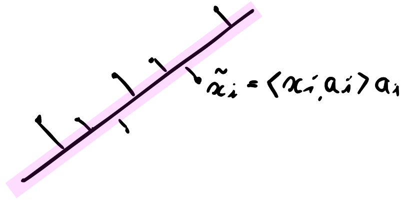
Unsupervised

I. Principal Component Analysis

- The 1^{st} principal axis is a direction maximizing the variance of projections.



- 1^{st} principal axis is solution to:

$$\max_{a \in \mathbb{R}^p} a^T V a \quad \text{s.t. } \|a\| = 1$$

$$\mathcal{L}(a, \lambda) = a^T V a - \lambda(\|a\|^2 - 1)$$

↳ $a^T V a = a^T \lambda a = \lambda \|a\|^2 = \lambda$ largest eigen-value.

Coordinates of x_i on projected principal axis a_1 are the 1^{st} principal component.

$C_{1,i} = \langle X_i, a_1 \rangle$

- 2^{nd} principal axis : $\max_a a^T V a$ s.t. $\|a\| = 1$, $a \perp a_1$

↳ $V a_2 = \lambda_2 a_2$. In large dimension $p \gg n$, $\frac{1}{n} K c = \lambda c$, $K_{i,j} = X_i^T X_j$

...

⇒ Reconstitution in the new base: $X_i = \sum_{k=1}^{r} C_{i,k}\, a_k$

Simplified representation in dimension $s \ll r$: $\tilde{X} = \tilde{C} \tilde{A}^T = (C_1 \cdots C_s)\begin{pmatrix} a_1 \\ \vdots \\ a_s^T \end{pmatrix}$

Explained variance by C_k $= \dfrac{\lambda_k}{\sum_{j=1}^{p} \lambda_j}$

Limitations:
- scale variant → Need standardization
- Complexity $O(p^3)$
- Might be impossible if p is too large.

- Kernel PCA

$\Phi(X)^T \Phi(X') = k(X, X')$

Solve $Ca = \lambda a$ where $C = \frac{1}{n} \sum \phi(X_i)\, \phi(X_i)^T$

Problem: $a_k = \frac{1}{n\lambda}(X^T C_k) = \frac{1}{n\lambda} \sum_{i=1}^{n} C_{i,k}\, \phi(X_i)$

In the standard case, c_1 is the linear combination of variables that maximize the sum of squared correlations of variables:

$$c_1 = \underset{c \in Vect(X^1 \dots X^p)}{\operatorname{argmax}} \sum_{j=1}^{p} R^2(X^j, c^j)$$

↳ Project on 2D correlation circle: $\cos \theta_{j,k} = \dfrac{\langle X^j, c^k \rangle}{\sqrt{X_j^T D X_j \cdot C_k^T D C_k^T}}$

angle between var. j and component k.

And length of line contribution of component c: $CTR(i, c) = \dfrac{p_i c_i^2}{\lambda}$

- Modified metric PCA: $Va = \lambda a$, $V = \langle x, x^T \rangle M \langle x, x^T \rangle$
- Modified weight PCA: $V = X^T D X - gg^T$, g barycenter, D matrix of weights.

II. Dependence Analysis

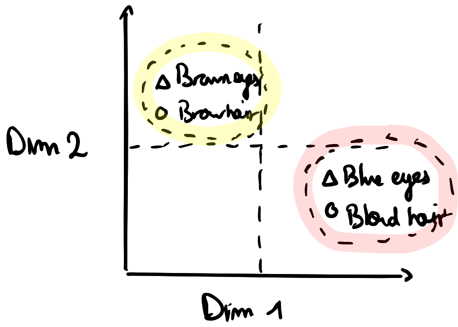
Used for discrete variables.

	Blue Eyes	Brown Eyes	Blond Hair	Brown Hair
Person 1	1		1	
Person 2	1			1
Person 3		1		1
...				

