CSE 291B 1/6

High dimensional data analysis

Problems

- Clustering
- Prediction
- Feature extraction

Approaches

- Generative models, probabilistic inference
- Spectral methods, matrix diagonalization
- Graph-based, nearest neighbor methods
- Kernel methods
- Deep architectures, multilayer networks

} not mutually exclusive

Mathematical techniques

- Variational methods
- Convex optimization
- Matrix factorizations
- Beyond Gaussian models

Topic #1: probabilistic models for simultaneous clustering and dimensionality reduction

Clustering

From unlabeled inputs $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$ with $\vec{x}_n \in \mathbb{R}^D$,
how to map $\vec{x}_n \longrightarrow \{1, 2, \ldots, k\}$?

More generally:
how to map $\vec{x} \in \mathbb{R}^D \longrightarrow \{1, 2, \ldots, k\}$?

For now: assume k is specified.

Also known as "vector quantization"

(Map a vector to an integer.)

k-means clustering

- define clusters via prototypes
  $\vec{\mu}_i \in \mathbb{R}^D \quad i = 1, 2, \ldots, k$

$D = 2$
$k = 3$

How to choose prototype $\vec{\mu}_i$?
How to assign inputs $\vec{x}_n$?

- Optimization – attempt to minimize:

$$\mathcal{E}(\vec{\mu}_i, Y_{in}) = \sum_{n=1}^{N} \sum_{i=1}^{k} \|\vec{x}_n - \vec{\mu}_i\|^2 Y_{in}$$

($\vec{\mu}_i$: prototypes; Y_{in}: binary assignment matrix)

$Y_{in} \in \{0, 1\}$ $k \times N$ matrix

- k-means algorithm
  1) choose $\vec{\mu}_i$ at random
  2) assign $\vec{x}_n$ to nearest prototypes (for all n)
     set $Y_{in} = \begin{cases} 1 & \text{if } i = \operatorname{argmin} \|\vec{x}_n - \vec{\mu}_i\|^2 \\ 0 \end{cases}$
  3) recompute prototypes (for all i)
     set $\vec{\mu}_i = \dfrac{\sum_n Y_{in} \vec{x}_n}{\sum_n Y_{in}}$ (mean of ith cluster)
  4) go to #2; repeat until convergence

- New inputs can be assigned to nearest prototype. $\vec{x} \in \mathbb{R}^D$

Guaranteed to converge? Yes, but only to local minimum of $\mathcal{E}(\mu, Y)$
Step 2 minimizes $\mathcal{E}$ wrt Y_{in}. At each iteration, cost decreases
Step 3 minimizes $\mathcal{E}$ wrt $\vec{\mu}_i$. (or stays the same).

Gaussian mixture modeling

Assume inputs $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$ are sampled (i.i.d.) from a joint probability density function (PDF) $P(\vec{x})$

Model $P(\vec{x})$ by belief network (BN) with hidden variable.

Estimate model to maximize log-likelihood: $\mathcal{L} = \sum_n \log P(\vec{x}_n)$

- Belief network

$z \in \{1, 2, \ldots, k\}$ hidden cluster label

$\vec{x} \in \mathbb{R}^D$ observed

- Conditional probability distributions

$P(z=i) = \pi_i$ fraction of data in ith cluster

$$P(\vec{x} \mid z=i) = \frac{1}{(2\pi)^{D/2}\sqrt{\det \Sigma_i}} \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x}-\vec{\mu}_i)\right\}$$

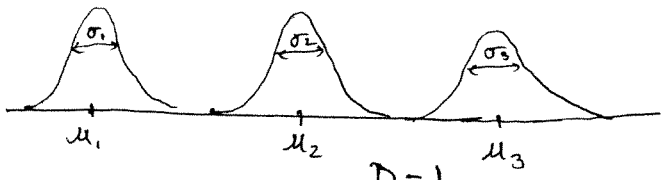
multivariate gaussian PDF
cluster-dependent mean $\vec{\mu}_i$
cluster-dependent covariance matrix Σ_i

- Marginal distribution

$$P(\vec{x}) = \sum_i P(\vec{x}, z=i) = \sum_i \underbrace{P(\vec{x} \mid z=i)}_{\text{gaussian}} P(z=i)$$

Gaussian mixture model (GMM)

$P(\vec{x})$ is <u>not</u> gaussian.



