来一块二向箔，分析用

- Python高维数据可视化

莫瑜
海豚浏览器

About Me

莫瑜

1.Microsoft - Dolphin

2.搜索引擎，推荐系统

3.关注和实践大规模数据算法性能优化和人工智能技术

4.喜欢读书，尤喜科幻

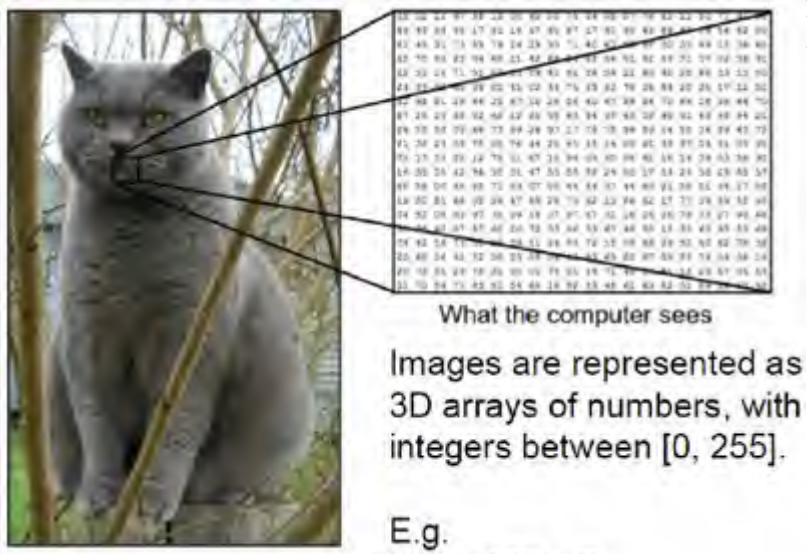
目　录

高维数据

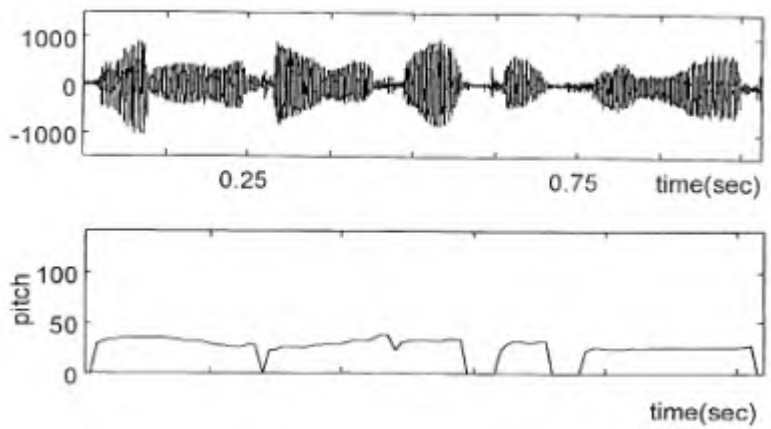
文本(词/句子/文档) one-hot encoding/BOW representation

$$w^{aardcark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \cdots \quad w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

图像



音频



.....

高维数据

Iris Data Set

Download: Data Folder, Data Set Description

Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1091271

.....
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
.....
5.1,2.5,3.0,1.1,Iris-versicolor
5.7,2.8,4.1,1.3,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
.....

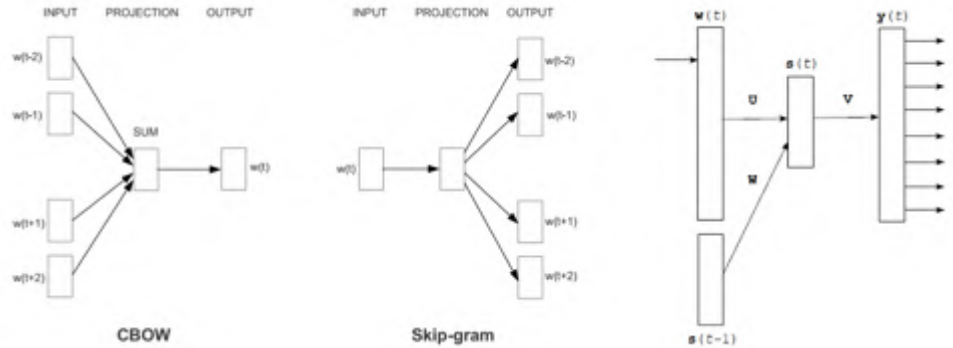
Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:

