This article was downloaded by: [University Of Maryland]
On: 04 August 2014, At: 07:38
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Human-Computer Interaction
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/hihc20

Exploring Data Distributions: Visual Design and Evaluation

Awalin Sopan [a], Manuel Freier [b], Meirav Taieb-Maimon [c], Catherine Plaisant [a], Jennifer Golbeck [a] & Ben Shneiderman [a]

[a] Department of Computer Science and Human-Computer Interaction Lab, University of Maryland, College Park, Maryland, USA

[b] Department of Computer Science, Universidad Complutense de Madrid, Spain

[c] Department of Information Systems Engineering, Ben-Gurion University, Beer-Sheva, Israel

Accepted author version posted online: 30 Apr 2012.Published online: 03 Jan 2013.

To cite this article: Awalin Sopan, Manuel Freier, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck & Ben Shneiderman (2013) Exploring Data Distributions: Visual Design and Evaluation, International Journal of Human-Computer Interaction, 29:2, 77-95, DOI: 10.1080/10447318.2012.687676

To link to this article: http://dx.doi.org/10.1080/10447318.2012.687676

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

Intl. Journal of Human–Computer Interaction, 29: 77–95, 2013
Copyright © Taylor & Francis Group, LLC
ISSN: 1044-7318 print / 1532-7590 online
DOI: 10.1080/10447318.2012.687676

Taylor & Francis
Taylor & Francis Group

Exploring Data Distributions: Visual Design and Evaluation

Awalin Sopan[1], Manuel Freier[2], Meirav Taieb-Maimon[3], Catherine Plaisant[1], Jennifer Golbeck[1], and Ben Shneiderman[1]

[1]*Department of Computer Science and Human-Computer Interaction Lab, University of Maryland, College Park, Maryland, USA*
[2]*Department of Computer Science, Universidad Complutense de Madrid, Spain*
[3]*Department of Information Systems Engineering, Ben-Gurion University, Beer-Sheva, Israel*

Downloaded by [University Of Maryland] at 07:38 04 August 2014

Visual overviews of tables of numerical and categorical data have been proposed for tables with a single value per cell. This article addresses the problem of exploring tables with columns that consist of cells that are distributions, for example, the distributions of movie ratings or trust ratings in recommender systems, age distributions in demographic data, usage distributions in logs of telephone calls, and so on. This article expands on heatmap approaches and proposes a novel way of displaying and interacting with distribution data. The usability study demonstrates the benefits of the heatmap interface in providing an overview of the data and facilitating the discovery of interesting clusters, patterns, outliers and relationships between columns.

1. INTRODUCTION

Many data sets include distributions, such as age, weight, or income distributions for numerous countries, counties, or cities. An activity log shows distributions of activity level by hours in a day or days in a month, whereas social media data can include distributions of movie or trust ratings from users. The traditional approach spreads the distribution information over multiple columns, placing values or range of values in each column, for example, age distributions using a separate column for each age range (0–9, 10–19, etc.). This simple strategy is convenient but makes it difficult to compare distributions as a whole. On the other hand, reducing the distributions to single-number statistics, such as median or average, is less informative than seeing the distribution itself. Distributions have important properties such as uniformity, skewness, or bimodality, which can be used to sort them. In addition, distributions can be clustered so that similar distributions are together. There would be value in a new technique that shows entire distributions in a single column. We call this a "distribution column," as each cell in this column contains a distribution. This notion of distribution columns was first introduced in ManyNets (Freire, Plaisant, Shneiderman, & Golbeck, 2010). For example, Figure 1 (A and B) contains a table of movies and the user ratings they received. Instead of displaying only average movie ratings, the table includes a column showing the ratings distribution.

Traditional table sorting, clustering, and interaction methods do not deal with groups of columns that represent distributions, and therefore offer no easy way to check for correlations between two distributions, (e.g., the age and height distribution of children). Our motivation was to help colleagues analyze trust and movie ratings from a recommender system using a tabular interface. These users wanted to answer questions such as, "Do raters use the whole rating scale or not?" or "Which films receive both very low and very high ratings?" Professional analysts are proficient users of table interfaces, but augmenting them with distribution functionalities would provide powerful tools to investigate such questions. There was no tool available for browsing or manipulating this data beside scrolling through long tables. To deal with these issues, we extended our visual analytics tool ManyNets. We added functionalities to it to enable users to conveniently explore distributions by their properties. To the best of our knowledge, this is the first visual analytics tool to offer such capability.

This article describes the functionalities that augment table interfaces to better handle distributions. In particular, we introduce "distribution column overviews" that allow users to analyze multiple large distribution columns with limited screen space. We focus on visual overviews that are customizable and manipulable (see Figure 1). In Figure 1, the compact row-based distribution column overview of (A) shows the entire column without having to scroll through the table. We also added the capability to sort the overview using distribution-specific

2010 *Mathematics Subject Classification*. 68U35[1].

Partial support for this research was provided by Lockheed Martin Corporation.

The second author was supported by an MEC/Fulbright Scholarship (Reference No. 2008–0306).

[1]*ACM 1998 CCS.* H.5.2 - Information Interfaces and Presentation - User Interfaces - Miscellaneous

Address correspondence to Awalin Sopan, Department of Computer Science and Human-Computer Interaction Lab, University of Maryland at College Park, 4320 Rowalt Drive, College Park, MD 20742. E-mail: awalins@cs.umd.edu

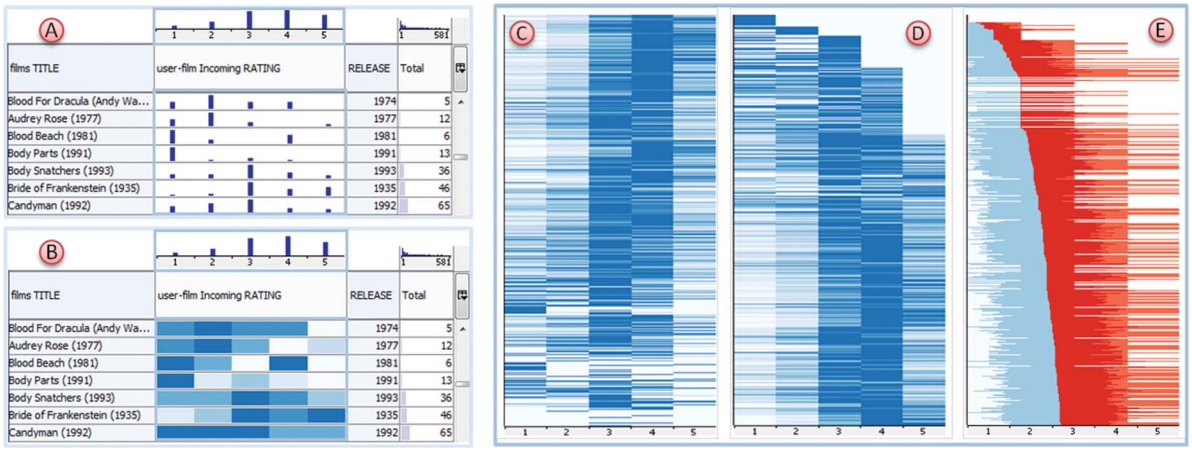

FIG. 1. On the left (i.e., A and B), a table where each row is a movie. *Note.* The columns are movie title, distributions of ratings received by movies, total rating count and release year. (A) Rating distributions are presented as histograms inside table cells in column "user-film Incoming RATING." (B) same table, with distributions presented as heatmaps. Only part of the whole column can be shown and scrolling is required to see the rest. An aggregated histogram overview of all ratings is visible at the top of the column, showing the global trend of ratings. On the right (C, D, and E) are examples where the the rating distributions are shown in compact "row-based" overviews (with one or more table-rows mapping to a single pixel-row), which require no scrolling. C and D use heatmaps (similar to those in B, sorted by two different table columns. C is sorted by movie title, whereas D is sorted by the highest values of the corresponding rating distributions; we can see that most of the movies received a highest possible rating of 5 at least once. Heatmap representations can be replaced by other compact views of the distribution, for example, (E) uses stacked box-plots, and sorted by the average value of the rating distributions (i.e., average movie rating) (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

properties such as skewness or bimodality and cluster it using similarity-based algorithms. The distribution overviews are linked to the original table, enabling analysts to perform standard table operations such as filter, sort, and select rows of table. We investigated the design space of distribution column overviews and report on designs we found useful. We present examples from multiple application domains and a usability study demonstrating the benefits while suggesting interface improvements.

Our main contributions are as follows:

1. An interface to produce and manipulate customizable visual overviews of distribution columns with features including distribution-aware sorting, clustering, and filtering. These overviews let users work with distributions as a single column, sort by distribution properties (skewness, bimodality, etc.), cluster similar distributions, and identify outlier distributions.
2. New methods to explore overviews of distributions coordinated with tabular visualization. Integration with a table lets users find correlations between distribution columns and other columns, along with the standard table manipulation features, such as sorting, filtering, and selecting rows.
3. Results from a usability study that demonstrate that after a short training session users can interpret the distribution column overviews correctly and can find clusters, outliers, and trends.

2. RELATED WORK

Early work on tables tackled the problem of overviews but did not consider distribution columns. In Table Lens (Rao & Card, 1994), 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, as in our tool, users can choose the type of aggregation used (e.g., average, min and max, etc.). Another early work—InfoZoom (Spenke, 2001)—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 (Kobsa, 2001).

We default to using heatmaps to represent distributions. Another option would have been to use parallel coordinates (Inselberg & Dimsdale, 1990), but they are generally less compact and harder to read, especially for high numbers of dimensions. Heatmaps are heavily used in bioinformatics to visualize microarray data (e.g., Gentleman et al., 2004; Kincaid, 2004; Vehlow Heinrich, Battke, Weiskopf, & Niesett, 2011). They provide high-information density and facilitate the spotting of blocks and outliers in the data. Spotting is much easier when similar rows and columns are placed close to each other using sorting and clustering. When there is a low number of columns (attributes), or when attributes are highly heterogeneous, it may only make sense to sort and cluster the rows; this is the approach we currently follow. In contrast, because bioinformatics applications deal with many relatively homogeneous attributes, they tend to consider the whole heatmap as a matrix, and reorganize both records and attributes to facilitate block and outlier detection. Heatmaps are also used outside of bioinformatics; for instance, Henry Goodell, Elmqvist, and Fekete, (2007) analyzed conference data using a matrix representation of a heatmap based visualization, displaying changes in keyword frequencies,

Downloaded by [University Of Maryland] at 07:38 04 August 2014

citation patterns and conference impact over time, with matrix cell color intensity representing values.

