
MSCIT5210: Knowledge Discovery and Data Mining

Acknowledgement: Slides modified by Dr. Lei Chen based on the slides provided by Jiawei Han, Micheline Kamber, and Jian Pei and Raymond Wong

©2012 Han, Kamber & Pei & Wong All rights reserved.

Outline of Advanced Clustering Analysis

- Probability Model-Based Clustering
 - Each object may take a probability to belong to a cluster
- Clustering High-Dimensional Data
 - Curse of dimensionality: Difficulty of distance measure in high-D space

Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering
- Clustering High-Dimensional Data
- Summary



Fuzzy Set and Fuzzy Cluster

- Clustering methods discussed so far
 - Every data object is assigned to exactly one cluster
- Some applications may need for fuzzy or soft cluster assignment
 - Ex. An e-game could belong to both entertainment and software
- Methods: fuzzy clusters and probabilistic model-based clusters
- Fuzzy cluster: A fuzzy set $S: F_S : X \rightarrow [0, 1]$ (value between 0 and 1)
- Example: Popularity of cameras is defined as a fuzzy mapping

Camera	Sales (units)
A	50
B	1320
C	860
D	270

$$\text{Pop}(o) = \begin{cases} 1 & \text{if } 1,000 \text{ or more units of } o \text{ are sold} \\ \frac{i}{1000} & \text{if } i \ (i < 1000) \text{ units of } o \text{ are sold} \end{cases}$$

- Then, $A(0.05)$, $B(1)$, $C(0.86)$, $D(0.27)$

Fuzzy (Soft) Clustering

- Example: Let cluster features be
 - C_1 : “digital camera” and “lens”
 - C_2 : “computer”
- Fuzzy clustering
 - k fuzzy clusters C_1, \dots, C_k , represented as a partition matrix $M = [w_{ij}]$
 - P1: for each object o_i and cluster C_j , $0 \leq w_{ij} \leq 1$ (fuzzy set)
 - P2: for each object o_i , $\sum_{j=1}^k w_{ij} = 1$, equal participation in the clustering
 - P3: for each cluster C_j , $0 < \sum_{i=1}^n w_{ij} < n$ ensures there is no empty cluster
- Let c_1, \dots, c_k as the center of the k clusters
- For an object o_i , sum of the squared error (SSE), p is a parameter:
- For a cluster C_j , SSE:

$$\text{SSE}(C_j) = \sum_{i=1}^n w_{ij}^p \text{dist}(o_i, c_j)^2$$

$$\text{SSE}(o_i) = \sum_{j=1}^k w_{ij}^p \text{dist}(o_i, c_j)^2$$
- Measure how well a clustering fits the data:

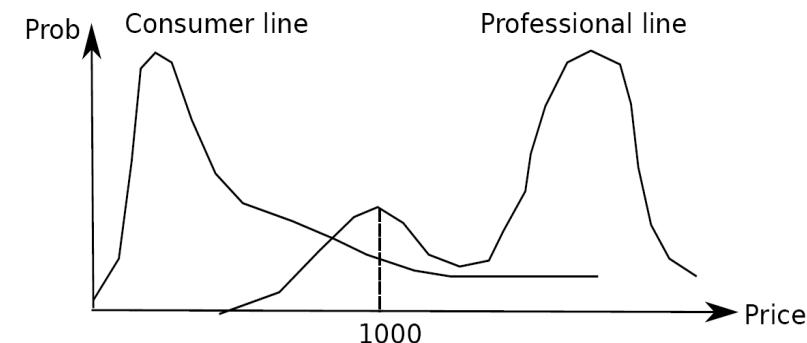
$$\text{SSE}(\mathcal{C}) = \sum_{i=1}^n \sum_{j=1}^k w_{ij}^p \text{dist}(o_i, c_j)^2$$

Review-id	Keywords
R_1	digital camera, lens
R_2	digital camera
R_3	lens
R_4	digital camera, lens, computer
R_5	computer, CPU
R_6	computer, computer game

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Probabilistic Model-Based Clustering

- Cluster analysis is to find hidden categories.
- A hidden category (i.e., *probabilistic cluster*) is a distribution over the data space, which can be mathematically represented using a probability density function (or distribution function).
- Ex. 2 categories for digital cameras sold
 - consumer line vs. professional line
 - density functions f_1, f_2 for C_1, C_2
 - obtained by probabilistic clustering
- A **mixture model** assumes that a set of observed objects is a mixture of instances from multiple probabilistic clusters, and conceptually each observed object is generated independently
- **Out task:** infer a set of k probabilistic clusters that is mostly likely to generate D using the above data generation process



Model-Based Clustering

- A set C of k probabilistic clusters C_1, \dots, C_k with probability density functions f_1, \dots, f_k , respectively, and their probabilities $\omega_1, \dots, \omega_k$.
- Probability of an object o generated by cluster C_j is $P(o|C_j) = \omega_j f_j(o)$
- Probability of o generated by the set of cluster C is $P(o|C) = \sum_{j=1}^k \omega_j f_j(o)$
- Since objects are assumed to be generated independently, for a data set $D = \{o_1, \dots, o_n\}$, we have,

$$P(D|C) = \prod_{i=1}^n P(o_i|C) = \prod_{i=1}^n \sum_{j=1}^k \omega_j f_j(o_i)$$

- Task: Find a set C of k probabilistic clusters s.t. $P(D|C)$ is maximized
- However, maximizing $P(D|C)$ is often intractable since the probability density function of a cluster can take an arbitrarily complicated form
- To make it computationally feasible (as a compromise), assume the probability density functions being some parameterized distributions

Univariate Gaussian Mixture Model

- $O = \{o_1, \dots, o_n\}$ (n observed objects), $\Theta = \{\theta_1, \dots, \theta_k\}$ (parameters of the k distributions), and $P_j(o_i | \theta_j)$ is the probability that o_i is generated from the j -th distribution using parameter θ_j , we have

$$P(o_i | \Theta) = \sum_{j=1}^k \omega_j P_j(o_i | \Theta_j) \quad P(\mathbf{O} | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \omega_j P_j(o_i | \Theta_j)$$

- Univariate Gaussian mixture model
 - Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters.
 - The probability density function of each cluster are centered at μ_j with standard deviation σ_j , $\theta_j = (\mu_j, \sigma_j)$, we have

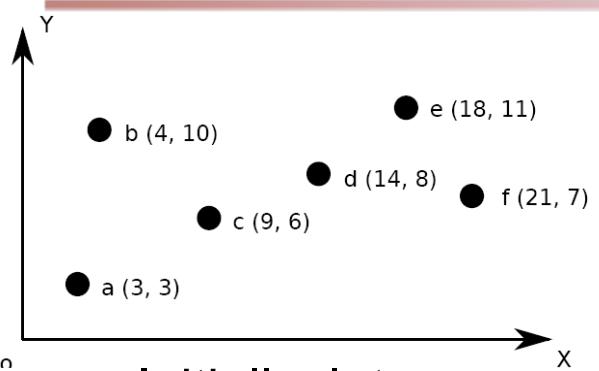
$$P(o_i | \Theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}} \quad P(o_i | \Theta) = \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

$$P(\mathbf{O} | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

The EM (Expectation Maximization) Algorithm

- The k-means algorithm has two steps at each iteration:
 - **Expectation Step (E-step):** Given the current cluster centers, each object is assigned to the cluster whose center is closest to the object: An object is *expected to belong to the closest cluster*
 - **Maximization Step (M-step):** Given the cluster assignment, for each cluster, the algorithm *adjusts the center so that the sum of distance* from the objects assigned to this cluster and the new center is minimized
- **The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
 - **E-step** assigns objects to clusters according to the current fuzzy clustering or parameters of probabilistic clusters
 - **M-step** finds the new clustering or parameters that maximize the sum of squared error (SSE) or the expected likelihood

Fuzzy Clustering Using the EM Algorithm



Iteration	E-step						M-step	
	$M^T =$	1	0	0.48	0.42	0.41	0.47	
1	$M^T =$	1	0	0.48	0.42	0.41	0.47	$c_1 = (8.47, 5.12)$, $c_2 = (10.42, 8.99)$
2	$M^T =$	0.73	0.49	0.91	0.26	0.33	0.42	$c_1 = (8.51, 6.11)$, $c_2 = (14.42, 8.69)$
3	$M^T =$	0.80	0.76	0.99	0.02	0.14	0.23	$c_1 = (6.40, 6.24)$, $c_2 = (16.55, 8.64)$

- Initially, let $c_1 = a$ and $c_2 = b$
- 1st E-step: assign o to c_1 , w. wt = $\frac{\frac{1}{dist(o,c_1)^2}}{\frac{1}{dist(o,c_1)^2} + \frac{1}{dist(o,c_2)^2}} = \frac{dist(o,c_2)^2}{dist(o,c_1)^2 + dist(o,c_2)^2}$
 - $w_{c,c_1} = \frac{41}{45+41} = 0.48$
- 1st M-step: recalculate the centroids according to the partition matrix, minimizing the sum of squared error (SSE)
$$c_j = \frac{\sum_{\text{each point } o} w_{o,c_j}^2 o}{\sum_{\text{each point } o} w_{o,c_j}^2}$$

$$c_1 = \frac{\left(\frac{1^2 \times 3 + 0^2 \times 4 + 0.48^2 \times 9 + 0.42^2 \times 14 + 0.41^2 \times 18 + 0.47^2 \times 21}{1^2 + 0^2 + 0.48^2 + 0.42^2 + 0.41^2 + 0.47^2}\right)}{\left(\frac{1^2 \times 3 + 0^2 \times 10 + 0.48^2 \times 6 + 0.42^2 \times 8 + 0.41^2 \times 11 + 0.47^2 \times 7}{1^2 + 0^2 + 0.48^2 + 0.42^2 + 0.41^2 + 0.47^2}\right)}$$

$$= (8.47, 5.12)$$
- Iteratively calculate this until the cluster centers converge or the change is small enough

Fuzzy Clustering Using the EM

- Next will give a detailed illustration of Slide 10 in [11ClusAdvanced.pdf](#)
- Consider following six points:

a(3,3), b(4,10), c(9,6), d(14,8), e(18,11), f(21, 7)

Complete the fuzzing clustering using the Expectation-Maximization algorithm.

Data Points

- ◆ $b(4, 10)$
- ◆ $c(9, 6)$
- ◆ $a(3, 3)$
- ◆ $d(14, 8)$
- ◆ $e(18, 11)$
- ◆ $f(21, 7)$

0

5

10

15

20

25

Fuzzy Clustering Using the EM

- Initially, let $c_1 = a$ and $c_2 = b$
- 1st E-step: assign objects to clusters: c_1 and c_2
- Calculate the weight for each object for each cluster: w_{ij} means the weight of object i in cluster j. **Below is a specific formula for this question only since there are only two clusters here.**
- w_{i1} means the weight of object i in cluster 1 (c_1).

$$w_{i1} = \frac{\frac{1}{dist(o_i, c_1)^2}}{\frac{1}{dist(o_i, c_1)^2} + \frac{1}{dist(o_i, c_2)^2}} = \frac{dist(o_i, c_2)^2}{dist(o_i, c_2)^2 + dist(o_i, c_1)^2}$$

Fuzzy Clustering Using the EM

- Calculate w_{i2} : the weight of object i in cluster 2 (c_2).

$$w_{i2} = \frac{\frac{1}{dist(o_i, c_2)^2}}{\frac{1}{dist(o_i, c_1)^2} + \frac{1}{dist(o_i, c_2)^2}} = \frac{dist(o_i, c_1)^2}{dist(o_i, c_2)^2 + dist(o_i, c_1)^2}$$

For this case particularly, we can use a simple way to calculate w_{i2}

$$w_{i2} = 1 - w_{i1} .$$

This is because there are only two clusters in this case, and it also obeys the rule $\sum_1^k w_{ij} = 1$

E-step in the 1st Iteration

- With this formula, we can calculate the weight for each object in c_1
- Next will calculate object c , because a is c_1 and b is c_2 , so**

$$w_{a1} = 1, w_{a2} = 0, w_{b1} = 0, w_{b2} = 1,$$

$$\begin{aligned} w_{c1} &= \frac{dist(c, c_2)^2}{dist(c, c_2)^2 + dist(c, c_1)^2} = \frac{(9-4)^2 + (6-10)^2}{(9-4)^2 + (6-10)^2 + (9-3)^2 + (6-3)^2} \\ &= \frac{41}{41+45} = \mathbf{0.48} \end{aligned}$$

Then use the simple method to calculate $w_{c2} = 1 - 0.48 = \mathbf{0.52}$

E-step in the 1st Iteration

- Take point **d** for another example:

$$\begin{aligned} \mathbf{w}_{d1} &= \frac{\text{dist}(d, c_2)^2}{\text{dist}(d, c_2)^2 + \text{dist}(d, c_1)^2} = \frac{(14-4)^2 + (8-10)^2}{(14-4)^2 + (8-10)^2 + (14-3)^2 + (8-3)^2} \\ &= \frac{104}{104+146} = \mathbf{0.42} \end{aligned}$$

Then use the simple method to calculate $\mathbf{w}_{d2} = 1 - 0.42 = \mathbf{0.58}$

Similarly, we can calculate the other weights:

$$\mathbf{w}_{e1} = \mathbf{0.41}, \mathbf{w}_{e2} = 1 - 0.41 = \mathbf{0.59}$$

$$\mathbf{w}_{f1} = \mathbf{0.47}, \mathbf{w}_{f2} = 1 - 0.47 = \mathbf{0.53}$$

Partition Matrix in the 1st Iteration

- Now we can draw the partition Matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.48 & 0.52 \\ 0.42 & 0.58 \\ 0.41 & 0.59 \\ 0.47 & 0.53 \end{bmatrix}$$

- Each row in Partition Matrix represents an Object(a Point in this case)
- Each column represents a Cluster.

Partition Matrix in the 1st Iteration

- $M = \begin{bmatrix} & c_1 & c_2 \\ a & 1 & 0 \\ b & 0 & 1 \\ c & 0.48 & 0.52 \\ d & 0.42 & 0.58 \\ e & 0.41 & 0.59 \\ f & 0.47 & 0.53 \end{bmatrix}$
- Each **row** in Partition Matrix represents an **Object**(a Point in this case)
- Each **column** represents a **Cluster**.

Transposition Matrix in the 1st Iteration

- $M^T = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline [1] & 0 & 0.48 & 0.42 & 0.41 & 0.47 & c_1 \\ [0] & 1 & 0.52 & 0.58 & 0.59 & 0.53 & c_2 \end{array}$

- For the **transposition** of the Matrix:
- Each **column** in Transposition Matrix represents an **Object**(a Point in this case)
- Each **row** represents a **Cluster**.

$$\mathbf{M}^T = \begin{bmatrix} 1 & 0 & 0.48 & 0.42 & 0.41 & 0.47 \\ 0 & 1 & 0.52 & 0.58 & 0.59 & 0.53 \end{bmatrix}$$

- ◆ e(18, 11)
- ◆ b(4, 10)
- ◆ d(14, 8)
- ◆ f(21, 7)
- ◆ c(9, 6)
- ◆ a(3, 3)

**Next is the M-step: recalculate the centroids
according to the partition matrix**

M-step in the 1st Iteration

- 1st M-step: **recalculate the centroids** according to the partition matrix.

- $c_j = \frac{\sum_{\text{each point } o} (w^2_{o,c_j} * o)}{\sum_{\text{each point } o} w^2_{o,c_j}}$, for example, calculate c_1 :

- $c_1 = \left(\frac{\sum_{\text{each point } o} (w^2_{o,c_1} * o_x)}{\sum_{\text{each point } o} w^2_{o,c_1}}, \frac{\sum_{\text{each point } o} (w^2_{o,c_1} * o_y)}{\sum_{\text{each point } o} w^2_{o,c_1}} \right)$

$$c_1 = \left(\frac{\frac{1^2 * 3 + 0^2 * 4 + 0.48^2 * 9 + 0.42^2 * 14 + 0.41^2 * 18 + 0.47^2 * 21}{1^2 + 0^2 + 0.48^2 + 0.42^2 + 0.41^2 + 0.47^2}, \frac{1^2 * 3 + 0^2 * 10 + 0.48^2 * 6 + 0.42^2 * 8 + 0.41^2 * 11 + 0.47^2 * 7}{1^2 + 0^2 + 0.48^2 + 0.42^2 + 0.41^2 + 0.47^2}} \right)$$

$$= (8.47, 5.12)$$

M-step in the 1st Iteration

- 1st M-step: recalculate the centroids for cluster 2.

- $c_2 = \left(\frac{\sum_{\text{each point } o} (w_{o,c_2}^2 * o_x)}{\sum_{\text{each point } o} w_{o,c_2}^2}, \frac{\sum_{\text{each point } o} (w_{o,c_2}^2 * o_y)}{\sum_{\text{each point } o} w_{o,c_2}^2} \right)$

$$c_2 = \left(\frac{\begin{matrix} 0^2 * 3 + 1^2 * 4 + 0.52^2 * 9 + 0.58^2 * 14 + 0.59^2 * 18 + 0.53^2 * 21 \\ 0^2 + 1^2 + 0.52^2 + 0.58^2 + 0.59^2 + 0.53^2 \end{matrix}}{\begin{matrix} 0^2 * 3 + 1^2 * 10 + 0.52^2 * 6 + 0.58^2 * 8 + 0.59^2 * 11 + 0.53^2 * 7 \\ 0^2 + 1^2 + 0.52^2 + 0.58^2 + 0.59^2 + 0.53^2 \end{matrix}}, \right)$$
$$= (10.42, 8.99)$$

The 1st Iteration Result

Iteration	E-Step	M-Step
1	$M^T = \begin{bmatrix} 1 & 0 & 0.48 & 0.42 & 0.41 & 0.47 \\ 0 & 1 & 0.52 & 0.58 & 0.59 & 0.53 \end{bmatrix}$	$c_1 = (8.47, 5.12)$ $c_2 = (10.42, 8.99)$

Now the first iteration is over, we should **repeat** the same process.

Go to the 2nd E-step: assign objects to new clusters: c_1 and c_2

Now is E-step in the 2nd Iteration

- New cluster centers: $c_1 = (8.47, 5.12)$ and $c_2 = (10.42, 8.99)$
- 2nd E-step: assign objects to new clusters: c_1 and c_2
- Calculate the weight for each object for each cluster: w_{ij}
- For example, calculate weight of a :
- $w_{a1} = \frac{dist(a, c_2)^2}{dist(a, c_2)^2 + dist(a, c_1)^2} = \frac{(3-10.42)^2 + (3-8.99)^2}{(3-10.42)^2 + (3-8.99)^2 + (3-8.47)^2 + (3-5.12)^2} = \frac{90.9365}{90.9365 + 34.4153} = \frac{90.9365}{125.3518} = 0.73$
- $w_{a2} = 1 - w_{a1} = 1 - 0.73 = 0.27$
- Similarly, we can calculate the other points' weight.

