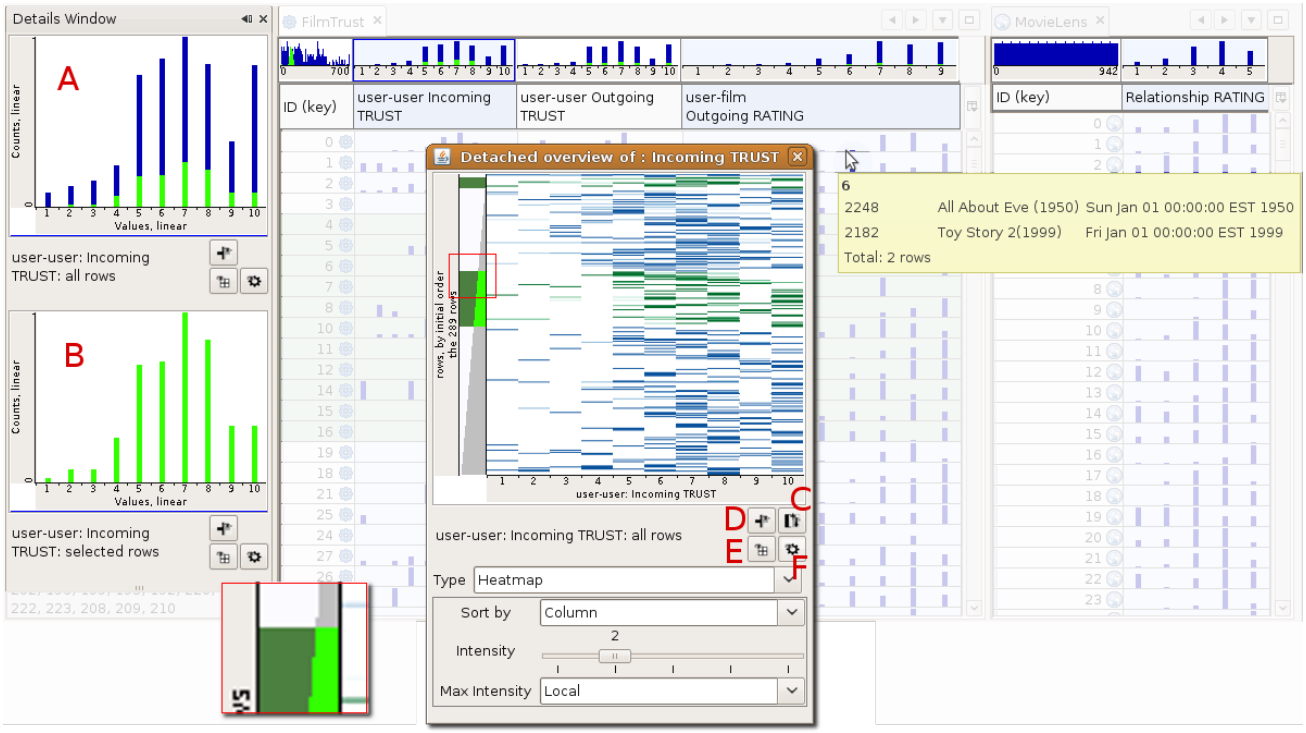

Fig. 8. General view, with current column overview (A) and current selection overview (B). The options for the current overview are displayed (or hidden) by clicking on the options button (F). Button (C) adds a sorting column with the current sorting to the table. The overview can be detached by clicking button (D). Finally, users can choose among recommended settings by clicking on the grid button (E); this brings up the dialog similar to that of Fig 10. The zoomed region demonstrates graphical labels for overviews and selected-row highlighting in row-based distributions.

Tasks

Seven tasks were used (see Table 1) for this study, starting with task 1 to 3 followed by task 4-7 in random order.

