Clustering

- major method in unsupervised learning:
  - find groups of similar instances $X_i \approx X_{i'}$ $\Rightarrow$ ~~put~~ them into clusters
    ($\approx$ "similar")
  - two benefits:
    - we can analyze data one cluster at a time
      $\Rightarrow$ if alg. complexity $O(N^p)$ $p \geq 1$ ~~$O(C \cdot N_1^p +$~~
      (N: entire data)
      for clusters $\sum_k O(N_k^p) < O(N^p)$
    - we can analyze global behavior by looking at cluster representatives instead of individual members

  $\Rightarrow$ simpler problem structure, faster, better interpretability for humans
- idea: TS $\{X_i\}_{i=1}^N$, assume that labels Y_i exist, but are unknown
  "latent" or "hidden" labels

  $\Rightarrow$ tasks: ① determine labels and clustering simultaneously

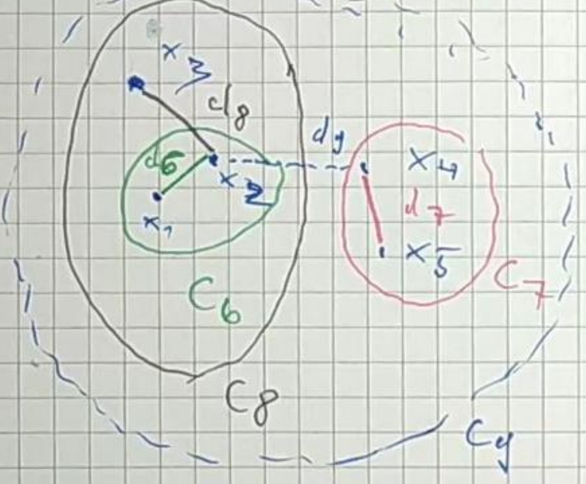
  ② find a good representative for each cluster, e.g.
  - mean: $\bar{X}_k = \arg\min_X \sum_{i \in C_k} (X_i - X)^2 = \frac{1}{N_k} \sum_{i \in C_k} X_i$
  - median: $\text{median}(C_k) = \arg\min_X \sum_{i \in C_k} \| X_i - X \|_1$ no simple solution in dimensions $D \geq 2$
  - medoid: $\text{medoid}(C_k) = \arg\min_{X \in \{X_i : i \in C_k\}} \sum_{i \in C_k} \| X_i - X \|_1$

    $\Rightarrow$ representative is chosen only among the cluster members

Hierarchical Clustering

⓪ init: each instance is a cluster of its own: $k = 1, \dots, N$ $\quad C_k = \{X_k\}$

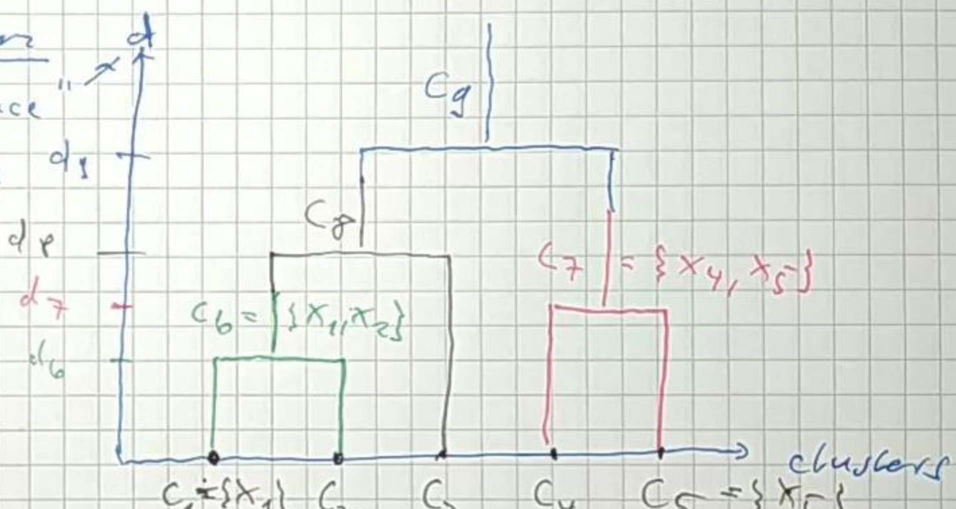
① repeat until all instance are in the same cluster: $(N-1)$ iterations
- find the two existing clusters closest to each other
- merge these two clusters into a new one $(k = N+1, \dots, 2N-1)$