General visual analytics packages, such as Spotfire (http://spotfive.tibco.com) provide heatmap views but do not take advantage of distribution-aware sorting and do not provide fine-grained control over the process of abstracting large numbers of data rows into a limited space. Developing better overviews for tables with distribution columns is a step in that direction. A fairly general-purpose visualization system that also relies on clustered heatmaps is the Hierarchical Clustering Explorer (HCE; Seo & Gordish-Dressman, 2007). HCE also provides several overviews of its multivariate data, separated from the actual data table, and its different overviews are linked to their original table by brushing and linking. We follow a similar approach but also allow the table to be filtered and sorted using the overview itself.

HCE relies heavily on its namesake sorting method (hierarchical clustering), also known as agglomerative clustering or dendrogram-based sorting, to place related rows together. In a standard dendrogram, the order of child branches within their parent branch is essentially random. It is possible to compute an optimal leaf ordering (Bar-Joseph, Gifford, & Jaakkola 2001) that rearranges leaves in order to maximize the similarity between all pairs of adjacent leaves, at the cost of a slight increase in computation time. We have added this refinement to our dendrogram ordering step. To achieve quicker cluster generation (the cost of building a similarity matrix is $O(n^2)$), some systems use several passes; for example, WireVis (Chang et al., 2007) sorts similar accounts together by using a fast keyword-based binning approach as a preprocessing step, and only then applying hierarchical clustering; similarly, in John Tominski, and Schumann, (2008), self-organizing maps are used to speed up the process. Regardless of the sorting method, a definition of similarity between pairs of objects to be sorted is always required. For example, similarity metrics are used when building self-organizing maps or generating nearest-neighbor traveling-salesman problem (NNTSP) routes (used in ZAME for fast adjacency matrix reordering; Elmqvist, Do, Goodell, Henry, Fekete, 2008). Matrix reordering methods, such as matrix diagonalization or principal component analysis (PCA), also rely on an implicit definition of similarity. In the case of PCA, Elmqvist et al., (2008) found it to be more complex and of lower general quality than simple NNTSP. Common explicit similarity metrics include Pearson correlation and Euclidean distance. Histogram similarity metrics are extensively used in the field of image search (Stricker & Orengo, 1995), and can be readily adapted to compare arbitrary ordinal distributions. Sung-Hyuk (Cha & Srihari, 2002) 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 use MDPA and Euclidean distances for global comparisons (comparisons between rows that were not internally normalized), and Kolmogorov-Smirnov, Euclidean, MDPA, and area metrics for all other cases.

In most of these tools, individual attributes are of a scalar nature, and heatmaps only arise when many of them are placed next to each other in a table; iHAT (Vehlow et al., 2011), a bioinformatics visualization tool, is an exception, as it supports distribution-aware, row-based aggregation. Distributions in iHAT must be externally binned and are termed "multivariate samples." Row aggregation is based on the chosen clustering, as users can choose to aggregate similar rows based on their "similarity depth." At the shallowest depth, all rows are aggregated into a single "consensus row," whereas at the lowest depth (no aggregation), each row is represented as an individual heatmap. Once aggregated, rows can be sorted according to any metadata available in the table. iHAT is oriented toward genomic data, which is generally nominal, although it also supports ordinal data. Within a nominal multivariate-sample section of a consensus row, hues are determined by dominant values, whereas intensity and saturation are used to convey the relative frequency of these values. iHAT does not provide out-of-table overviews of distribution columns (although a higher level abstraction can be considered to be an in-table overview) and has no support for brushing and linking between different views, as the interface is designed to be single view. Although it supports sophisticated sorting, it must be based on existing metadata, which, in the case of distribution aggregates such as skewness or bimodality, would have to be generated using external tools. 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 overview.

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. For 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 (LGE; Kincaid & Lam, 2006) 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, and clustering to bring similar line graphs closer together. LGE only allows the comparison of absolute values (e.g., using Euclidean distances), making the discovery of shape similarity improbable.

A typical question in table visualization is whether several columns are related. One option is to use parallel column overviews that share the same sorting. Another option, when testing for pairwise column correlation, is to use grids of small

Downloaded by [University Of Maryland] at 07:38 04 August 2014

plots or "trellis plots." Trellises are commonly used as secondary visualizations, though Polaris (Stolte Tang, & Hanrahan, 2002) uses them in a primary role. Again, trellises are tailored for single-value columns, not distributions; for example, they could not show the correlation between a distribution of children's ages in a set of regions and the distribution of their heights. We believe that distribution columns often provide additional flexibility when compared to flat (single-value-per-cell) tables and that generating overviews of these distribution columns facilitates analysis.

Another interesting example of related work is that of TimeMatrix (Yi, Elmqvist, & Lee, 2010), where the temporal strength of connections between network nodes can be displayed as miniature histograms within the cells of an adjacency matrix. In a sense, these histograms can be considered distributions; on the other hand, because they are embedded in fixed locations that correspond to individual nodes, and not to general record/attribute cells, the TimeMatrix visualization cannot strictly be considered a table display. Miniature histograms can also be used inside node headers to represent aggregated node connectivity over time, in a very similar row to our own distribution column overviews.

Although there is a large breadth of related work, we have found no examples, outside of iHAT (and perhaps TimeMatrix), of distributions being used *as such* within a tabular visualization. The closest work is probably LGE, which addresses the visual scalability of large collections of linegraphs. However, LGE uses a focus+context approach and does not deal with the issues of linking multiple partial overviews on the same data. In addition, LGE is intended to display a single column of line graphs 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.

3. DESIGN OF DISTRIBUTION OVERVIEWS

Because tables often have too many rows to fit on the screen, scrolling up and down is needed. When the data within a column exhibit a certain trend, or several columns are correlated, overviews of these columns can help users to identify and characterize them. The coordination of the overview with the table view is also very important for meaningful analysis.

We propose two types of distribution overviews (see Figure 2): (a) *Aggregated single-cell overviews* merge all the distributions on the column into a single distribution (e.g., by summing all histogram bars and rescaling to fit). They require only a single cell of the table to be displayed and find their natural place at the top of the column, (b) *Row-based overviews* draw compact versions of all distributions at once, merging them only as required to conserve limited space. Row-based overviews require more display space and are therefore placed in panels on the side of the main table, or in separate coordinated windows. In addition, row-based overviews can be combined into multicolumn overviews.

Distributions can be represented using several visual encodings, such as histograms, heatmaps, and box-plots. Heatmaps are by far the most popular encoding in high-density visualization applications, whereas histograms and box-plots are both frequently used in stand-alone representations. For unknown distributions, histograms are generally preferred. We adopt the usual convention, with bins placed along the horizontal axis, whereas vertical bar-height represents bin counts. A histogram in each cell of the column is useful in comparing values across bins, whereas heatmaps facilitate the global comparison of many distributions. Box-plots, on the other hand, are typically used to display and compare bell-shaped distributions. Within each box-plot row, we represent the maximum, minimum, average, and a standard deviation above and below the average by color-fields (see Figure 1E for an example). All of these encodings can be used to represent individual distributions within a table cell (or the single-cell aggregated overview at the top of the column), but only heatmaps and box-plots make sense for compact row-based overviews, as histograms become unreadable when vertically compressed.

Within a column, distribution cells can be compared by general shape or by actual quantitative values. To compare cells by general shape, their representations must first be normalized; that is, histogram bar heights or maximum-intensity values must be made relative to each cell's local maximum, so that height and intensity will only be relevant as compared to other heights and intensities within the same cell. We refer to this scaling as *local*, as opposed to *global* scaling, where bar heights and maximum-intensity values are assigned to the highest value among all the distributions in the column. Users can toggle all column and overview representations between local and global.

Distribution data can be either nominal (bins represent unrelated categories) or ordinal (bin order is important, as in continuous values discretized into bins); other data-types, such as ratio or interval, can be readily mapped to an ordinal dimension, following the approach in Vehlow et al. (2011). We allow users to choose among different transfer functions that map bin values to heights or colors. In the case of histogram bar-heights, users can choose between linear, square-root or logarithmic mappings. In the case of heatmaps, we map each normalized data-value $0 \leq v_{in} < 1$ to a v_{out} value using $v_{out} = 1 - (1 - v_{in})^m$; tuning m allows either high or low values to be more visually distinguishable. An alternative mapping which highlights the central ranges, used, for instance, in Kincaid and Lam (2006), is based on the sigmoid function: $v_{out} = 2/(1 + e^{-mv_{in}}) - 1$. When generating heatmaps for ordinal data, we have used sequential color schemes, based on those found in ColorBrewer (Brewer, 2004). Users can choose from preselected color schemes such as white-to-blue, red-black-green, yellow-to-green, or white-to-red. For nominal data, we use ColorBrewer "qualitative" schemes. There is no scheme that remains visually distinguishable for large numbers of categories; beyond size 12, we cycle the colors but recommend that analysts bin the categories to avoid this.

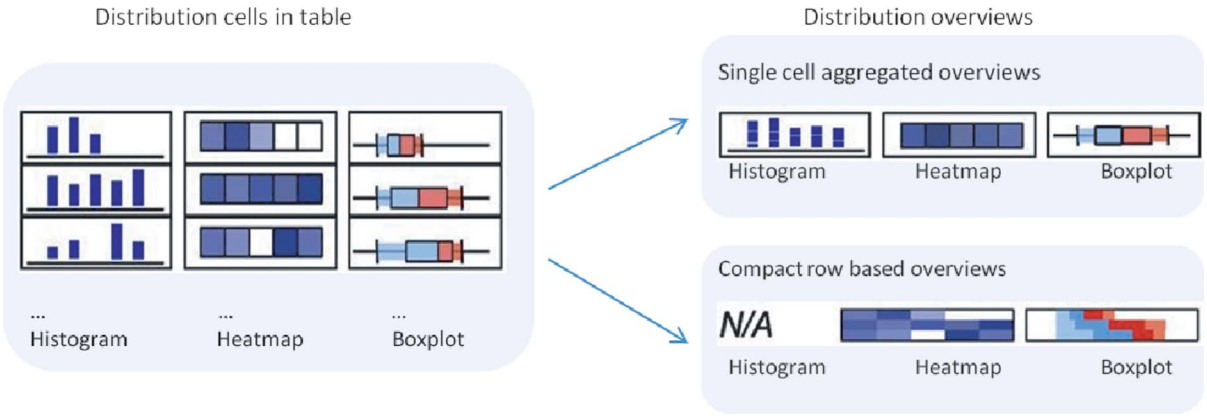

FIG. 2. Aggregated single-cell overviews and row-based overviews for a distribution column. *Note.* We encode distributions as histograms, heatmaps and boxplots. Multiple distributions can be merged to create single-cell overviews (top right), but important patterns can easily be lost in this process. Row-based stacked overviews (bottom-right) address this problem by squeezing visual encodings of each of the distribution cells vertically, at the cost of higher visual complexity (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

To guarantee that all rows are visible without overlaps, there should be at least one vertical view-row per data-row; otherwise, aggregation will be needed. If the number of vertical pixels is Y_p and the number of rows is Y_r, each vertical pixel will have to represent, on average, $\frac{Y_r}{Y_p}$ rows. When $Y_p > Y_r$, each data row will be present in several adjacent vertical pixels, posing no problem. When $Y_r > Y_p$, there are two main axes in the design space: which rows should be aggregated, and what values should be displayed for aggregated rows.