Results

The participants were able to select the suitable features and effectively interact with the tool to obtain the correct visualizations. Only two participants needed an additional attempt to produce the multicolumn overview (task 3). This was due to clicking the add/sort column button (which is used to add an additional column to the table sorted in the same order as the overview) instead of choosing the correct type of the overview (multicolumn). This suggests that we should have better labels for the buttons to clearly differentiate between these two options. One participant forgot to change the intensity scaling scheme from Local to Global while performing task 6. This suggests that the scaling scheme should be automatically changed to Global when choosing the Euclidean similarity metric and to Local while choosing the other similarity metrics. All of the participants were able to obtain the expected insights from the visualizations and provided accurate and full answers to the questions.

The Mean $\pm$ SD of the interaction and interpretation times of tasks 1 to 7 are presented in the table. Two separate one way ANOVAs with repeated measures revealed a significant difference in the interaction times ($F(6,54) = 4.3, p < 0.01$) and the interpretation times ($F(6,54) = 7.2, p < 0.001$) among the seven tasks. A Post-hoc Scheffe test showed that the interaction time for task 3 was significantly longer than for the other tasks ($p < 0.05$). Task 3 (multicolumn overview) was much more complex and required manipulating two columns and more interaction steps than the rest of the tasks. Users think aloud explored multiple ways of approaching the tasks, but most eventually find the right way. Similar analysis for the interpretation times revealed that the interpretation times of task 1 and Task 6 were significantly shorter than the interpretation times for the other tasks ($p < 0.05$). Task 1 was indeed much simpler than the rest of the tasks and only required the identification of the highest bin count in the aggregated histogram. Users familiar with standard keyboard shortcuts were able to accomplish the interactive tasks much faster than others. During the debriefing, typical comments included: "'The tool is useful and straightforward, easy to use after demonstration and it was not hard to learn"', "'The sorting options give different ways to visualize these types of data"', "'It allows handling a lot of information , includes a lot of options , "'The heatmap provides a nice improvement over distribution data presentation. When using the heatmap overview with the different types of sorting, it is easier to see patterns, and the differences are more obvious"', "'It shows the big picture and also the outliers". Two participants also asked to analyze the data from their own research with our tool: "'I can use it in the information retrieval domain. In this way I can present distributions of thousands of documents and compare them by the frequency of different terms"', "'I believe this tool can be very useful in the education domain (Educational Measurement and Statistics), for example comparing distributions of exam grades, binned by different questions."' Some participants were confused by the graphical label on the left of the row based overviews. It was meant to relate the position of rows in the overview with their position in the main table, but this had not been explained in the training because we had tried to focus the study on the overview interpretation and not on the overall ManyNets interface. Similarly participants requested access to more details about each row (beyond mouseover information), which is also provided by ManyNets.

Suggestions for improvement included: "'Sometimes I wish things can be circled/annotated. For example the clusters"', "'When comparing side by side you should make it on the same window so you don't have to match the size separately each time."', "'it would be better if I could see how many windows are open and if I could select the windows from the tool bar and switch between them easily"'.

In summary the study suggests that the distribution overview interface is learnable in a short period of time and its functionalities are beneficial in providing an overview of the distributions data, facilitating discoveries of distinctive patterns, clusters, and finding outliers.

