
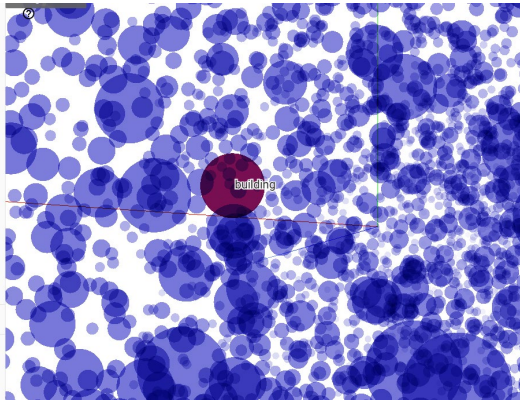


山东大学 计算机科学与技术 学院

可视化技术 课程实验报告


学号：201900130176	姓名：李伟国	班级：智能
实验题目：		
实验学时：	实验日期：	
实验目的：		
硬件环境： 处理器：AMD Ryzen 5 3600 6-Core Processor 3.60 GHz Ram 16.0 GB		
软件环境：		
实验步骤与内容： 1) 体验 tensorflow project 同样的数据集：  将其方法，然后用鼠标可以看见每一个样本如下。		



Google 这个交互式的网站，让数据仿佛活了起来

2) 使用 matlab toolbox 比较 t-sne , pca, isomap 等方法的区别

使用的实验数据如下：

 Abalone		Multivariate			
Data Set Characteristics:	Multivariate	Number of Instances:	4177	Area:	Life
Attribute Characteristics:	Categorical, Integer, Real	Number of Attributes:	8	Date Donated	1995-12-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	1156459

有 9 个维度，其中第一个是性别。

Sex / nominal / -- / M, F, and I (infant)
Length / continuous / mm / Longest shell measurement
Diameter / continuous / mm / perpendicular to length
Height / continuous / mm / with meat in shell
Whole weight / continuous / grams / whole abalone
Shucked weight / continuous / grams / weight of meat
Viscera weight / continuous / grams / gut weight (after bleeding)
Shell weight / continuous / grams / after being dried
Rings / integer / -- / +1.5 gives the age in years

The readme file contains attribute statistics

- t-sne：非线性的降维技术（non-linear）

t 是指 t-分布（也叫学生分布），stochastic Neighbor Embedding，叫做随机邻居嵌入。此方法相当的适合用来做高维度的数据可视化。该技术可以通过 Barnes-but 来近似的实现，这样的速度比较快。

t-sne : 相似的目标通过其附近的样本点来建模而不相似的目标通过远的目标点来建模, 有很高的概率。所以 t-sne 对原始空间中的数据集进行相似性的建模和降维后的空隆中的相似性建模都是概率密度。

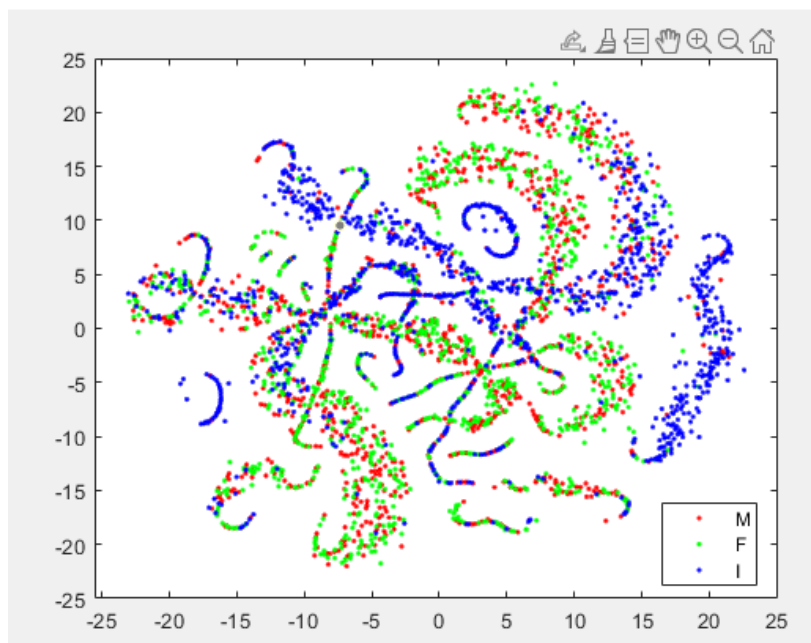
如果两个点在原始的空间中比较接近, 那么他们在降为后的空间中任然是比较接近的。

t-distributed: 设随机变量 X 和 Y 相互独立, 并且 $X \sim N(0, 1), Y \sim \chi^2(n)$, 则称随机变量 $t = \frac{X}{\sqrt{\frac{Y}{n}}}$ 是服从自由度为 n 的 t 分布;