- To map data-rows to pixel-rows, we currently use Bresenham's line-drawing algorithm, where each data row r_i contributes to the "nearest" pixel row r_p, as calculated by interpolation. Although it would be possible to split data-rows over several pixel-rows, having rows with extreme values split over several pixels would make overviews very sensible to the exact scaling. This corresponds to a straightforward visual mapping. Semantic mapping is also possible: Although we assign exactly $\frac{Y_r}{Y_p} \pm 1$ rows per pixel, iHAT can assign arbitrary "similar" data-rows per view-row. TreeJuxtaposer (Munzner, Guimbretière, Tasiran, Zhang, & Zhou, 2003), a tree-comparison tool designed to allow visual comparison of massive hierarchies, goes one step further: Important (currently-selected) data-rows receive greater level-of-detail than other rows.
- To represent several data-rows within a single pixel-row, ManyNets lets users choose a descriptor for each distribution bin. Users can choose between maximum, minimum, and average bin-values (as the number of rows per pixel-row is variable, it is not a good idea to use simple bin-sums as values). Each of these will result in very different overviews for large $\frac{Y_r}{Y_p}$ ratios, allowing the discovery of different types of outliers and trends.

4. MANIPULATION OF STANDALONE DISTRIBUTION OVERVIEW

When building row-based overviews, the sorting of the rows is critical to bring patterns into focus; in addition, in large tables, row order governs which data rows will be mapped to a single-view row. This means that relevant detail may be affected by neighbors. We support three sources of row-sorting information:

- Using a distribution property, for example, bimodality, skewness, average, median, standard deviation, kurtosis, and minimum and maximum values. The sorting helps reveal patterns and outliers in the distribution data.
- Using similarity: either from all rows to an individually selected row or to generate a hierarchical clustering, which is then linearized to generate a sorting order. This feature is useful when users want to see possible grouping and similarity among the distributions.
- Using the row-order of the table it is based upon. This feature helps to identify if the distribution column has correlation with some other table column.

Because all bins use the same scale, the overall shape of the distribution also conveys important information. By looking at the sorted overview, users can understand the reasons for the similarity, which may not be obvious without visualization. To search and browse shape-related information, appropriate similarity-based pattern-matching methods are required. Similarity-based methods rely on the notion of distance (or its complement, similarity) between two distributions. We compare all the distributions to one another to compute their pairwise 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

Downloaded by [University Of Maryland] at 07:38 04 August 2014

unrelated. We implemented both nominal and ordinal distance metrics:

- Euclidean—useful for nominal and ordinal distributions; uses the Euclidean distance when considering each distribution as a vector (where each bin represents a dimension).
- MDPA—ordinal version of the algorithm described in Cha and Srihari (2002). Cannot be used for nominal values.
- Area—ordinal only; computes the area between two CDFs. Uses normalized distributions to make this comparison.
- KS—ordinal only; uses Kolmogorov–Smirnov distance, that is, the maximal distance between CDFs. Uses normalized distributions.

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 perform normalization (because CDFs should add up to 1). Normalization is related to local versus. global scaling (see section 3) and should be used consistently with it. Using Euclidean or MDPA metrics and displaying the results as heatmaps of normalized histograms will generally result in the clusters being undetectable. After initial experiments, we updated the interface so that, 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; clustering, on the other hand, 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 Bar-Joseph et al., (2001). 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 Elmqvist et al., (2008); this approach is faster, but ordering in the last rows tends to suffer.

4.1. Sorting and Clustering: Methods and Examples

We illustrate available sorting options using data sets from movie recommendation systems and phone call domains. Our movie recommendation data included distribution columns from two movie rating systems, FilmTrust (Golbeck & Hendler, 2006) and MovieLens (GroupLens Research Project). Movies receive multiple ratings from reviewers; by analyzing how movies are rated by various groups, users 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 learned just by looking at the aggregated single cell overview.

Sorting using distribution properties. This is mostly useful for ordinal distributions, as many of these concepts are not applicable for nominal distributions. Although 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 Figure 3). For further analysis, we select the relevant section of the overview. We filter the table to show only those movies, and examine their rating pattern in a separate detached overview. This could also have been accomplished by adding ”movie bimodality” as a sorting column to the main table, and then sorting the whole table by bimodality. The most controversial movie (highest value of bimodality) in the data set was ‘*Double Indemnity*’, but there are several close contenders for this title.

Sorting by similarity to a distribution. This feature helps users find distributions similar to a given one. The example depicted in Figure 4, contains movies categorized as “science-fiction” in MovieLens. We select a popular movie, *Lost World: Jurassic Park* (1997), in the main table and then sort the overview by similarity, using the Kolmogorov-Smirnov similarity metric, to identify other movies with similar ratings. The details and overview both show that these movies have a bell-shaped distribution of ratings.

Sorting by clustering. Trends are easier to spot if similar rows are grouped together and clustering the distributions facilitates that. After being clustered, the overview immediately shows the similar distributions forming groups in the overview. 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 (Figure 5). On the left are the students, and we see a small cluster of “highly critical” raters (i.e., with 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.

Sorting by clustering: Multicolumn overview. Multicolumn overviews can help reveal correlation between columns, by displaying overviews of multiple columns side by side. The sorting order of the first column is carried over to the rest. 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 1 to 10. 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 data set hypothesized that users that rate movies very highly would also assign higher trust ratings to their colleagues. Using our system, no such correlation is visible (see Figure 6). Although users often give high rating to movies, trust

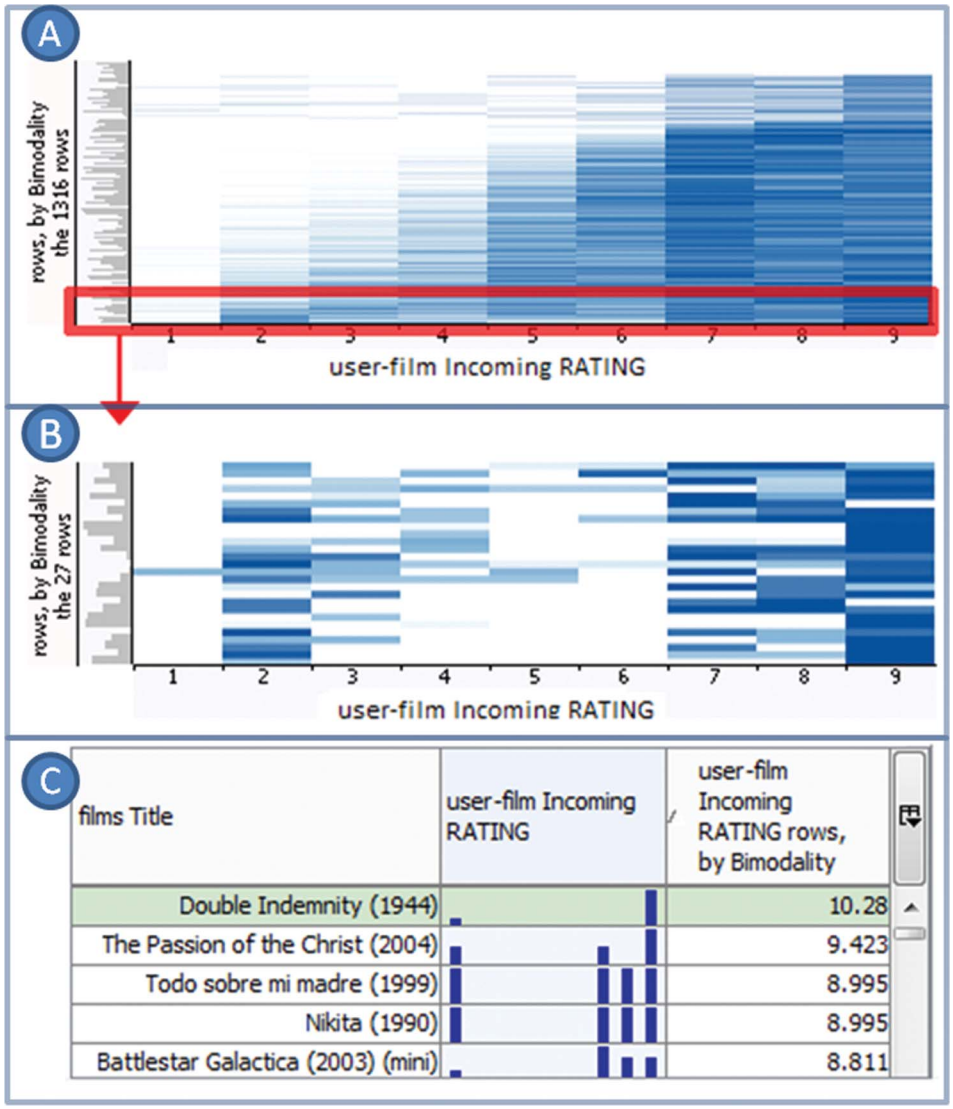

FIG. 3. A: Movie ratings in the FilmTrust dataset, sorted by bimodality. *Note*. At the very bottom of the overview are the movies with the highest bimodality: users either love them or hate them. These 27 highest bimodally rated movies are selected to create a separate heatmap overview as in B. In C: the portion of the filtered table containing only these movies (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

ratings tend to be more moderate. ManyNets allows 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.

