# Unsupervised Learning

# Unsupervised Learning

- Given feature vectors $x_1, \dots, x_n$
  No labels

Clustering: group "similar" objects together

## Main Ingredients

1) feature vectors $\underline{x}_j \in X$
2) # of clusters $k$
3) similarity or dissimilarity measure

Similarity $\longrightarrow$ $s(x, x')$
dissimilarity $\longrightarrow$ $d(x, x')$

Common conversion: $s(x, x') = \frac{e^{-d(x,x')}}{2\pi^2}$

$\|x - x'\|^2 = \|x\|^2 + \|x'\|^2 - 2\langle x, x' \rangle$

If feature vecs normalized to unit length
$\|x_j\|_2 = 1 \quad \forall j$, $s(x, x') := \langle x, x' \rangle$ then

$$d(x, x') = \|x - x'\|^2 = 1 + 1 - 2\langle x, x' \rangle$$
$$= 2 - 2s(x, x')$$
$$\boxed{d(x, x') = 2(1 - s(x, x'))}$$

- Feature based Clustering
  ↳ Works with features $x_1, \dots, x_n$

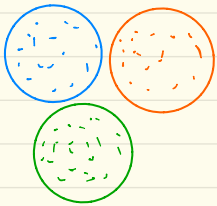
- Similarity-based Clustering
  ↳ works with the pairwise similarity matrix $(n \times n)$
  $[s(x_i, x_j)]$

**Given** : $x_1, \ldots, x_n \in \mathbb{R}^d$

**Goal** : Infer labels $y_1, \ldots, y_n \in \{1, k\}$

## Generative model for clustering

(not given colors)



$$p(x, y \mid \theta) = \underbrace{\frac{1}{k}}_{p(y)} \mathcal{N}(\mu_y, \sigma^2 I_d)(x)$$

$p(y) = \frac{1}{k} \quad \forall y$ (uniform)

$$p(x \mid y, \theta) = \mathcal{N}(\mu_y, \sigma^2 I_d)(x)$$

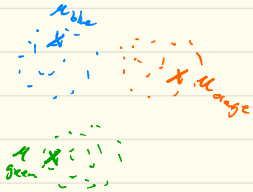
$$\theta = (\mu_1, \ldots, \mu_k, \sigma^2)$$

ML estimates of labels $y_1, \ldots, y_n$ & parameters $(\mu_1, \ldots, \mu_k)$

$$\begin{pmatrix} \hat{y}_1, \ldots, \hat{y}_k \\ \hat{\mu}_1, \ldots, \hat{\mu}_k \end{pmatrix} = \underset{\substack{y_1, \ldots, y_n \in \{1, k\} \\ \mu_1, \ldots, \mu_k \in \mathbb{R}^d}}{\operatorname{argmax}} \prod_{j=1}^{n} \underbrace{p(x_j \mid y_j, \theta)}_{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|x_j - \mu_j\|^2}{2\sigma^2}}}$$

$$= \underset{\substack{y_1, \ldots, y_n \\ \mu_1, \ldots, \mu_k}}{\operatorname{argmin}} \sum_{j=1}^{n} \|x_j - \mu_{y_j}\|_2^2$$

Within-cluster sum of squares (WCSS)



# K-means clustering (Lloyd's algorithm - 1957 Bell Labs)

**Initialize**: $\mu_1, \ldots, \mu_k$

For $t = 1$ until stopping condition

**Assign Labels**: $\forall j,\ y_j = \underset{\ell \in \{1,\ldots,k\}}{\operatorname{argmin}} \| x_j - \mu_\ell \|_2^2$

(For all $j$)

**Update Centers**: $\forall \ell,\ \mu_\ell = \frac{1}{|\{j : y_j = \ell\}|} \sum_{j : y_j = \ell} x_j$

end for

## Alternative way

For $t = 1$ until stopping condition

$$\forall j,\ y_j^{(t)} = \underset{\ell \in \{1,\ldots,k\}}{\operatorname{argmin}} \left\| x_j - \frac{1}{|\{j : y_j^{(t-1)} = \ell\}|} \sum_{j : y_j^{(t-1)} = \ell} x_j \right\|_2^2$$

end for

$$\left\| x_j - \frac{1}{n_\ell} \sum_{j : y_j^{(t-1)} = \ell} x_j \right\|^2 = \left\langle x_j - \frac{1}{n_\ell} \Sigma x,\ x_j - \frac{1}{n_\ell} \Sigma x \right\rangle$$

$$= \langle x_j, x_j \rangle - \frac{2}{n_\ell} \sum_{j'} \langle x_j, x_{j'} \rangle + \frac{1}{n_\ell^2} \sum \sum \langle x_j, x_{j'} \rangle$$