Task	What participants were expected to do	Time: Mean±SD(sec) Interaction, Interpretation
1)Across all counties, people of what age range are most prevalent? [aggregated Overview]	After obtaining the appropriate visualization, participants are expected to answer that the 35-44 age range is the most prevalent one.	22±14, 1±0.5
2)In which counties is the population distribution extremely skewed towards youths and in which is it extremely skewed towards elder adults? [Sort using a distribution descriptor]	Identify the two distinct sections: The cluster at the top of counties with distributions which are skewed to the right and the cluster at the bottom of distributions which are skewed to the left. Then, participants should describe what do these patterns imply (counties with high population density of Children and low percentage of elderly people and counties with of high population density of elderly people and low percentage of children, respectively). See Fig 12	30±26, 20±7.3
3)Do counties from the same state exhibit a similar pattern? Does any state stand out as different? [Sort according to a specific column(column State in this case) in the table + Multicolumn overview]	Point out at least one conspicuous pattern in the overview, and use the tooltip to check the name of the State (for example: After sorting by state we can see a group of counties at the top of the overview, all belong to Utah, exhibiting similar distributions) and describe what does this pattern imply (they all have higher percentage of children in their population relative to the other states). See Fig 13	65±46, 25±25
4)Say, you are running a business in Charlotte County in Florida which produces goods targeted towards the elderly population. Which counties in Texas have the most similar shape of age distribution to the distribution in your county so you can consider them as appropriate options for expanding the business in Texas? [Sort according to similarity to a selected distribution/ Search by an example]	To point out Charlotte County, FL at the topmost row in the overview after sorting. The following rows are sorted as higher to lower similarity of age distribution to the Charlotte County. The participants should visually verify the region of higher similarity (The counties at the top of the visualization) and that the similarity is decreasing towards the bottom of the visualization (lower density of elderly population).	44±26, 29±14
5)Use the data set of the topmost 508 populated counties to identify at least three different groups of counties with similar distribution shapes [Clustering, Local (KS or Area)]	Cluster and then identify at least three groups (see the different clusters and describe the different patterns; for example: high percentage of elderly residents, high percentage children, high percentage of middle age residents, etc). See Fig 14.	33±19, 27±11
6)You would like to start up business merchandise targeted for children. You have narrowed down your options to 26 selected counties. You would first like to group the counties according to their bin-counts (total number of people in each category) similarity. Then, you wish to learn which counties have a significant number of children and choose the best option. [Clustering: Euclidean]	Choose clustering and compare by a global similarity metric, since they are asked to compare bin-counts and the number of children (as opposed to distribution shapes and percentages). Identify the cluster of counties having a more children in their population and to choose the row with the distinct maximum color intensity of the younger ages' bins, as the correct answer. Mouse hovering over the topmost row will identify the county. If the participants choose by mistake any local clustering metric, it is re-explained and emphasized to them that the KS or the Area metrics compare the shapes of the distributions, and while using these methods distributions are clustered together according to their shapes, irrespective to the total number of population in each bin. They are then encouraged to make another attempt. See Fig 15.	35±13, 9±5
7)Compare the age distributions of the population of all counties of Florida with those of all the counties of Utah, both clustered by KS metric. Report on differences and similarities. [Side-by-side community comparison]	Identify that Utah has more intense color throughout its left side meaning it has more young population in its counties. In Florida there are only a few counties which exhibit the same pattern (The cluster at the bottom). There is also a distinct cluster at the top of high percentage of elderly people. See Fig 16	55±23, 21±7

Table 1. Task list and time for the user study

In addition, it supports exploration of global trends and relationships between columns.

6 Related Work

Early work on tables tackled the problem of overviews but did not consider distribution columns. In Table Lens [18], the default view is the overview of the whole table, and a focus+context approach is used to expand some areas. As the table grows larger, aggregation occurs at the cell level and like in our tool users can choose the type of aggregation used (e.g. average, min and max, etc.) Standard sorting is available, but not clustering. Another early work - InfoZoom [20], used an overview which is flipped, with attributes located in the rows instead of columns. In this overview mode, it is easy to browse and filter data; however, this method sacrifices the traditional table's basic property of having all attributes of a record aligned together [17].

Heatmaps are commonly used, particularly in bioinformatics for visualizing microarray data [3] because of their good information density and because a sorted heatmap makes it easier to spot blocks and outliers in the overviews. While they are highly sensitive to ordering, the sorting and clustering is typically done considering the heatmap as a matrix, so both rows and columns are reorganized. The values in each cells are not necessarily in the same scale or in any order. In our approach we use dendrogram-based sorting. It has also been used by others to cluster and bring similar rows and columns close together, e.g. [10]; Hierarchical Clustering Explorer [19](HCE) sorts in only one dimension, that of the data rows and in that respect is similar to our approach but does not include any other distribution specific manipulation techniques (e.g. skewness or kurtosis). Other systems offer overviews detached from the table itself, or the original data table is not visible at all, and its contents must be queried through the

Distribution column overviews	Parameters	Possible options
Row-based overview: •Heatmap •Stacked boxplots	**Sort Order**	
	❑ **Sort by Column**	According to table order
	❑Distribution descriptor	Average, Standard deviation, Skewness, Kurtosis, Bimodality
	❑Similarity/Distance based	Similarity to a selected distribution, Clustering
	Distance Metric	
	❑Ordinal	Area, KS, Euclidean
	❑Nominal	Euclidean
	Comparison	Global, local