External sorting. When sorting an overview according to the row-order in the table it is derived from, it will update its rendering whenever the table sorting changes. We can first sort the table rows using any standard table sorting method, such as sorting by the value of a selected column, and then apply the same sorting in the distribution overview. This is useful to observe correlation between that selected column and the distribution column that generated the overview.

The VAST 2008 Mini Challenge 3 data set (Grinstein et al., 2008) 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 Figure 7). Sorting the distribution of call destination IDs by start time, shows two different regions (see heatmap overview in Figure 7; alternative overviews are available in Figure 9). 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 8 to 10, represented by rows 71 to 99, shows a different pattern. Suddenly several destinations with higher IDs, including persons 308 and

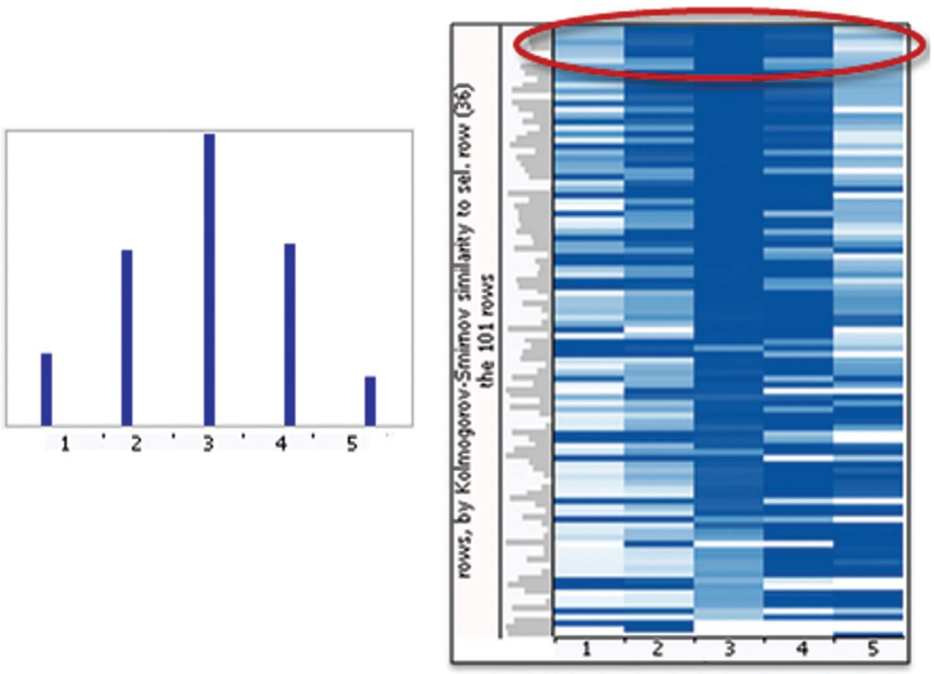

FIG. 4. Movies with rating distributions similar to that of *Jurassic Park* (marked with red oval). *Note*. The row for *Jurassic Park* is at the top of the overview. The histogram at the left shows the bell-shaped distribution of ratings for this movie (color figure available online).

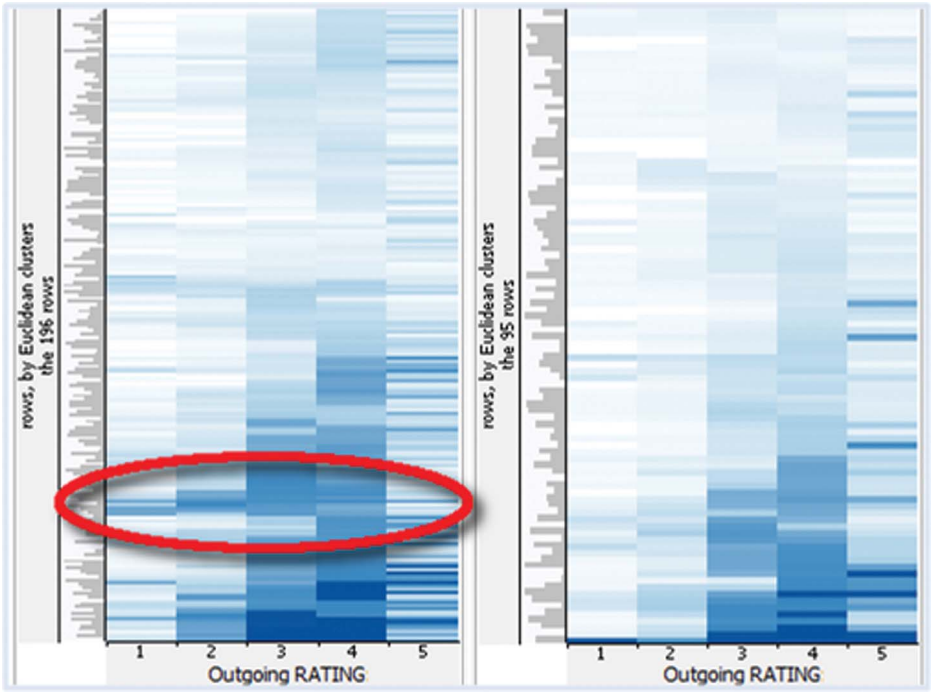

FIG. 5. Left: ratings from students; Right: ratings from educators. *Note*. In both cases the rating distributions are clustered using Euclidean distance as a distance metric. A cluster of low ratings by students is marked with red oval (color figure available online).

397, who were not specially active before, started to receive many 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, that is, the suspects had switched to different phone numbers to escape monitoring, but the structures of their ego networks remained unchanged.

Downloaded by [University Of Maryland] at 07:38 04 August 2014

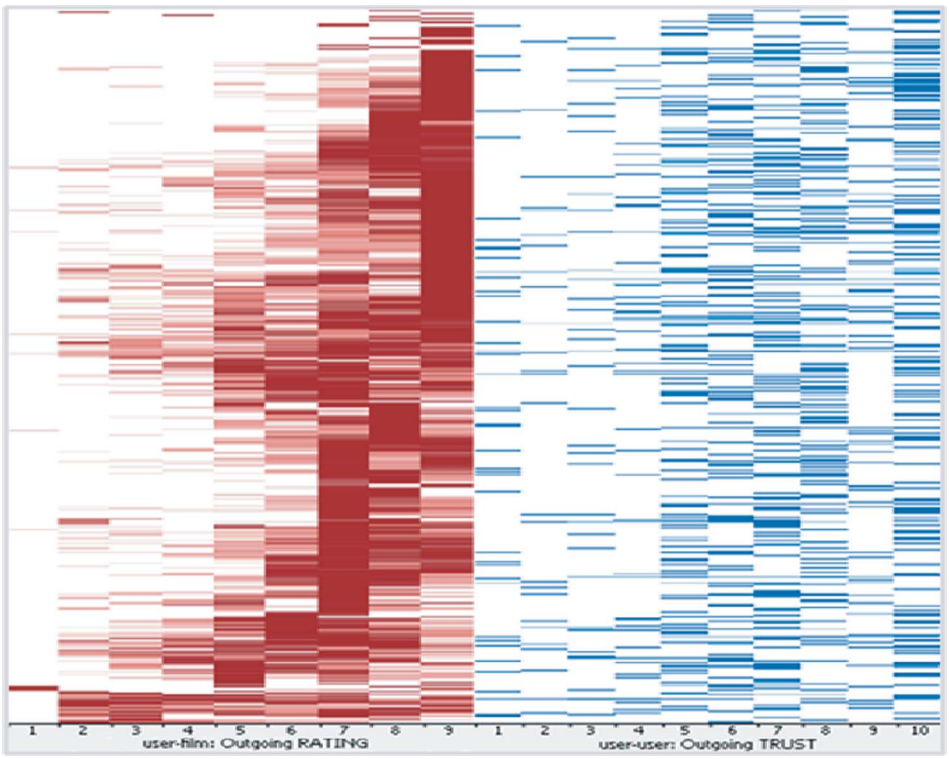

FIG. 6. Multicolumn overview: The left column is the distribution of ratings given by users to movies, the right column is the distribution of trust rating given by users to other users. *Note*. The first column is sorted by similarity; the second column uses the same order. At a glance, there appears to be no strong correlation between these variables (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

5. TABLE-OVERVIEW INTERACTION

Data sets are displayed as tables. Multiple tables can be displayed at once, and it is possible for some tables to be built out of aggregations of rows of other tables. For example, given a large collection of individual ratings of movies by users, we can aggregate this into a movie-centric table ("What ratings has this movie received?") or a user-centric table ("What ratings has this user made?"). In both of these examples, distribution columns arise naturally. This section describes how our tables interacts with their overviews.

5.1. Placement of Overviews

The top of each column presents a distribution overview of its contents, using the single-cell histogram style seen to the right of A or B in Figure 1. Whenever a cell on the table view is selected, a larger version of the corresponding column overview is displayed in a details-on-demand sidepane (see A in Figure 8). If several rows are selected, 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. Therefore, overviews are used in four roles: (a) at the top of each column; (b) at the sidepane, as a larger version (with additional controls to configure the column's representation); (c) as an overview that only includes values from currently-selected rows; and (d) detached.