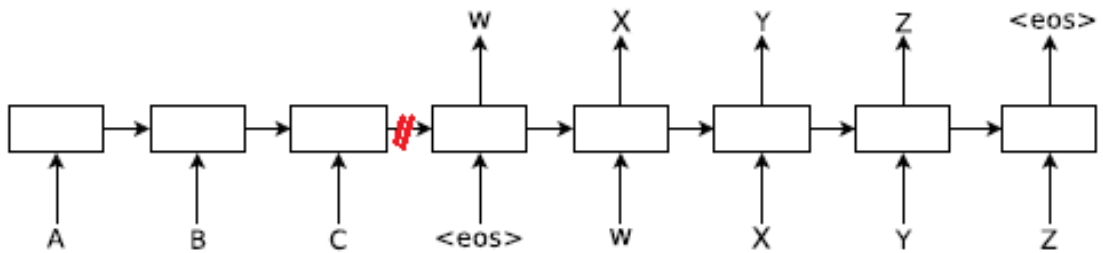
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

高维数据

Text




```
WALKER'S -0.090689 -0.383161 -0.569909 -0.626396 0.250820 -1.663088 -0.873118 0.411699 -0.367016 -0.
NERD -0.066861 0.649640 -1.157251 -0.526284 0.281073 -0.831471 -0.209744 -0.716850 -0.705707 -1.003
RAKED 0.158288 0.743522 -0.826185 0.120958 1.442295 -0.689943 -0.194061 -0.463098 1.212968 -0.376594
TAMOXIFEN -0.231270 0.118784 0.056632 -1.016629 0.667149 -0.262810 -1.220728 -0.636376 -0.481767 -0.
DEFRAUD 0.029352 -1.663714 -1.424621 -0.960925 -0.684159 -2.075946 -1.462572 -0.098368 1.635968 -0.4
MOBSTER -0.150721 -0.566374 -1.918836 -0.673172 -0.177824 -2.094826 0.053153 -0.944704 0.070066 -1.3
CHANEL 0.040901 0.802255 -2.308696 -1.048910 -1.141584 -0.186766 -0.472789 -0.173971 0.248361 -0.363
ZUCKERMAN -0.286497 0.247890 -1.512235 -0.086487 0.537495 -0.851625 -0.669148 -0.617157 -0.173713 -0
PUNDIT -0.087679 -0.599805 -1.980600 -0.228036 -0.636863 -0.693111 -0.502600 -0.905233 -0.726924 -1.
DUKES -0.382466 1.088892 -1.419441 -0.540625 -0.658076 -0.603717 -0.215285 0.272466 1.274940 0.32597
FAGER 0.097299 -0.282136 -0.769613 -0.611815 -1.574321 -0.775738 -0.622792 -0.651123 0.012215 1.0237
```



Image

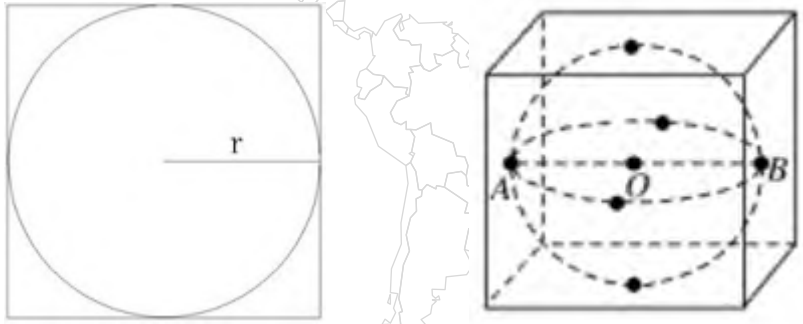
高维数据 – 为什么不容易理解高维数据

高维空间的反直觉现象

- **高维空间基本都分布在“边界/面”上**
- 边长为2r的d维超立方体体积为 $(2r)^d$，内切的半径为r的d维超球体的体积为 $\frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}$

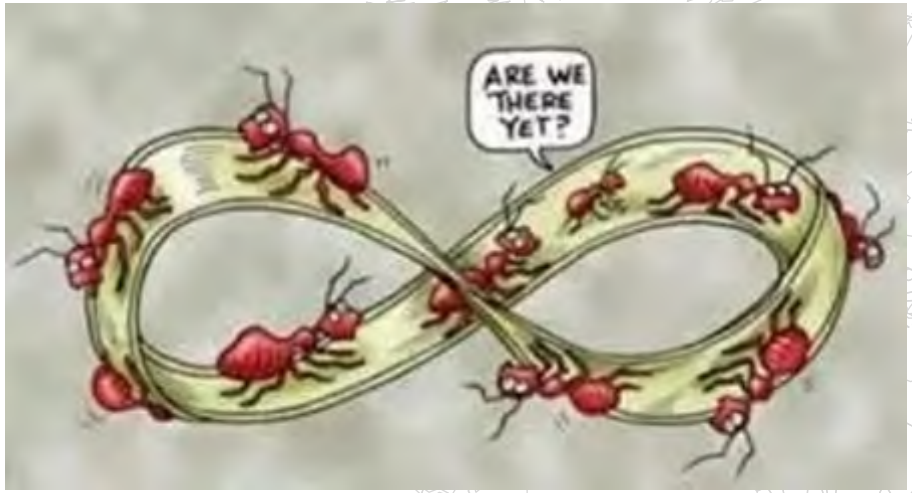
存在：

$$\lim_{d\to\infty}\frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)}\to 0$$