Dendrogram

"merge distance" = dist, where a pair of clusters got merged

example: single linkage clustering



By stopping the algorithm early (when a minimum number of clusters is reached, or when the next merge edge exceeds a given threshold) ⇒ control the granularity of the clustering (hyperparameter)

Define distance between clusters: given (hyperparameter): distance between instances $d(X_i, X_{i'})$

- single linkage: $d(C_k, C_{k'}) = \min\limits_{i \in C_k,\, i' \in C_{k'}} d(X_i, X_{i'})$
- complete linkage: $d(C_k, C_{k'}) = \max\limits_{i \in C_k,\, i' \in C_{k'}} d(X_i, X_{i'})$

- average linkage: $d(C_k, C_{k'}) = \frac{1}{N_k N_{k'}} \sum_{i \in C_k} \sum_{i' \in C_{k'}} d(x_i, x_{i'})$
- representative linkage $d(C_k, C_{k'}) = d(\text{repr.}(C_k), \text{repr.}(C_{k'}))$

properties:
- single linkage is not robust against outliers => "chaining"

=> complete linkage more robust against this problem

chain incorrectly connects the clusters

- complete linkage ~~fails~~ may fail on the "closeness property" ≙ within cluster distance are less than between cluster distances => single linkage better
- average and representative linkage are in between
- if data clusters well, all criteria give similar results

<u>k-means clustering</u> K ≙ predefined number of clusters (here: C)

- consider total distance of TS: $d_{total} = \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d(x_i, x_{i'})$
- any given clustering separates edges into "within" and "between" cluster edges

$d_{total} = d_{in} + d_{betw.}$ $\quad d_{in} = \frac{1}{2} \sum_{k=1}^{C} \sum_{i \in C_k}^{N_k} \sum_{i' \in C_k}^{N_k} d(x_i, x_{i'})$

$$d_{betw.} = \frac{1}{2} \sum_{k=1}^{C} \sum_{i \in C_k} \sum_{i' \notin C_k} d(x_i, x_{i'})$$

- a good clustering minimizes d_{in} or maximize $d_{betw.} = d_{total} - d_{in}$ (both objectives are equivalent)

- exact optimization requires exhaustive search over all possible clustering
  $\Rightarrow$ too expensive
- k-means alg. simple heuristic to minimize d_{in} for special choice

$$d(X_i, X_{i'}) = \| X_i - X_{i'} \|_2^2$$

$\Rightarrow$ d_{in} simplifies: $$d_{in} = \sum_{k=1}^{C} N_k \sum_{i \in C_k} \| X_i - \bar{X}_k \|_2^2$$

$$\bar{X}_k = \frac{1}{N_k} \sum_{i \in C_k} X_i \quad \text{mean of cluster } k$$

- introduce hidden labels: Y_i : $Y_i = k$ means $X_i \in C_k$

  when means were known: $$Y_i = \arg\min_k \| X_i - \bar{X}_k \|_2^2$$

  $\Rightarrow$ assign X_i to the nearest cluster representative

$\Rightarrow$ optimize alternatingly:

(0) define initial guess for cluster centers $\bar{X}_k^{(0)}$

(1) repeat until labels Y_i do no longer change (always happens after finite steps)

  (a) update labels $Y_i^{(t)} = \arg\min_k \| X_i - \bar{X}_k^{(t-1)} \|_2^2$

  (b) update means $\bar{X}_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i \in Y_i^{(t)} = k} X_i \qquad N_k^{(t)} = \# \{ Y_i^{(t)} = k \}$

- converges to a local optimum of d_{in} (not the global one in general)

$\Rightarrow$ quality of result critically depends on initial guess

- k-means++ : improved initial guess

① choose $\bar{X}_1^{(0)} \sim \{X_i\}_{i=1}^{N}$ uniformly at random

② for $k = 2, \dots, C$: (other cluster centers)

- define distances to closest existing center:

$$\forall i: \quad d_i = \min_{k' \in 1,\dots,k-1} d(X_i, \bar{X}_{k'}^{(0)})$$

- define probabilities by normalization: $\cancel{\bar{d} = \frac{d_i}{\sum d_i}}$ $\quad p_i = \cancel{\frac{d_i}{\bar d}} \frac{d_i}{\sum_i d_i}$