Replacing $\langle x, x' \rangle$ with $k(x, x')$ gives us the

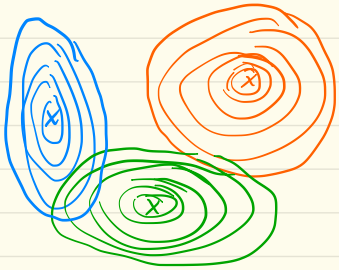
Kernel k-means Algorithm

## Initialization

1) Randomly (uniformly, without replacement) choose $k$ points among training set
2) Randomly assign labels (uniformly over $1, \dots, k$ to each training point.
3) k-means ++ (farthest point clustering)
   - 1st point picked uniformly at random
   - sequentially pick from remaining points with the probability which is proportional to squared distance to closest cluster point among cluster points chosen thus far.

k-means ++ is guaranteed to obtain a within-cluster sum of squares which is not more than $O(\log k)$ of minimum possible WCSS.

## Mixture of Gaussians (GMM)

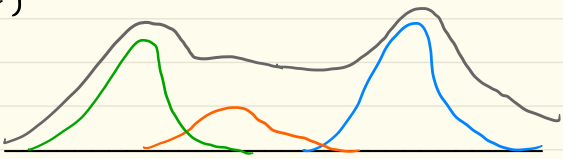


$$p(x, y \mid \theta) = \underbrace{\lambda_y}_{p(y \mid \theta)} \underbrace{N(\mu_y, \Sigma_y)(x)}_{p(x \mid y, \theta)}$$

$$\theta = \{(\lambda_y, \mu_y, \Sigma_y), y = 1, \dots, k\}$$

$$\Rightarrow p(x \mid \theta) = \sum_{y=1}^{k} p(x, y \mid \theta)$$

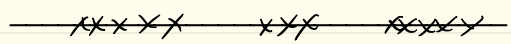
$$= \lambda_1 N(\mu_1, \Sigma_1)(x) + \lambda_2 N(\mu_2, \Sigma_2)(x) + \vdots + \lambda_k N(\mu_k, \Sigma_k)(x)$$

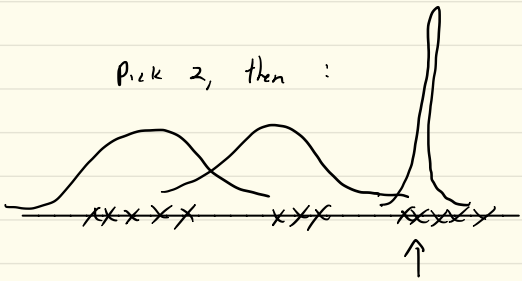


Idea of soft membership to clusters

$\{ P(Y_j = \ell \mid x_j, \theta),\ \ell = 1, \ldots, k \}$ Prob. of membership to cluster $\ell$

given points,






Finding ML estimates of $\theta$ is ill-posed:

Can make likelihood go to $\infty$ by choosing a $\delta$ function for one of the components centered at one of the data points.

$\Rightarrow$ need a prior for $\theta$, specifically for $\Sigma_1, \ldots, \Sigma_k$

## E-M algorithm (Expectation Maximization)

Idea $(X, Y) \sim p(x, y \mid \theta)$, $\theta \sim \pi(\theta) \rightarrow$ prior

Complete data: $(x, y)$

Incomplete data: $x$

Ideally: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} [\ln p(\overset{\downarrow}{x}, y \mid \theta) + \ln \pi(\theta)]$

but, $Y$ is not given, only $X$ is (latent, hidden)

$$\int \ln \left( p(\mathbb{X}, \mathbb{Y} \mid \theta) \pi(\theta) \right) p(\mathbb{Y} \mid \mathbb{X}, \theta^{t-1}) \, d\mathbb{Y}$$

Solution:

