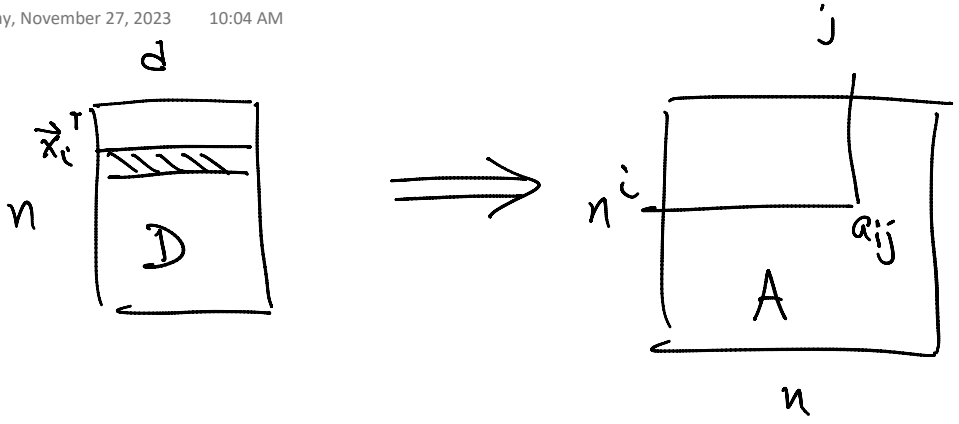
Lecture 23

Monday, November 27, 2023 10:04 AM



weighted adjacency matrix

similarity matrix

$n \times n$

symmetric

$a_{ij} = a_{ji}$

$a_{ij} \geq 0$

$A = \underbrace{K}$

kernel matrix

using Gaussian kernel

$$a_{ij} = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}}$$

↑ user chosen variance value

Markov matrix

$$M = \underbrace{\Delta^{-1} A}$$

Δ = degree matrix

$$\Delta = \begin{bmatrix} d_1 & & 0 \\ & d_2 & \\ & & \ddots \\ 0 & & d_n \end{bmatrix}$$

$$d_i = \underbrace{\sum_{j=1}^{n} a_{ij}}$$

sum of each row

$$M = \begin{bmatrix} 1/d_1 & & 0 \\ & 1/d_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \end{bmatrix}$$

$M = \begin{bmatrix} 1/d_1 & & \\ & 1/d_2 & \\ & & \ddots \\ & & & 1/d_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \end{bmatrix}$

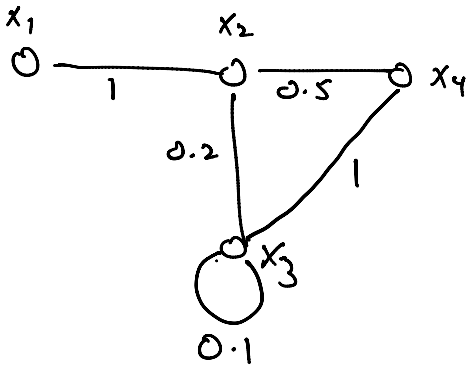
sum of each row

Δ^{-1}

$$M = \begin{bmatrix} a_{11}/d_1 & a_{12}/d_1 & \cdots & a_{1n}/d_1 \\ a_{21}/d_2 & a_{22}/d_2 & \cdots & a_{2n}/d_n \\ \vdots & & & \end{bmatrix}$$

$= \{ m_{ij} \}_{i,j=1\ldots n}$

m_{ij} is the probability of 'jumping from i to j'



each row is a probability vector

$m_{ij} \geq 0$

$\sum_j m_{ij} = 1$ ← for each row

	x_1	x_2	x_3	x_4
x_1	0	1	0	0
x_2	1	0	0.2	0.5
x_3	0	0.2	0.1	1
x_4	0	0.5	1	0

A

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 0 & 1.3 & 0 \\ 0 & 0 & 0 & 1.5 \end{bmatrix}$$

