Road to Junior Data Scientist

Module 5: Unsupervised learning

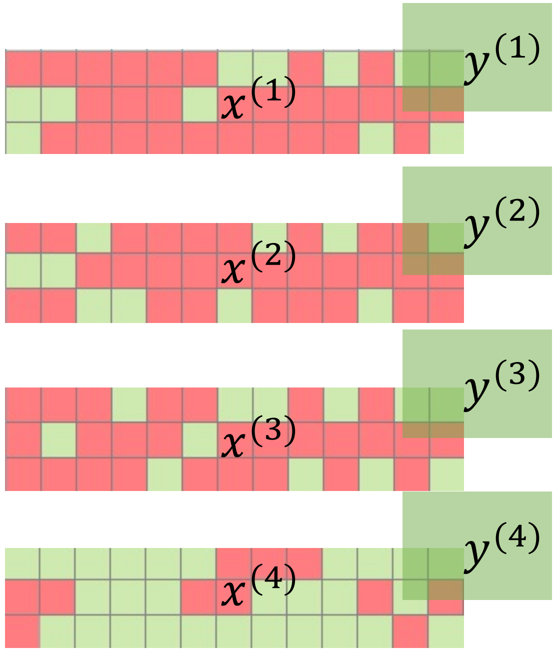
Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

What is unsupervised learning?

Supervised learning

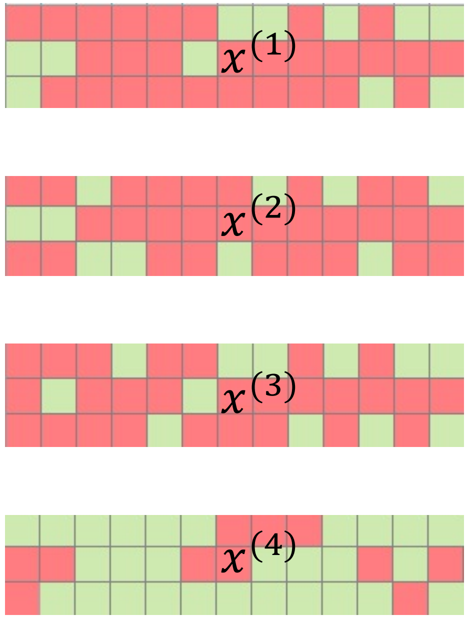
$$y = \hat{f}(x)$$



Labeled data

Unsupervised learning

$$y = ?$$

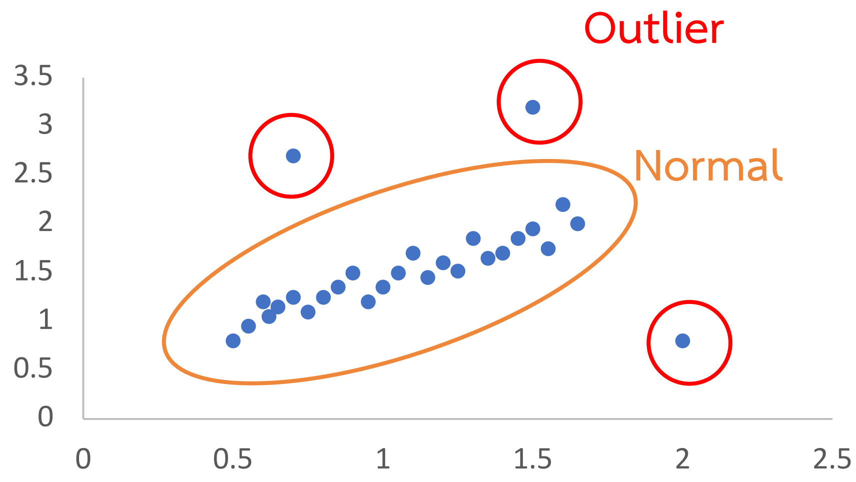


- Grouping
- Finding repeating patterns

Unlabeled data

Outlier detection

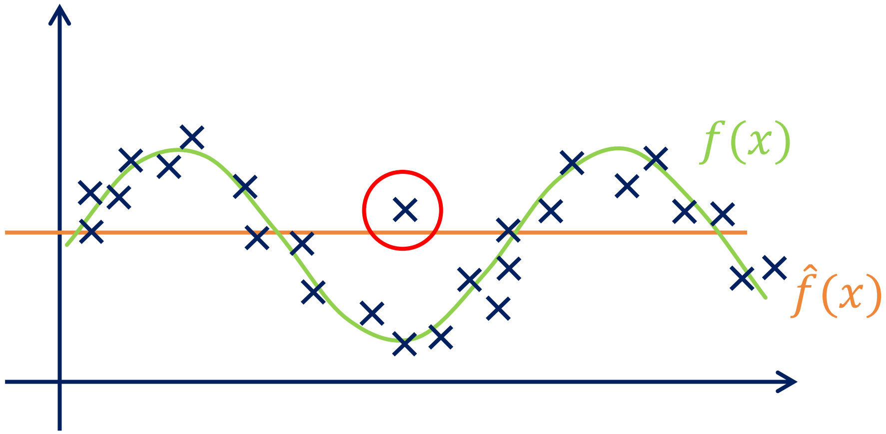
Ex 1:



Ex 2:

Timestamp	Productivity
2021-06-01	100
2021-06-02	120
2021-06-03	110
2021-06-04	110
2021-06-05	300
2021-06-06	100
2021-06-07	120
2021-06-08	110

Ex 3:



Fraud detection

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.77	0
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053527	24.79	0
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.88	0
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.00	0
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.00	0

284315 rows × 31 columns

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
541	406.0	-2.312227	1.951992	-1.609851	3.997906	-0.522188	-1.426545	-2.537387	1.391657	-2.770089	...	0.517232	-0.035049	-0.465211	0.320198	0.044519	0.177840	0.261145	-0.143276	0.00	1
623	472.0	-3.043541	-3.157307	1.088463	2.288644	1.359805	-1.064823	0.325574	-0.067794	-0.270953	...	0.661696	0.435477	1.375966	-0.293803	0.279798	-0.145362	-0.252773	0.035764	529.00	1
4920	4462.0	-2.303350	1.759247	-0.359745	2.330243	-0.821628	-0.075788	0.562320	-0.399147	-0.238253	...	-0.294166	-0.932391	0.172726	-0.087330	-0.156114	-0.542628	0.039566	-0.153029	239.93	1
6108	6986.0	-4.397974	1.358367	-2.592844	2.679787	-1.128131	-1.706536	-3.496197	-0.248778	-0.247768	...	0.573574	0.176968	-0.436207	-0.053502	0.252405	-0.657488	-0.827136	0.849573	59.00	1
6329	7519.0	1.234235	3.019740	-4.304597	4.732795	3.624201	-1.357746	1.713445	-0.496358	-1.282858	...	-0.379068	-0.704181	-0.656805	-1.632653	1.488901	0.566797	-0.010016	0.146793	1.00	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885	0.537503	0.788395	0.292680	0.147968	390.00	1
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049	0.521875	0.739467	0.389152	0.186637	0.76	1
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070	-0.739695	0.471111	0.385107	0.194361	77.89	1
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659	-0.328168	0.606116	0.884876	-0.253700	245.00	1
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261	0.313267	-0.289617	0.002988	-0.015309	42.53	1

492 rows × 31 columns

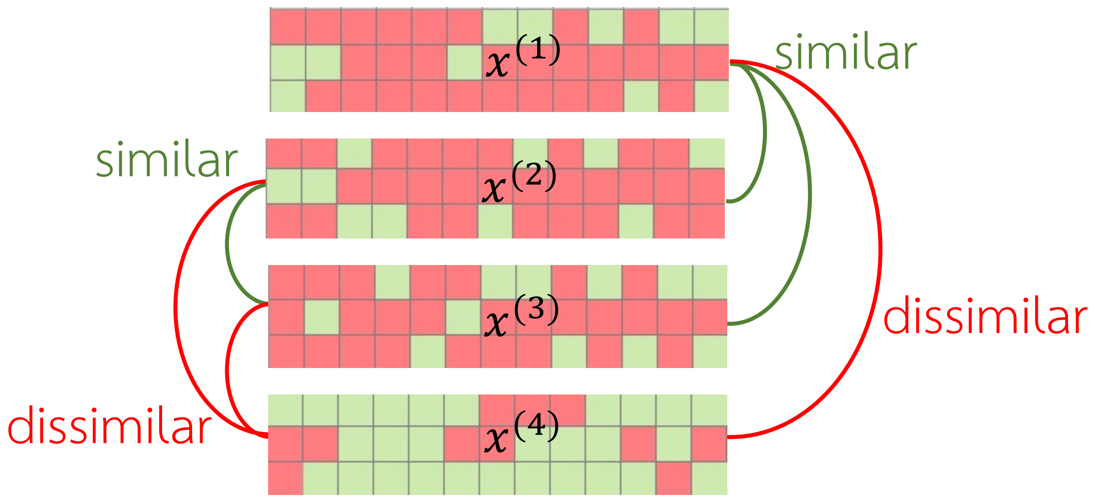
Customer segmentation

Customer ID	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Country
13085.0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	United Kingdom
13085.0	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	United Kingdom
13085.0	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	United Kingdom
13085.0	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	United Kingdom
13085.0	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	United Kingdom
...	...	...	...	...	...	...	...
12680.0	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	France
12680.0	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	France
12680.0	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	France
12680.0	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	France
12680.0	581587	POST	POSTAGE	1	2011-12-09 12:50:00	18.00	France

1067371 rows × 7 columns

General concepts

- Similarity and dissimilarity : group similar things together
- Inter-cluster / intra-cluster : separate dissimilar things from each other



Objective function

$$\sum_k max_{x^{(i)},x^{(j)} \epsilon c_k} sim\left(x^{(i)}, x^{(j)}\right)$$

$$\sum_m \sum_k min_{x^{(i)} \epsilon c_n, x^{(j)} \epsilon c_m} sim\left(x^{(i)}, x^{(j)}\right), \mathrm{n} \neq m$$

Bad cluster

$x^{(1)}$

$x^{(2)}$

$x^{(3)}$

$x^{(4)}$

1 item per cluster — Maximum intra-cluster similarity

1 cluster — Minimum inter-cluster similarity

We need some constraints

e.g. minimum number of items in a cluster

maximum number of clusters

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Data types

- Numerical values: sales amount, number of transaction, balance, debt, age
- Categorical values: sex, educational background, province code

Age	Sex	Educational background	Province code	Number of monthly transaction	Balance
[0, ∞)	{0, 1}	{1, 2, 3, 4, 5, 6}	{1, 2, 3, ..., 77}	[0, ∞)	(-∞, ∞)

$x^{(1)}$ = (24, 0, 4, 8, 3, 20000)

$x^{(2)}$ = (38, 1, 6, 10, 20, 100000)

$x^{(3)}$ = (35, 1, 6, 10, 10, 60000)

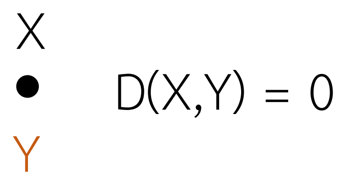
$x^{(4)}$ = (30, 0, 5, 50, 10, 50000)