M-step in the 2nd Iteration

- After all weights are calculated, we now get the Matrix again.
- $M^T = \begin{bmatrix} 0.73 & 0.49 & 0.91 & 0.26 & 0.33 & 0.42 \\ 0.27 & 0.51 & 0.09 & 0.74 & 0.67 & 0.58 \end{bmatrix}$
- The 2nd E-step is over, continue the 2nd M-step:
- For example, calculate new c_1 :
- $c_1 = \left(\frac{\sum_{\text{each point } o} (w_{o,c_1}^2 * o_x)}{\sum_{\text{each point } o} w_{o,c_1}^2}, \frac{\sum_{\text{each point } o} (w_{o,c_1}^2 * o_y)}{\sum_{\text{each point } o} w_{o,c_1}^2} \right)$
 $c_1 = \left(\frac{0.73^2 * 3 + 0.49^2 * 4 + 0.91^2 * 9 + 0.26^2 * 14 + 0.33^2 * 18 + 0.42^2 * 21}{0.73^2 + 0.49^2 + 0.91^2 + 0.26^2 + 0.33^2 + 0.42^2}, \frac{0.73^2 * 3 + 0.49^2 * 10 + 0.91^2 * 6 + 0.26^2 * 8 + 0.33^2 * 11 + 0.42^2 * 7}{0.73^2 + 0.49^2 + 0.91^2 + 0.26^2 + 0.33^2 + 0.42^2} \right)$
= (8.51, 6.11)

M-step in the 2nd Iteration

- 2nd M-step: recalculate the centroids for cluster 2.
- $c_2 = \left(\frac{\sum_{\text{each point } o} (w_{o,c_2}^2 * o_x)}{\sum_{\text{each point } o} w_{o,c_2}^2}, \frac{\sum_{\text{each point } o} (w_{o,c_2}^2 * o_y)}{\sum_{\text{each point } o} w_{o,c_2}^2} \right)$
$$c_2 = \left(\frac{0.27^2 * 3 + 0.51^2 * 4 + 0.09^2 * 9 + 0.74^2 * 14 + 0.67^2 * 18 + 0.58^2 * 21}{0.27^2 + 0.51^2 + 0.09^2 + 0.74^2 + 0.67^2 + 0.58^2}, \frac{0^2 * 3 + 1^2 * 10 + 0.52^2 * 6 + 0.58^2 * 8 + 0.59^2 * 11 + 0.53^2 * 7}{0.27^2 + 0.51^2 + 0.09^2 + 0.74^2 + 0.67^2 + 0.58^2} \right)$$
$$= (14.42, 8.69)$$

The 2nd Iteration Result

Iteration	E-Step	M-Step
2	$M^T = \begin{bmatrix} 0.73 & 0.49 & 0.91 & 0.26 & 0.33 & 0.42 \\ 0.27 & 0.51 & 0.09 & 0.74 & 0.67 & 0.58 \end{bmatrix}$	$c_1 = (8.51, 6.11)$ $c_2 = (14.42, 8.69)$

Now the second iteration is over, but the centers do not converge, so we need to **repeat** the same process.

Go to the 3rd E-step: assign objects to new clusters: c_1 and c_2

The 3rd Iteration Result

Iteration	E-Step	M-Step
3	$M^T = \begin{bmatrix} 0.80 & 0.76 & 0.99 & 0.02 & 0.14 & 0.23 \\ 0.20 & 0.24 & 0.01 & 0.98 & 0.86 & 0.77 \end{bmatrix}$	$c_1 = (6.40, 6.24)$ $c_2 = (16.55, 8.64)$

Now the third iteration is over.

But it looks like the result is still not good because the cluster centers do not converge.

So we need to continue repeating again, **until the cluster centers converge or the change is small enough.**

Univariate Gaussian Mixture Model

- $O = \{o_1, \dots, o_n\}$ (n observed objects), $\Theta = \{\theta_1, \dots, \theta_k\}$ (parameters of the k distributions), and $P_j(o_i | \theta_j)$ is the probability that o_i is generated from the j -th distribution using parameter θ_j , we have

$$P(o_i | \Theta) = \sum_{j=1}^k \omega_j P_j(o_i | \Theta_j) \quad P(\mathbf{O} | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \omega_j P_j(o_i | \Theta_j)$$

- Univariate Gaussian mixture model
 - Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters.
 - The probability density function of each cluster are centered at μ_j with standard deviation σ_j , $\theta_j = (\mu_j, \sigma_j)$, we have

$$P(o_i | \Theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}} \quad P(o_i | \Theta) = \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

$$P(\mathbf{O} | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

Computing Mixture Models with EM

- Given n objects $O = \{o_1, \dots, o_n\}$, we want to mine a set of parameters $\Theta = \{\theta_1, \dots, \theta_k\}$ s.t., $P(O|\Theta)$ is maximized, where $\theta_j = (\mu_j, \sigma_j)$ are the mean and standard deviation of the j -th univariate Gaussian distribution
- We initially assign random values to parameters θ_j , then iteratively conduct the E- and M- steps until converge or sufficiently small change
- At the E-step, for each object o_i , calculate the probability that o_i belongs to each distribution,

$$P(\Theta_j|o_i, \Theta) = \frac{P(o_i|\Theta_j)}{\sum_{l=1}^k P(o_i|\Theta_l)}$$

- At the M-step, adjust the parameters $\theta_j = (\mu_j, \sigma_j)$ so that the expected likelihood $P(O|\Theta)$ is maximized

$$\mu_j = \sum_{i=1}^n o_i \frac{P(\Theta_j|o_i, \Theta)}{\sum_{l=1}^n P(\Theta_j|o_l, \Theta)} = \frac{\sum_{i=1}^n o_i P(\Theta_j|o_i, \Theta)}{\sum_{i=1}^n P(\Theta_j|o_i, \Theta)} \quad \sigma_j = \sqrt{\frac{\sum_{i=1}^n P(\Theta_j|o_i, \Theta)(o_i - \mu_j)^2}{\sum_{i=1}^n P(\Theta_j|o_i, \Theta)}}$$

Univariate Gaussian Mixture Model

- Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters.

The probability density function of each cluster are centered at μ_j with standard deviation σ_j , $\theta_j = (\mu_j, \sigma_j)$, we have:

$$P(o_i | \Theta_j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(o_i - \mu_j)^2}{2\sigma^2}},$$

$$P(\mathbf{o}_i | \Theta) = \sum_{j=1}^k w_j P(o_i | \Theta_j) = \sum_{j=1}^k w_j \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma^2}} \text{ (probability of } \mathbf{o}_i \text{)}$$

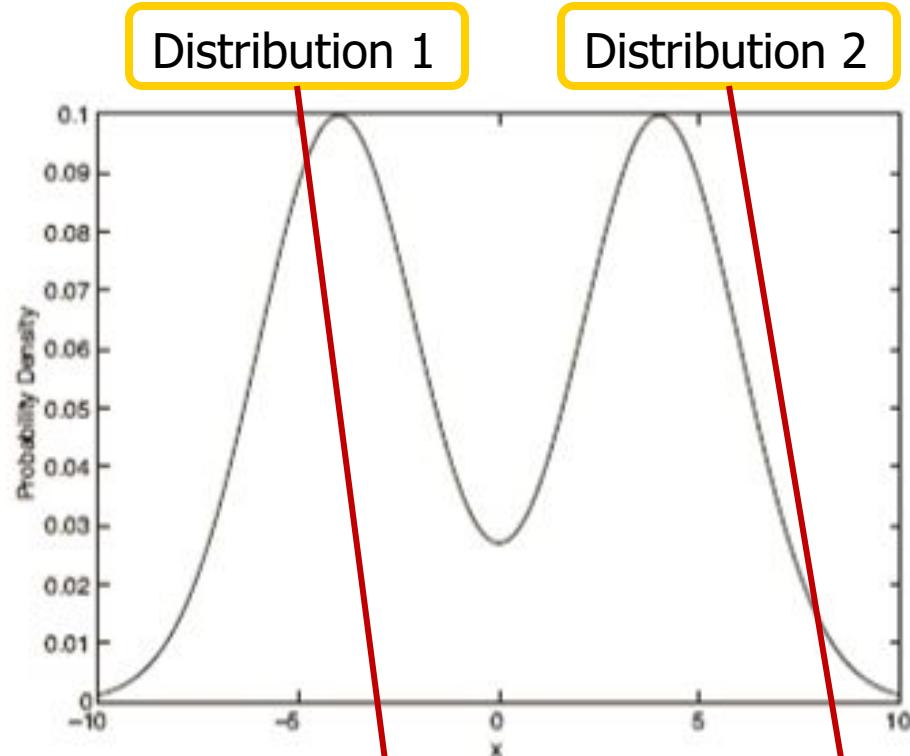
$$P(\mathbf{o} | \Theta) = \prod_{i=1}^n P(o_i | \Theta) = \prod_{i=1}^n \sum_{j=1}^k w_j \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma^2}}$$

Simple Mixture Model

Assume there are two Gaussian distributions, with a common standard deviation(σ) of 2 and means(μ) of -4 and 4, respectively. Also assume that each of the two distributions is selected with equal probability, i.e., $w_1 = w_2 = 0.5$.

So the probability of an object o is given:

$$P(o|\Theta) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(o_i+4)^2}{8}} + \frac{1}{2\sqrt{2\pi}} e^{-\frac{(o_i-4)^2}{8}}$$

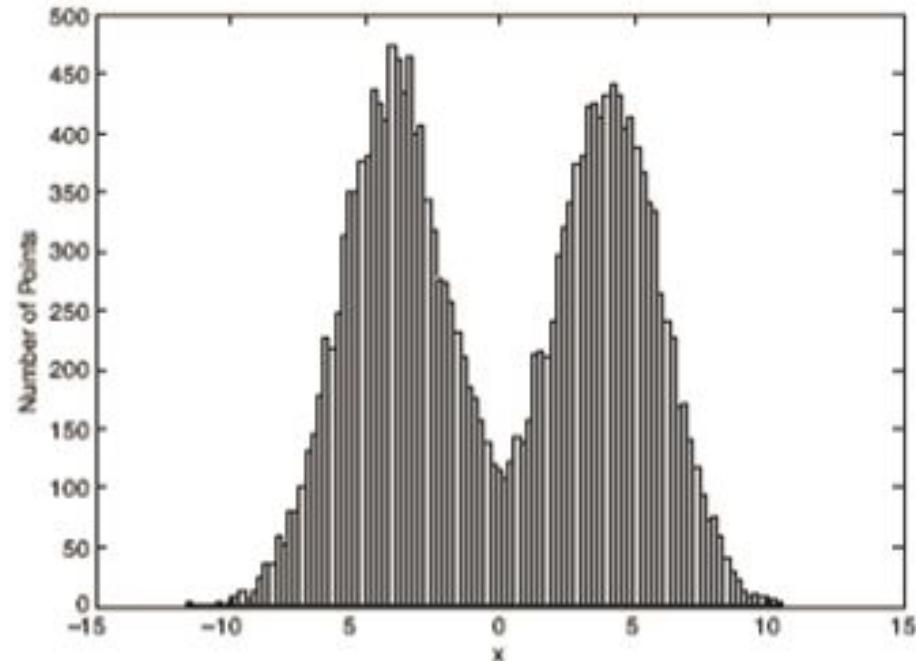


(a) Probability density function for the mixture model

model

Mixture model consisting of two normal distributions with means of -4 and 4, respectively. Both distributions have a standard deviation of 2.

$$P(o_i | \Theta) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(o_i+4)^2}{8}} + \frac{1}{2\sqrt{2\pi}} e^{-\frac{(o_i-4)^2}{8}}$$



(b) 20,000 points generated from the mixture model

Univariate Gaussian Mixture Model

- Given n objects $O = \{o_1, \dots, o_n\}$, we want to mine a set of parameters $\Theta = \{\theta_1, \dots, \theta_k\}$ s.t., $P(o|\Theta)$ is maximized, where $\theta_j = (\mu_j, \sigma_j)$ are the mean and standard deviation of the j -th univariate Gaussian distribution
- We initially assign random values to parameters θ_j , then iteratively conduct the E- and M- steps until converge or sufficiently small change
- At the E-step, for each object o_i , calculate the probability that o_i belongs to each distribution,

$$P(\theta_j|o_i, \Theta) = \frac{w_j P(o_i|\theta_j)}{\sum_{j=1}^k w_j P(o_i|\theta_j)}$$

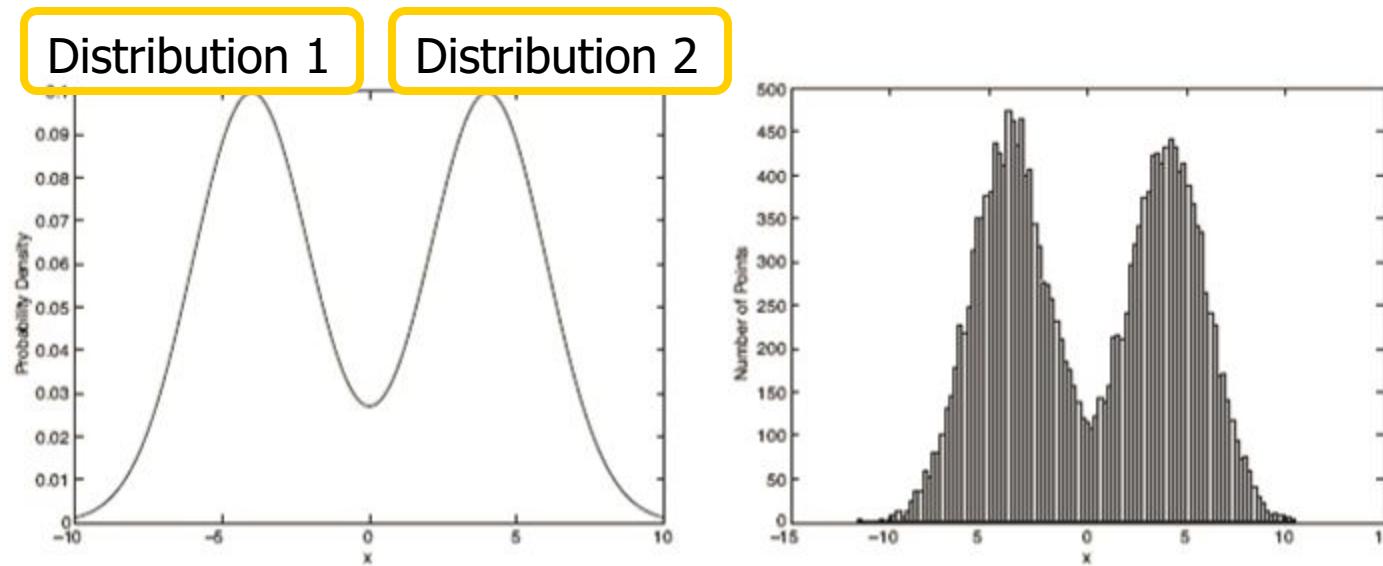
- At the M-step, adjust the parameters $\theta_j = (\mu_j, \sigma_j)$ so that the expected likelihood $P(o|\Theta)$ is maximized
- $$\mu_j = \sum_{i=1}^n o_i \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^k P(\theta_l|o_i, \Theta)} = \frac{\sum_{i=1}^n o_i P(\theta_j|o_i, \Theta)}{\sum_{i=1}^n P(\theta_j|o_i, \Theta)}, \sigma_j = \sqrt{\frac{\sum_{i=1}^n P(\theta_j|o_i, \Theta)(o_i - \mu_j)^2}{\sum_{i=1}^n P(\theta_j|o_i, \Theta)}}$$

EM algorithm

- 1. Select an **initial** set of model parameters.
 2. **Repeat**
 3. **E-Step:** for each object, calculate the probability that each object belongs to each distribution, i.e., calculate $P(\Theta_j | o_i, \Theta)$
 4. **M-Step:** given the probabilities from E-Step, find the new estimates of the parameters that maximize the expected likelihood.
 5. **Until** The parameters do not change.
 6. (Alternatively, stop if the change in the parameters is below a specified threshold)

A Simple Example of EM Algorithm

Consider the previous distributions, we assume that we know the **standard deviation(σ) of both distributions is 2.0** and that points were generated with **equal probability** from both distributions. We will refer to the left and right distributions as distributions 1 and 2, respectively.



Initialize Parameters and E-Step

- We can begin with $\mu_1 = -2$ and $\mu_2 = 3$. Thus, the initial parameters, $\Theta = (\mu, \sigma)$, for the two distributions are, respectively $\Theta_1 = (-2, 2)$ and $\Theta_2 = (3, 2)$. The set of parameters for the entire mixture model is $\Theta = \{\Theta_1, \Theta_2\}$

For the **Expectation** Step of EM, we want to compute the probability that a point came from a particular distribution; i.e., we want to compute $P(\Theta_j | o_i, \Theta)$ and $P(\Theta_2 | o_i, \Theta)$. These values can be expressed by the following equation, which is a straightforward application of Bayes rule:

$$P(\Theta_j | o_i, \Theta) = \frac{w_j P(o_i | \Theta_j)}{\sum_{j=1}^k w_j P(o_i | \Theta_j)},$$

here we have only 2 distributions and each weight w is 0.5, so:

$$P(\Theta_j | o_i, \Theta) = \frac{0.5 P(o_i | \Theta_j)}{0.5 P(o_i | \Theta_1) + 0.5 P(o_i | \Theta_2)}$$

Expectation Step of EM

- For instance, assume one the points is 0. We can calculate the $p(0|\Theta_1)$ according to Gaussian density function: $P(o_i|\Theta_j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}}$, and we also know that $\Theta_1 = (-2, 2)$ and $\Theta_2 = (3, 2)$,

$$P(0|\Theta_1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(0-\mu_1)^2}{2\sigma^2}} = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(0+2)^2}{8}} = 0.12$$

$$P(0|\Theta_2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(0-\mu_2)^2}{2\sigma^2}} = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(0-3)^2}{8}} = 0.06$$

Then use $P(\Theta_j|o_i, \Theta) = \frac{0.5 P(o_i|\Theta_j)}{0.5 P(o_i|\Theta_1) + 0.5 P(o_i|\Theta_2)}$ to calculate $P(\Theta_1|0, \Theta)$ and $P(\Theta_2|0, \Theta)$:

$$P(\Theta_1|0, \Theta) = 0.12/(0.12+0.06) = 0.66$$

$$P(\Theta_2|0, \Theta) = 0.06/(0.12+0.06) = 0.33$$

Expectation Step of EM

- We can see that for point 0, $P(\Theta_1|0, \Theta) = 0.66$ and $P(\Theta_2|0, \Theta) = 0.33$, which means that the point 0 is twice likely to belong to distribution 1 as distribution 2 based on the current assumptions for the parameter values.

And in Expectation Step of EM, we need to calculate $P(\Theta_j|o_i, \Theta)$ for each object. ($P(\Theta_j|o_i, \Theta)$ means the probability that each object belongs to each distribution)

So for this case, we need to compute the cluster membership probabilities for all 20,000 points. After the calculation, the E-Step of the first iteration is over.