"average-out" $Y$ in cost, then maximize over $\theta$

- But "arg-out" requires $P(Y \mid X, \theta)$
- Use previous $\theta$ to do arg'ing (expectation step) then maximize over $\theta$ (M-step)

Expectation step (E-step)

Replace $\ln p(x, y \mid \theta) + \ln \pi(\theta)$ with $E\left[\ln p(x, y \mid \theta) + \ln \pi(\theta) \,\middle|\, X = x, \theta = \theta\right]$

$Q(\theta, \theta^{(t-1)})$

Maximization step (M-step)

$$\theta^{(t)} = \underset{\theta}{\operatorname{argmax}} \; Q(\theta, \theta^{(t-1)})$$

Geometric

$$p(x \mid \theta^{t-1}) \pi(\theta^{t-1}) \leq p(x \mid \theta^{t}) \pi(\theta^{t})$$

unconditioned likelihood

## E-M algorithm

Initialize $\hat{\theta}_0$

For $t = 1, \ldots, t_{max}$ or stopping cond.

E-step: $Q(\theta, \hat{\theta}_{t-1}) = E_{\bar{Y}}\left[ \ln\left( p(\underline{X}, \bar{Y} \mid \theta)\, \pi(\theta) \right) \middle| \underline{X}, \hat{\theta}_{t-1} \right]$

M-step: $\hat{\theta}_t = \underset{\theta}{\operatorname{argmax}}\ Q(\theta, \hat{\theta}_{t-1})$

end for

- $E_{\bar{Y}}\left[ \ln\left( p(\underline{X}, \bar{Y} \mid \theta)\, \pi(\theta) \right) \mid \underline{X}, \hat{\theta}_{t-1} \right] = \int \ln\left( p(\underline{X}, \bar{Y} \mid \theta)\, \pi(\theta) \right) \cdot p(\bar{Y} \mid \underline{X}, \hat{\theta}_{t-1})\, d\bar{Y}$
- key property

$$p(\underline{X} \mid \hat{\theta}_{t+1})\, \pi(\hat{\theta}_{t+1}) \geq p(\underline{X} \mid \hat{\theta}_t)\, \pi(\hat{\theta}_t)$$

## GMM

$$p(x, y \mid \theta) = \underbrace{\lambda_y}_{p(Y=y \mid \theta)}\ \underbrace{N(\mu_y, \Sigma_y)(x)}_{p(x \mid Y=y, \theta)}$$

$\theta = \{ (\lambda_y, \mu_y, \Sigma_y),\ y = 1, \ldots, k \}$

$\underline{X} = (\underline{x}_1, \ldots, \underline{x}_n)$

$\bar{Y} = (y_1, \ldots, y_n)$ → cluster labels (hidden)

$$p(\mathbf{x}, \mathbf{y} \mid \theta) = \prod_{j=1}^{n} p(x_j, y_j \mid \theta)$$

$(\pi(\theta) \sim \text{Unif.})$

(no prior)

$$= \prod_{j=1}^{n} \lambda_{y_j} \mathcal{N}(\mu_{y_j}, \Sigma_{y_j})(x_j)$$

$$= \prod_{j=1}^{n} \prod_{\ell=1}^{k} \left( \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)(x_j) \right)^{\mathbb{1}(y_j = \ell)}$$

$$\ln p(\mathbf{x}, \mathbf{y} \mid \theta) = \sum_{j=1}^{n} \sum_{\ell=1}^{k} \mathbb{1}(y_j = \ell) \ln\left( \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)(x_j) \right)$$

E-step $\quad E_{\mathbf{y}}\left[ \ln p(\mathbf{x}, \mathbf{y} \mid \theta) \mid \mathbf{x}, \hat{\theta}_{t-1} \right]$

$$= \sum_{j=1}^{n} \sum_{\ell=1}^{k} E\left[ \mathbb{1}(y_j = \ell) \mid x_j = x_j, \hat{\theta}_{t-1} \right] \cdot \ln\left( \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)(x_j) \right)$$

$$E\left[ \mathbb{1}(y_j = \ell) \mid x_j = x_j, \hat{\theta}_{t-1} \right]$$

$$=: \gamma_{\ell j}^{t-1}$$

$$\sum_{j=1}^{n} \sum_{\ell=1}^{k} \gamma_{\ell j}^{t-1} \ln\left( \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)(x_j) \right)$$

