Star Plot Visualization of Ultrahigh Dimensional Multivariate Data

Shabana Sangli1, Gurminder Kaur1 and Bijaya B. Karki12,3
1School of Electrical Engineering and Computer Science, 2Department of Geology and Geophysics, 3Center for Computation and Technology, Louisiana State University
Baton Rouge, USA

Abstract - Visualization-based analysis of multivariate data suffers from a high degree of clutter when the number of dimensions (variables) becomes too large. Here, we extend the standard star plot technique to visualize large datasets of ultrahigh number of dimensions by a) drawing overlapped star plots with one star per data item, b) shifting the origins of radial axes away from the central point to open space in the low-value scale, and c) dynamically partitioning the dimensions into groups and mapping them to different concentric circular regions to provide multilevel star plot visualization. Our test on multivariate datasets of high dimensionality suggests that the proposed extensions with appropriate interaction options can handle large number of dimensions of potential relevance to big data analytics.

Keywords: Multivariate data; star plot; information visualization; big data analytics

1 Introduction

Analysis of multivariate data poses tremendous challenge not only when the size of the dataset increases but also when the number of variables or attributes (also referred to as dimensions) increases. Such high-dimensional multivariate data perhaps are one of the most important contributors to today’s big data problems [1,2]. Extracting useful information from large, complex datasets is too difficult by directly looking at the data presented in a tabular form and/or simply using standard query languages. Visualization aims to take maximal advantage of human perception capabilities by graphically mapping a given dataset in its entirety, irrespective of its size, to a visual form for gaining insight into the data [3]. The visualization output though it is likely cluttered may still convey some information about the overall nature and structure of the data. When used interactively, visualization can provide us with a qualitative overview of large and abstract datasets thereby helping us quickly search for interesting features such as patterns, trends, anomalies, relationships, and clusters [4,5]. These features in turn can serve as basis for further analysis, which usually involves more focused (quantitative) explorations of selected data regions or variables. The goal is thus to find a visual representation of a given big data problem with a minimum clutter possible. So, the user can actually view the data as a whole to make some sense out of the display and then further dig into the data. This process may be referred to as an exploratory visualization-based analysis.

Several visualization techniques currently exist for analyzing multivariate data, which involve three or more quantitative variables (dimensions). Popular examples include star plot [6], parallel coordinates [7,8], star coordinates [9], scatterplot matrix [10], etc. In principle, these techniques should work for dataset of arbitrary size and arbitrary number of dimensions. However, there is still a need for assessing the potential of effectively using these techniques in visualization of very large datasets and big data. This is true in the case of star plot, which is a simple, widely used method for multivariate data visualization. The star plot method maps an n-dimensional data space into a 2-dimensional display space by representing n (≥ 3) variables on axes starting from the same central point [6,11]. In this mapping, each data item is displayed as a star-shaped icon in which radial spoke represents the value for a variable. All icons, one for each data observation, are usually displayed in a rectangular arrangement on a single page or screen. This visual presentation is effective and mostly used for small- to moderate-sized datasets perhaps containing no more than one hundred observations for no more than a couple of dozens of variables [6]. As the dimensionality and size of data increase, star plot visualization becomes increasingly overwhelming and the individual stars eventually become too small to represent recognizable shapes.

In this paper, we discuss the weaknesses of the classic star plot visualization technique for large multivariate datasets and mainly deal with ultra-high dimensionality of the data. In the situations involving a large number of variables/attributes, the radial axes are very closely packed and the star plot display becomes too cluttered. We propose different ways of adopting/extending the star plot method to analyze and understand large, complex multivariate datasets. We call these approaches as overlapped star plot, shifted origin star plot, and multilevel star plot visualization techniques. Tested on different multivariate datasets, the proposed extensions enable us to get a better visual display of data in their entirety by effectively mapping an ultrahigh number of dimensions as radial axes and rendering all data values for these dimensions on a given display plane.

2 Related work and motivation

The star plot is a multivariate data visualization method, which represents each data item as a star icon consisting of a sequence of equi-angular spokes (called radii) with each spoke/ray representing one of the variables [6,11]. It is also called a radar chart or web chart or circular parallel coordinates [12]. The star plot for single observation can provide information about relative dominance of a variable with respect to other variables in the observation. A set of multiple star plots helps identify the similarities in the observations and form clusters. The multi-plot also helps in the detection of outliers and anomalies. With these capabilities, the star plot enables the user to comprehend the data and extract useful information in a simple way, thus serving as one of the most useful visualization techniques.
However, the star plot technique falls short when the number of variables or dimensions becomes too large, needing its further improvement as we show in this paper. The human eye can perceive and understand the information conveyed by a visual display more easily when the display is presented in a bigger size. Instead of plotting every data item as a rather small star at a separate location, it is better to map all data items together and display all stars at the same location. This approach has been previously considered [13,14], but here we further emphasize its essence.

