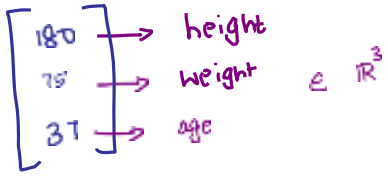
GOAL: UNSUPERVISED LEARNING

↳ REPRESENTATION LEARNING

- Given a set of "data points", "understand" some thing "useful" about them.

DATA POINTS → VECTORS



understand ? useful ?

- RUNNING THEME

" COMPREHENSION IS COMPRESSION ".

- GEORGE CHAITIN

→ understand

→ Learn

Problem:

Input: $\{x_1, x_2, \ldots, x_n\}$ $\quad x_i \in \mathbb{R}^d$ ← # features

output: Some "COMPRESSED" representation of the dataset.

Example:

$$\left\{ \underbrace{\begin{bmatrix} -5 \\ 2.5 \end{bmatrix}}_{x_1} \quad \underbrace{\begin{bmatrix} -2 \\ 1 \end{bmatrix}}_{x_2} \quad \underbrace{\begin{bmatrix} 4 \\ -2 \end{bmatrix}}_{x_3} \quad \underbrace{\begin{bmatrix} 5 \\ -2.5 \end{bmatrix}}_{x_4} \right\}$$

Question: How many numbers are needed to store this dataset? 8

Note:

Using this representation we can "exactly" reconstruct the dataset

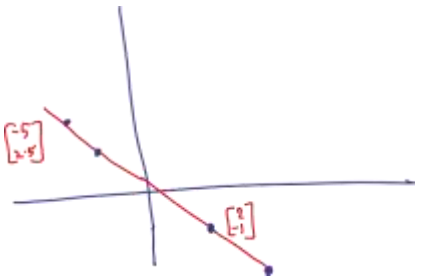
Representative

$$\begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Co-efficient

$$\{2.5,\ 1,\ -2,\ -2.5\}$$

GEOMETRIC VIEW



R

$$\begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

C

$$\{2.5,\ 1,\ -2,\ -2.5\}$$

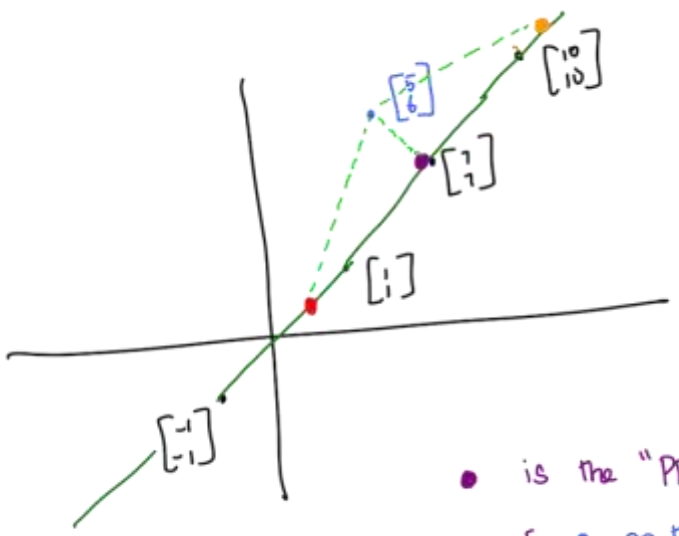
$$\begin{bmatrix} -2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$$

$$\{2.5\sqrt{5},\ \sqrt{5},\ -2\sqrt{5},\ -2.5\sqrt{5}\}$$

NOTE: R can be any point on the line (other than $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$)

original set of numbers $2n$

Compressed representation $2 + n$



Repr

$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$

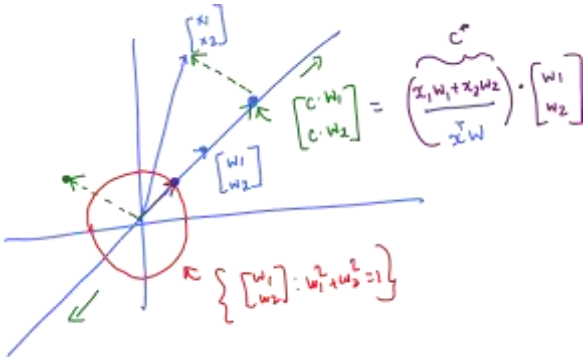
Co-eff

$\{(7,7), (1,1), (-1,-1), (10,10)\}$

$7 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 7 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 7 \end{bmatrix}$

$\text{length}\left(\begin{bmatrix} a \\ b \end{bmatrix}\right) = \sqrt{a^2+b^2}$

● is the "PROJECTION" of ● on the green line.



$$\begin{bmatrix} cw_1 \\ cw_2 \end{bmatrix} + \begin{bmatrix} ? \\ ? \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} x_1 - c \cdot w_1 \\ x_2 - c \cdot w_2 \end{bmatrix} \Rightarrow \text{ERROR VECTOR}$$

$$\underset{c}{\text{minimize}} \quad (x_1 - c \cdot w_1)^2 + (x_2 - c \cdot w_2)^2$$

$$f(c) = (x_1)^2 + c^2 (w_1)^2 - 2c x_1 w_1 + (x_2)^2 + c^2 w_2^2 - 2c x_2 w_2$$

$$f'(c) = 2c\,w_1^2 - 2x_1w_1 + 2c\,w_2^2 - 2x_2w_2$$

$$= 0$$

$$\Rightarrow \quad c^* = \frac{x_1w_1 + x_2w_2}{w_1^2 + w_2^2}$$

INNER PRODUCT / DOT - PRODUCT

$$\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right) = x_1w_1 + x_2w_2$$

$$\begin{matrix} x_1 \\ \vdots \\ x_d \end{matrix}, \begin{matrix} w_1 \\ \vdots \\ w_d \end{matrix} = x_1w_1 + \cdots + x_dw_d = x^T w$$

Summary

- "Comprehension is compression"
- Rep, Co-efficients
- Proxy → $(x^T W)\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ (Projection)

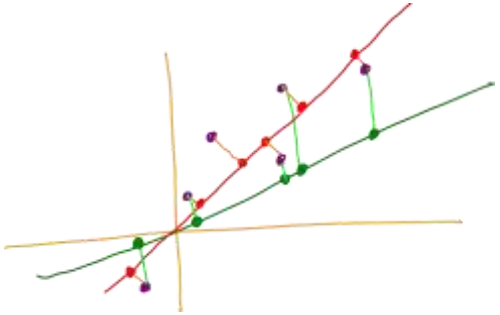
Question - Given a line & a point, we know how to find proxy for the point along the line

- who gives us the line ?

LECTURE 3

Goal: Develop a way to find a "compressed" representation of data when points non-necessarily fall on a line

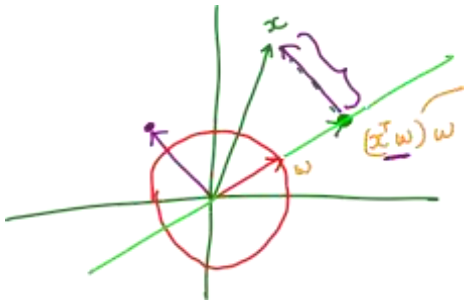
↳ RULE not an EXCEPTION



Goal : Find the line that has least "error"

↳ Reconstruction

Dataset $\{x_1, x_2, \cdots, x_n\}$ $x_i \in \mathbb{R}^d$



$x_1 w_1 + x_2 w_2 + \cdots + x_d w_d$

$(x^T w) \cdot \vec{w} + ? = \vec{x}$

$x - (x^T w)w$

$$\text{ERROR}(\text{line}, \text{Dataset}) = \sum_{i=1}^{n} \text{error}(\text{line}, x_i)$$

$$= \sum_{i=1}^{n} \text{length}^2\left(x - (x^T w)w\right)$$

$$\text{error}(w, D) = \sum_{i=1}^{n} (x - (x^T w)w)^T (x - (x^T w)\cdot w)$$

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

GOAL:

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

$$f(w) = \frac{1}{n} \sum_{i=1}^{n} \left(x - (x^T w)\cdot w\right)^T \left(x - (x^T w)\cdot w\right)$$

$$x^T\left((x^T w)\cdot w\right)$$

$$(x^T w)\cdot(x^T w)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left(x^T x - (x^T w)^2 - (x^T w)^2 + (x^T w)^2 \right)$$

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

$$g(w) = -\frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2$$

$$\max_{w:\, w^T w = 1} R(w) = \frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_i^T w)\,(x_i^T w)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(w^T x_i)\cdot(x_i^T w)$$

$$= \frac{1}{n}\sum_{i=1}^{n} w^T (x_i x_i^T) w$$

(w^T: $1\times d$, $x_i x_i^T$: $d\times d$, w: $d\times 1$)

$$= w^T \left[\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T\right] w$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}_{d\times 1} \qquad \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}_{d\times 1}$$

$$\underset{1\times d}{[x_1 \;\cdots\; x_d]}$$

$$w^T(a+b+c)$$

$$w^T a + w^T b + w^T c.$$

$$\frac{1}{n}\sum_{i=1}^{n} w^T (x_i x_i^T) w$$

$$\max_{w:\ w^T w = 1} \quad w^T C w \qquad -(*)$$

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

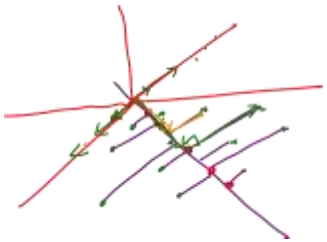
Co-variance matrix

- What is the solution to (*)?
- For the moment, lets say it's simple to solve (*).

So are we done?

$w_1 - (x^T w_1)$

$w_2 - (x^T w_2)$



$x \in \mathbb{R}^d$

↓ Find w

x becomes

$(x^T w)$ w

↓

$x - (x^T w) \cdot w$

might have information.

Possible Algorithm

Input: $\{x_1, \ldots, x_n\}$

$x_i \in \mathbb{R}^d$

→ Find "best" line

$w_1 \in \mathbb{R}^d$

→ Replace

$x \to x - (x^T w) w$

→ Repeat to obtain w_2, ..

$x \in \mathbb{R}^d$

↓ FIND w

$x \to (x^T w)\, w$

$\in \mathbb{R}^d \qquad \in \mathbb{R}^d$

↓ RESIDUE / ERROR

$x - (x^T w) \cdot w$

MIGHT NOT BE ERROR BUT HAS INFORMATION

RECAP

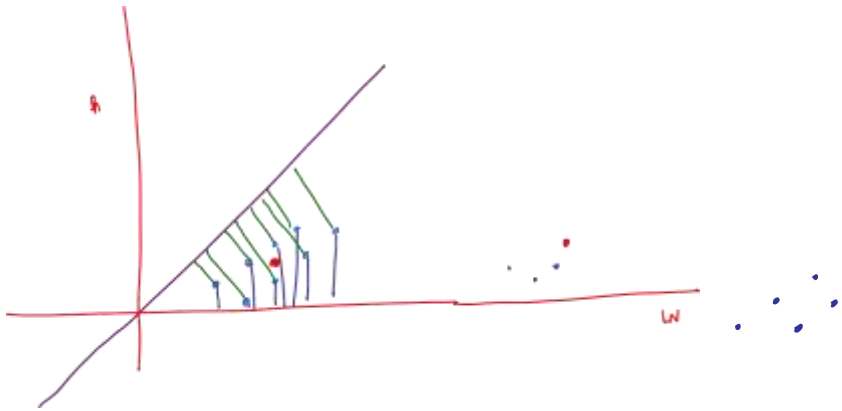
POSSIBLE ALGORITHM

INPUT DATASET $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$

- FIND "BEST" LINE w_1 $\mathbb{R}^d$
- REPLACE $x_i \to x_i - (x_i^T w_1) w_1$ $\forall i$
- REPEAT procedure to obtain $w_2, w_3, \ldots$

QUESTIONS

- We look only for lines passing through ORIGIN. Is this ok?
- How to solve $\max_{w : \|w\|^2 = 1} w^T C w$?
- How many times to repeat procedure ?
- Where exactly is "COMPRESSION" happening?
- What "REPRESENTATIONS" are we learning ?



$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$x_i' \to x_i - \mu \quad \forall i$$

$$\frac{1}{n}\sum_{i=1}^{n} x_i' = 0$$

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu) = \frac{1}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}\sum_{i=1}^{n}\mu$$

$$\mu - \mu = 0$$

$$x_1' \longrightarrow x_1 - (x_1^T w_1) w_1$$

$$x_2' \longrightarrow x_2 - (x_2^T w_1) w_1$$

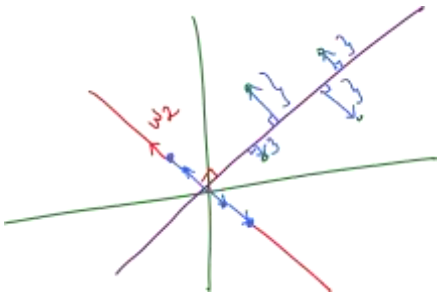
$$\vdots$$

$$x_n' \longrightarrow x_n' - (x_n'^T w_1) w_1$$

$$w_2 = \max_{\substack{w \\ \|w\|^2 = 1}} w^T C' w$$

$$C' = \frac{1}{n} \sum_{i=1}^{n} x_i' x_i'^T$$

QUESTION: What can we say about w_1 and w_2?



w_1

$$w_1^T w_2 = 0$$

By continuing the same procedure, we get $\{w_1, w_2, \ldots, w_k\}$ of ORTHOGONAL VECTORS (ORTHONORMAL)

RESIDUES after Round 1

$$\left\{ x_1 - (x_1^T w_1) w_1 , \ \ldots , \ x_n - (x_n^T w_1) w_1 \right\}.$$

$w_2 \rightarrow$ Fits Residues, $\quad w_1^T w_2 = 0.$

RESIDUES after Round 2.

$$\left\{ x_1 - (x_1^T w_1) w_1 - \left(\left(x_1 - (x_1^T w_1) w_1 \right)^T w_2 \right) w_2 , \ \ldots \right\}$$

$$\left\{ x_1 - (x_1^T w_1) w_1 - \left(x_1^T w_2 - (x_1^T w_1)(w_1^T w_2) \right) w_2 , \ \ldots \right\}$$

$$\left\{ x_1 - (x_1^T w_1) w_1 - (x_1^T w_2) w_2 , \ldots \right\}$$

$x_i \in \mathbb{R}^d$.

Residues after d rounds

$$\forall i \quad x_i \longrightarrow x_i - (x_i^T w_1) w_1 - (x_i^T w_2) w_2 - \ldots - (x_i^T w_d) w_d = 0.$$

$$x_i = (x_i^T w_1) w_1 + (x_i^T w_2) w_2 + \ldots + (x_i^T w_d) w_d \quad \forall i$$

$x_i \in \mathbb{R}^d$; $(x_i^T w_1) \in \mathbb{R}$, $w_1 \in \mathbb{R}^d$; $(x_i^T w_d) \in \mathbb{R}$, $w_d \in \mathbb{R}^d$.

$$x_i = \left(x_i^T \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right)\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \left(x_i^T \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}\right)\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots \left(x_i^T \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}\right)\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

What have we gained?

- If data lies in a "low" dimensional space, then residues become 0 much earlier than d rounds.

Dataset $= \{x_1, \ldots, x_n\}$ $\quad x_i \in \mathbb{R}^{100}$ $\forall i$

$$x_i = \underbrace{(x_i^T w_1) w_1 + (x_i^T w_2) w_2 + (x_i^T w_3) w_3}_{\in \mathbb{R}^{100}}$$

$x_i \in \mathbb{R}^{100}$ $\quad \in$

Rep. $\{w_1, w_2, w_3\}$

Co-eff

$x_i \in \mathbb{R}^{100} \rightarrow \left[(x_i^T w_1) \quad (x_i^T w_2) \quad (x_i^T w_3)\right] \in \mathbb{R}^3$

Original : $100 \times n$

Now : $\underbrace{100 \times 3}_{w_1, w_2, w_3} + 3 \times n$

$[a, b, c] \rightarrow [x^T w_1 \quad x^T w_2]$

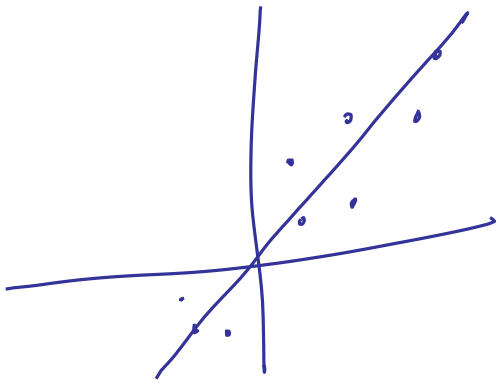
$x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$

$w_1 = \begin{bmatrix} w_{11} \\ w_{12} \\ w_{13} \end{bmatrix}$ $w_2 = \begin{bmatrix} w_{21} \\ w_{22} \\ w_{23} \end{bmatrix}$

$a \cdot w_{11} + b w_{12} + c w_{13}$

$a w_{21} + b w_{22} + c w_{23}$

QUESTION: What if data approximately low dimensional

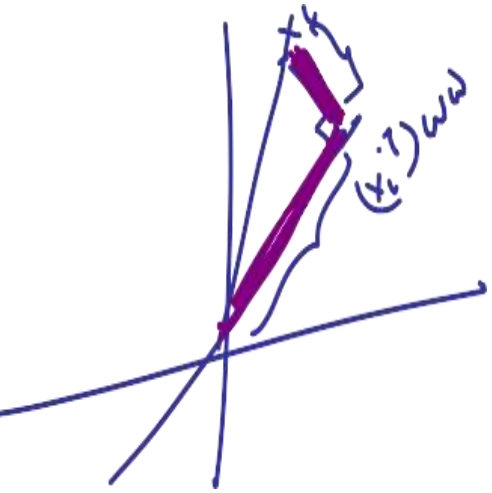


Summary So Far

Algorithm:
Given data $\{x_1, x_2, \ldots x_n\}$
-> Center data so that mean is 0 (origin shift)
-> Use the "magic" box to find the best line represented by w_1
-> Create dataset with residues
-> Iterate the process to find w_2, w_3, …

Observations:
- All the lines obtained from the above process are orthogonal to each other
- The residues must necessarily become 0 after 'd' rounds
- We gain (compression wise) if residues becomes 0 much earlier than d rounds.

Question:
What if data lives in an approximately low dimensional space?