degree

Sim

$M = \Delta^{-1} A =$

0	1	0	0
0.59	0	0.12	0.29

Inflate $r = 2$

0	1^2	0	0

0	1	0	0

$M = \Delta A =$

0.59	0	0.12	0.29
0	0.15	0.08	0.77
0	0.33	0.67	0

0	1	0	0
0	$\frac{1}{3}$	$\frac{2}{3}$	0

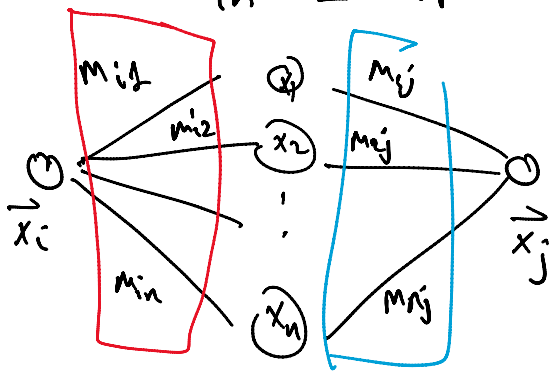
0	1	0	0
0	$\frac{1}{9}$	$\frac{4}{9}$	1

0.2 0.8

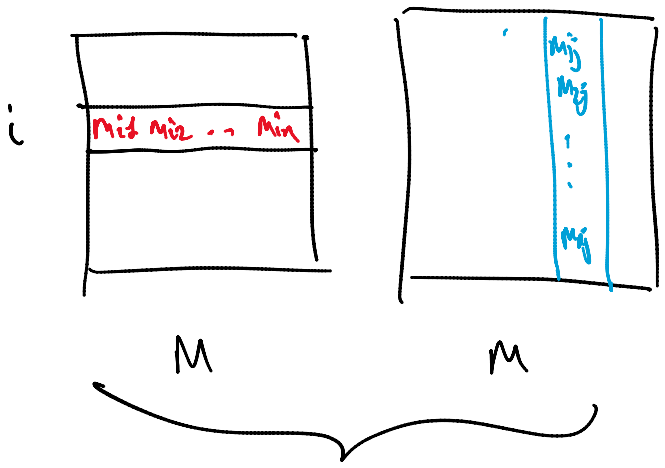
$\frac{1}{9}$ / $\frac{5}{9}$ $\frac{4}{9}$ / $\frac{5}{9}$

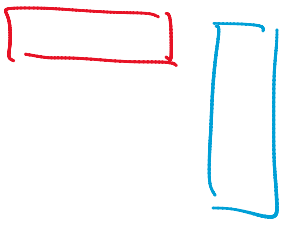
$M \equiv$ 1 – step transition matrix

Q: what is the prob of ending up on $\vec{x}_j$ starting from $\vec{x}_i$ in 2 steps



independent of step or time (homogeneous)





$= M_{ij}^{(2)}$ prob from i to j in 2 steps

$M^2 \equiv$ 2 – step transition matrix

$M^t \equiv$ t – step transition matrix

Π is a initial prob vector

= initial prob of starting on node i (page i)

Col vector

⇓

what's the prob of ending up on j (buy j)

$$\vec{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.1 \\ 0 \\ 0 \\ 0.5 \\ \vdots \end{bmatrix}$$

$\Sigma = 1$

$\vec{x}_0 = \vec{\pi}$

$M\vec{x}_0 = \vec{x}_1$

$M\vec{x}_1 = \vec{x}_2$

$M(M\vec{x}_0) = \vec{x}_2$

$M^2 x_0 = \vec{x}_2$

⋮

Converge to dominant eigen vector of M

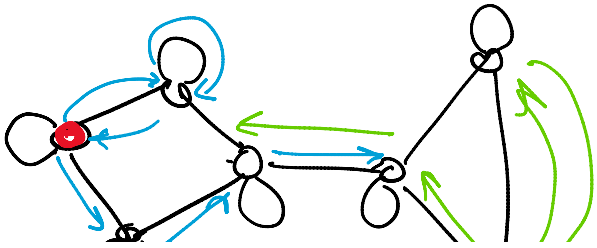
$\pi, x_1, x_0 \in \mathbb{R}^n$

final "destination" probabilities are independent of the initial prob vector

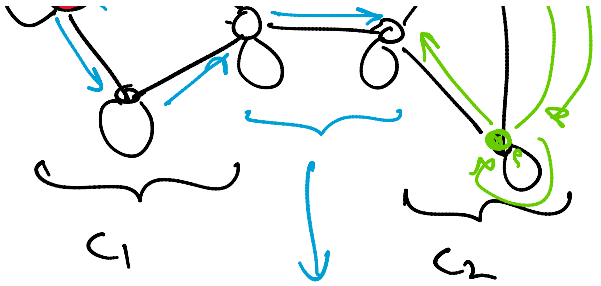
---

Q: how to use M matrix for clustering

random walk



all nodes will have self-loops (for clustering)