接下来, 对从网站上下载的数据集进行 t-sne 降维, 在运用 t-sne 之前, 人们总是先对数据进行 PCA 降维处理, 再进行 tsne 降维。

Tsne 有一种能够聚类的效果

下面看一下该方法降维后的效果



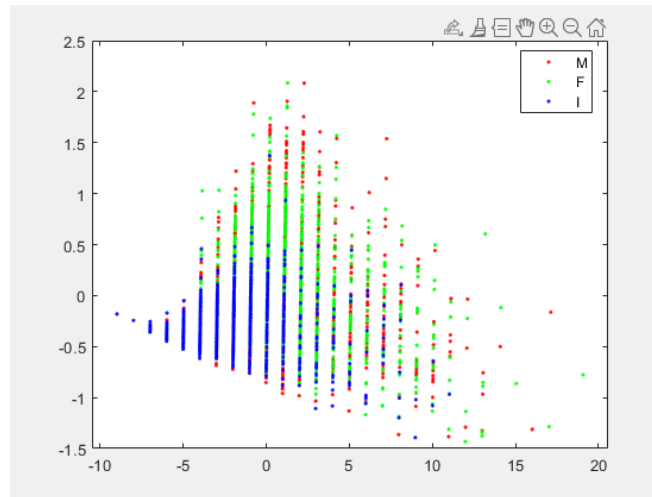
显而易见, 我们可以从图中发现一些拥挤的感觉

随着迭代次数的增加, error 一直在减少, 但是并不是线性的减少。

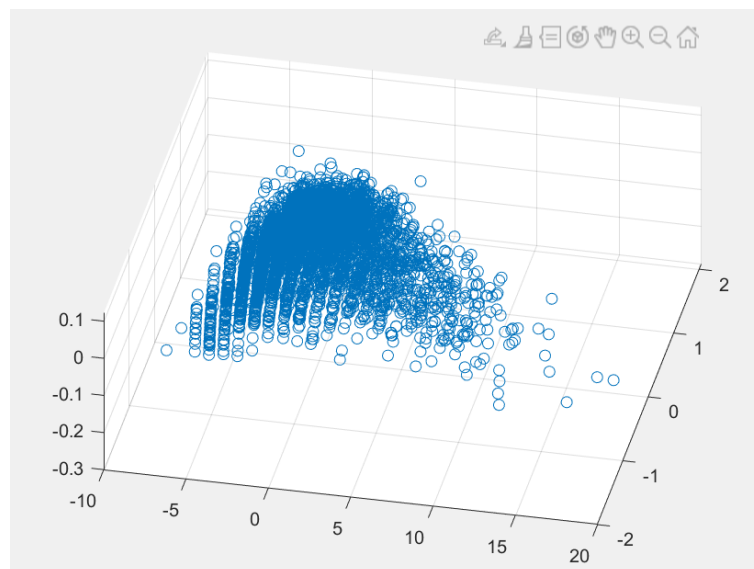
```
Iteration 20: error is 4.8127
Iteration 40: error is 3.6463
Iteration 60: error is 2.8432
Iteration 80: error is 2.3719
Iteration 100: error is 2.0457
Iteration 120: error is 1.8082
Iteration 140: error is 1.6344
Iteration 160: error is 1.5012
Iteration 180: error is 1.3907
Iteration 200: error is 1.2934
Iteration 220: error is 1.2098
Iteration 240: error is 1.1387
Iteration 260: error is 1.0768
```

- PCA

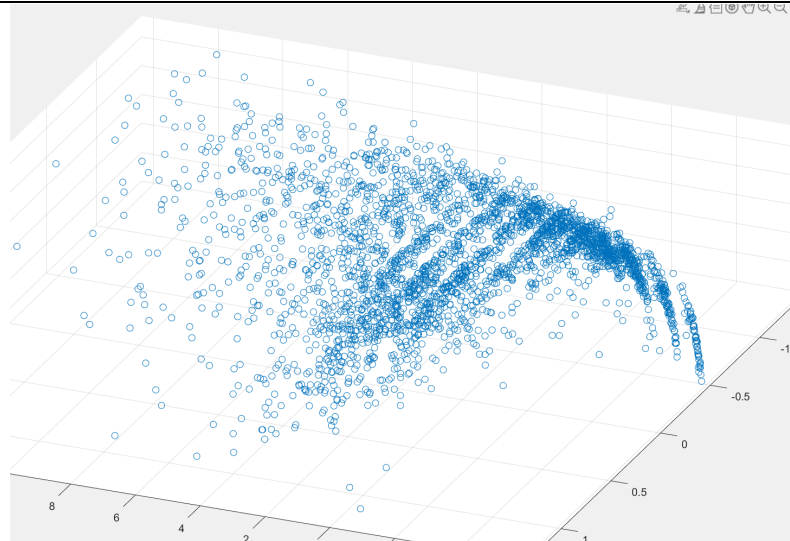
PCA 计算的核心就是计算一个 `covariance matrix`，并对其特征分解（奇异值分解），并按照特征值从大到小的顺序进行选择其对应的特征向量，特征向量的方向就是主成分的方向，然后根据情况选择前 k 个最大的特征向量作为方向，此时就可以把高维的数据降到了 K 维，在这里， K 取 3 或者 2，这样就可以将我们的高维数据在低维度的空间中可视化出来。



