

パターン認識 Pattern Recognition

佐藤真一
Shin'ichi Satoh

国立情報学研究所
National Institute of Informatics

July 4, 2023

最終レポート

- いわゆるトップジャーナル・トップ国際会議の任意のパターン認識に関する論文についてまとめる
- e.g., IEEE TPAMI, IJCV, CVPR, ICCV, NeurIPS, ICML, ACMMM...
- 以下について記述せよ:
 - 論文の書誌事項
 - 論文の概要
 - どのような問題を扱っているのか、なぜその問題は重要なのか、どのように解決しているのか、どのように評価しているか
 - なぜその論文を選んだのか、何が特に興味深いのか
 - 講義への意見・コメント
- A4 2-4 ページ程度
- 締め切り: 07/31/2023
- ITC-LMS より提出

Final Report

- Geman and Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, TPAMI, 1984.
- Moghaddam and Pentland, Probabilistic Visual Learning for Object Detection, TPAMI, 1997.
- Belhumeur, Hespanha, and Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, TPAMI, 1997.
- Shi and Malik, Normalized Cuts and Image Segmentation, TPAMI, 2000.
- Belkin and Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, NIPS, 2001.
- Zhang and Sim, When Fisher meets Fukunaga-Koontz: A New Look at Linear Discriminants, CVPR, 2006.
- Felzenszwalb, Girshick, McAllester, and Ramanan, Object Detection with Discriminatively Trained Part Based Models, TPAMI, 2009.
- Antonio Torralba and Alexei A. Efros, Unbiased look at dataset bias, CVPR 2011.

クラスタリング実験

データセット 4 種

- Two Gaussians (two flattened Gaussians)
- Four squares
- Four Gaussians
- Swiss role

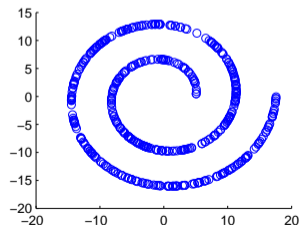
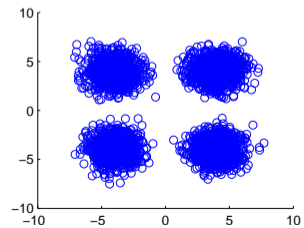
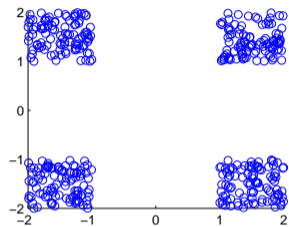
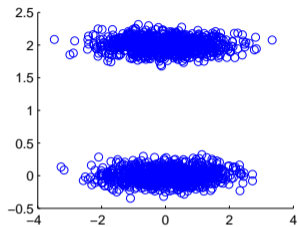
クラスタリング手法 3 種

- k-means
- Single linkage
- Complete linkage

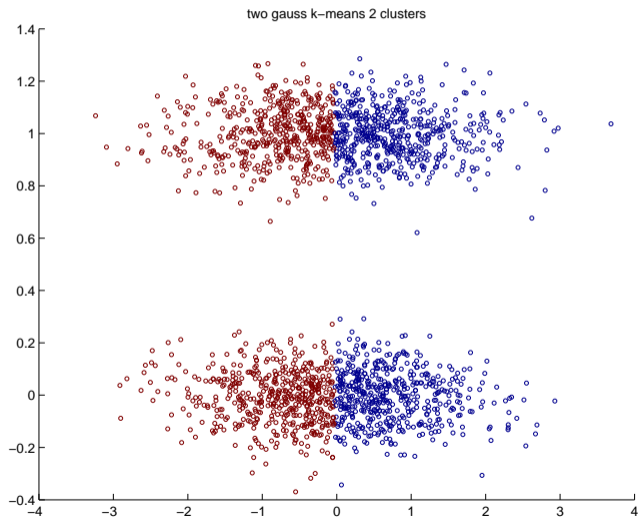
クラスタリング手法 (再掲)

- k-means** k 個の代表点をランダムに選び、各データを最も近い代表点に所属させ、代表点を所属しているデータの平均値で置き換えるという処理を繰り返す
- Single linkage** 階層的凝集型クラスタリング手法であり、クラスタ間の距離を要素間の最短距離とする方法
- Complete linkage** 階層的凝集型クラスタリング手法であり、クラスタ間の距離を要素間の最長距離とする方法

