Supervised: $Y \in \mathbb{R}$ or $Y \in \mathcal{C}$ and $X \in \mathbb{R}^p$

If (X, Y) as random then they have some joint dist. $p(x, y)$.

Supervised Learning is basically density estimation where we want to estimate $p(y|x)$.

prob. of seeing a y given a x

often we just want

$$\hat{f} = \arg\min_f E_{Y|X} L(f)$$

some mean we estimating

Bayes Theorem says that

supervised interest in this

$$p(x, y) = p(y|x)p(x)$$

not so much this

Unsupervised Learning Just have $X \in \mathbb{R}^p$ (no Y)

We want to estimate $p(x)$

→ summarize

Quirks of Unsupervised

→ no direct measure of success

→ so many methods & opinions

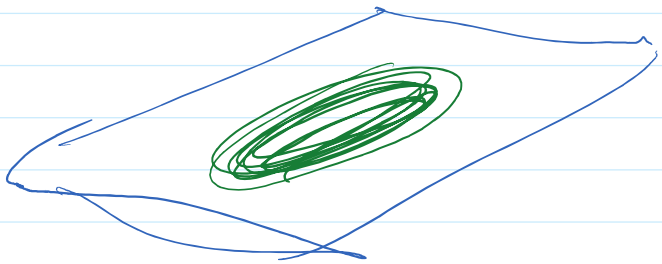
How to summarize $p(x)$

→ If P is low (low-dim'l problem)

directly estimate $p(x)$ / visualize

→ If P is high ⇒ ??? summary stats?

or PCA: find a low-dim'l subspace that contains high density parts of $p(x)$

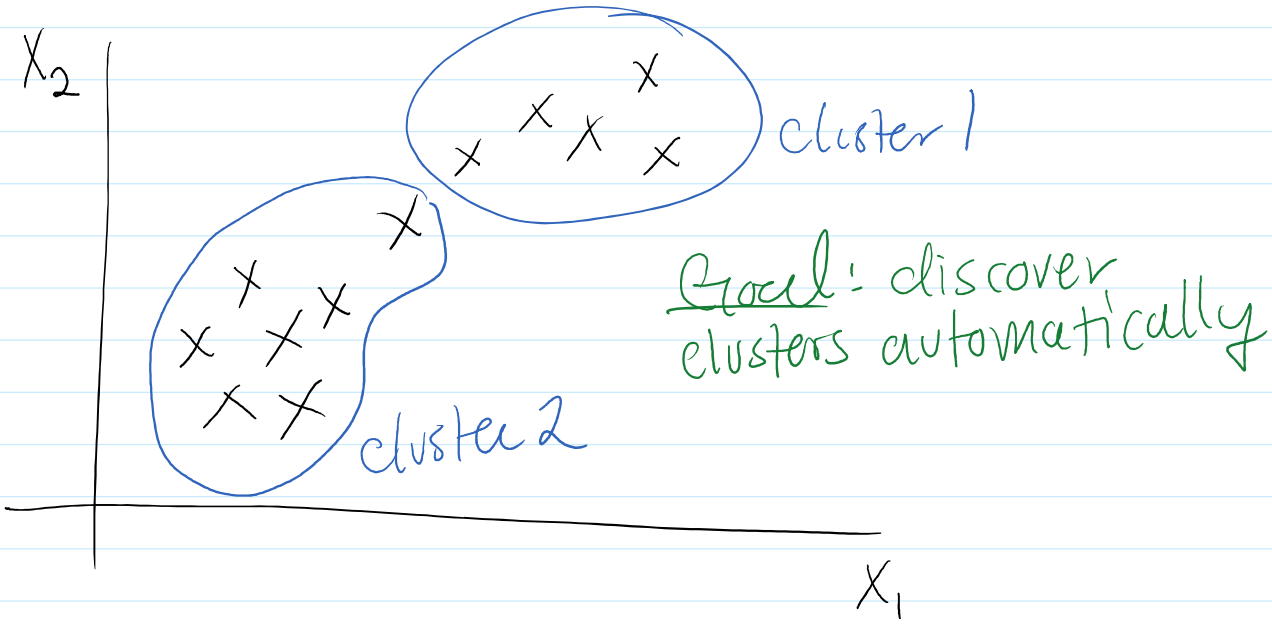


Clustering:

Basically tries to find convex sets of

high density parts of $p(x)$.

Classic: want to create "clusters" w/in our data so that pts w/in a cluster are more similar to each other than points in different clusters.



Goal: discover clusters automatically

Sim/Dissim

To do clustering, we need some measure of either similarity or dissimilarity

Ultimately (almost) all clustering algs just need a measure of sim/disim btwn all pts.