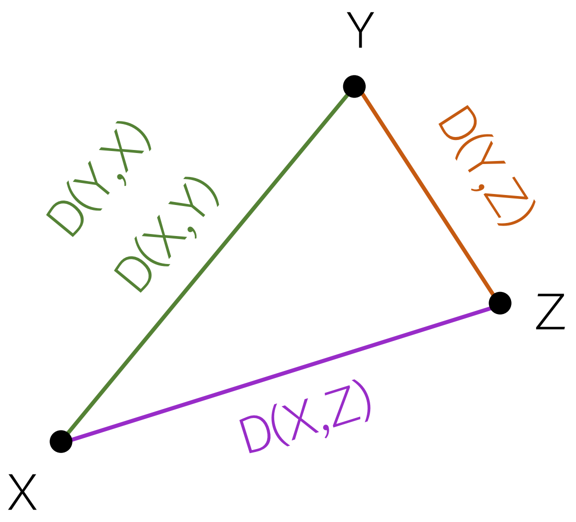
20000 is closer to 50000 than 100000

Province 8 is NOT closer to province 10 than province 50

Distance metrics

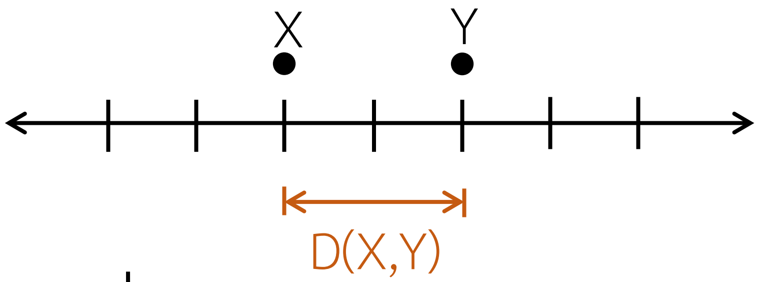
- Minimum = 0 (Similar)
- Maximum = ∞ (Dissimilar)
- D(X,Y) = 0 if and only if X = Y
- D(X,Y) = D(Y,X)
- D(X,Z) ≤ D(X,Y) + D(Y,Z)



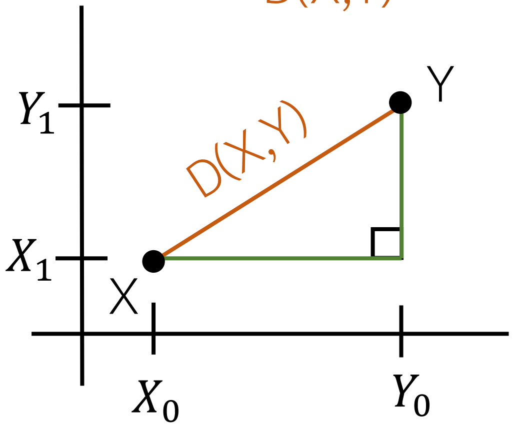


Euclidean distance

- 1-d

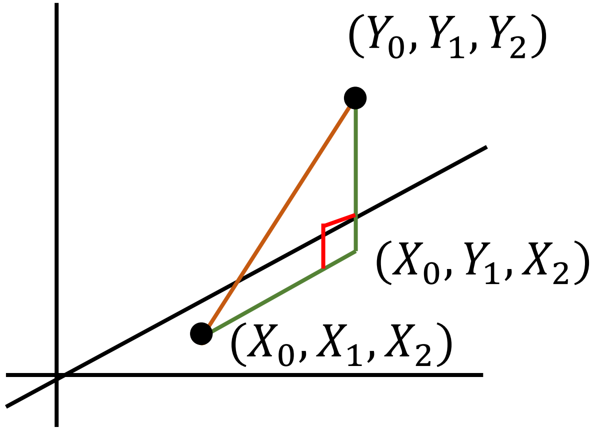


- 2-d



$$D(X,Y) = \sqrt{(X_0 - Y_0)^2 + (X_1 - Y_1)^2}$$

- 3-d



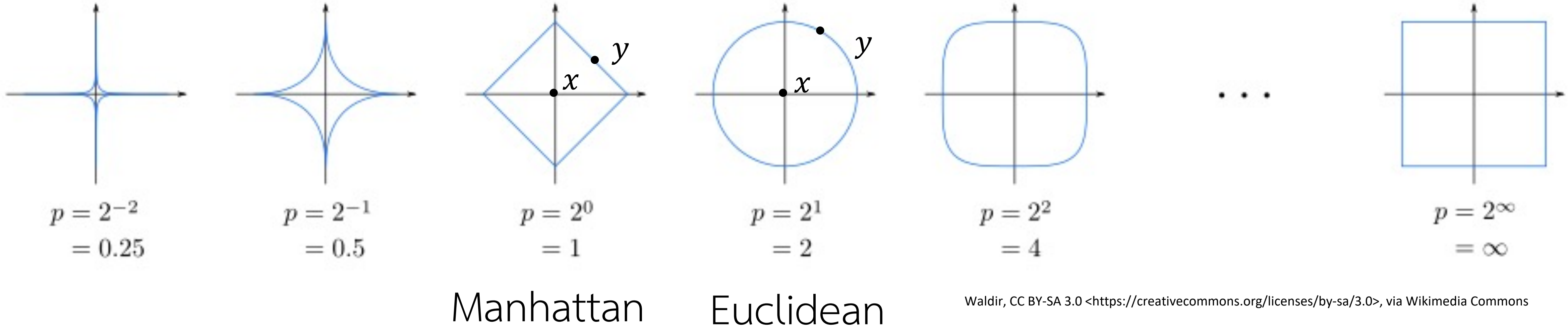
$$D(X,Y) = \sqrt{(X_0 - Y_0)^2 + (X_1 - Y_1)^2 + (X_2 - Y_2)^2}$$

$$D(X,Y) = \sqrt{\sum_{i=0}^{2} (X_i - Y_i)^2}$$

- D-dimensional vector $D(X,Y) = \sqrt{\sum_{i=0}^{d} (X_i - Y_i)^2}$

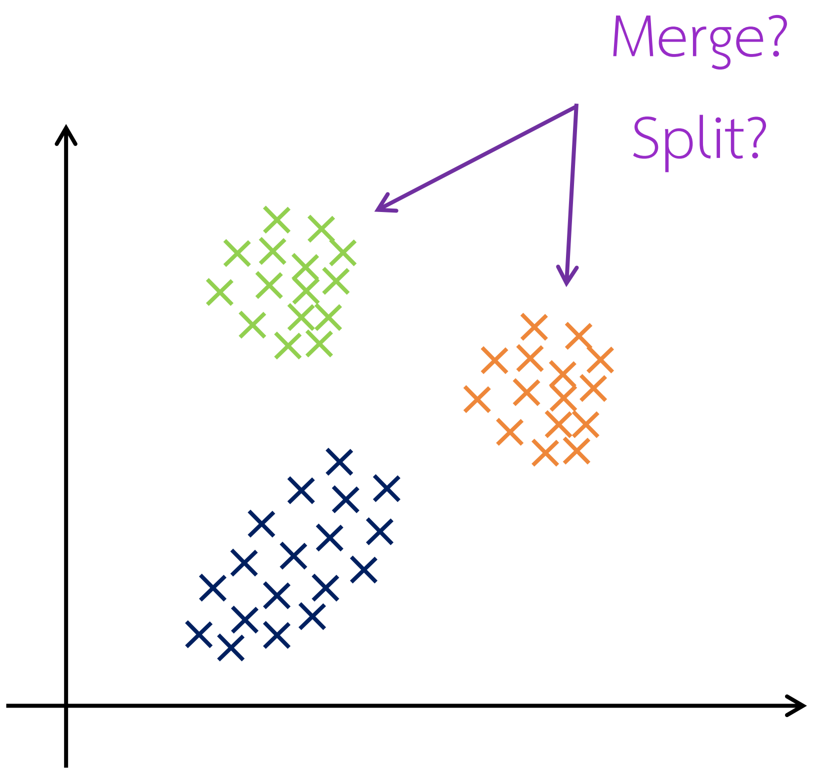
Minkowski distance

- $D(X,Y) = \left(\sum_{i=1}^{n} |X_i - Y_i|^p\right)^{1/p}$
- Euclidean distance: p = 2
- Manhattan distance: p = 1



Variance

- Minimum = 0 (similar)
- Maximum = ∞ (dissimilar)
- Cluster property



Hamming distance

Count dissimilar positions; suitable for categorical values

$$x : 0\ 1\ 0\ 0$$
$$y : 1\ 0\ 0\ 1$$

$$D(x, y) = 3$$

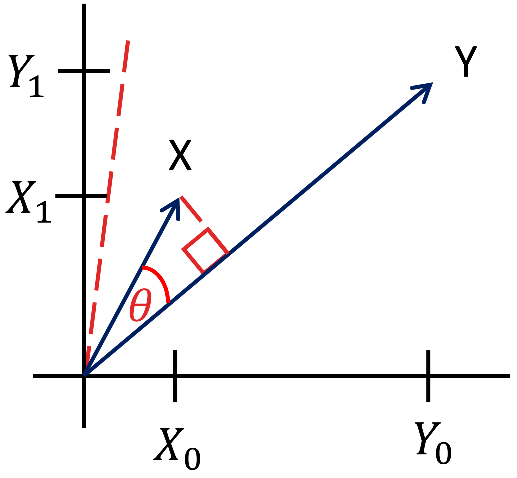
$$x^{(2)} = (38, 1, 6, 10, 20, 100000)$$
$$x^{(3)} = (35, 1, 6, 10, 10, 60000)$$

$$\mathrm{D}(x^{(2)}, x^{(3)}) = 3$$

Similarity function

- Loosely speaking, it's the inverse of a distance metric
- High similarity -> large value $\propto$
- Low similarity -> small value (negative or zero)

Cosine similarity



Minimum = -1

Maximum = 1

$$\cos\theta = \frac{X \cdot Y}{||X|| \cdot ||Y||}$$

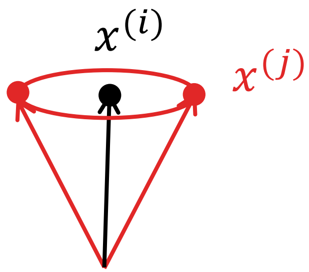
$$= \frac{\sum_{i=1}^{d} X_i \cdot Y_i}{||X|| \cdot ||Y||}$$

Advantage

- Consider only non-zero values

Disadvantage

- Small range



$$x^{(2)} = (38, 1, 6, 10, 20, 100000)$$
$$x^{(3)} = (35, 1, 6, 10, 10, 60000)$$

$$\cos\theta = \frac{(38 * 35 + 1 * 1 + 6 * 6 + 10 * 10 + 20 * 10 + 100000 * 60000)}{\sqrt{38^2 + 1^2 + 6^2 + 10^2 + 20^2 + 100000^2} \cdot \sqrt{35^2 + 1^2 + 6^2 + 10^2 + 10^2 + 60000^2}}$$

Pearson's correlation

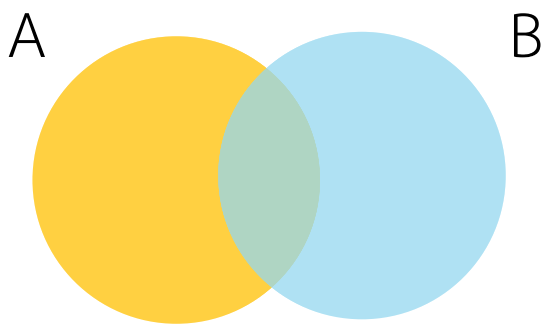
From $$\rho(x,y) = \frac{E[(x-E[x])(y-E[y])]}{\sqrt{E[(x-E[x])^2]}\sqrt{E[(y-E[y])^2]}},$$

We want to find the correlation coefficient between feature j and feature k

- The dataset contains d training samples
- $x_j^{(i)}$ is feature j of input $x^{(i)}$
- x_j is the d dimensional vector containing all the values of j^{th} feature for all training samples

$$C_{j,k} = \frac{\left|\sum_{i=1}^{d}\left(x_j^{(i)} - \overline{x_j}\right)\left(x_k^{(i)} - \overline{x_k}\right)\right|}{\sqrt{\sum_{i=1}^{d}\left(x_j^{(i)} - \overline{x_j}\right)^2}\sqrt{\sum_{i=1}^{d}\left(x_k^{(i)} - \overline{x_k}\right)^2}}$$

Jaccard index



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Intersection Over Union (IOU)

