Visualizing Nearest Neighbours of Large High Dimensional Real Datasets

Mounica Maddela and Kamalakar Karlapalem

Centre for Data Engineering, International Institute of Information Technology, Hyderabad, India

kamal@iiit.ac.in

ABSTRACT

The primary goal of this work is present a novel way to visualize nearest neighbours in a high dimensional dataset. It focuses on representing the data distribution in d-dimensional space on the surface area of a cone. This poster presents the design process and the capabilities the nearest neighbour visualization over cone.

1 INTRODUCTION

The nearest neighbours of the points of the dataset help to understand the distribution of the points in the dataset. As the dimension of the dataset increases, it becomes difficult to visualize the points of the dataset and its neigbours. It is difficult to find suitable mapping of high-dimensional data into a 3D visualization. Reduction in dimensions results in loss of information. Further, reduction in dimensions should preserve spatial and relative distance orientation, and the problem space to be visualized increases exponentially with dimensions. The data distribution also becomes very sparse to get compact visualization. Cone visualization, (which goes beyond parallel coordinates[1] and polar map[2]) addresses these issues and gives a visualization of the nearest neighbours.

The Cone visualisation presents a *d-dimensional data distribution* as a *three-dimensional plot*. The d-dimensional space is divided into concentric *d-spheres* with centre as the centroid of the data. The number of points in each *concentric d-sphere shell* gives the density of the data distribution. This information about density is mapped to the surface of the cone. The concentric *d-spheres* become slices on the surface of the cone and the centroid of the data becomes the tip of the cone. A d-dimensional space has 2^d number of quadrants. So, each *d-sphere* is divided into 2^d quadrants which are shown as sectors on the surface of the cone. In this scenario, the selected point and its k-nearest neighbours are highlighted. Figure 1 shows the cone visualization of iris dataset with 4 dimensions and 150 instances.

The features of CONE system are:

- Visualize high-dimensional data as a three dimensional representation and plot the nearest neigbors.
- Visualize k-neighbour graph for the points of the dataset.
- Give information about k-neighbors

2 CONE VISUALIZATION

The input to the system is the d-dimensional data set to be visualized to give Cone visualization. The approach of the system is as follows:

d-dimensional space: The centroid of d-dimensional is set as the origin of d-dimensional coordinate system. The d-dimensional space is bounded by calculating the farthest point(rmax) and neareast point(rmin) from the centroid. The radius range rmax - rmin is divided into intervals with variable length such that the number of points in each layer between two concentric d-spheres is almost the same.

Placement of the d-dimensional data points: To represent the d-dimensional data points in three-dimensional plot, three parameters are needed: radius, δ and quadrant.

The radius is given by the Euclidean distance from centroid. To represent the quadrants, we use the following convention: 0 implies that the value for a dimension is positive whereas 1 implies negative value. For each point, we check each coordinate and construct a binary string. The decimal value represented by the binary string is the quadrant number. For example, let P (1,-1,-1, 1) be a point in 4 dimensional space. It results in the binary string 0110 = 6, the point lies in quadrant 6. δ is the angle made by the vector joining point and origin with one selected dimension (i.e., base axis/dimension).

$$\cos\delta = \frac{\text{value of base-dimension}}{\text{radius}}$$

$\delta \in [0, \pi]$. But all the data points in the quadrant should be plotted within the sector corresponding to the quadrant. So, the sector angle is scaled between $[0, \pi]$ to plot the points within the sector. Here, sector angle is the angle subtended by a sector at the center. Using δ and quadrant we get Θ. If a point is in sector *i*, then

Θ = sum of sector angles of (i-1) sectors + Offset , where offset is scaled with respect to sector angle of the quadrant of the point.