⇒

	Blond Hair	Brown Hair
Blue Eyes	16	8
Brown Eyes	4	29

Idea: We have 2 profiles (line, column) and we want to assess which categories of those profiles are correlated. It does however not say anything about intensity of relation.

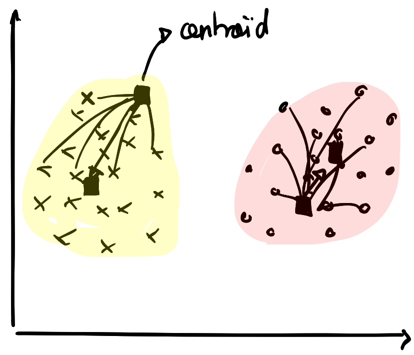


Correspondance analysis can be generalized.
↳ Multiple components > 2.

III. K-means

$$\min_{C_1 \ldots C_k} \sum_{j=1}^{k} \sum_{i \in C_j} \| X_i - \mu_j \|^2 \ , \quad \mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} X_i$$ center of the cluster.

Start from arbitrary locations of clusters
- Find partition induced by nearest cluster
- Update cluster center
- Iterate until cluster convergence



Can be expressed as:

$$\min \sum_{k=1}^{K} \sum_{X_i, X_j \in C_k} \| X_i - X_j \|^2$$ → minimize distance of points belonging to a cluster

$$\max \sum_{1 \leq k \neq l \leq K} \sum_{(X_i, X_j) \in C_k \times C_l} \| X_i - X_j \|^2$$ → maximize distance between points belonging to different clusters.

- k-means ++

Select initial clusters far away from each other (squared distance criteria)

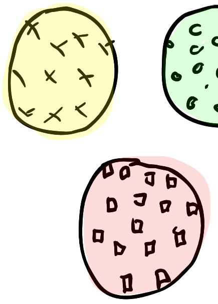
- Minibatch k-means

Update centroid using mini-batch of b samples chosen randomly
↳ Complexity $O(bk)$ instead of $O(nk)$

⇒ Always a finite number of configurations → these algorithms converge.

- **Soft clustering**

We suppose a gaussian distribution around a cluster. Each point has a probability $p_{ij} \propto \exp(-\|x_i-\mu_j\|^2/2\sigma^2)$ to belong to cluster j



Start from arbitrary location of cluster center
- Compute p_{ij}
- Update cluster center: $\mu_j = \dfrac{\sum_{i=1}^{n} p_{ij} x_i}{\sum_{i=1}^{n} p_{ij}}$
- Iterate until convergence

- **Fuzzy k-means**

Pareto (heavy tail) version of soft clustering: $p_{ij} \propto \dfrac{1}{\|x_i-\mu_j\|^{\alpha}}$, $\alpha > 0$

- **Weighted k-means**

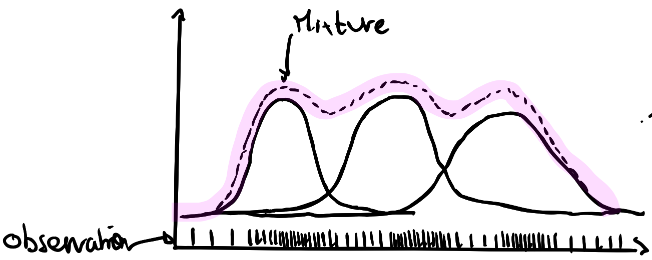
Assign relative importance to each sample w_i.

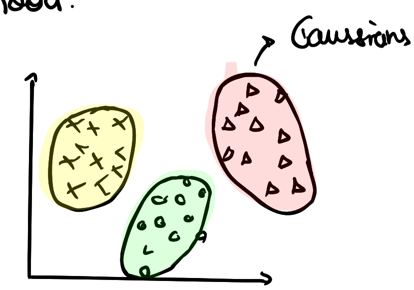
$$\min \sum_{j=1}^{k} \sum_{i \in C_j} w_i \|x_i-\mu_j\|^2 \quad \text{where} \quad \mu_j = \frac{\sum_{i \in C_j} w_i x_i}{\sum_{i \in C_j} w_i}$$

IV Gaussian Mixture model

Recall: $f(x) = \dfrac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$, $|\Sigma|$ determinant of Σ

Idea: Data may come from one of the gaussian distribution among the mixture. We need to set the number of clusters and identify best parameters by maximum likelihood.