PYTHAGOROUS

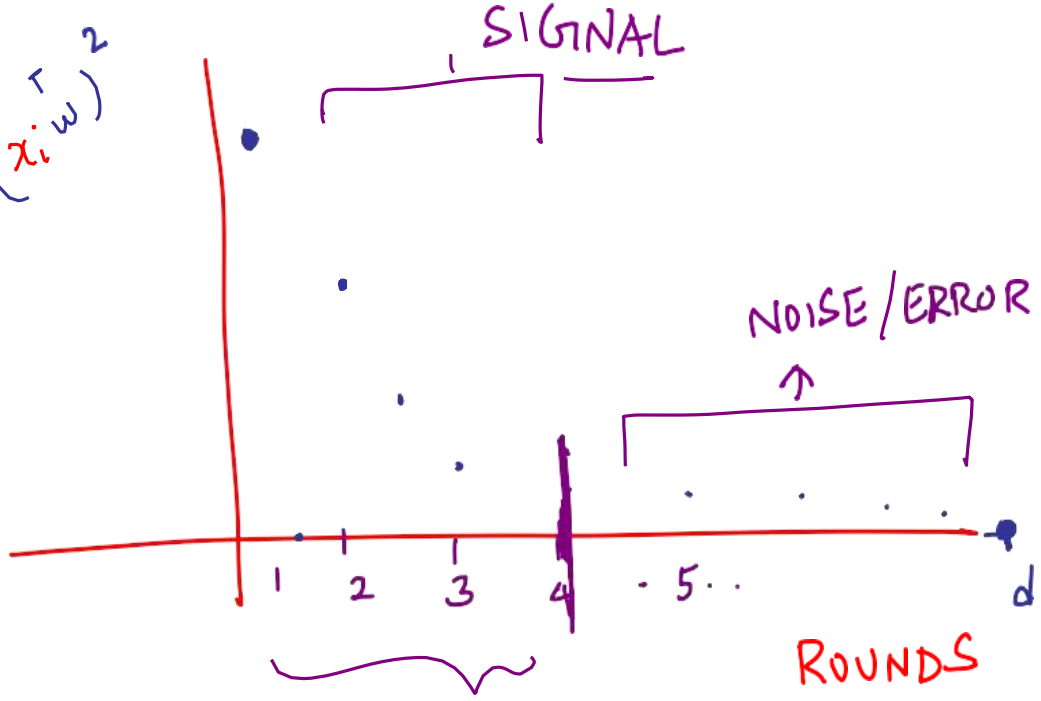
$$\forall i \quad \|x_i\|^2 = \|x_i - (x_i^T w)w\|_2^2 + \|(x_i^T w)w\|^2$$

$$\frac{1}{n}\sum_{i=1}^{n}\|x_i\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|x_i - (x_i^T w)w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2$$

Smaller the value
better the fit

larger the value
better the fit

$$\frac{1}{n}\sum_{i=1}^{n}(x_i \cdot w)^2$$



PCA

PRINCIPAL COMPONENT ANALYSIS.

Question

$$\max_{w \in \mathbb{R}^d, \|w\|^2 = 1} \quad w^T C w$$

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

x_i: $d \times 1$, x_i^T: $1 \times d$, $C \in \mathbb{R}^{d \times d}$

$\rightarrow$ CO-VARIANCE MATRIX

SOLUTION:

w_1 is eigenvector corresponding to the "largest" eigenvalue of C

[HILBERT'S MIN-MAX THEOREM]

In Fact the eigenvectors $\{w_1, \ldots, w_d\}$ of C form
$\lambda_1 \qquad \lambda_d$

an orthonormal basis

w_k -> Best line that we get in round k.

By definition

$$C w_1 = \lambda_1 w_1$$

$$w_1^T (C w_1) = w^T (\lambda_1 w_1)$$

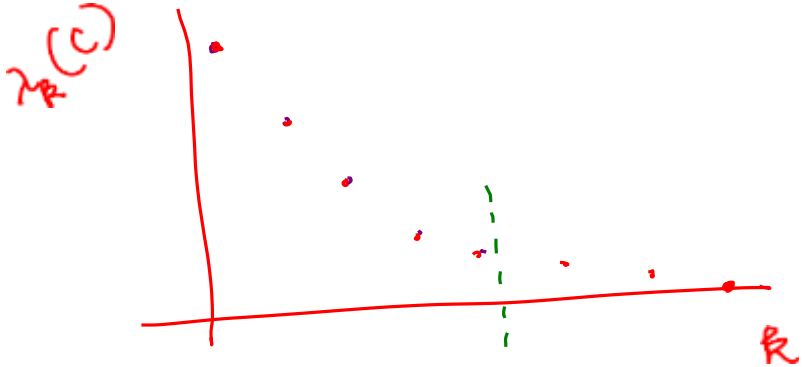
$$w_1^T C w_1 = \lambda_1$$

$$w_1^T \left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) w_1 = \lambda_1$$

$$\frac{1}{n} \sum_{i=1}^{n} (w_1^T x_i)(x_i^T w_1) = \lambda_1$$

$$\Rightarrow \boxed{\lambda_1 = \frac{1}{n} \sum_{i=1}^{n} (x_i^T w_1)^2} \rightarrow$$

Which is the term we used earlier



$$\left(\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \right) \geq 0.95$$

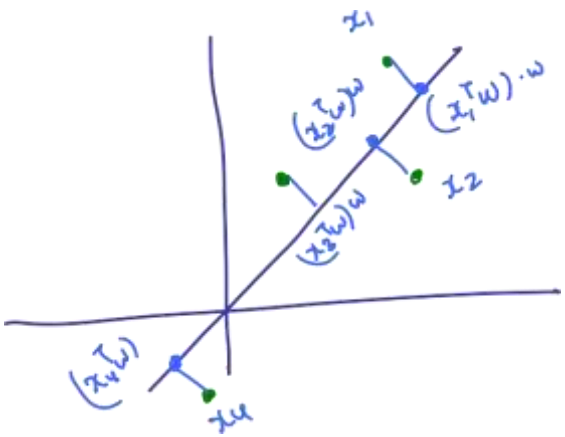
PCA

$\{w_1, \ldots, w_k\}$

PRINCIPAL COMPONENTS

$x_i \in \mathbb{R}^d \rightarrow [x_i^T w_1 \ \ldots \ x_i^T w_k]$

A DIFFERENT VIEW OF PCA



$x_1 \rightarrow (x_1^T w)\, w$

$\vdots$

$\{ (x_1^T w), \ldots, (x_n^T w) \}$

Average.

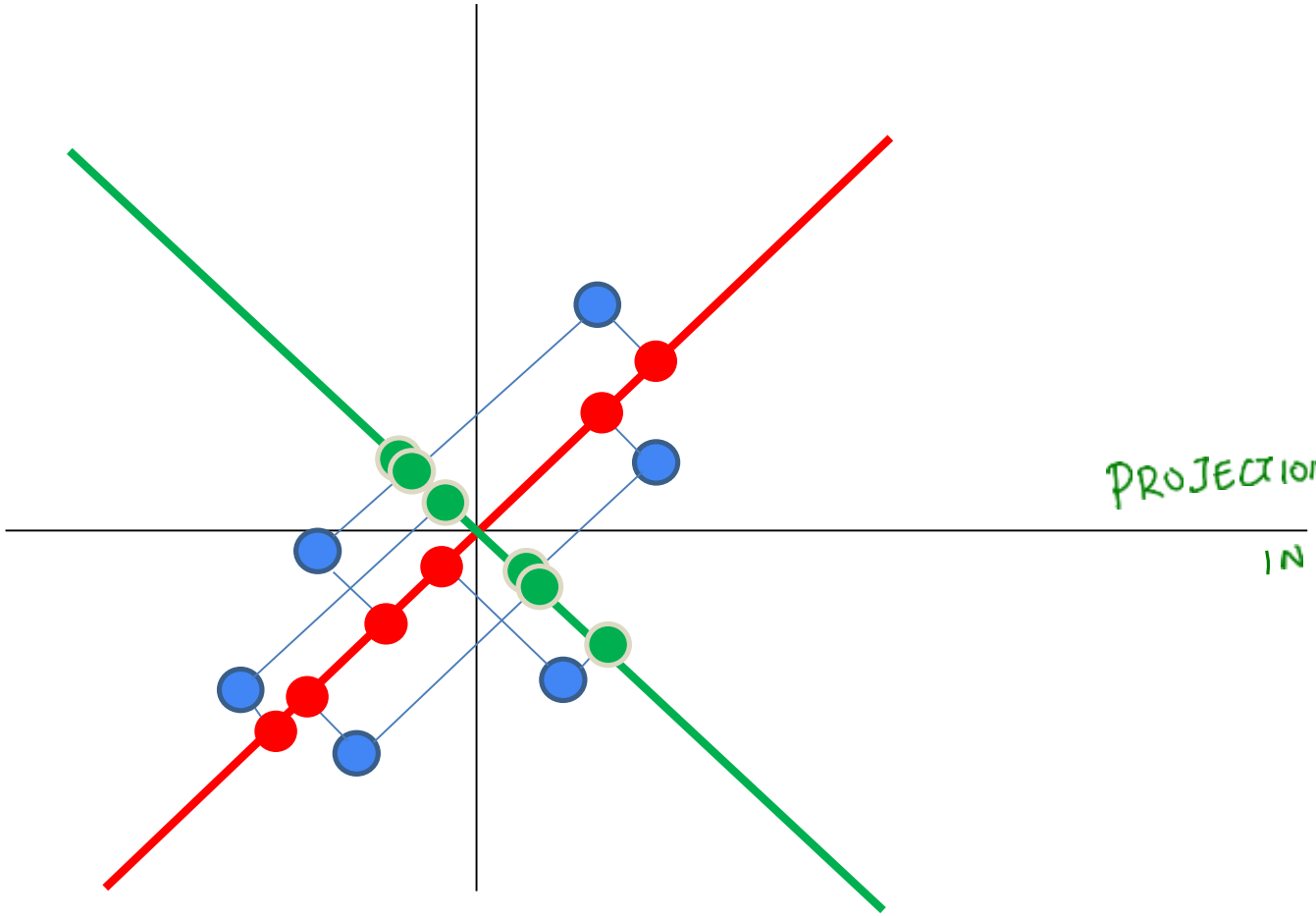
$$\frac{1}{n}\sum_{i=1}^{n}(x_i^T w) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^T w$$

$$0^T w = 0$$

Variance

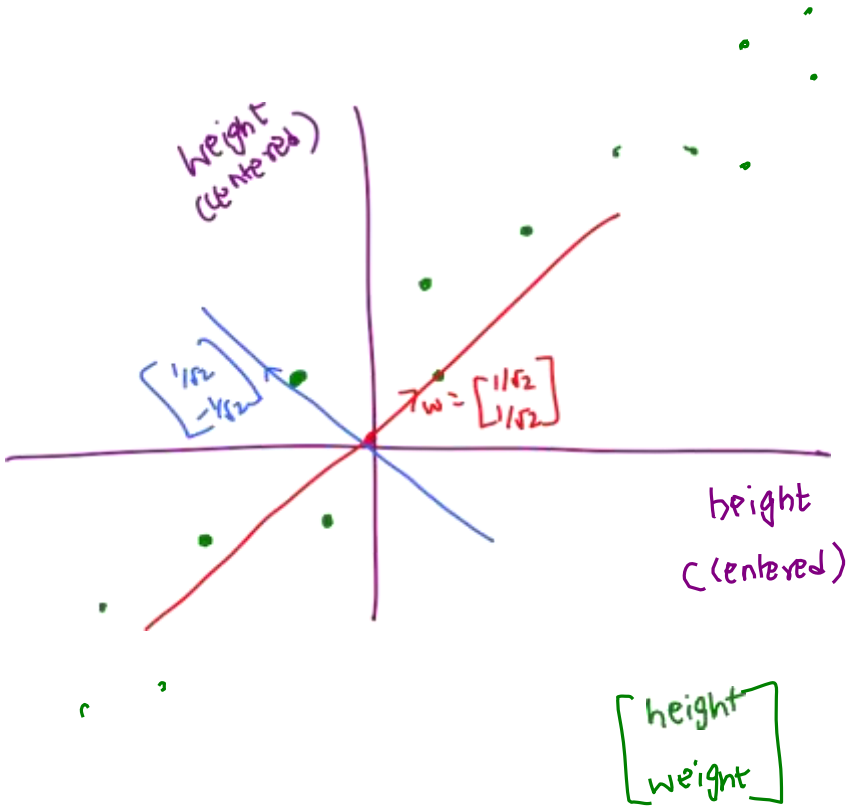
$$\frac{1}{n}\sum_{i=1}^{n}\left(x_i^T w - 0\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^T w\right)^2$$

ERROR MINIMIZATION ON CENTERED DATA = VARIANCE MAXIMIZATION

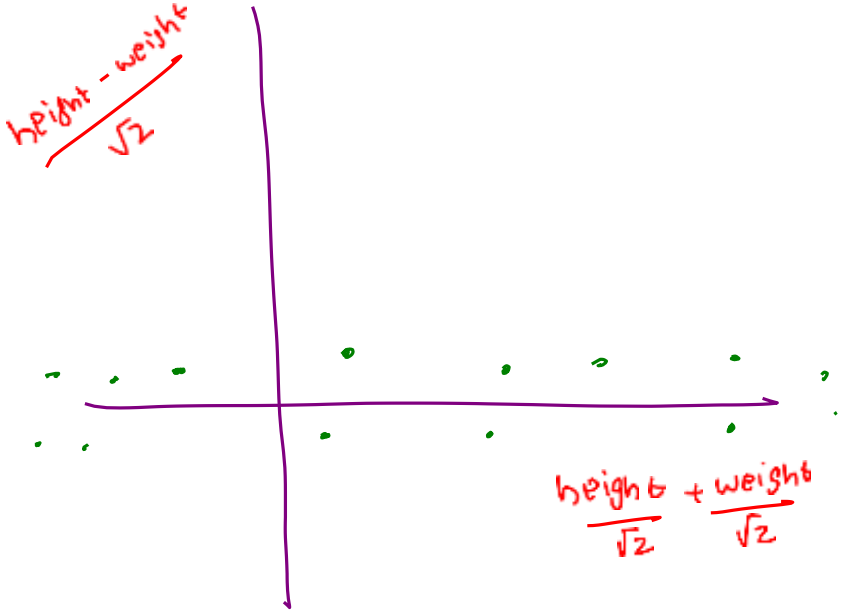


PROJECTIONS CROWD UP IN GREEN DIRECTION.

ONE MORE EXAMPLE



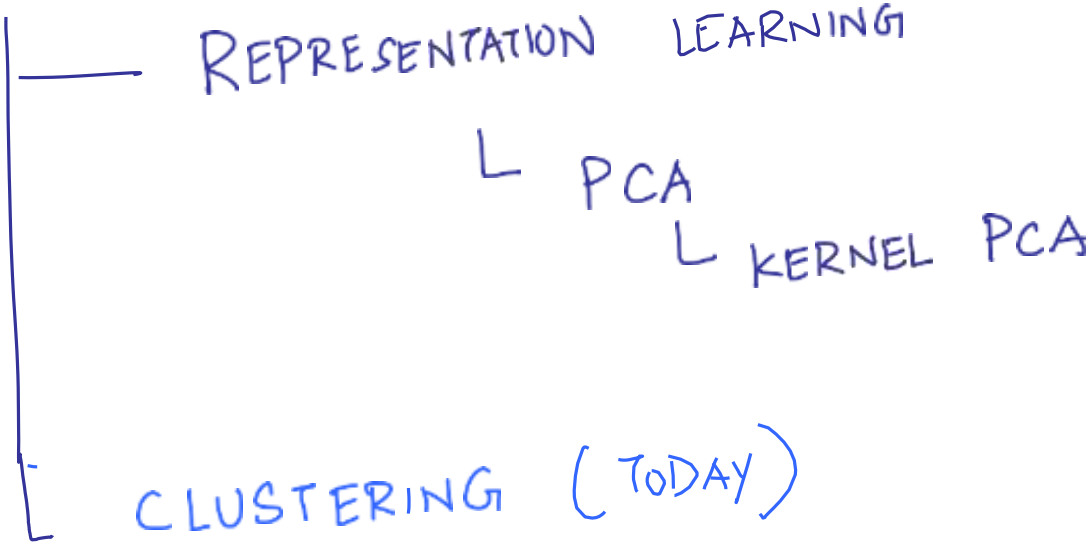
$\begin{bmatrix} \text{height} \\ \text{weight} \end{bmatrix}$



PCA finds combinations of features that are "decorrelated"

So far

unsupervised Learning



REAL WORLD EXAMPLE

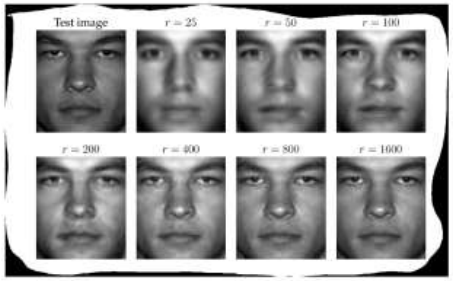
faculty.washington.edu/sbrunton/me565/pdf/L29secure.pdf

EXTENDED YALE FACE DATASET

~ 32000

EACH Datapoint $x_i \in \mathbb{R}^{\sim 32000}$

$d = \sim 32000$; $n = \sim 20000$



TEST IMAGE 1

TEST IMAGE 2

PRINCIPLE COMPONENTS

PC1 PC2 ... PCK

Summary of PCA

-> Given a dataset, we wanted to find a compressed representation.

-> In other words, we wanted to find a subspace that best “represents” the data

-> Best representation => Error is minimum ⇔ variance is maximum

-> The principal directions can be chosen greedily ! (In practice using SVD)

-> Extremely useful for real world data such as images.

PCA

Time complexity → Find Eigen vectors & Eigen values

$C \in \mathbb{R}^{d \times d}$

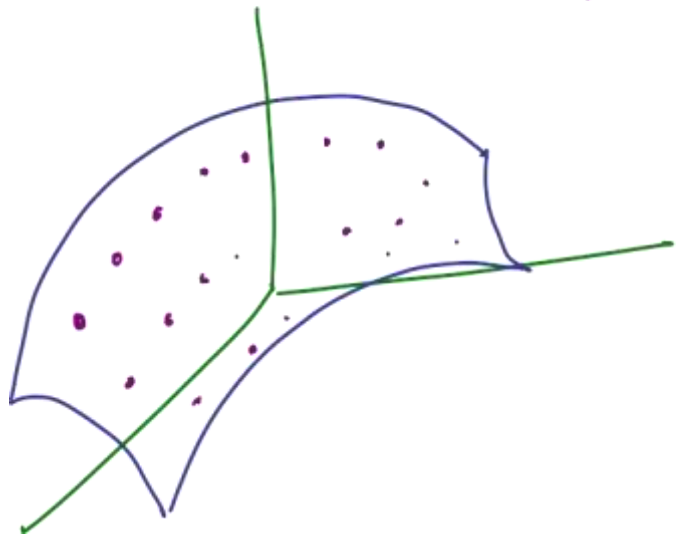
Typically $O(d^3)$

Issue 1: $d \gg n$ (Features: d; number of data points: n)

Example: Face recognition

ISSUE 2.