Maximization Step of EM

- After computing the probabilities for all 20,000 points, we need to find the new estimators for μ_1 and μ_2 in the M-Step of EM.
We use these equations:

$$\mu_j = \sum_{i=1}^n o_i \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^n P(\theta_l|o_l, \Theta)} = \frac{\sum_{i=1}^n o_i P(\theta_j|o_i, \Theta)}{\sum_{i=1}^n P(\theta_j|o_i, \Theta)}, \quad \sigma_j = \sqrt{\frac{\sum_{i=1}^n P(\theta_j|o_i, \Theta)(o_i - \mu_j)^2}{\sum_{i=1}^n P(\theta_j|o_i, \Theta)}}$$

For instance, compute the new μ_1 and μ_2 :

$$\mu_1 = \sum_{i=1}^{20,000} o_i \frac{P(\theta_1|o_i, \Theta)}{\sum_{l=1}^{20,000} P(\theta_l|o_l, \Theta)} = \frac{\sum_{i=1}^{20,000} o_i P(\theta_1|o_i, \Theta)}{\sum_{i=1}^{20,000} P(\theta_1|o_i, \Theta)} = -3.74$$

$$\mu_2 = \sum_{i=1}^{20,000} o_i \frac{P(\theta_2|o_i, \Theta)}{\sum_{l=1}^{20,000} P(\theta_l|o_l, \Theta)} = \frac{\sum_{i=1}^{20,000} o_i P(\theta_2|o_i, \Theta)}{\sum_{i=1}^{20,000} P(\theta_2|o_i, \Theta)} = 4.10$$

First few iterations of EM

Iteration	μ_1	μ_2
0	-2.00	3.00
1	-3.74	4.10
2	-3.94	4.07
3	-3.97	4.04
4	-3.98	4.03
5	-3.98	4.03

We repeat these E-Step and M-Step until the estimates of μ_1 and μ_2 either don't change or change very little.

The table above gives the first few iterations of EM algorithm when it is applied to the set of 20,000 points.

Advantages and Disadvantages of Mixture Models

- Strength
 - Mixture models are more general than partitioning and fuzzy clustering
 - Clusters can be characterized by a small number of parameters
 - The results may satisfy the statistical assumptions of the generative models
- Weakness
 - Converge to local optimal (overcome: run multi-times w. random initialization)
 - Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
 - Need large data sets
 - Hard to estimate the number of clusters

Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering
- Clustering High-Dimensional Data 
- Clustering Graphs and Network Data
- Clustering with Constraints
- Summary

Clustering High-Dimensional Data

- Clustering high-dimensional data (How high is high-D in clustering?)
 - Many applications: text documents, DNA micro-array data
 - Major challenges:
 - Many irrelevant dimensions may mask clusters
 - Distance measure becomes meaningless—due to equi-distance
 - Clusters may exist only in some subspaces
- Methods
 - **Subspace-clustering:** Search for clusters existing in subspaces of the given high dimensional data space
 - CLIQUE, ProClus, and bi-clustering approaches
 - **Dimensionality reduction approaches:** Construct a much lower dimensional space and search for clusters there (may construct new dimensions by combining some dimensions in the original data)
 - Dimensionality reduction methods and spectral clustering

Traditional Distance Measures May Not Be Effective on High-D Data

- Traditional distance measure could be dominated by noises in many dimensions
- Ex. Which pairs of customers are more similar?

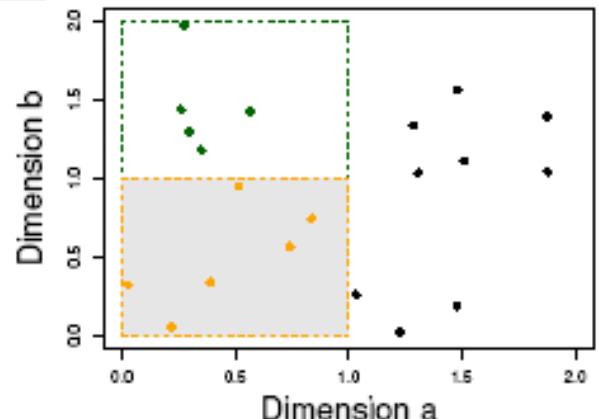
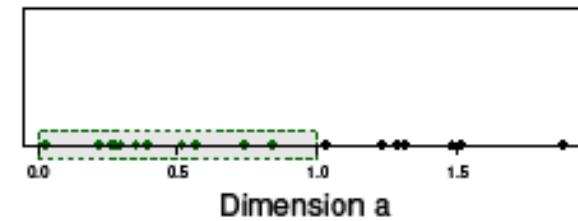
Customer	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Ada	1	0	0	0	0	0	0	0	0	0
Bob	0	0	0	0	0	0	0	0	0	1
Cathy	1	0	0	0	1	0	0	0	0	1

- By Euclidean distance, we get,
 $dist(Ada, Bob) = dist(Bob, Cathy) = dist(Ada, Cathy) = \sqrt{2}$
 - despite Ada and Cathy look more similar
- Clustering should not only consider dimensions but also attributes (features)
 - Feature transformation: effective if most dimensions are relevant (PCA & SVD useful when features are highly correlated/redundant)
 - Feature selection: useful to find a subspace where the data have nice clusters

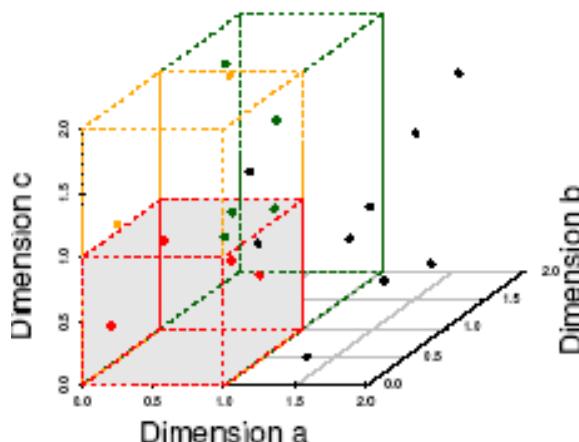
The Curse of Dimensionality

(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance



(b) 6 Objects in One Unit Bin

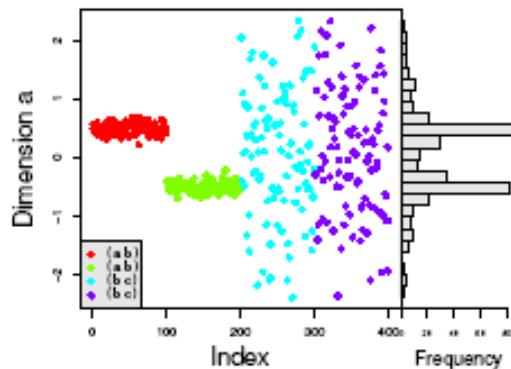
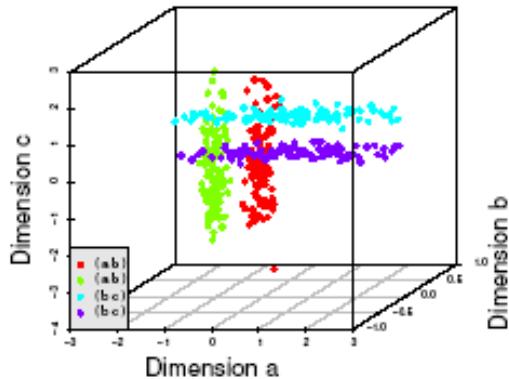


(c) 4 Objects in One Unit Bin

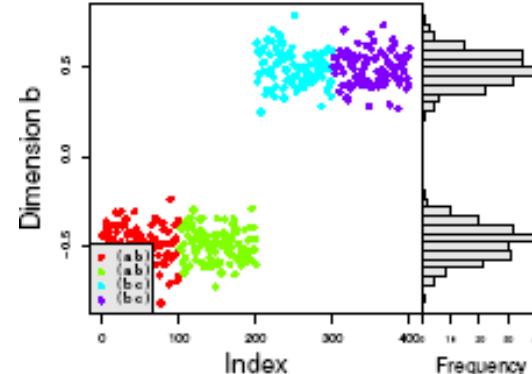
Why Subspace Clustering?

(adapted from Parsons et al. SIGKDD Explorations 2004)

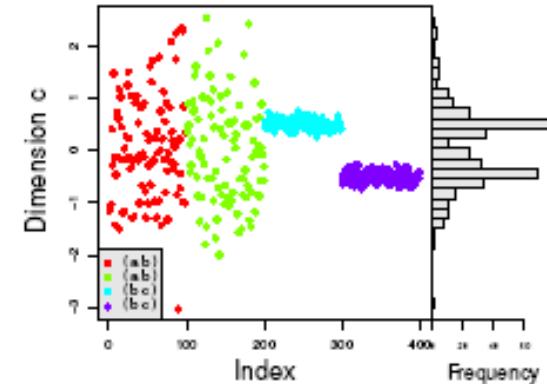
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



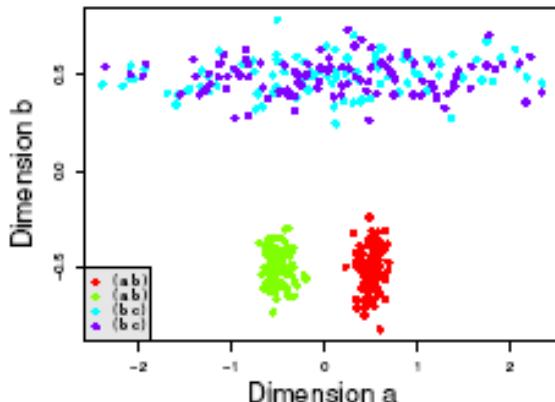
(a) Dimension a



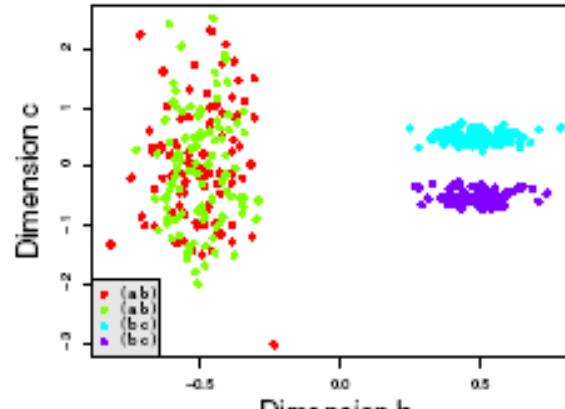
(b) Dimension b



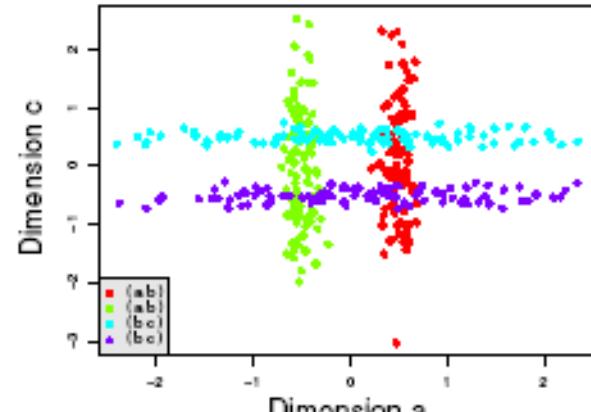
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



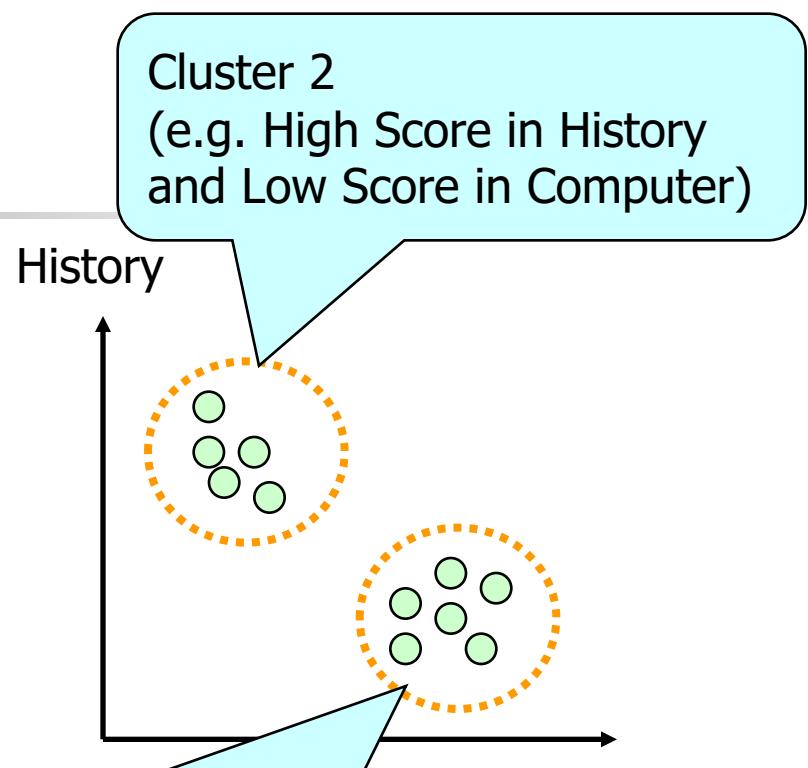
(c) Dims a & c

Subspace Clustering Methods

- Subspace search methods: Search various subspaces to find clusters
 - Bottom-up approaches
 - Top-down approaches
- Correlation-based clustering methods
 - E.g., PCA based approaches
- Bi-clustering methods
 - Optimization-based methods
 - Enumeration methods

Clustering

	Computer	History
Raymond	100	40
Louis	90	45
Wyman	20	95
...



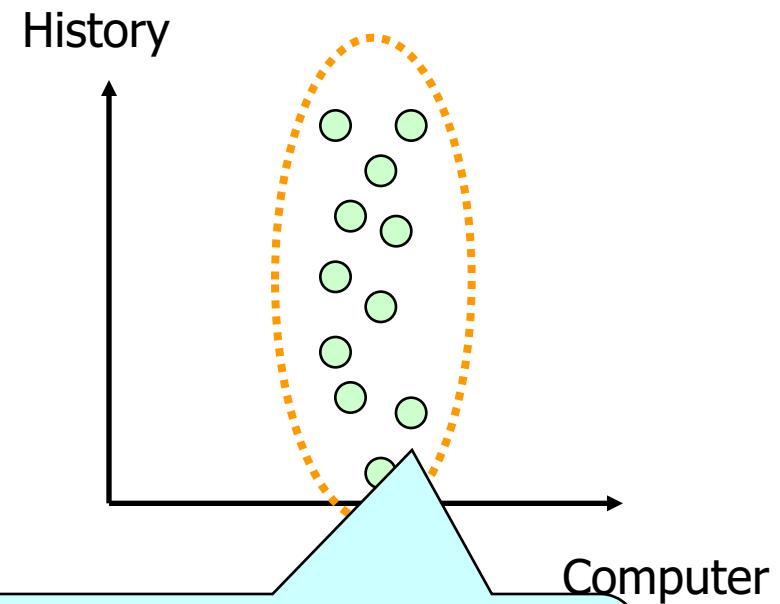
Cluster 1
(e.g. High Score in Computer
and Low Score in History)

Problem: to find all clusters

This kind of clustering considers only FULL space (i.e.
computer and history)!

Subspace Clustering

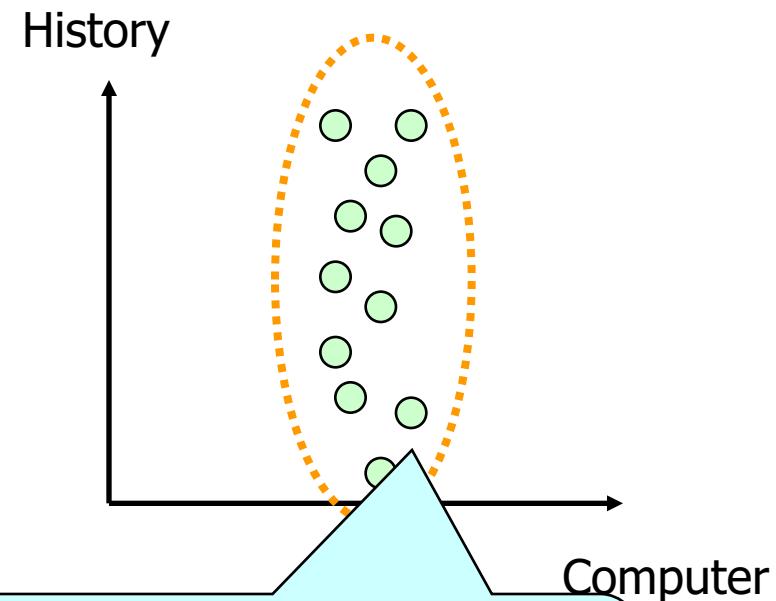
	Computer	History
Raymond	50	40
Louis	60	45
Wyman	40	95
...



Cluster 1
(e.g. Middle Score in Computer
and Any Score in History)

Subspace Clustering

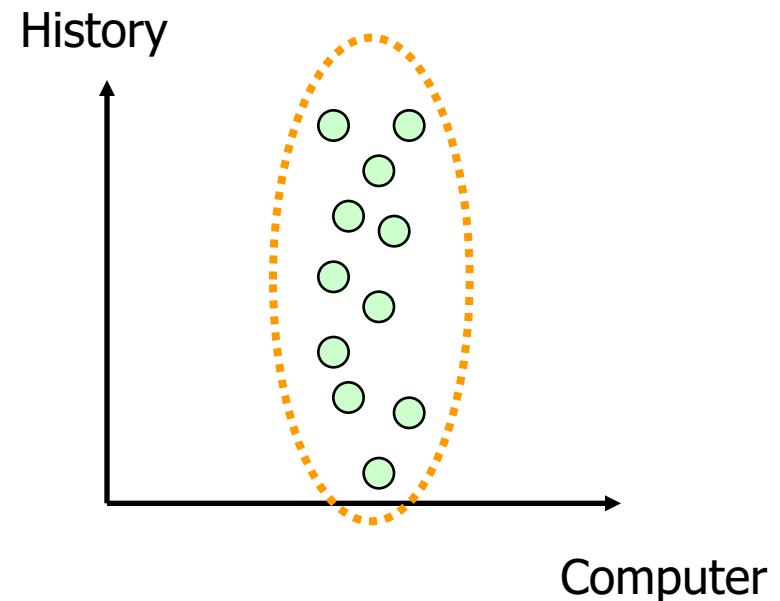
	Computer	History
Raymond	50	40
Louis	60	45
Wyman	40	95
...