- 当单位d维超立方体的边长缩为 α，超立方体的体积缩小为 α^d

高维数据 – 为什么不容易理解高维数据



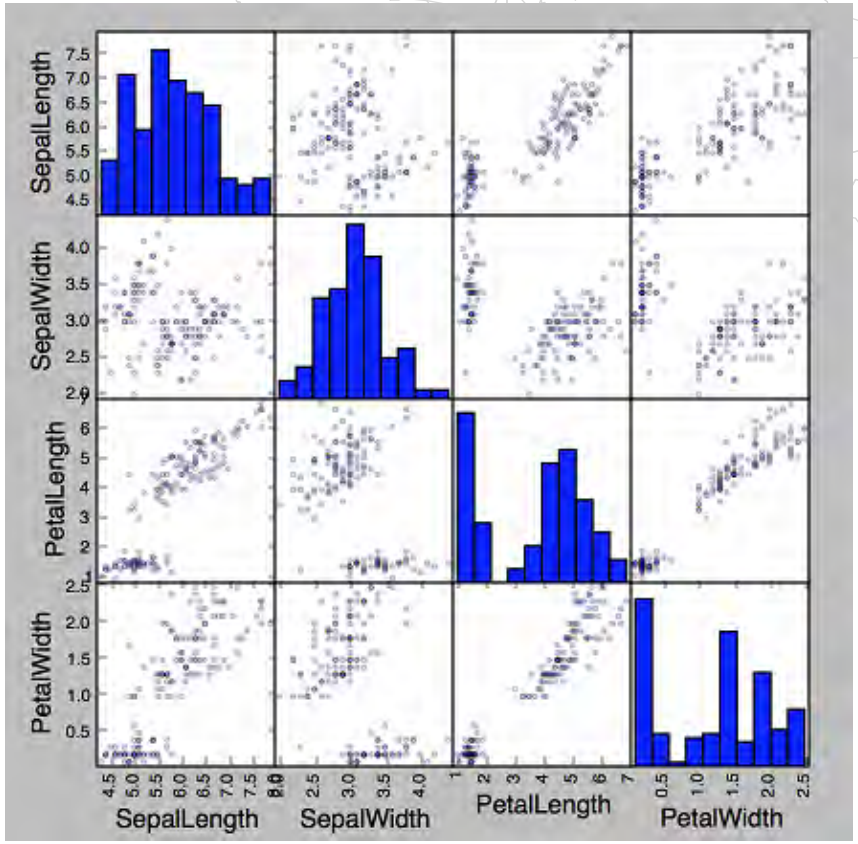
- 怎么理解高维数据？
  - 真的高维数据吗？是否信息冗余？
    - Fitted Curve？{(x_0, x_1, …, x_n)}　x_i = f(i)
  - 考察单维度
    - 统计信息(均值，方差，直方图(Histogram)等)
  - “一图胜千言”，能否数据可视化？
    - 多元数据可视化
    - 高维数据可视化(降维)

多元数据可视化

- Geometric projection
  - Line graph
  - **Scatterplot Matrix**
  - Prosection matrix
  - Hyberslice, Hyberbox
  - **Parallel coordinates**/Radical coordinate
  - Table lens
  - **Andrews curves**
- Pixel-oriented techniques
  - Space filling curve
  - Recursive pattern
  - Spiral and axes techniques
  - Circle segment
- Hierarchical display
  - Hierarchical axis, Dimensional stacking, Worlds within worlds
- Iconography
  - Star glyph, Chernoff faces