Minimum = 0

Maximum = 1

Note: If $A = B = \emptyset$ then $J(A,B) = 1$

Jaccard distance

$$d_{J(A,B)} = 1 - J(A,B)$$

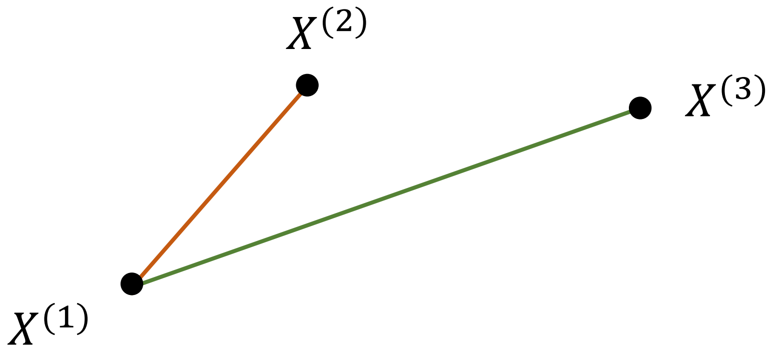
Minimum = 0

Maximum = 1

$$x^{(2)} = (38, 1, 6, 10, 20, 100000)$$
$$x^{(3)} = (35, 1, 6, 10, 10, 60000)$$

$$J\left(x^{(2)}, x^{(3)}\right) = \frac{3}{3+3}$$

Comparison



Choosing similarity/dissimilarity metrics

Numerical/Categorical

Sparsity

Similarity distribution

Ordering

Range boundary

Hamming: $X^{(1)} \neq X^{(2)}$ and $X^{(1)} \neq X^{(3)}$

Euclidean,

Manhattan, $: |X^{(1)} - X^{(2)}| < |X^{(1)} - X^{(3)}|$

Minkowski

Clustering evaluation

- Internal evaluation
  - Variance
  - Similarity/distance
    - DB – index
    - Dunn index
    - Silhouette Coefficient

- External evaluation
  - Require class labels
  - Purity
  - Rand index
  - Mutual information
  - F- measure

Code samples

[Online Retail II UCI | Kaggle (https://www.kaggle.com/mashlyn/online-retail-ii-uci)](https://www.kaggle.com/mashlyn/online-retail-ii-uci)

- InvoiceNo: nominal
- StockCode: nominal
- Description: nominal
- Quantity: numeric
- InvoiceDate: numeric
- UnitPrice: numeric
- CustomerID: nominal
- Country: nominal

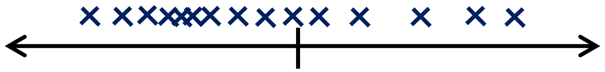
Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

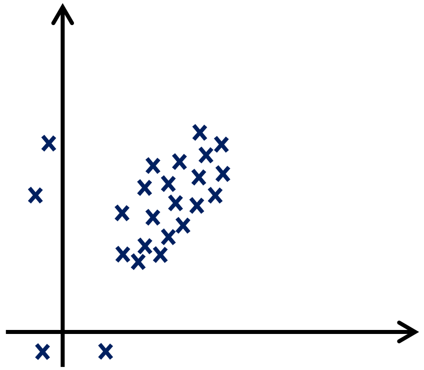
Curse of dimensionality

- Processing time/memory
- Visualization
- Redundancy

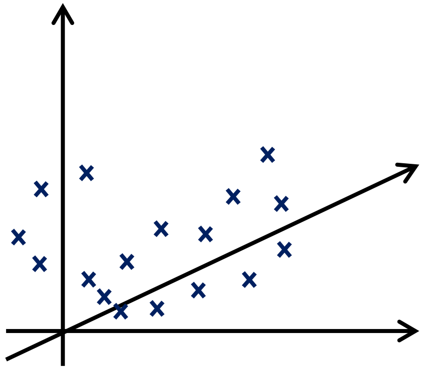
1-d



2-d



3-d



4-d??

5-d??

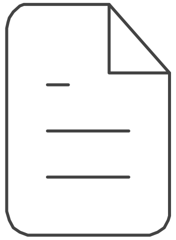
Negatively correlated → Also redundant

10-d

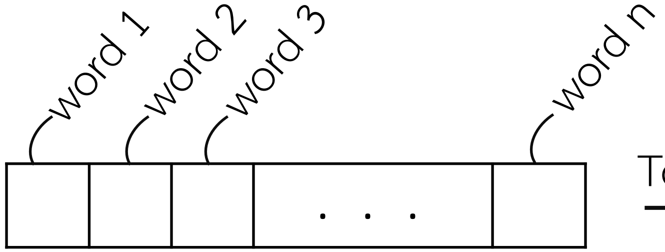
$x^{(1)}$	1	-1	-2	2	2	3	2	-2	6	5
$x^{(2)}$	2	-1	-1	4	4	4	1	-4	5	-2
$x^{(3)}$	6	5	4	-2	-2	4	1	2	4	-2
$x^{(4)}$	3	4	3	-6	-6	3	2	6	-1	3

same value/
highly correlated → redundant

Feature selection



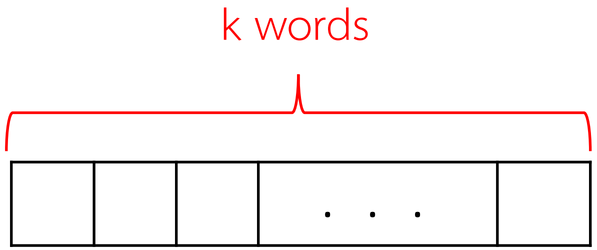
Text document



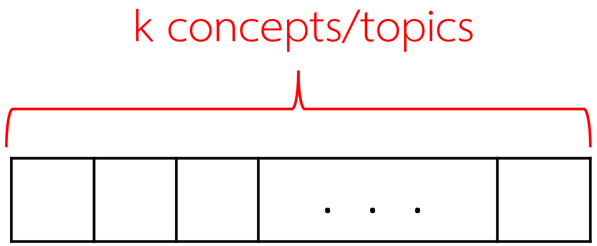
Document vector

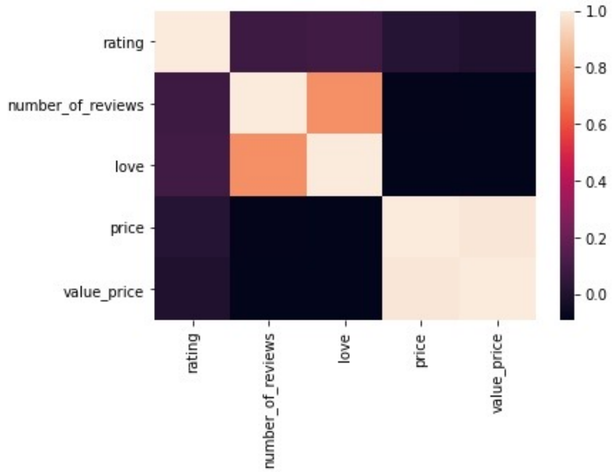
- Frequency
- Importance score

Top k words



Transform





price and value_price are positively correlate

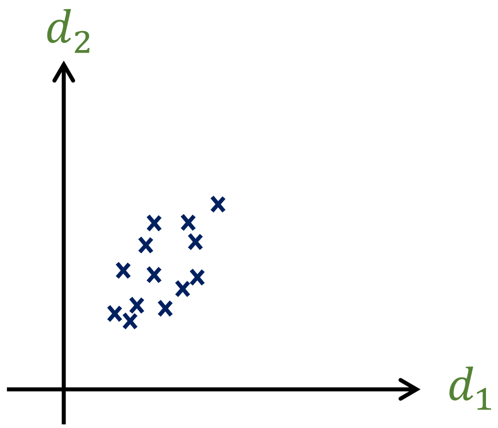
number_of_review and love are positively correlate

- drop one of them

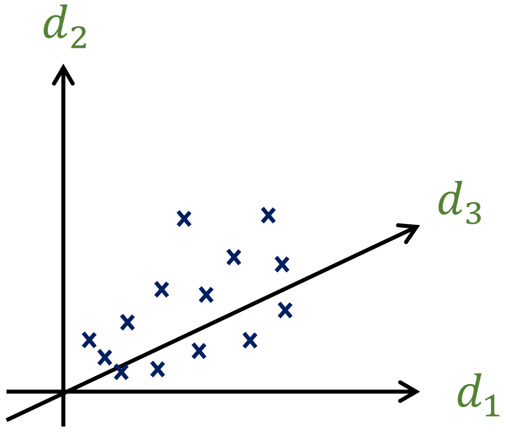
Visualization

Map K-dimensional vector to

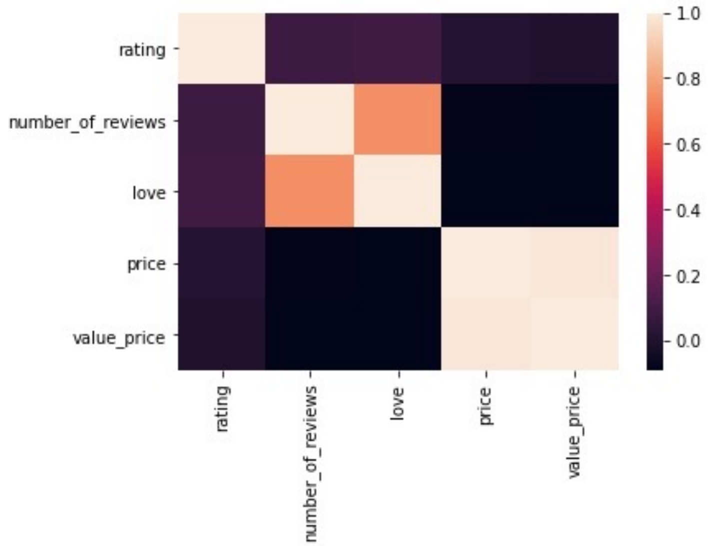
- 2-d plot



- 3-d plot



- 2-d with heat map



We may see some insight from low dimensional space

Pearson's correlation

From $$\rho(x,y) = \frac{E[(x-E[x])(y-E[y])]}{\sqrt{E[(x-E[x])^2]}\sqrt{E[(y-E[y])^2]}},$$

We want to find the correlation coefficient between feature j and feature k

- The dataset contains d training samples
- $x_j^{(i)}$ is feature j of input $x^{(i)}$
- x_j is the d dimensional vector containing all the values of j^{th} feature for all training samples

$$C_{j,k} = \frac{\left|\sum_{i=1}^{d}\left(x_j^{(i)} - \overline{x_j}\right)\left(x_k^{(i)} - \overline{x_k}\right)\right|}{\sqrt{\sum_{i=1}^{d}\left(x_j^{(i)} - \overline{x_j}\right)^2}\sqrt{\sum_{i=1}^{d}\left(x_k^{(i)} - \overline{x_k}\right)^2}}$$

Principal component analysis (PCA)