5.2. View Coordination

When rows are selected, all nondetached overviews immediately echo the selection. Histogram bars highlight the relative contribution to each bin of selected rows, whereas row-based overviews highlight the selected rows themselves. When several data rows are mapped to the same overview row, the overview row will be selected as soon as one of its data rows is selected. The converse is also true: If overview rows are selected, all related data rows will be selected in the corresponding table. Choosing values from histograms results in all data rows with a nonzero value in the corresponding bin being selected. Therefore, overview-selection can be used to answer the question "What rows contribute to these values?" whereas table-selection can answer, "What values are contributed by these rows?" Therefore, whenever users select through the overview, the corresponding table rows are also selected and highlighted with different color, so they can go back to the table view and observe detailed information of the selected entries. For example, if a movie's rating distribution is selected from the overview, the table row for that movie is also highlighted, which contains all the original information about the movie title,

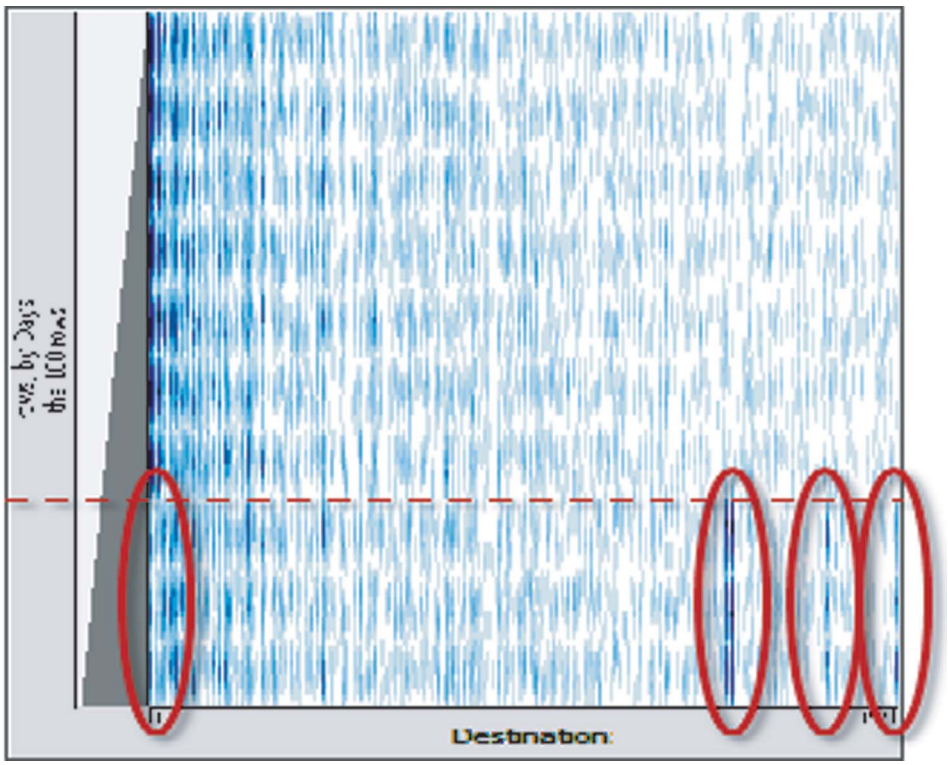

FIG. 7. Overview of destination column of VAST 2008 Mini Challenge 3 time-sliced telephone call data; after the first 7 days of calls (red dashed line), represented by 70 rows, several trends change (marked by red ovals). *Note*. Highly active low-ID destinations (leftmost oval) stop receiving calls, whereas previously inactive high-ID destinations (rightmost three ovals) start taking in calls. See Figure 9 for alternative overviews of this data set (color figure available online).

release year, genre, and so on. Selected data rows can quickly be filtered into a new table. The new table can contain either only previously selected rows or only previously unselected rows. This allows interesting subsets of the data to be isolated and explored. Alternatively, keeping the mouse pointer over an overview displays a pop-up that describes the data rows that correspond to that position. Both selection and hovering can be used to provide details-on-demand.

5.3. Sorting Tables

Row-based overviews can be sorted in more ways than the table columns themselves: Clicking on any of the table's headers sorts the corresponding column according to its 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 Figure 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 users can sort the pixel rows in the overview by the table and vice versa.

Because the ordering of row-based overviews can be set independently from that of the main table, ManyNets has a "graphical label" (see callout in Figure 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.

5.4. Configuration

Overviews can be configured in a settings panel in the details pane; the settings themselves are hidden unless requested. Figure 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. So if users choose to generate heatmap overview, first they need to configure how they want to sort the overview: sort by original table, sort by distribution properties (such as skewness, bimodality, etc.), sort by similarity and cluster by similarity are the available options. If they choose sort by similarity or clustering, they will be provided with another list of available similarity metrics from where they can choose how to calculate the similarity metrics for the distributions. The intensity scale is to vary the color intensity and contrast of the heatmaps. Tool-tips are available for all the configuration options, and incompatible options are highlighted with red borders.

Downloaded by [University Of Maryland] at 07:38 04 August 2014

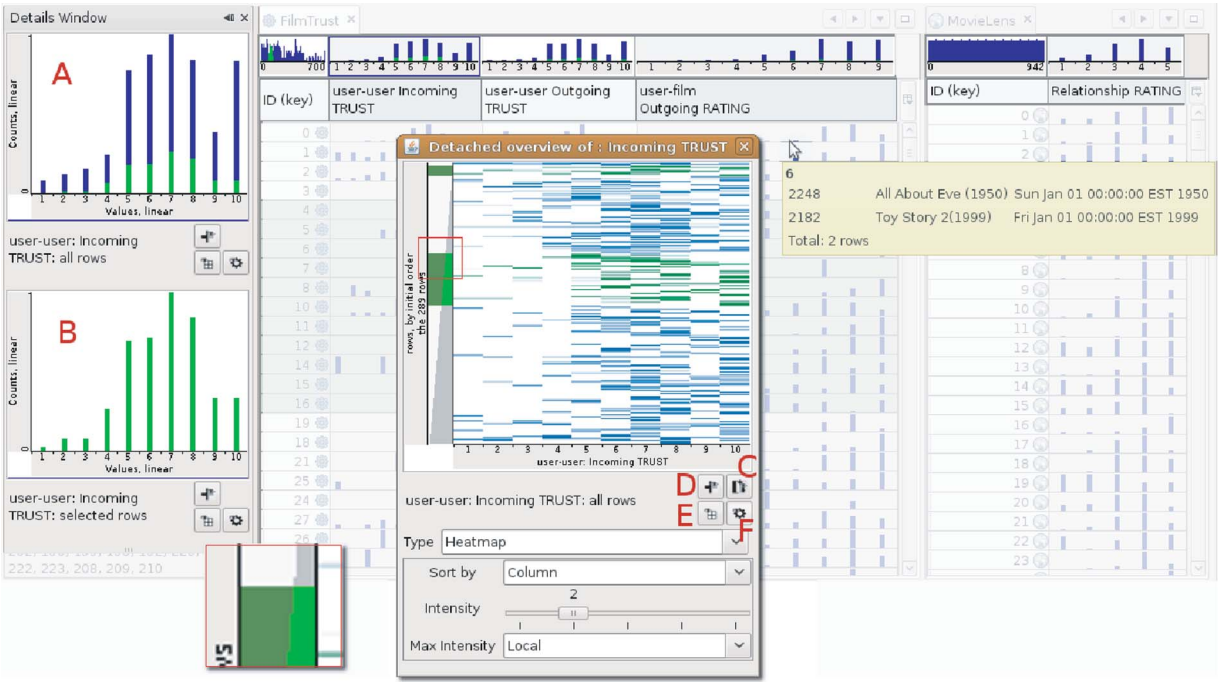

FIG. 8. General view, with current column overview (A) and current selection overview (B). *Note*. The options for the current overview are displayed (or hidden) by clicking on the options button (F). Button (C) adds a sorting column with the current sorting to the table. The overview can be detached 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 Figure 9. The zoomed region demonstrates graphical labels for overviews and selected-row highlighting in row-based distributions (color figure available online).

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 Figure 8). This brings up a dialog, displayed in Figure 9, with a set of alternative fully configured overviews appropriate for the current data-type; clicking on any of these options selects the corresponding settings. Because 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.

6. USABILITY STUDY

We believe that ManyNets is unique in supporting distribution data in a single cell and providing a very rich set of interaction techniques to manipulate and analyze distribution data. For example Microsoft Excel and Tableau (http://www.tableausoftware.com) can represent single column or groups of columns as heatmap, but these columns cannot be manipulated as a group. On the other hand, Spotfire provides clustering of distributions but it lacks features like distribution aware sorting, multicolumn overview, global vs local comparison of values, sorting by similarity, rich integration with table, etc. Hence, we have chosen an objective-based approach [Friedman and Wyatt, 1997]. In such an approach, certain reasonable objectives are defined for a new system, and the evaluation strives to demonstrate that these objectives have been achieved. Our goal in this study was to investigate if users could use the interface to obtain visualizations that allowed them to answer representative questions effectively and efficiently. We also wanted to observe the strategies that users chose, and the problems they encountered, gathering feedback and suggestions for further improvement.

6.1. Procedure

The data set included 3,018 records of census data of population age-distributions in U.S. counties (see Figure 10). Because analysts have very little availability, are hard to recruit for a user study, and the data used in the study are simple enough to be understood by students, the participants were 10 graduate students from various departments of the University of Maryland. None of them was a member of the development team.

Training consisted of reading a printed manual, watching a demonstration and interacting with the tool according to a predefined script (including examples of analyzing movie ratings distributions from the MovieLens data set), and finally answering two training questions. When the participants could answer the training questions correctly, they were considered ready to perform the first 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 hr, of which about 30 min consisted of training. Participants were encouraged to think aloud while performing the tasks. Observers recorded tasks-completion