Cluster 1
(e.g. Middle Score in Computer
and Low Score in History)

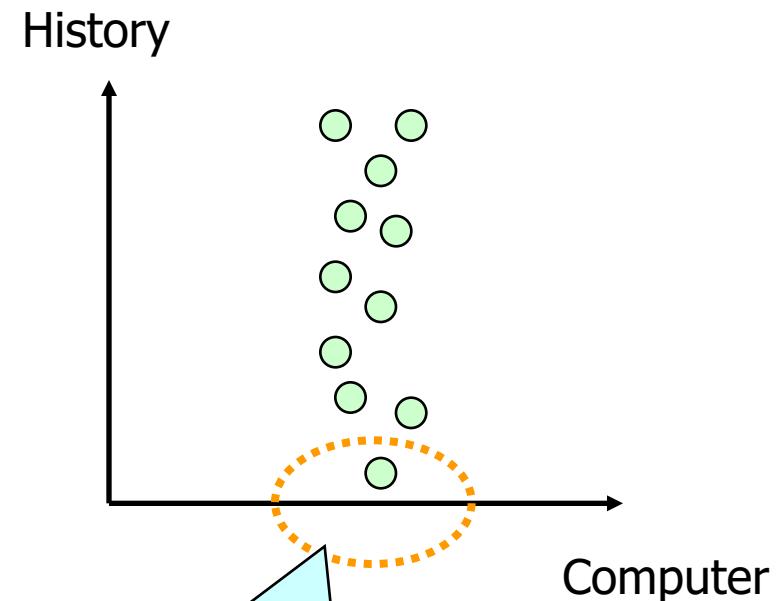
Subspace Clustering

	Computer	History
Raymond	50	40
Louis	60	45
Wyman	40	95
...



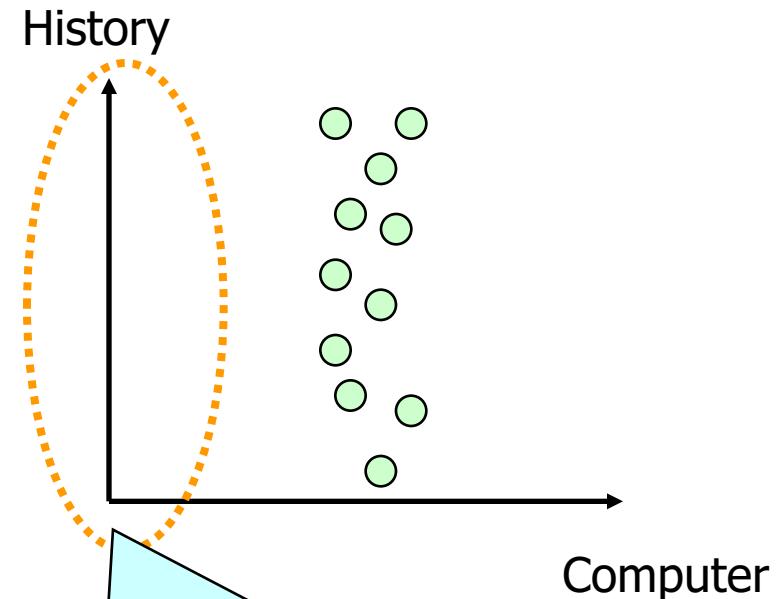
Subspace Clustering

	Computer	History
Raymond	50	40
Louis	60	45
Wyman	40	95
...



Subspace Clustering

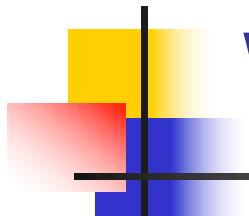
	Computer	History
Raymond	50	40
Louis	60	45
Wyman	40	95
...



No Cluster!

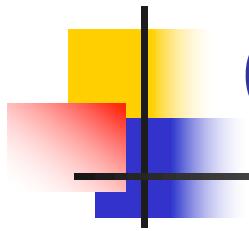
The data points span along history dimension.

Problem: to find all clusters in the subspace (i.e. some of the dimensions)



Why Subspace Clustering?

- Clustering for Understanding
 - Applications
 - Biology
 - Group different species
 - Psychology and Medicine
 - Group medicine
 - Business
 - Group different customers for marketing
 - Network
 - Group different types of traffic patterns
 - Software
 - Group different programs for data analysis



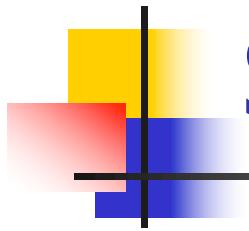
Curse of Dimensionality

- When the number of dimensions increases,
 - the distance between any two points is nearly the same



Surprising results!

This is the reason why we need to study subspace clustering



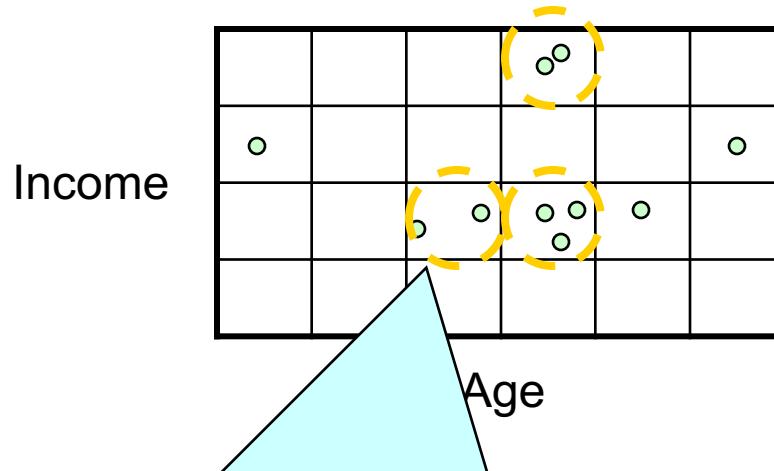
Subspace Clustering Methods

- Dense Unit-based Method
- Entropy-Based Method
- Transformation-Based Method

Dense Unit-based Method for Subspace Clustering

Density

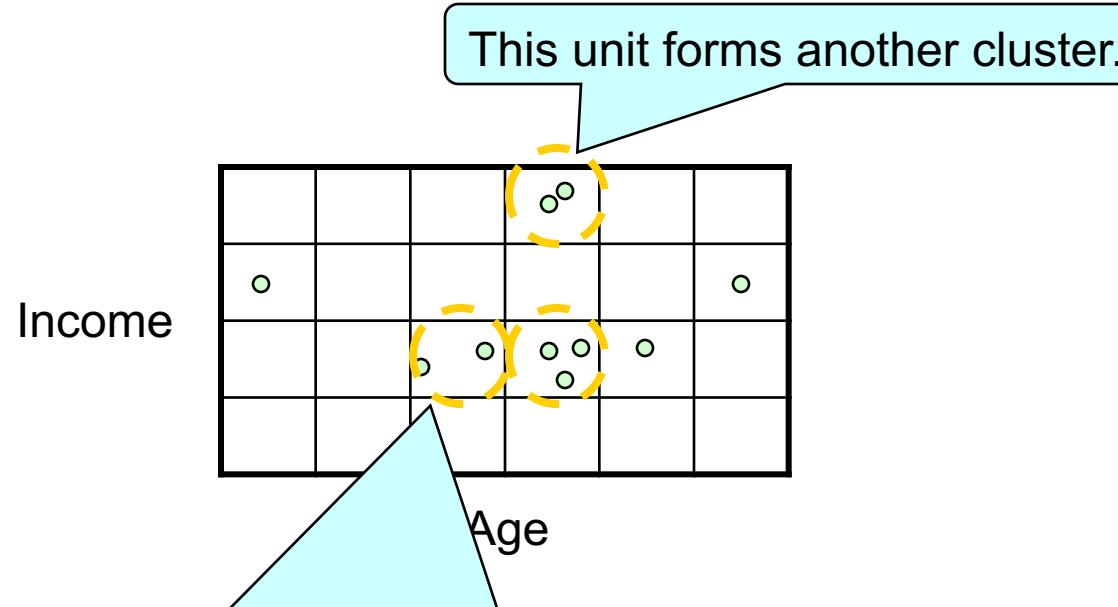
Dense unit: a unit if the fraction of data points contained in it is at least a threshold, T



If $T = 20\%$, these three units are dense.

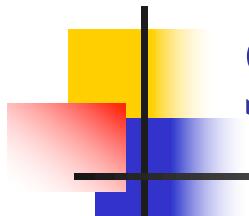
Dense Unit-based Method for Subspace Clustering

Cluster: a maximal set of connected dense units in k-dimensions



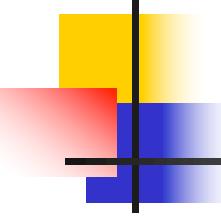
The problem is to find which **sub-spaces** contain dense units.

The second problem is to find clusters from each sub-space containing dense units



Dense Unit-based Method for Subspace Clustering

- **Step 1:** Identify sub-spaces that contain dense units
- **Step 2:** Identify clusters in each sub-spaces that contain dense units



Step 1

Suppose we want to find all dense units (e.g.,
dense units with density $\geq 20\%$)

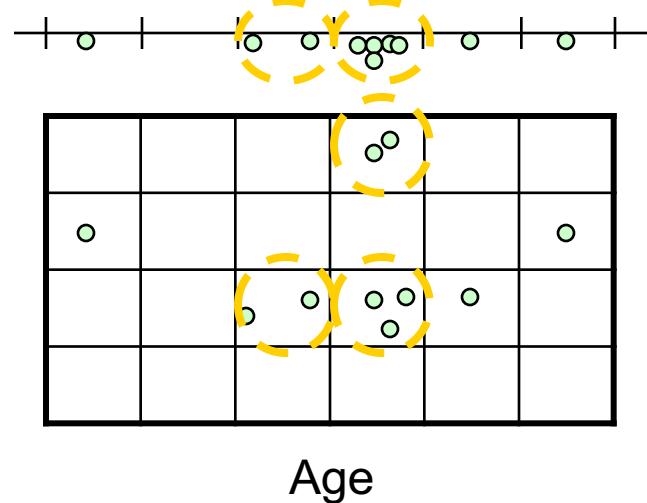
■ Property

- If a set S of points is a cluster in a k -dimensional space, then S is also part of a cluster in any $(k-1)$ -dimensional projections of the space.

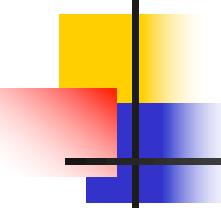
Step 1

Suppose we want to find all dense units (e.g., dense units with density $\geq 20\%$)

If $T = 20\%$, these two units are dense.



If $T = 20\%$, these three units are dense.



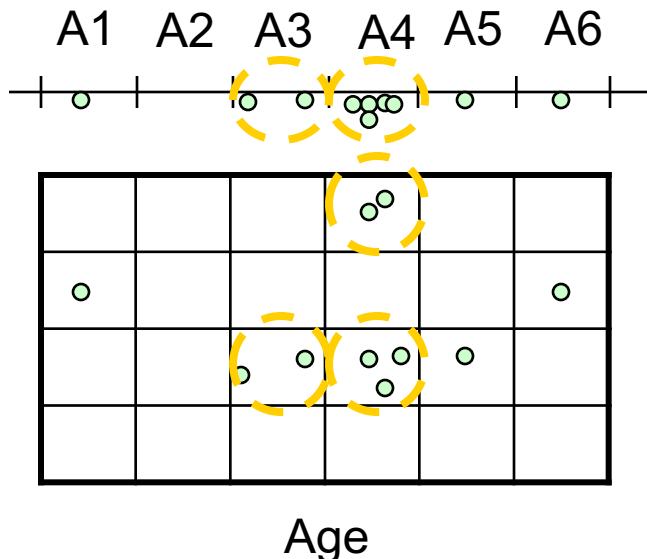
Step 1

Suppose we want to find all dense units (e.g.,
dense units with density $\geq 20\%$)

- We can make use of apriori approach to solve the problem

Step 1

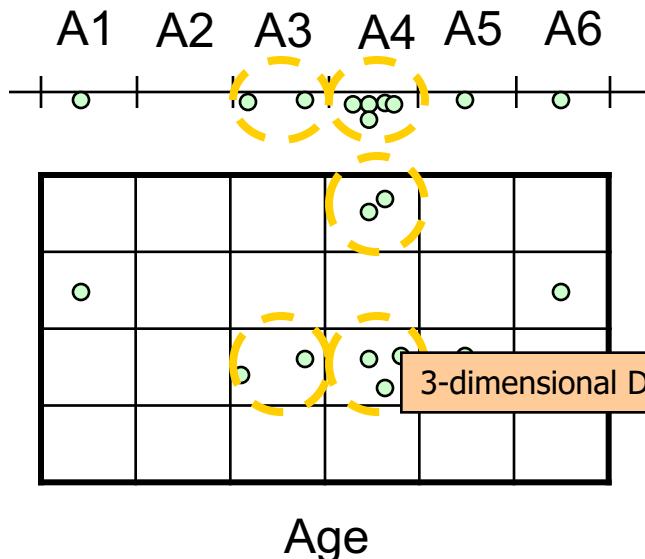
Suppose we want to find all dense units (e.g.,
dense units with density $\geq 20\%$)



With respect to dimension Age,
A3 and A4 are dense units.
With respect to dimension Income,
I1, I2 and I3 are dense units

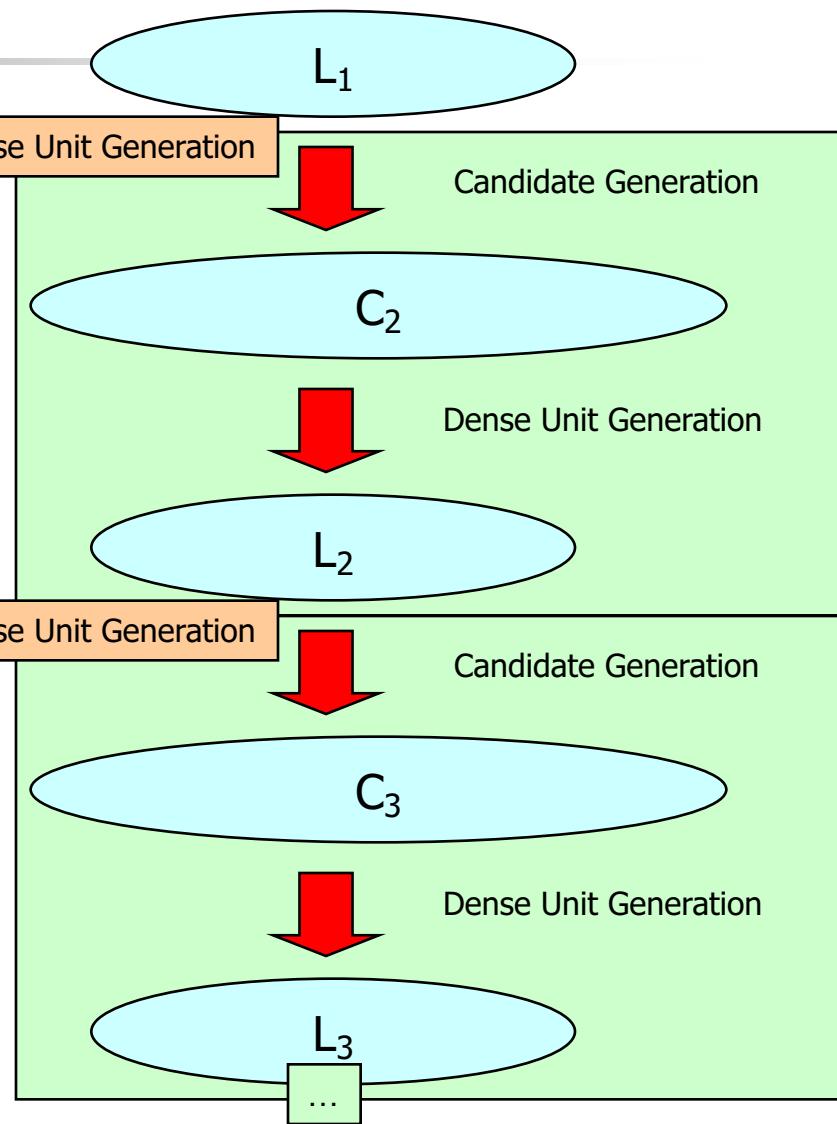
Apriori

Suppose we want to find all dense units (e.g., dense units with density $\geq 20\%$)

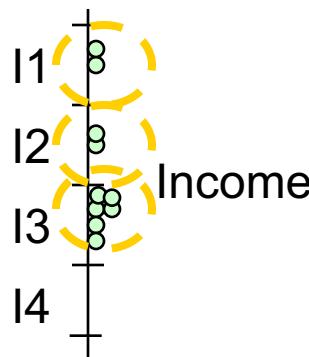


With respect to dimension Age,
A3 and A4 are dense units.

