Three Dimensional Star Coordinates: an Alternative Visualization Technique

Zepiao Han, Jason Yao, Dazhou Yu

The Ohio State University, Computer Science and Engineering

Abstract

Star coordinates can show multi-dimensional data in a 2D plane space. However, observing clusters in a 3D space is more intuitive for humans. By rotating one axis perpendicular to the screen, we can even find possible clusters in a n-1 dimensional subspace (n is the total number of the axis).

In this project, we extend the traditional 2D star coordinates into a 3D space. To improve the possibility of forming clusters, instead of representing each patient as a point, we use several points with a specific color to represent a patient. Several interactive options are also offered to give users more freedom to explore the EHR dataset.

Keywords: Visualization, three dimensions, interactivity, star coordinate

1 Background

Data is becoming increasingly important in many aspects and the number of dimension of datasets is also becoming larger as more data get collected. To analyze data, one way of doing that is visualization which means viewers can clearly see the relations among the data instead of getting lost in oceans of numbers. By visualizing data, the audience can do the data analysis quickly, intuitively and precisely.

Given a dataset, two-dimensional coordinate is a common visualization method. Two-dimensional coordinate is easy to design and adequate for low dimensional dataset, but when the data owns multiple dimensions, viewers would still get lost by mountains of features in the charts.

Beyond the two-dimensional coordinate, the three-dimensional coordinate is able to visualize the same amount of data but presents a much acceptable view in a richer visualization space. In three-dimensional space, instead of a plane, data is visualized in a volume which provides viewers with more degrees of freedom to choose different angles to see the data. Another possible advantage of the three-dimensional coordinate is that data in volume can be more of interactivity, which allows viewers to enjoy a better visualization experience.

Apart from dimensionality, we make use of star coordinates in our visualization. Star coordinates abandons the orthogonality of axes in order to fit the higher-dimensional dataset in limited dimension. As shown in Figure 1, data within 8 dimensions can be presented in plane using 2-dimensional star coordinate. In particular, we use star coordinates to analyze the similarity of clusters of data and to ignore unnecessary dimension in visualization.

2 Idea

Our idea is to extend the star coordinates from two-dimensional space into three-dimensional space by adding one-dimensional data.

2.1 Motivation

There are advantages of three-dimensional star coordinates. First, points are distributed in volume instead of plane, making it possible to visualize large dataset, so that viewers are able to have a global view of points. Second, three-dimensional star coordinates enable interactions with data, such as coordinates rotation and zoom in and out, which delivers an immersive experience.

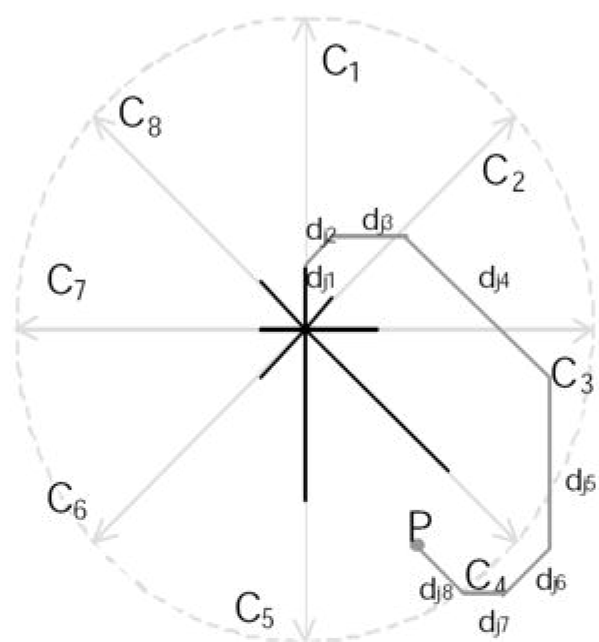

Figure 1: Computation of 2d star coordinate

3 Design

The design of our three-dimensional star coordinate have 2 parts, visualization, interaction. Figure 2 shows the ideal version of our star coordinate.