PCA finds "LINEAR" relationships only.



SURPRISING RESULT.

SAME SOLUTION FOR BOTH ISSUES!!

ISSUE 1: $d >> n$ $O(d^3)$ for Eigen decomposition

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

$$XX^T = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix}_{d \times n} \begin{bmatrix} — x_1 — \\ — x_2 — \\ \vdots \\ — x_n — \end{bmatrix}_{n \times d} = \sum_{i=1}^{n} x_i x_i^T$$

[Show this]

$$\Rightarrow \quad \boxed{C = \frac{1}{n} X X^T}$$

Let w_k be the eigenvector corresponding to k^{th} largest e.v of C (λ_k)

$$C \cdot w_k = \lambda_k w_k$$

$$\left(\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right) w_k = \lambda_k w_k$$

$$\Rightarrow \quad \boxed{w_k = \frac{1}{n\lambda_k} \sum_{i=1}^{n} (x_i^T w_k)\, x_i}$$

w_k is a LINEAR COMBINATION OF data points.

$$\begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_{k1} \\ \vdots \\ \alpha_{kn} \end{bmatrix} = \sum_{i=1}^{n} \alpha_{ki} \cdot x_i$$

$$w_k = \sum_{i=1}^{n} \left(\frac{x_i^T w_k}{n \lambda_k} \right) x_i$$

Assume for the moment we know α_k $\forall k$ [and not w_k].

Why is this useful?

$X_{test} \in \mathbb{R}^d \rightarrow \left[x_{test}^T w_1, \ldots, x_{test}^T w_k \right]$

$$x_{test}^T w_k = x_{test}^T \left(\sum_{i=1}^{n} \alpha_{ki} x_i \right) = \sum_{i=1}^{n} \alpha_{ki} \boxed{x_{test}^T x_i}$$

Can obtain compressed representation only by knowing alphas

Question: How to get α_k without computing w_k?

Some algebra

$$w_k = X\alpha_k \qquad * k$$

$$C w_k = \lambda_k w_k \quad \text{[by definition]}.$$

$$\left(\frac{1}{n} X X^T\right) w_k = \lambda_k w_k$$

$$\frac{1}{n} X X^T (X\alpha_k) = \lambda_k (X\alpha_k)$$

$$X X^T (X\alpha_k) = n\lambda_k \, X\alpha_k$$

Premultiply by X^T on both sides

$$X^T \left(X X^T X \alpha_k \right) = X^T \left(n\lambda_k \, X\alpha_k \right)$$

$$(X^T X)(X^T X)\alpha_k = n\lambda_k (X^T X)\alpha_k$$

$$(X^T X)^2 \alpha_k = n\lambda_k (X^T X)\alpha_k$$

Lets call $X^T X$ as K

Want α_k s.t $K^2 \alpha_k = n \lambda_k K \alpha_k$.

Observation:

$X\alpha_k = w_k$

$w_k^T w_k = 1$

$(X\alpha_k)^T (X\alpha_k) = 1$

$\alpha_k^T (X^T X) \alpha_k = 1$

$\alpha_k^T K \alpha_k = 1$

Summary so far

PCA

ORIGINAL GOAL : $\{x_1, \ldots, x_n\} \to XX^T \to \{w_1, \ldots, w_l\}$

$x_i \in \mathbb{R}^d$

nC

$n\lambda_1 \geq \ldots \geq n\lambda_l$

OBSERVATION 1 : $w_k = X\alpha_k$ for some $\alpha_k \in \mathbb{R}^n$

$d \times n$

OBSERVATION 2 : Suffices to find α_k satisfying

→ (P1) $K\alpha_k = (n\lambda_k)\alpha_k$ $\quad K = X^TX \in \mathbb{R}^{n \times n}$

(P2) $\alpha_k^T K \alpha_k = 1$

OBSERVATION 3:** $nC = \underset{d \times d}{XX^T} \rightarrow \{ w_1, \ldots, w_\ell \}$ $w_i \in \mathbb{R}^d$, $n\lambda_1 \geq \ldots n\lambda_\ell$, $\|w_k\|^2 = 1$

$k = \underset{n \times n}{X^T X} \rightarrow \{ \beta_1, \ldots, \beta_\ell \}$ $\beta_i \in \mathbb{R}^n$, $n\lambda_1 \geq \ldots n\lambda_\ell$, $\|\beta_k\|^2 = 1$

Thus α_k that satisfies (P1) and (P2) can be obtained as

$$\alpha_k := \frac{\beta_k}{\sqrt{n\lambda_k}}$$

$k \rightarrow \square \rightarrow \{\beta_1, \ldots, \beta_k\}$, $n\lambda_1, \ldots n\lambda_\ell$

[verify (P1) & (P2) using observations (1), (2) & (3)]

** - The non-zero Eigenvalues of XX^T and X^TX are the same – proved using SVD of X

PROCEDURE

Centered Dataset - $\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$ $[d >> n]$

- **Step 1:** Calculate $K = X^T X$
- **Step 2:** Eigendecompose K to get

$\{\beta_1, \ldots, \beta_\ell\}$ $\beta_k \in \mathbb{R}^n$ $\|\beta_k\|^2 = 1$

$\{n\lambda_1 \geq \ldots \geq n\lambda_\ell\}$

- **Step 3:** $\alpha_k = \frac{\beta_k}{\sqrt{n\lambda_k}}$ $\forall k$ $\alpha_k \in \mathbb{R}^n$

- **Step 4:** $w_k = X\alpha_k$ $w_k \in \mathbb{R}^d$

$\|w_k\|^2 = 1$?

Key advantages of this method

we need only $X^T X$.

i.e., only "pairwise dot products"

Step 2: Eigendecompose K to get

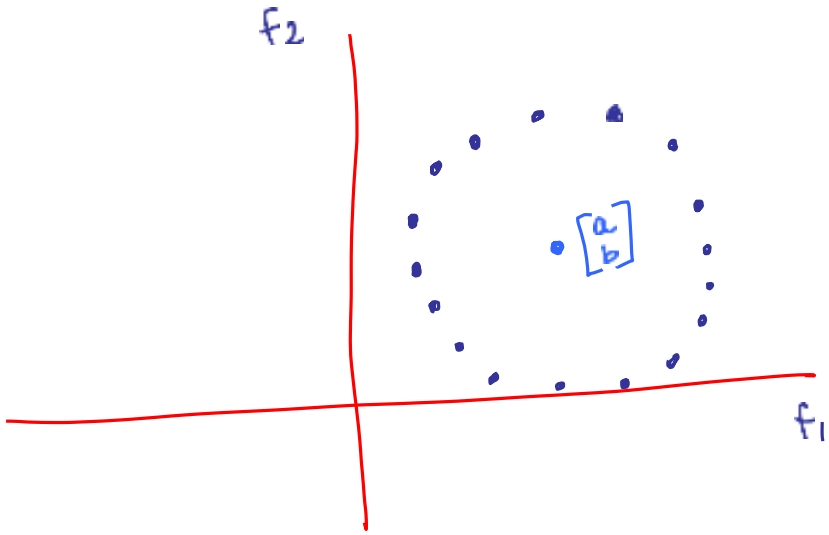
$\{\beta_1, \ldots, \beta_\ell\}$ $\quad \beta_k \in \mathbb{R}^n$, $\|\beta_k\|^2 =$

$\{n\lambda_1 \geq \ldots \geq n\lambda_\ell\}$

$O(n^3)$

$K \in \mathbb{R}^{n \times n}$

ISSUE 2 → Features could be non-linearly related



Standard PCA would pick 2 eigen directions

$$(f_1 - a)^2 + (f_2 - b)^2 = r^2$$

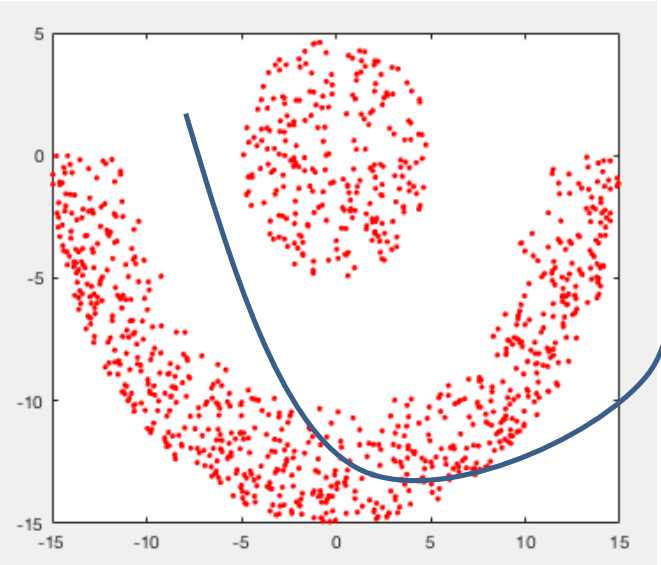
Idea: Transform features from low dimension $\mathbb{R}^d$ to high dimension $\mathbb{R}^D$

$x \rightarrow \phi(x)$
$\mathbb{R}^d \quad \mathbb{R}^D$

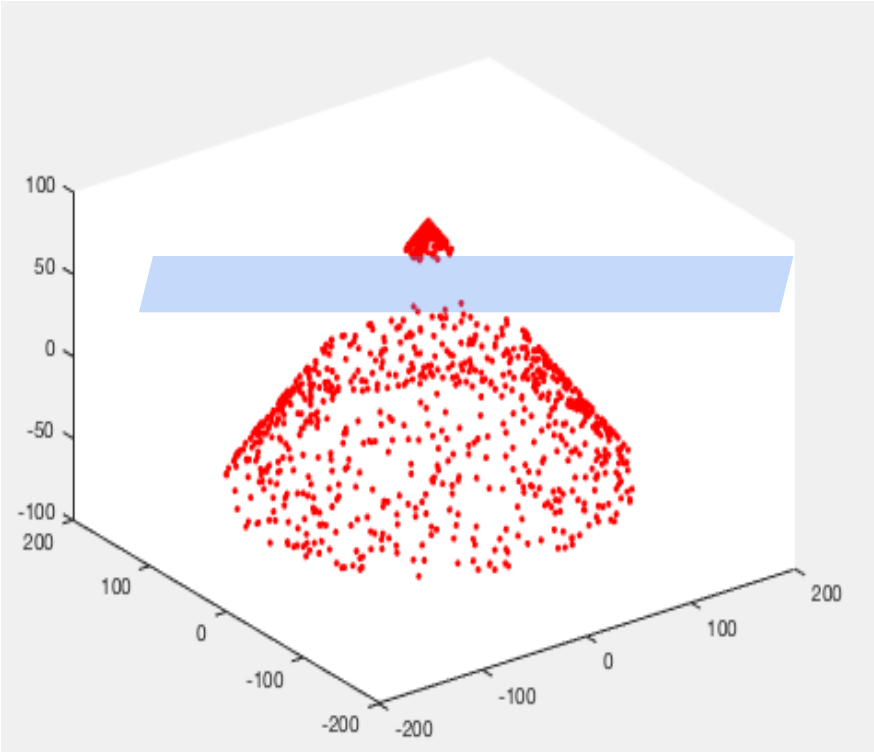
We already know how to handle Case when $D >> n$

[use $\phi(x)^T \phi(x)$ instead of $\phi(x) \phi(x)^T$]

ORIGINAL DATA



3D REPRESENTATION



Quadratic Map φ to 6 Dimensions

PCA

w1,w2,w3 Top 3 principal components

Reconstruct each point x as [ϕ(x)'w1 ϕ(x)'w2 ϕ(x)'w3]

KERNEL-PCA

Input: Dataset $\{x_1, \dots, x_n\}$ $\quad x_i \in \mathbb{R}^d$

Kernel k: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

Step 1: Compute $K \in \mathbb{R}^{n \times n}$

$$K_{ij} = k(x_i, x_j)$$

Step 2: Compute $\{\beta_1, \dots, \beta_\ell\}$ (e-vec of K)

$\{n\lambda_1, \dots n\lambda_\ell\}$ (e-v of K)

$$\Rightarrow \alpha_k := \frac{\beta_k}{\sqrt{n\lambda_k}}$$

✗ Step 3.

$$w_k = \phi(x)\,\alpha_k$$

→ "Defeats the purpose"

We cannot "reconstruct" eigenvectors of the covariance matrix

=

$$\phi(x_i)^T w_k \quad \Rightarrow \quad \phi(x_i)^T \left(\sum_{j=1}^{n} \alpha_{kj}\, \phi(x_j) \right)$$

$$\sum_{j=1}^{n} \alpha_{kj}\, k(i,j)$$

Typically good enough for "downstream" tasks.

What about Centering?

- Even if center the original data, it does not guarantee that the data will be centered after applying the transformation ϕ

- We want to work with the centered data w.r.t to $\{\phi(x_1), \phi(x_2), \ldots, \phi(x_n)\}$

- Let $\mu = 1/n \, \Sigma\phi(x_\ell)$

- The (i,j) term in the pairwise dot product matrix in the high dimension will

- $(\phi(x_i) - \mu)^T(\phi(x_j) - \mu) = \phi(x_i)^T\phi(x_j) - \phi(x_i)^T\mu - \phi(x_i)^T\mu + \mu^T\mu$

$$= \phi(x_i)^T\phi(x_j) - \phi(x_i)^T (1/n \, \Sigma\phi(x_\ell)) - \phi(x_i)^T (1/n \, \Sigma\phi(x_\ell)) + (1/n \, \Sigma\phi(x_\ell))^T (1/n \, \Sigma\phi(x_\ell))$$

$$= K(i,j) - 1/n\Sigma K(i, \ell) - 1/n\Sigma K(j, \ell) - 1/n^2\Sigma\Sigma K(\ell, \ell')$$

Good news - Each term depends only on K

KERNEL-PCA

Input: Dataset $\{x_1, \ldots, x_n\}$ $\quad x_i \in \mathbb{R}^d$

Kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

Step 1: Compute $K \in \mathbb{R}^{n \times n}$

$$K_{ij} = k(x_i, x_j)$$

Step 1.5 Center Kernel using ①

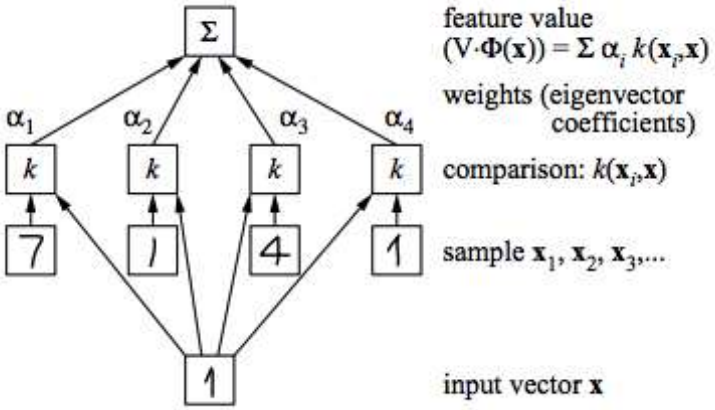
Step 2: Compute $\{\beta_1, \ldots, \beta_\ell\}$ e-vec of K $\quad \Rightarrow \quad \alpha_k := \frac{\beta_k}{\sqrt{n\lambda_k}}$

$\{n\lambda_1, \ldots, n\lambda_\ell\}$ e-v of K

$\phi(x_i)^T w_k$

$\phi(x_i)^T \left(\sum_{j=1}^{n} \alpha_{kj} \phi(x_j) \right)$

$\sum_{j=1}^{n} \alpha_{kj} k(i,j)$ ← Typically good enough for "downstream tasks"



Nonlinear Component Analysis as a Kernel Eigenvalue Problem

Bernhard Schölkopf
Max-Planck-Institut für biologische Kybernetik, 72076 Tübingen, Germany

Alexander Smola
Klaus-Robert Müller
GMD First (Forschungszentrum Informationstechnik), 12489 Berlin, Germany

Application of Kernel-PCA to face recognition

400 images of 40 subjects with different expression/poses.
Each image is (23*28) = 644 dimensional

- Apply Kernel PCA to obtain features
- Use features as input to a simple supervised learning algorithm (which we will see later)

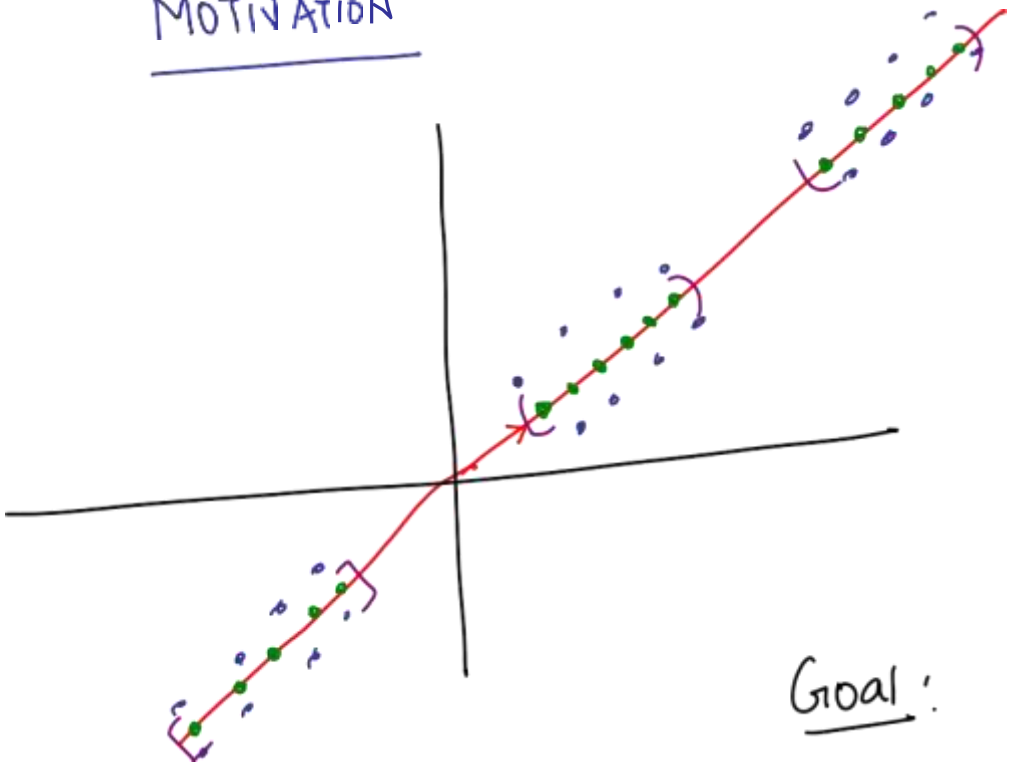
Results:

