Clustering Evaluation

(applications of clustering: market segmentation, grouping text documents, anomaly detection, social network analysis, search results grouping, ...)

- silhouette score, ($\in [\overset{\text{bad}}{-1}, 1]$) : $S_i \approx 1 - \dfrac{\text{typical intercluster size}}{\text{typical intra cluster size}}$
  (average over all i points)

- evaluate likelihood on held out data (if clustering method is probabilistic)
  - can help you decide between probabilistic models (e.g. can help you decide which K to use in GMM)

cluster, then assign class labels

- Use class labels in ~~predicting~~ training and testing a supervised classifier. Assuming your classifier is good, if the performance is good, you can expect a decent clustering

- clustering performance really depends on how helpful it is in domain application (e.g. market segmentation, anomaly detection, etc...), thus, there's no <u>one</u> method for evaluation

- Choosing K: elbow plots, domain knowledge, choosing K to maximize likelihood on a holdout set (if probabilistic)

K - means

- performs a hard clustering by minimizing sum of squared distances to assigned cluster center

$$\{a_{ij}\}^{*\, i=m;\, j=K}_{i,j=1} = \underset{\{a\}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{K} \sum_{i=1}^{m} a_{ij} \| x^{(i)} - \mu_j \|^2 \right\}$$

where $a_{ij} = \begin{cases} 1 & \text{if } i \text{ is assigned to cluster } j \\ 0 & \text{otherwise} \end{cases}$

$$\mu_j = \frac{\sum_{i=1}^{m} a_{ij} x^{(i)}}{\sum_{i=1}^{m} a_{ij}}$$

Algo to find $\{a_{ij}\}^*$:

① randomly initialize all μ_j to data points

repeat until conv {

① assign data points to closest centroid

② update centroids

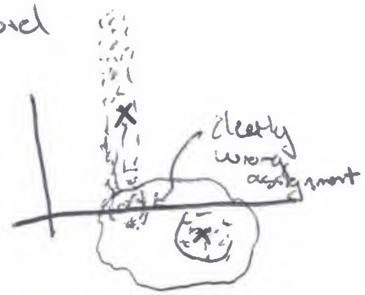
}

Pros

- easy to understand
- fast to train

Cons

- clusters need to be roughly spherical since objective minimize <u>distances</u> to centroid



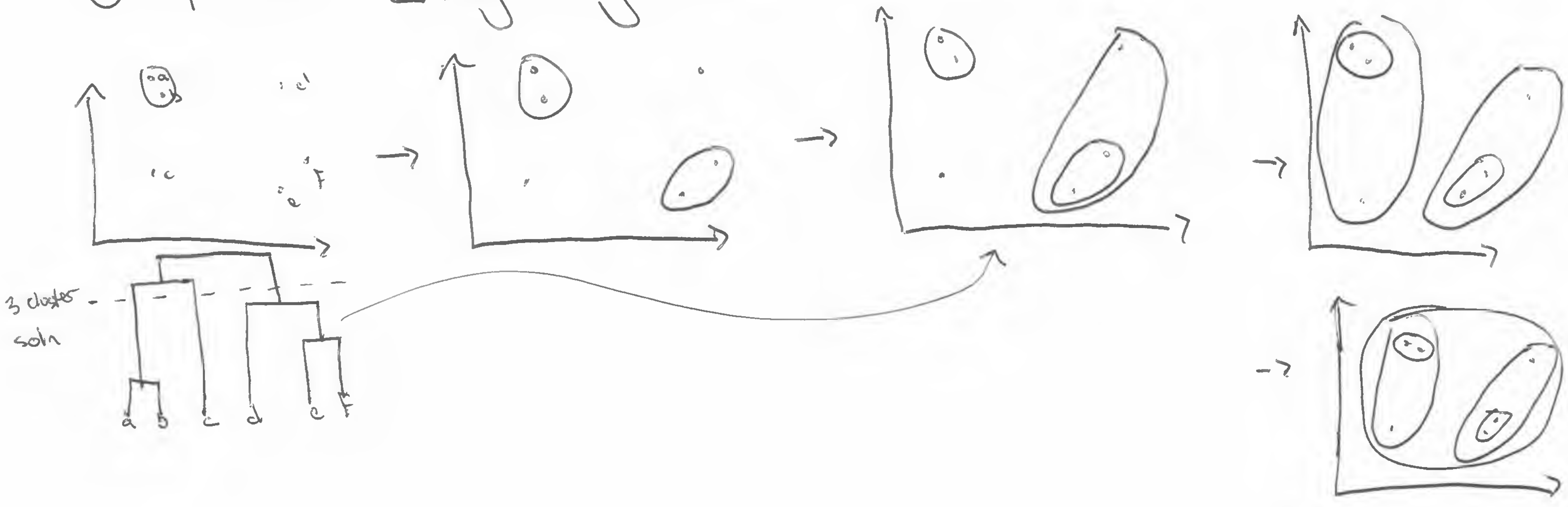
- provides only a "hard" clustering (which may or may not be best for problem at hand)
- subject to local minima
- soln. depends on scaling of data
- need to specify k
- only works for $X \in \mathbb{R}^n$ (not categorical)