$X \sim \mathcal{N}(\mu_Z, \Sigma_Z)$, $Z \sim \pi$ latent variables

The parameters are the following: $\theta = (\pi, \mu, \Sigma)$

- $\pi = (\pi_1 \dots \pi_k)$ mixing distribution (weights)
- $\mu = (\mu_1 \dots \mu_k)$
- $\Sigma = (\Sigma_1 \dots \Sigma_k)$

↳ $p_\theta(X|Z) = \prod_{i=1}^n f_{z_i}(x_i)$

Maximum log likelihood: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ l(\theta; Z)$, $l(\theta, Z) = \sum_{i=1}^n \log \pi_{z_i} + \sum_{i=1}^n \log f_{z_i}(x)$

↳ Find solutions $\hat{\pi}_j$, $\hat{\mu}_j$, $\hat{\Sigma}_j$

↳ Estimate latent variables : $p_\theta(Z|X) = \frac{p_\theta(X,Z)}{p_\theta(X)} \propto p_\theta(X,Z) = \prod_{i=1}^n \pi_{z_i} f_{z_i}(x_i)$

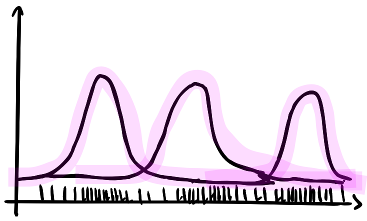
V. Expectation - maximization

Expected likelihood: $\sum_{j=1}^k \sum_{i=1}^n p_{ij} (\log \pi_j + \log f_j(x_i))$, $p_{ij} \propto \pi_i f_j(x_i)$ is the probability that observation i belongs to cluster j.

What is different with GMM ? We now consider for each observation its probability to belong to each cluster, and EM allows a clear MLE solution.

↳ $\hat{\pi}_j = \frac{n_j}{n}$, $\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^n p_{ij} x_i$, $\hat{\Sigma}_j = \frac{1}{n_j} \sum_{i=1}^n p_{ij} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T$

⇒ This is soft k-means. When $\sigma \to 0$, this is k-means.



Distance metric between 2 proba. distributions:

- KL divergence: $D(p \| q) = \sum_{j=1}^k p_k \log\left(\frac{p_k}{q_k}\right)$
- Entropy: $H(p) = -\sum_{j=1}^k p_k \log p_k$

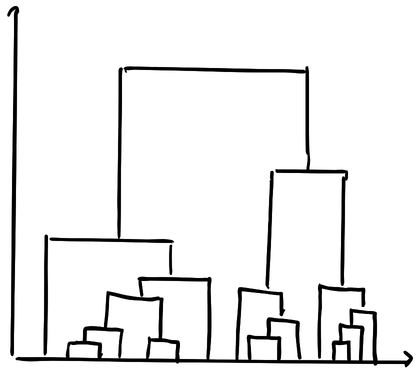
- Symmetric Gaussian mixture model.

↳ EM with the set of covariance matrices $\Sigma = (\sigma^2 I, \dots, \sigma^2 I)$

VI. Hierarchical clustering

Aim : Reflect multiscale nature of data.

2 approaches to build a dendogram



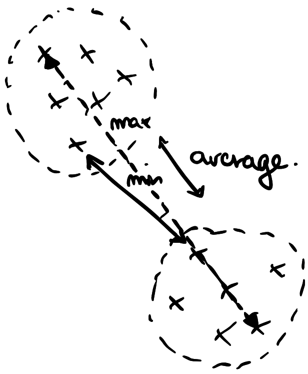
↓ Divisive (top down) → Best cut while at least two nodes

↑ Agglomerative → Best merge of 2 clusters

But how do we decide to split 2 far clusters or merge 2 close ones ?