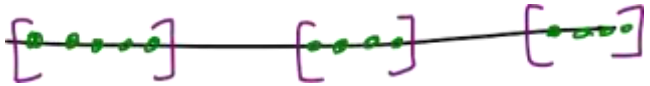
Method	Reduced space	Error Rate (%)
Eigenface	30	2.75
Kernel PCA, d=2	50	2.50
Kernel PCA, d=3	50	2.00
Kernel PCA, d=4	60	2.25
Kernel PCA, d=10	80	2.25

FACE RECOGNITION USING KERNEL EIGENFACES

Ming-Hsuan Yang Narendra Ahuja David Kriegman

MOTIVATION





Goal: Given $\{x_1, x_2, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$,

PARTITION the data into

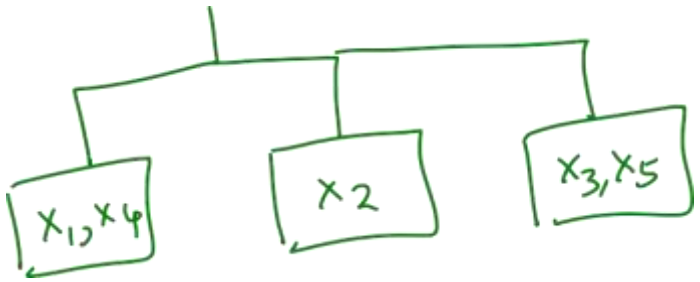
K CLUSTERS

EXAMPLE

$\{x_1, x_2, x_3, x_4, x_5\}$

Say $K=3$

one way



another way

3^5 possibilities

$x_1, x_2, \ldots, x_n$ ← DATA POINTS

$z_1, z_2, \ldots, z_n$ ← CLUSTER INDICATOR

$z_i \in \{1, \ldots, k\}$ $\forall i$

Given a cluster assignment, how good is it

$$F(z_1, \ldots, z_n) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2$$

$$\forall k \quad \mu_k = \frac{\sum_{i=1}^{n} x_i \, \mathbb{1}(z_i = k)}{\sum_{i=1}^{n} \mathbb{1}(z_i = k)}$$

$k = 1, \ldots k$

INDICATOR

$\mathbb{1}(p) = 1$ if p is true

$= 0$ p is false

Eg

x_1, x_2, x_3, x_4, x_5 $\quad k = 2$

$z_1 \; z_2 \; z_3 \; z_4 \; z_5$

$1 \; 2 \; 1 \; 1 \; 2$

$$\mu_1 = \frac{x_1 \cdot 1 + x_2 \cdot 0 + x_3 \cdot 1 + x_4 \cdot 1 + x_5 \cdot 0}{3}$$

Goal :

$$\min_{\{z_1, \cdots, z_n\}} \sum_{i=1}^{n} \| x_i - \mu_{z_i} \|^2$$

NP - HARD

LLYOD'S ALGORITHM

?? Initialization

$$z_1^0, z_2^0 \ldots, z_n^0 \quad \in \{1, \ldots, k\}$$

until Convergence.

$\forall k$ Mean of Cluster k at iteration t

$$\mu_k^t = \frac{\sum_{i=1}^{n} x_i \, \mathbb{1}(z_i^t = k)}{\sum_{i=1}^{n} \mathbb{1}(z_i^t = k)}$$

REASSIGNMENT STEP

$$z_i^{t+1} = \arg\min_k \| x_i - \mu_k^t \|^2$$

$$\in \{1, \ldots, k\}$$

FACT: Llyod's Algorithm converges!

QUESTIONS

- CONVERGENCE ?
- INITIALIZATION [sensitive to initialization]
- CHOICE OF K.
- NATURE OF CLUSTERS

LLYOD'S ALGORITHM CONVERGES

FACT -1.

Let $x_1, \dots, x_\ell \in \mathbb{R}^d$

$$v^* = \underset{v \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{\ell} \| x_i - v \|^2$$

$$f(v) = \sum_{i=1}^{\ell} \| x_i - v \|_2^2 \qquad (v \in \mathbb{R}^d)$$

Solution:

$$\boxed{v^* = \frac{1}{\ell} \sum_{i=1}^{\ell} x_i}$$

$$\nabla f(v) = 2 \sum_{i=1}^{n} (x_i - v)$$

[Exercise]

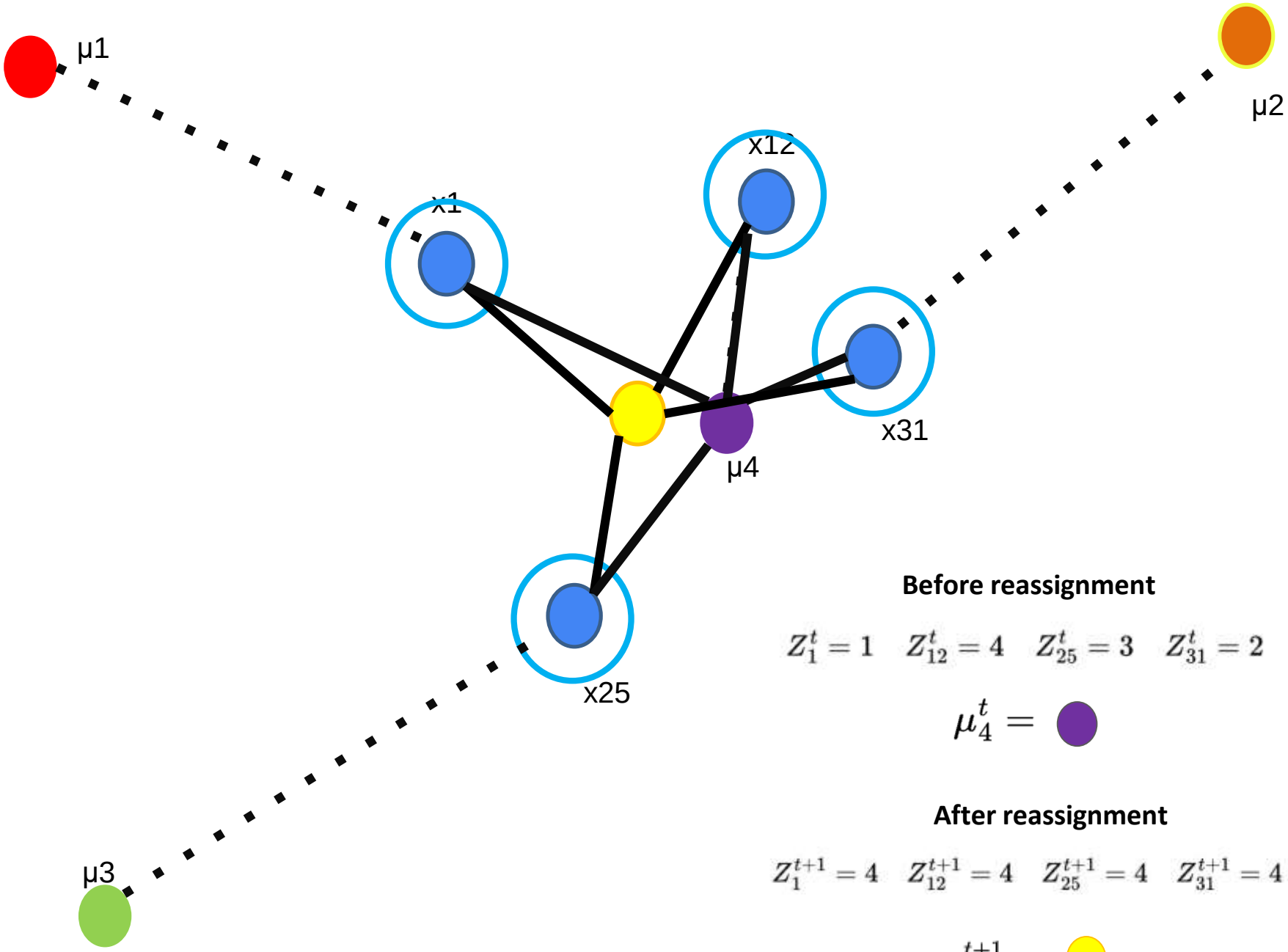
- Say we are at iteration t of Llyod's

- $z_1^t, z_2^t, \ldots, z_n^t \in \{1, \ldots, k\}$

CURRENT ASSIGNMENTS

- $z_1^{t+1}, z_2^{t+1}, \ldots, z_n^{t+1} \in \{1, \ldots, k\}$

What can we say about the update?



Before reassignment

$$Z_1^t = 1 \quad Z_{12}^t = 4 \quad Z_{25}^t = 3 \quad Z_{31}^t = 2$$

$$\mu_4^t =$$

After reassignment

$$Z_1^{t+1} = 4 \quad Z_{12}^{t+1} = 4 \quad Z_{25}^{t+1} = 4 \quad Z_{31}^{t+1} = 4$$

$$\mu_4^{t+1} =$$

By definition

$$\sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t+1}} \|^2 \quad < \quad \sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t}} \|^2$$

MEAN of Cluster where x_i wants to be

Mean of Current Cluster x_i is assigned to

$$\sum_{i=1}^{n} \| x_i - \mu^{t+1}_{z_i^{t+1}} \|^2 \quad < \quad \sum_{i=1}^{n} \| x_i - \mu^{t}_{z_i^{t+1}} \|^2$$

using (FACT 1)

If Algo does not converge in round t $\Rightarrow$

$$\sum_{i=1}^{n} \| x_i - M_{z_i^{t+1}}^{t+1} \| < \sum_{i=1}^{n} \| x_i - M_{z_i^t}^{t} \|^2$$

$$F(z_1^{t+1}, \ldots z_n^{t+1}) < F(z_1^t, \ldots, z_n^t)$$

$\Rightarrow$ convergence?

→ OBJECTIVE FUNCTION STRICTLY REDUCES IF RE-ASSIGNMENT HAPPENS

→ only "FINITE" possibilities (Partitions)

→ Algorithm must converge.

NATURE OF CLUSTERS

Say $K=2$

- Llyod's produces 2 clusters with means μ_1, μ_2

- What can we say about points assigned to Cluster 1?

Where are

Clusters?

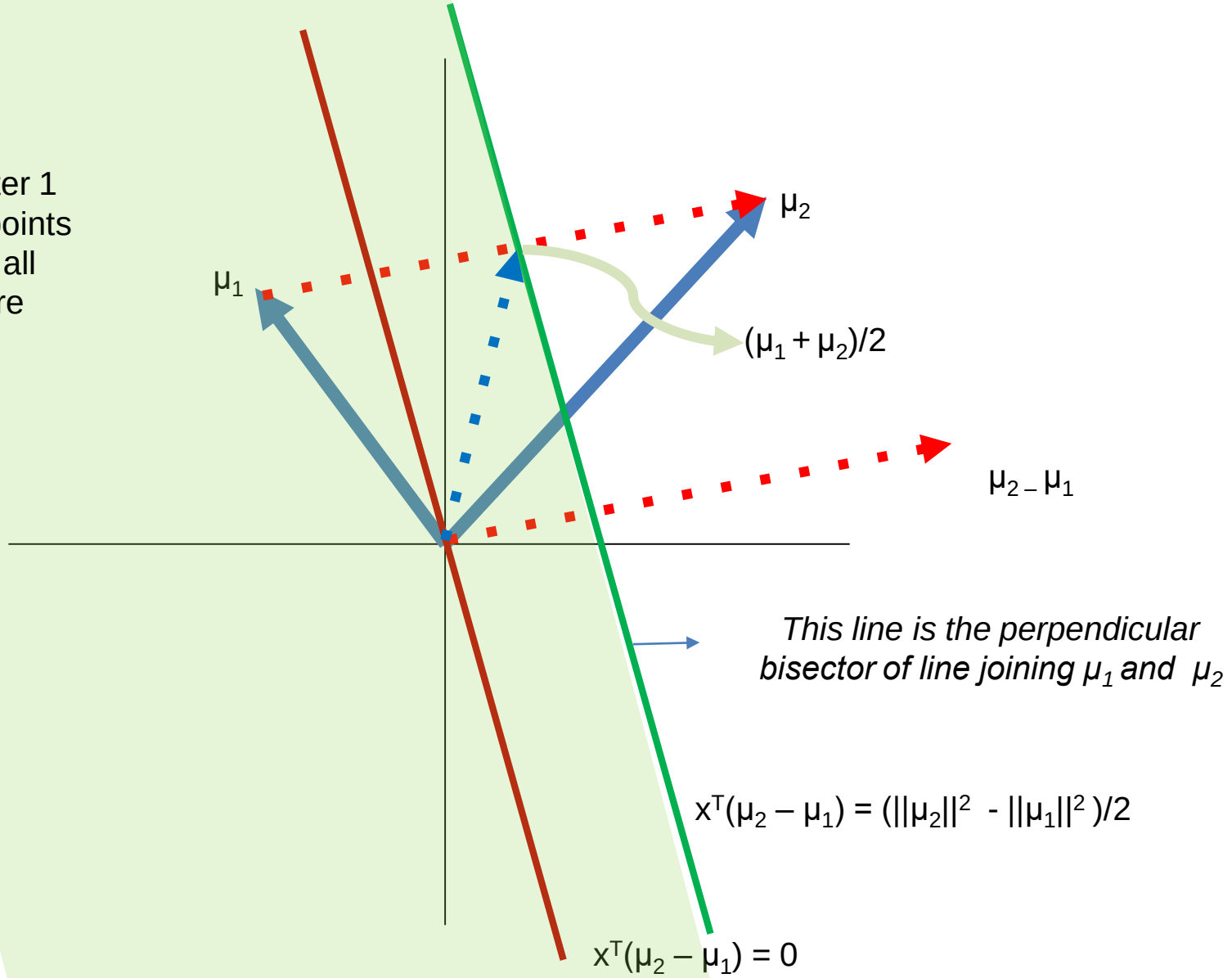
Cluster 1.

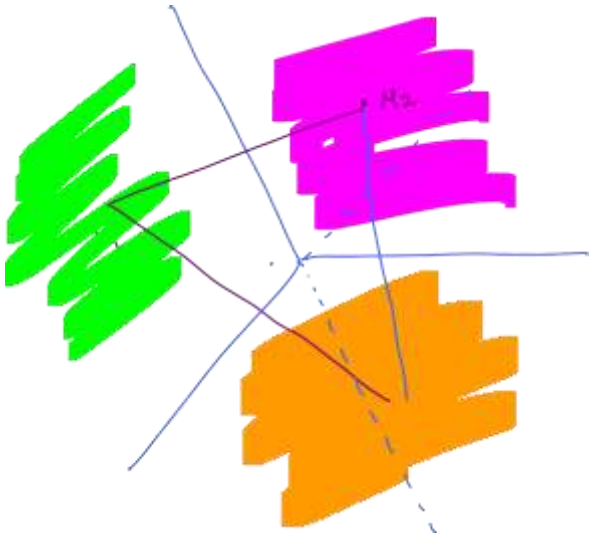
$$\|x - \mu_1\|^2 \leq \|x - \mu_2\|^2$$

$$\|x\|^2 + \|\mu_1\|^2 - 2x^T\mu_1 \leq \|x\|^2 + \|\mu_2\|^2 - 2x^T\mu_2$$

$$x^T(\mu_2 - \mu_1) \leq \left(\|\mu_2\|^2 - \|\mu_1\|^2\right)/2$$

$\underline{x}^T(\mu_2 - \mu_1) \leq (\|\mu_2\|^2 - \|\mu_1\|^2)/2$





N /CELLS

So far

- CONVERGENCE ?
- INITIALIZATION
- CHOICE OF K.
- NATURE OF CLUSTERS

INITIALIZATION

→ RANDOM PARTITION.

→ uniformly Sample K-means from dataset

K-MEANS ++

→ Choose first mean μ_1^0 (Iteration) uniformly at random from $\{x_1, \ldots x_n\}$

→ for $\ell = 2, \ldots, k$

Choose μ_ℓ^0 "probabilistically"

according to score $S(x) = \min_{j=1,\ldots,\ell-1} \| x - \mu_j^0 \|_2^2$

GUARANTEE

$$\mathbb{E}\left[\sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2\right] \leq O(\log k)\left(\min_{z_1,\ldots,z_n} \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2\right)$$

over randomness
over algorithm's choices

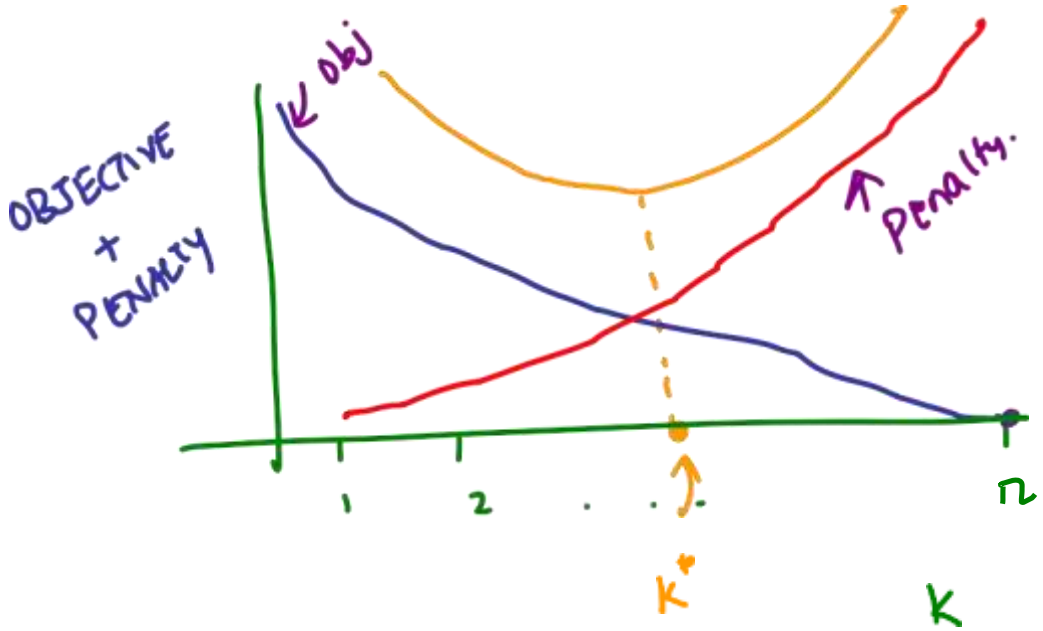
Choice of k ?