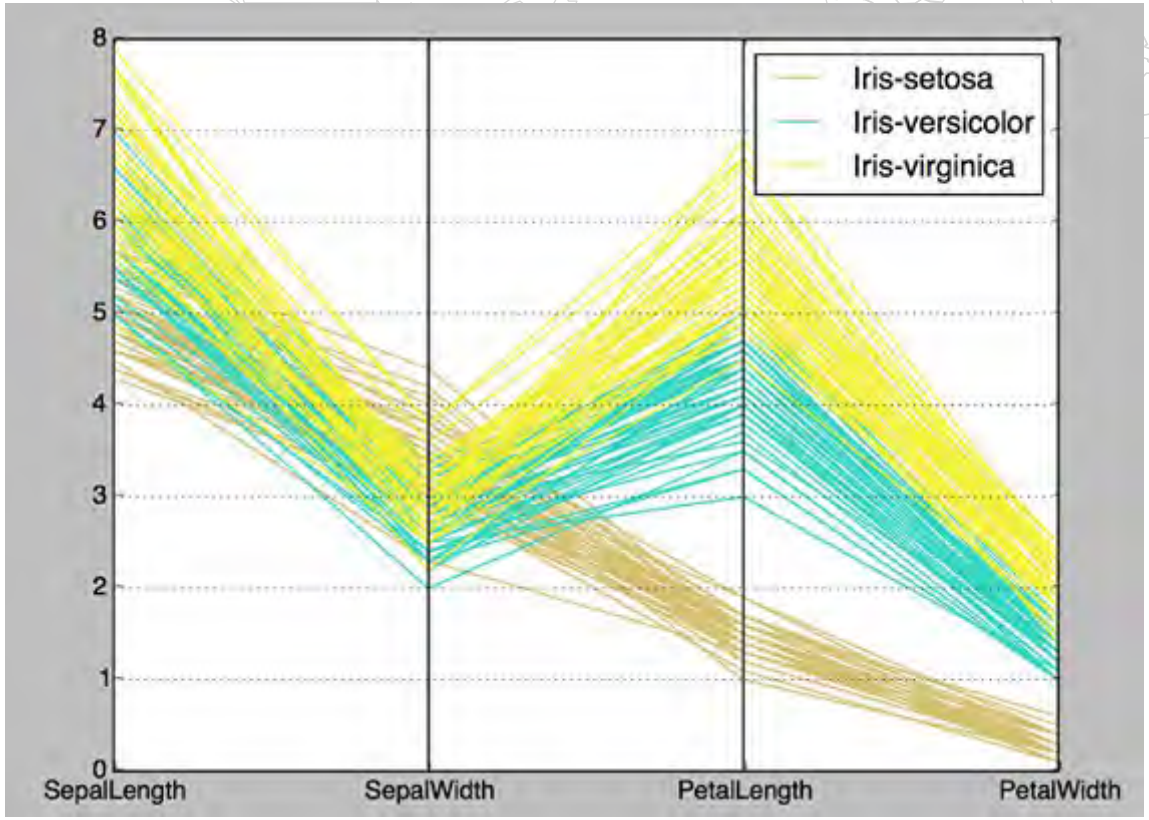
多元数据可视化 - Scatterplot Matrix

```
32  import matplotlib.pyplot as plt
33  import pandas as pd
34  from pandas.tools.plotting import scatter_matrix
35  data = pd.read_csv('data/iris.csv')
36  plt.figure()
37  scatter_matrix(data, figsize=(6, 6), diagonal='hist')
38  plt.show()
```



多元数据可视化 - Parallel coordinates

```
41  import matplotlib.pyplot as plt
42  import pandas as pd
43  from pandas.tools.plotting import parallel_coordinates
44  data = pd.read_csv('data/iris.csv')
45  plt.figure()
46  parallel_coordinates(data, 'Name')
47  plt.show()
```

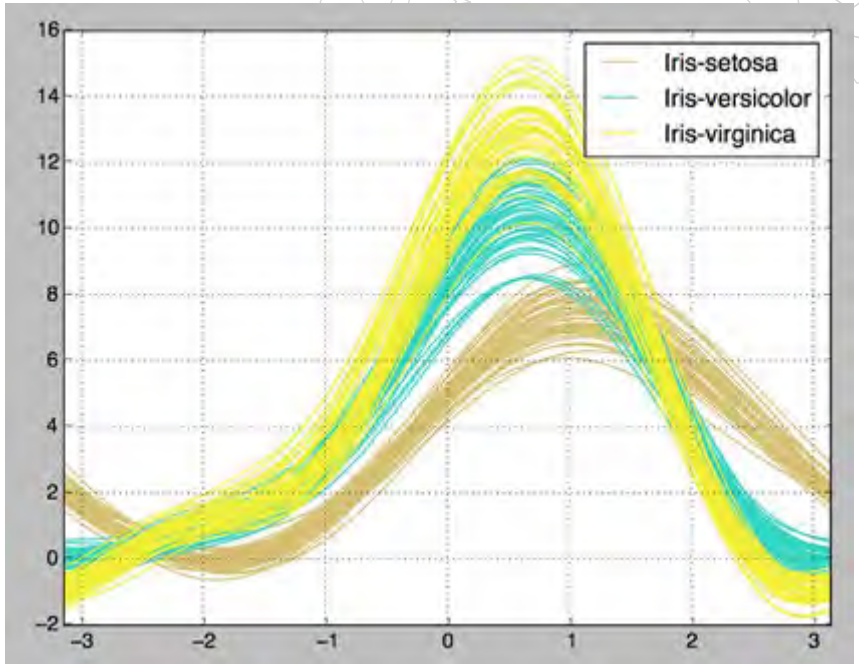


多元数据可视化 – Andrews Curves

Each data point $x = \{x_1, x_2, \ldots x_d\}$ defines a finite Fourier series:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + \ldots$$