降到 2 维



降到 3 维度



- Isomap 一种无监督的算法 (unsupervised)

这是一种非线性的降维方式，是一种嵌入的方式。该算法提供了一种简单的方法，用于基于流形上的每个数据点的邻居的粗略估计来估计数据流形的内在几何形状。这是一种对数据种类和数目很丰富的有效的可行的方法。其能够使用非线性的方式去降维并能保留局部的结构

Isomap 使用嵌入的邻域图的 geodesic distance 而不是欧式距离。

给定数据集，经过最近邻等方式构造一个数据图 (data graph, neighborhood graph)。而后，计算任意两个点之间的最短路径 (即测地距离)。对于全部的任意两个点对，指望在低维空间中保持其测地距离。

算法步骤如下：

- 1) 决定每一个点的邻居：K nearest neighbors

任然用欧氏距离来确定每个 point 的最近的 k 个邻居，而非最近的 k 个邻居的距离设置为无穷大

- 2) 构建邻域图 (neighborhood graph)

如果数据点和数据点是邻居关系，那么就将他们相连接。如果不是就保持不连接的状态。

- 3) 计算两个点的最短路径 (Dijkstra 或者 floyd warshall)

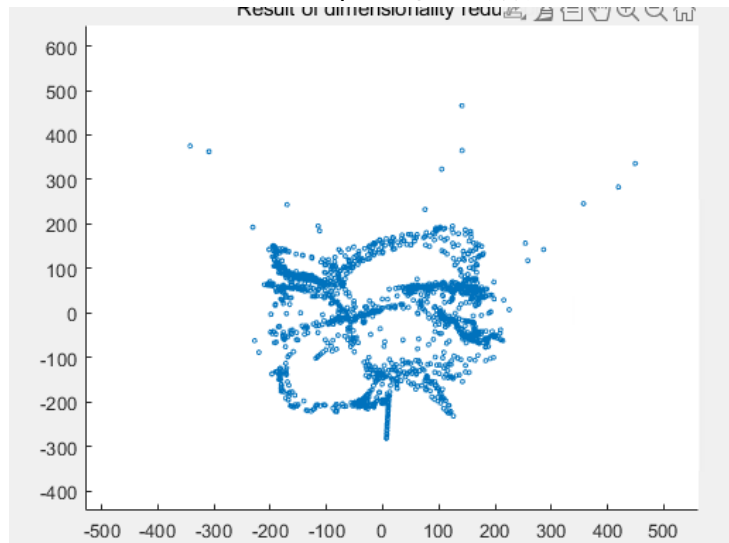
这一步也叫做在两个点之间通过邻居图

求 geodesic distance,

- 4) 计算低维空间的嵌入 (MDS)

因为现在任意两个点之间的额距离已经知道了，MDS 尽可能的保持任意两个点之间的距离在 embedding space 和 original space 不变

经过 isomap 降维后的数据



结论分析与体会：

- 1) iosmap （等距特征映射）可以保留一些数据点之间的非线性关系，isomap 也经常用于 NLP 分析中去。让两个点之间的距离近似的等于依次多个临近点的连线的长度之和
- 2) PCA 实际上式一个线性变换，将原始的数据 变换到一个新的坐标系中，使得任何数据的投影的方差在一个坐标上最大，在二个坐标上第二大，etc。该方法减少数据的维数的同时保持数据集的对方差贡献最大的特征。
- 3) 不同于 ISOMAP 中距离不变的思想，而是现将欧式距离转变成条件概率，来表示点与点的相似度，在优化 loss function (KL 散度)，从而保证点一点之间的分布概率不变并且在低维空间先使用更重长尾的 t-distributed 来避免拥挤的问题

附录：程序源代码