- Keep different points away from each other

  → high variance

- Finding the best projection

$x \xrightarrow[\text{project}]{} x' \xrightarrow[\text{project}]{} x$

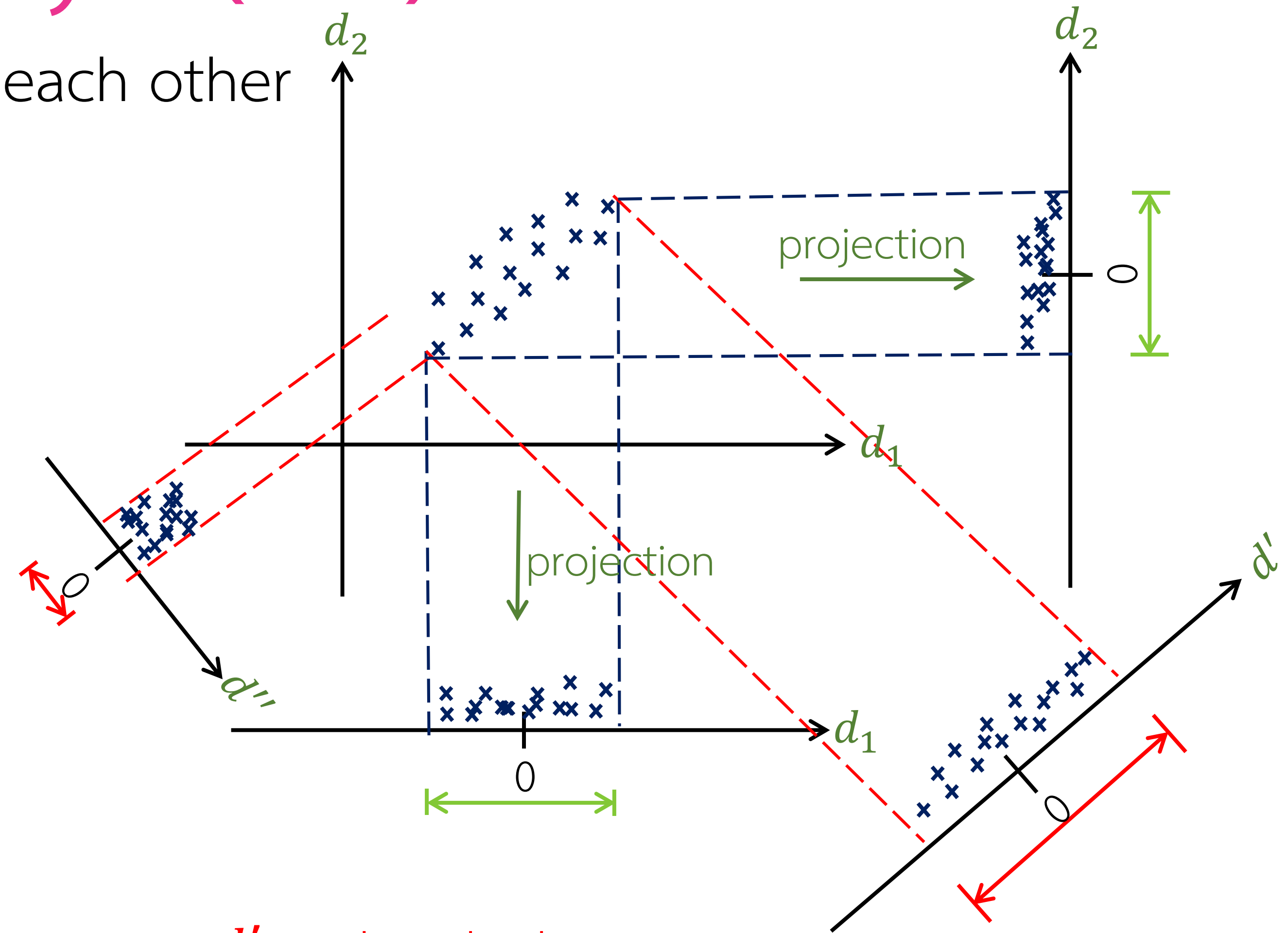
(encode) (decode)

Choose linear transformation

↓

Eigen decomposition to eigen axes

= principal components



The projection on d' axis has the largest variance.

Dimensionality reduction

Diagonal matrix sort descendingly

$$\begin{bmatrix} A \end{bmatrix}_{d\times d} \times \begin{bmatrix} \ \ \ \end{bmatrix}_{d\times d} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix}_{d\times d} \times \begin{bmatrix} | & | & & | \\ v^{(1)} & v^{(2)} & \dots & v^{(d)} \\ | & | & & | \end{bmatrix}_{d\times d}$$

$$\begin{bmatrix} A \end{bmatrix}_{d\times d} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix} \begin{bmatrix} | & | & & | \\ v^{(1)} & v^{(2)} & \dots & v^{(d)} \\ | & | & & | \end{bmatrix} \begin{bmatrix} | & | & & | \\ v^{(1)} & v^{(2)} & \dots & v^{(d)} \\ | & | & & | \end{bmatrix}^{-1}$$

Choose top λ with corresponding axes → eigen axes

highest variance

Limitation: Eigen decomposition requires A to be a square matrix

$$= \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix} \begin{bmatrix} | & | & & | \\ v^{(1)} & v^{(2)} & \dots & v^{(d)} \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & v^{(1)} & — \\ — & v^{(2)} & — \\ & \vdots & \\ — & v^{(d)} & — \end{bmatrix}$$

Singular value decomposition (SVD)

Choose top σ

Diagonal matrix sort descendingly

$$\begin{bmatrix} — x^{(1)} — \\ — x^{(2)} — \\ \vdots \\ — x^{(n)} — \end{bmatrix}_{n\times d} = \begin{bmatrix} \\ U \\ \\ \end{bmatrix}_{n\times n} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & \Sigma & \\ & & \ddots & \\ & & & \sigma_d \end{bmatrix}_{n\times d} \begin{bmatrix} — v^{(1)} — \\ — v^{(2)} — \\ \vdots \\ — v^{(d)} — \end{bmatrix}_{d\times d}$$

$$\begin{bmatrix} — x^{(1)} — \\ — x^{(2)} — \\ \vdots \\ — x^{(n)} — \end{bmatrix}_{n\times d} \begin{bmatrix} | & | \\ v^{(1)} & v^{(2)} \\ | & | \end{bmatrix}_{d\times 2} = \underbrace{\begin{bmatrix} & \\ & \\ & \end{bmatrix}_{n\times n} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix}_{n\times 2}}_{\begin{bmatrix} & \\ & \end{bmatrix}_{n\times 2}} \overbrace{\boxed{\begin{bmatrix} — v^{(1)} — \\ — v^{(2)} — \end{bmatrix}_{2\times d} \begin{bmatrix} | & | \\ v^{(1)} & v^{(2)} \\ | & | \end{bmatrix}_{d\times 2}}}^{I}$$

$$\begin{bmatrix} — x^{(1)\prime} — \\ — x^{(2)\prime} — \\ \vdots \\ — x^{(n)\prime} — \end{bmatrix}_{n\times 2}$$

We can approximate d-dimensional vectors with 2-dimensional vectors

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

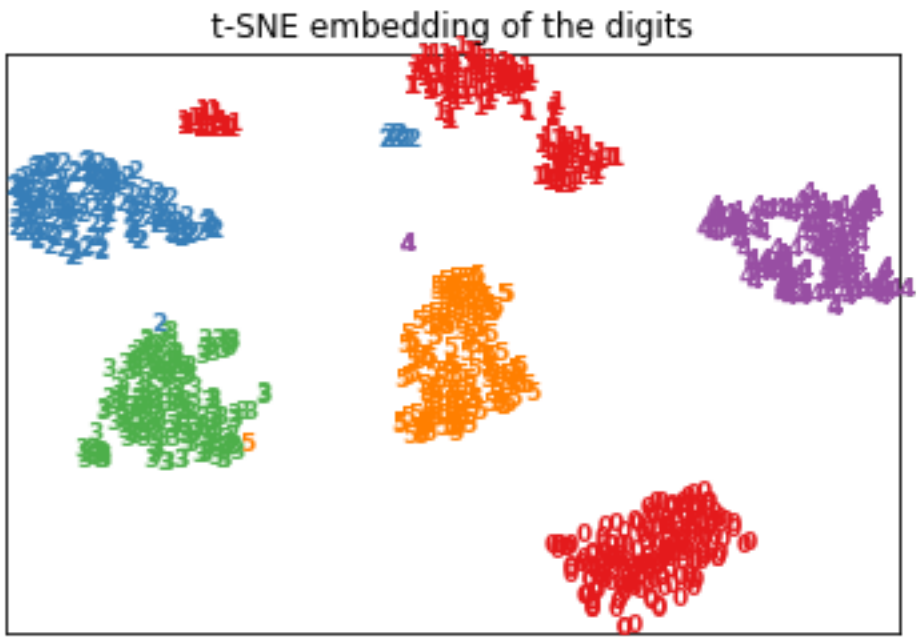
1. Construct a probability distribution over pairs of high-dimensional objects in such a way that
   - Similar objects have a high probability of being picked
   - Dissimilar points have an extremely small probability of being picked
2. Define a similar probability distribution over the points.
   - The location of the point in the low-dimensional space is the where it minimizes the KL divergence between its distribution in the low-dimensional and the high-dimensional space.

with respect to the locations of the points in the map

Sample codes

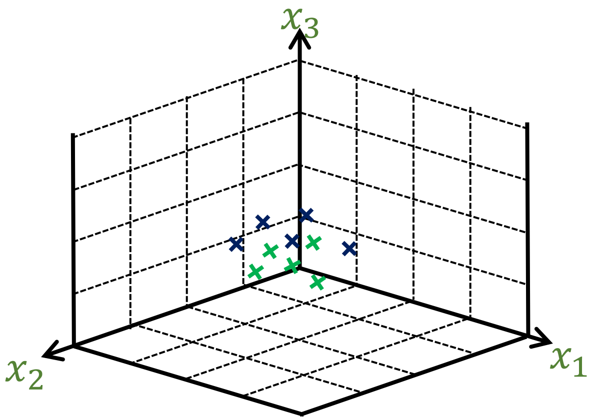
```
tsne = manifold.TSNE(n_components=2,
init='pca')
X_tsne = tsne.fit_transform(X)

plot_embedding(X_tsne)
```

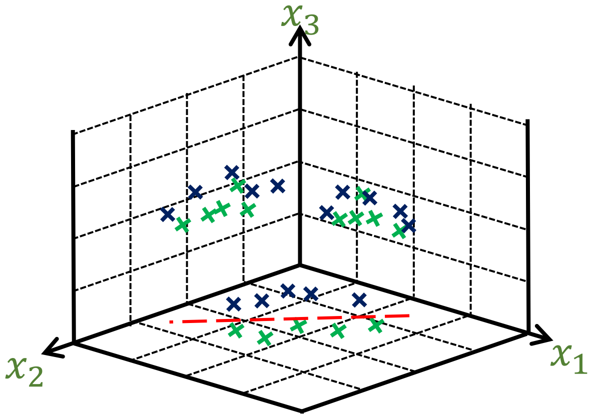


Common mistakes

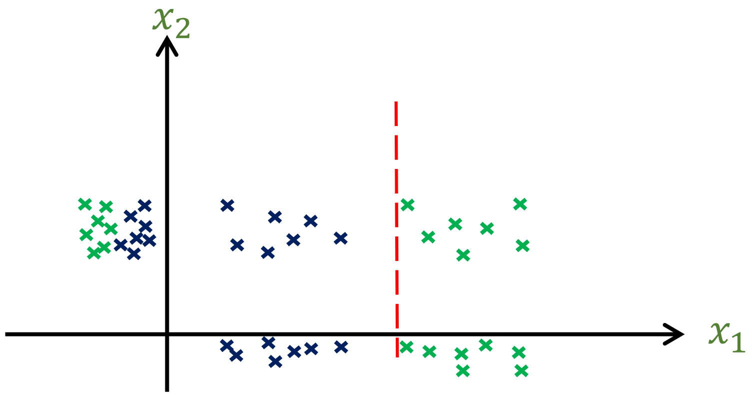
- Independence
- Linearly independent
- Correlation