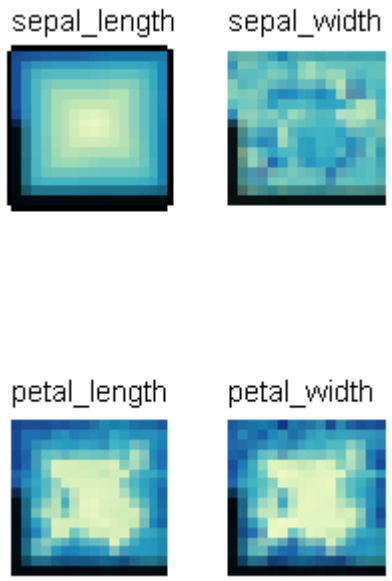
```
51  import matplotlib.pyplot as plt
52  import pandas as pd
53  from pandas.tools.plotting import andrews_curves
54  data = pd.read_csv('data/iris.csv')
55  plt.figure()
56  andrews_curves(data, 'Name')
57  plt.show()
```



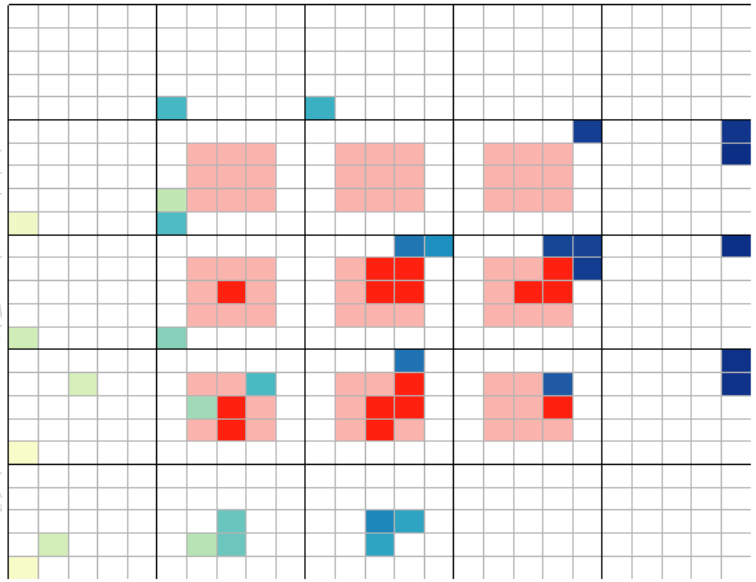
多元数据可视化 – 更多

- Python pandas
  - E.g. pandas.tools.plotting
- XmdvTool

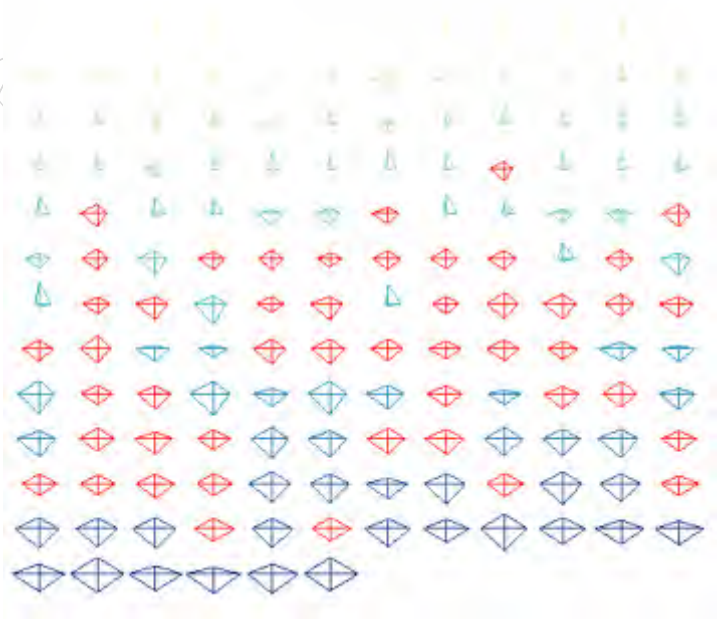
Pixel oriented display



Dimensional Stacking



Glyph



高维数据可视化(降维)

寻找

$$\phi: \quad \mathbb{R}^m \quad \rightarrow \mathbb{R}^t$$
$$x_i \quad \mapsto y_i, \text{ for } \quad 1 \leq i \leq n,$$