M-step $\quad$ maximize over

$\theta = \{(\lambda_\ell, \mu_\ell, \Sigma_\ell)\}$ $\quad$ similar to GDA with weights $\gamma_{\ell j}^{(t-1)}$

$$\hat{\lambda}_\ell^{(t)} = \frac{1}{n} \sum_{j=1}^{n} \gamma_{\ell j}^{(t-1)}$$

$$\hat{\mu}_\ell^{(t)} = \frac{\sum_{j=1}^{n} \gamma_{\ell j}^{(t-1)} x_j}{\sum_{j=1}^{n} \gamma_{\ell j}^{(t-1)}}$$

$$\hat{\Sigma}_\ell^{(t)} = \frac{\sum_{j=1}^{n} \gamma_{\ell j}^{(t-1)} (x_j - \hat{\mu}_\ell^{(t)})(x_j - \hat{\mu}_\ell^{(t)})^T}{\sum_{j=1}^{n} \gamma_{\ell j}^{(t-1)}}$$

$$\gamma_{\ell j}^{(t)} = P(y_j = \ell \mid x_j = x_j, \theta_t)$$

$$= \frac{\hat{\lambda}_\ell^{(t)} \mathcal{N}(\hat{\mu}_\ell^{(t)}, \hat{\Sigma}_\ell^{(t)})(x_j)}{\sum_{\ell=1}^{k} \hat{\lambda}_\ell^{(t)} \mathcal{N}(\hat{\mu}_\ell^{(t)}, \hat{\Sigma}_\ell^{(t)})(x_j)}$$

★ Spectral Clustering Tutorial on class webpage

(+ pseudocode)

# Spectral Clustering

- Graph-cut based clustering algorithms

Similarity score : high => similar  Can use to form a weighted or unweighted Graph

Gaussian kernal:

$$s(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$$

another is : $s(x_i, x_j) = \begin{cases} 1 & \text{if } \|x_i - x_j\| < \xi \\ 0 & \text{else} \end{cases}$

another : (kNN) $s(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \in NN \text{ of } x_j \text{ OR } x_j \in NN \text{ of } x_i \\ 0 & \text{else} \end{cases}$

another : " " " AND "

## How to cut the graph

### Graph terminology

- similarity graph on $n$ nodes [node ≡ feature/training vector]
- Weighted adjacency matrix

$W_{n \times n}$, $w_{ij} = s(x_i, x_j)$

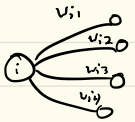
$$\begin{array}{c} \\ i \\ \\ n \end{array} \begin{array}{c} \begin{array}{ccc} & j & \\ & \vdots & \\ \cdots & w_{ij} = s(x_i, x_j) & \cdots \end{array} \\ \begin{array}{c} \\ \\ \end{array} \end{array} \begin{array}{c} \\ \\ n \end{array}$$

Assume symmetric $s(\cdot,\cdot)$

$s(x, x') = s(x', x)$

$\Rightarrow W = W^T$

### Node degree




degree of node $i$ : $d_i := \sum_{j=1}^{n} w_{ij}$

### Degree Matrix

$$D_{n \times n} := \text{diag}(d_1, \ldots, d_n) = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$

## Graph Laplacian $L_{n \times n}$

$$L := D - W$$

### Properties of $L$

1) If $\underline{f} \in \mathbb{R}^n$, $\underline{f} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}_{n \times 1}$

$$\underbrace{\underline{f}^T L \underline{f}}_{1 \times 1} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (f_i - f_j)^2$$