Fig. 9. The configuration options for distribution column overview

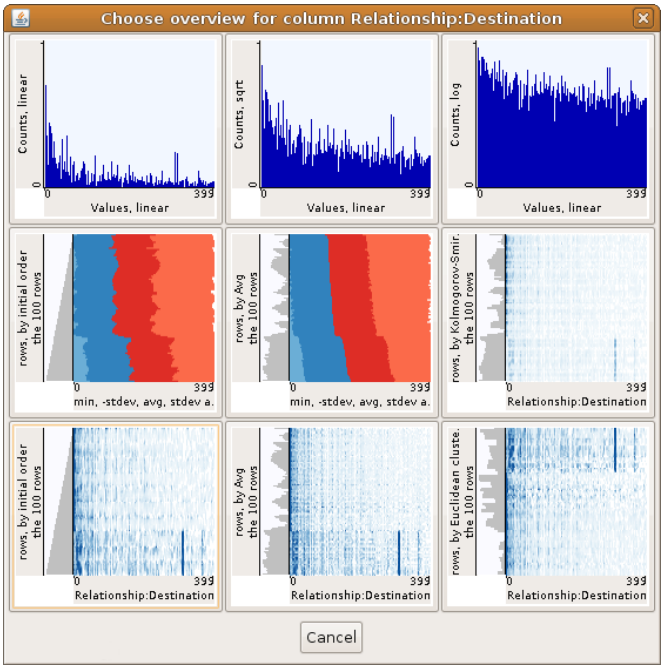

Fig. 10. Grid interface to visually select a fully-configured overview. One of the overviews was used to generate Fig 7

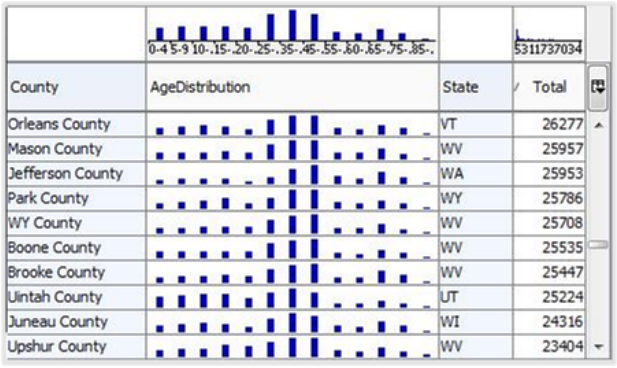

Fig. 11. A sample US county wise population table where each row represents a county and the columns are, from left to right: US county names, Age distribution (distribution of population of different age groups), State and Total (total population counts). The table is sorted by the column Total.

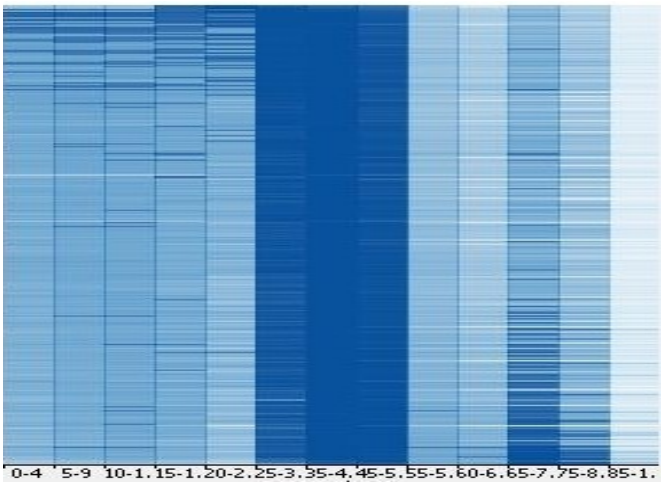

Fig. 12. Task2: Sorting according to skewness of the age distributions of all US counties

overview. HCE also provides several overviews of the multivariate data, all physically separated from the data table. As is expected the different overviews are linked to the table by brushing and linking, but it does not allow filtering the data using the overview.