目标：

$$\varepsilon_\phi = \sum_{1 \leq i,j \leq n} W_{i,j} \ (\delta_m(x_i, x_j) - \delta_t(y_i, y_j))^2, \ \ for \ W \in \mathbb{R}^{(n \times n)},$$

- 线性
  - PCA, LDA, etc
- 非线性 (Manifold Learning)
  - Isomap, LLE, t-SNE, etc

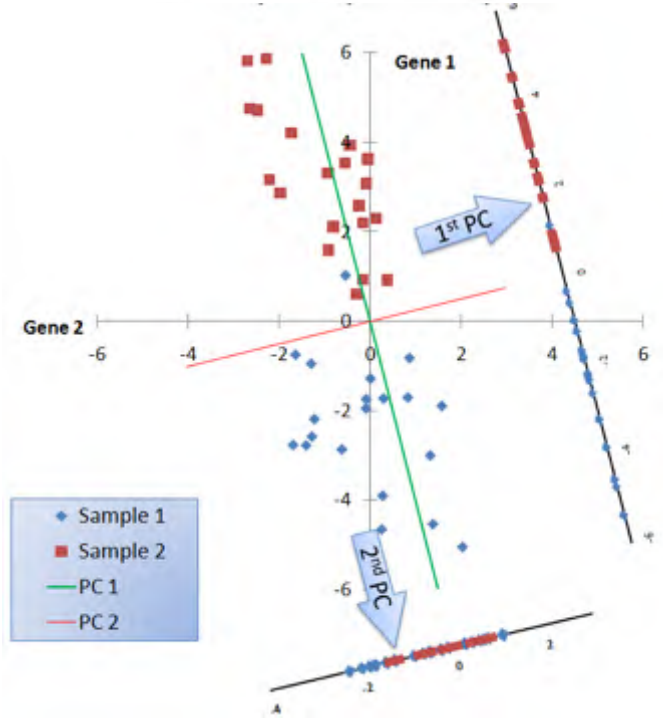
高维数据可视化 – 数据:sklearn手写数字

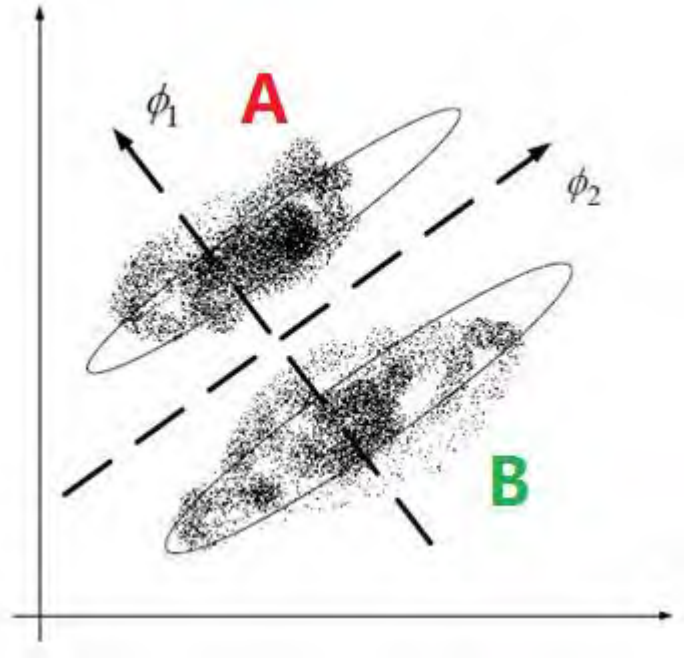
```
33  digits = datasets.load_digits(n_class=6)
34  X = digits.data
35  y = digits.target
36  n_samples, n_features = X.shape
37  n_neighbors = 30
38
39  #----------------------------------------------------------------------
40  # Plot images of the digits
41  n_img_per_row = 20
42  img = np.zeros((10 * n_img_per_row, 10 * n_img_per_row))
43  for i in range(n_img_per_row):
44      ix = 10 * i + 1
45      for j in range(n_img_per_row):
46          iy = 10 * j + 1
47          img[ix:ix + 8, iy:iy + 8] = X[i * n_img_per_row + j].reshape((8, 8))
48
49  plt.imshow(img, cmap=plt.cm.binary)
50  plt.xticks([])
51  plt.yticks([])
52  plt.title('A selection from the 64-dimensional digits dataset')
```