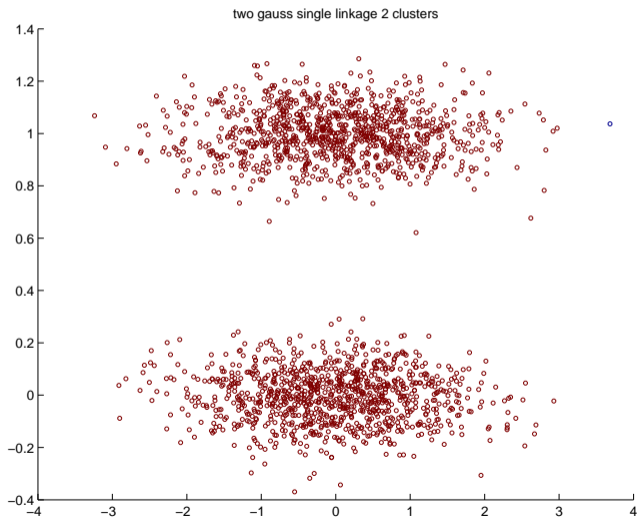
データ



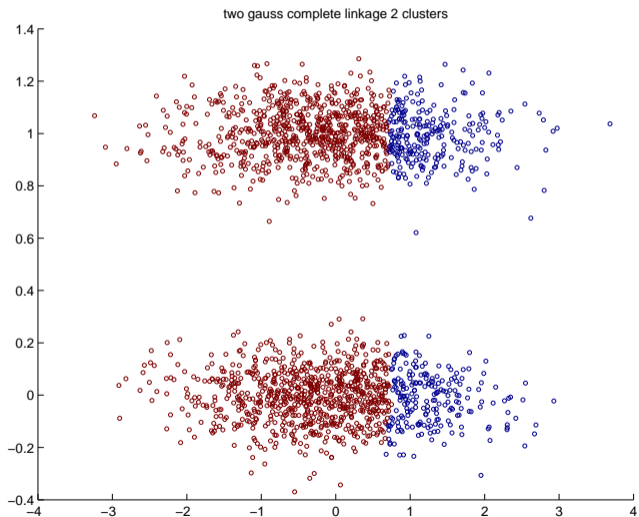
Two Gaussians



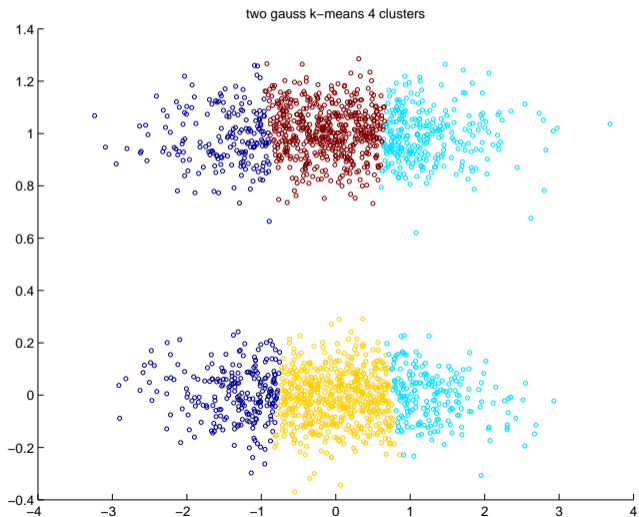
Two Gaussians



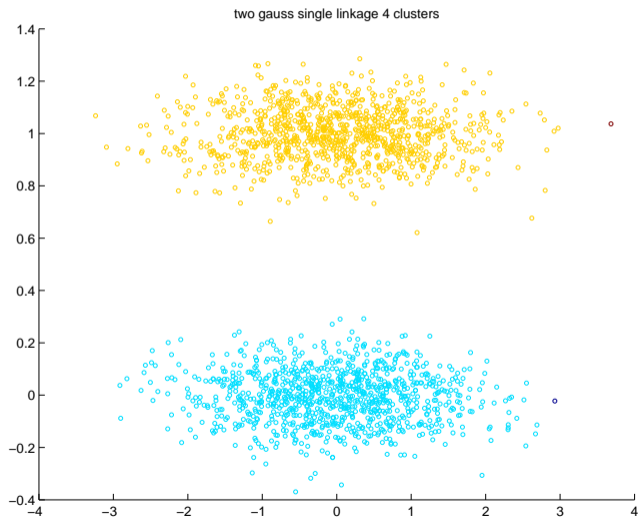
Two Gaussians



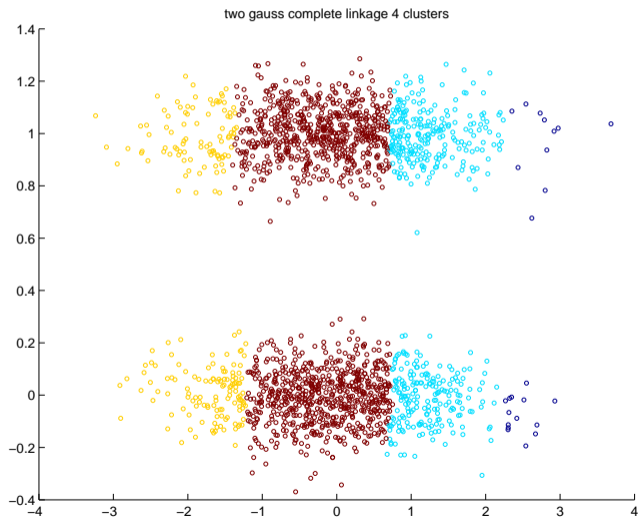
Two Gaussians



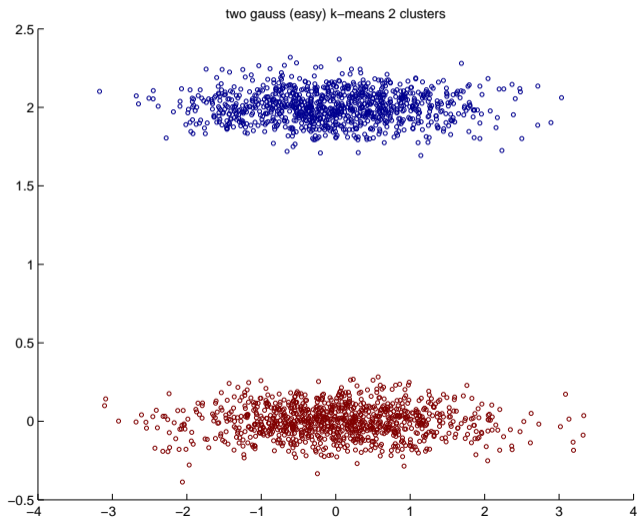
Two Gaussians



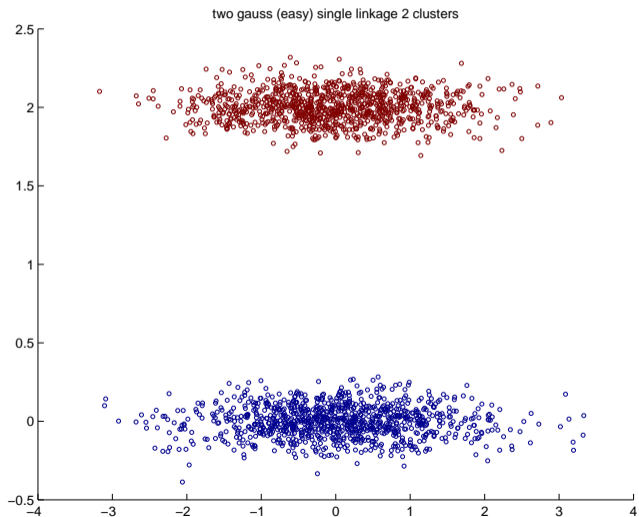
Two Gaussians



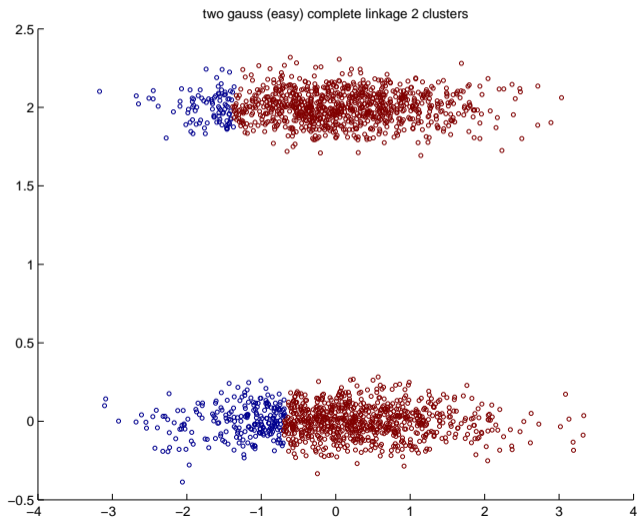
Two Gaussians



Two Gaussians

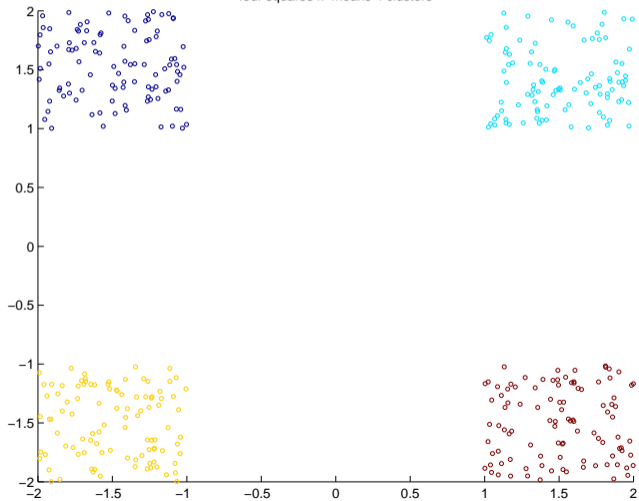