D = 1

- Aside: ML estimation for single Gaussian PDF

$$\vec{\mu} = \frac{1}{N}\sum_{n=1}^{N} \vec{x}_n$$

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N} (\vec{x}_n - \vec{\mu})(\vec{x}_n - \vec{\mu})^T$$

- EM algorithm for GMM
  - E-step: compute posterior probability (i.e. of hidden given observed)

$$P(z=i \mid \vec{x}_n) = \frac{P(\vec{x}_n \mid z=i)\, P(z=i)}{\sum_j P(\vec{x}_n \mid z=j)\, P(z=j)} \quad \text{(Bayes rule)}$$

  - M-step: update parameters of GMM

$$\vec{\mu}_i \leftarrow \frac{\sum_{n=1}^{N} P(z=i \mid \vec{x}_n)\, \vec{x}_n}{\sum_{n=1}^{N} P(z=i \mid \vec{x}_n)}$$

weighted mean of points "softly" assigned to ith cluster

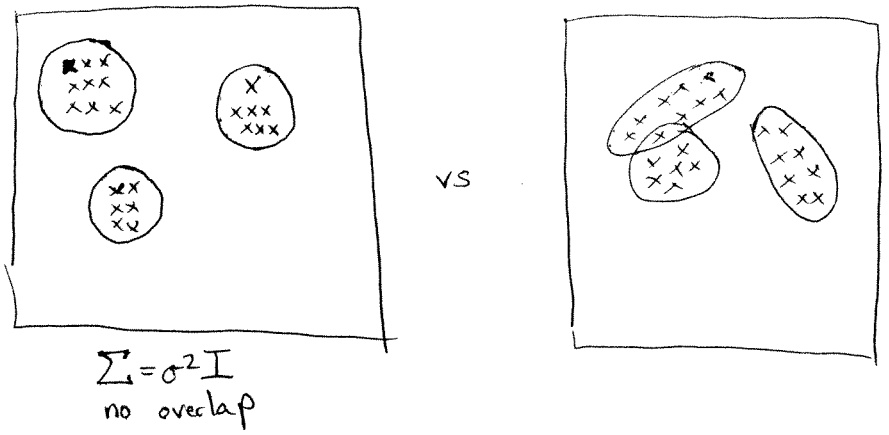
$$\Sigma_i \leftarrow \frac{\sum_{n=1}^{N} P(z=i \mid \vec{x}_n)\, (\vec{x}_n - \vec{\mu}_i)(\vec{x}_n - \vec{\mu}_i)^T}{\sum_{n=1}^{N} P(z=i \mid \vec{x}_n)}$$

$$P(z=i) = \pi_i \leftarrow \frac{1}{N} \sum_{n=1}^{N} P(z=i \mid \vec{x}_n)$$

  - Converges to local maximum of log-likelihood $\sum_n \log P(\vec{x}_n)$

- Relation to k-means
  - Suppose clusters have:
    - same shape $\Sigma_i = \Sigma$ for all i
    - spherical symmetry $\Sigma = \sigma^2 I$ ← identity matrix ($D \times D$)
    - no overlap

↑ global scalar variance



$\Sigma = \sigma^2 I$
no overlap

In "winner-take-all" limit $\sigma^2 \to 0$ of hard clustering, EM algorithm reduces to k-means.

- k-means implicitly making strong parametric assumptions

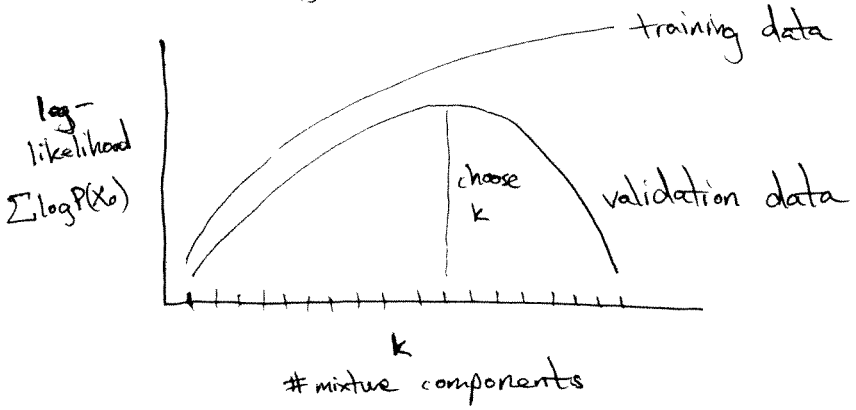
- How to choose k?

  Simplest method: cross validation

  Divide data into training & validation set.

  Estimate GMMs for different k on training set.

  Compute log-likelihoods:



- Curse of dimensionality

  $O(D^2)$ to store full covariance matrix

  $O(D^3)$ to invert " " "

  $O(2^D)$ to grid or sample uniformly D-dimensional space

How to exploit low dimensional structure in high dimensional data?

Next lecture: answer for GMMs