Data visualization

Heidi Matrix Implementation

Problem statement:

Follow up with Heidi Matrix : nearest neighbour Driven High Dimensional Data visualization

As discussed following tasks were expected to be done

1. Clustering
2. Ordering of data points based on 2 approaches
    a) Ordering based on nearness to centroid
    b) Connected order minimum distance
3. Heidi Matrix Computation.
4. GUI where if we click on any point it gives corresponding data points and bit vector at that point
5. Clustering on heidi matrix

Solution:

Dataset used is : IRIS dataset. (3 classes)

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

class:

-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

Class distribution : 33% for each class

1. **Clustering :**

   K-means clustering was done to find 3 clusters and their corresponding centroids

2. **Ordering :**
   a) Ordering based on nearness to centroid
      Distance of all data points within a cluster from its centroid were computed. The list obtained above was sorted in increasing order to get the desired order.
   b) Connected order minimum distance
      From centroid of cluster nearest point was computed (p_1) and then nearest point from p_1 is found and this is repeated iteratively.
   c) Minimum spanning tree approach
      Minimum spanning tree of a cluster is computed and initially a point nearest to centroid is picked up and order is order of traversal through minimum spanning tree.

3. **Heidi Matrix Computation:**

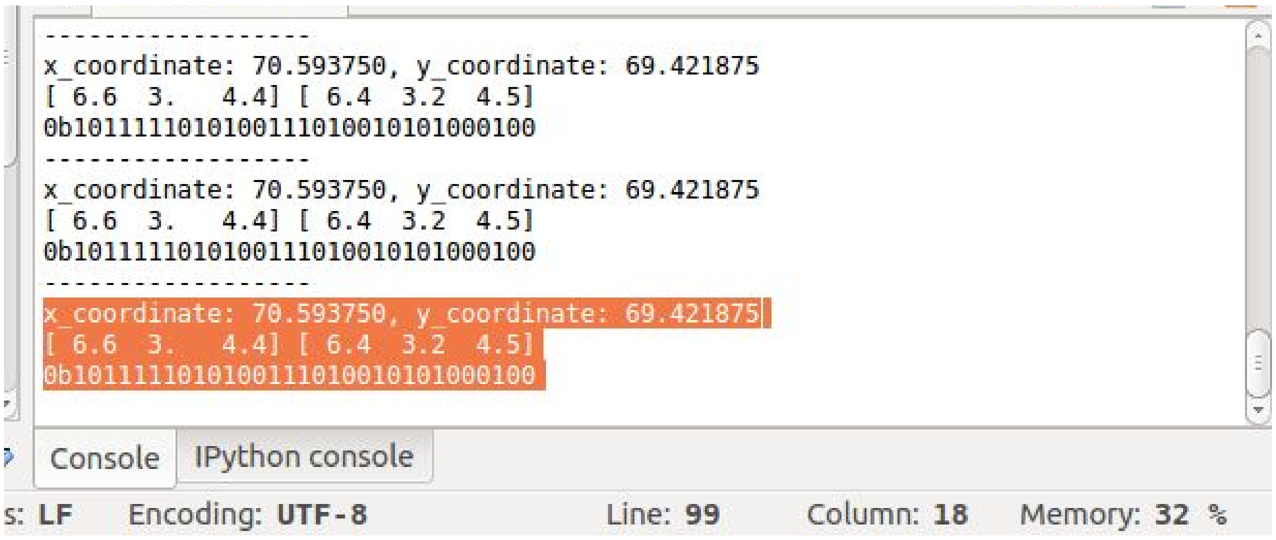
   Computed heidi Matrix

   Algorithm:
   Given n X d dataset . (n : no. of rows ; d : no. of columns)

   1. compute set of all possible subspaces. (2^n -1)
   2. For each subspace compute knn for every pair of points
   3. Merge these (2^n -1) matrices into one (bitwise)
   4. Grouping and ordering of points
   5. Draw image

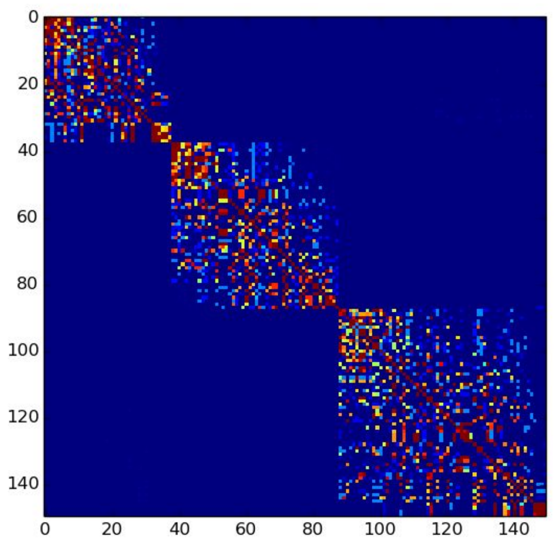
4. GUI :