Parallel coordinates [14] are aimed at multivariate data, and do not address the needs of distribution exploration. Line-graphs, such as stock-market quotes or network traffic statistics, are conceptually similar to distributions. Given a binning strategy for a distribution, line-graphs can be treated as a sequence of numerical values, one per bin. In the case of a set of adjacent numerical columns, each row of cells from these columns also fits this general description (each column will map to a bin). The semantic differences between line-graphs, binned distributions and adjacent columns mandate which types of operations make sense. For instance, the standard deviation is not defined for time-series, but it does make sense for an ordinal distribution. However, in general, overviews for one are readily transferable to another.

Kincaid's Line Graph Explorer [16] (LGE) uses a Table Lens-like interface to display line-graph data with a fisheye effect to reveal details. Similar to our approach, LGE uses color to provide a compact heatmap overview of the data (but again does not consider distribution columns). LGE also uses clustering to bring similar line-graphs closer together. However, in a naive dendrogram ordering, the order of the child branches is essentially random. In comparison the optimal leaf ordering [4] we used rearranges leaves in order to maximize the similarity between adjacent leaves, at the cost of a slight increase in computation time. LGE only allows the comparison of absolute values (e.g. using euclidean distances) making the discovery of shape similarity improbable. When agglomerative clustering is used to sort objects (be these heatmap rows, distributions or time-series), a definition of similarity between pairs of objects to be sorted is required. Similarity metrics are also required in other sorting schemes for visualizations, such as self-organizing maps (used in [15] to lower the $O(n^2)$ cost of building a similarity matrix) or nearest-neighbor TSP routes (used in ZAME [8] for fast adjacency matrix reordering). Common similarity metrics include Pearson correlation and Euclidean distance. Histogram similarity metrics are extensively used in the field of image search [22]. Sung-Hyuk [6] compares different similarity metrics, and introduces the Minimum Difference of Pair Assignments (MDPA) metric, with both nominal and ordinal versions. There is no clear consensus as to which methods are best. We used MDPA and Euclidean distances for global comparisons. A typical question in table visualization is whether several columns are related or not. One option is to use parallel column overviews that share the same sorting. Another option, when testing for pair-wise column correlation, is to use grids of small plots or "trellis plots". Trellises are commonly used as secondary visualizations, though Polaris [21] uses them in a primary role. Again, trellises are tailored for single value columns,

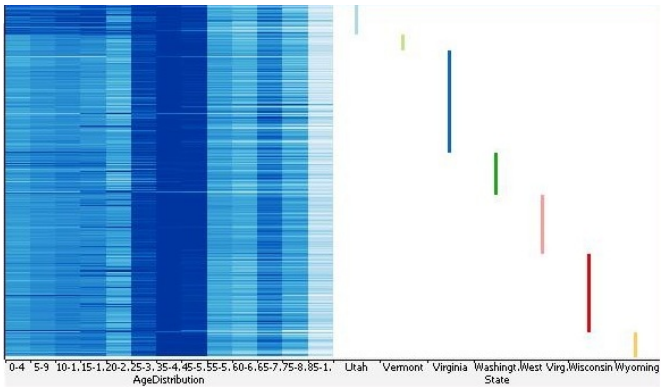

Fig. 13. Task 3: Multicolumn overview sorted by column State. Distribution column age-distribution and categorical column State are appearing side by side. Each State is presented by different color. At the top we can observe all the counties from Utah stand out from the rest as they have more younger population.