Downloaded by [University Of Maryland] at 07:38 04 August 2014

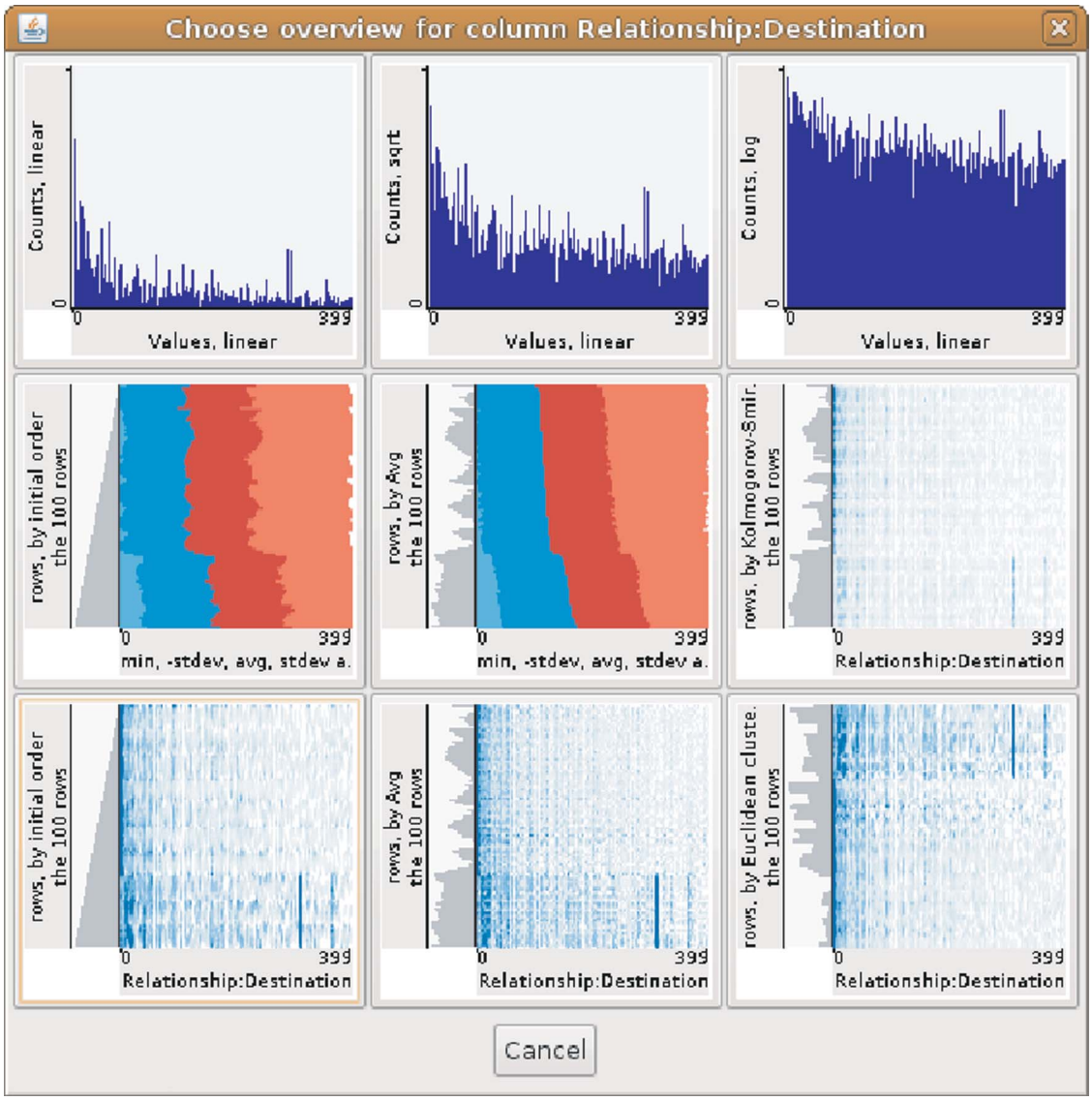

FIG. 9. Grid interface to visually select a fully configured overview. *Note*. One of the overviews was used to generate Figure 7 (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

times 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 had 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 an example, see task 6) and were encouraged to try again.

6.2. Tasks

Seven tasks were used (see Table 1) for this study, starting with Tasks 1 to 3 followed by Tasks 4 to 7 in random order. The tasks were designed to test the usability of the main features of the interface (as noted in parentheses in Table 1).

6.3. Results and Discussion

The participants were able to select the suitable features and to effectively interact with the tool in order to obtain the

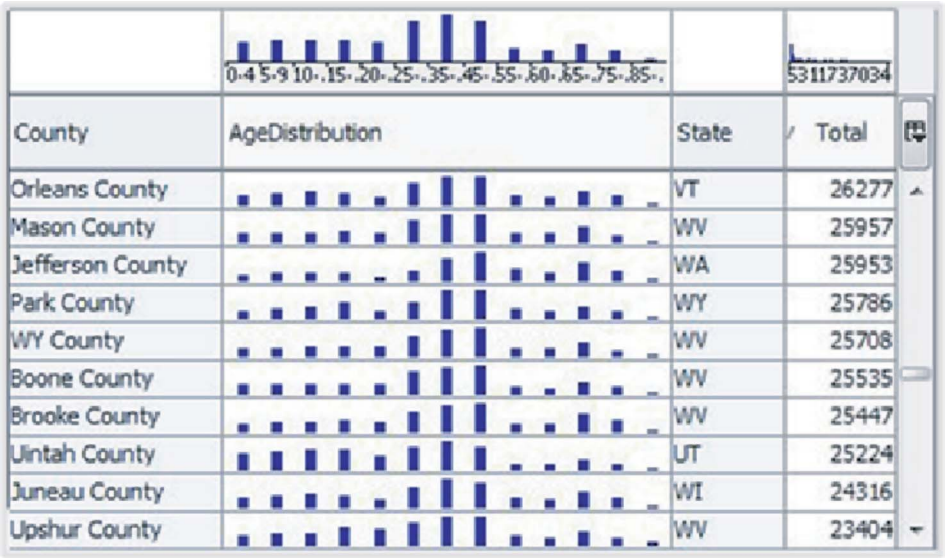

FIG. 10. A sample U.S. county-wise population table, where each row represents a county and the columns are, from left to right: U.S. county names, age distribution (distribution of population of different age groups), state, and total (total population counts). *Note.* The table is sorted by the Total column (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

correct visualizations. Only two participants needed an additional attempt to produce the multicolumn overview (Task 3). This was due to mistakenly clicking the "add/sort column" button (which is used to add an additional column to the main table sorted in the same order as the overview, not to add additional column to the multicolumn overview) instead of choosing the correct type of the overview (multicolumn). This suggested 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. 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. Moreover, participants requested access to more details about each row (beyond mouseover information), which is also provided in the main table. All of the participants were able to obtain the expected insights from the visualizations and provided accurate and full answers to the questions.

The $M \pm SD$ of the interaction and interpretation times of Tasks 1 to 7 are presented in the Table 1. For all tasks, participants were able to produce produce the required visualizations ($M \pm SD = 40 \pm 24$ s) and interpret the information so as to correctly answer the questions ($M \pm SD = 18 \pm 9$ s). However, there is a large difference in both interaction and interpretation times among the tasks; for example, the time for Task 3 was much longer than for the other tasks. Task 3 (multicolumn overview) was much more complex and required manipulating two columns and more interaction steps than the other tasks. In contrast, the interpretation times of Tasks 1 and 6 were much shorter than for the other tasks. Task 1 was indeed much simpler and required only the identification of the highest bin count in the aggregated histogram. The interpretation of Task 6 also required pointing out a row which had much higher intensity than the other rows. Users thinking aloud explored multiple ways of approaching the tasks, but most of them eventually found the right way. Users familiar with standard keyboard shortcuts were able to accomplish the interactive tasks much faster than others. For example, one of the participants was already familiar with netbeans platform (that we used to develop our tool), and its window manipulation features and took much less time to interact with the interface. This might explain the high standard deviations of the interaction times across tasks. 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" and "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".

Downloaded by [University Of Maryland] at 07:38 04 August 2014

TABLE 1
Task List and Time for the User Study

Tasks	What Participants Were Expected to Do	$M \pm SD$ (s): Interaction, Interpretation
1) Aggregated overview: Across all counties, people of what age range are most prevalent?	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) Sort using a distribution descriptor: In which counties is the population distribution extremely skewed toward youths and in which is it extremely skewed toward elder adults?	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 Figure 11.	30 ± 26, 20 ± 7.3
3) Sort by table column + multicolumn overview: Do counties from the same state exhibit a similar pattern? Does any state stand out as different?	Point out at least one conspicuous pattern in the overview, and use the tooltip to check the name of the State (e.g., 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 Figure 12.	65 ± 46, 25 ± 25
4) Sort by similarity to a distribution/Search by example: 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?	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) Clustering, local (KS or Area): Use the data set of the topmost 508 populated counties to identify at least three different groups of counties with similar distribution shapes	Cluster and then identify at least three groups (see the different clusters and describe the different patterns; e.g., high percentage of elderly residents, high percentage children, high percentage of middle-age residents, etc.). See Figure 13.	33 ± 19, 27 ± 11
6) Clustering, Euclidean or MDPA: 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.	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 reexplained 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 Figure 14.	35 ± 13, 9 ± 5
7) Comparing overviews side by side: 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.	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 Figure 15.	55 ± 23, 21 ± 7

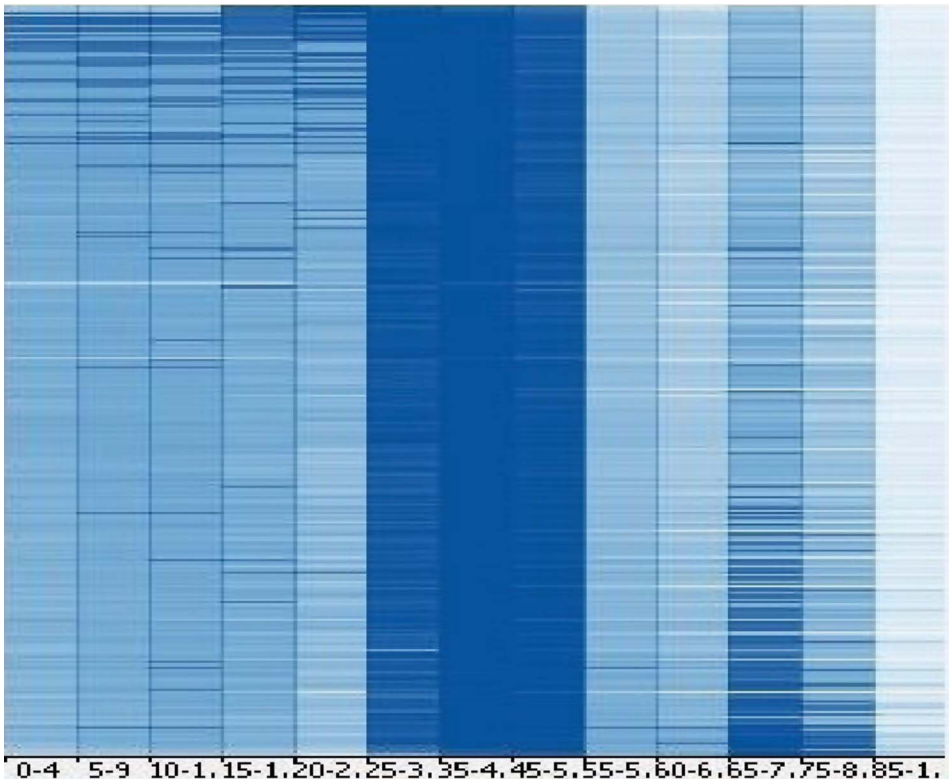

FIG. 11. Task 2: Sorting according to skewness of the age distributions of all U.S. counties (color figure available online).