$r\cos\Theta$ and $r\sin\Theta$ give *x* and *y*-coordinates for the plot. Here, *r* is the scaled version of radius as described in next section. We get the *z*-coordinate (height of the point from the tip of cone) using *r* and applying the concept of similar triangles.

$$\frac{\textit{height of cone}}{\textit{r corresponding to rmax}} = \frac{\textit{z-cordinate (height)}}{\textit{r corresponding to the datapoint}}$$

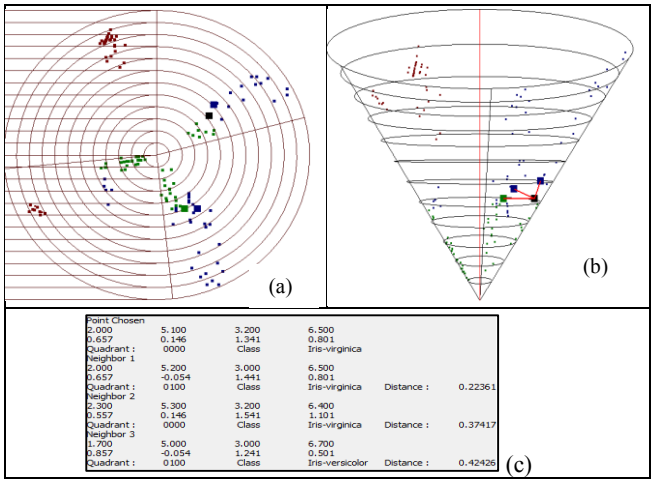

Figure 1 (a) 2D visualization of Iris dataset (4 dimensions and 150 points). (b) 2D visualization is converted into Cone visualization. The selected point (in black) and its neighbours are highlighted. (c) Information about the nearest neighbours.

A *d-dimensional* dataset has 2^d quadrants. But the cone visualization considers only the populated quadrants. We take each populated quadrant as a bit vector and come up with a simplified Boolean expression. In the reduced expression each term represents a group of quadrants and becomes a sector on the surface to the cone. Here, if a quadrant occurs in more than one term then the term with highest number of quadrants retains it quadrant and rest of the terms drop it. The angle of the sector is proportional to the data points in that sector.

3 INTERPRETING CONE VISUALIZATION

The red line represents the line from where the sectors are ordered clockwise. Note that the intervals are not uniform. The d-spheres are divided such that each concentric sphere has same number of points. The radius(r) is scaled by finding in which interval it falls. If the number of terms is very high in the final boolean expression, then we club some terms and show them as one sector.

The highlighted points give the selected point (in black) and its neighbours.The k-neighbours do not necessarily lie in the same sector of the selected point. The selected point and their neighbours are connected by line segments. We display the information about the original data values, shifted data values, distance from the selected point, class (if labelled data) and the quadrant of the neighbours and the selected point. By extending this concept, we can also create k-neighbour graph of the d-dimensional dataset. Here, all the points are connected to their k-nearest neighbours. Figure 2 gives the k-neigbour graph of the points in Iris dataset.

4 CONE VISUALIZATION SNAPSHOTS

Cone Visualization is implemented in Qt4 and opengl.
Figure 2 gives the Cone visualization of the k-neigbour graph(k=3) of Iris dataset. The dataset has 4 dimensions, 150 instances and 3 classes. The classes are Iris-setosa(brown), Iris-versicolor(green) and Iris-virginica(blue).In the figure, we can see that Iris-setosa forms a cluster far from the other two classes. So, we can say that this class is separable from other classes. On the other hand, points of Iris-versicolor and Iris-virginica are close to each other.

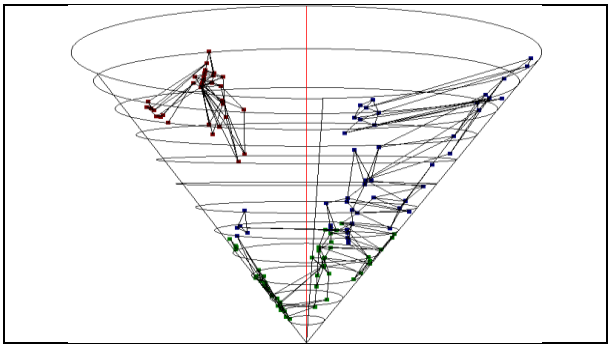

Figure 2: k-neighbour graph of Iris dataset.