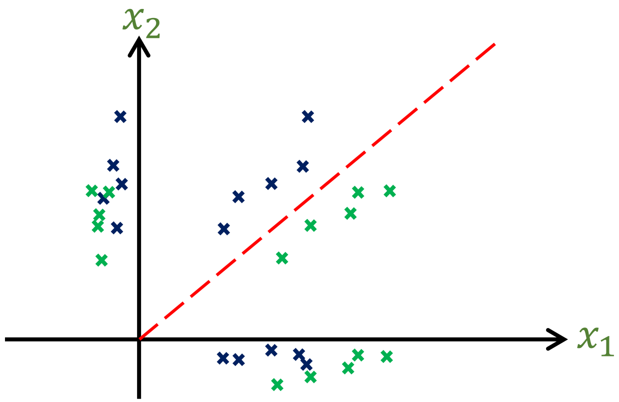
3-d example



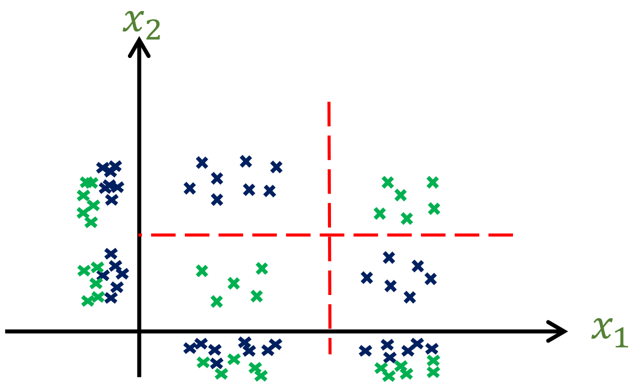
3-d projection



x_1 is informative

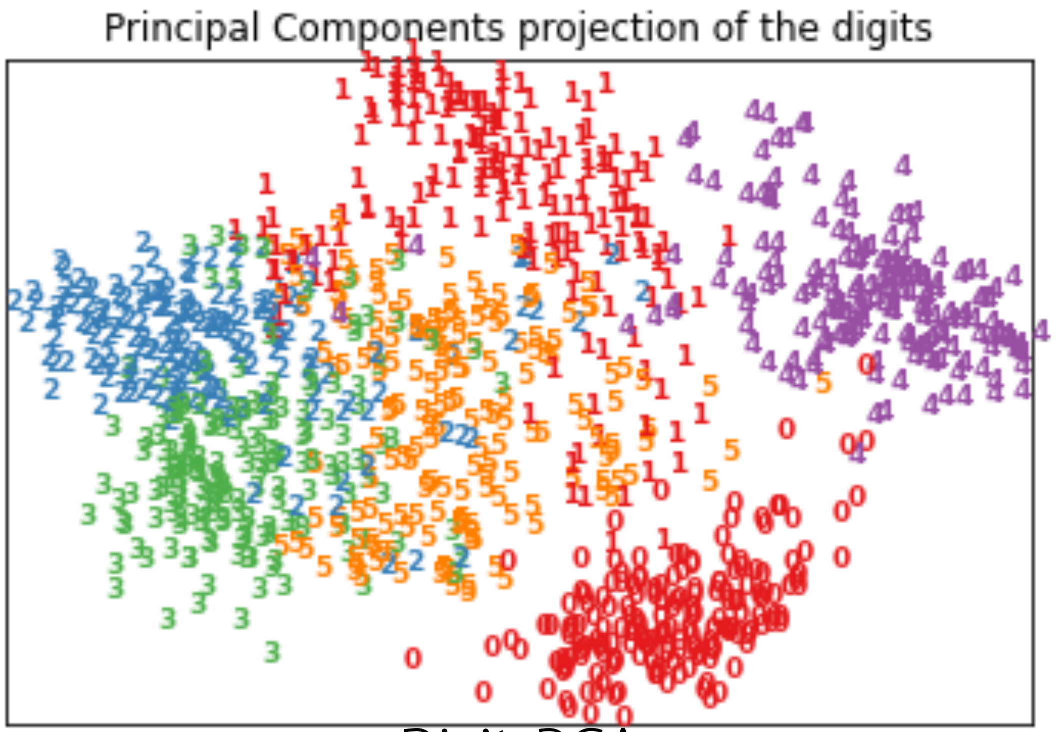


Either x_1 and x_2 alone is not informative

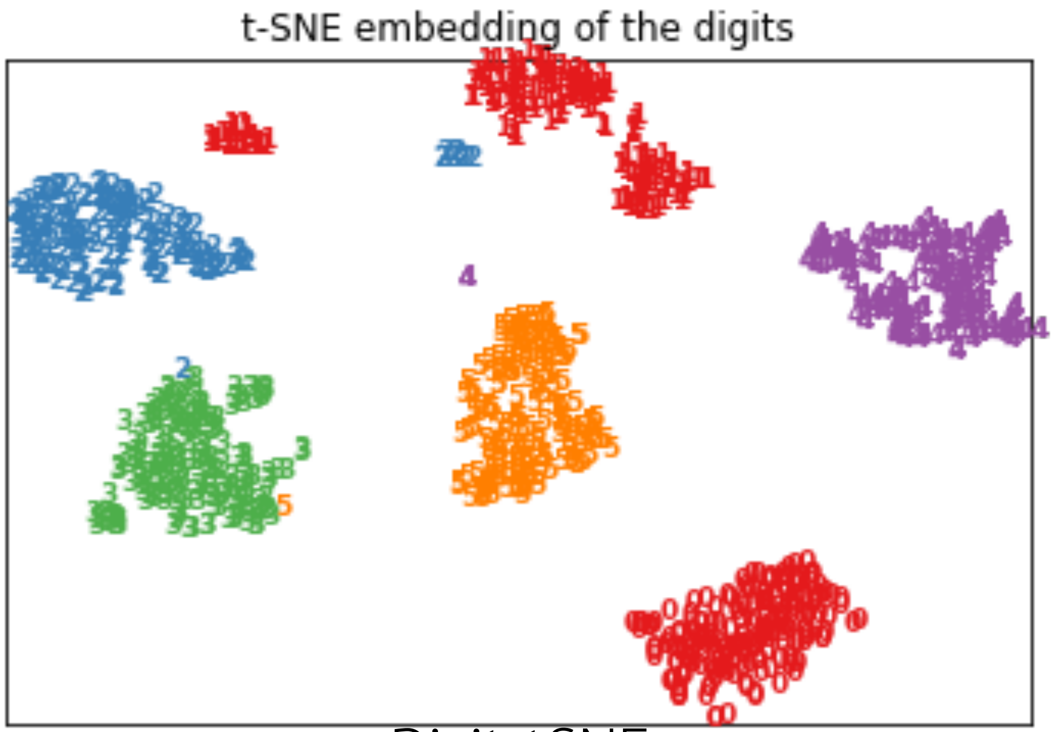


Chessboard pattern

Examples



Digit-PCA

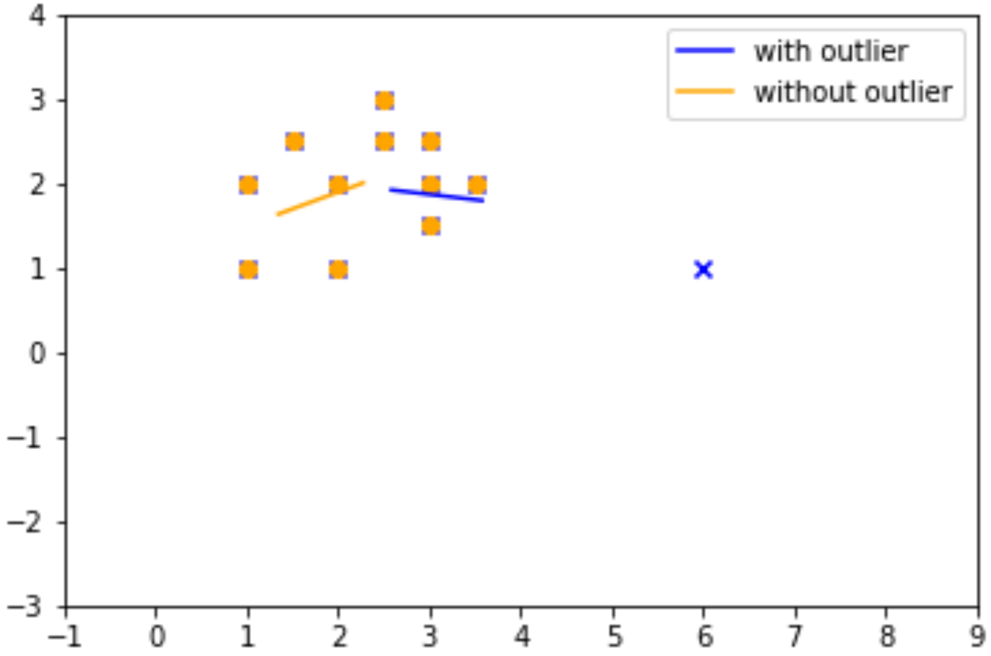


Digit-tSNE

Outlier problem

PCA tries to minimize the reconstruction error $(x' - x)^2$

therefore, it is sensitive to outliers



Code samples

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.77	0
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053527	24.79	0
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.88	0
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.00	0
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.00	0

284315 rows × 31 columns

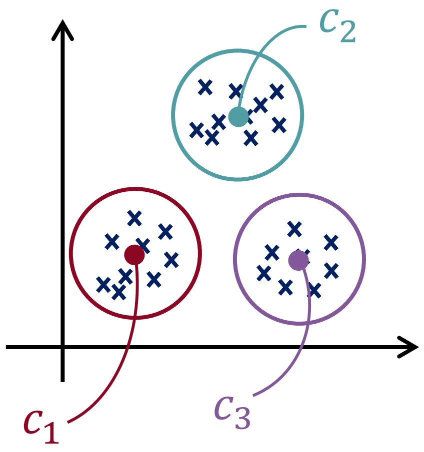
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
541	406.0	-2.312227	1.951992	-1.609851	3.997906	-0.522188	-1.426545	-2.537387	1.391657	-2.770089	...	0.517232	-0.035049	-0.465211	0.320198	0.044519	0.177840	0.261145	-0.143276	0.00	1
623	472.0	-3.043541	-3.157307	1.088463	2.288644	1.359805	-1.064823	0.325574	-0.067794	-0.270953	...	0.661696	0.435477	1.375966	-0.293803	0.279798	-0.145362	-0.252773	0.035764	529.00	1
4920	4462.0	-2.303350	1.759247	-0.359745	2.330243	-0.821628	-0.075788	0.562320	-0.399147	-0.238253	...	-0.294166	-0.932391	0.172726	-0.087330	-0.156114	-0.542628	0.039566	-0.153029	239.93	1
6108	6986.0	-4.397974	1.358367	-2.592844	2.679787	-1.128131	-1.706536	-3.496197	-0.248778	-0.247768	...	0.573574	0.176968	-0.436207	-0.053502	0.252405	-0.657488	-0.827136	0.849573	59.00	1
6329	7519.0	1.234235	3.019740	-4.304597	4.732795	3.624201	-1.357746	1.713445	-0.496358	-1.282858	...	-0.379068	-0.704181	-0.656805	-1.632653	1.488901	0.566797	-0.010016	0.146793	1.00	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885	0.537503	0.788395	0.292680	0.147968	390.00	1
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049	0.521875	0.739467	0.389152	0.186637	0.76	1
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070	-0.739695	0.471111	0.385107	0.194361	77.89	1
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659	-0.328168	0.606116	0.884876	-0.253700	245.00	1
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261	0.313267	-0.289617	0.002988	-0.015309	42.53	1

492 rows × 31 columns

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

K-means



c_1, c_2, c_3 are centroids

Find centroids (center of clusters) that minimize the distances between cluster members and cluster centroids

Euclidean distance

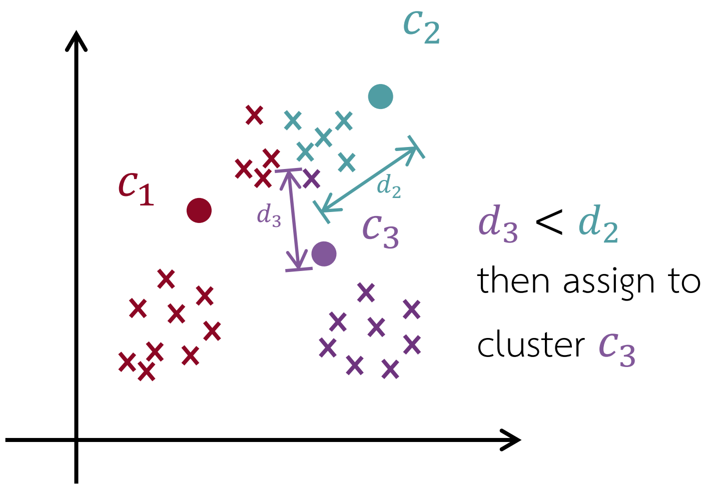
$$\underset{S}{argmin} \sum_{j=1}^{k} \sum_{x \in S_j} \left\|x^{(i)} - c_j\right\|_2$$

S is a cluster assignment

e.g. assign $x^{(i)}$ to cluster j

K-means

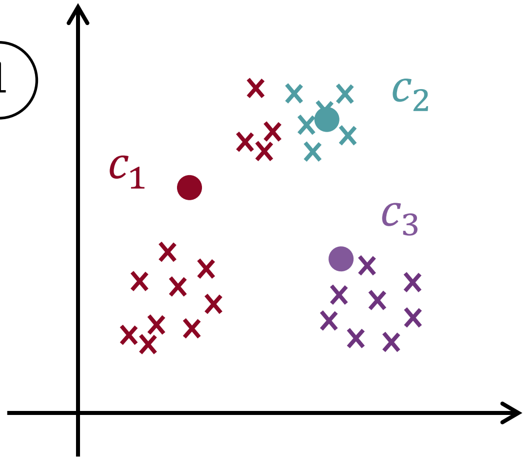
1., 2.



