so far: $x \to y$ (prediction)

New task $x \to$ summary (x)

Why??

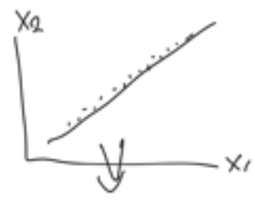
summarization: news / synthesis of info & organization
visualization
compression (image/video)
latent causes [hypotheses]

What

unsupervised learning / representation learning

clustering

$\Leftrightarrow$ classification



manifold learning / embedding

$\Leftrightarrow$ regression

discrete vs. continuous

supervised vs. unsupervised

~~Bayes~~ non-prob vs. probabilistic

How evaluate.

reconstruction error : $\mathcal{L}\Big(x, \underbrace{\text{decode}(\underbrace{\text{encode}(x)}_{\text{summary}(x)})}_{\text{recovery of } x}\Big)$

but: we can also create very method-specific criteria
(ex. clustering: stability, relative cluster sizes, etc.)

How to do it?
(clustering)

how many?

what shape?

Set-up: Data: $x_1 \ldots x_N$

Similiarity/Distance $d(x,x')$ e.g. $d(x,x') = \|x - x'\|$

goal is to group similar points together

for now, suppose K groups, z_{nk} indicate group assignment

z_n to be a binary vec. of length K

$z_{nk} = 1$ for k is the group of datum x_n $\quad [0\,1\,0\,0\,0]$

Specific Approach #1: K-means

goal: define μ_k as the cluster "prototypes" $\{\mu_1 \dots \mu_K\}$

assign z_{nk}, find μ_k s.t.

$$\min_{\{z\}\{\mu\}} \sum_n \underbrace{\sum_k z_{nk}}_{\text{picking out which cluster } x_n \text{ was assigned to}} \underbrace{\|x_n - \mu_k\|_2^2}_{\text{Loss}}$$

x_n = encoding of x_n

μ_k = decoding of x_n

unfortunately, this is NP-hard!

can we find a local optima?

- iterative block update/coordinate ascent
- [K-means/Lloyd's Alg]
  - → start by randomly assigning z_n's
  - → loop:

    update $\{\mu\}$ given $\{z\}$:

    for each k, $\mu_k = \frac{1}{\underbrace{N_k}_{\sum_n z_{nk}}} \sum_n z_{nk} x_n$ } set μ_k to the mean of x_n's assigned to cluster k

    update $\{z\}$ given $\{\mu\}$:

    for each n, assign x_n's cluster to $\arg\min_k \|x_n - \mu_k\|$

Computation:

- nice, parallelized/distributed

Theory:

- nonconvex → need to do many random restarts / K-means++
- but, procedure always is improving the obj.
  - → by assigning z_n to be $\|x_n - \mu_k\|$, loss has to go down (or stay constant)

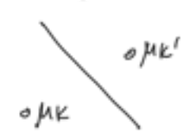
→ $\mathcal{L}(\mu_k) = \sum_n z_{nk} (x_n - \mu_k)^T (x_n - \mu_k)$

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = -2 \sum_n z_{nk} (x_n - \mu_k) = 0 \Rightarrow \mu_k = \frac{\sum_n z_{nk} x_n}{\sum_n z_{nk}}$$

↳ implementation note: you can check this!

Concepts:

- What kind of boundary?



linear boundaries → may need to $x \to \phi(x)$, but K-means will only do linear boundaries

- What about different $d(x, x')$?

↳ z assignment is still easy $\min_k d(x_n, \mu_k)$

↳ μ_k may get hard → K-medoids which forces μ_k to be a $\{x_n\}$

- How many clusters?
  problem-specific

<u>Specific Approach #2</u> : deterministic, nonparametric (tree), non-linear boundaries

Hierarchical Agglomerative Clustering

"dendrogram"

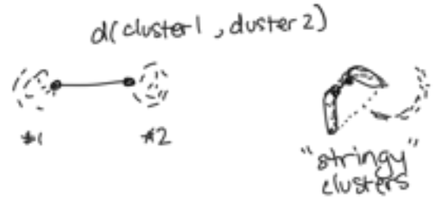
Alg: $\{x_n\}$, $d(x, x')$

1) everyone starts in their own cluster
2) while #clusters > 1, merge closest clusters

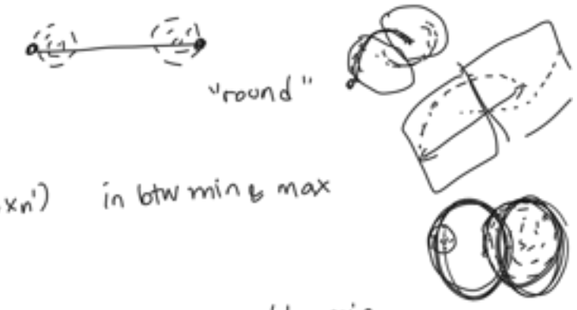
Obvious question: what is the "closest" cluster? we have $d(x, x')$

"linkage"

- $\min_{\substack{n \in \text{cluster 1} \\ n' \in \text{cluster 2}}} d(x_n, x_{n'})$



- $\max_{\substack{n \text{ in } C1 \\ n' \text{ in } C2}} d(x_n, x_{n'})$



- average $\frac{1}{N_1 N'} \sum_{n \text{ in } C_1} \sum_{n' \text{ in } C_2} d(x_n, x_{n'})$ in btw min & max

- centroid/ward $d(\mu_{\#1}, \mu_{\#2})$ in btw min & max

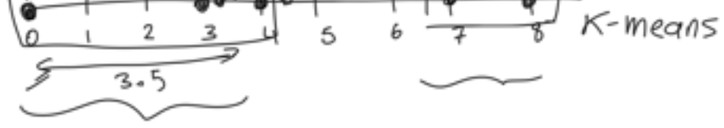
  mean cluster 1, mean cluster 2

<u>Notes</u>:

- very sensitive to choices of $d(x, x')$ & linkage
- high dims can get tricky w/ Euclidean $d(x, x')$ because all data will ≈ equally far apart. [non-specific to HAC]
- obj is not as clear (2016, 2017 papers..)
  - ↳ cluster sizes
  - ↳ $\sum d(x, x')$ crossing a boundary

K-means, HAC : Concept Exercise

HAC: @ 2 clusters
K-means: 2 clusters



K-means

one more example:

large # M of random [binary w.p. $\frac{1}{2}$].

x_1: 0000 0101101

⋮

x_n: 1111 0000

⋮

K-means: 0 0 0 0 $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ ...

1 1 1 1 $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ ...

HAC: for large M, essentially random $d(x, x')$

Clustering