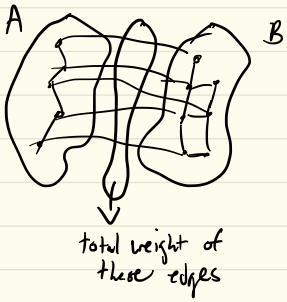
2) $L = L^T$, symmetric

3) (1) $\Rightarrow L \geq 0$ positive semidefinite

$Cut(A, B)$ $\qquad CUT(A, B) := \sum_{i \in A} \sum_{j \in B} w_{ij}$

$A \cap B = \{\}$

$A, B = \{1, \ldots, n\}$



## $k$ clusters

$A_1, A_2, \ldots A_k$ = position of nodes for the $k$ adjacent graphs

if $i \neq j$, $A_i \cap A_j = \{\}$,

$\bigcup_{i=1}^{k} A_i = \{1, \ldots, k\}$

$A^c = \{j : j \notin A\} = \{1, \ldots, n\} \setminus A$

Ratio Cut : $\sum_{l=1}^{k} \frac{Cut(A_l, A_l^c)}{|A_l|}$

$|A_l| = |\{j : j \in A_l\}|$ = # nodes in cluster $l$

*

$F^{\text{Ratio Cut}} = [f_1, \ldots, f_k]_{n \times k}$

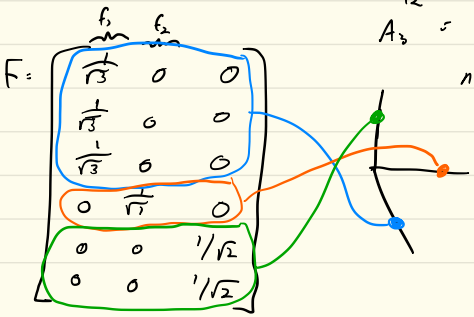
$F_{il} = \begin{cases} \frac{1}{\sqrt{|A_l|}} & \text{if } j \in A_l \\ 0 & \text{else} \end{cases}$

$F = \left[ \frac{1_{A_1}}{\sqrt{|A_1|}} \quad \cdots \quad \frac{1_{A_k}}{\sqrt{|A_k|}} \right]$

e.g. Suppose $A_1 = \{1, 2, 3\}$, $A_2 = \{4\}$, $A_3 = \{5, 6\}$, $n = 6$, $k = 3$

$F =$

| | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| | $\frac{1}{\sqrt{3}}$ | 0 | 0 |
| | $\frac{1}{\sqrt{3}}$ | 0 | 0 |
| | $\frac{1}{\sqrt{3}}$ | 0 | 0 |
| | 0 | $\frac{1}{\sqrt{1}}$ | 0 |
| | 0 | 0 | $1/\sqrt{2}$ |
| | 0 | 0 | $1/\sqrt{2}$ |



**Results:** $\underset{k\times n}{F^T}\underset{n\times k}{F} = I_k$

Ratio Cut $(A_1, A_2, \ldots A_k) = \sum_{\ell=1}^{k} \underbrace{f_\ell^T L f_\ell}_{1\times 1}$

Recall $abL$ = Trace $(abc)$
(scalar) = $Tr(bca)$
= $Tr(cab)$

$f_\ell^T L f_\ell = Tr(f_\ell^T L f_\ell) = Tr(L f_\ell f_\ell^T)$

$\Rightarrow \sum_\ell f_\ell^T L f_\ell = \sum_\ell Tr(L f_\ell f_\ell^T) = Tr(L \sum_\ell f_\ell f_\ell^T)$

$= Tr(L F F^T) = Tr(F^T L F)$

Ratio Cut minimization for clustering

$= \underset{A_1,\ldots,A_k}{\text{argmin}}$ Ratio Cut$(A_1, \ldots, A_k)$

$= \underset{F \text{ of form } *}{\text{argmin}} Tr(F^T L F)$

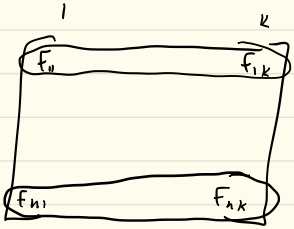
$\underset{F^TF=I}{\text{argmin}} Tr(F^T L F)$ ← relax requirement of $F$

# Solution to Relaxed Ratio Cut Problem

→ Rayleigh-Ritz Theorem

cols of best F

≡ orthonormal eigenvectors of the k SMALLEST eigenvalues of L

$F^{best}$ = 


→ cluster the rows of $F^{best}$ into k groups using k-means

Intuition

D = degree matrix

V = F

SC-2

RW: random walk or simple pagerank

NCut

$$NCut(A_1 .. A_K) = \sum_{\ell=1}^{K} \frac{Cut(A_\ell, A_\ell^c)}{Vol(A_\ell)}$$

$$Vol(A_\ell) = \sum_{j \in A_\ell} d_j$$

$$WCSS(\underset{\mu_1, \ldots, \mu_k}{y_1, \ldots, y_n}) = \sum_{j=1}^{n} \| x_j - \mu_{y_j} \|_2^2$$