高维数据可视化 - PCA

- 线性降维变换
- 目标：最大化低维空间中点的方差

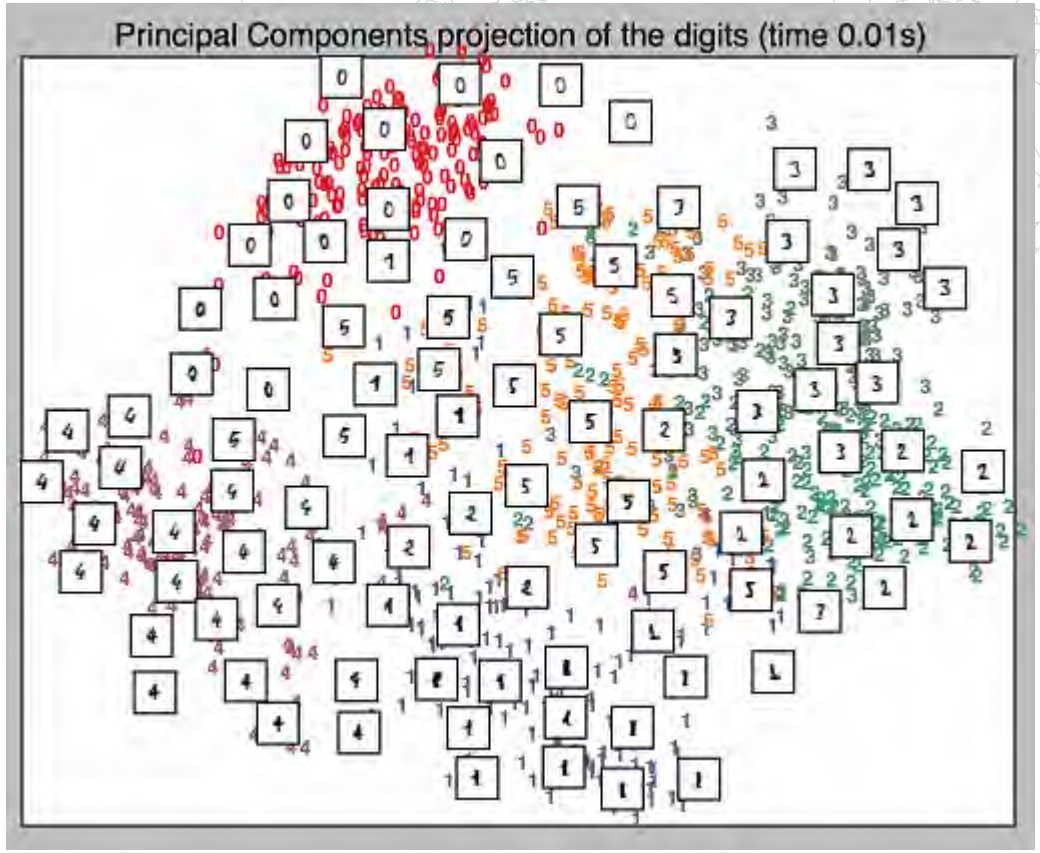
$$\max_{W^{m*t}} \frac{1}{N-1} \Sigma_{i=1}^{N} \left\| W^T x_i - W^T \bar{x} \right\|$$





高维数据可视化 - PCA

```
94  #----------------------------------------------------------------------
95  # Projection on to the first 2 principal components
96
97  print("Computing PCA projection")
98  t0 = time()
99  X_pca = decomposition.PCA(n_components=2).fit_transform(X)
100 plot_embedding(X_pca,
101                "Principal Components projection of the digits (time %.2fs)" %
102                (time() - t0))
```



高维数据可视化 – t-SNE

- SNE(Stochastic Neighbor Embedding)的改进版本
- 主要思想
  - 点对距离在高维与低维空间之间相似性，构造概率分布，使用KL距离衡量
  - 相比SNE
    - 解决KL距离不对称问题
    - 低维空间使用t-分布，高维空间仍然使用高斯分布

高维空间概率分布

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

低维空间概率分布

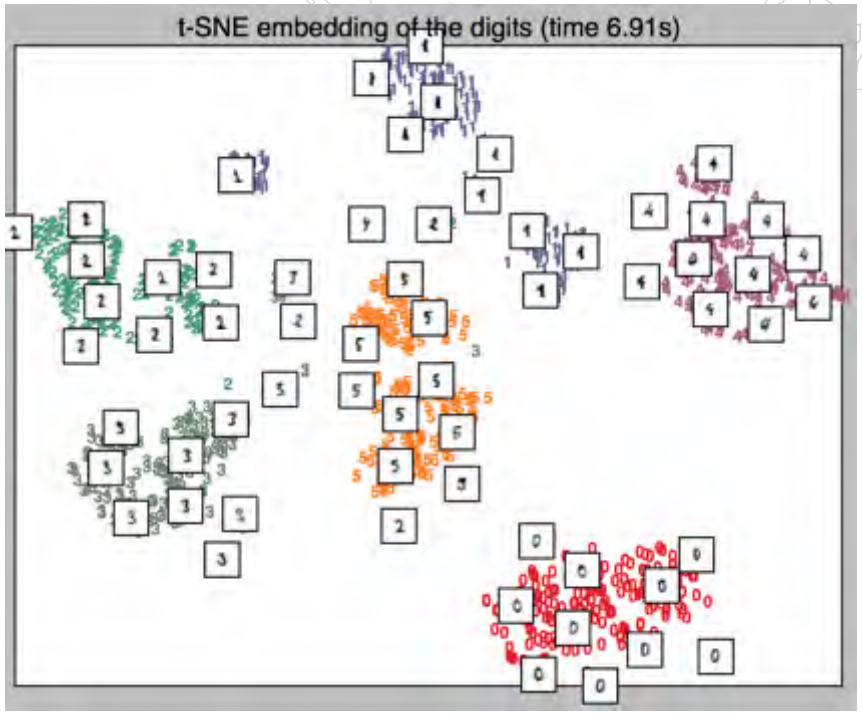
$$q_{ij} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