The star plot technique treats dimensions uniformly as in other radial methods of information visualization [6,12,14]. As the number of dimensions increases, it allows more compaction to accommodate all dimensions in display plane of fixed size. When data values are plotted on a compact axial layout, the result obviously is a cluttered visualization. To minimize the clutter, we consider two approaches: First, if the numbers of data items and dimensions are excessive near the origin, we can widen space between successive axes in their lower value ends. Second, to handle a very high number of dimensions, we can divide the dimensions into multiple groups and map them in different regions (levels), which are concentric circular areas. Finally, each approach is expected to work more effectively when appropriate interaction options are supported, as is the case with any multivariate data visualization [5].

3 Test datasets and implementation

Three datasets used for illustrating and testing the proposed extensions of the star plot method were taken from the UCI Machine Learning Repository [15]. The first is the data about 21 attributes of a car. The second US census 1990 dataset involves 68 variables of both numerical and categorical type. The third is the data about hand movements for LIBRAS (the official Brazilian signal language) with 91 variables representing the coordinates of the movement. Further details about the scaling and coding of the variable values in these datasets can be obtained from the repository [15]. Here, we provide some information for the “cars” dataset in Table I, which consists of 7 categorical variables (numbered as 0, 1, 2, 3, 4, 10 and 20) and the remaining 14 numerical variables.

The proposed improvements on the star plot visualization method were implemented using C++ with OpenGL and glut libraries for graphics rendering. A few interaction options were implemented to assess how visualization can be further enhanced. They include options for highlighting one or more data items of interest, adjusting the number of dimensions to be mapped to each display ring (circular region), and changing the boundaries between the display rings.

4 Proposed star plot extensions

A major concern with star plot is its limited ability to convey information under situations when the number of data items increases, when the values lie near the low end of the rays of the star plot, and when the dimensionality is too high. Therefore, we seek to adopt and further develop the star plot technique to visualize and explore large multivariate datasets as described in this paper.

4.1 Overlapped star plot

The star plot method displays a given dataset of \( N \) items as an array of \( N \) stars, one for each data item of \( k \) dimensions (or variables). In each plot, the values of a data point of the dimensions \( d_1, d_2, \ldots, d_k \) are linearly scaled from the fixed origin \( O_x, O_y \) and mapped at values \( v_1, v_2, \ldots, v_k \) on the \( k \) radial axes. The intersection points on these axes are then joined together to form a closed polylines of star shape, which graphically depicts the corresponding data item. These individual stars drawn separately need to be shrunk to accommodate all of them in the available display space, so that it is increasingly hard to view and comprehend these small stars. Here, we consider displaying all the data items together at the same position using the same radial axes instead of mapping each data item as a separately located star icon. This overlapped star plot approach uses the entire space for displaying each data point. In Fig. 1 (top), all stars are drawn together in the same large space for the cars dataset so they are visible (except some clutter caused by overlapping) and easy to compare.

