11/2/2023 Finishing GMMs, Starting Dimensionality Reduction

Reminder: Naive Bayes

Dataset: $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

maximize log likelihood of the data:

$$\sum_{i=1}^{n} \log p(x^{(i)}, y^{(i)}) = \sum_{i=1}^{n} \log p(y^{(i)}) + \log p(x^{(i)} \mid y^{(i)})$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}[y^{(i)} = j] \left(\log P(Y=j) + \log p(x^{(i)} \mid Y=j) \right)$$

Back to GMMs: Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$

Idea: Replace the $\mathbb{I}[z_i = j]$ quantity with R_{ij} = inferred probability that $z_i = j$

If: $R_{i1} = 0.8$, $R_{i2} = 0.2$,

maximize

$$0.8 \cdot \left[\log P(x^{(i)}, z_i = 1)\right] + 0.2 \left[\log P(x^{(i)}, z_i = 2)\right]$$

this is the objective called Expected Complete log likelihood:

$$ECLL(\pi_{1:k}, \mu_{1:k}, \Sigma_{1:k})$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \log P(X_i = x^{(i)}, Z_i = j; \pi_{1:k}, \mu_{1:k}, \Sigma_{1:k})$$

Goal: maximize ECLL wrt. $\pi_{1:k}$, $\mu_{1:k}$, $\Sigma_{1:k}$

Today: Compute optimal μ_j, intuit optimal π_j, Σ_j

Plan: Take ∇_{μ_j} ECLL, set it to 0

$$= \sum_{i=1}^{n} \nabla_{\mu_j} R_{ij} \log P(x_i = x^{(i)}, z_i = j)$$

$$= \sum_{i=1}^{n} R_{ij} \nabla_{\mu_j} \left[\cancel{\log P(z_i = j)} + \log P(x_i = x^{(i)} \mid z_i = j) \right]$$

only depends on π_j not μ_j

$$\cancel{\frac{1}{(2\pi)^{d/2}}} \; \cancel{\frac{1}{\sqrt{\det(\Sigma_j)}}} \cdot \exp\left(-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)\right)$$

constant

depends on Σ_j not μ_j

$$= \sum_{i=1}^{n} R_{ij} \nabla_{\mu_j} \left[-\frac{1}{2} (x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j) \right]$$

Fact: $\nabla_x x^T A x = 2Ax$

$$= -\frac{1}{\cancel{2}} \sum_{i=1}^{n} R_{ij} \cdot \cancel{2}\, \Sigma_j^{-1} \cdot (x^{(i)} - \mu_j) \cdot \cancel{-1} = 0$$

$$= \sum_{i=1}^{n} R_{ij} \cancel{\Sigma_j^{-1}} (x^{(i)} - \mu_j) = 0$$

multiply both sides by Σ_j (on left)

$$= \sum_{i=1}^{n} R_{ij} \, x^{(i)} = \sum_{i=1}^{n} R_{ij} \mu_j = \left(\sum_{i=1}^{n} R_{ij} \right) \mu_j$$

$$\mu_j = \boxed{\frac{\sum_{i=1}^{n} R_{ij} \, x^{(i)}}{\boxed{\sum_{i=1}^{n} R_{ij}}}}$$

weighted average of $x^{(i)}$ weighted by probability that it's in cluster j

P(Example 1 in cluster j)
+ P(Example 2 in cluster j)
+ ...

Expected # of points in cluster j

M-step formulas for π_j & Σ_j:

$$\pi_j = \frac{\boxed{\sum_{i=1}^{n} R_{ij}}}{n}$$

Soft version of counting

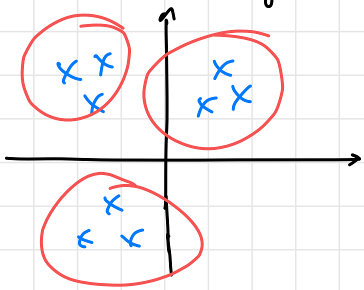
$\frac{\text{\# of points in cluster j}}{\text{total \# of points}}$

$$\Sigma_j = \boxed{\frac{\sum_{i=1}^{n} R_{ij} \left(x^{(i)} - \mu_j\right)\left(x^{(i)} - \mu_j\right)^T}{\sum_{i=1}^{n} R_{ij}}}$$