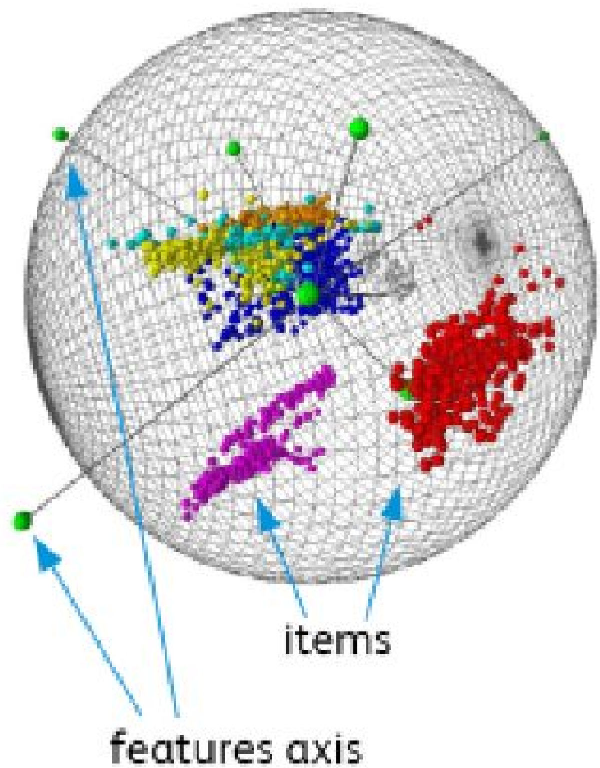

Figure 2: an ideal version of 3d star coordinate

3.1 Visualization

A 3d faceted sphere with radius 1 will be used to constraint all the feature axis and our data space. Each feature axis is viewed as a vector in Cartesian coordinate system, whose final point along the axis will be shown on the sphere. Because the points will be projected on Cartesian coordinate system, we will show the unit axis x, y, z, which have vector value (1, 0, 0), (0, 1, 0), (0, 0 ,1) respectively. The position and angle of feature axis will be pre-computed. The size of points may represent one feature if there are comparable quantitative data.

Notice that our 3d star coordinate focus on transactional dataset. The dataset have several transactions, each transaction have several items. The items may or may not be the same. We visualize the items as the points. For the points, we will assign the same color for items under the same transaction or with the same person. So that we have natural cluster inside one transaction.

3.2 Interaction

Interaction of our three-dimensional star coordinate is the key role to keep the visualization flexible and fit the need of users. Our system can do scaling and rotation over the whole sphere, or rotation over the feature axis if the user think there are better angle to represent the relationship between features rather than the rotation.

The number of items inside a transaction are also changeable based on the form of a slider. The size of points is also changeable because there may be conditions where we have few points that scatter around the space. The system also has an informative interaction that when user moves the cursor on one point, the information of the point shows.

4 Method

We view each feature axis as a vector in the three dimensional Cartesian coordinate system. The computation of feature vector is shown in Figure 3.

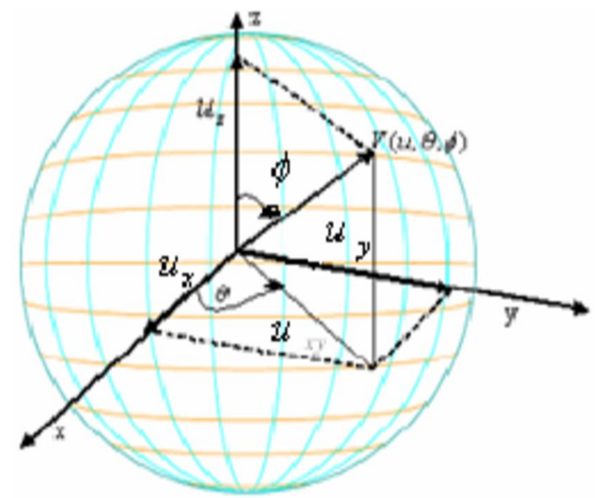

Figure 3: computation of feature vector

4.1 pre-compute angle

At initial stage, we need to assign angle for all the feature axis. In order to show the data in Cartesian coordinate system, we first choose 3 features that overlap with the base x, y, z axis.