<table>
<thead>
<tr>
<th>Variable Number</th>
<th>Values of the variable with their numerical scaled equivalent used. The appropriate values of ( \min ) and ( \max ) were used for scale.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fuel-type: diesel=1, gas=2 ; ((\min=0, \max=2))</td>
</tr>
<tr>
<td>1</td>
<td>Aspiration: std = 1, turbo = 2</td>
</tr>
<tr>
<td>2</td>
<td>Number-of-doors: two = 2, four = 4</td>
</tr>
<tr>
<td>3</td>
<td>Body-style: convertible = 1, hatchback = 2, sedan = 3, wagon = 4, hardtop = 5</td>
</tr>
<tr>
<td>4</td>
<td>Engine-location: front = 1, rear = 2</td>
</tr>
<tr>
<td>5</td>
<td>Wheel-base: 86.6 to 108.0</td>
</tr>
<tr>
<td>6</td>
<td>Length: 141.1 to 192.7</td>
</tr>
<tr>
<td>7</td>
<td>Width: 60.3 to 71.4</td>
</tr>
<tr>
<td>8</td>
<td>Height: 47.8 to 59.8</td>
</tr>
<tr>
<td>9</td>
<td>Curb-weight: 1488 to 3296</td>
</tr>
<tr>
<td>10</td>
<td>Number-of-cylinders: two = 2, three = 3, four = 4, five = 5, six = 6</td>
</tr>
<tr>
<td>11</td>
<td>Engine-size: 61 to 181</td>
</tr>
<tr>
<td>12</td>
<td>Bore: 2.91 to 3.94</td>
</tr>
<tr>
<td>13</td>
<td>Stroke: 2.19 to 3.90</td>
</tr>
<tr>
<td>14</td>
<td>Compression-ratio: 7 to 21.9</td>
</tr>
<tr>
<td>15</td>
<td>Horsepower: 48 to 200</td>
</tr>
<tr>
<td>16</td>
<td>Peak-rpm: 4150 to 6000</td>
</tr>
<tr>
<td>17</td>
<td>City-mpg: 17 to 49</td>
</tr>
<tr>
<td>18</td>
<td>Highway-mpg: 20 to 54</td>
</tr>
<tr>
<td>19</td>
<td>Price: 5151 to 23875</td>
</tr>
<tr>
<td>20</td>
<td>Make: alfa-romero = 1, audi = 2, bmw = 3, chevrolet = 4, dodge = 5, honda = 6, porshe = 7, benz = 8, mitsubishi = 9, nissan = 10, peugeot = 11</td>
</tr>
</tbody>
</table>
For a uniform radial layout of $k$ dimensions, the angle between the neighboring axes is $360^\circ/k$, which becomes too small when the dimensionality is too high. It is desirable to have a large interaxial angle. For instance, an angle of 30° that corresponds to one dozen variables is generally preferred. To have such wide-diverging axes, we can use one of half of the total space (i.e., 180° angular region) to display a few selected dimensions, say 6 axes. The remaining $k$-6 axes are then packed in other half space. This non-uniform axial layout shown in Fig. 1 (bottom) for the cars dataset helps display data distributions and relationships for selected dimensions more clearly (by giving a zoom-in like view) while still accommodating the remaining dimensions. The total display area can be split in two halves horizontally, vertically, or perhaps at any orientation in an interactive manner.

### 4.2 Shifted origin star plot

In the normal star plot technique, data values are mapped onto the radii from the fixed center point ($O_x, O_y$). When many dimensions are represented by closely packed radial axes emanating from the same origin (Fig. 2, top), high degree of crowdedness occurs towards the lower ends of the plot thereby making the data lines hardly visible. To overcome this problem, we propose the *shifted origin star plot* technique, in which the data values ($d_1, d_2, \ldots, d_k$) are scaled from different origin points ($O_1, O_2, \ldots, O_k$), where each dimension $d_i$ has its own origin $O_i$ which is at some shifted

![Fig. 1. Visualization of cars data set containing 21 variables using the overlapped star plot with uniform (top) and non-uniform (bottom) axial layout.](image1)

![Fig. 2. Layout of 68 axes for the census dataset using single origin (top) and shifted origins (bottom).](image2)
distance from the center. This radial outward shift opens extra space between the successive axes at their lower ends thereby improving visibility even when data points are crowded toward the origin.

Before laying out the axes and finding data locations along each radial axis, we need to calculate the shift amount \( l \), which depends on the total number of dimensions \( k \) and the maximum scale \( L \) available for plotting:

\[
l = (k/360)L
\]

As \( k \) increases, \( l \) increases. However, the shift distance cannot be arbitrarily large otherwise the inner void space eventually covers the entire display area. We constrain the shift distance to be the maximum of \( l \) and \( 0.5L \). Fig. 2 (bottom) illustrates the shifted origin star plot using a shift of \( 0.2L \) for the census dataset. The user can adjust the size of the circular perimeter containing all variable origins by dragging it inward or outward in an interactive manner.