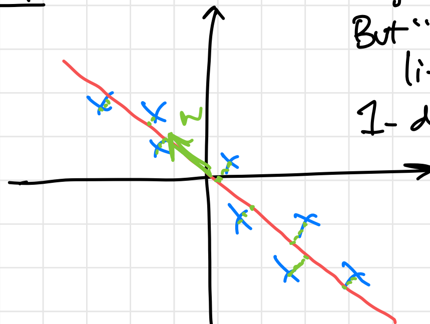
Note: Formula for covariance matrix of dataset is $\frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu)(x^{(i)} - \mu)^T$

* This is covariance but weighted by probability of each point being in cluster j

Dimensionality Reduction



Clustering



Original data: $\mathbb{R}^2$
But "mostly" lies in 1-dimensional subspace

Dimensionality Reduction:
Find a low-dimensional subspace that preserves most of the info in our dataset

Method: Principal Component Analysis (PCA)

Common use case: high dim data → 2D for visualization

Starting point: Try to find best 1-D projection

Key assumption: Data has mean 0

ie $\frac{1}{n} \sum_{i=1}^{n} x^{(i)} = 0$

Ensure by computing mean, subtract it from every example

What is our parameter?

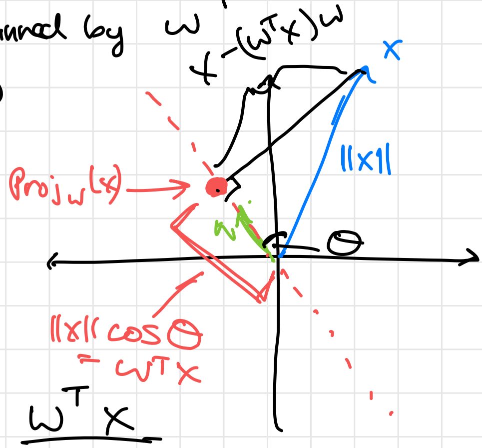
We have 1 parameter vector $w \in \mathbb{R}^d$ that defines the 1-D subspace we will project onto

Force $\|w\| = 1$

What is a good loss function for w?

"<u>Reconstruction Error</u>": How well can we reconstruct $\{x^{(1)}, \dots, x^{(n)}\}$ based on only the projections of $x^{(1)}, \dots, x^{(n)}$ onto subspace spanned by w

Math: $\boxed{\text{minimize}}$ $\sum_{i=1}^{n} \| x^{(i)} - \underbrace{\text{Proj}_w}_{\text{project onto span}(w)} (x^{(i)}) \|^2$



$\boxed{\text{minimize}}$

$$= \sum_{i=1}^{n} \| x^{(i)} - (w^T x^{(i)}) x^{(i)} \|^2$$

$$\cos\theta = \frac{w^T x}{\underbrace{\cancel{\|w\|}}_{1} \|x\|}$$

$$\Rightarrow \|x\| \cos\theta = w^T x$$

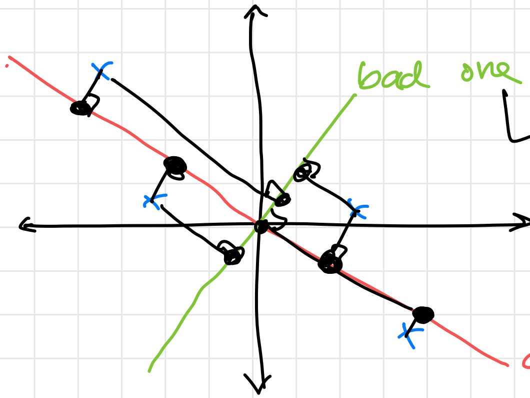
$$\Rightarrow \text{Proj}_w(x) = \underbrace{(w^T x)}_{\text{right distance}} \underbrace{w}_{\text{right direction}}$$

Equivalent view: minimizing reconstruction error is same as maximizing variance of points after projection

Pythagorean Theorem: $\underbrace{(w^T x)^2}_{\text{want to maximize}} + \underbrace{\text{ReconError for example } i}_{\text{want to minimize}} = \underbrace{\|x\|^2}_{\text{Fixed}}$

New goal:
Maximize $\sum_{i=1}^{n} (w^T x^{(i)})^2$

Note: $\frac{1}{n} \sum_{i=1}^{n} (w^T x^{(i)})^2 = \text{Variance of } w^T x$ (since $\mathbb{E}[x] = 0$)



large distance = high reconstruction error ✗

less spread out after projection = low variance ✗

good subspace

small distance to line = small reconstruction error ✓

resulting points spread out = high variance ✓