(for clustering)

[prob of staying in C_1 is high] — weak Similarity edge — [prob of staying in C_2 is high]

MCL: Markov chain clustering

Starting with $M \leftarrow$ 1-step transition

a) walk one more step

$$\underline{M} = M \cdot M = M^2$$

b) Inflate M (prob that are high should be made even higher & prob that are low " " " even lower)

$r \equiv$ inflation parameter

[i) Compute r-th power of each element of M
ii) normalize]

$$M_{ij} = \frac{M_{ij}^{(r)}}{\sum_{j=1}^{n} M_{ij}^{(r)}}$$

(each row should remain prob vector)

c) repeat a) & b) until Convergence

c) repeat a) & b) until Convergence

$$\| M^{(t)} - M^{(t-1)} \|_F \leq \varepsilon \quad \text{Stop}$$

(t) step number

sqrt (sum of difference)

d) extract clusters

Using the final M we create a directed graph G

Sparse

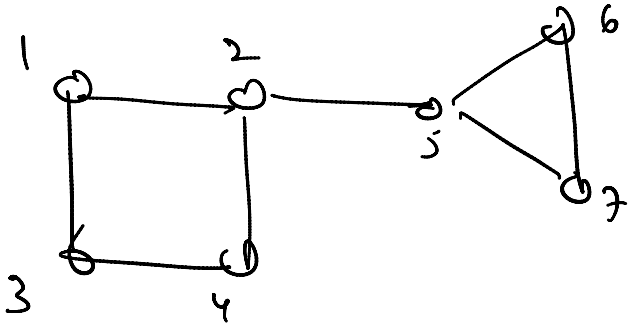
$$E = \{ (i,j) \mid M(i,j) > 0 \}$$

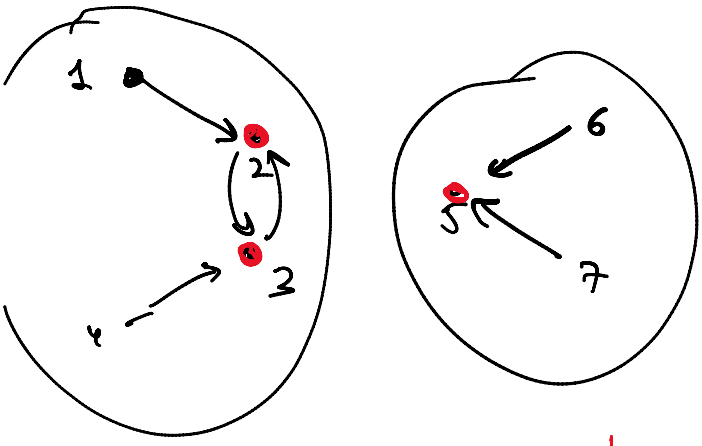
find weakly connected components in G

If $M(i,j) > 0 \Rightarrow$ i can transition to j