4.3 Multilevel Star Plot

As the number of dimensions increases, effectively using available fixed display area to accommodate all dimensions without obscuring cognitive information about the data becomes even more challenging. The shifted origin star plot method discussed above may not be good enough because of high visual clutter arising from closely packed radial axes and connecting data lines. A possible solution is to reduce mapping of radial lines (axes) in a given area as much as possible. Note that this shifted origin star plot creates a void circular space around the center (Fig. 2, bottom). It is good idea to not waste this space. We can use the space for representing a subset of the dimensions and displaying the corresponding data values. We now divide the total dimensions into two groups and map them at two levels (Fig. 3, top), with the selected (fewer) dimensions assigned to the inner region (first level) and the remaining (majority) dimensions assigned to the outer region (second level). Thus, two stars of which the larger one lies in the outer region and the smaller one lies in the inner region together display each data item. We can extend this idea to multiple levels in which different groups of dimensions are represented in different concentric rings that constrain the respective display regions out of the total space. Our multilevel star plot thus manages many dimensions in different subsets thereby mapping the axes and displaying the corresponding data values in multiple levels (regions) instead of plotting all together in a single region as one star icon. The result is that a star displaying each data item is now split into multiple stars of different sizes drawn in different regions.

Before laying out the radial axes and calculating data locations along each axis, a decision is to be made on how many dimensions appear at each level. Let us consider a scenario in which the original large circular display space is divided into concentric rings of equal concentric radii. Obviously, the area of the outer ring is larger than the ring below it. So, it makes sense that a higher-level (outer ring) plot be assigned more dimensions than a lower-level (inner ring) plot. The number of dimensions assigned to the \( i^{th} \) level depends on the total number of dimensions \( k \), the total number of plot levels \( m \), and the level of plotting \( N_i \). We require that the number of dimensions represented in a level be proportional to the area of the corresponding ring. For the \( i^{th} \) ring, the area is given by

\[
A_i = \pi r_i^2 - \pi (r_{i-1})^2 = (i + i - 1)\pi r^2,
\]

where \( r \) represents the concentric radii. The number of dimensions \( D_i \) can be thus calculated as:

\[
D_i = \frac{k}{N_m^2} (N_i + N_{i-1}) \text{, where } i = 1, 2, ..., m
\]

Here, \( N_i \) represents the \( i^{th} \) level. Fig. 3 (top) shows the star plot with 2 shift levels \( N_m = 2 \) for the 91-dimensional LIBRAS dataset where \( 1/4^{th} \) of the total dimensions are assigned to the inner circle and the remaining \( 3/4^{th} \) to the outer ring. Fig. 3 (bottom) shows a similar plot with 3 shift levels \( N_m = 3 \) for the same dataset with approximately \( 1/9^{th} \) dimensions in the inner circle (first level), \( 3/9^{th} \) dimensions in the middle ring (second level), and the remaining \( 5/9^{th} \) dimensions in the outer ring (third level).
5 Visualization-based analysis

We now present some analysis of the proposed star plot extensions on two high-dimensional multivariate datasets, namely the US census data with 68 variables and the LIBRAS movement data with 91 variables. When the number of radial axes is large, all data points mapped to the scale may not be visible. As shown in Fig. 4 (top) for the census dataset, the data values that lie in the upper end of the scale are visible. However, the dense region around the origin does not give much information about any data values that lie closer to the lower ends of the scale. This visual clutter can be attributed to two factors. First, the dimensionality is too high which would result in mapping many radial axes from the fixed origin. Second, many values lie in the lower ends of the scale further adding to the clutter. This is the suppression of data with the loss of information. Hypothesizing information that has concealed the data likely produces a hypothesis, which may not match with the information presented by the actual data. Consider the values plotted on the radial axis numbered 65, representing the variable YEARSCH in Fig. 4 (top). A careful observation of the display reveals that the range of values in the dataset for this variable are in the mid to upper range of education with considerable range of points falling in the mid-range. One can make hypothesis that the most people have an educational qualification of 9th Grade-10th Grade or more. However, the information in the lower ends of the scale is hidden due to visual clutter.

As shown in Fig. 4 (bottom), the shifted origin star plot of the census dataset reduces the clutter in the low-value region and reveals information that might otherwise be hidden. For the variable YEARSCH, notable information that becomes apparent after the origin shift is that many data points do fall in the lower end scale of the axis. This means that a considerable portion of the population has an educational qualification lesser than 9th Grade, an important information that remained obscure earlier because of high clutter in the Fig. 4 (top).