$F(z_1, \ldots, z_n)$

Penalize large k.

Find the k that has the

Smallest objective value + Penalty(k)



AIC - Akaike Information Criterion

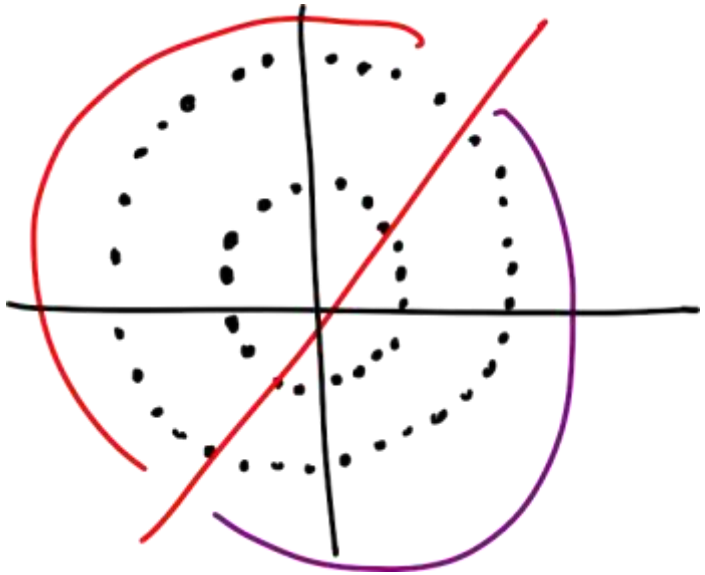
$$2K - 2\log(L(\theta^*))$$

Likelihood

BIC - Bayesian Information Criterion.

$$K\log(n) - 2\log(L(\theta^*))$$

Likelihood



KERNELIZE K-MEANS

ORIGINAL

$$\min_{z_1,\dots,z_n} F(z_1,\dots,z_n) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|^2$$

NEW

$$\min_{z_1,\dots z_n} \sum_{i=1}^{n} \| \phi(x_i) - \mu_{z_i} \|_2^2$$

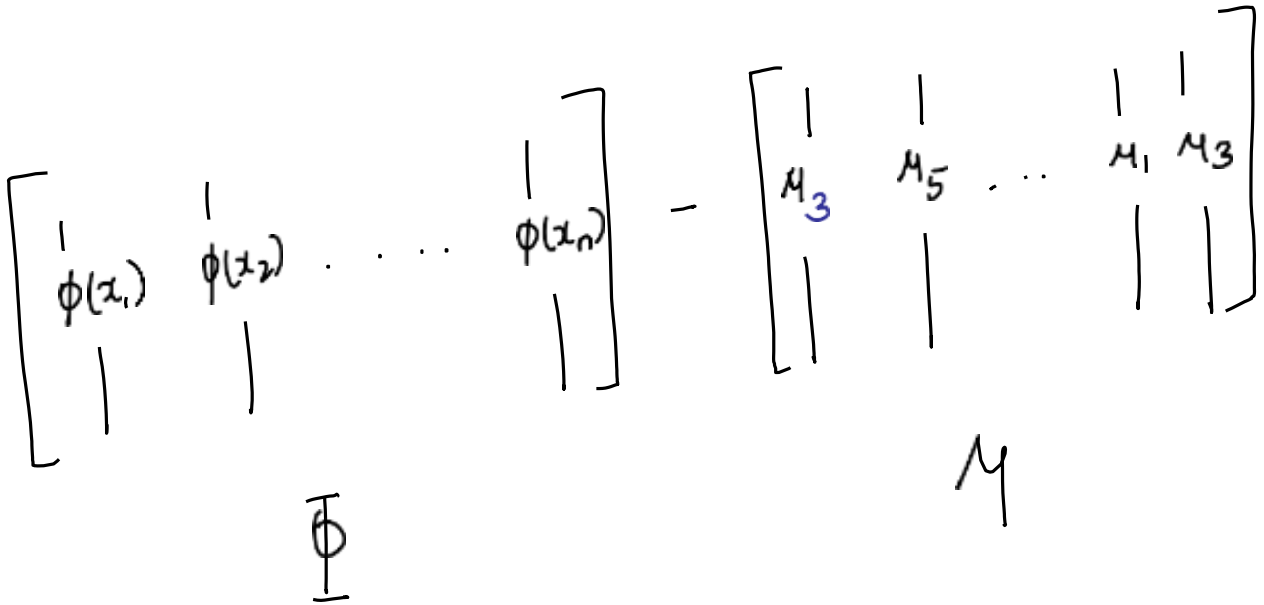
↓

NOT "KERNELIZED" YET!

$$\forall k \quad \mu_k = \sum_{i=1}^{n} \phi(x_i)\, \mathbb{1}(z_i = k) \Big/ \sum_{i=1}^{n} \mathbb{1}(z_i = k)$$

Goal: Can we rewrite the new objective using dot-products.

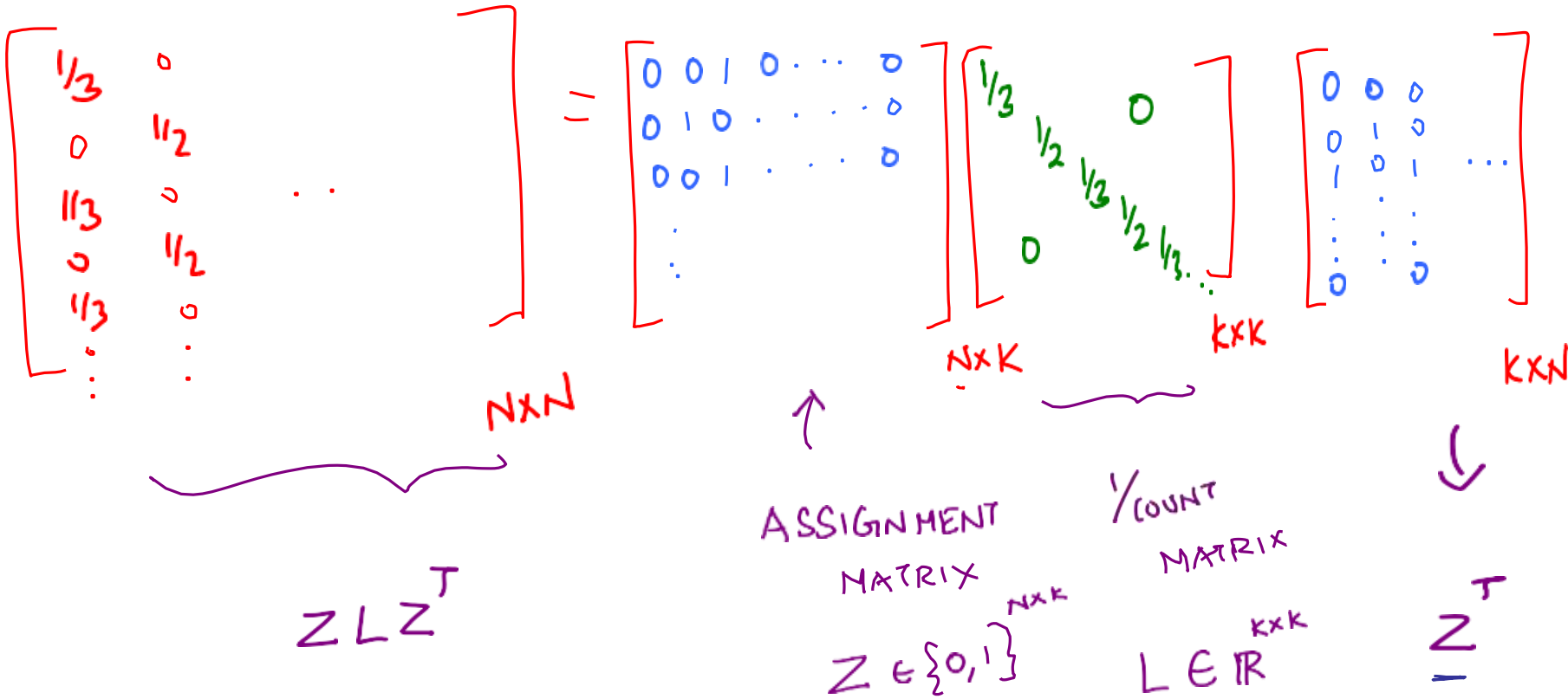
Consider



$$M = \begin{bmatrix} | & | & & | & | \\ x_3 & x_5 & \cdots & x_1 & x_3 \\ | & | & & | & | \end{bmatrix}_{D \times N} = \begin{bmatrix} | & | & | \\ \frac{\phi(x_1)+\phi(x_3)+\phi(x_5)}{3} & \frac{\phi(x_2)+\phi(x_4)}{2} & \cdots \\ | & | & | \end{bmatrix}_{D \times N}$$

$$= \begin{bmatrix} | & | & & | \\ \phi(x_1) & \phi(x_2) & \cdots & \phi(x_N) \\ | & | & & | \end{bmatrix}_{D \times N} \begin{bmatrix} 1/3 & 0 & 1/3 & \cdots \\ 0 & 1/2 & 0 & \\ 1/3 & 0 & 1/3 & \\ 0 & 1/2 & 0 & \\ 1/3 & 0 & 1/3 & \\ 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \end{bmatrix}_{N \times N}$$

$$= \Phi$$



$$\underline{\Phi} - \underline{M} = \underline{\Phi} - \underline{\Phi} Z L Z^T$$

$\underline{M} \hookrightarrow D \times N$

Want to consider $\sum_{i=1}^{n} \|\phi(x_i) - M_{z_i}\|^2$

$$= \text{trace}\left[(\phi - M)^T(\phi - M)\right]$$

$$= \text{trace}\left[(\phi - \phi Z L Z^T)^T(\phi - \phi Z L Z^T)\right]$$

$$= \text{trace}\left[\phi^T\phi - \phi^T\phi Z L Z^T - Z L Z^T \phi^T\phi + \underbrace{Z L Z^T}_{A}\ \underbrace{\phi^T\phi}_{B}\ \underbrace{Z L Z^T}_{C}\right]$$

$$A = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix}$$

$$A^T A = \begin{bmatrix} - a_1 - \\ - a_2 - \\ \vdots \\ - a_n - \end{bmatrix} \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix}$$

$$= \begin{bmatrix} \|a_1\|^2 & & & \\ & \|a_2\|^2 & & \\ & & \ddots & \\ & & & \|a_n\|^2 \end{bmatrix}$$

$$\text{trace}(A^T A) = \sum_{i=1}^{n} \|a_i\|_2$$

Facts about trace

- $\text{trace}(A+B) = \text{trace}(A) \quad \text{trace}(B)$
- $\text{tr}(\cdot \ \cdot C) = (C \cdot \ \cdot) = \text{tr}(\cdot \ \cdot A).$

$$\left(\| \quad - \quad z \|\ \| - \phi \| \quad z \qquad \sim^{L} \quad |^{T}\phi \qquad \cdot \right)$$

L

$$\left(\phi | \quad - \quad {}^{T}| \quad z \right)$$

Facts about trace

- $\text{trace}(A+B) = \text{trace}(A) + \text{trace}(B)$
- $\text{trace}(A \cdot B \cdot C) = \text{trace}(C \cdot A \cdot B) = \text{trace}(B \cdot C \cdot A)$.

$$\text{trace}\Big(\phi^T\phi - ZLZ^T\phi^T\phi - \phi^T\phi ZLZ^T + ZLZ^T ZLZ^T \phi^T\phi \Big)$$

($Z^TZ = L^{-1}$, verify)

$$ZLZ^T\phi^T\phi$$

$$\text{trace}\Big(\phi^T\phi - \phi^T\phi ZLZ^T \Big)$$

Goal:

$$\min \; \text{trace}\left(\phi^T\phi - \phi^T\phi\, Z L Z^T\right)$$

Z s.t it is a valid "assignment" matrix

$\equiv$

$$\max \; \text{trace}\left(\phi^T\phi\, Z L Z^T\right)$$

Z s.t Z is valid assng matrix

$\phi^T\phi$ → kernel.

$$\text{trace}\left(K Z L Z^T\right)$$

$$= \max_{Z} \operatorname{trace}\left(\underbrace{K}_{A} \underbrace{Z L^{1/2}}_{B} \underbrace{L^{1/2} Z^{T}}_{C} \right)$$

$$= \max_{Z} \operatorname{trace}\left(\boxed{(Z L^{1/2})^{T}} K (Z L^{1/2}) \right)$$

Z is a valid assignment matrix.

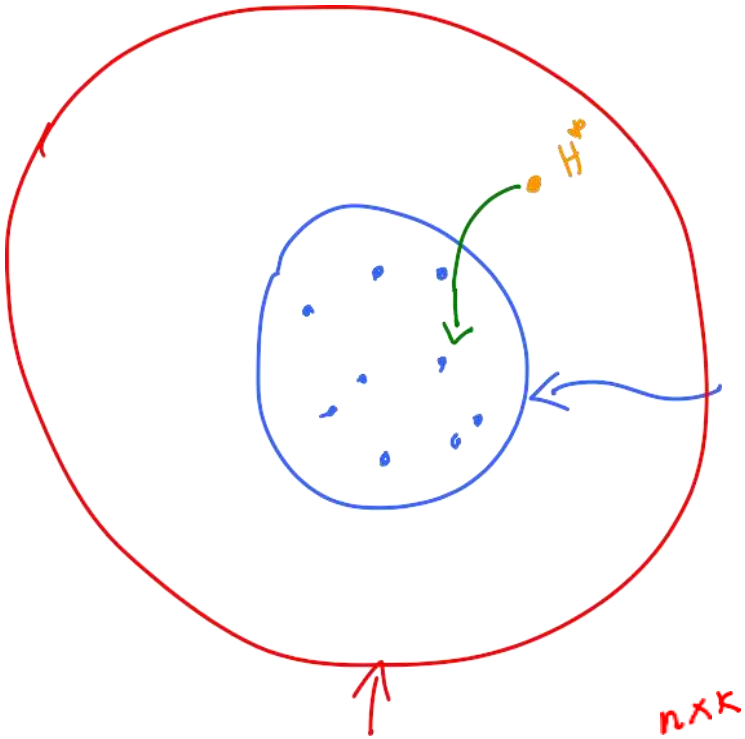
↳ NP-HARD

Notice that

$$(Z L^{1/2})^{T} (Z L^{1/2}) = L^{1/2} Z^{T} Z L^{1/2}$$

$$= L^{1/2} L^{-1} L^{1/2} = I$$

Can we solve a relaxed version?



valid assignments

$Z \in \{0,1\}^{n \times k}$ s.t

$ZL^{1/2}$ satisfies

$(ZL^{1/2})^T ZL^{1/2} = I$

Set of all $H \in \mathbb{R}^{n \times k}$

s.t $H^T H = I$

RELAXATION

$$\max_{H \in \mathbb{R}^{n \times K}} \operatorname{trace}\left(H^T K H\right)$$

$$\text{s.t} \quad H^T H = I$$

Solution: Top K ($K \in \mathbb{N}$) eigen vectors of K

kernel.

SPECTRAL CLUSTERING

→ Given $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$, kernel k

→ Compute $H^* = \begin{bmatrix} | & & | \\ h_1 & \cdots & h_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{n \times k}$

↳ top k eigen vectors of K

→ Normalize rows of H^*

→ Run Lloyd's assuming each row is "new" data.

Why Normalize?

Observation:

-> H is a proxy for $ZL^{1/2}$

-> We need a way to go from H to Z

-> if we were lucky, H = $ZL^{1/2}$ for some Z.

-> How can we go from H to Z in that case?