In heidi matrix if we click on any point it prints at console corresponding two data points at X-axis and Y-axis respectively and bit value at that point

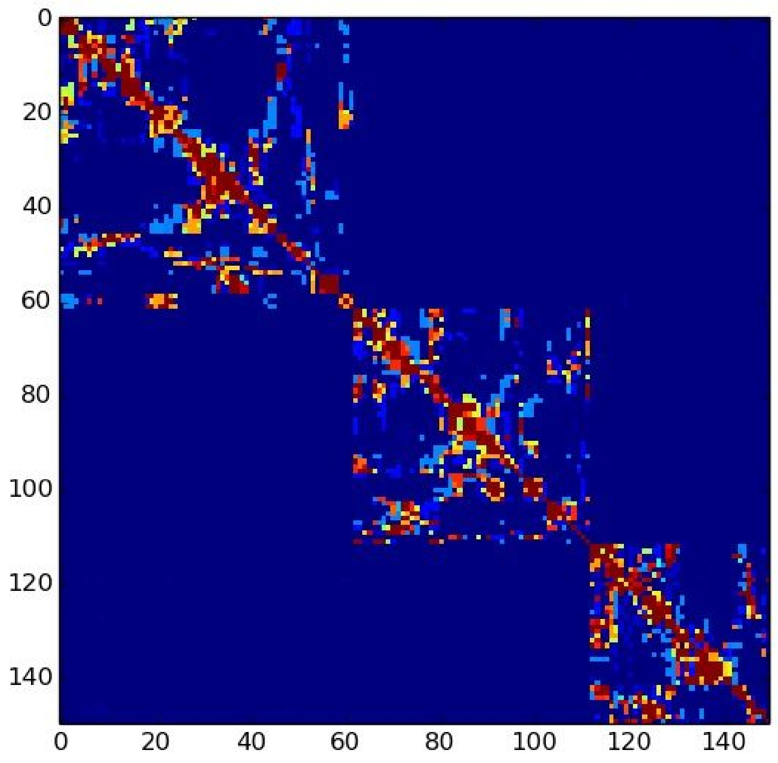


```
------------------
x_coordinate: 70.593750, y_coordinate: 69.421875
[ 6.6  3.   4.4] [ 6.4  3.2  4.5]
0b1011111010100111010010101000100
------------------
x_coordinate: 70.593750, y_coordinate: 69.421875
[ 6.6  3.   4.4] [ 6.4  3.2  4.5]
0b1011111010100111010010101000100
------------------
x_coordinate: 70.593750, y_coordinate: 69.421875
[ 6.6  3.   4.4] [ 6.4  3.2  4.5]
0b1011111010100111010010101000100
```

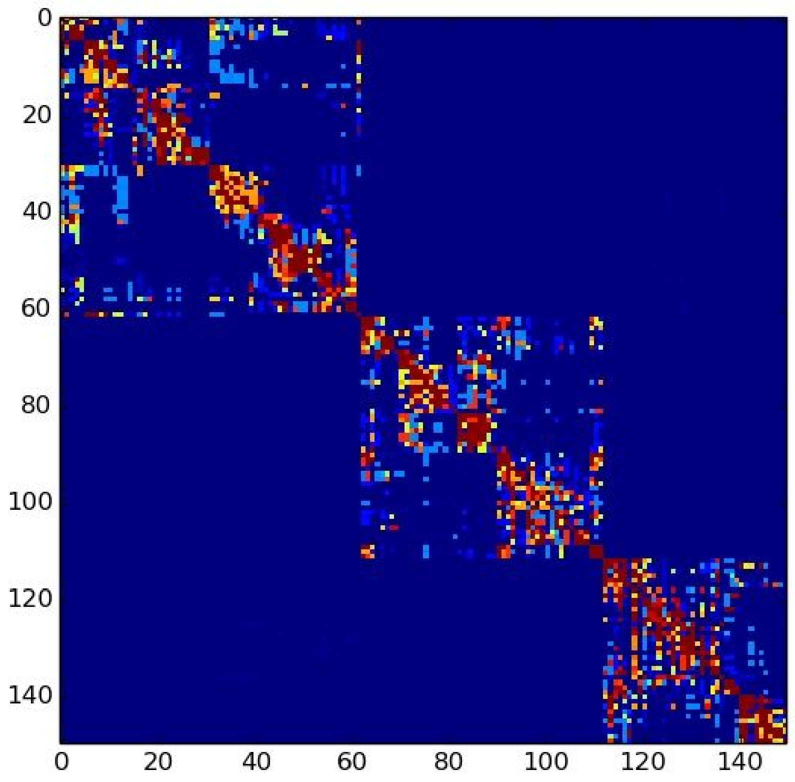
1) Ordering based on nearness to centroid



2) Connected order minimum distance



3) Minimum spanning tree approach


0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140