The overlapped star plot and origin shifted star plot map all dimensions and all data in the same region. They may not be effective when the number of dimensions is very large because the inter-ray angles become too small. Fig. 5 (top) illustrates such a scenario for the 91-dimensional dataset. The clutter becomes prevalent over a wide region around the center of the display. Splitting the dimensions at three groups considerably reduces the clutter and makes all star-polylines visible (Fig. 5, bottom). One can see open space between the rays, which arises because fewer dimensions are drawn at each level and the outer regions have bigger origin shifts. It is remarkable that the multilevel star plot visualization can give a better presentation of multivariate dataset consisting of large number of dimensions by mapping each data star to a multiple sub-star icon, for example, a three sub-star drawn in Fig 5 (bottom).

If there is simply too much data, the display becomes too cluttered because of a lot of over-plotting and closely packed axes. So, normal star plot visualization may not be of much help in analyzing large multivariate dataset of high dimensionality. We need specialized visualizations and effective modes of interaction to get to know such data [5]. Our proposed approaches for star plot visualization attempt to systematically increase inter-axial angular spacing. As such, it is possible to trace all axes individually and then encode extra information about the data along them. For example, we can draw a box plot to display distribution of numerical data values through their quartiles [12]. Similarly, we can draw circles to display the relative sizes of each value belonging to a particular categorical variable, as demonstrated in Fig. 6 for the cars dataset for seven categorical variables (Table I). Also, outliers may be displayed as individual points.

Employing interaction rather than simple renderings can help manage large-scale data [5]. Highlighting and brushing may be applied to aid interpretation of star plot visualization. A straightforward option is to highlight the lines representing selected observations or outliers with different color and/or thickness. As shown in Fig. 6, the shapes of two highlighted stars are clearly recognizable for their analysis and comparison. A brushing allows us to specify a region of interest along one axis and to focus on the corresponding
stars. Brushing can be combined with other selection functionalities to make the star plot more interactive.

Moreover, it is possible to reduce visual clutter and discover interesting features in data by dimension suppression and reordering [4,16]. We can draw fewer dimensions by omitting dimensions of little or no importance. The star plot technique is generally considered to be effective only when quantitative variables are represented. The dimensions can also be ordered or clustered according to similarity [17]. Some axes may be reversed as well. While a linear scaling is often used for mapping data values on the axes, other scaling options such as a multi-scale mapping may be worth considering. If multiple pairwise relationships are of interest, the corresponding axes should be drawn next to each other.

Displaying the aggregation information derived by combining many data cases and variables can further reduce visual clutter. Data items that are similar in most dimensions should be drawn together rather than individually drawing them. Several techniques exist for forming clusters of high-dimensional data with respect to all dimensions. For instance, hierarchical clustering is a popular technique, which has been previously used with other visualization techniques including parallel coordinates [8] but not yet used with star plot visualization. Thus derived clusters can be displayed at different levels of abstraction with proximity-based coloring and structure-based brushing [8].

6 Conclusions

The star plot visualization method has been used in a wide variety of data domains. However, it becomes less effective under situations when the number of data items increases in the dataset and when the dimensionality of the data becomes too high. Many rays/dimensions have to be closely packed within a small circular area for each data item and individual star icons become too small. To overcome these problems, we have proposed different ways of effectively using the star plot visualization method.

First, instead of displaying multiple star-shaped icons spatially separately (one for each data item), we plot all data items together in the same star plot setting. Such overlapping allows the use of a larger visual display for all data and also makes identification of clusters and other features easier. Second, the shifting of the origin away from the fixed center for each ray (radial axis) widens the space between the axes. This shifted origin star plot reduces the ray crowdedness in the lower-value region of the scale. Third, our multilevel star...
plot divides the drawing of dimensions in different groups (levels) and maps them to different concentric circular regions (rings). This approach can handle very large number of dimensions such as those expected in big data problems. By visualizing three multivariate data sets of large dimensions (21, 68 and 91 variables or attributes), we anticipate that the proposed star plot visualization extensions can be potentially useful in analyzing ultra-high dimensional multivariate data.

As we have shown in this study, specialized visualizations and methods of interaction with the data are required to gain insight onto large multivariate data set of high dimensionality. Our focus has been mainly on effective representation of a large number of dimensions (variables) for star plot visualization. The supported user interface allows the user to choose desired star plot option, to highlight certain stars (data cases) and to dynamically adjust the dimension partitioning and the ring boundaries. Clearly, there is much to be done to make the proposed star plot extensions useful in a practical sense. Our further work will explore effective ways of managing large-scale data (dimension manipulation, data breakdown), enhancing line rendering (compositing, opacity), deriving and displaying aggregation information (e.g., clustering), and employing user interaction.

7 References