Algorithm

1. Initialize cluster centroids
2. Assign instances to their closest centroids

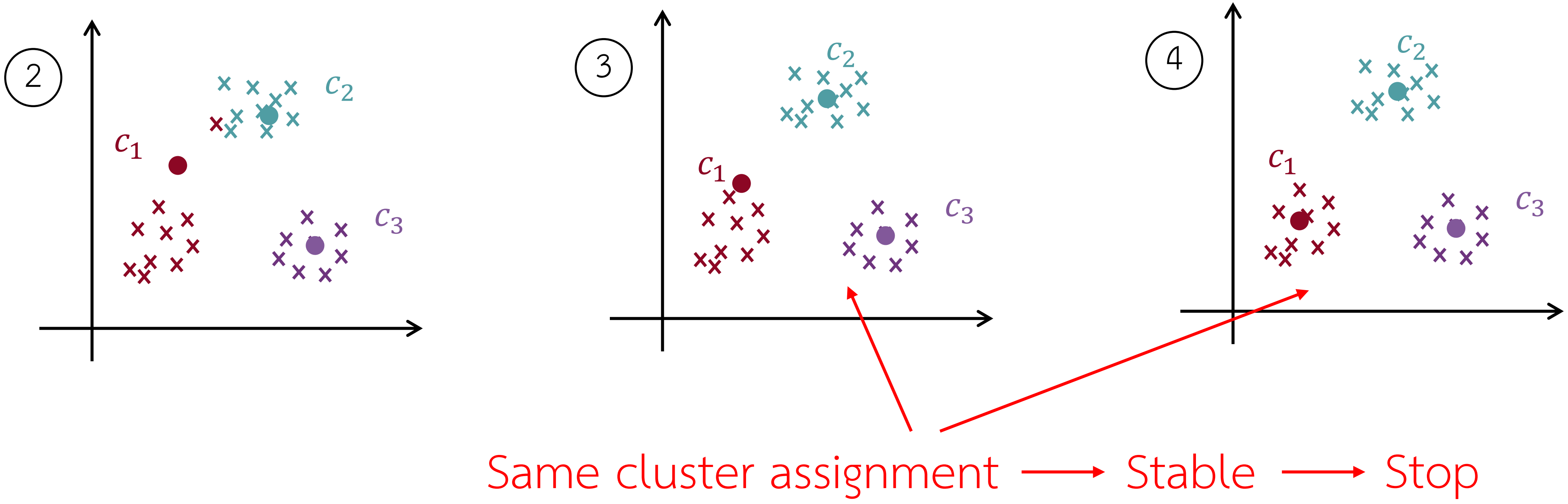
3., 4. 1



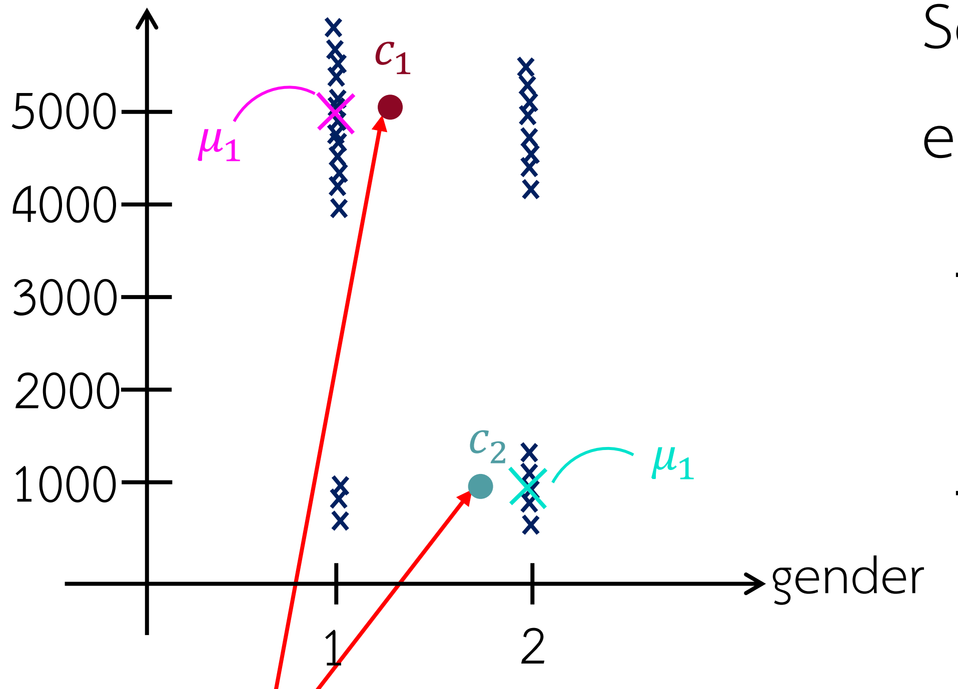
3. Find new centroids
4. Re-assign instances to their closest centroids

K-means

5. Repeat from 2. until the assignment becomes stable



K-modes



Mean centroid

Numerical values

Spending = 5000

Categorical values

Gender = 60% male 40% female?

Sometimes, the mean is not a good representation especially for a categorical value

K-medoids slow → algorithm

→ Choose the closest data point to the completed centroid

→ Choose the mode as the cluster representation and measure dissimilarities using Manhattan distance especially for categorical values

K-modes

K-modes

$$\underbrace{\underbrace{argmin \sum_{j=1}^{k} \sum_{x^{(i)} \epsilon S} \sum_{m \epsilon F_1} \left\| x_m^{(i)} - c_j \right\|_2}_{\text{Numerical features} \,\to\, \text{K-means}} + \underbrace{\gamma \sum_{n \in F_2} \left\| x_m^{(i)} - c_j \right\|_1}_{\text{Categorical features} \,\to\, \text{K-modes}}}_{\text{K-prototypes}}$$

Euclidean distance

Manhattan distance

weight

Numerical features

K-means

Categorical features

K-modes

K-prototypes

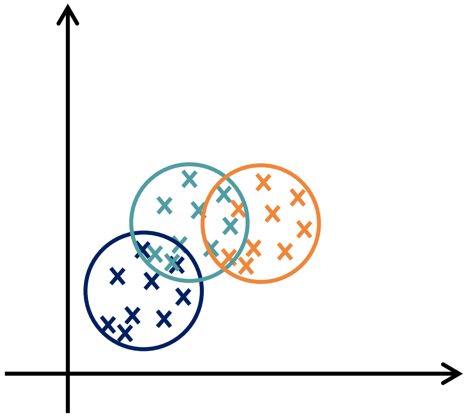
Finding a good k

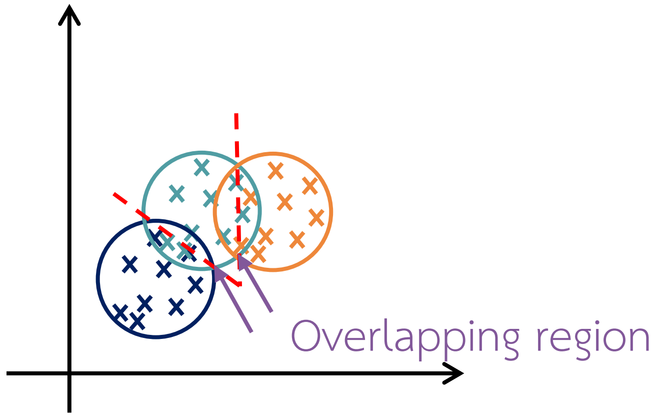
- Known number of clusters
  - Doesn't guarantee to produce desired clusters
- Try them all!!!
  - And choose the best k : requires evaluation metrics

Grid search: find k that has the highest cluster scores

```
for each k in range(...)
      s = find k clusters
      s_score = evaluate(s)
```

Overlapping cluster





One instance belongs to only one cluster

- Not suitable for overlapping structures

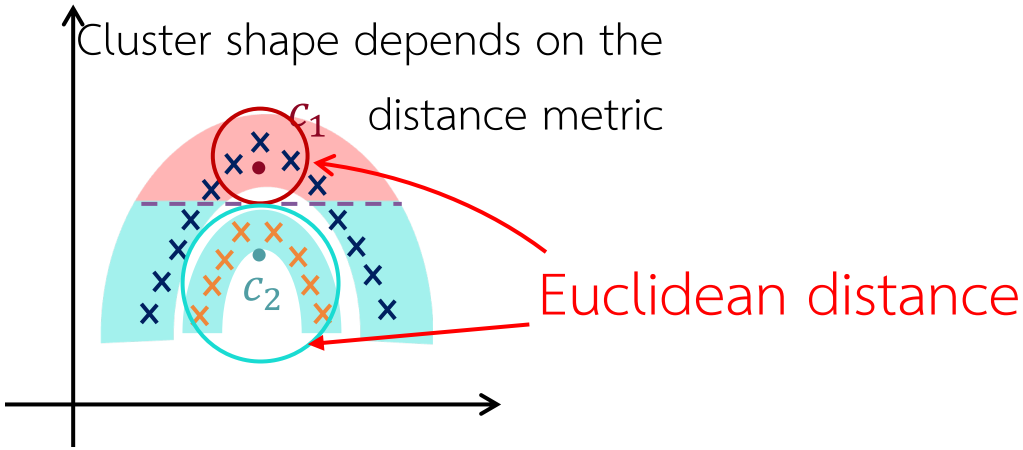
Solution

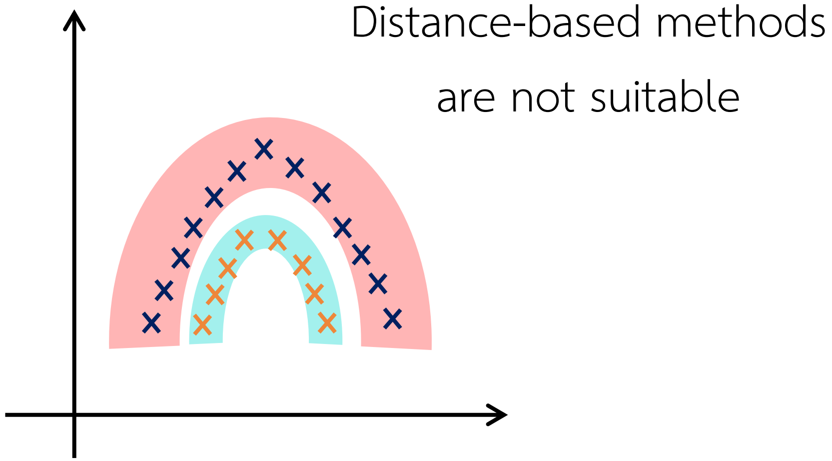
1. Map instances to higher dimensional space ⟶ tends to be more sparse