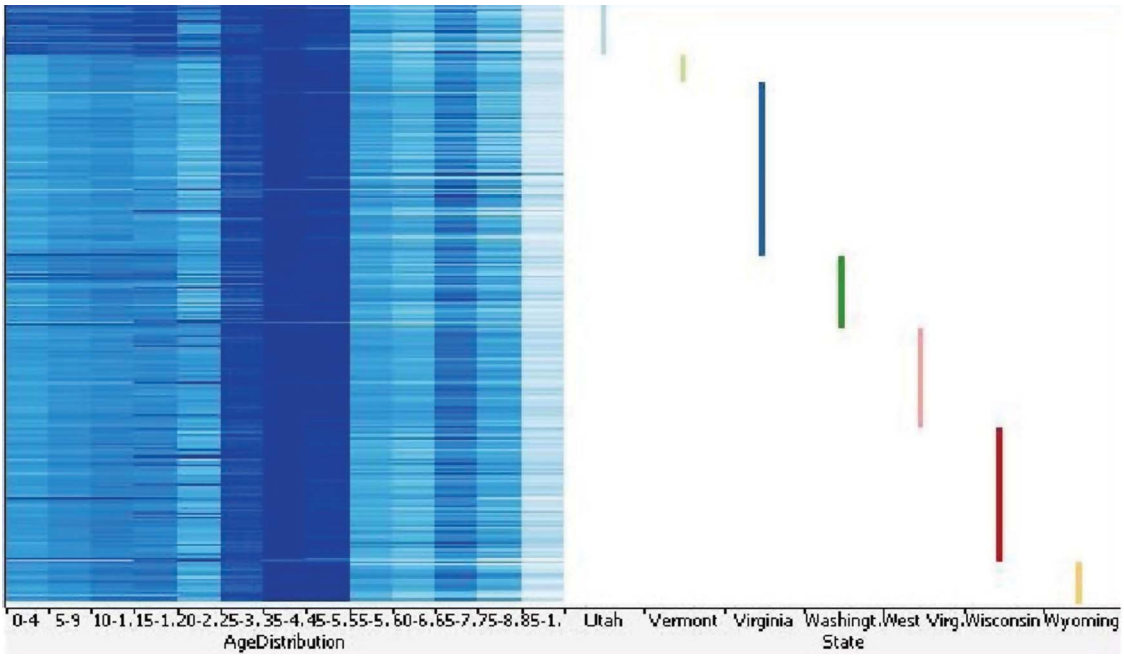

FIG. 12. Task 3: Multicolumn overview sorted by states. *Note.* The AgeDistribution column and the categorical State column are displayed side by side. Each state is presented by different color. At the top, we can observe that the counties from Utah stand out from the rest; their population is younger than that of other states (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

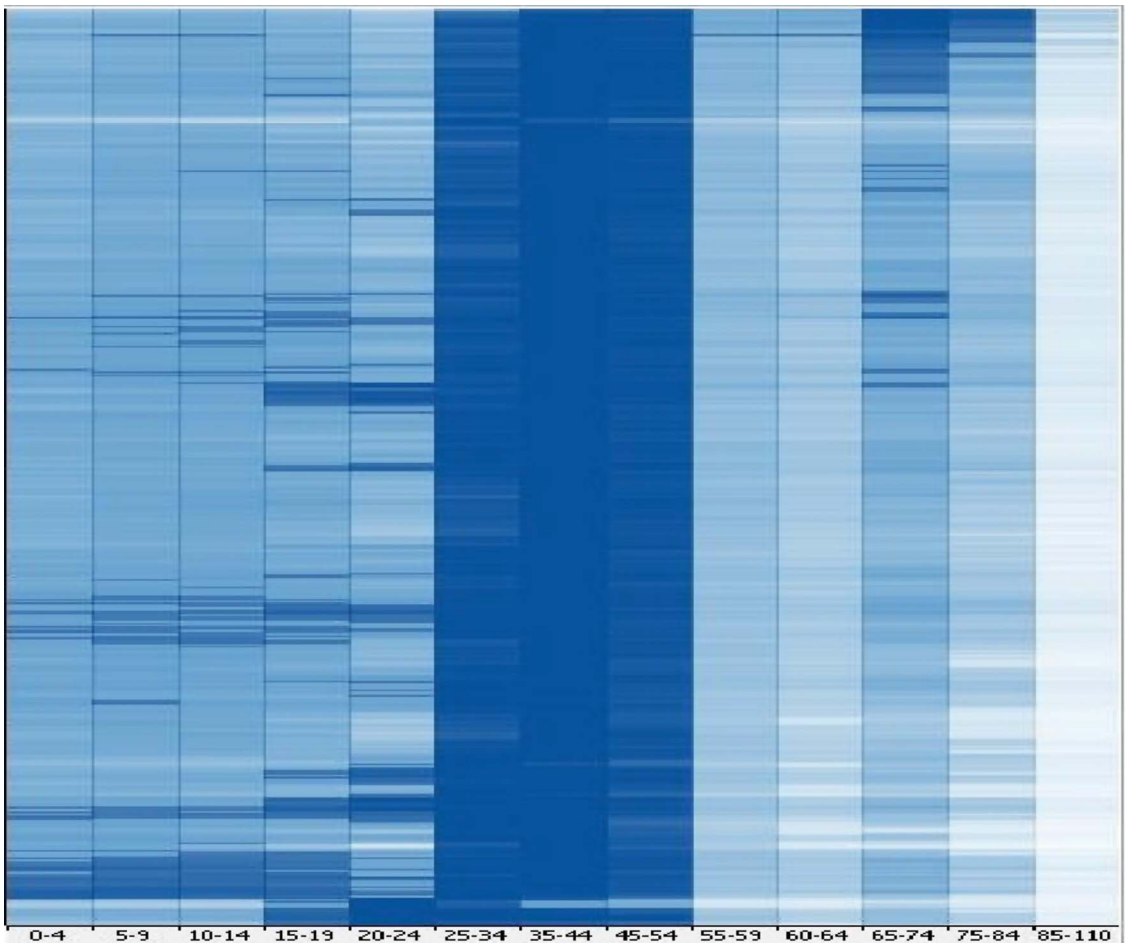

FIG. 13. Task 5: Cluster age-distribution column to compare shapes of distributions, using the Kolmogorov-Smirnov metric (color figure available online).

Downloaded by [University Of Maryland] at 07:38 04 August 2014

Suggestions for improvement included, "Sometimes I wish things could 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." Participants also requested access to more details about each row (beyond mouseover information).

In summary, the study demonstrated that after a short training period, participants were able to answer representative questions using the tool, at high level of accuracy and within a reasonable period. Using the overview interface, participants were able to effectively and efficiently produce overviews of the distribution data and to use the sorting options to discover distinctive patterns, clusters, and outliers, pointing out global trends and relationships between columns.

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. ManyNets 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 good code modularity. Having distributions spread over multiple columns does not let the users operate on them as a group, whereas the ManyNets interface handles distribution data as a single column and makes it possible to manipulate them all together considering their distribution specific properties. Visual overviews enable users to see patterns, similarities, and outliers. Moreover, the coordination with the table provides understanding of correlation between columns and the option to filter and select subset of data easily.

This work analyzes several aspects of distribution overviews, including generation, sorting, and clustering. Similarity-based ordering in general and clustering in particular are specially important, because statistical properties (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

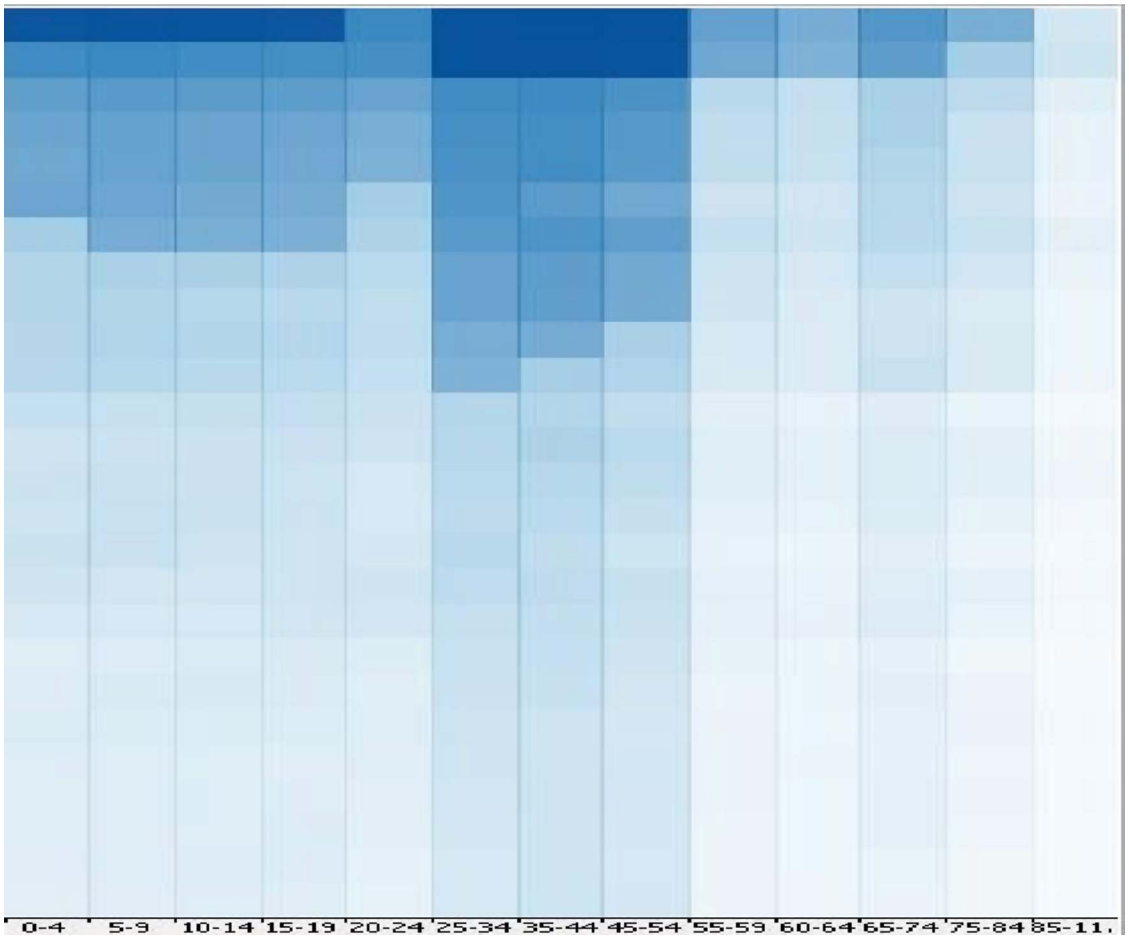

FIG. 14. Task 6: Cluster age-distributions to compare bin-count values of distributions, using the Euclidean distance similarity metric (color figure available online).