Two Gaussians

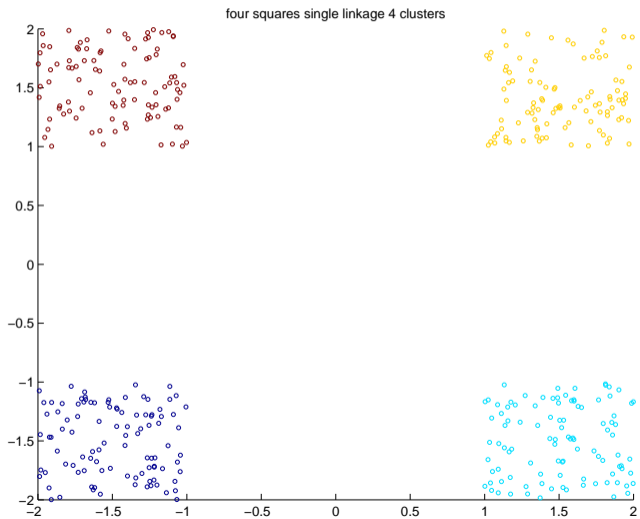


Four Squares

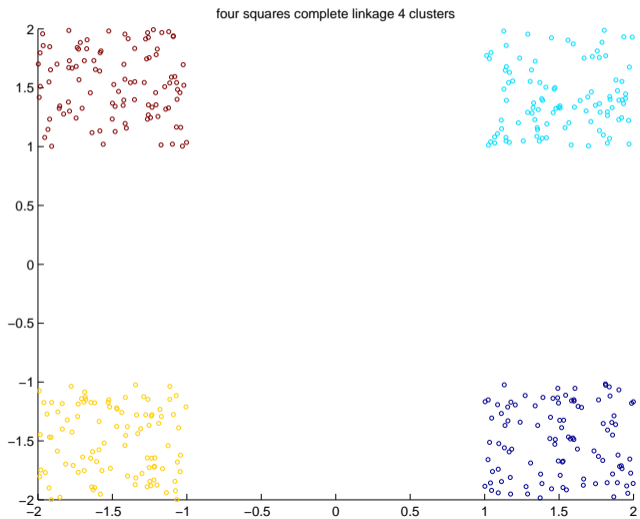
four squares k-means 4 clusters



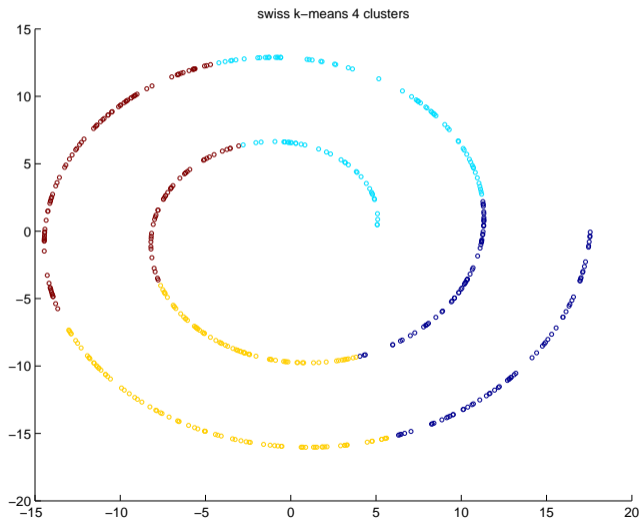
Four Squares



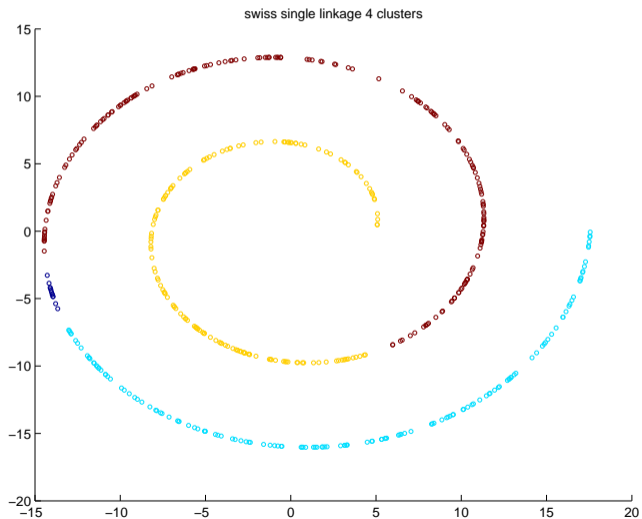
Four Squares



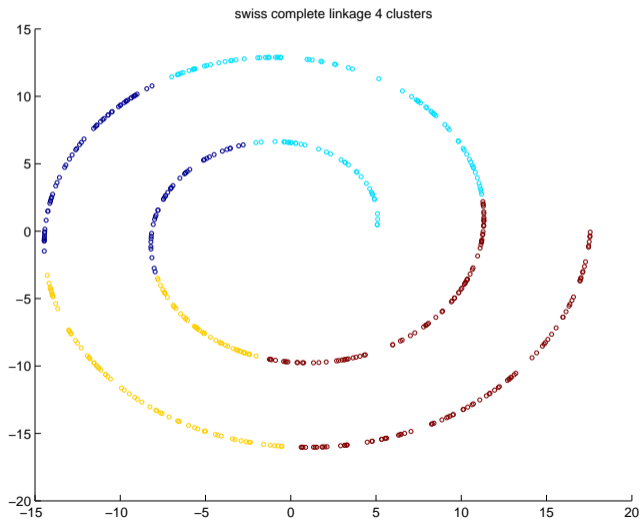
Swiss Role



Swiss Role



Swiss Role

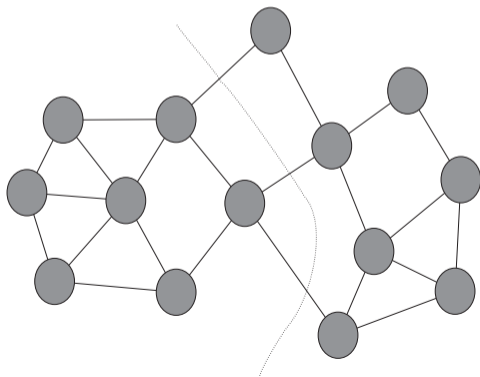


グラフスペクトラルクラスタリング (Graph Spectral Clustering)

- これまで、クラスタリングの対象となるデータはベクトルとして表現され、距離はユークリッド距離で計測できることを仮定してきた