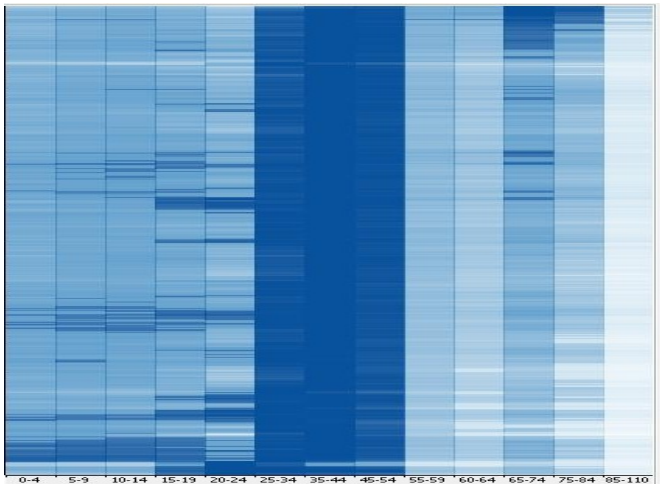

Fig. 14. Task 5: Cluster age-distribution column to compare shapes of distributions, by Kolmogorov-Smirnov metric

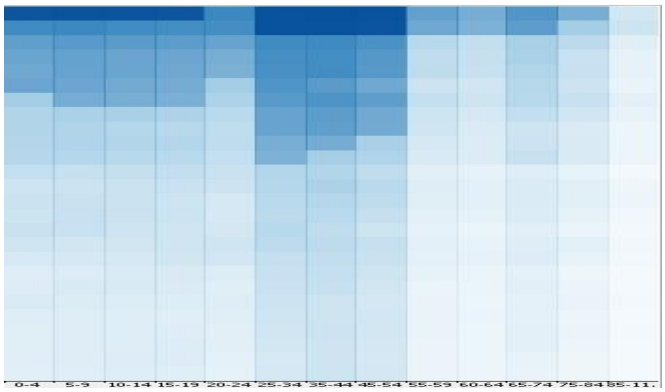

Fig. 15. Task 6: Cluster age-distributions to compare bin-count values of distributions, Euclidean distance metric

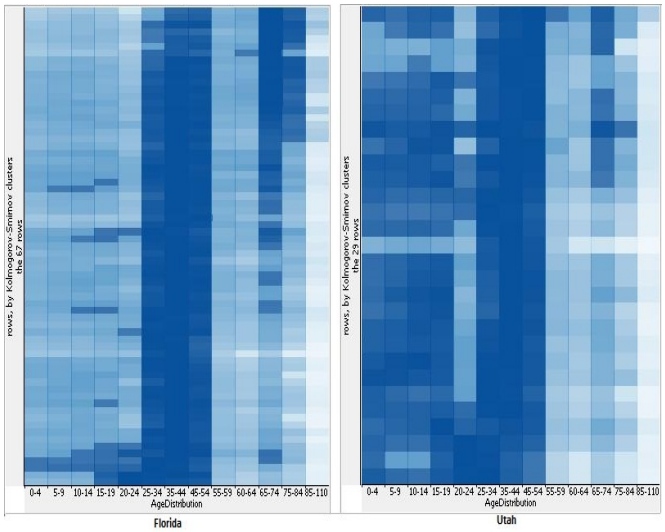

Fig. 16. Task 7:Side-by-side comparison of distributions from two sets after clustering, here age distribution of FL and UT, both overviews are clustered using Kolmogorov-Smirnov metric

not distributions, for example they could not show the correlation between the distribution of children's age and the distribution of their heights. Although there is such a large breadth of related work, we have found no examples of distributions being used as such within a tabular visualization. The closest work is LGE, which has addressed the visual scalability of large collections of line-graphs. Line-graphs are less flexible than distributions, since they cannot be used to aggregate non-ordinal data. LGE uses a focus+context approach, and does not deal with the issues of linking multiple partial overviews on the same data. Additionally, LGE is intended to display a single column of linegraphs next to (but not inside) a table with traditional single-valued cells; line-graph columns are not intended to be freely mixed with traditional columns. WireVis [7] provides fast sorting of heatmaps to bring similar patterns together but different to our approach as it does not consider distribution. General visual analytics package, such as Spotfire [2] or Tableau [1], do not support distribution or line-graph columns. We believe that distribution columns often provide additional flexibility to flat (single-value-per-cell) tables. Developing better overviews for tables with distribution columns is a step towards that direction.

