Unsupervised Learning

↳ How to use ML when no labels are provided?

↳ Labels can be expensive to acquire, UL allows for insights despite no labels

Clustering

- k-means
- hierarchical

Gaussian Mix. Models

- Density est.
- Comparison w/ k-means

↳ the same under certain conditions

What is UL used for?

(Bayesianism?)

- density estimation (est of PDF)
- Latent variables
- Dimension reduction
  - ↳ understand data
  - ↳ see if models work well

Why use clustering?

- can be used on any dataset.
- Also allows for speed of algorithms if you need to search for similar items
  - recommendation systems
- Density estimation
  - generate new things based on old things (Hidden Markov Models)

k means clustering

No labels to data

↳ there might still be patterns to find

→ Challenges:

- High dimensions can't be visualised
- Not all interesting info is shown in clustering
  ↳ How do we evaluate this?

k-means

Based on 2 facts:

- clusters have a center of mass

$$\text{com.} = \frac{1}{c} \sum \vec{c}_i$$

- new points get assigned to closest cluster

More on k-means

- Start w/ random centroids
- map points to closest pt.
- recenter to CoM
- repeat until converge

} n centroids = k

Soft k-means

"fuzzy" membership

- membership is not binary but a continuous value

$$r_k^{(n)} = \frac{e^{-\beta\, d(m_k, x^n)}}{\sum_{j=1}^{K} e^{-\beta\, d(m_j, x^n)}}$$

dist between centroid & point

similar to gaussian

β ← "sensitivity term"

Objective Function

$$J = \sum_n \sum_k \| \mu_k - x^{(n)} \|^2$$ } coordinate descent

↳ will always decrease but not convex

↳ Multiple local minima

When can k-means fail

- donut problem
- two clusters can also be split awkwardly depending on initialization
- less dense functions that are next to very dense ones

→ **Disadv. of k means:**

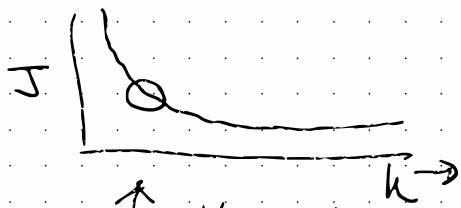
- k needs to be determined
- local min.
- can only work for spherical clusters

} experimentation needs to be done.

→ How to choose k

→ Plot scree plot

→ Choose k that reduces the cost the most.



↑ Elbow has most info per cluster

Cost function Alternatives

current — inter-cluster dist ○←→○
intra cluster dist ○

↳ Decreases every round
Low ○, high ○-○ is desirable

Disadv.: problems w/ large dataset
- cost is sensitive to scale of data
- cost also sensitive to k
  ↳ high k overfitting?

Not really - unsupervised ML - remember?

Davies-Bouldin Index

↳ "internal validation"

$$DBI = \frac{1}{k} \sum_{k=1}^{k} \max_{j \neq k} \left[\frac{\sigma_j + \sigma_k}{d(c_k, c_j)} \right]$$

- $\sigma_j + \sigma_k$: dist from each pt. to cluster centre
- $j \neq k$: different clusters
- $d(c_k, c_j)$: dist between clusters (centre)

Hierarchical Clustering

"Agglomerative Clustering"

↳ grow clusters from bottom-up

↳ closest together get clustered first

⇒ Create dendrogram

Where to decide clusters?

Options for clustering

Distance measurements:

- Euclidean dist
- Sq Euclid. dist.
- Manhattan
- Max dist
- Mahalanobis dist.

} experiment to find best results

↳ valid based on certain properties

→ what are some properties distances must have?

1) $d(x, y) > 0$
2) iff 1 then $x = y$
3) $d(x, y) = d(y, x)$
4) $d(x, z) \leq d(x, y) + d(y, z)$

→ How to join clusters together?

→ Single Linkage (closest points)

→ complete linkage (furthest pts)

→ mean dist (UPGMA)

→ Ward's criterion

↳ Minimise increase in variance

$$cost = \frac{n_a n_b}{\sum_{i \in \{a,b\}} n_i} (\bar{a} - \bar{b})^2$$

↳ Problem w/ single linkage

- always grab nearest obj but miss pattern

Why does this happen?

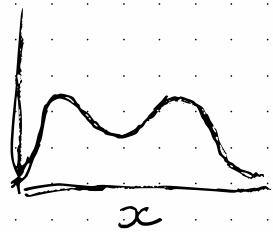
↳ Always grabbing obj that's very close so it looks like no new cluster is forming

Gaussian Mixtive Models

↳ form of density estimation

- approx of prob dist of our data

- used when data is multi modal



hmm...

↳ estimate w/ multiple normal distributions?

yes! $p(X=x) = \sum_{i=0}^{n} \pi_i \, N(\mu_i, \Sigma_i)$

- π_i: weight of norm
- μ_i: mean
- Σ_i: cov matrix

NB: - π_i = prob that X belongs to ith norm.

- $\int p(X=x)\,dx = 1$
  - ↳ since each N has int of 1
  - ↳ $\sum_i \pi_i = 1$

$\vec{\pi}$ is thus its own PMF

→ Training a Gaussian Mixture Model

· Similar to training kmn

2 steps:

1) Calculate responsibilities (γ)

$$\gamma_k^{(n)} = p(z^{(n)} \mid x)$$

$$= \frac{\pi_k \, N(x^{(n)} \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{k} \pi_j \, N(x^{(n)} \mid \mu_i, \Sigma_i)}$$

prop of 1 norm over responsibility of all norms

2) Given responsibilities, recalculate the rest

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} x^{(n)}$$ ← mean of resp. pt

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (x^{(n)} - \mu_k)(x^{(n)} - \mu_k)^T$$

norm for sample size

resp of kth mean

sq dist. of each pt. from mean

$$\pi_k = \frac{N_k}{N} \quad \text{w/} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)}$$

↳ renorm π or something

→ Comparison betw. GMM & k-means

↳ very similar algorithm

1. initialize points
2. evaluate data
3. update
4. rinse & repeat until convergence

k-means → no π var, each cluster has equal weight.

Σ vs β → Σ is full cover matrix
↳ increases flexibility of shape

k-means is gmm where π is uniform & is perfectly spherical

→ Practical issues w/ gmms

→ what happens if var = 0

↳ if some var/covar in Σ is 0 then it becomes singular

(ie squishes space into lower dim than dim of full rank matrix)

Solution use diagonal covar

by assumption that each dim is indep then all covars are 0 w/ only vars on diag.

↳ Also assume all covars are equal

Also resolves local min.

→ Kernel density estimation

↳ fitting of prob. dist. w/ kernels

→ Simplest version is a histogram!

GMM

↳ kde using a normal dist as kernel

↳ possible to use various kerns

→ Expectation maximisation

~~MLE~~ MLE Max $P(X|\theta)$ or log $P(X|\theta) = L(\theta)$

↳ add hidden var z

$$\therefore P(X|\theta) = P(X, z|\theta)$$
$$= \Sigma_z P(X, z|\theta)$$
$$= \Sigma_z P(X|\theta, z) P(z|\theta) \text{ by LOTP}$$
$$\vdots$$

2 steps: Adjusting z & Θ

1. E-step

find dist of $z|X, \Theta_n$ ← current setting of Θ

2. M-step

find Θ that maximizes joint dist

$P(X, z|\Theta)$

↳ Max expected log lik. of this over dist. in step 1

ie $\Theta_{n+1} = \underset{\max}{\arg} \, E[\log P(X, z|\Theta]$

(Just use a library)