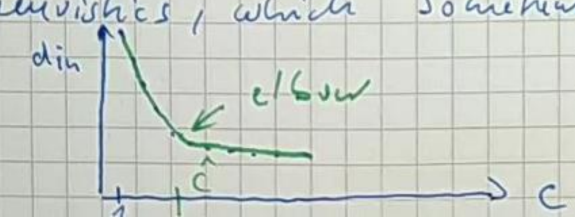
- choose $\bar{X}_k^{(0)} \sim \text{discrete}(\{p_i\})$

(already chosen points have $d_i = 0$, $p_i = 0$ ⇒ will not be picked again, p_i large ≙ d_i relatively large ≙ point i far away from existing centers ⇒ $\bar{X}_k^{(0)}$ are well spread out over the training distribution)

- variant: k-medoids : instead of mean, represent each cluster by its medoid ⇒ works for any distance $d(X_i, X_{i'})$ but more expensive than k-means

- unsolved: how to choose optimal number C of clusters ≙ hyperparameter

- several heuristics, which sometimes work, for example elbow method



EM algorithm (expectation - maximization)

- k-means recovers <u>hard cluster assignments</u> Y_i $\hat{=}$ analogous to discriminative model in classification
- EM alg corresponds to a generative model:
  - soft assignments: for each X_i, define posterior $p(Y_i = k \mid X_i)$
  - model posterior by learning RHS of Bayes formula $p(Y_i \mid X_i) = \dfrac{p(X_i \mid Y_i)\, p(Y_i)}{p(X_i)}$

$\Rightarrow$ <u>mixture model</u>: $p(Y_i = k)$ probab. that a point belongs to cluster k

$p(X_i \mid Y_i = k)$ shape of cluster k

as in QDA, but the labels Y_i are now hidden in TS

most common: $p(X_i \mid Y_i = k) = \mathcal{N}(\mu_k, S_k)$ $\Rightarrow$ Gaussian mixture model GMM

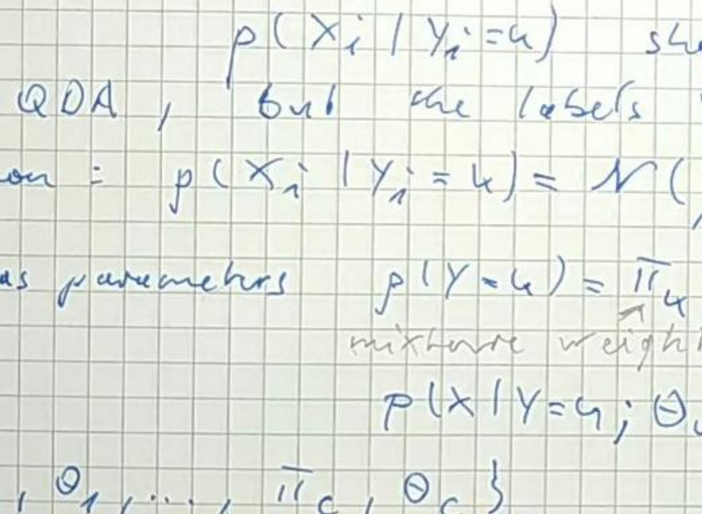
$\Rightarrow$ solution has parameters $p(Y = k) = \underset{\text{mixture weights}}{\pi_k} \approx \dfrac{N_k}{N}$ $\quad \sum_k \pi_k = 1$

$p(X \mid Y = k; \Theta_k) = \mathcal{N}(\mu_k, S_k)$ $\quad \Theta_k = \{\mu_k, S_k\}$

$\Theta = \{\pi_1, \Theta_1, \dots, \pi_C, \Theta_C\}$

example: $p(x_i)$

$C = 3$



$\leftarrow$ cluster 1 $= \pi_1 \cdot \mathcal{N}(\mu_1, \sigma_1)$ $\quad$ π_1 large ($\pi_1 = 0.45$), σ_1 small

cluster 2 $= \pi_2\, \mathcal{N}(\mu_2, \sigma_2)$ $\quad$ π_2 large ($\pi_2 = 0.45$), σ_2 large

cluster 3 $= \pi_3\, \mathcal{N}(\mu_3, \sigma_3)$ $\quad$ π_3 small, $\pi_3 = 0.1$, σ_3 small