If I have N objects then we can form a matrix

$$D_{N \times N}$$

so that $D_{ii'}$ = dissim btwn i^{th} and i'^{th} objects

[If I have a measure of sim. btwn objects, I can create D using a decreasing transformation.]

Many/most cluster algos only need D, don't even need some N x P matrix X.

Properties req. for D

(1) $D = \begin{bmatrix} 0 & & \\ & \ddots & \\ & & 0 \end{bmatrix}$ ($D_{ii} = 0$) zeros down main diag

(2) $D_{ii'} \geq 0$ (non-neg.)

(3) $D = D^T$ (symmetric)

If we have features $X_{N \times p}$ we can form attribute-based dissims

i.e. use diffs btwn feats to calc D.

dissim for

$$D_{ii'} = d(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

dissim meas for feat j

$$D_{ii'} = d(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_i, x_{i'})$$

rows i and i' of X

for feat j

Caveat: careful w/ scale

Ex. ① Numeric Feature j   Euclidean (Squared) dist.

$$d_j(x_i, x_{i'}) = (x_{ij} - x_{i'j})^2$$

② Ordinal Feature j

transform level i to $\frac{i - 1/2}{M}$ for $i = 1, \ldots, M$

and treat as numeric (eg. use Euclidean)

③ Categorical Feature j

$$d_j(x_i, x_{i'}) = \mathbb{1}(x_{ij} = x_{i'j})$$

$\bar{0}$ $\begin{cases} 1 & \text{if same} \\ & \text{else} \end{cases}$

Combinatorical Clustering Algos

Assume data comes from one of K clusters

$$G_1, \ldots, G_K$$

Want to do: assign each x_i to some G_k

How: I do this assignment to minimize some measurement ("Loss") of not being clustered.

Classic measurement

$$W = \text{total within cluster dissim} = \sum_{k=1}^{K} \sum_{i,i' \in G_k} D_{ii'}$$

= total dissim measure across elements in same cluster.

Should be small if clustered well
" large if not clustered well

Note that if

$$T = \sum_{i,i'} D_{ii'} = \text{total dissim}$$

and $\text{total within cluster} = \sum_{k=1}^{K} \sum \sum D_{ii'}$

and B = total btwn cluster dissim $= \sum_{k=1}^{K} \sum_{i \in G_k} \sum_{i' \notin G_k} D_{ii'}$

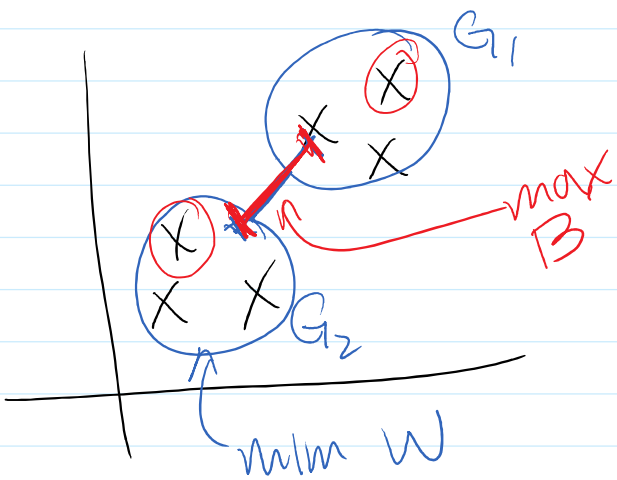
= total dissim btwn pairs in diff clusters

Note: $T = W + B$

To find good Gs we can

(1) minimize W

or (2) Since $W = T - B$ we can maximize B



How to find Gs?

Ideally, consider all possible Gs

Ideally, consider all possible C's

Not computationally tractible

Ex. $N = 19$ and $K = 4$

then # of possible assignments is $\sim 10^{10}$

Soln: Greed.

K-Means:

Today: all features are numeric, and

$$D_{ii'} = \|x_i - x_{i'}\|^2$$

↑ squared euclidean dist.