- しかし、もし距離 (あるいは類似度) のみが与えられていた場合はどのようにするか
- あるいは、距離が近傍領域のみでしか有効でない場合はどうするか
- グラフ表現に基づくクラスタリング技術はこうした状況に有効である
- データとそれらの間の類似度をグラフとして表現し、クラスタリングをグラフの分割問題として解く

Graph Spectral Clustering

- 各データをノードとする
- 二つのノード間の類似度があれば辺とする
- 類似度を辺の重みとする
- グラフのカットとして、カットされる辺の重みの和を用いる



Graph Spectral Clustering

n ノードのグラフを $n \times n$ 行列 W により、 $w_{i,j}$ が i 番目と j 番目のノードの類似性に対応するように表現する

まず、単純にデータを二つの集合に分割する方法について考える

メンバーシップ指示ベクトル q_i を決定すればよい

$$q_i = \begin{cases} 1 & \text{if } i \in A \\ -1 & \text{if } i \in B \end{cases}$$

$$J = \text{CutSize} = \frac{1}{4} \sum_{i,j} w_{i,j} (q_i - q_j)^2 = \frac{1}{4} \sum_{i,j} w_{i,j} (q_i^2 + q_j^2 - 2q_i q_j)$$

$$= \frac{1}{2} \sum_{i,j} q_i (d_i \delta_{i,j} - w_{i,j}) q_j$$

$$= \frac{1}{2} q^T (D - W) q$$

$$d_i = \sum_j w_{i,j}$$

Graph Spectral Clustering

$J = \frac{1}{2}q^T(D - W)q$ の最小化は困難である (NP 完全)

そこで q を離散値から連続値へと緩和する: この場合 $\min J(q)$ の解は以下の固有ベクトルとして得られる

$$(D - W)q = \lambda q$$

グラフラプラシアン (Graph Laplacian) の性質

$L = D - W$ はグラフのラプラシアン行列と呼ばれる

L は半正定値行列: $x^T L x \geq 0$ for any x .