With respect to dimension Income,
I1, I2 and I3 are dense units



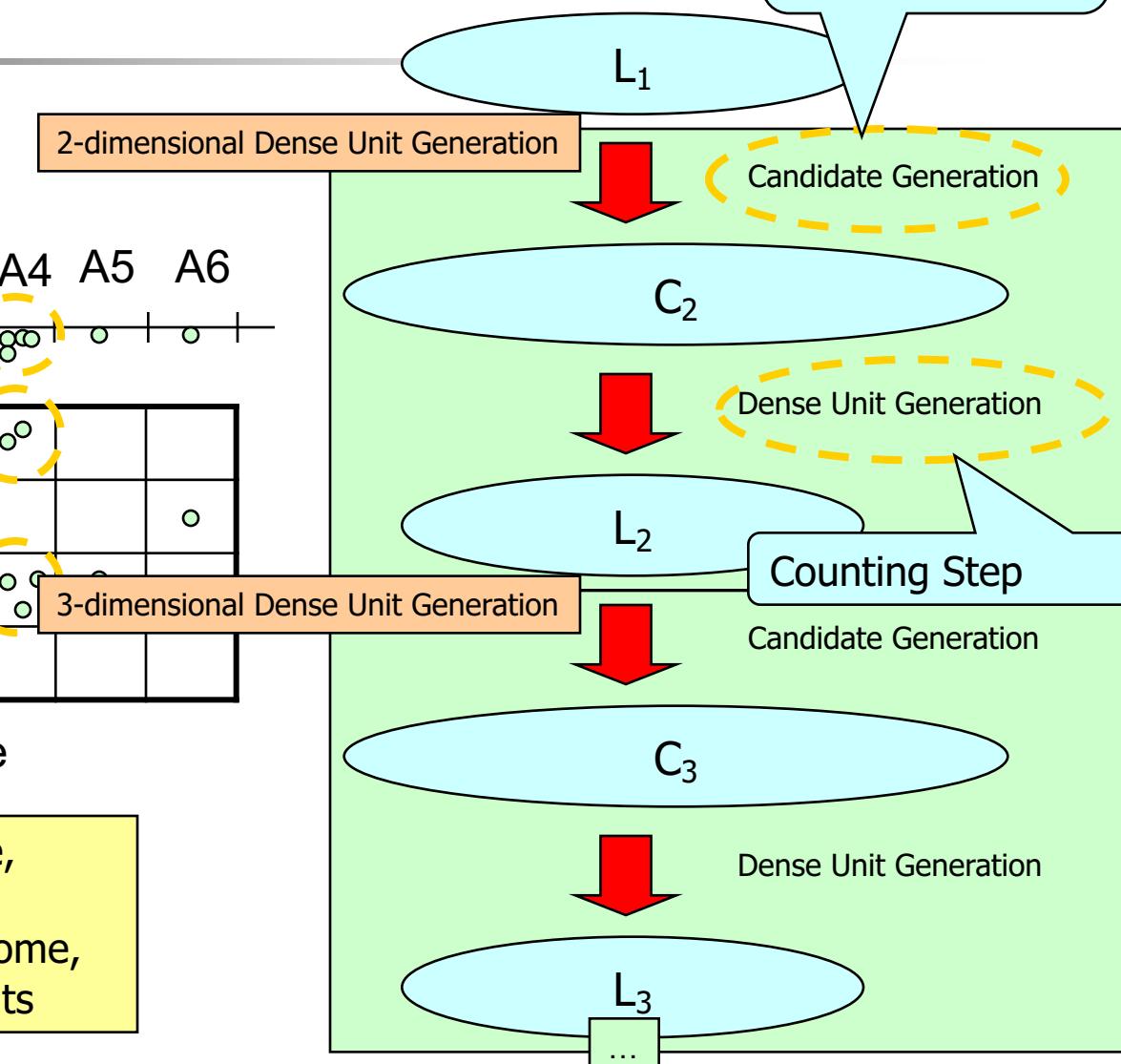
Apriori

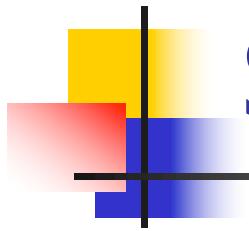


With respect to dimension Age,
A3 and A4 are dense units.
With respect to dimension Income,
I1, I2 and I3 are dense units

Suppose we want to find all dense units
(dense units with density $\geq 20\%$)

1. Join Step
2. Prune Step





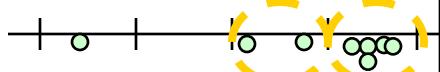
Dense Unit-based Method for Subspace Clustering

- **Step 1:** Identify sub-spaces that contain dense units
- **Step 2:** Identify clusters in each sub-spaces that contain dense units

Step 2

Suppose we want to find all dense units (e.g.,
dense units with density $\geq 20\%$)

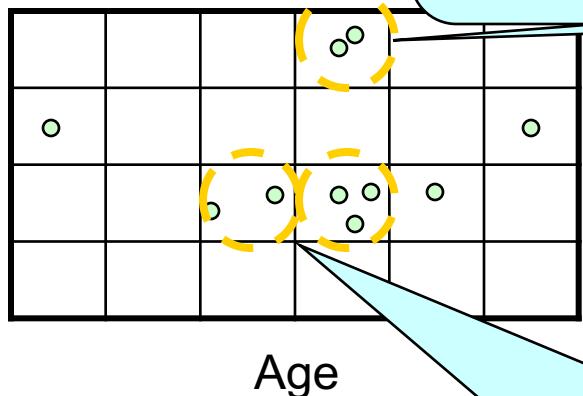
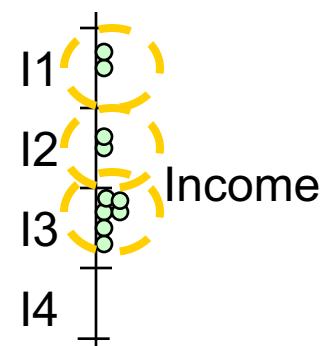
A1 A2 A3 A4



Cluster 2: A4 and I1

e.g., A4 = 21-25 and I1 = 10k-15k

Cluster 2: Age= 21-25 and Income= 10k-15k



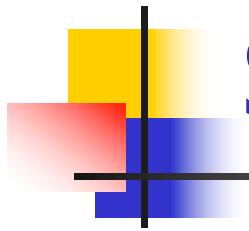
With respect to dimension Age,
A3 and A4 are dense units.

With respect to dimension Income,
I1, I2 and I3 are dense units

Cluster 1: (A3 or A4) and I3

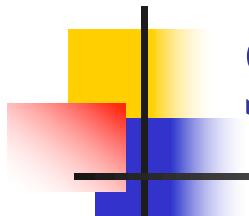
e.g., A3 = 16-20, A4 = 21-25 and I3 = 20k-25k

Cluster 1: Age=16-25 and Income=20k-25k



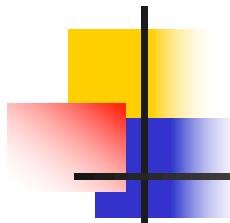
Subspace Clustering Methods

- Dense Unit-based Method
- Entropy-Based Method
- Transformation-Based Method



Entropy-Based Method for Subspace Clustering

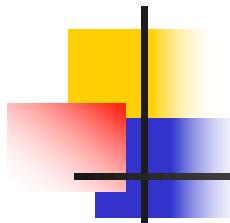
- Entropy
- Problem
 - Good subspace Clustering
- Algorithm
 - Property
 - Apriori



Entropy

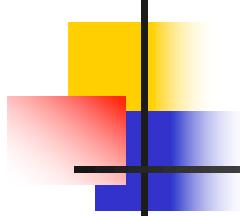
■ Example

- Suppose we have a horse race with eight horses taking part.
- Assume that the probabilities of winning for the eight horses are
- $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$



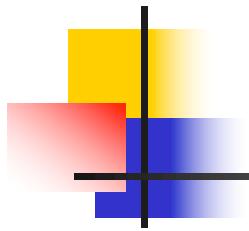
Entropy

- Suppose we want to send a message to another person indicating which horse won the race. One method is to send a 3 bit string to denote the index of the winning horse



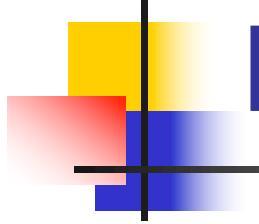
Entropy

- Another method is to use a variable length coding set (i.e. 0, 10, 110, 1110, 11110, 111101, 111110, 111111) to represent the eight horses.
- The average description length is
 - $\frac{1}{2} \log \frac{1}{2} - \frac{1}{4} \log \frac{1}{4} - \frac{1}{8} \log \frac{1}{8} -$
 - $\frac{1}{16} \log \frac{1}{16} - 4 \times \frac{1}{64} \log \frac{1}{64}$
 - = 2 bits



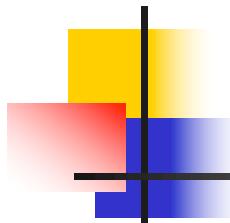
Entropy

- The **entropy** is a way to measure the amount of information.
- The smaller the entropy (viewed as the average length of description length in the above example), the more informative we have.



Entropy

- Assume that the probabilities of winning for the eight horses are
- $(1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8)$
- Entropy of the horse race:
- $H(X) = -(1/8 \log 1/8) \times 8$
= 3 bits

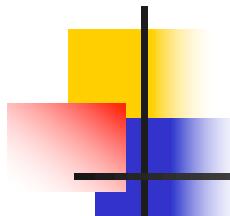


Entropy

- Assume that the probabilities of winning for the eight horses are
- (1, 0, 0, 0, 0, 0, 0, 0)
- Entropy of the horse race:
- $H(X) = - 1 \log 1 - 7 (0 \log 0)$
= 0 bits

We use the convention that $0 \log 0 = 0$

justified by continuity since $x \log x \rightarrow 0$ as $x \rightarrow 0$



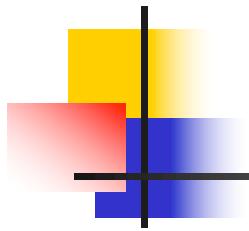
Entropy

- Let A be the set of possible outcomes of random variable X .
- Let $p(x)$ be the probability mass function of the random variable X .
- The entropy $H(X)$ of a discrete random variable X is

$$H(X) = - \sum_{x \in A} p(x) \log p(x)$$

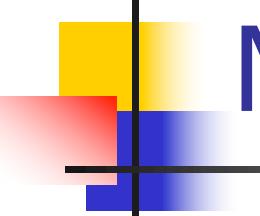
Unit: bit

con If the base of log is 2, the unit for entropy is bit.



Entropy

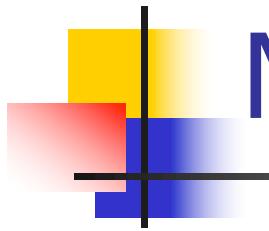
- $H(X) \geq 0$
- Because $0 \leq p(x) \leq 1$



More variables

- When there are more than one variable, we can calculate the **joint entropy** to measure their uncertainty
- X_i : the i-th random variable
- A_i : the set of possible outcomes of X_i
- Entropy:

$$H(X_1, \dots, X_n) = - \sum_{x_1 \in A_1} \dots \sum_{x_n \in A_n} p(x_1, \dots, x_n) \log p(x_1, \dots, x_n)$$



More variables

$X_1 \setminus X_2$	1	2
1	$1/4$	$1/2$
2	0	$1/4$

$$p(1, 1) = \frac{1}{4}$$

$$p(1, 2) = \frac{1}{2}$$

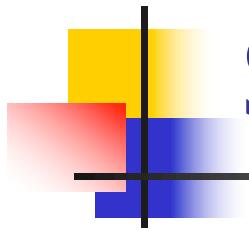
$$p(2, 1) = 0$$

$$p(2, 2) = \frac{1}{4}$$

$$H(X_1, X_2)$$

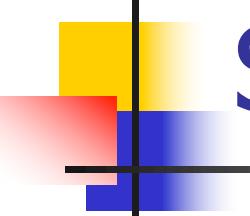
$$= -\frac{1}{4} \log \frac{1}{4} - \frac{1}{2} \log \frac{1}{2} - 0 \log 0 - \frac{1}{4} \log \frac{1}{4}$$

$$= 1.5 \text{ bits}$$



Entropy-Based Method for Subspace Clustering

- Entropy
- Problem
 - Good subspace Clustering
- Algorithm
 - Property
 - Apriori

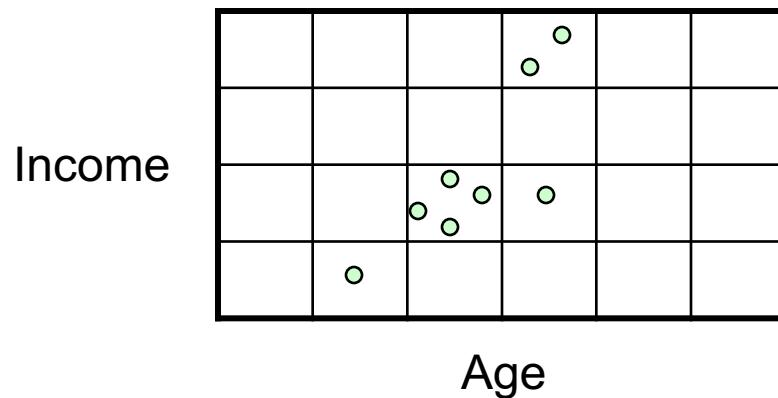


Subspace Clustering

- We divide each dimension into intervals of equal length Δ , so the subspace is partitioned into a grid.

Subspace Clustering

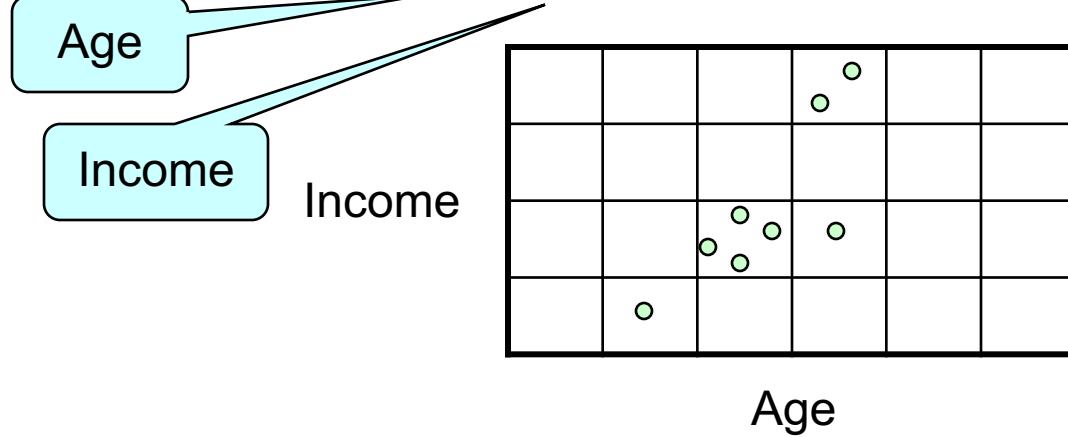
- We divide each dimension into intervals of equal length Δ , so the subspace is partitioned into a grid.

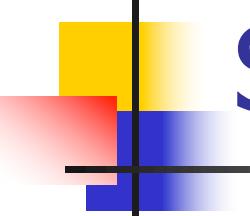


Subspace Clustering

- Let A be the set of all cells.
- $d(x)$ be the density of a cell x in terms of the percentage of data contained in x .
- We can define the entropy to be:

$$H(X_1, X_2) = - \sum_{x \in A} d(x) \log d(x)$$





Subspace Clustering

- Let A be the set of all cells.
- $d(x)$ be the density of a cell x in terms of the percentage of data contained in x .
- We can define the entropy to be:

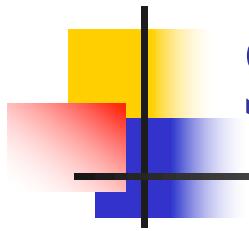
$$H(X_1, X_2) = - \sum_{x \in A} d(x) \log d(x)$$

Age

Income

Given a parameter ω ,
k dimensions (or random variables) are said
to have **good clustering** if

$$H(X_1, X_2, \dots, X_k) \leq \omega$$

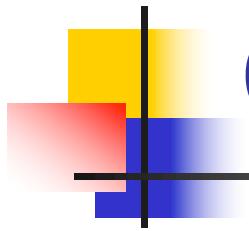


Subspace Clustering

- **Problem:** We want to find all subspaces with good clustering.

e.g., we want to find sub-spaces with entropy $H(X_1, X_2, \dots, X_k) \leq \omega$

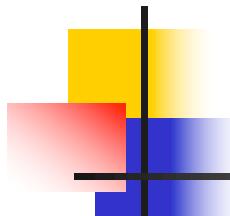
Given a parameter ω ,
k dimensions (or random variables) are said
to have **good clustering** if
$$H(X_1, X_2, \dots, X_k) \leq \omega$$



Conditional Entropy

- The conditional entropy $H(Y|X)$ is defined as

$$H(Y|X) = \sum_{x \in A} p(x)H(Y|X = x)$$



Conditional Entropy

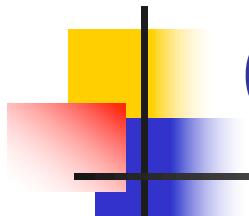
X\Y	1	2
1	0	$\frac{3}{4}$
2	$\frac{1}{8}$	$\frac{1}{8}$

$$H(Y|X=1) = 0 \text{ bit}$$

$$H(Y|X=2) = 1 \text{ bit}$$

$$\begin{aligned} H(Y|X) &= \frac{3}{4} \times H(Y|X=1) + \frac{1}{4} \times H(Y|X=2) \\ &= 0.25 \text{ bit} \end{aligned}$$

$$H(Y|X) = - \sum_{x \in A} \sum_{y \in B} p(x, y) \log p(y|x)$$



Conditional Entropy

- A : a set of possible outcomes of random variable X
- B : a set of possible outcomes of random variable Y
- $H(Y|X) = - \sum_{x \in A} \sum_{y \in B} p(x, y) \log p(y|x)$

$$H(Y|X) = - \sum_{x \in A} \sum_{y \in B} p(x, y) \log p(y|x)$$

Conditional Entropy

X\Y	1	2
1	0	3/4
2	1/8	1/8

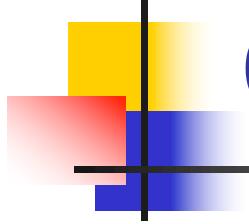
$$p(Y = 1 | X= 1) = 0$$

$$p(Y = 2 | X= 1) = 1$$

$$p(Y = 1 | X= 2) = \frac{1}{2}$$

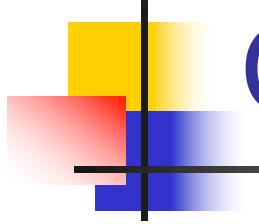
$$p(Y = 2 | X= 2) = \frac{1}{2}$$

$$\begin{aligned} H(Y|X) &= - 0 \log 0 - \frac{3}{4} \log 1 - \frac{1}{8} \log \frac{1}{2} - \frac{1}{8} \log \frac{1}{2} \\ &= 0.25 \text{ bit} \end{aligned}$$



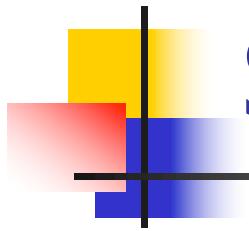
Chain Rule

- $H(X, Y) = H(X) + H(Y | X)$



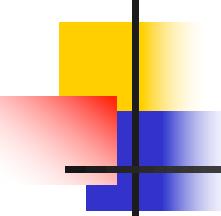
Chain Rule

- $H(X, Y) = H(X) + H(Y | X)$
- $$\begin{aligned} H(X_1, \dots, X_{k-1}, X_k) \\ = H(X_1, \dots, X_{k-1}) + H(X_k | X_1, \dots, X_{k-1}) \end{aligned}$$



Entropy-Based Method for Subspace Clustering

- Entropy
- Problem
 - Good subspace Clustering
- Algorithm
 - Property
 - Apriori



Property

Given a parameter ω ,
k dimensions (or random variables) are said
to have **good clustering** if

$$H(X_1, X_2, \dots, X_k) \leq \omega$$

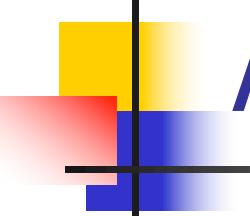
- **Lemma:** If a k-dimensional subspace X_1, \dots, X_k has good clustering, then each of the $(k-1)$ -dimensional projections of this space has also good clustering.

Proof: Since the subspace X_1, \dots, X_k has good clustering,

$$H(X_1, \dots, X_k) \leq \omega$$

Consider a $(k-1)$ -dimensional projections, say X_1, \dots, X_{k-1} :

$$\begin{aligned} H(X_1, \dots, X_{k-1}) &\leq H(X_1, \dots, X_{k-1}) + H(X_k | X_1, \dots, X_{k-1}) \\ &= H(X_1, \dots, X_k) \\ &\leq \omega \end{aligned}$$



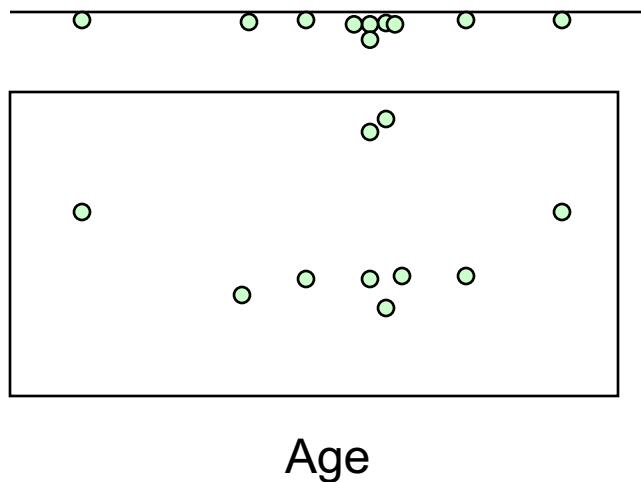
Apriori

Suppose we want to find sub-spaces with entropy ≤ 0.2

- We can make use of apriori approach to solve the problem

Apriori

Suppose we want to find sub-spaces with entropy ≤ 0.2



e.g.,
 $H(\text{Age}) = 0.12$
 $H(\text{Income}) = 0.08$

Apriori

Suppose we want to find sub-spaces with entropy ≤ 0.2

Income



Age

e.g.,
 $H(\text{Age}) = 0.12$
 $H(\text{Income}) = 0.08$

COMP5331

Size 2 subspace Generation

L_1

Candidate Generation

C_2

Good Subspace Generation

L_2

Size 3 subspace Generation

C_3

Candidate Generation

L_3

Good Subspace Generation

...