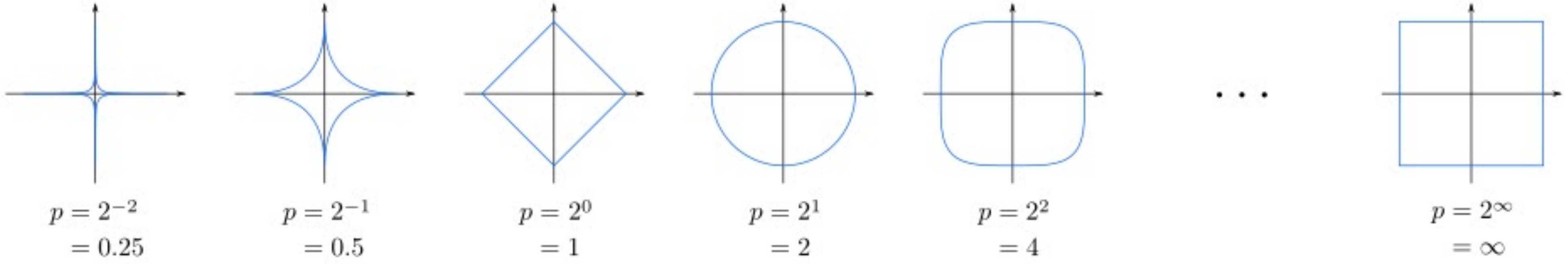
   ⟶ not overlap

2. Fuzzy solution: partial membership

Non-spherical distribution







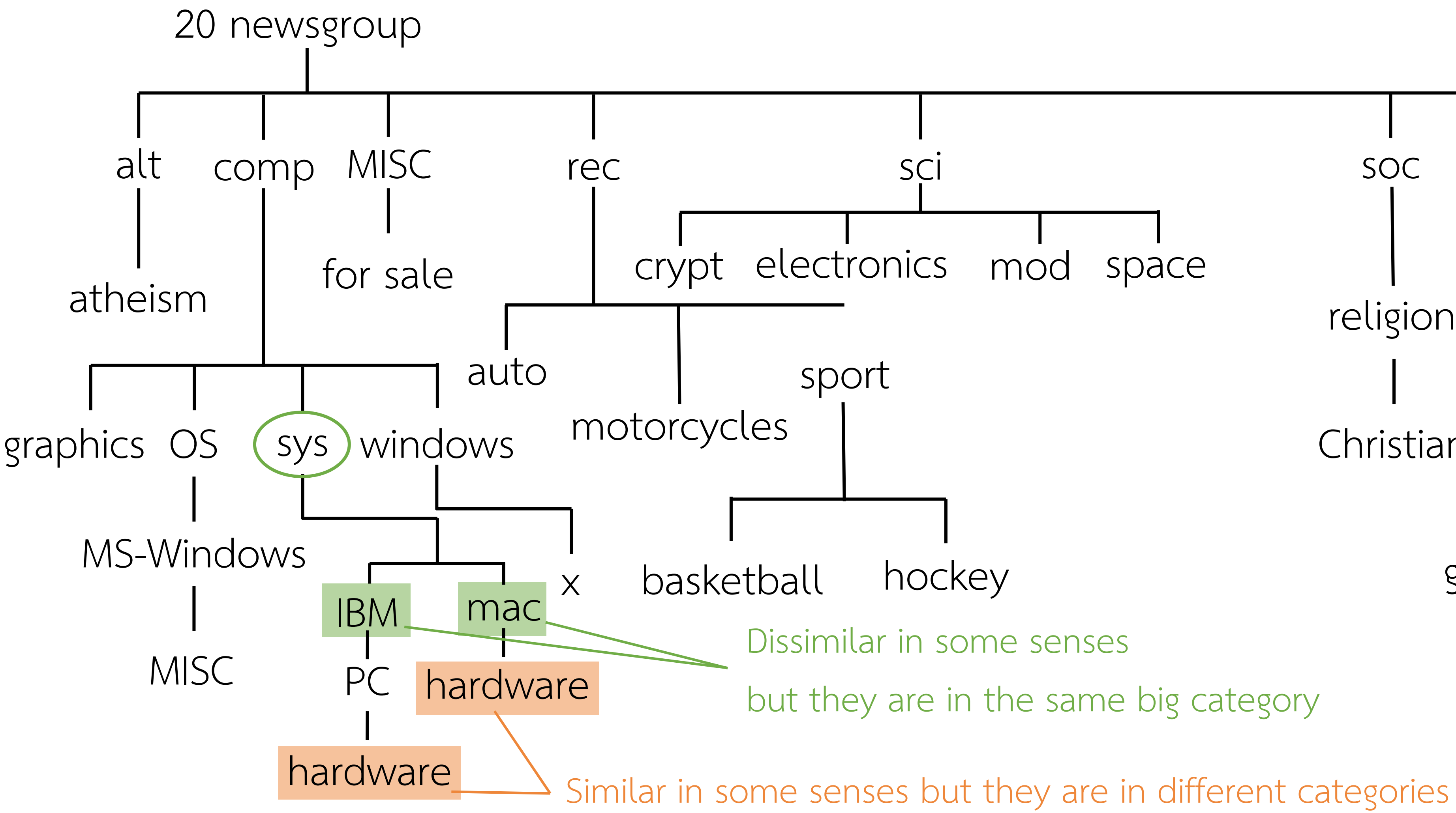
Project

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Dendrogram

Hierarchical structure



Hierarchical clustering

If we use k-mean to find clusters,

1. k = ?
   - Number of the leaf clusters?
   - Number of the top leaf clusters?
   - Number of all possible clusters?
2. Hierarchical structure
   - Splitting a big cluster into small clusters
     → Top-down construction
   - Merging small clusters into a bigger cluster
     → Bottom-up construction

How many clusters in each level?

Note that this is an unsupervised learning

Cluster dissimilarity metrics

- Euclidean distance: $\sum_i ||x_i - y_i||_2$
- Manhattan distance: $\sum_i ||x_i - y_i||_1$
- Mahalanobis distance: $\sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

Covariance matrix — eigen values

Eigen axes/cluster mean

Hierarchical clustering

Linkage criteria

- Single linkage clustering: $\min\{d(a,b): a \in A, b \in B\}$ Closest point between two clusters
- Complete linkage clustering: $\max\{d(a,b): a \in A, b \in B\}$ Farthest points between two clusters
- Unweighted average linkage clustering: $\frac{1}{|A|\cdot|B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$
- Weighted average linkage clustering: $d(i,j,k) = \frac{d(i,k)+d(j,k)}{2}$
- Sum of intra-cluster variance The new distance between the merged cluster and another cluster is the weighted average distance of its member clusters
- The increase in variance for the cluster being merged (Ward)

Hierarchical clustering

Bottom-up construction: Agglomerative clustering

- Merge smaller clusters into a larger clusters
- Based on the dissimilarity metric and linkage criterion

$O(n^3)$ quite slow

Have better algorithms

in some special cases

Top-down construction: Divisive clustering

- Find the split
- Based on the dissimilarity matric and linkage criteria

$O(2^n)$ very slow

Need good heuristics

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

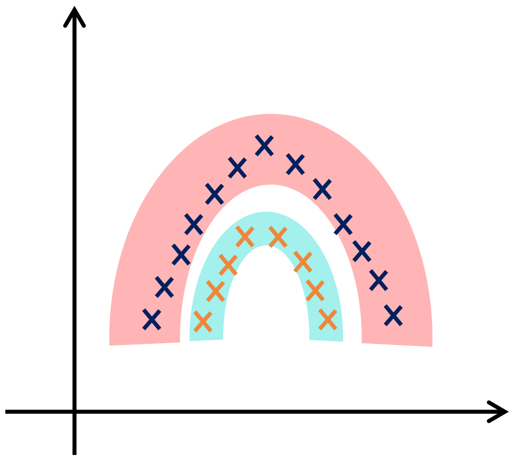
DBSCAN

Density Based Spatial Clustering of Application with Noise

- Given a point, find all neighboring points in the small radius
- If the neighboring area is dense enough, connect nodes together and form a cluster

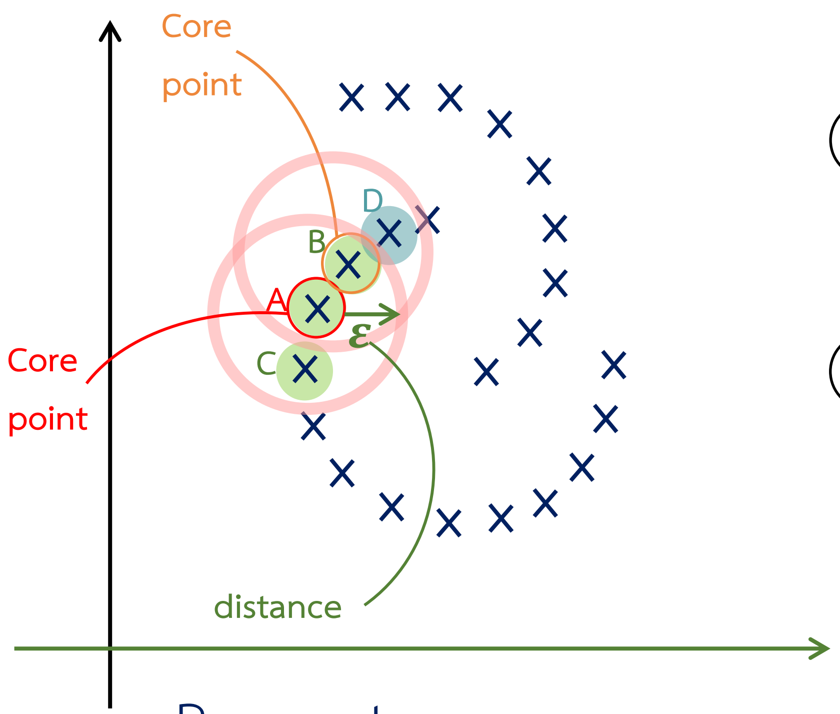
No pre-defined number of clusters. Allow non-spherical cluster.