最小固有値は $\lambda_1 = 0$ 固有ベクトルは $q_1 = [1 \ 1 \ \dots \ 1]^T = e^T$ (どういうグラフ分割に対応?)
二番目の固有ベクトル q_2 が欲しい解 (Fiedler ベクトルと呼ばれる)

$$\lambda_2 = \frac{\text{cutsizesize}}{|A|} + \frac{\text{cutsizesize}}{|B|}$$

J は以下のように q に対する定数の加算に不変なので

$$J = \frac{1}{4} \sum_{i,j} w_{i,j} ((q_i + c) - (q_j + c))^2$$

例えば単純に q_2 をソートして真ん中で切ればよい

様々なクラスタリング目的関数

- Ratio Cut

$$J_{Rcut} = \frac{s(A, B)}{|A|} + \frac{s(A, B)}{|B|}$$

- Normalized Cut

$$\begin{aligned} J_{Ncut} &= \frac{s(A, B)}{d_A} + \frac{s(A, B)}{d_B} \\ &= \frac{s(A, B)}{s(A, A) + s(A, B)} + \frac{s(A, B)}{s(B, B) + s(A, B)} \end{aligned}$$

- Min-Max Cut

$$J_{MMcut} = \frac{s(A, B)}{s(A, A)} + \frac{s(A, B)}{s(B, B)}$$
$$s(A, B) = \sum_{i \in A} \sum_{j \in B} w_{i,j}, \quad d(A) = \sum_{i \in A} d_i$$

ラプラシアンアイゲンマップ (Laplacian Eigenmaps)

各ノードを低次元のベクトル表現に埋め込む (embed) 方法を考える
そのため、以下の一般化固有値問題を解く

$$Lf = \lambda Df.$$

(これは Normalized Cut と同じ解法)

固有値 $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_n$ に対応する固有ベクトル f_0, f_1, \dots, f_n について
 $[f_1(i) \ f_2(i) \ \dots \ f_m(i)]^T$ を i 番目のノードに対応付ける

- 辺の重みを以下の通り定義する:

$$w_{i,j} = \begin{cases} e^{-\frac{d_{i,j}^2}{\sigma^2}} & \text{if } i \in knn(j) \text{ or } j \in knn(i) \\ 0 & \text{otherwise} \end{cases}$$

- Laplacian eigenmaps 法により、 i 番目のノードを $[f_1(i) f_2(i) \dots f_m(i)]^T$ と表現する
- 変換後のベクトルに対して k-means をかける
- 定数: k of knn: 5, σ : 10, m : 2 等

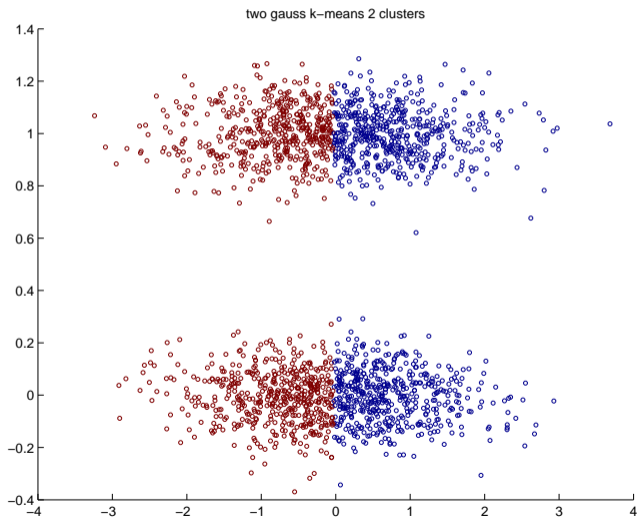
Laplacian Eigenmaps (Matlab)

```
k=10;
sigma=10;
n=size(x,1);
dt=squareform(pdist(x));
[sdt,idx]=sort(dt,'ascend');
dt=sdt(1:k+1,:);
nidx=idx(1:k+1,:);
tW=exp(-dt.^2/(sigma^2));
ii= repmat(1:n,k+1,1);
W=sparse(ii,double(nidx(:)),tW(:),n,n);
W=full(W);
W=max(W,W');
D=diag(sum(W,2));
[v,d]=eigs(D-W,D,10,'sa');
xx=v(:,2:3);
c=kmeans(xx,numc);
```