- Minimum distance (single linkage):

$$d(a,b) = \min_{i\in a, j\in b} \| x_i - x_j \|$$



- Maximum distance (complete linkage):

$$d(a,b) = \max_{i\in a, j\in b} \| x_i - x_j \|$$

- Average (average linkage):

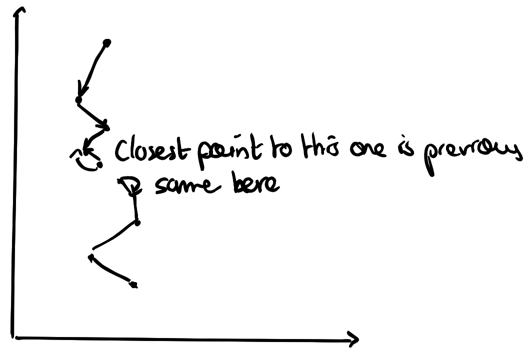
$$d(a,b) = \frac{1}{|a||b|} \sum_{i\in a, j\in b} \| x_i - x_j \|$$

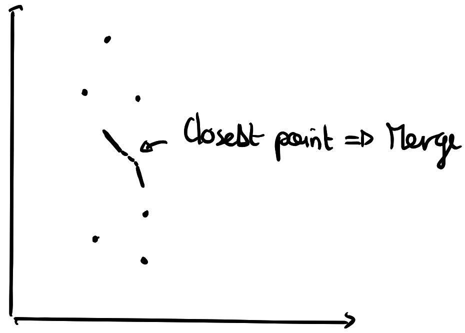
- Ward's method : Minimize sum of squared errors

$$S = \sum_{c\in C} \sum_{i\in n} \| x_i - g(c) \|^2 \quad \text{where } g(c) = \frac{1}{|C|} \Sigma_{i\in c} x_i$$

- Nearest neighbor chain
  1) Start from any cluster
  2) Build directed chain of nearest neighbors until 2 clusters are jointly nearest neighbors

3) Merge these 2 clusters and proceed with rest of chain until empty
4) Go to step 1 if there are at least 2 clusters left.



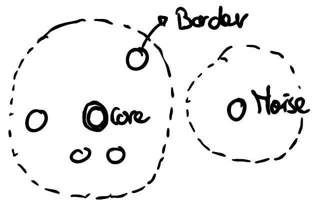


How do we measure performance for clustering?

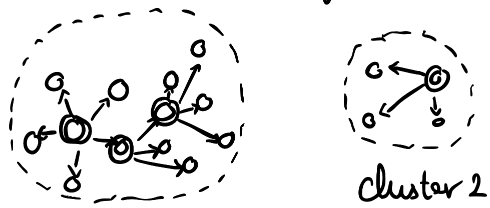
- square distance → Optimized for k-Means
- log-likelihood → Optimized for gaussian mixture
- mutual information (normalized)
- precision, recall, F1-Score

VIII. Density based spatial clustering of applications with noise

- A core is a point that has at least k points in its neighborhood.
- A border is a non-core point that has at least 1 core in its neighborhood.
- A noise is an outlier



- Pick a 1^{st} core point and perform core-search
  Assign all points recursively in neighborhood to the same cluster as X



Advantage
- Can fit any shape and size of cluster
- Can ignore outliers

Limitations: Sensitive to choice of neighborhood.

IX. Non-negative matrix factorisation model (NMF)

$$V_{F \times N} \simeq W_{F \times K} \times H_{K \times N}$$

Data $\simeq$ Expl. Var. $\times$ Regressors

We want to decompose V into positive matrices: $\begin{cases} W = [w_{fk}] \text{ s.t } w_{fk} \geq 0 \\ H = [h_{kn}] \text{ s.t } h_{kn} \geq 0 \end{cases}$

$\Leftrightarrow V_n \simeq \sum_{k=1}^{K} h_{kn} w_k$

Used for:
- topic modeling (LSA)
- clustering (K-Means)
- temporal segmentation (video)
- Source separation (ICA).