In this case we can show that

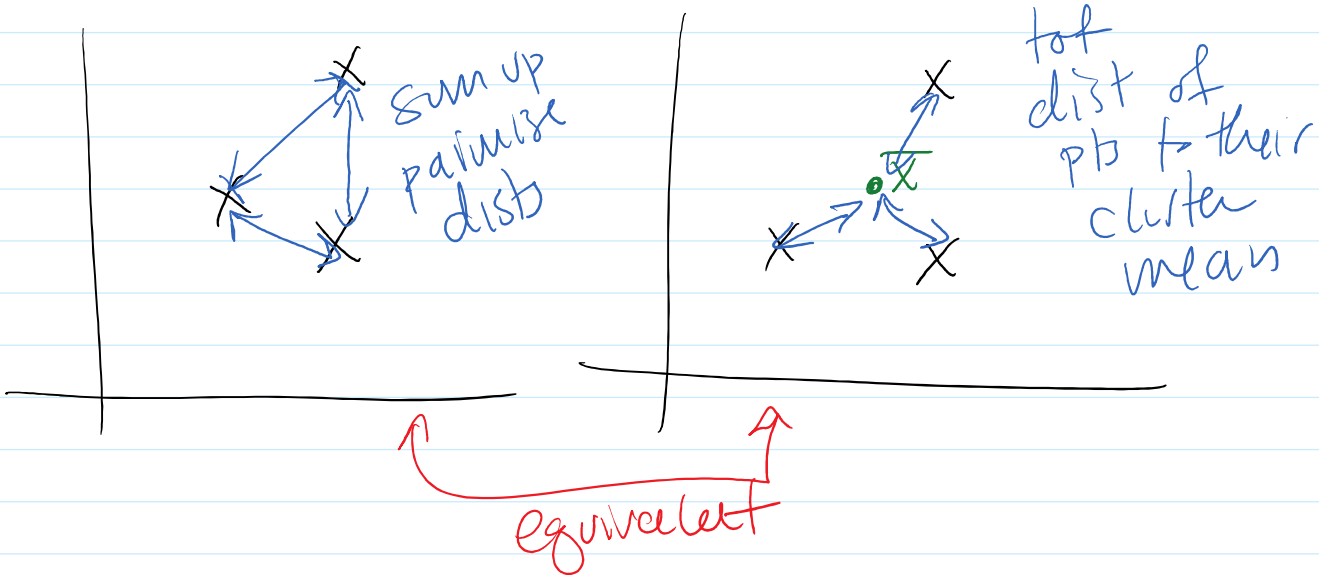
$$W = \sum_{k=1}^{K} N_k \sum_{i \in G_k} \|x_i - \bar{x}_k\|^2$$

N_k = # in k^{th} cluster

$\bar{x}_k$ = mean of k^{th} cluster

[Recall: $W = \sum_{k=1}^{K} \sum_{i,i' \in G_k} D_{ii'} = \sum_{k=1}^{K} \sum_{i,i' \in G_k} \|x_i - x_{i'}\|^2$]

tot. pairwise dist. of pts in a cluster



Of Course Goal: choose $G_1, \ldots, G_K$ so that we minimize W.

Classic Way of doing: Lloyd's Algorithm

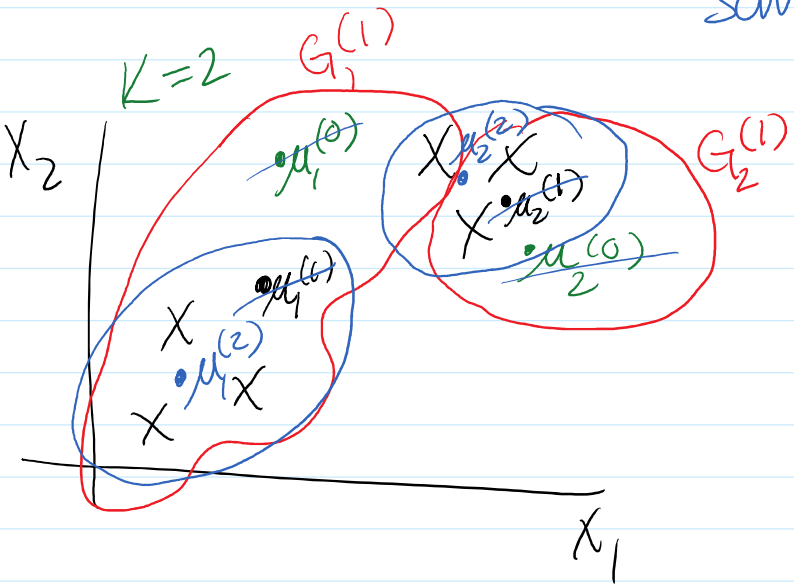
(0) Initialize Step:

Make initial (random) guesses of grp means: $\mu_1^{(0)}, \ldots, \mu_K^{(0)}$

Loop over some iterations $t = 1, 2, 3, 4, \ldots$ where at the t^{th} iteration

(1) Assignment Step assign each x_i to the Grp w/ the closest mean $\mu_k^{(t-1)}$

(2) Update Step: re-compute μs as sample means of new assignments



Do this until means stop moving.