machine learning

supervised learning

training dataset

$D = \{(x^{(1)}, y^{(1)}) \cdots (x^{(n)}, y^{(n)})\}$

Learn a function from

$X \to Y$

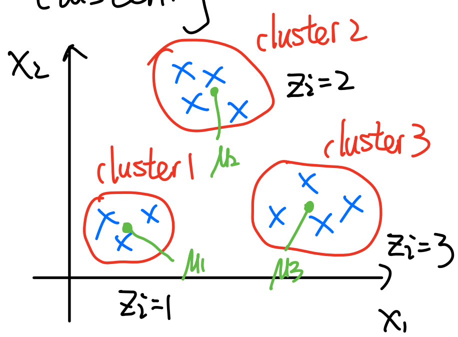
unsupervised learning

$D = \{x^{(1)}, \dots, x^{(n)}\}$

goal: learn about structure of dataset

1. clusters
2. sequential structure
3. subspace / low-dimensional structure
4. similarity or relationships structure

Clustering



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

Assume K clusters $1, \dots, k$

Goal of clustering:

produce an assignment $z_1, \dots, z_n$
where $z_i \in \{1, \dots, k\}$ and denotes
cluster assigned to $x^{(i)}$

Need loss function that measures how bad an assignment is.

K-means clustering:

each cluster has a centroid μ_j for $j = 1, \dots, k$

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

$$L(z_{1:n}; \mu_{1:k}) = \sum_{i=1}^{n} \| x^{(i)} - \mu_{z_i} \|^2$$

cluster ID assigned to $x^{(i)}$

centroid for $x^{(i)}$

"reconstruction error"

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

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

Strategy: alternating minimization

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

alternate until convergence

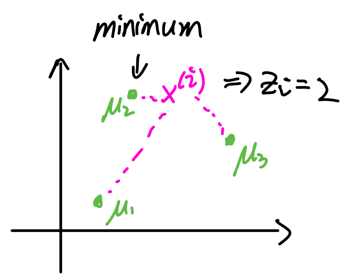
② choose $z_{1:n}$ to minimize L given $\mu_1, \dots, \mu_k$

③ choose $\mu_{1:k}$ to minimize L given $z_1, \dots, z_n$

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

step ②: minimize w.r.t. $z_{1:n}$

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

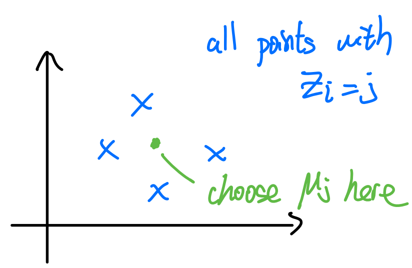


step ③: minimize w.r.t. $\mu_{1:k}$

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

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

consider each j independently



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

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

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

← average of all points in cluster 1

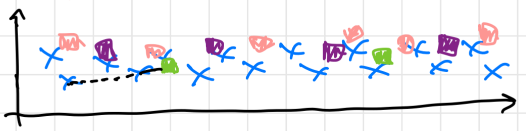
Note: 1. eventually will converge

2. at every step L decreases or stay the same

3. not guaranteed to find global optimal

How to choose k?

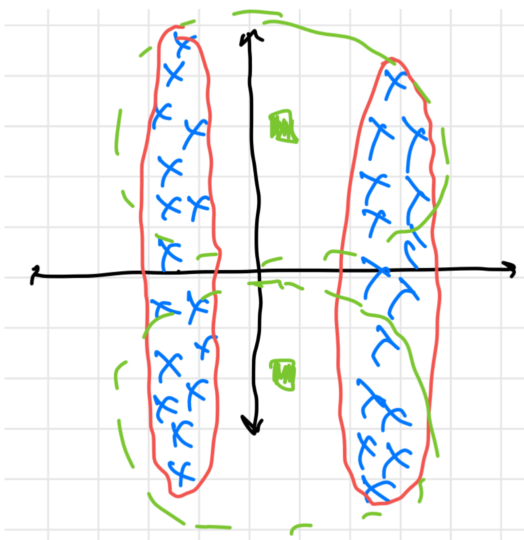
Wrong Answer: Choose based on dev set



Larger k essentially always decreases loss

"Elbow criterion"

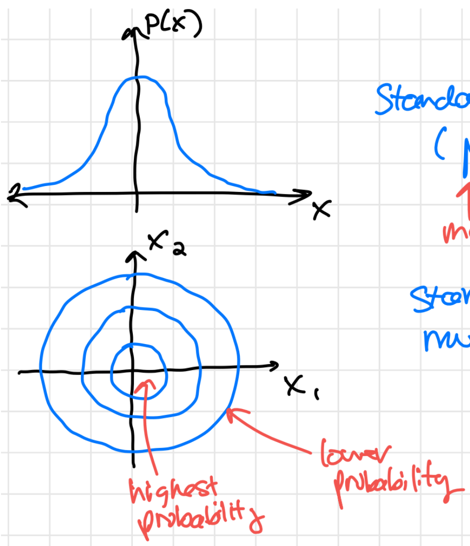
elbow - transition between loss going down rapidly & going down slowly



K-means is looking for special clusters (because it uses Euclidean distance)

Need new algorithm that learn both location and shape of clusters

Plan : describe clusters as multivariate Gaussian dist.



Standard univariate Gaussian

$(\mu = 0, \ \sigma^2 = 1)$

mean　　variance

Standard multivariate Gaussian

$\left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$

mean　　covariance matrix

Covariance Matrix $\Sigma = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_2) \end{pmatrix}$

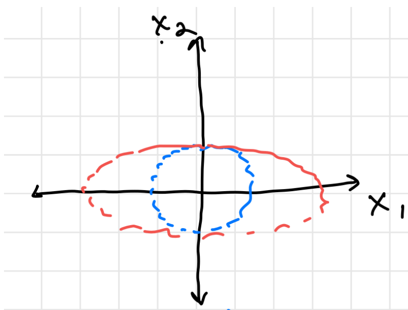
$Var(X_1) = \mathbb{E}\left[(X_1 - \mathbb{E}[X_1])^2 \right]$

$Cov(X_1, X_2) = \mathbb{E}\left[(X_1 - \mathbb{E}X_1)(X_2 - \mathbb{E}X_2) \right]$

$Correlation(X_1, X_2) = \dfrac{Cov(X_1, X_2)}{SD(X_1)\ SD(X_2)}$

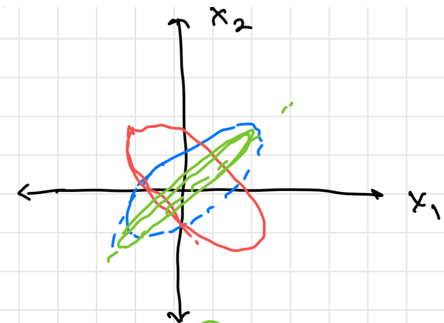
$\text{Cov} > 0 \iff$ pos correlated

$\text{Cov} < 0 \iff$ neg correlated



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

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



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

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

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