Apriori

Suppose we want to find sub-spaces
entropy ≤ 0.2

1. Join Step
2. Prune Step

Income

Age

e.g.,

$$H(\text{Age}) = 0.12$$

$$H(\text{Income}) = 0.08$$

COMP5331

Size 2 subspace Generation

Size 3 subspace Generation

L_1

C_2

L_2

C_3

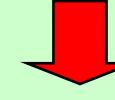
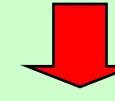
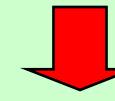
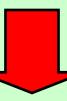
L_3

Counting Step

Candidate Generation

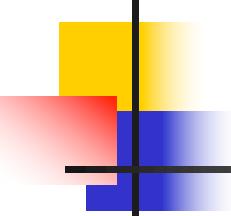
Good Subspace Generation

...



Candidate Generation

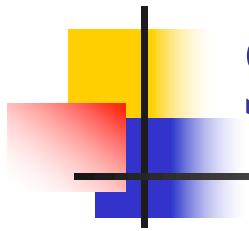
Good Subspace Generation



Cluster

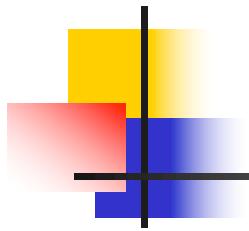
Suppose we want to find sub-spaces with entropy ≤ 0.2

- After finding the subspaces with entropy ≤ 0.2 ,
- We can find the real clusters by existing methods (e.g., k-mean) in each of the subspaces found.



Subspace Clustering Methods

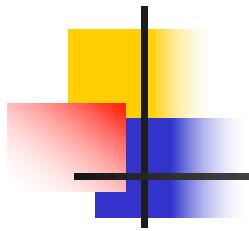
- Dense Unit-based Method
- Entropy-Based Method
- Transformation-Based Method



KL-Transform

Karhunen-Loeve Transform

- The two previous approaches find the sub-space in the original dimensions
- KL-Transform “transforms” the data points from the original dimensions into other dimensions



KL-Transform

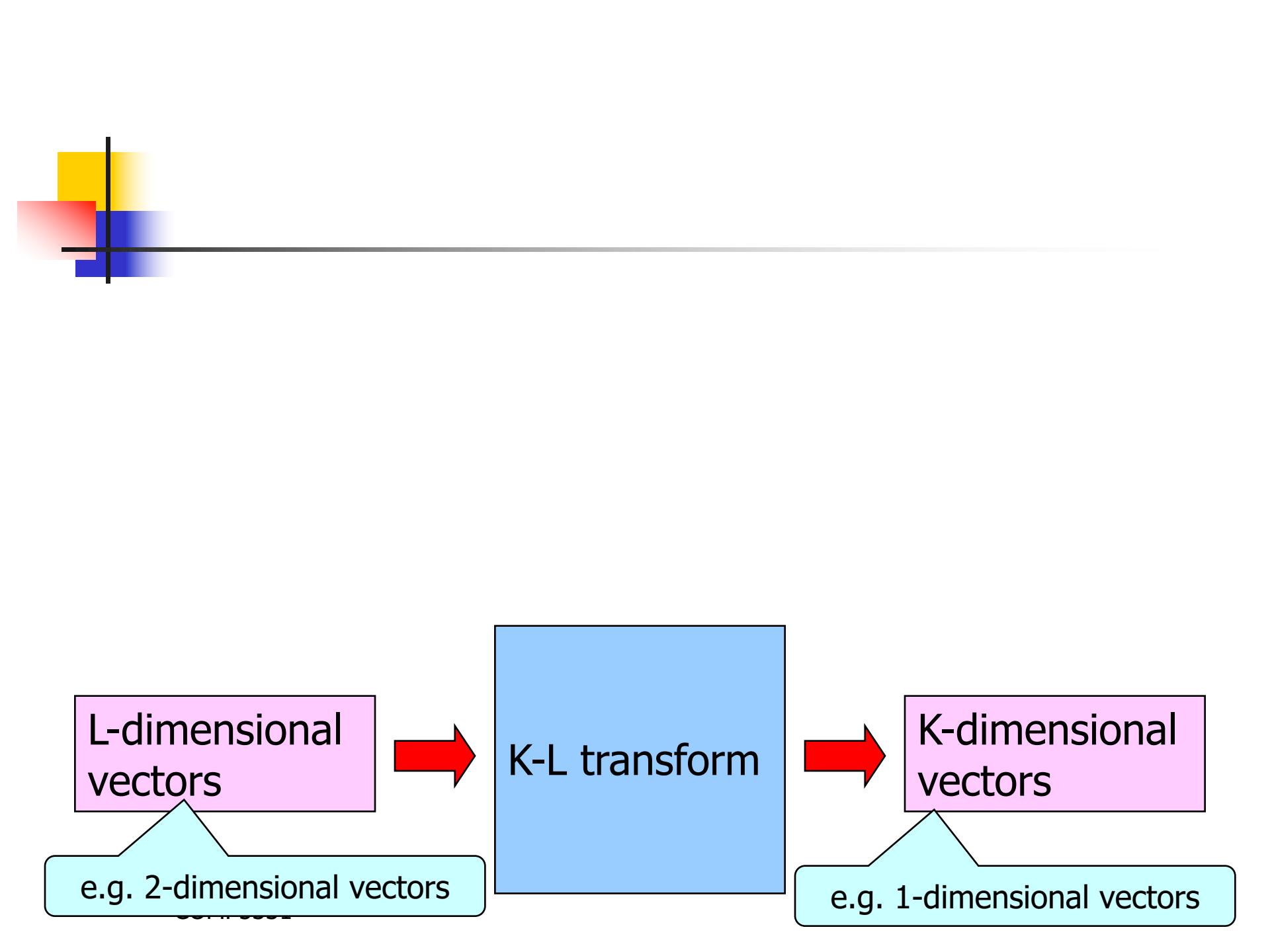
L-dimensional
vectors

e.g. 2-dimensional vectors

K-L transform

K-dimensional
vectors

e.g. 1-dimensional vectors





Step 1

For each dimension i ,
calculate the mean e_i (expected value)
For each L-dimensional data $\{x_1, x_2, \dots, x_L\}$
find $\{x_1 - e_1, x_2 - e_2, \dots, x_L - e_L\}$

Step 2

Obtain the covariance matrix Σ

Step 3

Find the eigenvalues and eigenvectors of Σ
Choose the eigenvectors of unit lengths

Step 4

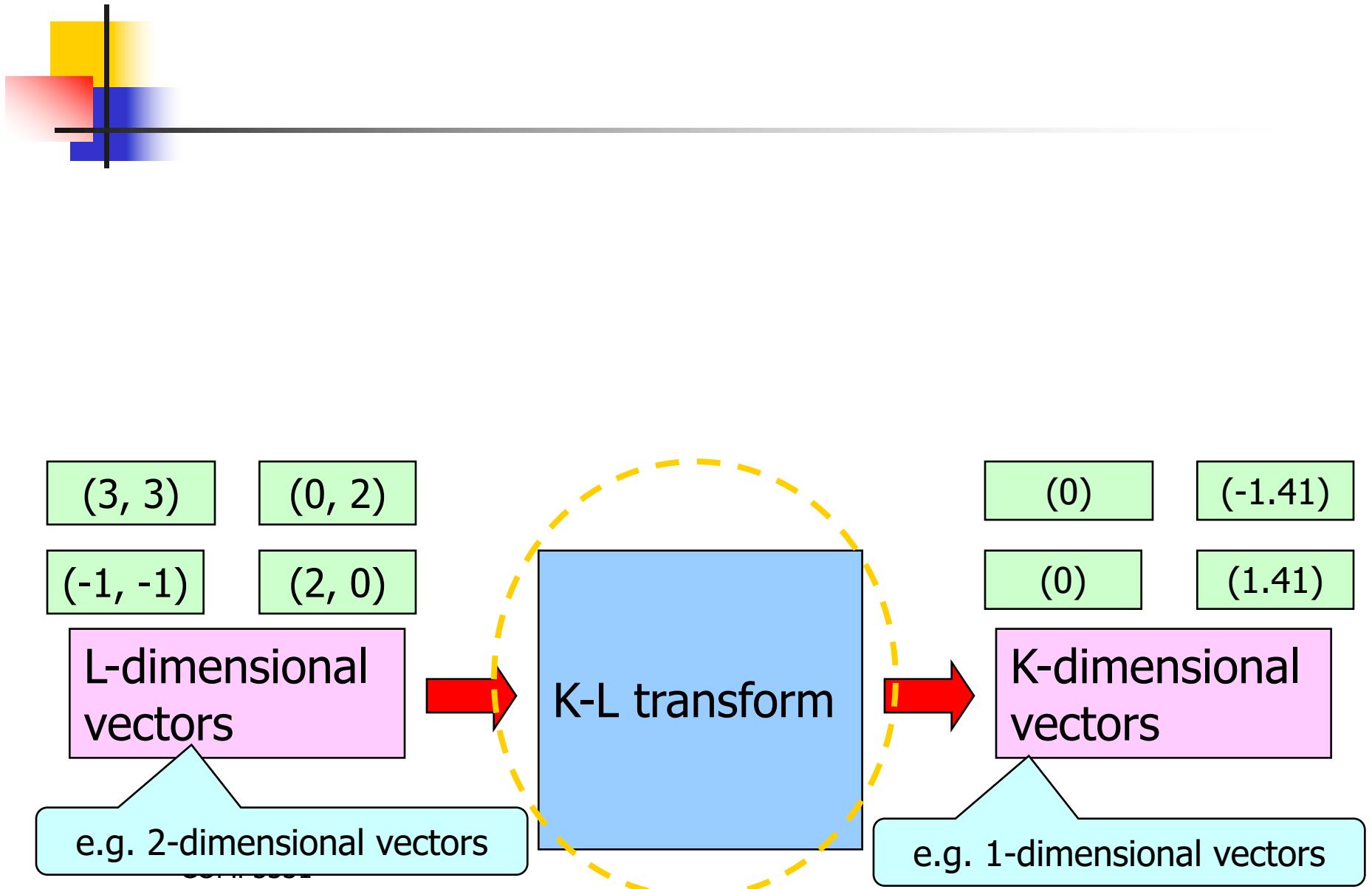
Arrange the eigenvectors in descending order of the eigenvalues

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

Step 6

For each “transformed” L-dimensional vector, keep only the K values $\{y_1, y_2, \dots, y_K\}$ corresponding to the smallest K eigenvalues.



2-dimensional vectors

K-L transform

1-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 1

For each dimension i ,
calculate the mean e_i (expected value)
For each L-dimensional data $\{x_1, x_2, \dots, x_L\}$
find $\{x_1 - e_1, x_2 - e_2, \dots, x_L - e_L\}$

For **dimension 1**,

$$\text{mean} = (3 + 0 + (-1) + 2)/4 = 4/4 = 1$$

For **dimension 2**,

$$\text{mean} = (3 + 2 + (-1) + 0)/4 = 4/4 = 1$$

$$\text{mean vector} = (1, 1)$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 1

For each dimension i ,
calculate the mean e_i (expected value)
For each L-dimensional data $\{x_1, x_2, \dots, x_L\}$
find $\{x_1 - e_1, x_2 - e_2, \dots, x_L - e_L\}$

For **dimension 1**,

$$\text{mean} = (3 + 0 + (-1) + 2)/4 = 4/4 = 1$$

For **dimension 2**,

$$\text{mean} = (3 + 2 + (-1) + 0)/4 = 4/4 = 1$$

$$\text{mean vector} = (1, 1)$$

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 1

For each dimension i ,
calculate the mean e_i (expected value)

For each L-dimensional data $\{x_1, x_2, \dots, x_L\}$
find $\{x_1-e_1, x_2-e_2, \dots, x_L-e_L\}$

mean vector = (1, 1)

For **data 1**, (3, 3)

Difference from mean vector = $(3-1, 3-1) = (2, 2)$

For **data 2**, (0, 2)

Difference from mean vector = $(0-1, 2-1) = (-1, 1)$

For **data 3**, (-1, -1)

Difference from mean vector = $(-1-1, -1-1) = (-2, -2)$

For **data 4**, (2, 0)

Difference from mean vector = $(2-1, 0-1) = (1, -1)$

COMP

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 1

mean vector = (1, 1)

For **data 1**, (3, 3)

Difference from mean vector = (2, 2)

For **data 2**, (0, 2)

Difference from mean vector = (-1, 1)

For **data 3**, (-1, -1)

Difference from mean vector = (-2, -2)

For **data 4**, (2, 0)

Difference from mean vector = (1, -1)

COMP

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 1

mean vector = (1, 1)

For **data 1**, Difference from mean vector = (2, 2)

For **data 2**, Difference from mean vector = (-1, 1)

For **data 3**, Difference from mean vector = (-2, -2)

For **data 4**, Difference from mean vector = (1, -1)

Step 2

Obtain the covariance matrix Σ

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

$$\Sigma = \frac{1}{4} YY^T =$$

$$\frac{1}{4} \begin{pmatrix} 2 & -1 & -2 & 1 \\ 2 & 1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ -1 & 1 \\ -2 & -2 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 1

mean vector = (1, 1)

For **data 1**, Difference from mean vector = (2, 2)

For **data 2**, Difference from mean vector = (-1, 1)

For **data 3**, Difference from mean vector = (-2, -2)

For **data 4**, Difference from mean vector = (1, -1)

Step 2

$Y =$

$$\begin{pmatrix} 2 & -1 & -2 & 1 \\ 2 & 1 & -2 & -1 \end{pmatrix}$$

$\Sigma =$

$$\begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 1 mean vector = (1, 1)

For **data 1**, Difference from mean vector = (2, 2)

For **data 2**, Difference from mean vector = (-1, 1)

For **data 3**, Difference from mean vector = (-2, -2)

For **data 4**, Difference from mean vector = (1, -1)

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

Find the eigenvalues and eigenvectors of Σ
Choose the eigenvectors of unit lengths

$$\begin{vmatrix} 5/2 - \lambda & 3/2 \\ 3/2 & 5/2 - \lambda \end{vmatrix} = 0$$
$$(5/2 - \lambda)^2 - (3/2)^2 = 0$$

$$25/4 - 5\lambda + \lambda^2 - 9/4 = 0$$

$$4 - 5\lambda + \lambda^2 = 0$$

$$(\lambda - 4)(\lambda - 1) = 0$$

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\begin{vmatrix} 5/2 - \lambda & 3/2 \\ 3/2 & 5/2 - \lambda \end{vmatrix} = 0$$

$$(5/2 - \lambda)^2 - (3/2)^2 = 0$$

$$25/4 - 5\lambda + \lambda^2 - 9/4 = 0$$

$$4 - 5\lambda + \lambda^2 = 0$$

$$(\lambda - 4)(\lambda - 1) = 0$$

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} 5/2-4 & 3/2 \\ 3/2 & 5/2-4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} -3/2 & 3/2 \\ 3/2 & -3/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x_1 - x_2 = 0 \\ x_1 = x_2$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a \\ a \end{pmatrix} \quad \text{where } a \in R$$

We choose the eigenvector of unit length

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

2-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

When $\lambda = 1$

$$\begin{pmatrix} 5/2-1 & 3/2 \\ 3/2 & 5/2-1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &= 0 \\ x_1 &= -x_2 \end{aligned}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a \\ -a \end{pmatrix} \quad \text{where } a \in R$$

We choose the eigenvector of unit length

$$\begin{pmatrix} 3/2 & 3/2 \\ 3/2 & 3/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ -\sqrt{1/2} \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

When $\lambda = 1$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ -\sqrt{1/2} \end{pmatrix}$$

2-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional
vectors

Step 2

$$\Sigma = \begin{pmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{pmatrix}$$

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

When $\lambda = 1$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ -\sqrt{1/2} \end{pmatrix}$$

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 3

$$\lambda = 4 \quad \text{or} \quad \lambda = 1$$

When $\lambda = 4$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ \sqrt{1/2} \end{pmatrix}$$

When $\lambda = 1$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1/2} \\ -\sqrt{1/2} \end{pmatrix}$$

Step 4

Arrange the eigenvectors in descending order of the eigenvalues

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

2-dimensional vectors	
(3, 3)	(0, 2)
(-1, -1)	(2, 0)



1-dimensional vectors

Step 4

Arrange the eigenvectors in descending order of the eigenvalues

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

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 4

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

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

$$Y = \Phi^T X$$

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 4

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

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

2-dimensional
vectors

K-L transform

1-dimensional
vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

Step 4

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

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

2-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional vectors

Step 4

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

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

For **data 1**, (3, 3)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} 3 \\ 3 \end{pmatrix} = \begin{pmatrix} 4.24 \\ 0 \end{pmatrix}$$

For **data 3**, (-1, -1)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1.41 \\ 0 \end{pmatrix}$$

For **data 2**, (0, 2)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1.41 \\ -1.41 \end{pmatrix}$$

For **data 4**, (2, 0)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.41 \\ 1.41 \end{pmatrix}$$

2-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional vectors

Step 4

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

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

For **data 1**,

(3, 3)

$$= \begin{pmatrix} 4.24 \\ 0 \end{pmatrix}$$

For **data 3**,

(-1, -1)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1.41 \\ 0 \end{pmatrix}$$

For **data 2**,

(0, 2)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1.41 \\ -1.41 \end{pmatrix}$$

For **data 4**,

(2, 0)

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.41 \\ 1.41 \end{pmatrix}$$

2-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional vectors

Step 4

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

$$Y = \begin{pmatrix} \sqrt{1/2} & \sqrt{1/2} \\ \sqrt{1/2} & -\sqrt{1/2} \end{pmatrix} X$$

Step 5

Transform the given L-dimensional vectors by eigenvector matrix

For **data 1**, (3, 3)

(4.24, 0)

For **data 2**, (0, 2)

(1.41, -1.41)

For **data 3**, (-1, -1)

(-1.41, 0)

For **data 4**, (2, 0)

(1.41, 1.41)

2-dimensional vectors

(3, 3)	(0, 2)
(-1, -1)	(2, 0)

K-L transform

1-dimensional vectors

Step 6

For each “transformed” L-dimensional vector, keep only the K values $\{y_1, y_2, \dots, y_K\}$ corresponding to the smallest k eigenvalues.