Example:

Z =

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

L =

$$\begin{pmatrix} 0.5000 & 0 & 0 & 0 \\ 0 & 0.5000 & 0 & 0 \\ 0 & 0 & 0.5000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

$ZL^{1/2} =$

$$\begin{pmatrix} 0 & 0.7071 & 0 & 0 \\ 0.7071 & 0 & 0 & 0 \\ 0 & 0.7071 & 0 & 0 \\ 0 & 0 & 0.7071 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0.7071 & 0 & 0 & 0 \\ 0 & 0 & 0.7071 & 0 \end{pmatrix}$$

After row normalization

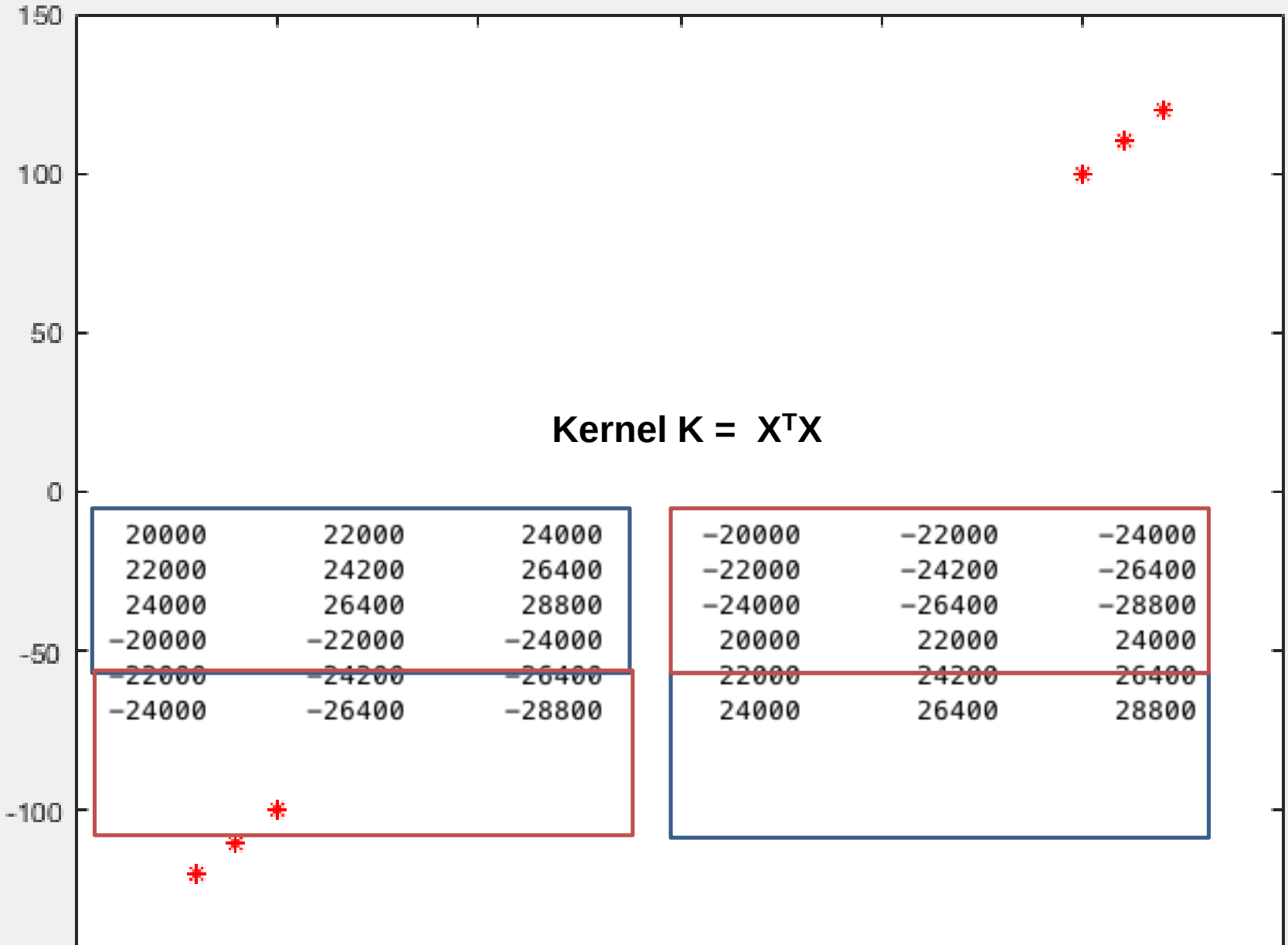
$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- In
ther

- H

- T

- N


150
100
50
0
-50
-100
Kernel K = X^T X
20000 22000 24000
22000 24200 26400
24000 26400 28800
-20000 -22000 -24000
-22000 -24200 -26400
-24000 -26400 -28800
-20000 -22000 -24000
-22000 -24200 -26400
-24000 -26400 -28800
20000 22000 24000
22000 24200 26400
24000 26400 28800

Observations:

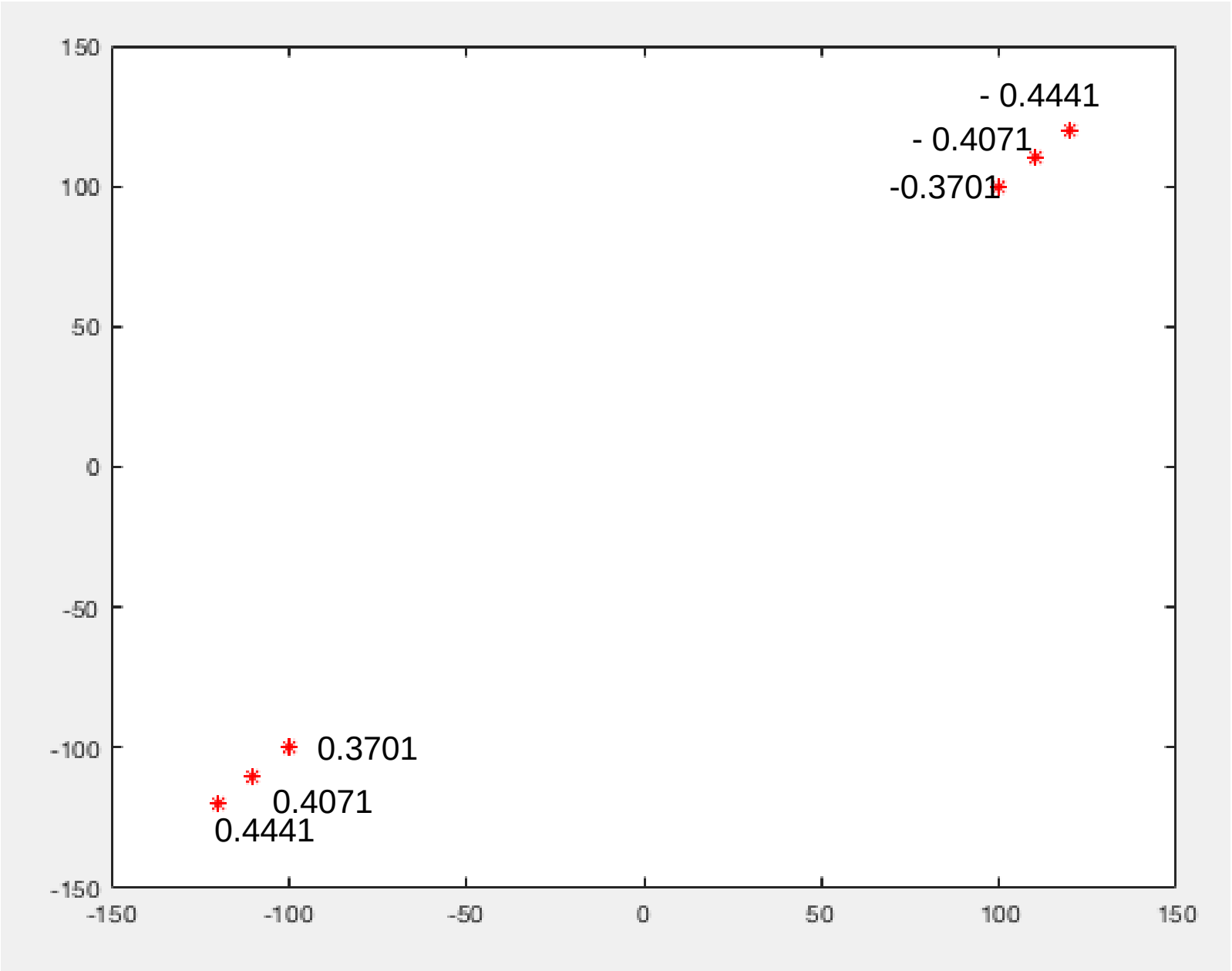
- K is a rank 1 matrix; K is real and symmetric

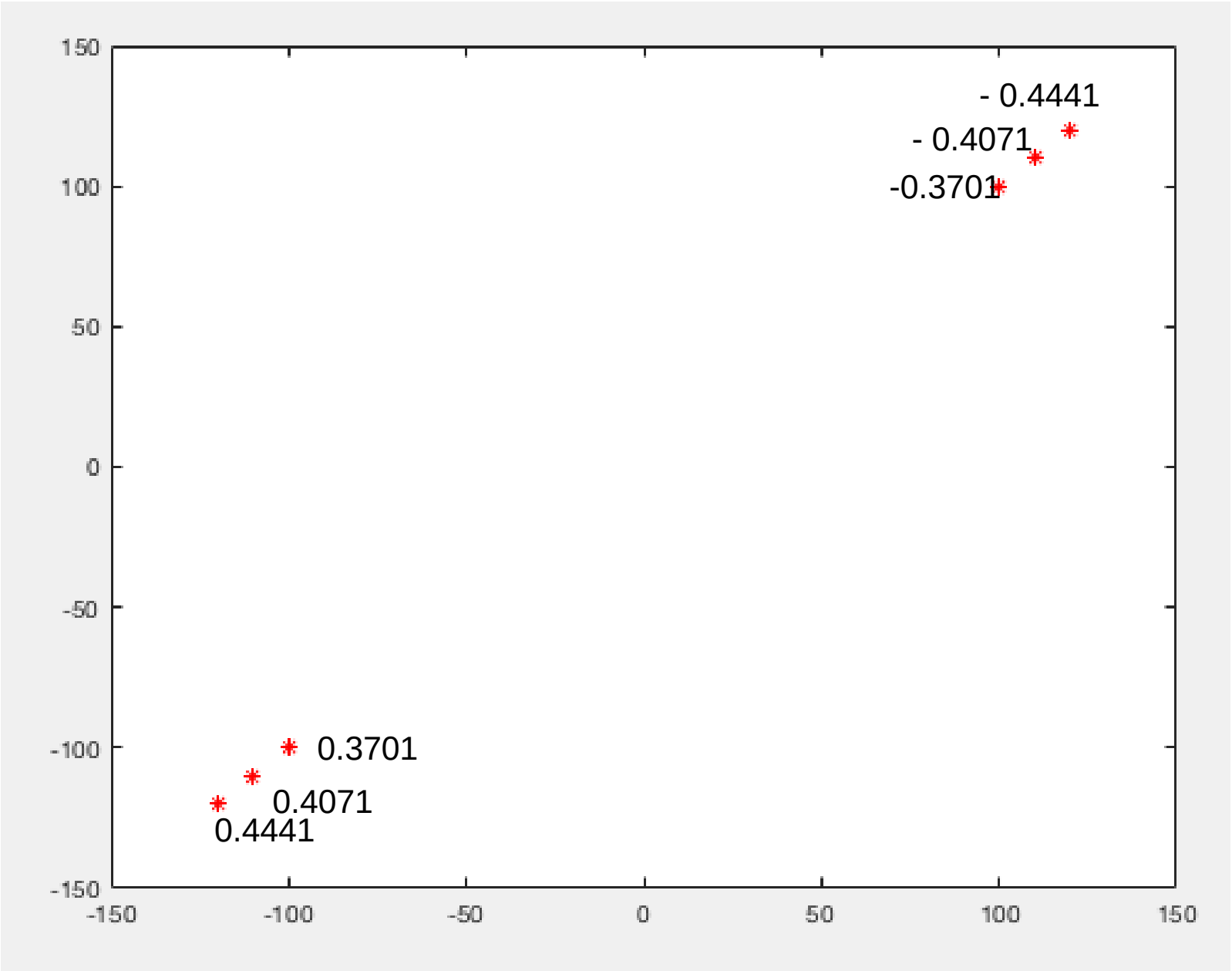
$\Rightarrow K = \lambda(\alpha\alpha^T)$ for positive real number λ and vector α

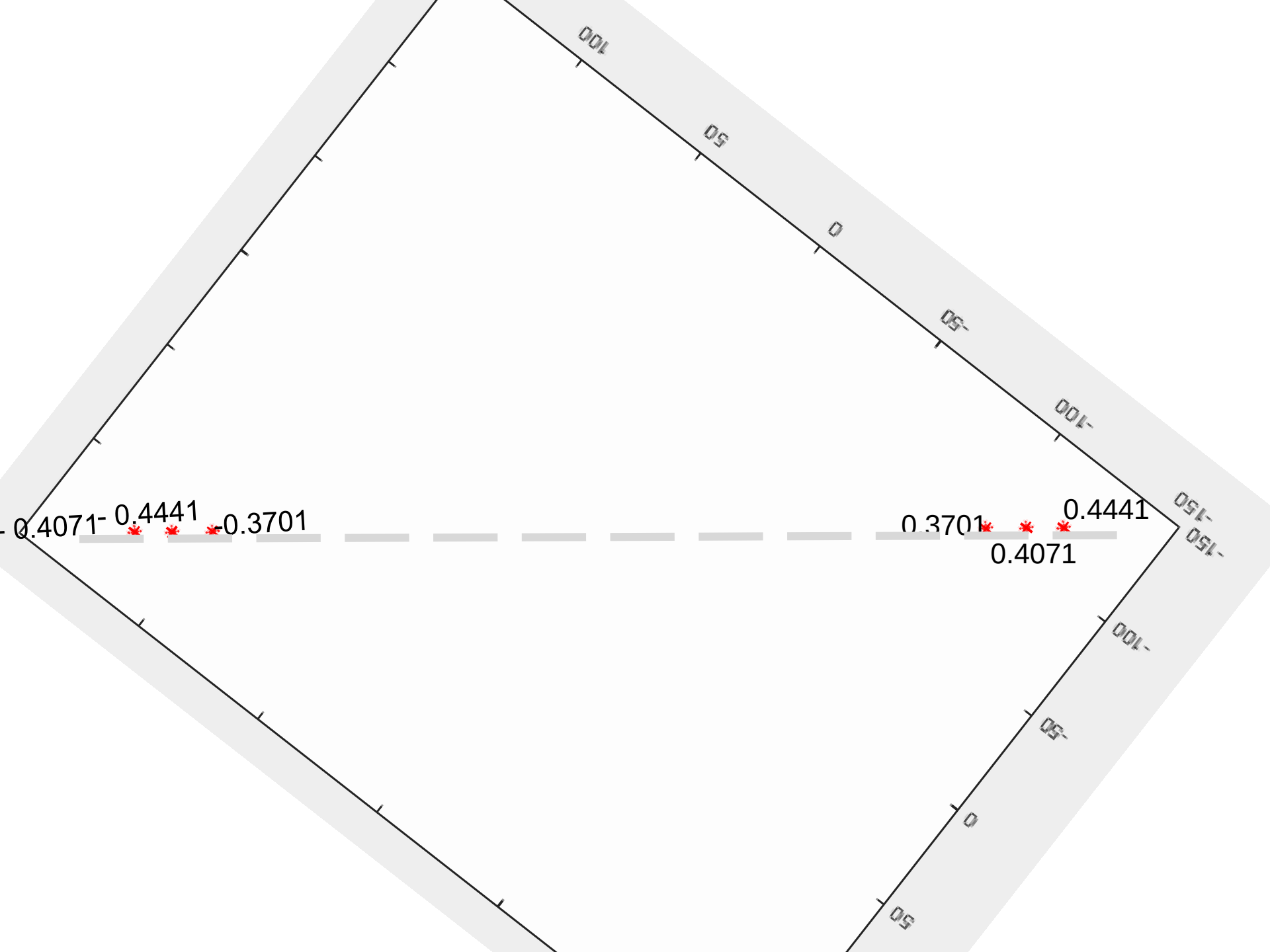
$\Rightarrow \lambda$ is the Eigen value and α is the corresponding Eigen ve

Thus, for all (i,j)

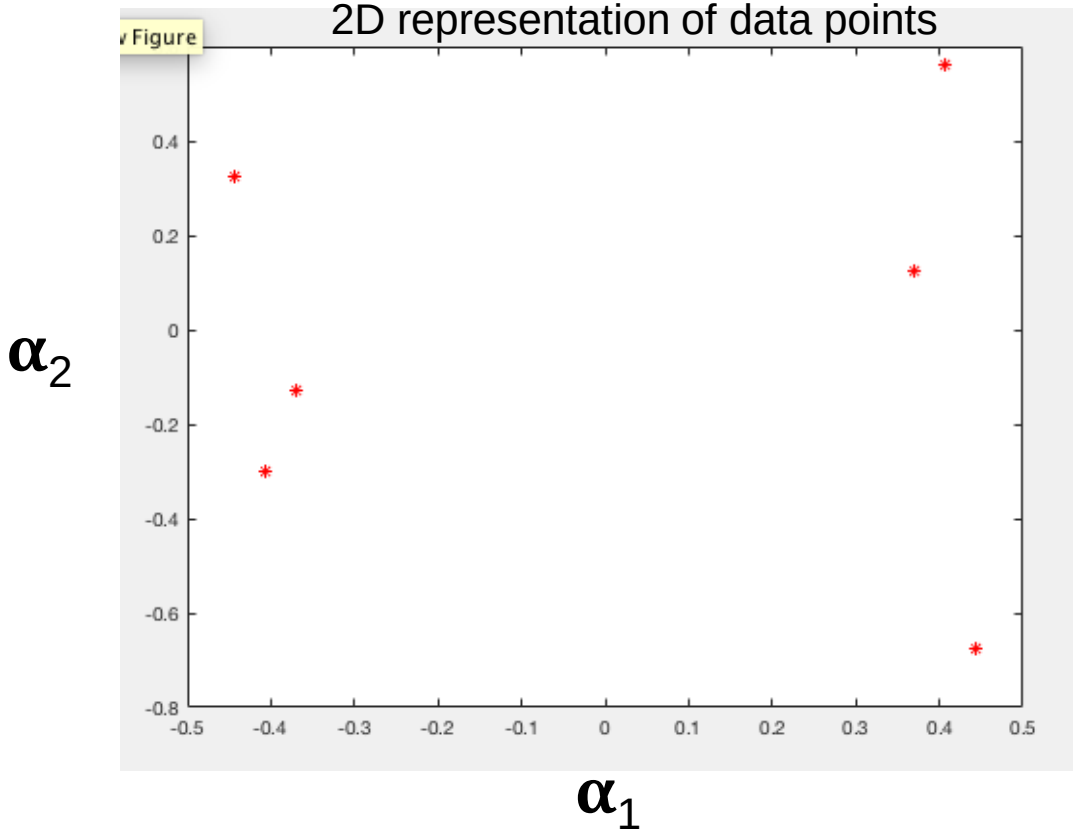
$$K_{ij} = \lambda \alpha_i \alpha_j$$


150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
- 0.4441
- 0.4071
-0.3701
0.3701
0.4071
0.4441