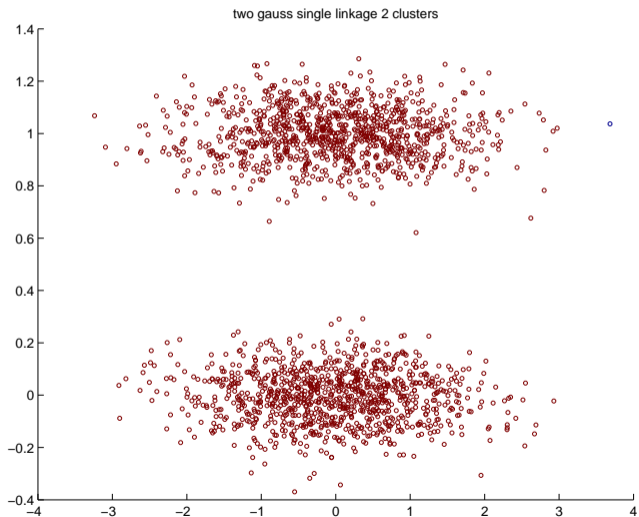
Laplacian Eigenmaps (Python)

```
k = 10
sigma = 5.
n = x.shape[0]
dt = squareform(pdist(x))
idx = np.argsort(dt)
dt = np.array([dt[i, idx[i, range(k)]] for i in range(n)])
W = np.zeros([n, n])
for i in range(n):
    for j in range(k):
        W[i, idx[i, j]] = np.exp(-(dt[i, j]**2) / (sigma**2))
W = np.maximum(W, W.T)
D = np.diag(np.sum(W, axis=0))
dd, v = scipy.sparse.linalg.eigs(D - W, k=5, M=D, which='SR')
v = np.real(v)
idxdd = np.argsort(np.real(dd))
xx = v[:, idxdd][:, range(1, 4)]
c = sklearn.cluster.KMeans(n_clusters=numc).fit_predict(xx)
```