For **data 1**,

(3, 3)



(4.24, 0)



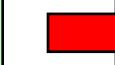
(0)

For **data 2**,

(0, 2)



(1.41, -1.41)



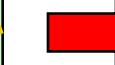
(-1.41)

For **data 3**,

(-1, -1)



(-1.41, 0)



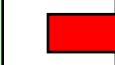
(0)

For **data 4**,

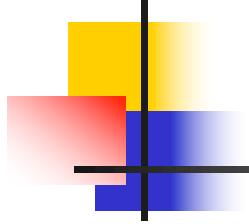
(2, 0)



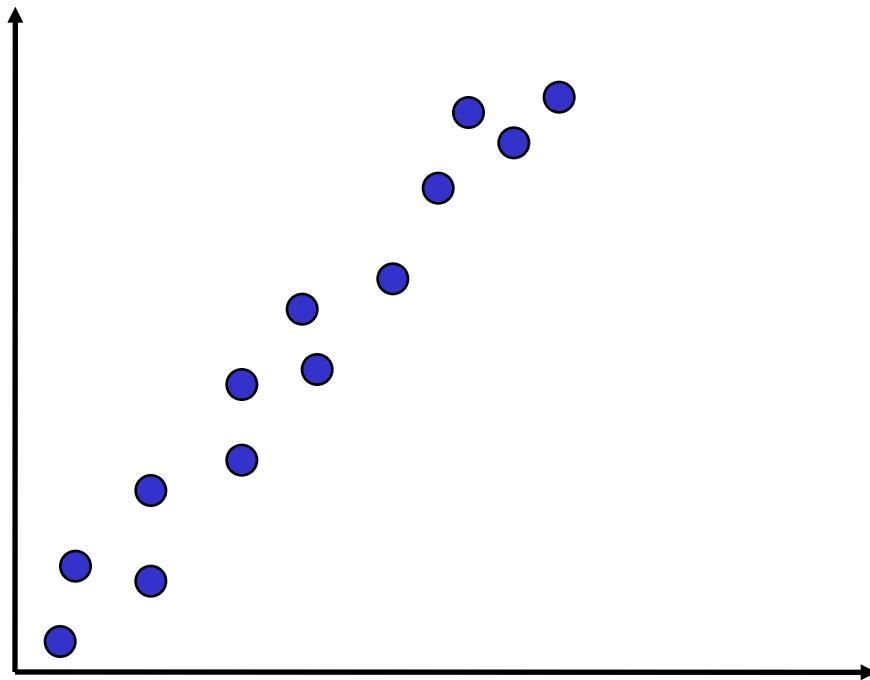
(1.41, 1.41)



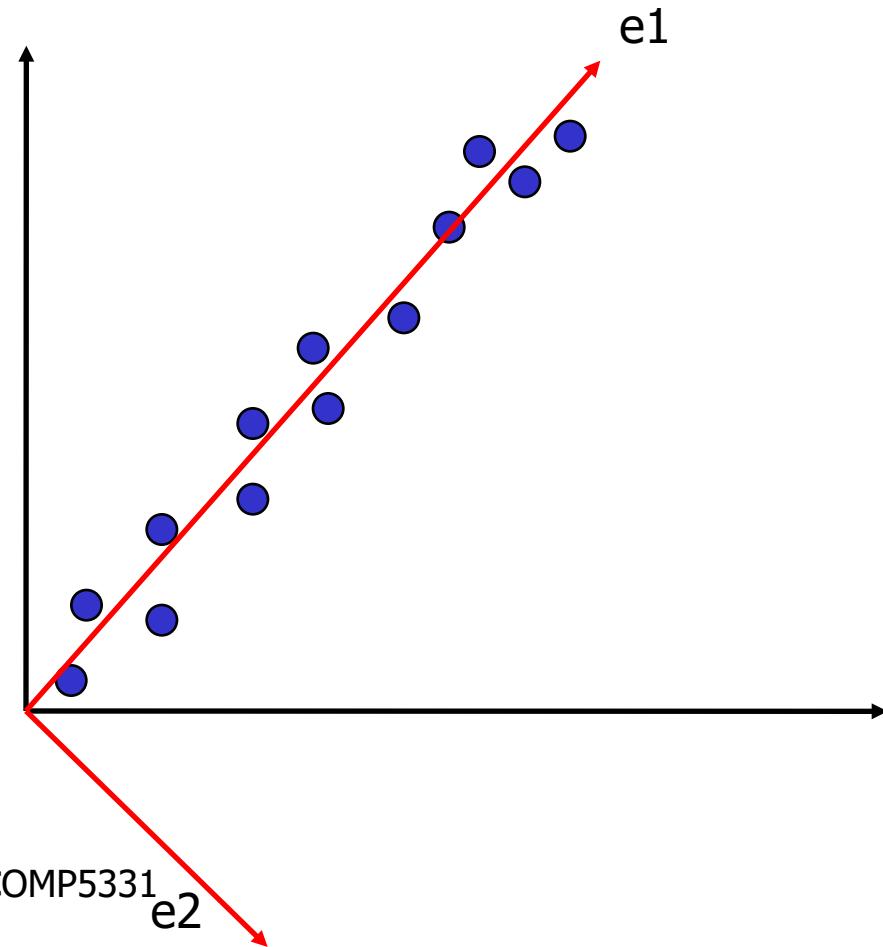
(1.41)

- 
- Why do we need to do this KL-transform?
 - Why do we choose the eigenvectors with the smallest eigenvalues?

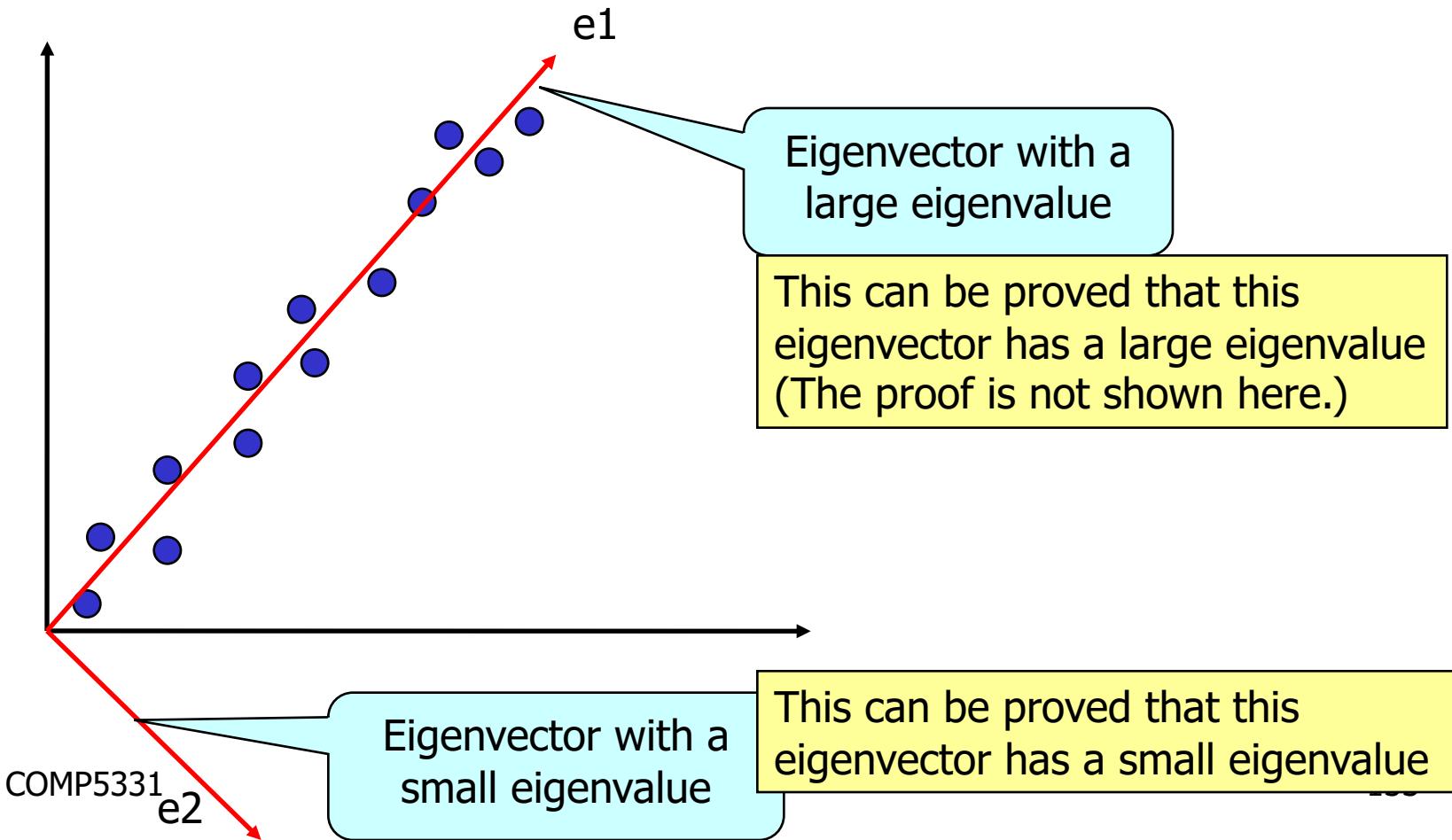
Suppose we have the
following data set

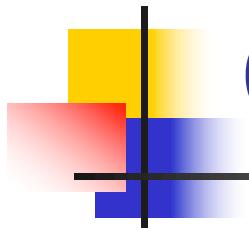


According to the data, we find the following eigenvectors (marked in red)



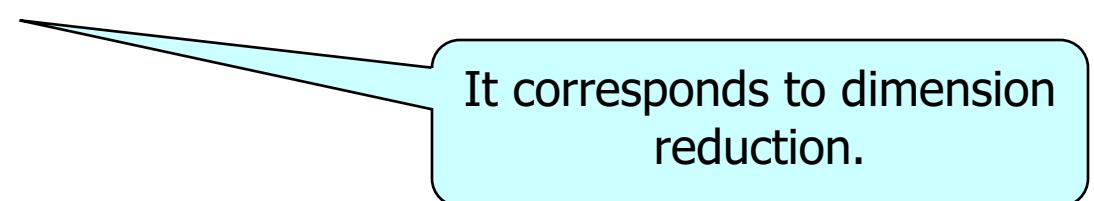
According to the data, we find the following eigenvectors (marked in red)



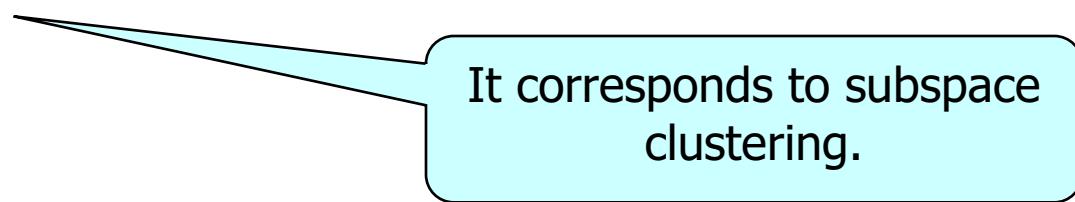


Consider two cases

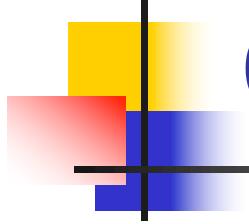
- Case 1
 - Consider that the data points are projected on e_1
- Case 2
 - Consider that the data points are projected on e_2



It corresponds to dimension reduction.



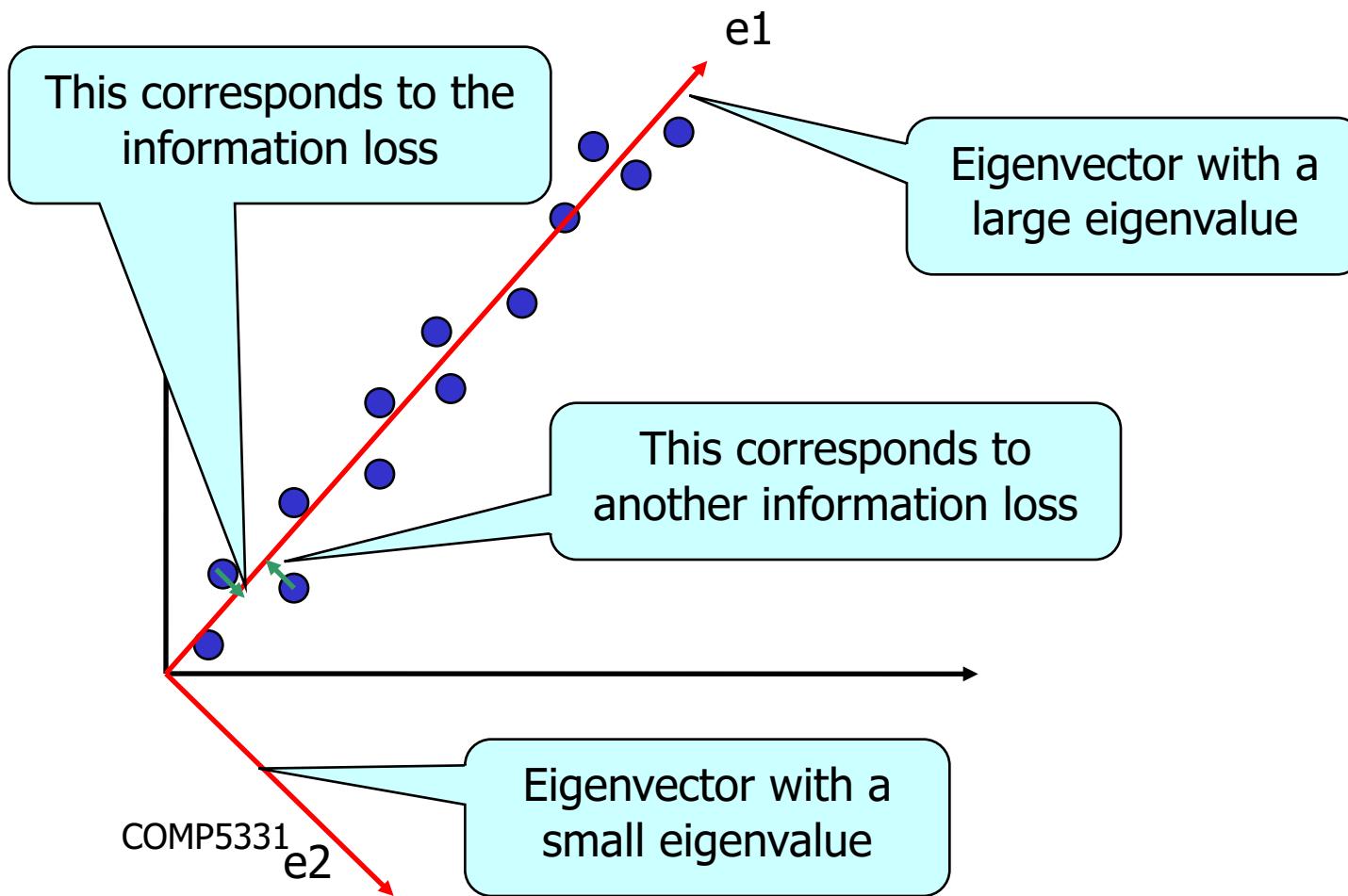
It corresponds to subspace clustering.