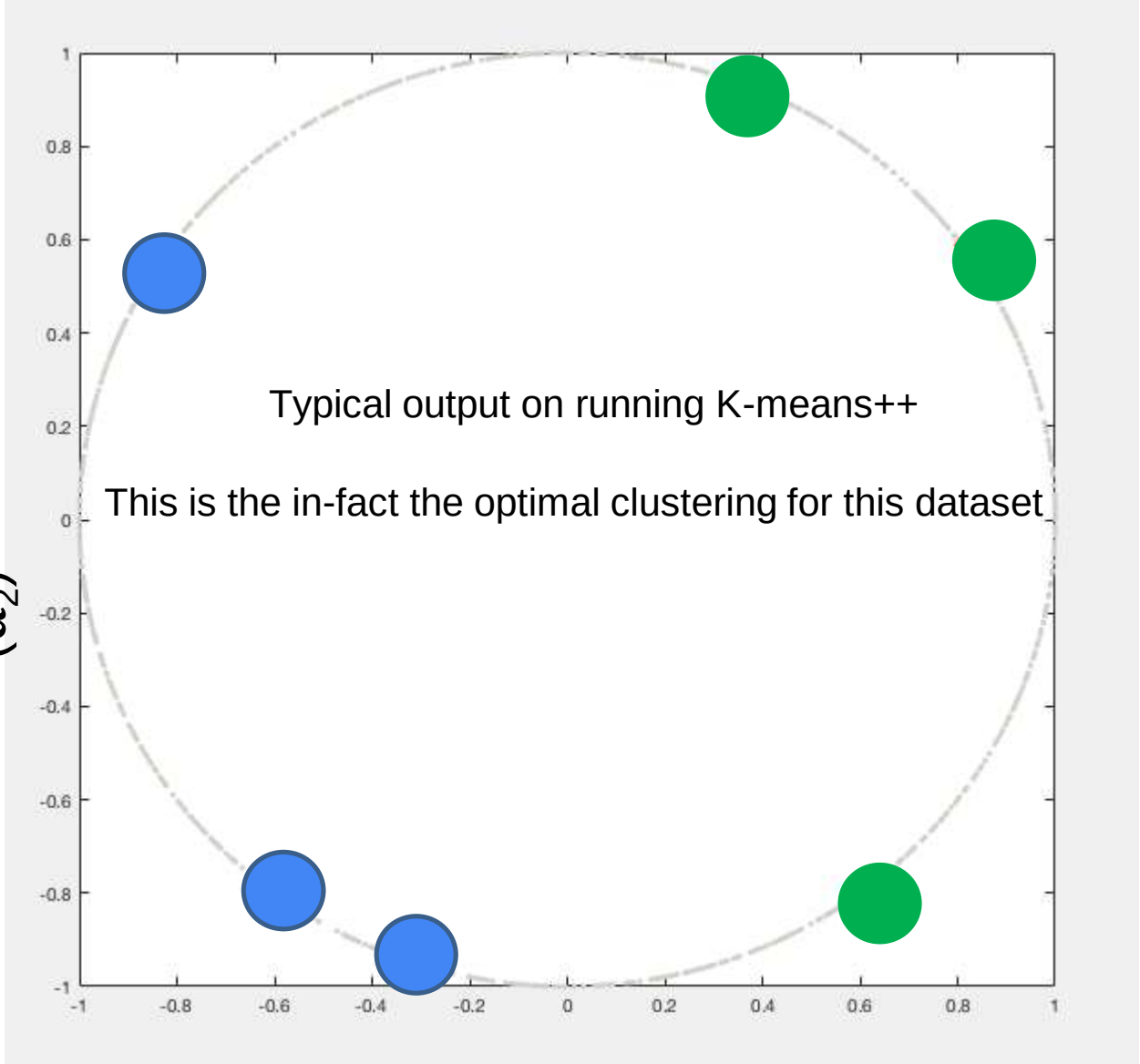

150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
- 0.4441
- 0.4071
-0.3701
0.3701
0.4071
0.4441


100
50
0
-50
-100
-150
-0.4071
-0.4441
-0.3701
0.3701
0.4441
0.4071

α_6				α_2	α_1
-0.1865	0.4087	0.7858	-0.1648	-0.1290	-0.3701
-0.5742	-0.5908	-0.1177	-0.2281	-0.2992	-0.4071
0.5929	-0.2236	-0.0124	-0.5426	0.3265	-0.4441
0.1441	-0.6512	0.6067	0.1849	0.1261	0.3701
-0.4648	0.0938	0.0112	-0.5421	0.5616	0.4071
0.2171	0.0320	0.0187	-0.5462	-0.6752	0.4441



2D normalized representation of data points



Last point about Spectral Clustering

- If you search for spectral clustering on Google, you may find a graph-based algorithm

- Though **not covered in this course**, you are encouraged to read about it. In fact there are many variants that are called spectral clustering.

- If you think for a bit, you will realize it is nothing but kernel-K-means that we have seen with a specific choice of kernel (pseudo inverse of the Laplacian matrix associated with data)

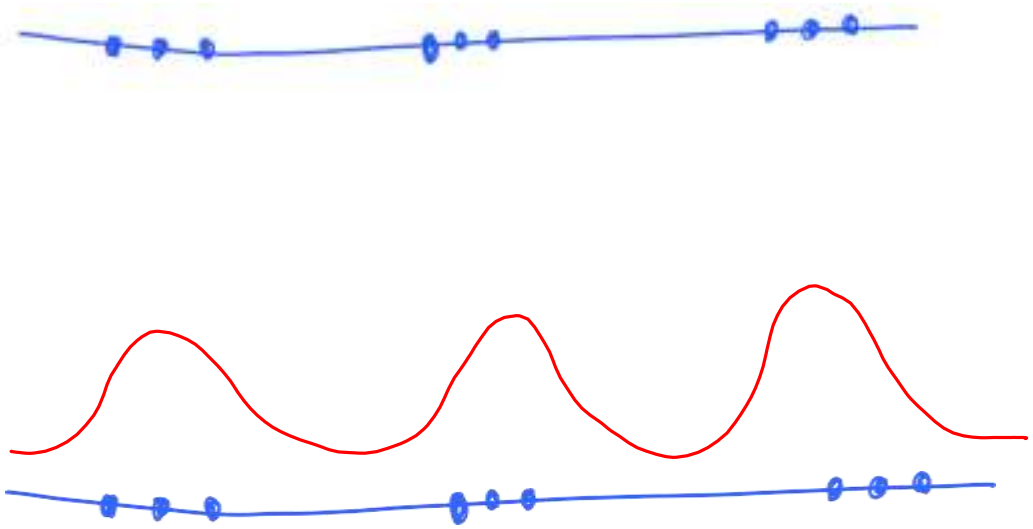
On Spectral Clustering: Analysis and an algorithm

Andrew Y. Ng
CS Division
U.C. Berkeley
ang@cs.berkeley.edu

Michael I. Jordan
CS Div. & Dept. of Stat.
U.C. Berkeley
jordan@cs.berkeley.edu

Yair Weiss
School of CS & Engr.
The Hebrew Univ.
yweiss@cs.huji.ac.il

Data



$x_i \in \mathbb{R}$

A new GENERATIVE STORY – MIXTURE OF GAUSSIANS

STEP 1: Pick which mixture a data point comes from

STEP 2: Generate a datapoint from that mixture

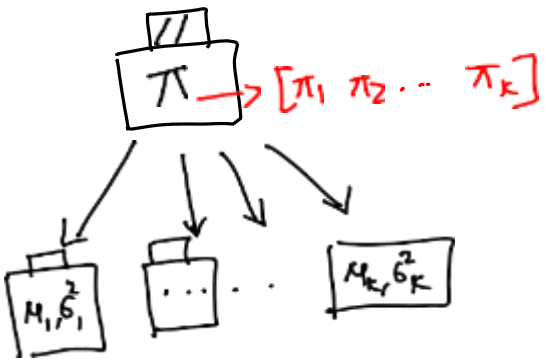
STEP 1: Generate a mixture component among $\{1, \dots, k\}$ $z_i \in \{1, \dots, k\}$

$$P(z_i = \ell) = \pi_\ell$$

$$\left[\sum_{\ell=1}^{k} \pi_\ell = 1 \quad 0 \le \pi_\ell \le 1 \quad \forall \ell \right.$$

STEP 2: Generate $x_i \sim N(\mu_{z_i}, \sigma^2_{z_i})$

Gaussian Mixture Model



$\{x_1, \ldots, x_n\}$ → OBSERVED

$\{z_1, \ldots, z_n\}$ → UNOBSERVED/ LATENT

LATENT VARIABLE MODEL

PARAMETERS ?

$\pi = [\pi_1, \ldots, \pi_K]$

μ_k, σ_k^2 $\forall k$

$2K + K - 1$

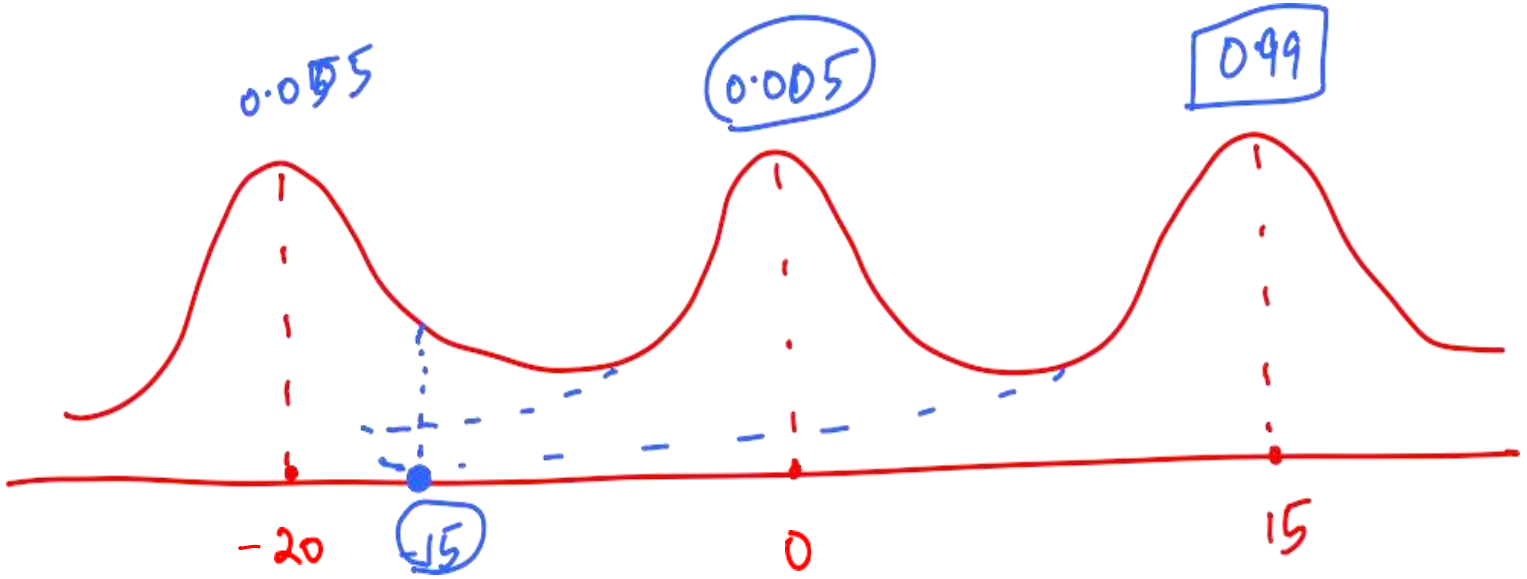
Total: $3K - 1$

MAXIMUM LIKELIHOOD

$$L\left(\begin{matrix}\mu_1,\dots,\mu_K\\ \sigma_1^2,\dots,\sigma_K^2\\ \pi_1,\dots,\pi_K\end{matrix};\ x_1,\dots,x_n\right) = \prod_{i=1}^{n} f_{mix}\left(x_i;\ \begin{matrix}\mu_1,\dots,\mu_K\\ \sigma_1^2,\dots,\sigma_K^2\\ \pi_1,\dots\pi_K\end{matrix}\right)$$

$$= \prod_{i=1}^{n}\left[\sum_{k=1}^{K}\pi_k\, f\left(x_i;\mu_k,\sigma_k^2\right)\right]$$

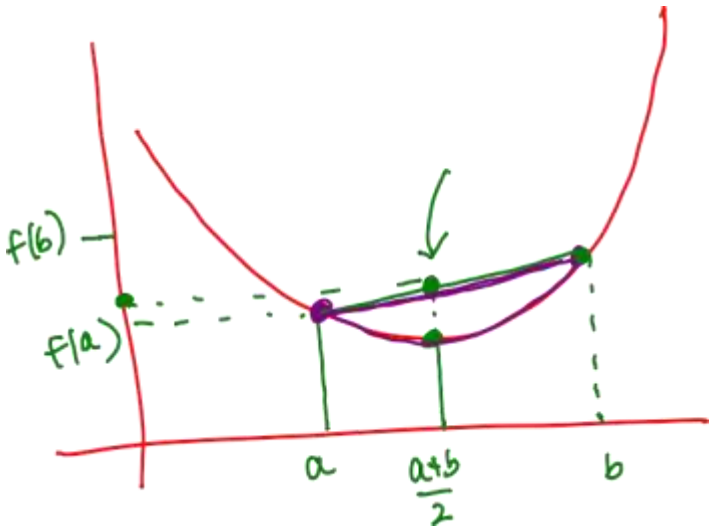
Gaussian density



$$\log L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_k} \right)$$

- Not possible to solve analytically.
- Need some alternate ways to solve $\max_{\theta} \log L(\theta)$

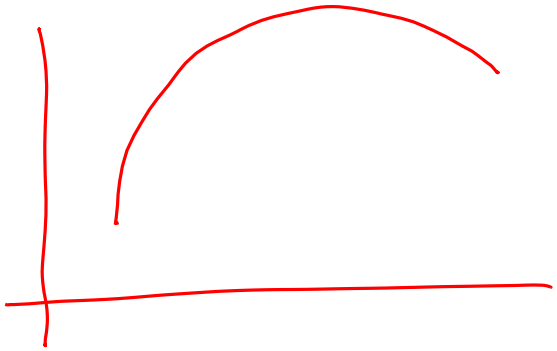
Quick Detour - Convex functions



$\forall a, b$

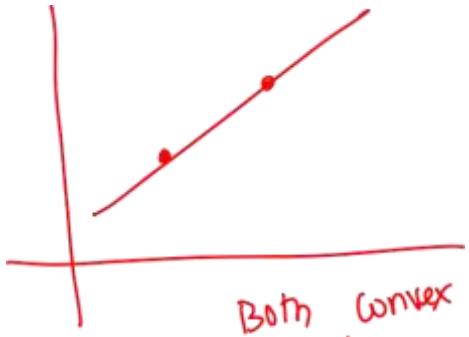
$$f\left(\frac{a+b}{2}\right) \leq \frac{f(a)+f(b)}{2}$$

$\Rightarrow$ convex function

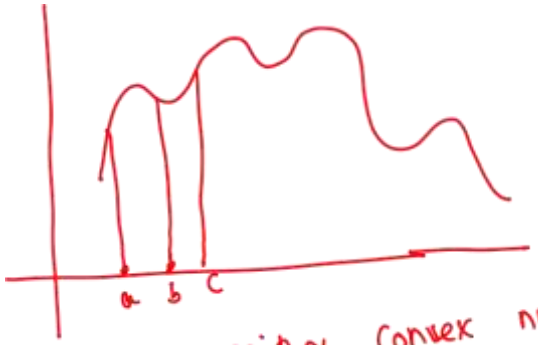


$$f\left(\frac{a+b}{2}\right) \geq \frac{f(a)+f(b)}{2}$$

$\Rightarrow$ concave functions.



Both Convex and Concave.



Neither Convex nor Concave.

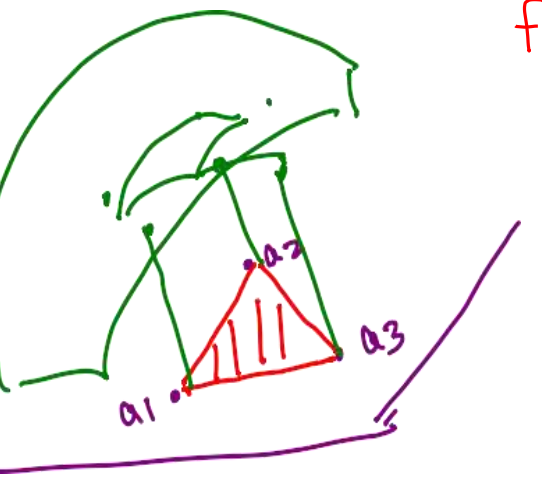
$$\forall a, b \quad f\left(\frac{1}{2}a + \frac{1}{2}b\right) \leq \frac{1}{2}f(a) + \frac{1}{2}f(b)$$

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda) f(b) \quad * \lambda \in [0,1]$$

f - concave

$$f(\lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_k a_k) \geq \lambda_1 f(a_1) + \cdots + \lambda_k f(a_k)$$

$$\sum_{i=1}^{k} \lambda_i = 1 \qquad 0 \leq \lambda_i \leq 1 \quad \forall i$$



JENSEN'S INEQUALITY

$$f\left(\sum_{k=1}^{K} \lambda_k a_k\right) \geq \sum_{k=1}^{K} \lambda_k f(a_k)$$

LOG IS A CONCAVE FUNCTION!

Recall

$$\log L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \left(\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k} \right) \right)$$

Introduce for every datapoint i, $\{\lambda_1^i, \ldots, \lambda_K^i\}$ s.t

$$\sum_{k=1}^{K} \lambda_k^i = 1 \quad \forall i \qquad 0 \le \lambda_k^i \le 1 \quad \forall i, \forall k.$$

$$\log L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \lambda_k^i \left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right) \right)$$

$$\log L(\theta) \geq \text{modified-}\log L(\theta, \lambda)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log \left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right)$$

[JENSEN'S]

- Note the mod. $\log L(\theta, \lambda)$ gives a lower bound for $\log L(\theta)$ for any choice of λ

• What are we gaining?

KEY INSIGHT →

- If we fix λ, it is "easy" to maximize w.r.t θ
- If we fix θ, maximize w.r.t λ is easy.

Fix λ and max over θ (mid-logL(θ, λ))

$$\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{k} \lambda_k^i \log \left(\frac{\pi_k \, e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right)$$

Take derivative w.r.t μ, σ^2, we get

$$\hat{\mu}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\hat{\sigma}_k^{2\,MML} = \frac{\sum_{i=1}^{n} \lambda_k^i \left(x_i - \hat{\mu}_k^{MML}\right)^2}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\max_{\pi_1, \ldots, \pi_k} \; mod\_logL(\theta, \lambda)$$

$$\text{s.t} \quad \sum_{i=1}^{k} \pi_i = 1$$

$$\hat{\pi}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}$$

Fix θ and maximize w.r.t λ

$$\max_{\lambda} \sum_{i=1}^{n} \left(\sum_{k=1}^{K} \lambda_k^i \log \left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}}{\lambda_k^i} \right) \right)$$

$\downarrow$

max for each i separately.

Solving this gives.

$$\hat{\lambda}_k^{i\,MML} = \frac{\left(\frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \right) (\pi_k)}{\sum_{l=1}^{K} \frac{1}{\sqrt{2\pi}\sigma_l} e^{-\frac{(x_i - \mu_l)^2}{2\sigma_l^2}} \pi_l}$$

(Dempster et al) 1970s

EM - ALGORITHM

→ Initialization $\theta^0 = \{ \mu_1^0, \ldots, \mu_k^0, \sigma_1^{2^0}, \ldots, \sigma_k^{2^0}, \pi_1^0, \ldots, \pi_k^0 \}$

Iteration

Tolerance.

until convergence $(\| \theta^{t+1} - \theta^t \| \leq \epsilon)$

EXPECTATION STEP ——— $\lambda^{t+1} = \underset{\lambda}{\operatorname{argmax}}$ modified_log L (θ^t, λ)

MAXIMIZATION STEP ←——— $\theta^{t+1} = \underset{\theta}{\operatorname{argmax}}$ modified_log L (θ, λ^{t+1})

end.