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 a context-sensitive grid display of recommended settings.

We illustrate our approach using examples drawn from the domains of recommender systems, VAST 2008 MiniChallenge 3 phone call data, and U.S. Census 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 data set is clearly visible in our overview. The mere calculation of the statistical metrics (e.g., standard deviation, skewness, etc.) of the distributions and sorting them according to those metrics cannot provide the users sufficient understanding on the overall distributions. Presenting them as a visual overview can help users see the patterns and the shifts. The usability study also confirms that the participants not only were able to spot the clusters and the outliers, but also could understand, among all the distributions in a column, why and where the similarity and the differences are. For example in Task 7 they could spot easily that several counties from Florida have more younger population than people of other age range by looking at the clustered distribution overview. For Task 4 they could find similar patterns and interpret why they were similar. Only sorting the distribution data and presenting them as a sorted list would provide the similarity measure but would not generate the understanding about why they are similar; our overview facilitates this understanding.

The results of the usability study suggest that the distribution overview interface is learnable in a short period and its functionality is beneficial to provide overviews of the distribution data, which facilitates discoveries of distinctive patterns, clusters, and outliers. In addition, our interface supports exploration of global trends and relationships between columns. The usability study also provided valuable interface improvement suggestions. We improved ManyNets by changing the button labels to make them unambiguous. We made it possible that global and local scaling would change automatically depending on the selected similarity metric; for example, choosing MDPA would make the scaling global and choosing Area metric would

Downloaded by [University Of Maryland] at 07:38 04 August 2014

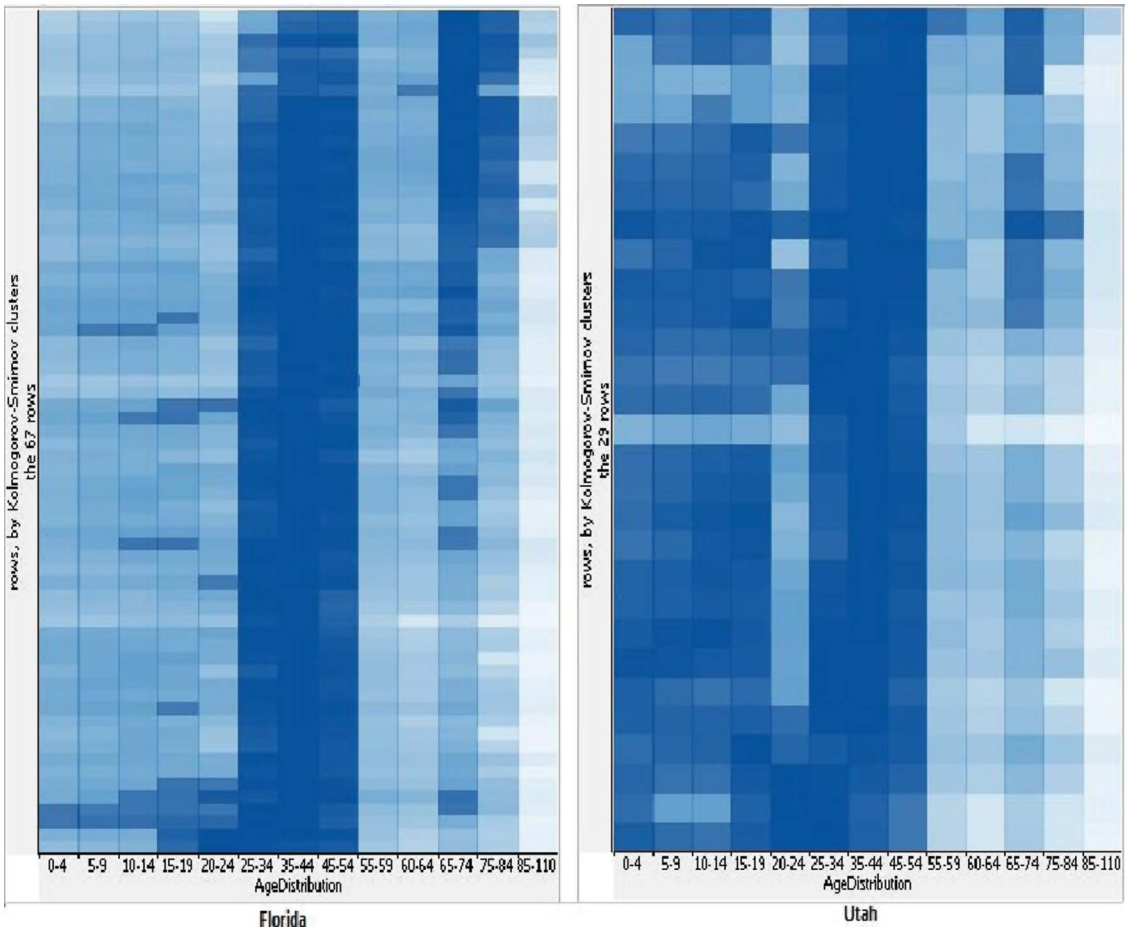

FIG. 15. Task 7: Side-by-side comparison of distributions from two sets of counties after clustering, here age distribution of Florida (FL) and Utah (UT); both overviews have been clustered using the Kolmogorov-Smirnov similarity metric (color figure available online).

make it local. Even though details about the rows are in the main table view, participants suggested more details in the tooltip. Participants also wanted to annotate interesting regions in the overview, annotate the clusters, and save the image to share with others.

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

REFERENCES

Bar-Joseph, Z., Gifford, D. K., & Jaakkola, T. (2001). Fast optimal leaf ordering for hierarchical clustering. pages 22–29.

Brewer, C. A. (2004). Color research applications in mapping and visualization. In *Color Imaging Conference*, (pp. 1–3). The Society for Imaging Science and Technology.

Cha, S.-H., & Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, *35*, 1355–1370.

Chang, R., Ghoniem, M., Kosara, R., Ribarsky, W., Yang, J., Suma, E., Sudjianto, A. (2007). Wirevis: Visualization of categorical, time-varying data from financial transactions. *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, 155–162.

Elmqvist, N., Do, T.-N., Goodell, H., Henry, N., & Fekete, J.-D. (2008). ZAME: Interactive large-scale graph visualization. 215–222.

Freire, M., Plaisant, C., Shneiderman, B., & Golbeck, J. (2010). ManyNets: An interface for multiple network analysis and visualization. *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI)*. pp. 213–222.

Friedman, C., & Wyatt, J. (1997). *Evaluation methods in medical informatics*. Springer.

Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., & Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, *5*, R80.

Golbeck, J., & Hendler, J. (2006). Filmtrust: Movie recommendations using trust in web-based social networks. *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference CCNC 2006*, 282–286.

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

GroupLens Research Project. Movielens 100K Dataset. Retrieved from http://www.grouplens.org/node/73.

Henry, N., Goodell, H., Elmqvist, N., & Fekete, J.-D. (2007). 20 years of four HCI conferences: A visual exploration. *International Journal of Human–Computer Interaction*, *23*, 239–285.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multi-dimensional geometry. *Proceedings of the 1st Conference on Visualization'90*, 378.

Downloaded by [University Of Maryland] at 07:38 04 August 2014

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

Kincaid, R. (2004). Vistaclara: An interactive visualization for exploratory analysis of DNA microarrays. *Proceedings of the 2004 ACM Symposium on Applied Computing*, 167–174.

Kincaid, R., & Lam, H. (2006). Line graph explorer: Scalable display of line graphs using focus+context. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces*, 404–411.

Kobsa, A. (2001). An empirical comparison of three commercial information visualization systems. *INFOVIS*, 123–130.

Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., & Zhou, Y. (2003). Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, 453–462.

Rao, R., & Card, S. K. (1994). The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. *CHI '94: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 318–322.

Seo, J., & Gordish-Dressman, H. (2007). Exploratory data analysis with categorical variables: An improved rank-by-feature framework and a case study. *International Journal of Human–Computer Interaction*, *23*, 287–314.

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

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

Stricker, M. A., & Orengo, M. (1995). Similarity of color images. pages 381–392.

Vehlow, C., Heinrich, J., Battke, F., Weiskopf, D., & Nieselt, K. (2011). ihat: Interactive hierarchical aggregation table. *IEEE Symposium on Biological Data Visualization*, 63–69.

Yi, J. S., Elmqvist, N., & Lee, S. (2010). Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, *26*, 1031–1051.

ABOUT THE AUTHORS

Awalin Sopan is a Ph.D. student at the department of Computer Science of the University of Maryland from where she received her master's in Computer Science in 2011. She is a research assistant at the Human–Computer Interaction Lab. Her research interest is in Information Visualization and Social Network Analysis.

Manuel Freier's interests are in information visualization, authoring, e-learning, and plagiarism detection. In May 2009 he received Fulbright scholarship to work as a postdoctoral researcher at the University of Maryland's Human–Computer Interaction Lab, where he developed ManyNets. As of October 2010, he is an Assistant Professor at the Universidad Complutense de Madrid, Spain.

Meirav Taieb-Maimon is a Lecturer at the Department of Information Systems Engineering in Ben-Gurion University. She received her M.Sc. (Summa Cum Laude) and Ph.D. in Industrial Engineering and Management from Ben-Gurion University. Her research interests include human factors, human–computer interaction, information visualization and evaluation of information systems, interfaces and visual analytics systems.

Catherine Plaisant is a senior research scientist at the Human–Computer Interaction Lab of the University of Maryland. Her research covers various topics such as information visualization, evaluation methodologies, digital libraries, help, digital humanities, and so on. She coauthored with Ben Shneiderman the 4th and 5th editions of *Designing the User Interface.*

Jennifer Golbeck is the director of Human–Computer Interaction Lab and Assistant Professor of College of Information Studies in University of Maryland. Her core research interests are in understanding how people use social media to improve the way they interact with information.

Ben Shneiderman is a Professor in the Department of Computer Science, Founding Director of the Human–Computer Interaction Laboratory. He is a member of the National Academy of Engineering. He was elected as a Fellow of the ACM, a Fellow of IEEE, and a Fellow of the AAAS.

Downloaded by [University Of Maryland] at 07:38 04 August 2014