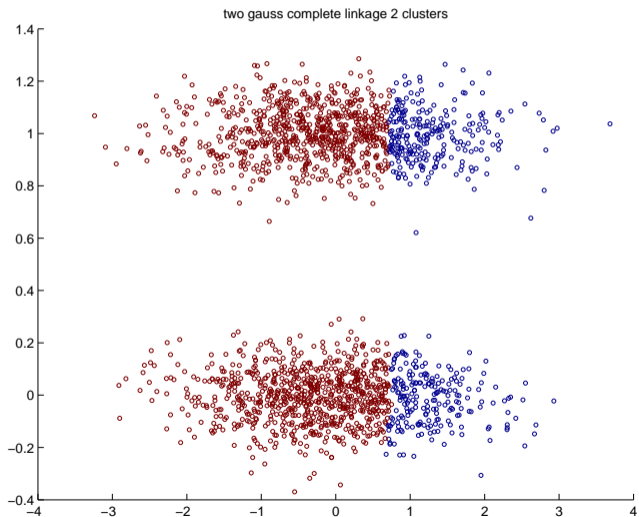
Two Gaussians



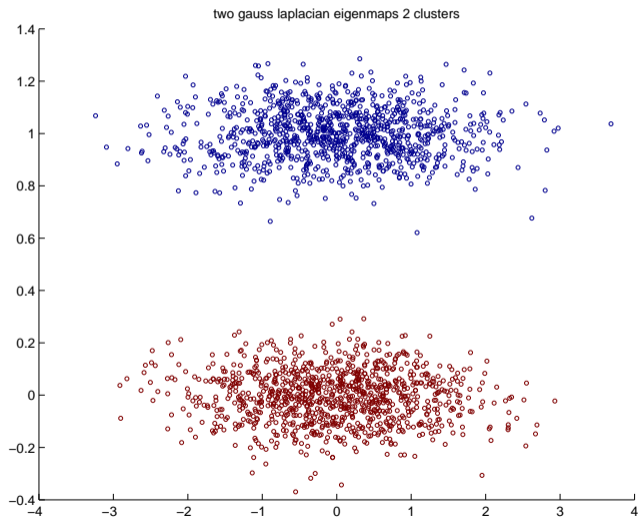
Two Gaussians



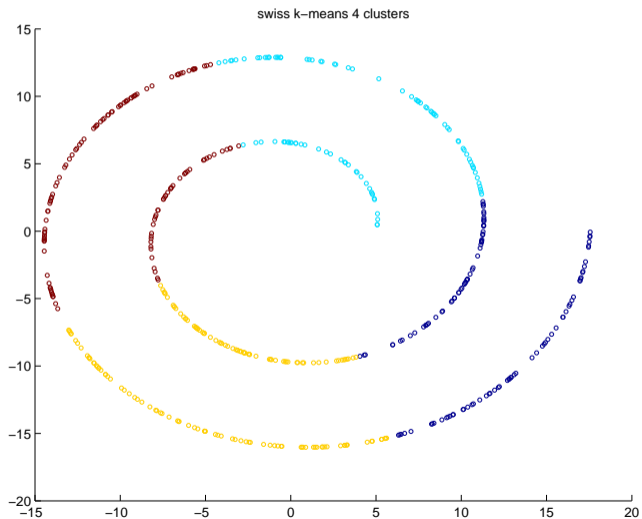
Two Gaussians



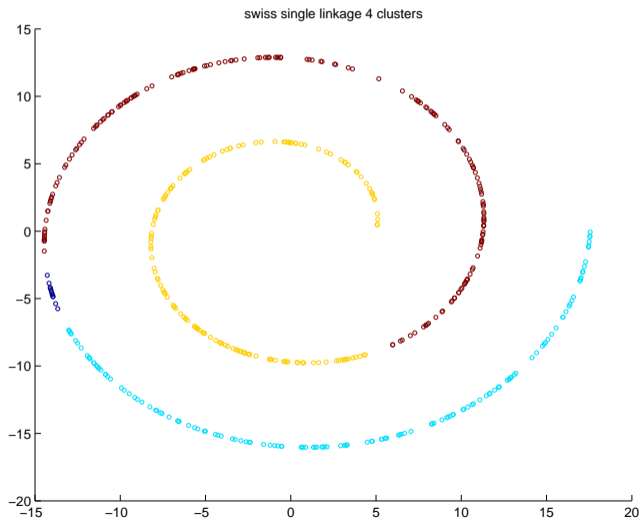
Two Gaussians



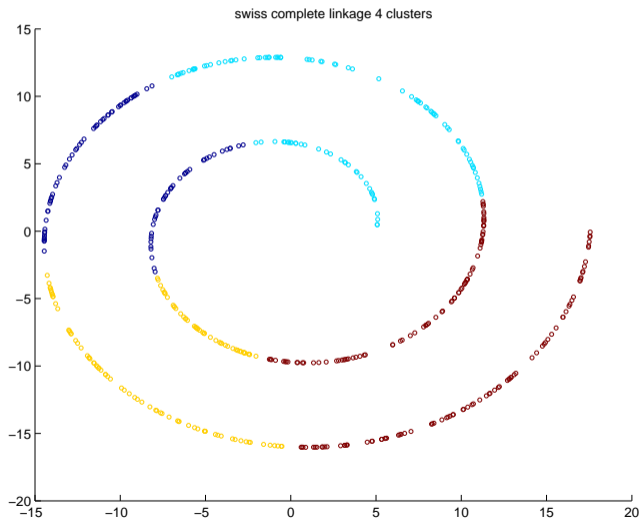
Swiss Role



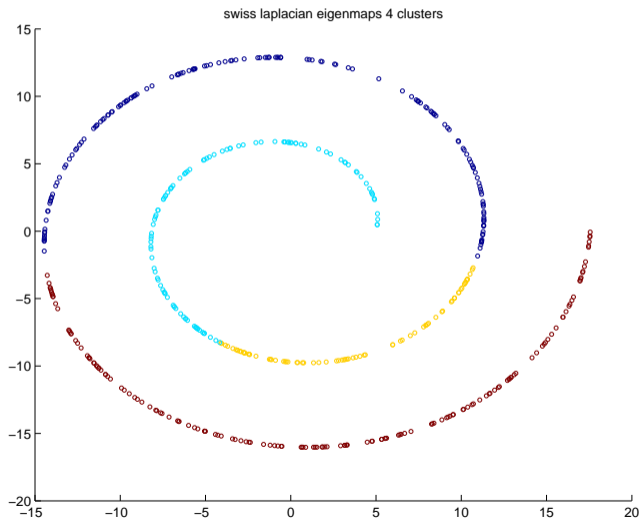
Swiss Role



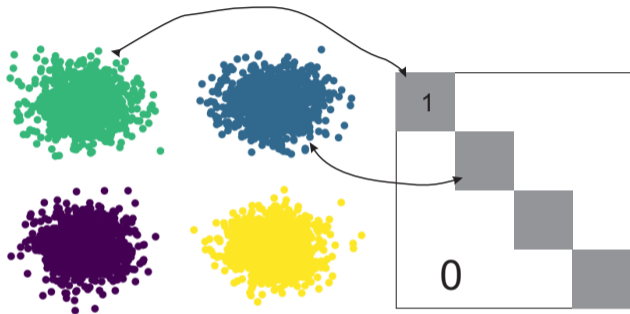
Swiss Role



Swiss Role



例



例