$M_{ij} > 0$

we also say that j is an attractor for i





Weakly Connected Components

γ parameter

γ is large

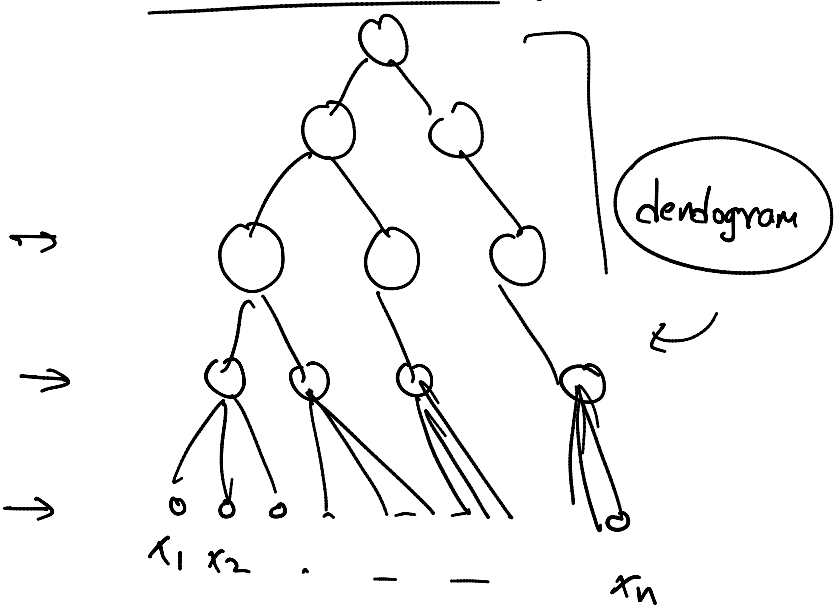
↳ many clusters, small

γ is small

↳ fewer, larger clusters

$\gamma \geq 1$

Hierarchical clustering



K-mean ← hard

EM ← soft

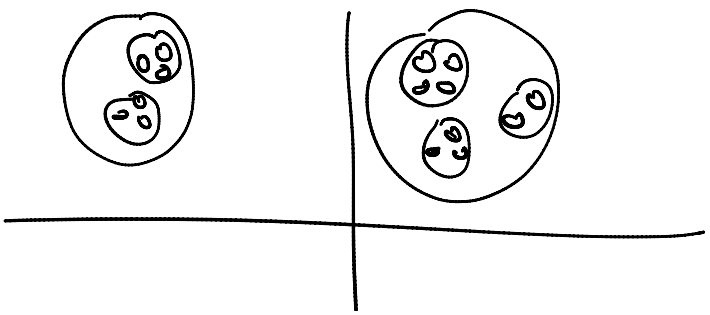
density ← non-convex

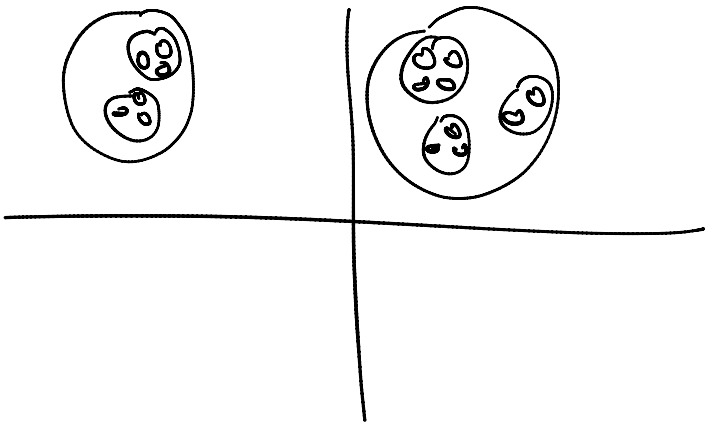
Overlap

spectral → hard

MCL → overlap

flat





Agglomerative hierarchical clustering ($K \leftarrow$ find K clusters)

1) $\mathcal{C} = \{ x_i \text{ in its own cluster} \}$

$= \{ C_1 \; C_2 \; - \; - \; - \; C_n \}$ $\qquad C_i = \{ \vec{x_i} \}$

---

repeat until $|\mathcal{C}| = K$ (or $|\mathcal{C}| = 1$) → full tree

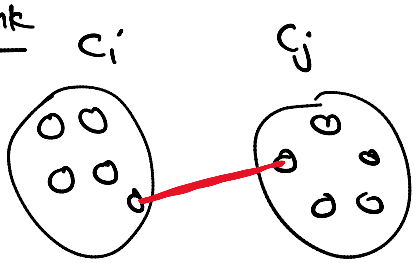
- find the closest pair of clusters C_i & C_j and merge
- $C_{ij} = C_i \cup C_j$ $\qquad \Omega(n^2)$
- to $\mathcal{C}$ add C_{ij} and remove C_i and C_j $\qquad O(n^2 \log n)$
- update distances from C_{ij} to all other C_k