7 Conclusion and Future Work

We have addressed the problem of creating overviews of tables that contain distribution columns. Distribution columns arise naturally in a number of scenarios that involve aggregation. Our approach is general enough for use in cases of distribution-like data, such as independent, adjacent columns or line-graphs. It can be readily ported to any tabular interface, such as Table Lens, and the effort to add additional metrics or visualizations should be small due to the high modularity of the code. The work analyzes several aspects of overviews, including generation, sorting, and clustering. Similarity-based ordering in general and clustering in particular are specially important, since statistical descriptors (e.g. average, variance, bimodality) are not useful in nominal distributions, and often not even in ordinal distributions. Placement of overviews is also a concern; we identify three likely candidates: at the top of each column, in a details-on-demand sidepane, and in a detachable pane or dialog. We address the problem of choosing the best options for a given overview by providing users with context-sensitive a grid display of recommended settings. We illustrate our approach using examples drawn from the domains of recommender systems and VAST 2008 Mini Challenge 3 phone calls data. We have found several interesting trends and outliers. In the case of MovieLens, we characterize differences in film ratings between students and educators. In the case of FilmTrust, we disprove the hypothesis that high-rating users will also assign high trust ratings, and quickly locate films with high bimodality. The key shift in call patterns from the VAST dataset is clearly visible in our overview.

Future work should include dealing with clustering scalability as

our algorithms are still slow when used on large datasets ($O(n^2)$ complexity). The most appropriate clustering algorithm from a visualization point of view (a trade-off between quality and interactive speeds) remains an open question.

ACKNOWLEDGEMENTS

The second author is supported by a MEC/Fulbright Scholarship (Reference No. 2008-0306). Partial support for this research was provided by Lockheed Martin Corporation

REFERENCES

[1] Tableau Software. http://www.tableausoftware.com/. [Online; accessed 21-Mar-2010].

[2] TIBCO Spotfire. http://spotfire.tibco.com/. [Online; accessed 21-Mar-2010].

[3] *VistaClara: an interactive visualization for exploratory analysis of DNA microarrays*, New York, NY, USA, 2004. ACM.

[4] Z. Bar-Joseph, D. K. Gifford, and T. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. pages 22–29, 2001.

[5] C. A. Brewer. Color research applications in mapping and visualization. In *Color Imaging Conference*, pages 1–3. IS&T - The Society for Imaging Science and Technology, 2004.

[6] S.-H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370, June 2002.

[7] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 155–162. IEEE Computer Society, 2007.

[8] N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete. ZAME: Interactive large-scale graph visualization. pages 215–222, 2008.

[9] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. ManyNets: An interface for multiple network analysis and visualization. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI)*. ACM Press, 4 2010.

[10] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004.

[11] J. Golbeck and J. Hendler. Filmtrust: movie recommendations using trust in web-based social networks. In *Proc. 3rd IEEE Consumer Communications and Networking Conference CCNC 2006*, volume 1, pages 282–286, 2006.

[12] G. Grinstein, C. Plaisant, S. Laskowski, T. O'Connell, J. Scholtz, and M. Whiting. VAST 2008 Challenge: Introducing mini-challenges. In *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST'08*, pages 195–196, 2008.

[13] GroupLens Research Project. Movielens 100K Dataset. http://www.grouplens.org/node/73. [Online; accessed 21-Mar-2010].

[14] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization'90*, page 378. IEEE Computer Society Press, 1990.

[15] M. John, C. Tominski, and H. Schumann. Visual and analytical extensions for the table lens. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6809, page 7, 2008.

[16] R. Kincaid and H. Lam. Line graph explorer: scalable display of line graphs using focus+context. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 404–411, New York, NY, USA, 2006. ACM.

[17] A. Kobsa. An empirical comparison of three commercial information visualization systems. In *INFOVIS*, pages 123–130, 2001.

[18] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322. ACM Press, 1994.

[19] J. Seo and B. Shneiderman. Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):311–322, May/June 2006.

[20] M. Spenke. Visualization and interactive analysis of blood parameters with infozoom. *Artificial Intelligence in Medicine*, 22(2):159–172, 2001.

[21] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, pages 52–65, 2002.

[22] M. A. Stricker and M. Orengo. Similarity of color images. pages 381–392, 1995.