相似性

$$KL(P||Q) = \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}}$$

高维数据可视化 - tSNE

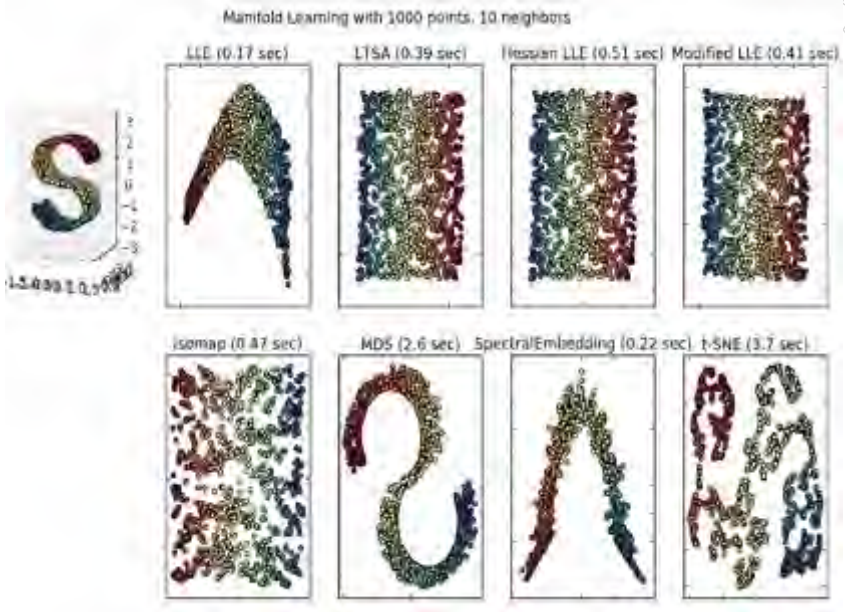
```
216  #----------------------------------------------------------------------
217  # t-SNE embedding of the digits dataset
218  print("Computing t-SNE embedding")
219  tsne = manifold.TSNE(n_components=2, init='pca', random_state=0)
220  t0 = time()
221  X_tsne = tsne.fit_transform(X)
222
223  plot_embedding(X_tsne,
224                 "t-SNE embedding of the digits (time %.2fs)" %
225                 (time() - t0))
```



高维数据可视化 – 更多

- sklearn.decomposition: Matrix Decomposition
- sklearn.discriminant_analysis: Discriminant Analysis
- sklearn.manifold: Manifold Learning

`manifold.LocallyLinearEmbedding([...])`	Locally Linear Embedding
`manifold.Isomap([n_neighbors, n_components, ...])`	Isomap Embedding
`manifold.MDS([n_components, metric, n_init, ...])`	Multidimensional scaling
`manifold.SpectralEmbedding([n_components, ...])`	Spectral embedding for non-linear dimensionality reduction.
`manifold.TSNE([n_components, perplexity, ...])`	t-distributed Stochastic Neighbor Embedding.
`manifold.locally_linear_embedding(X, ...[, ...])`	Perform a Locally Linear Embedding analysis on the data.
`manifold.spectral_embedding(adjacency[, ...])`	Project the sample on the first eigenvectors of the graph Laplacian.



高维数据可视化 – 更多

- 应该选择什么方法？
  - 考察不同方法的算法假设
  - 应用场景数据特点：线性/非线性？ 度量数据(metric-data, non-metric data)？
  - 全局结构信息？局部结构信息？
  - 计算性能
- Tips
  - 统一不同维度的值域
  - More：http://scikit-learn.org/stable/modules/manifold.html#tips-on-practical-use

总结

- 高维数据无处不在
- 维度较小时，可以尝试多元数据可视化(Pandas, XmdvTool)
- 高维数据，结合实际情况，尝试降维可视化(PCA, sklearn.manifold)

谢谢观看

b9875641

finalbob@hotmail.com

13810971657

dolphin.com