For each of the rest of feature axis, we first compute the pearson correlation coefficient of this axis ki with x, y, z feature respectively using the formula 1, where cov is the covariance of two feature, σX is the standard deviation of X, similar for Y.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (1)$$

We normalize the coefficient $\rho x, ki$ and $\rho y, ki$ on the coefficient $\rho x, ki$ and $\rho y, ki$, by computing the ratio of two coefficients we get an angle ratio αx with respective to x axis in the x-y plane. Multiply the ratio with 90 we get the angle θxy between x and ki. Also by normalizing $\rho z, ki$ to (0,1) and multiply 90 we get the angle θz with respect to z. So the vector of the feature ki will be

(2). This vector also represent the final point along this feature axis.

$$ki = [cos(\theta xy), sin(\theta xy), cos(\theta z)] \quad (2)$$

4.2 compute points

We normalize all data under the same feature to range (0, 1). And by multiply feature value with the feature vector, we get all the feature vector for that point. Then by summing up the vector, we get the position of the point in 3d Cartesian coordinate system.

4.3 re-compute point

In the interaction we rotate the angle, here we only can rotation the angle inside the xy plane and along the z axis, which is that we only change the value of θxy and θz. Then we can recompute the feature vector by (2).

5 Data description

We are using the EHR dataset in assignment 4 (https://drive.google.com/drive/folders/1pb0CCT6Pnp0WZagkqr KtTHVpDsGsokEf). The dataset contains 5673 records, 41 patients, 34 attributes. Attributes information is provided along with the dataset:

1) Demographics information (PatientID, Gender, Age, Age_Group)
2) Injury Information (Age_TBI [age when their 1st TBI occurred], Days_From1stTBI [negative means days before, positive days after], Date of Injury, Type of Injury Code [NSFINJ = actual day of the injury, VCODE = sometime after the actual injury])
3)Encounter Information (EncounterID, Encounter_date, Encounter_Source, Provider_Specialty, Provider_type, Product_Line)
4) Other flags (TBI_encounter_flag, WarRelated_flag, PRE_max_days (to know how much data we have pre), POST_max_days (to know how much data we have Post injury)
5) Diagnosis flags [0=no, 1=yes] (Stress, PTSD, Speech, Anxiety, Depression, Headache, Sleep, Audiology, Vision, Neurologic, Alzheimer, Cognitive, PCS, Endocrine, Skull_inj, NON_skull_inj) Note that many encounters will have all the diagnosis flags as zeros. In those cases we know that that specific encounters wasn't a TBI-related encounter... they went to the doctor for something completely different such as the flu, surgery, fever, rash, annual check-up, etc...

6 Conclusion

References

[1] Cooprider, N. D., & Burton, R. P. (2007, January). Extension of star coordinates into three dimensions. In Visualization and Data Analysis 2007 (Vol. 6495, p. 64950Q). International Society for Optics and Photonics.

[2] Shaik, J. S., Kothari, C. R., Russomanno, D. J., & Yeasin, M. (2006). 3D Visualization of Relation Clusters from OWL Ontologies. In SWWS (pp. 17-23).

[3] Sun, Y., Tang, J., Tang, D., & Xiao, W. (2008, July). Advanced star coordinates. In 2008 The Ninth International Conference on Web-Age Information Management (pp. 165-170). IEEE.

[4] Kandogan, E. (2000, October). Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In Proceedings of the IEEE Information Visualization Symposium (Vol. 650, p. 22). Citeseer.

[5] Shaik, J. S., & Yeasin, M. (2006, July). Visualization of high dimensional data using an automated 3D star co-ordinate system. In The 2006 IEEE International Joint Conference on Neural Network Proceedings (pp. 1339-1346). IEEE.