3/21/2022 : K-Means Clustering

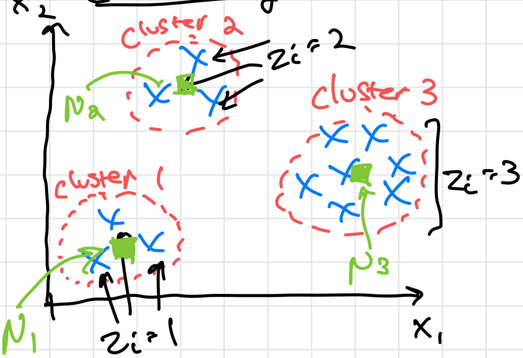
Machine Learning

Supervised Learning	Unsupervised Learning
Training Dataset	Dataset only contains x's
$D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$	$D = \{x^{(1)}, \ldots, x^{(n)}\}$
$x^{(1)}$: input to model; $y^{(1)}$: correct output	No input → output mapping learnable from dataset
Goal: Learn a function from $x \to y$	Goal: Learn about structure of dataset

Unsupervised Learning structures:

1. Clusters
2. Sequential structure
3. Subspace structure (low-dimensional structure)
4. Similarity or relationships structure

Clustering



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

Assume K clusters $1, \ldots, K$

Goal of clustering:

Produce an assignment $z_1, \ldots, z_n$ where $z_i \in \{1, \ldots, K\}$ and denotes cluster assigned to $x^{(i)}$

Need loss function that measures how bad an assignment is

Today: K-Means clustering

Idea: Each cluster has a centroid μ_j for $j = 1, \ldots, K$

Loss = how far each $x^{(i)}$ is to its assigned centroid

Loss function more formally:

$$L(\underbrace{z_{1:n}}_{\text{assignments}}, \underbrace{\mu_{1:k}}_{\text{centroids}}) = \sum_{i=1}^{n} \| x^{(i)} - \underbrace{\mu_{\boxed{z_i}}}_{\text{centroid of cluster for example } i} \|^2$$

cluster ID assigned to example i

"Reconstruction Error"

If we replaced each $x^{(i)}$ with its cluster centroid, how wrong is that?

Can't directly do gradient descent – z_i's are discrete

Strategy: <u>Alternating minimization</u>

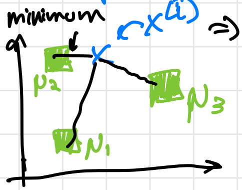
1. start with random choice of $\mu_1, \dots, \mu_k$

Alternate until convergence:

2. Choose $z_{1:n}$ to minimize L given choice of $\mu_{1:k}$
3. Choose $\mu_{1:k}$ to minimize L given choice of $z_{1:n}$

Step 1: Choose each μ_j to be a random example in dataset

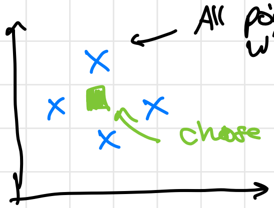
Step 2: Minimizing w.r.t. $z_{1:n}$



$\Rightarrow z_i = 2$

For each i, set $z_i = \underset{j=1\dots k}{\operatorname{argmin}} \| x^{(i)} - \mu_j \|^2$

Step 3: minimizing w.r.t. $\mu_{1:k}$



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

$$= \sum_{j=1}^{k} \sum_{i: z_i = j} \| x^{(i)} - \mu_j \|^2$$

Now consider each j independently

For $j=1$: $\nabla_{\mu_1} L(z_{1:n}, \mu_{1:k}) = \nabla_{\mu_1} \sum_{i: z_i = 1} \| x^{(i)} - \mu_1 \|^2$

$$= \sum_{i: z_i = 1} \cancel{2} (x^{(i)} - \mu_1) \cdot \cancel{(-1)} = 0$$

$$\mu_1 = \frac{1}{|\{i : z_i = 1\}|} \cdot \sum_{i: z_i = 1} x^{(i)}$$

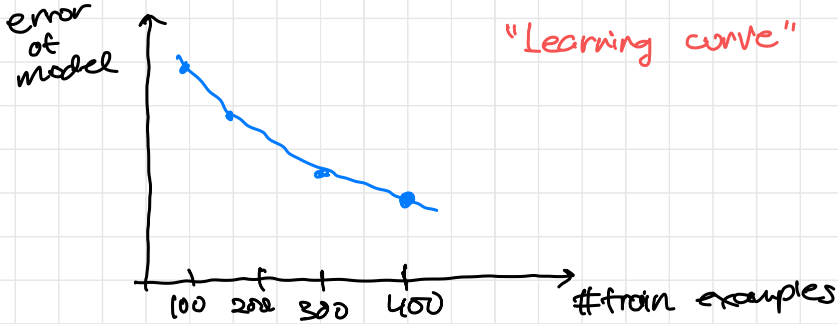
Average of all points in cluster 1

Note: Eventually we will converge
At every step L decreases (or stays same)

This is not guaranteed to find global optimum

Announcements

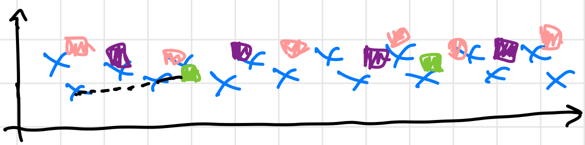
- Midterm graded
- Progress reports due Thurs
- HW 3 due 4/11
- "How much training data?"