- k-means algorithm
- k-means ++
- GMMs → E-M algorithm
- spectral methods:

$$\underset{n \times n}{W} \rightarrow \text{weighted adjacency matrix}$$

$w_{ij} \equiv$ similarity between $x_i, x_j$

Graph Laplacian : $L = D - W$

$D = \text{diag}(W 1_n)$

$A_1, A_2, \ldots, A_k \rightarrow$ k clusters (set of nodes)

$$\text{RatioCut}(A_1, \ldots, A_k) = \sum_{l=1}^{k} \frac{\text{cut}(A_l, A_l^c)}{|A_l|}$$

$$* \quad \underset{n \times k}{F} = \left[ \frac{1_{A_1}}{\sqrt{|A_1|}}, \ldots, \frac{1_{A_k}}{\sqrt{|A_k|}} \right] \qquad F^T F = I_k$$

$$\text{RatioCut}(A_1, \ldots, A_k) = Tr(F^T L F)$$

RatioCut minimization

$$\equiv \underset{A_1, \ldots, A_k}{\text{argmin}} \ \text{RatioCut}(A_1, \ldots, A_k)$$

$$= \underset{F \text{ of form } *}{\text{argmin}} \ Tr(F^T L F)$$

## Relaxed Ratio Cut minimization

$$\underset{F:\, F^TF = I_k}{\operatorname{argmin}} \quad Tr(F^TLF)$$

Solution: Rayleigh-Ritz Theorem

cols of $F_{\substack{best \\ (relaxed)}}$ = $k$ o.n. eigenvalues of $L$ correspond to the $k$ smallest eigenvalues of $L$

$$L = U \Lambda U^T = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} - & u_1^T & - \\ & \vdots & \\ - & u_n^T & - \end{bmatrix}$$

$$0 \le \lambda_1 \le \cdots \le \lambda_n$$

$$F_{\substack{best \\ (relaxed)}} = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{bmatrix}$$

WCSS $(y_1 \cdots y_n, \mu_1 \cdots \mu_k)$

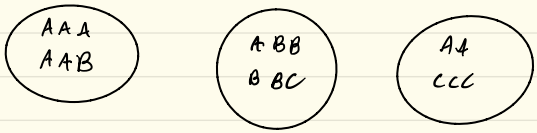
$$= \underset{F \text{ of form } *}{\operatorname{argmin}} \; Tr(F^T \underbrace{(I-K)}_{\substack{n \times n \\ \text{identity}}} F)$$

$K = X^TX$, $X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix}$

# Metrics

(for clustering when given a reference clustering)

1) Purity




dominant label counts

$$\frac{5 + 4 + 3}{6 + 6 + 5}$$

Formally, $n_{ij}$ = # class $j$ points in cluster # $i$

$$n_i = \max_j n_{ij}, \quad \text{purity} = \frac{\sum_{i=1}^{k} n_i}{n}$$

2) Normalized mutual information (NMI)

$$P_{uv}(i,j) = \frac{n_{ij}}{n}$$

$$P_u(i) = \sum_j P_{uv}(i,j)$$

$$P_v(j) = \sum_i P_{uv}(i,j)$$


Mutual Information

$$I(u,v) := D(P_{uv} \,//\, P_u \cdot P_v)$$

$$NMI = \frac{I(u;v)}{(H(u)+H(v))/2} \qquad H = \text{entropy}$$

$\in [0,1]$

## If K is not known

1) penalize # clusters: $\underset{\substack{y_1,\dots,y_n \\ \mu_1,\dots,\mu_k,\, k}}{\operatorname{argmin}} \; WCSS(y_1,\dots,y_n,\mu_1,\dots,\mu_k) + \lambda k$

($\lambda$: regularization parameter)

2) DP-means (DP = Dirichlet Process means)

- like k-means
- Idea: if there exists a point whose distance to closest $\mu_1,\dots,\mu_k$ is greater than some distance $\gamma$, then $k \to k+1$ & begin new cluster.

3) Sum of Norms (SON) clustering

$$\underset{\mu_1,\dots,\mu_n}{\operatorname{argmin}} \left[ \sum_{j=1}^{n} \| x_j - \mu_j \|_2^2 + \lambda \sum_{j=2}^{n} \sum_{i<j} \| \mu_i - \mu_j \|_p \right]$$

4) Chinese Restaurant Process