Expectation Maximization for GMM

-> Initialize all parameters (mean, variance, mixture proportions)
that one has to estimate:

-> Until convergence,

-> Assume the current parameters are correct and find the chance that each point came from every mixture

-> Assume the chance that each point comes from every mixture is correct and find the best parameters.

(Dempster et al) 1970s

EM - ALGORITHM

→ Initialization $\theta^0 = \{ \mu_1^0, \ldots, \mu_k^0, \sigma_1^{2^0}, \ldots, \sigma_k^{2^0}, \pi_1^0, \ldots, \pi_k^0 \}$

Iteration

Tolerance.

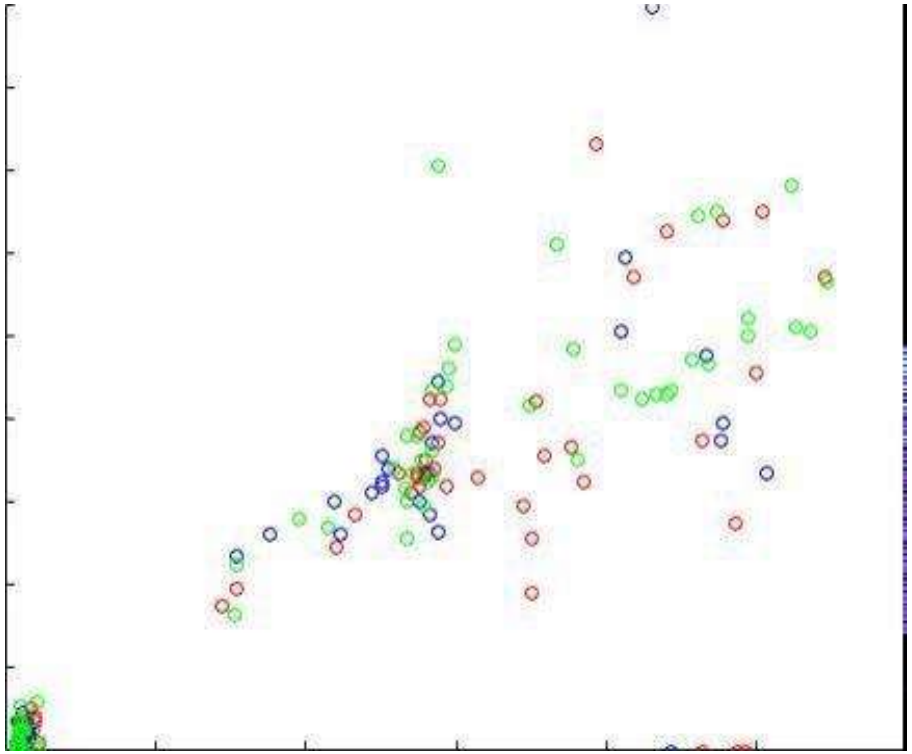
until convergence $(\| \theta^{t+1} - \theta^t \| \leq \epsilon)$

EXPECTATION STEP ← $\lambda^{t+1} = \underset{\lambda}{\text{argmax}}$ modified_log L (θ^t, λ)

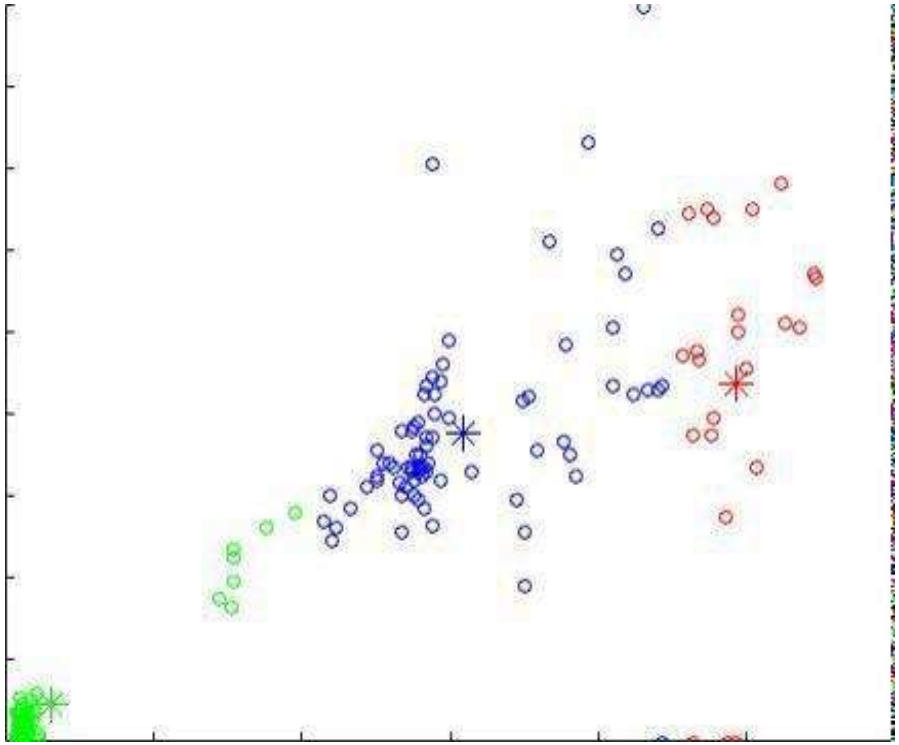
MAXIMIZATION STEP ← $\theta^{t+1} = \underset{\theta}{\text{argmax}}$ modified_log L (θ, λ^{t+1})

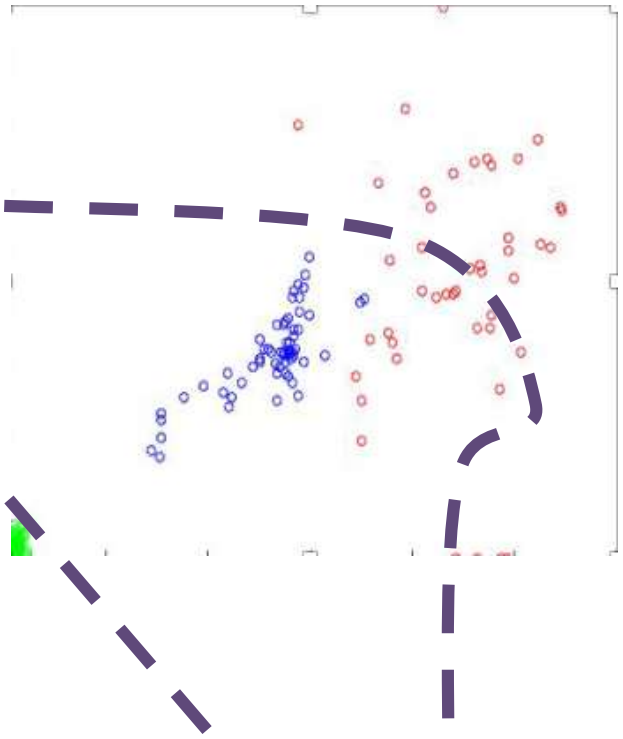
end.

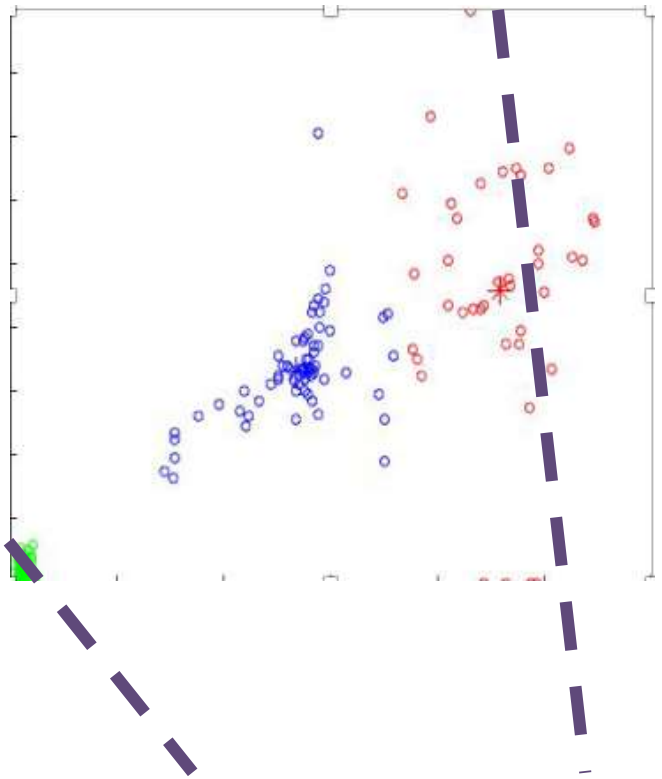
EM-illustration



K means Illustration



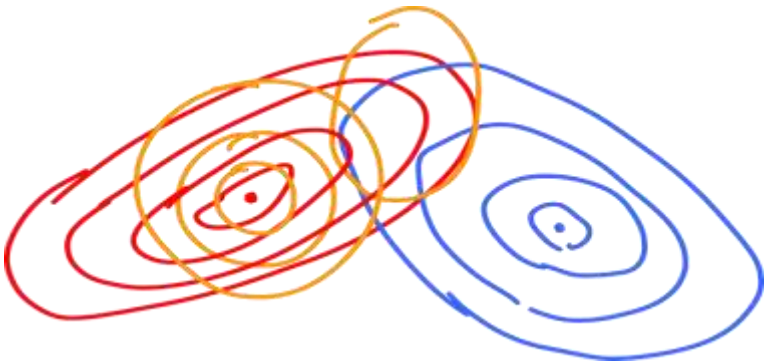




A hard clustering is producing at every round by
assigning data to the cluster with highest probability

- EM produces "soft clustering" $\lambda_k^i - P(z_i|x_i)$

- EM takes into account variance of clusters as well



Why does EM work?

Recall

$$MML(\theta^t, \lambda)\ \ \theta^t) = \sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i . \log\left(\pi_k^t . f(x_i; \mu_k^t, (\sigma_k^2)^t)/\lambda_k^i\right)$$

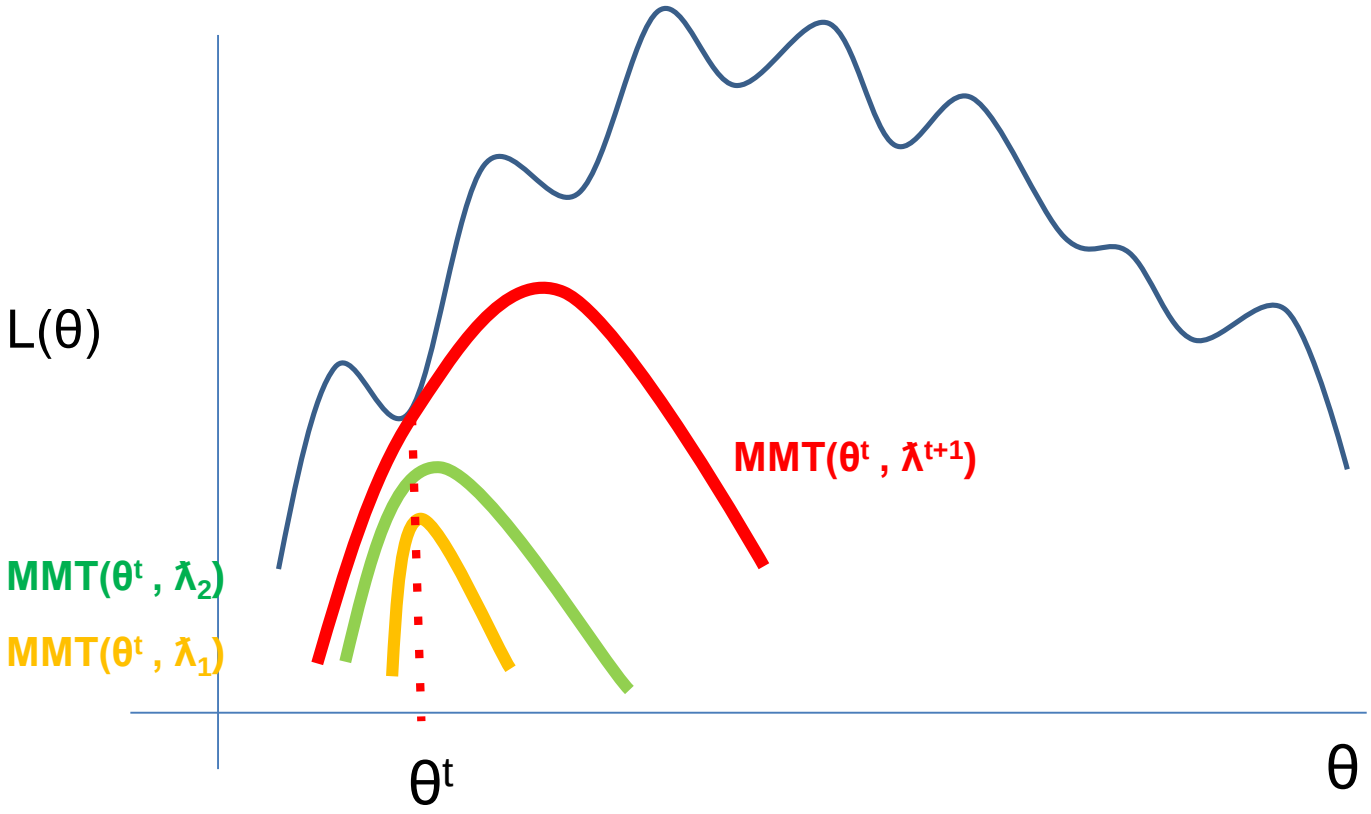
We know If we believed in the parameters Θ^t, then the best guess for λ is

$$\lambda_k^{i*}{}^{t+1} = \frac{f(x_i; \mu_k^t, \sigma_k^{2t}) . \pi_k^t}{f(x_i)}$$

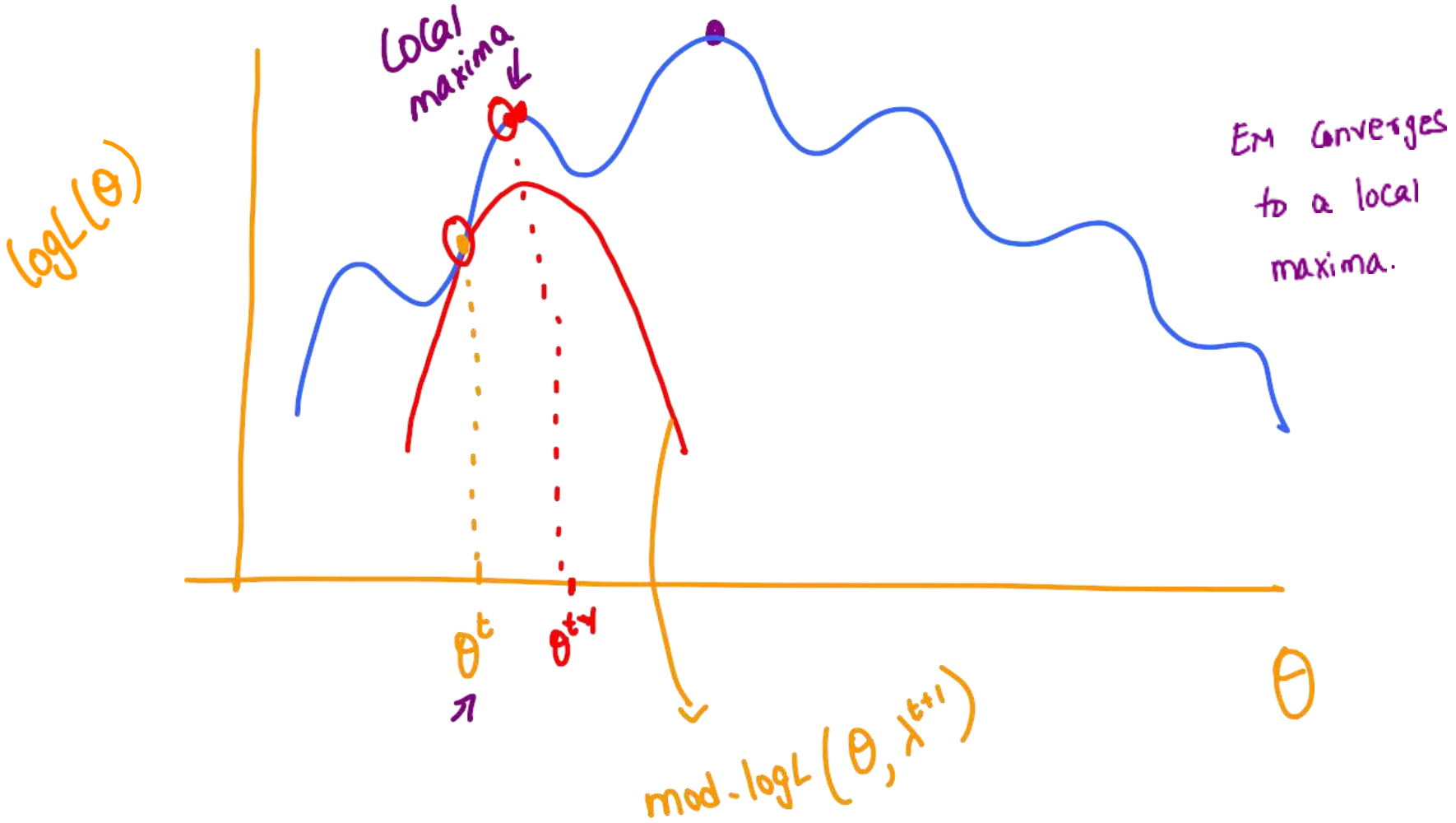
Substituting back the best values in MML, we can easily show that

$$MML(\theta^t, \lambda^{t+1}) = Loglikelihood(\theta^t)$$

So what?



$$MML(\lambda^{t+1}, \theta^t) = Loglikelihood(\theta^t)$$



$$\log L(\theta^t)) = MML(\theta^t, \lambda^{t+1}) \leq MML(\theta^{t+1}, \lambda^{t+1}) \leq \log L(\theta^{t+1}))$$

PREVIOUS SLIDE

MAXIMIZATION

JENSEN

A few more points

Latent Variable Models are very popular.

EM is a general idea – not just for GMM

There is a probabilistic variant of PCA as well – called Probabilistic PCA

Summary

Unsupervised Learning

Representation learning
PCA, Kernel PCA

Clustering
K-means, Spectral Clustering

Estimation
Max-Likelihood, Bayesian
EM