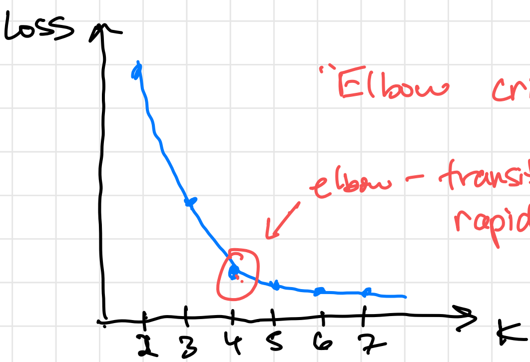
"Learning curve"

How do you choose K?

Wrong Answer: Choose based on dev set

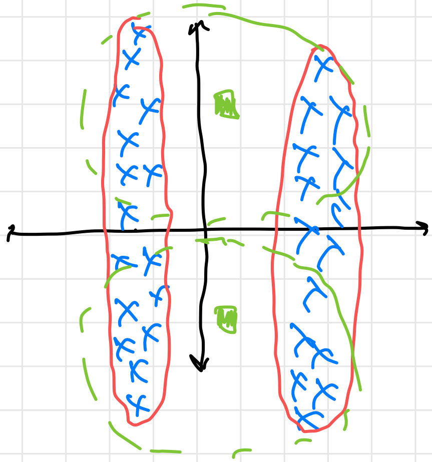


Larger K essentially always decreases loss



"Elbow criterion"

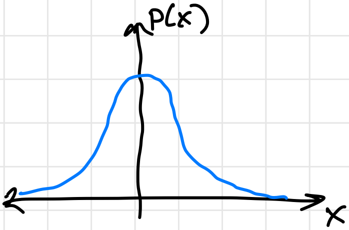
elbow - transition between loss going down rapidly & going down slowly



K-Means is looking for spherical clusters (because it uses Euclidean distance)

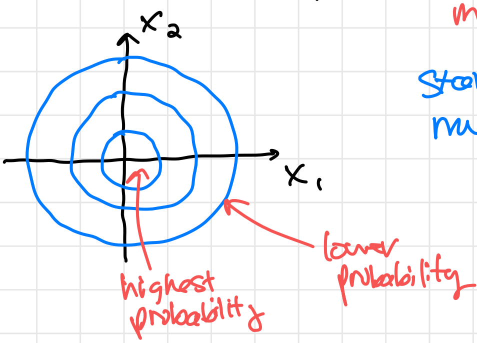
Need new algorithm that can learn both <u>location</u> AND <u>shape</u> of clusters

<u>Plan</u>: Describe clusters as multivariate Gaussian distribution



Standard univariate Gaussian

$$\left(\underbrace{\mu}_{\text{mean}} = 0,\ \underbrace{\sigma^2}_{\text{variance}} = 1\right)$$



Standard multivariate Gaussian

$$\left(\underbrace{\mu}_{\text{mean}} = \begin{bmatrix}0\\0\end{bmatrix},\ \underbrace{\Sigma}_{\text{covariance matrix}} = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right)$$

Covariance Matrix

$$\Sigma = \begin{pmatrix} Var(x_1) & Cov(x_1, x_2) \\ Cov(x_1, x_2) & Var(x_2) \end{pmatrix}$$

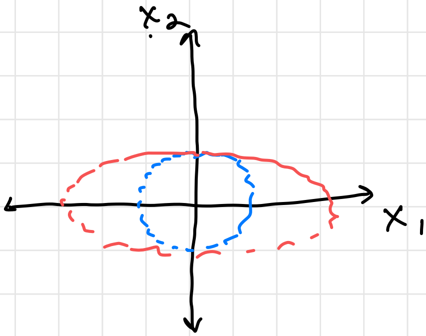
$$Var(x_1) = \mathbb{E}\left[(x_1 - \mathbb{E}[x_1])^2\right]$$

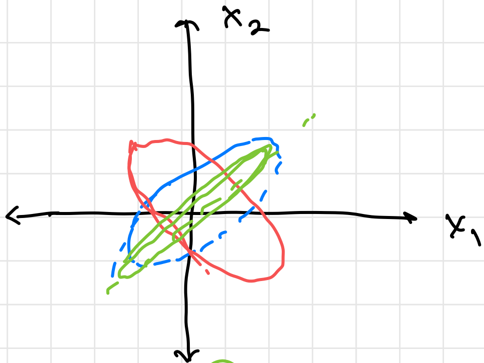
$$Cov(x_1, x_2) = \mathbb{E}\left[(x_1 - \mathbb{E}[x_1])(x_2 - \mathbb{E}[x_2])\right]$$

$$\text{Correlation}(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)} \cdot \sqrt{\text{Var}(X_2)}}$$

Cov > 0 $\iff$ positively correlated

Cov < 0 $\iff$ negatively correlated





$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & -1/2 \\ -1/2 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$