Hierarchical Clustering

- choose distance/dissimilarity measure (euclidean, correlation) ← good for clustering time series as well as when you don't care about absolute mag.
- choose "linkage" (defines distances between clusters)

Algo

1. start w/ each point in its own cluster
2. join closest 2 clusters
3. repeat until 1 single big cluster



Pros

- get to understand hierarchical structure in data
- clusters need not be spherical
- Can use on many data types (just need to supply pairwise distance matrix)

Cons

- scaling matters
- need to pick linkage + distance metric
- need to choose K
- computationally expensive
- not super easy to interpret

DB-Scan - regions w/ high contiguous density get clustered together (if there are at least n_min points within a radius of ε of current points these points are part of the current cluster).

Pros

- k not needed
- can use any distance metric you want
- can find very nonlinear, non-spherical shaped clusters
- does not need to include every point in a cluster (has a notion of outliers)

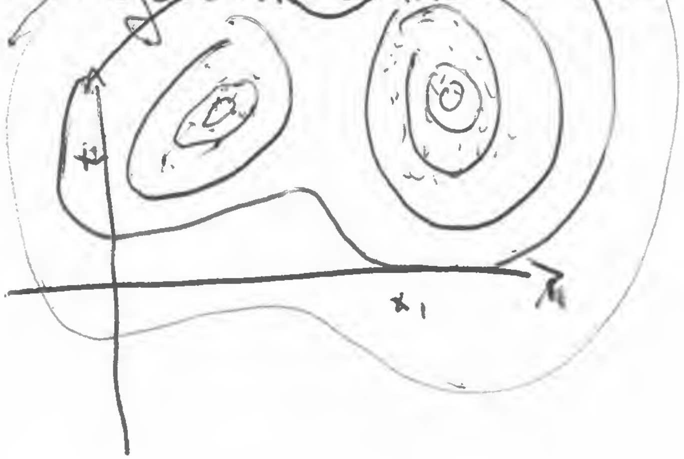
Cons

- doesn't work well w/ datasets with large variations in density
- could result in different clusterings depending on initial seed

Gaussian Mixture Model

- density estimation
- soft clustering
- generative model with latent RVs.

- unsupervised model



generative
Model: ① toss a k-faced die to determine class label

② sample from that class' multivariate normal

$$z^{(i)} \sim \mathrm{Cat}(\phi) \qquad \phi \in \mathbb{R}^k \text{ and } \sum \phi_j = 1$$

$$x^{(i)} \mid z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j) \qquad \mu_j \in \mathbb{R}^n, \quad \Sigma_j \in \mathbb{R}^{n \times n} \text{ and is symmetric PD}$$

MLE estimation of parameters

$$P(z^{(i)}; \phi) = \prod_{j=1}^{k} \phi_j^{\mathbb{1}\{z^{(i)}=j\}}$$

$$P(x^{(i)} \mid z^{(i)}=j) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\{-\tfrac{1}{2}(x^{(i)}-\mu_j)^T \Sigma^{-1} (x^{(i)}-\mu_j)\}$$

joint: $P(x^{(i)}, z^{(i)}) = P(x^{(i)} \mid z^{(i)}) P(z^{(i)})$

$$\ell(\Theta) = \log \prod_{i=1}^{m} P(x^{(i)}; \Theta) = \sum_{i=1}^{m} \log P(x^{(i)}; \Theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} P(x^{(i)}, z^{(i)}; \Theta)$$

- This is intractable to maximize analytically
- Thus we use the EM algo since all dists. are of the exponential family
- In the M step, we must use the method of Lagrange Multipliers since $\sum \phi_j = 1$
- Can constrain Σ_js to be spherical, diagonal etc. if not enough data to estimate full Σ_js

Pros

- provides a "soft" clustering w/ $P(z^{(i)} \mid x^{(i)}; \hat{\Theta})$ (posteriors of $z^{(i)}$)
- can also make into hard clusters by $z^{(i)*} = \arg\max_{z^{(i)}}(\cdot)$
- can accomodate non-spherical geometries

Cons

- local minima
- still need to specify K (could possibly do this on a holdout set by maximizing likelihood).
- clusters need to be ellipsoidal