Next, we visualize the page-block classification dataset[3] for document classification. Here, the page layout of a document are divided into blocks, which belong to 5 classes. The attributes give information about these blocks. The dataset has 10 dimensions, 5472 instances and 5 classes. The classes are *text, horizontal line, vertical line, picture and graphic*.

We analyze the distribution of *Horizontal line* class. We can see that the points of are concentrated at the origin. By examining the values at these points, we can tell that they mainly lie in quadrants 1001101101, 1101101101, 1000101101 and 1001101001. For a clutter-free visualization and a better understanding of the distribution, we visualize the k-neighbours (k=3) of the points of this class, which belong to other classes. Figure 3(a) gives top view of cone visualization of the k-neighbour graph of all the points of class *Horizontal line*, whose neighbours belong to other classes. By analyzing the points which have neigbours of other classes, we can get information about the distribution of the horizontal line class. For example, in figure 3(b), we analyse one such point belonging to the quadrant 1101101101. One neighbor is of the same class and same quadrant, while the other two neighbours belong to class *Picture*. Both these points belong to the quadrant 1111111110.

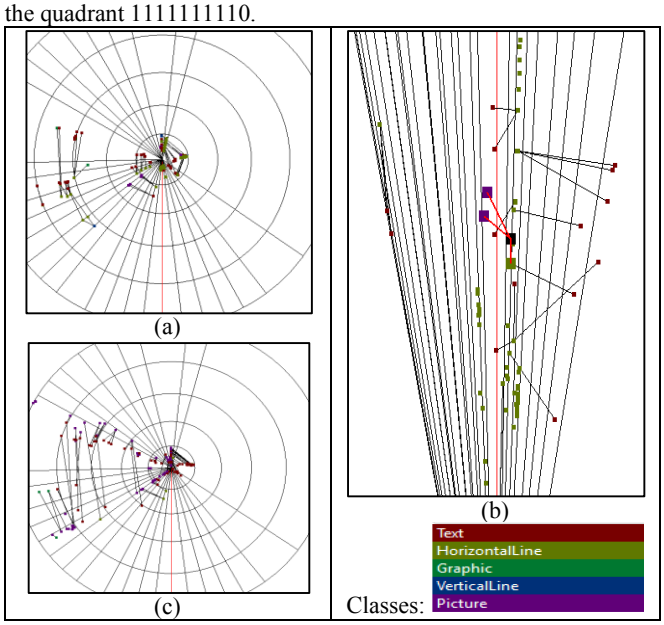

Figure 3(a):Top-view of k-neighbour graph of the class, Horizontal line,with neighbours in other classes. (b)Zoomed version giving the points closer to the origin.The selected point (in black) and its neighbors are highlighted. (c)Top-view of k-neighbour graph of the class Picture.

Figure 3(c) shows the k-neighbour graph of the class Picture. The k-neigbours of points from this class are mostly of other classes. Therefore, we can conclude that this class is widely distributed for further analytics can be conducted on these data points.

5 SUMMARY

Visualizing high dimensional large real data sets is a challenging problem. Further, determining the spatial orientation and relative distance among nearest neighbours of points is difficult. 3-d conic visualization explicitly shows neighbours across quadrants, and helps users to comprehend nearest neighbours to perform further analytics.

REFERENCES

1. *Ying-Huey Fua, Matthew O.Ward & Elk A. ,Hierarchial parallel coordinates for exploration of large datasets, VIS '99 Proceedings of the conference on Visualization '99, pages 43-50.*
2. *Frank R., Frank K. & Rudolf Kruse, POLAR MAP- Efficient Visualization of High Dimensional Data, Tenth International Conference on Information Visualization (IV'06).*
3. *Esposito F., Malerba D., & Semeraro G. , Multistrategy Learning for Document Recognition ,Applied Artificial Intelligence, 8, pp. 33-84, 1994.*