Limitations:
- Choice of K
- Non-unique solution

$V \approx WH$ usually obtained through $\min_{W,H \geq 0} D(V \mid WH)$ reconstitution error.
where $D(V \mid \hat{V}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d(v_{fn} \mid \hat{v}_{fn})$.

What divergence metrics can we use ?

- Euclidean distance : $d_{euc}(x, y) = (x-y)^2$

  $d_{euc}(\lambda x | \lambda y) = \lambda^2 d_{euc}(x|y)$ not scale invarent.

- Kullback - Leibler : $d_{KL}(x, y) = x \log \frac{x}{y} - x + y$

  $d_{KL}(\lambda x | \lambda y) = \lambda d_{KL}(x|y)$

- Itakuri - Saito (IS) : $d_{IS}(x, y) = \frac{x}{y} - \log \frac{x}{y} - 1$

  $d_{IS}(\lambda x | \lambda y) = d_{IS}(x|y)$ is scale invarient.

We solve optimal H and W values by multiplicative update (MU) and Majoration - minimization.

Limitations :
- monotonicity not guaranteed
- convergence not so good.

X. Independent Component Analysis (ICA)

$X_{nF} \simeq A_{F \times K} \cdot S_{nK}$

data expl. var. regressors

Used for source separation :

Mixed signal = Mixing matrix × Sources

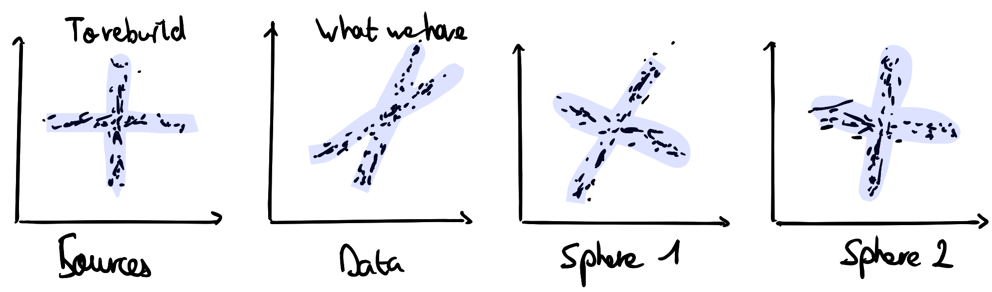
- 1) Sphering

Variance of entries of S must be normalized to 1.

$S_{sph} = A_{sph}^T X$ where $A_{sph} = E_{F \times K} D_K^{-\frac{1}{2}}$

$= Z$

$\hookrightarrow \mathbb{E}(ZZ^T) = I \Rightarrow S_{ISA} = U_{ISA} \cdot Z \Rightarrow$ Find U rotation



Once data is sphered:

- Construct numerical criteria $C(Y)$ measuring independence of entries
- Solve $\max_{\mu} C(\mu Z)$

↳ Signals that add up: CLT. We want to be as non-gaussian as possible

How to measure non-gaussianity?

- Kurtosis $\text{kurt}\{Y\} = \mathbb{E}(Y^4) - 3\mathbb{E}(Y^2)^2$
- Negentropy: $J(Y) = H\{Y_{\text{gaussian}}\} - H\{Y\}$

- Fast ICA algorithm

$$\max_{\mu} C(\mu) = |\text{kurt}\{\mu^T Z\}| \text{ s.t } \mu^T\mu = I$$

$$\Leftrightarrow \mathcal{L}(\mu, \lambda) = c(\mu) + \lambda(1 - \|\mu\|^2)$$

There must be at most 1 gaussian component inside., and we need to know how many sources there are.