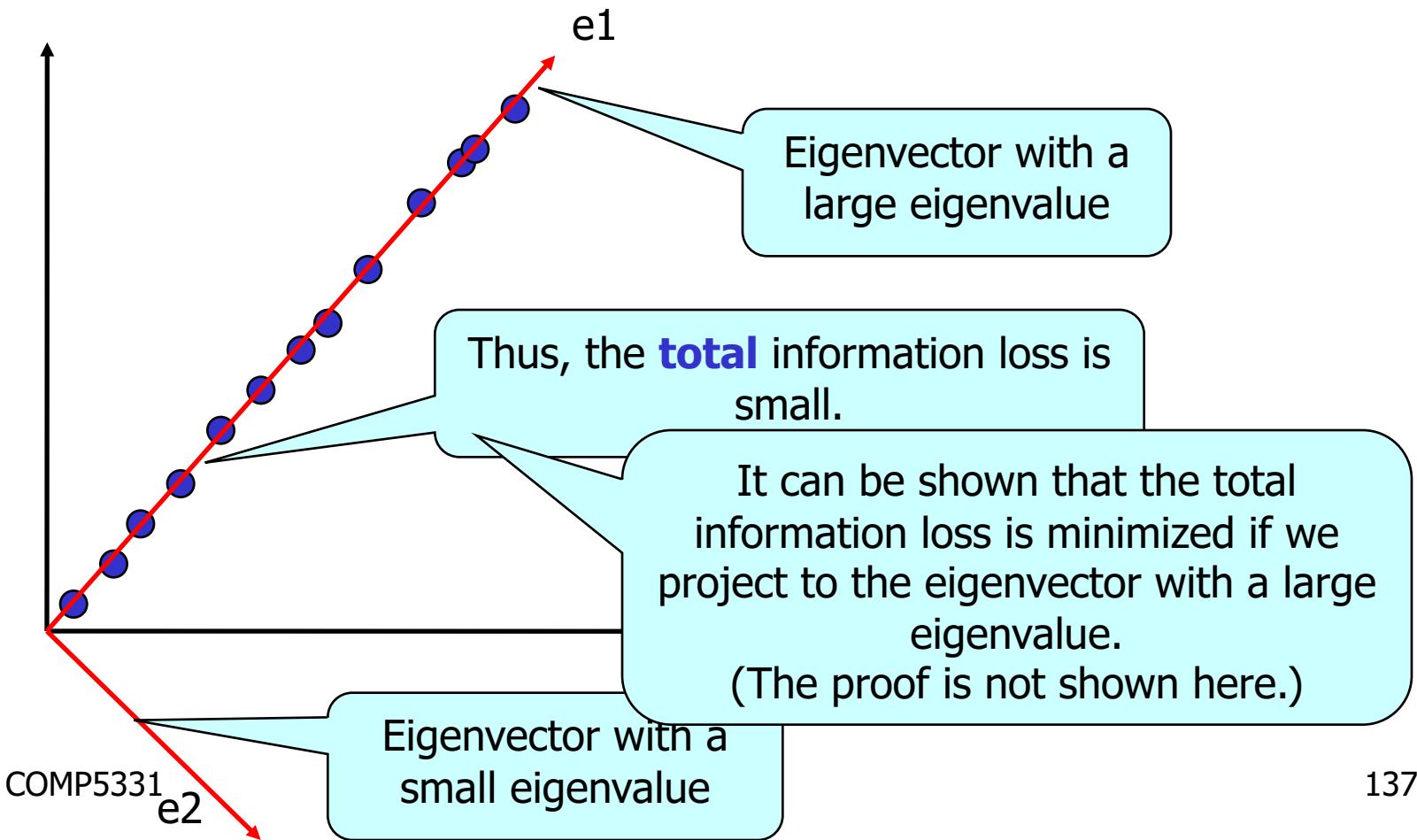
Case 1

- Consider that the data points are projected on e_1

Suppose all data points are projected on vector e_1

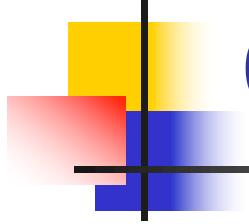


After all data points are projected on vector e_1



Objective of Dimension Reduction

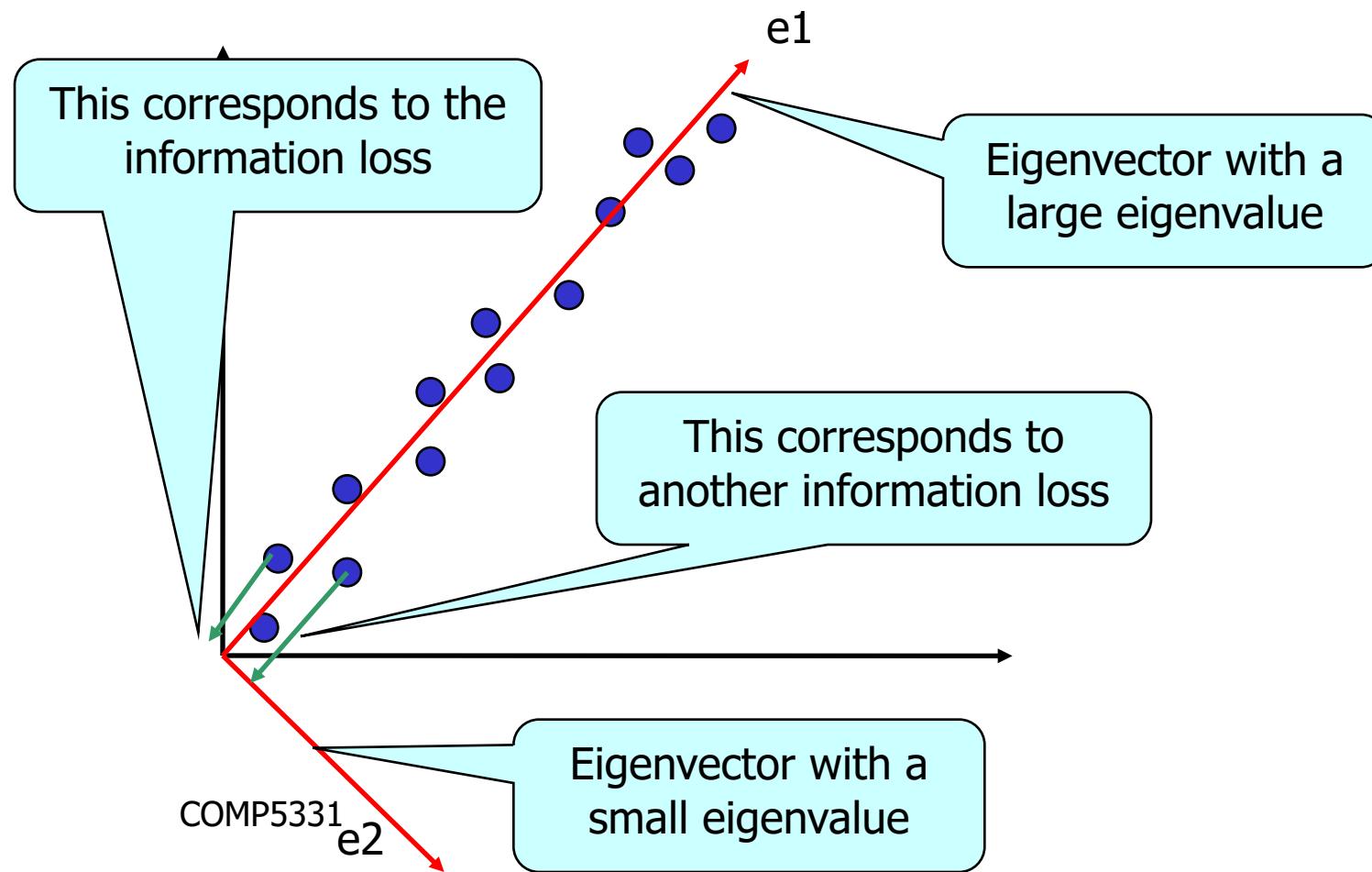
- The objective of Dimension Reduction
 - To **reduce** the total number of dimensions
 - At the same time, we want to keep information as much as possible.
(i.e., **minimize** the information loss)
- In our example,
 - We reduce from two dimensions to one dimension
 - The eigenvector with a large eigenvector corresponds to this dimension
 - After we adopt this dimension, we can minimize the information loss



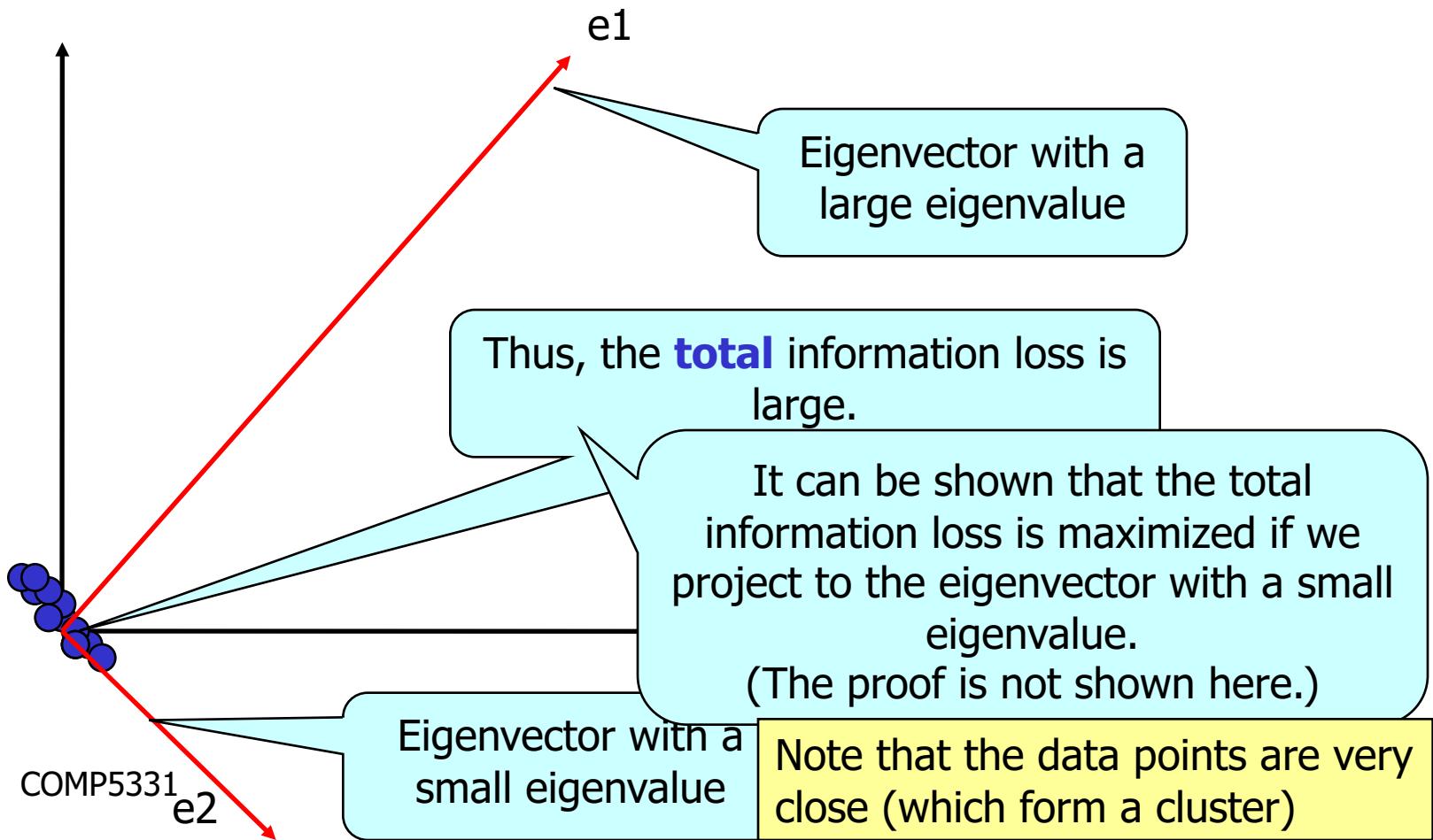
Case 2

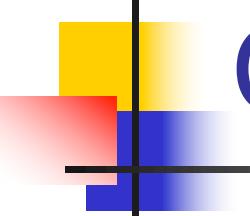
- Consider that the data points are projected on e_2

Suppose all data points are projected on vector e_2



After all data points are projected on vector e_2





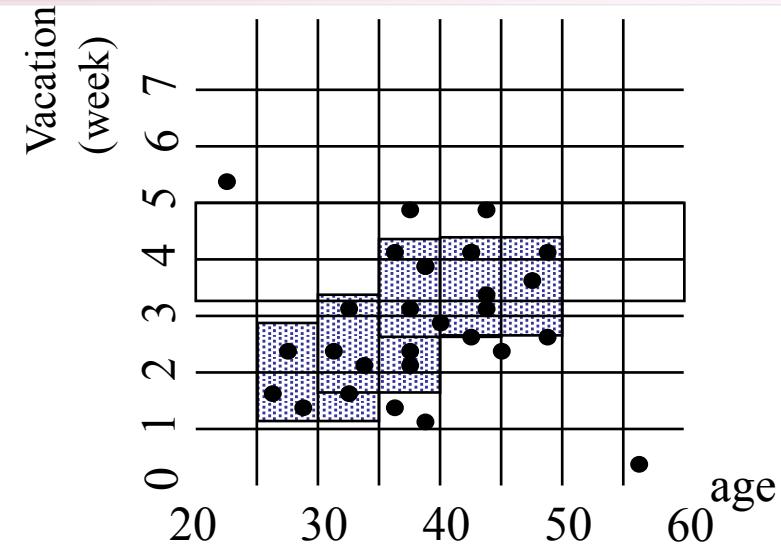
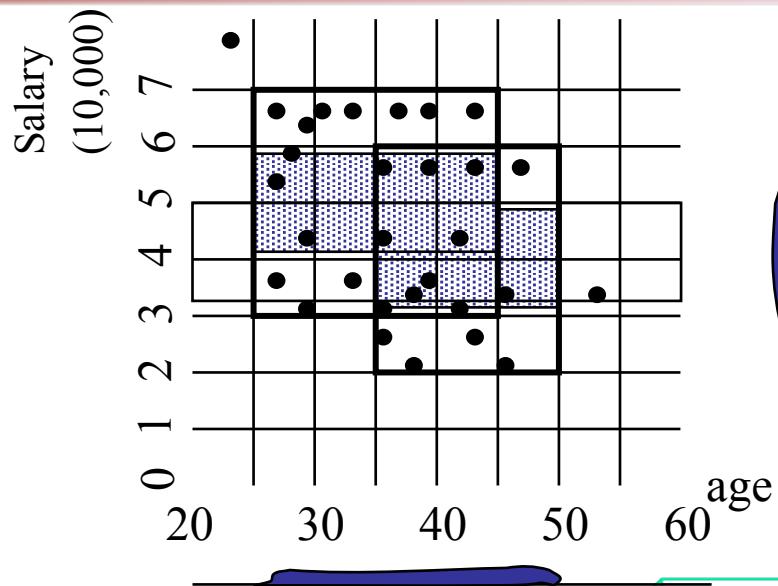
Objective of Subspace Clustering (KL-Transform)

- The objective of Subspace clustering
 - To **reduce** the total number of dimensions
 - At the same time, we want to find a cluster
 - A cluster is a group of “close” data points
 - This means that, after the data points are transformed, the data points are very close.
 - In KL-transform, you can see that the information loss is **maximized**.
- In our example,
 - We reduce from two dimensions to one dimension
 - The eigenvector with a small eigenvalue corresponds to this dimension
 - After we adopt this dimension, we can maximize the information loss

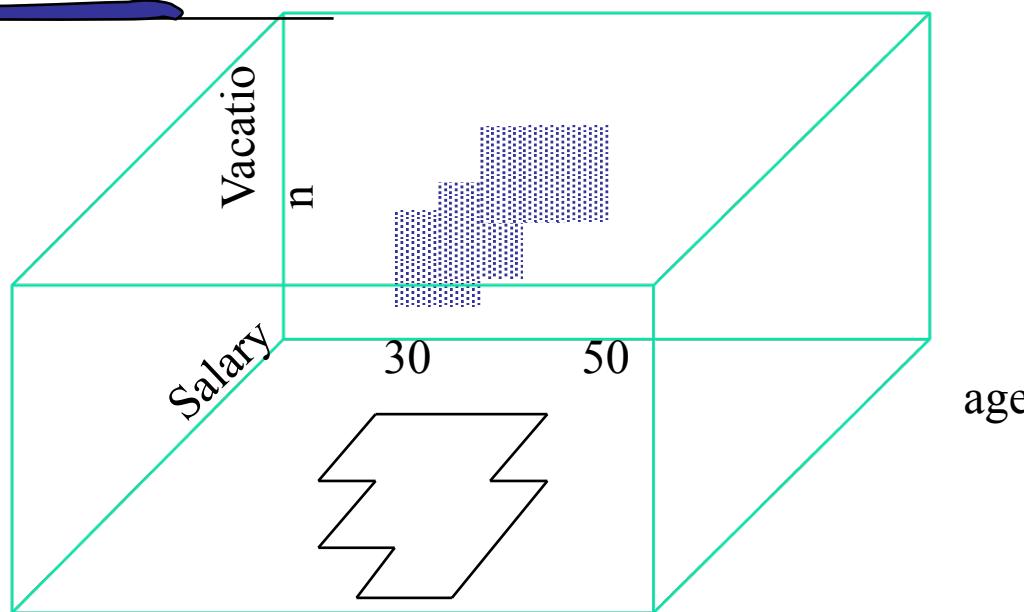
Subspace Clustering Method (I): Subspace Search Methods

- Search various subspaces to find clusters
- *Bottom-up approaches*
 - Start from low-D subspaces and search higher-D subspaces only when there may be clusters in such subspaces
 - Various pruning techniques to reduce the number of higher-D subspaces to be searched
 - Ex. CLIQUE (Agrawal et al. 1998)
- *Top-down approaches*
 - Start from full space and search smaller subspaces recursively
 - Effective only if the *locality assumption* holds: restricts that the subspace of a cluster can be determined by the local neighborhood
 - Ex. PROCLUS (Aggarwal et al. 1999): a k -medoid-like method

CLIQUE: SubSpace Clustering with Aprori Pruning



$$\tau = 3$$

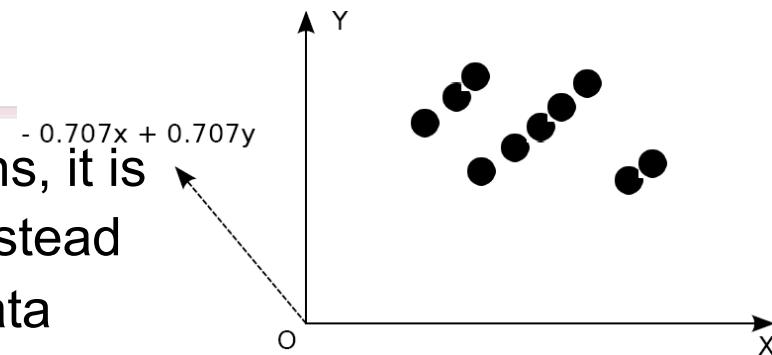


Subspace Clustering Method (II): Correlation-Based Methods

- Subspace search method: similarity based on distance or density
- Correlation-based method: based on advanced correlation models
- Ex. PCA-based approach:
 - Apply PCA (for Principal Component Analysis) to derive a set of new, uncorrelated dimensions,
 - then mine clusters in the new space or its subspaces
- Other space transformations:
 - Hough transform
 - Fractal dimensions

Dimensionality-Reduction Methods

- Dimensionality reduction: In some situations, it is more effective to construct a new space instead of using some subspaces of the original data
- Ex. To cluster the points in the right figure, any subspace of the original one, X and Y, cannot help, since all the three clusters will be projected into the overlapping areas in X and Y axes.
 - Construct a new dimension as the dashed one, the three clusters become apparent when the points projected into the new dimension
- Dimensionality reduction methods
 - Feature selection and extraction: But may not focus on clustering structure finding
 - Spectral clustering: Combining feature extraction and clustering (i.e., use the *spectrum* of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions)
 - Normalized Cuts (Shi and Malik, CVPR'97 or PAMI'2000)
 - The Ng-Jordan-Weiss algorithm (NIPS'01)



Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering
- Clustering High-Dimensional Data
- Summary 

Summary

- Probability Model-Based Clustering
 - Fuzzy clustering
 - Probability-model-based clustering
 - The EM algorithm
- Clustering High-Dimensional Data
 - Subspace clustering
 - Dimensionality reduction

References (I)

- K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, "When is Nearest Neighbor Meaningful?", ICDT 1999
- R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", SIGMOD 1998
- C.-H. Cheng, A. W.-C. Fu and Y. Zhang, "Entropy-based Subspace Clustering for Mining Numerical Data", SIGKDD 1999
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD'98*
- C. C. Aggarwal, C. Procopiuc, J. Wolf, P. S. Yu, and J.-S. Park. Fast algorithms for projected clustering. *SIGMOD'99*
- S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56:5:1–5:37, 2009.
- J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
- K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? *ICDT'99*
- Y. Cheng and G. Church. Bioclustering of expression data. *ISMB'00*
- I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. *SDM'05*
- I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. *PKDD'06*
- C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *J. American Stat. Assoc.*, 97:611–631, 2002.
- F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, 1999.
- G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. *KDD'02*

References (II)

- G. J. McLachlan and K. E. Bkasford. *Mixture Models: Inference and Applications to Clustering*. John Wiley & Sons, 1988.
- B. Mirkin. Mathematical classification and clustering. *J. of Global Optimization*, 12:105–108, 1998.
- S. C. Madeira and A. L. Oliveira. Bioclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1, 2004.
- A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. NIPS'01
- J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. *ICDM'03*
- M. Radovanović, A. Nanopoulos, and M. Ivanović. Nearest neighbors in high-dimensional data: the emergence and influence of hubs. *ICML'09*
- S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.
- A. K. H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. *ICDE'01*
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based clustering in large databases. *ICDT'01*
- A. Tanay, R. Sharan, and R. Shamir. Bioclustering algorithms: A survey. In *Handbook of Computational Molecular Biology*, Chapman & Hall, 2004.
- K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. *ICML'01*
- H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. *SIGMOD'02*
- X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: A structural clustering algorithm for networks. *KDD'07*
- X. Yin, J. Han, and P.S. Yu, “Cross-Relational Clustering with User's Guidance”, KDD'05