$\leftarrow$ $p(X_i) = \sum_k \pi_k\, \mathcal{N}(\mu_k, S_k)$

optimize via maximum likelihood principle :

$$\hat{\theta} = \arg\max_{\theta} \prod_i p(x_i;\theta) \iff \hat{\theta} = \arg\max_{\theta} \sum_i \log p(x_i;\theta)$$

$$= \arg\max_{\theta} \sum_i \log \sum_k \pi_k \, N(\underbrace{\mu_k, S_k}_{\theta_k})$$

$$\frac{\partial \text{Loss}}{\partial \theta_k} = \sum_i \frac{\partial}{\partial \theta_k} \log p(x_i;\theta) = \sum_i \frac{1}{p(x_i;\theta)} \frac{\partial}{\partial \theta_k} p(x_i;\theta)$$

$$= \sum_i \frac{1}{p(x_i;\theta)} \frac{\partial}{\partial \theta_k} \cancel{\pi_k \log} \, \pi_k \, p(x_i;\theta_k) = \sum_i \frac{\pi_k}{p(x_i;\theta)} \frac{\partial}{\partial \theta_k} p(x_i;\theta_k)$$

$$= \sum_i \frac{\pi_k \, p(x_i;\theta_k)}{p(x_i;\theta)} \cdot \frac{1}{p(x_i;\theta_k)} \frac{\partial}{\partial \theta_k} p(x_i;\theta_k)$$

$$= \sum_i \frac{\pi_k \, p(x_i;\theta_k)}{p(x_i;\theta)} \frac{\partial}{\partial \theta_k} \log p(x_i;\theta_k)$$

[Bayes: $\pi_k = p(Y_i = k)$ $\quad \pi_k \, p(x_i;\theta_k) = p(x_i, Y_i = k; \theta_k) = p(x_i, Y_i = k; \theta)$

$= p(Y_i = k \mid x_i; \theta_k) \, p(x_i \mid \theta)$]

$$\boxed{\frac{\partial \text{Loss}}{\partial \theta_k} = \sum_i p(Y_i = k \mid x_i;\theta) \cdot \frac{\partial}{\partial \theta_k} \log p(x_i \mid \theta_k) \overset{!}{=} 0}$$

we cannot solve analytically for θ_k, because $\theta_k \in \Theta$, the two dependencies cannot be rewritten in closed form $\Rightarrow$ <u>alternating optimization</u>

EM alg.: alternating optimization of the loss:

(0) initialization: choose number of clusters C (hyperparameter)

initial cluster parameters $\Theta_k^{(0)} \; (= \mu_k^{(0)}, S_k^{(0)})$, $\pi_k^{(0)}$

(1) for $t = 1, \dots, T$ or until convergence

(a) define "responsibilities": auxilliary variables $\gamma_{ik} \hat{=}$ how well is instance i explained by cluster k in the current guess

$$\gamma_{ik} = p(Y_i = k \mid X_i, \Theta_k^{(t-1)}) = \frac{\pi_k^{(t-1)} \cdot \mathcal{N}(X_i \mid \mu_k^{(t-1)}, S_k^{(t-1)})}{\sum_{k'=1}^{C} \pi_{k'}^{(t-1)} \cdot \mathcal{N}(X_i \mid \mu_{k'}^{(t-1)}, S_{k'}^{(t-1)})}$$

(b) E-step: optimize π_k as the expectation of γ_{ik}

$$\pi_k^{(t)} = \frac{1}{N} \sum_i \gamma_{ik}$$

(c) M-step: maximize data likelihood with $\pi_k^{(t)}$ fixed

$\Rightarrow$ choose $\Theta_k = \{\mu_k, S_k\}$ to fit the data as well as possible

(equivalent to QDA, but with guessed soft class labels γ_{ik})

$\Rightarrow$ weighted mean and covariance:

$$\mu_k^{(t)} = \frac{\frac{1}{N}\sum_i \gamma_{ik} X_i}{\underbrace{\frac{1}{N}\sum_i \gamma_{ik}}_{\pi_k^{(t)}}} \qquad S_k^{(t)} = \frac{\frac{1}{N}\sum_i \gamma_{ik} (X_i - \mu_k^{(t)})^T (X_i - \mu_k^{(t)})}{\underbrace{\frac{1}{N} \cancel{\mu_k^{(t)}} \sum_i \gamma_{ik}}_{\pi_k^{(t)}}}$$