$\delta(C_{ij}, C_k) \quad \forall \; k$

$\delta(C_i, C_j) \quad \forall \; i, j$

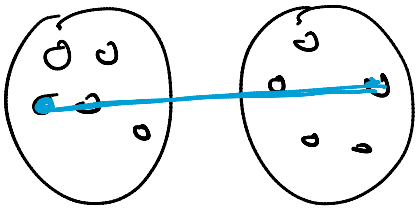
→ we need to define what is the distance between two clusters

Single link



$$\delta(c_i, c_j) = \min \{ \|\vec{x} - \vec{y}\| \}$$
$$\vec{x} \in c_i$$
$$\vec{y} \in c_j$$

Complete link



$$\delta(c_i, c_j) = \max \{ \|\vec{x} - \vec{y}\| \}$$
$$\vec{x} \in c_i$$
$$\vec{y} \in c_j$$

Intuitively the farthest pair includes all other pair

Group average

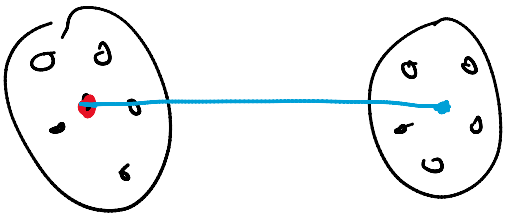
$$\delta(c_i, c_j) = \operatorname{avg} \{ \|\vec{x} - \vec{y}\| \} = \frac{\sum_{x}^{c_i} \sum_{y}^{c_j} \|\vec{x} - \vec{y}\|}{n_i \cdot n_j}$$
$$\forall \vec{x} \in c_i$$
$$\vec{y} \in c_j$$

$$n_i = |c_i|$$

mean distance

$$\delta(c_i, c_j) = \|\vec{\mu_i} - \vec{\mu_j}\|$$

$$\vec{\mu_i} = \sum_{\vec{x_j} \in c_i} \vec{x_j} \Big/ n_i$$

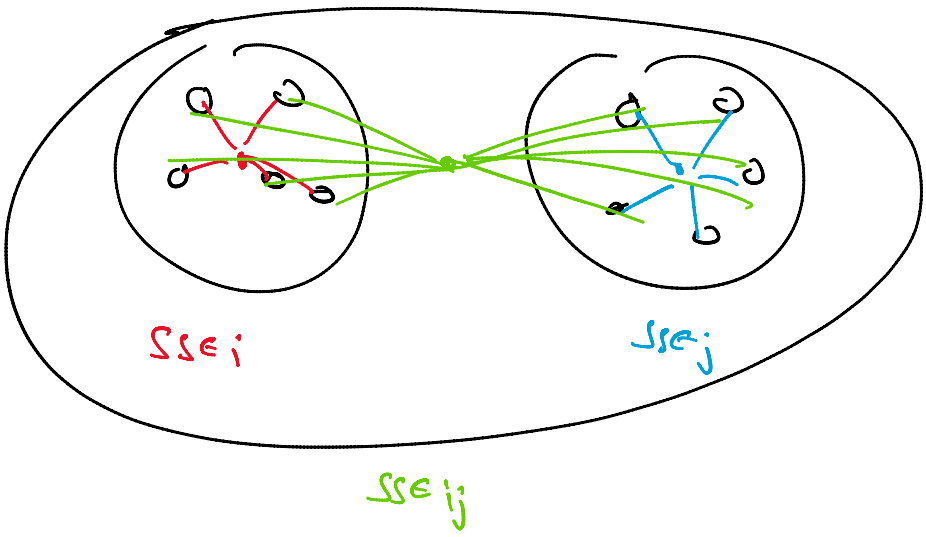


Minimum Variance (Wards Method)

$$\delta(C_i, C_j) = \underbrace{\Delta SSE_{ij}}_{\text{Change in squared error}}$$

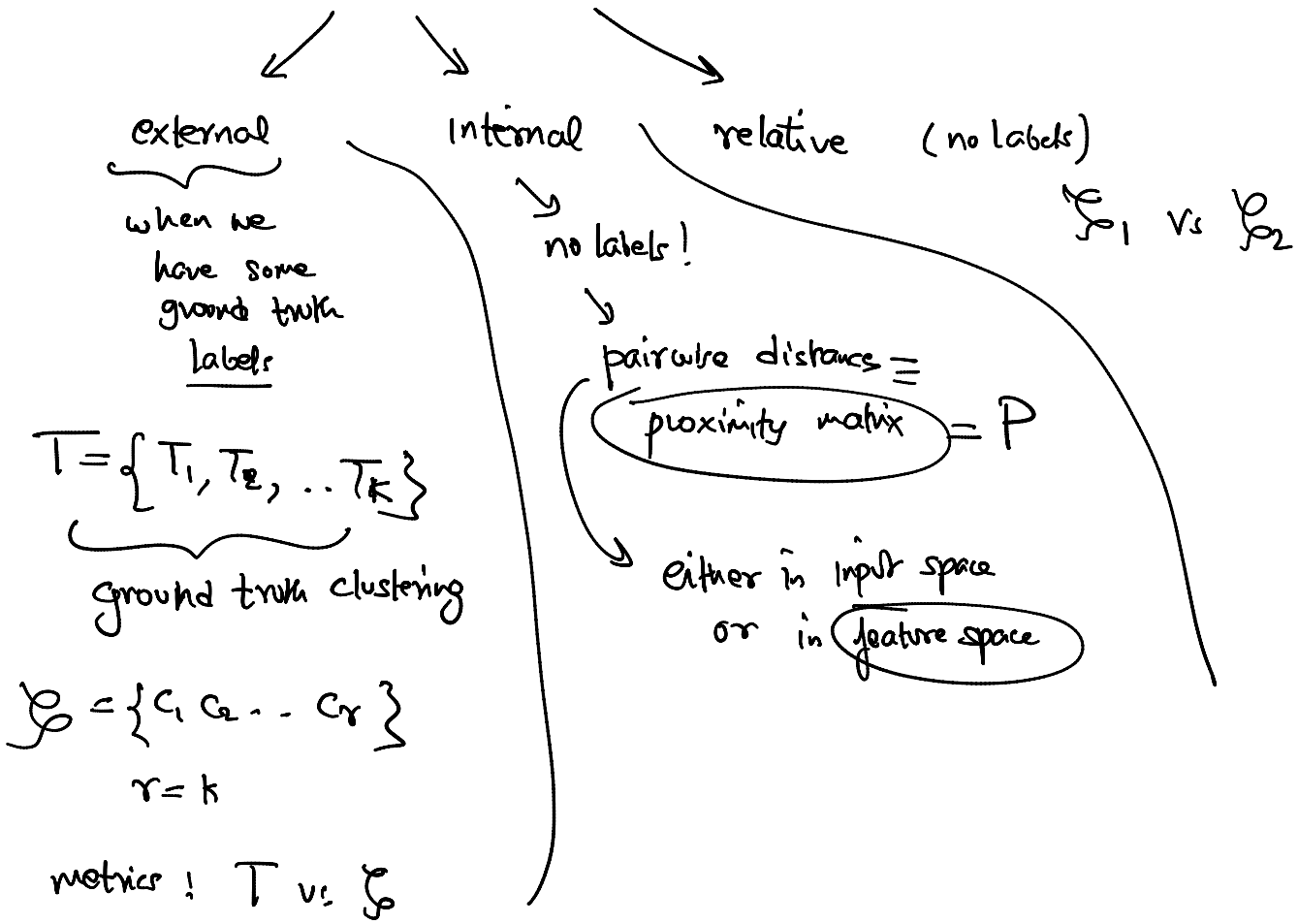
$$= SSE_{ij} - SSE_i - SSE_j = \left(\frac{n_i n_j}{n_i + n_j}\right) \| \vec{\mu}_i - \vec{\mu}_j \|^2$$

$$SSE_i = \sum_{\vec{x}_j \in C_i} \| \vec{x}_j - \vec{\mu}_i \|^2$$



Evaluate Clusters?

↳ unsupervised


external
when we have some ground truth labels
$T = \{T_1, T_2, \ldots T_k\}$
ground truth clustering
$\zeta = \{C_1 C_2 \ldots C_r\}$
$r = k$
metrics : T vs ζ
internal
no labels !
pairwise distances ≡
proximity matrix = P
either in input space
or in feature space
relative (no labels)
ζ_1 vs ζ_2