Use cases



- Non-spherical structures
- An instance is closed to some of its neighbors in the same clusters ⟶ high density

DBSCAN



Parameters

- ε
- Min points

1
- A, B, C
  3 points in the distance ε from the core point A
- Add B, C to the cluster

2
- B, A, D
  3 points in the distance ε from the core point B
- Add D to the cluster

Repeat the process

Project

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Usecases

Market basket analysis

Transaction list

- Shampoo, conditioner, cola, chips
- Shampoo, conditioner, toothpaste, chips
- Shampoo, toothbrush, toothpaste, water
- Shampoo, conditioner toothbrush, cola
- Conditioner, toothpaste, cola, chips

Frequent patterns ⟶ promotional campaign

Medical diagnosis

- Symptoms ⟶ diagnosis

e.g. glucose level, insulin level,

blood pressure ⟶diabetes

Interestingness

Itemset: set of co-occurrence items

Transaction list: T

$$\underbrace{\overbrace{\text{Shampoo, conditioner,}}^{\text{Itemset}}\ \overbrace{\text{cola, chips}}^{\text{Itemset}}}_{\text{transaction}}$$

- Shampoo, conditioner, cola, chips

antecedent e.g. Shampoo

consequence e.g. Conditioner

(Association) Rule: $X \rightarrow Y$

If a customer buy X, they also buy Y

If there are symptom X, the diagnosis is Y

Association rules

We want to find interesting rules

- Support: How often we see itemset X in the transaction list T

$$Supp(X) = \frac{|X \cap T|}{|T|}$$

High support $\longrightarrow$ interesting pattern

- Confidence: How often we see rule $X \rightarrow Y$ in the transaction list T

$$Conf(X \rightarrow Y) = \frac{Supp(X \cap Y)}{Supp(x)} \text{ or } P(Y|X)$$

- Lift: Independence of X and Y

$$\boxed{P(X|Y)P(Y) \text{ or } P(Y|X)P(X)}$$

$$lift(X \rightarrow Y) = \frac{Supp(X \cap Y)}{Supp(X) \cdot Supp(Y)} \text{ or } \frac{\boxed{P(X \cap Y)}}{P(X) \cdot P(Y)}$$

lift = 1	X and Y are independent
lift > 1	X and Y are dependent, possible consequent
lift < 1	X and Y are substitution to each other

Association rules

- All-confidence

$$allconf(X \to Y) = \frac{Supp(X \cap Y)}{\max(Supp(X), Supp(Y))}$$

- Cosine

$$cos(X \to Y) = \frac{Supp(X \cap Y)}{\sqrt{Supp(X) \cdot Supp(Y)}}$$

- Conviction: dependency of X and Y

$$conv(X \to Y) = \frac{1 - Supp(Y)}{1 - conf(X \to Y)} = \frac{P(X) \cdot P(\bar{Y})}{P(X \cap \bar{Y})}$$

Apriori algorithm

Frequency counting algorithm

- Constructing small to large itemsets e.g. $|x|$ in $1 \ldots n$
- Filter cut patterns with low support
- Extend the pattern length

Association rules

Example

For each itemset

Count for the pattern

Frequency count

Itemsets

- $\{1, 2, 3, 4\}$
- $\{1, 2, 4\}$
- $\{1, 2\}$
- $\{2, 3, 4\}$
- $\{2, 3\}$
- $\{3, 4\}$
- $\{2, 4\}$

$|x| = 1$

x	support
$\{1\}$	3
$\{2\}$	6
$\{3\}$	4
$\{4\}$	5

$|x| = 2$

x	support
$\{1, 2\}$	3
~~$\{1, 3\}$~~	~~1~~
~~$\{1, 4\}$~~	~~2~~
$\{2, 3\}$	3
$\{2, 4\}$	4
$\{3, 4\}$	3

$|x| = 3$

x	support
~~$\{1, 2, 3\}$~~	low support
~~$\{1, 2, 4\}$~~	low support
~~$\{1, 3, 4\}$~~	low support
$\{2, 3, 4\}$	2

For each pattern

Extend the previous pattern

Filter out patterns with low support

e.g. support < 3

Market-basket analysis (Affinity analysis)

Understand the purchase behavior of a buyer

e.g. beers are often bought with diapers

Then

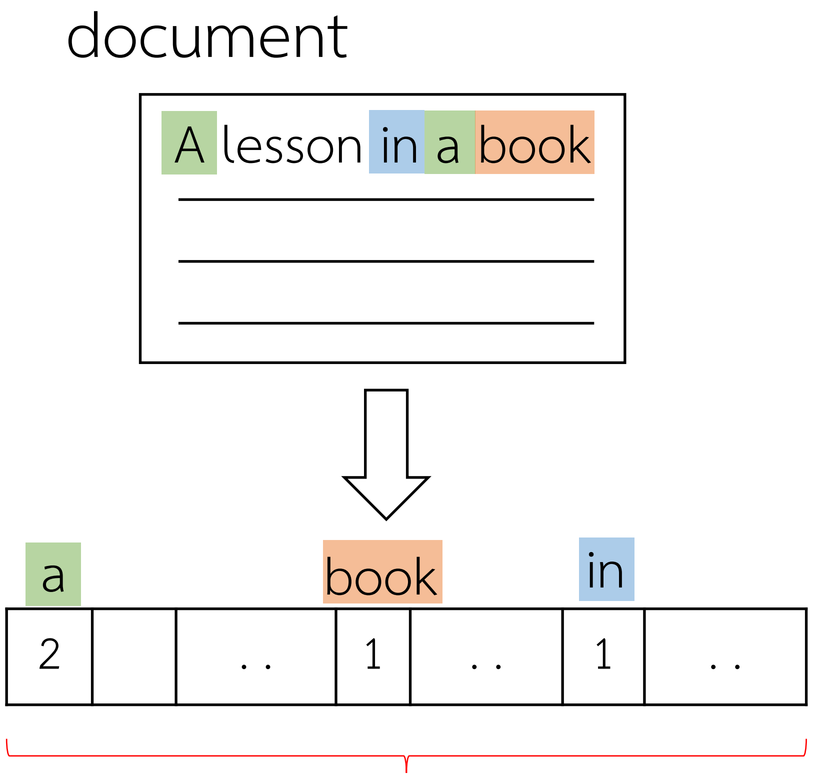
- Cross-selling
- Customer segmentation

Project

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Bag-of-word

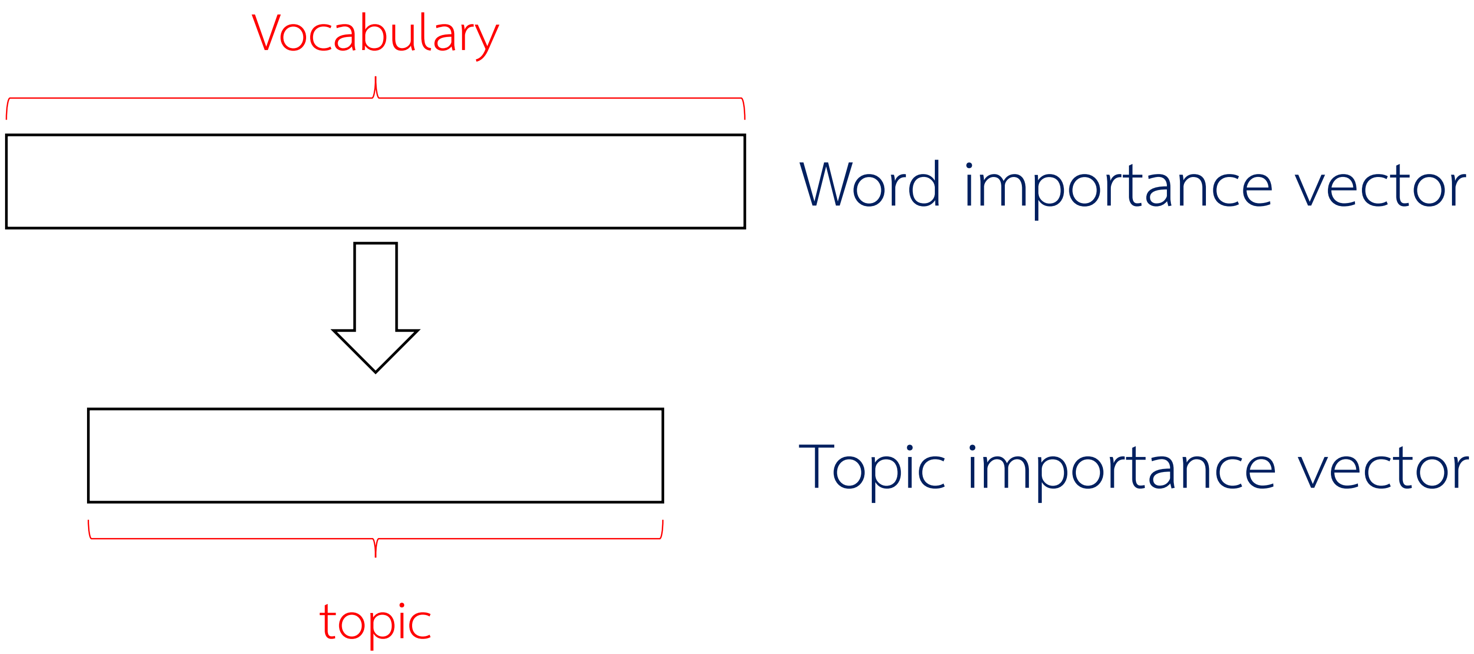


Frequency count vector

word importance vector

Vector dimension = size of vocabulary → very large

Topic model



E.g. sports → score, win, lose, soccer, tennis, referee, foul, etc.
topic

Topic model

How?

1. Matrix factorization

$$\begin{bmatrix} \text{---}\ d^{(1)}\ \text{---} \\ \text{---}\ d^{(2)}\ \text{---} \\ \vdots \\ \text{------------} \end{bmatrix} |D|\times|V| = \begin{bmatrix} \text{---}\ d^{(1)}\ \text{---} \\ \text{---}\ d^{(2)}\ \text{---} \\ \vdots \\ \text{------------} \end{bmatrix} |D|\times|T| \begin{bmatrix} \text{---}\ t^{(1)}\ \text{---} \\ \text{---}\ t^{(2)}\ \text{---} \\ \vdots \\ \text{------------} \end{bmatrix} |T|\times|V|$$

Number of documents

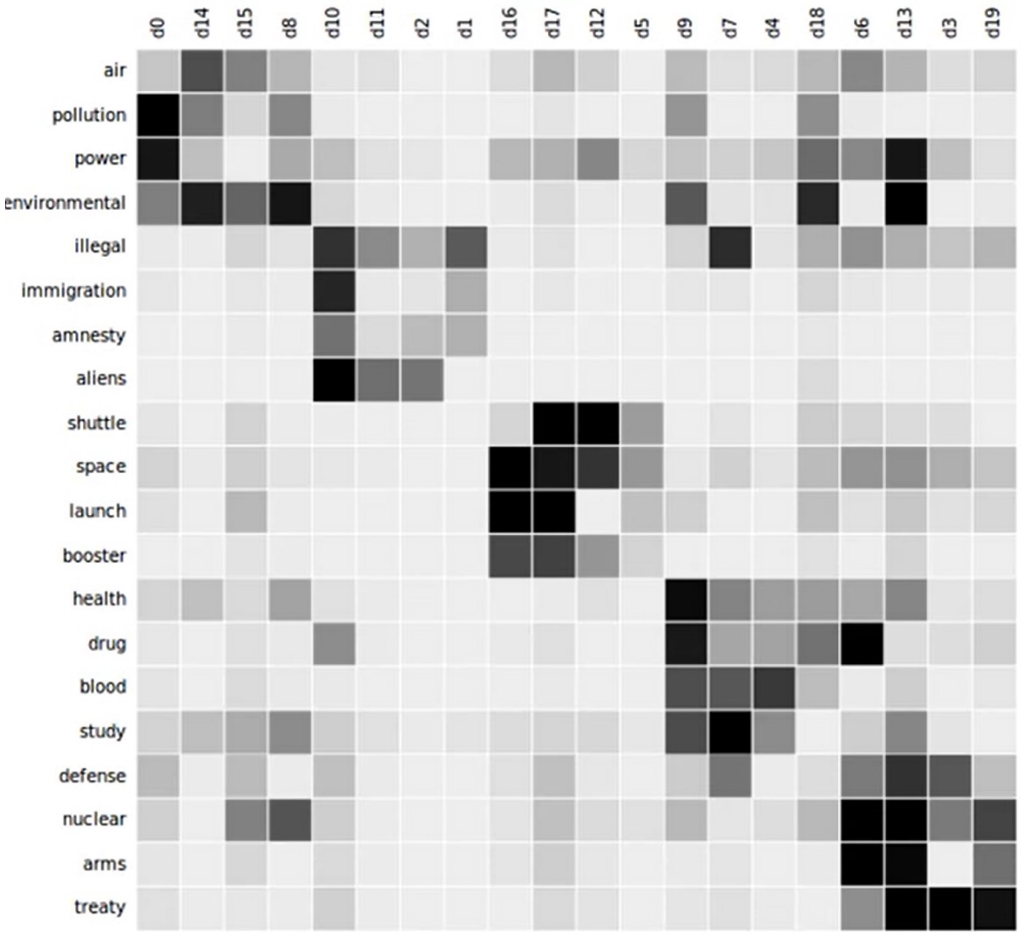
Vocabulary size

Number of topics

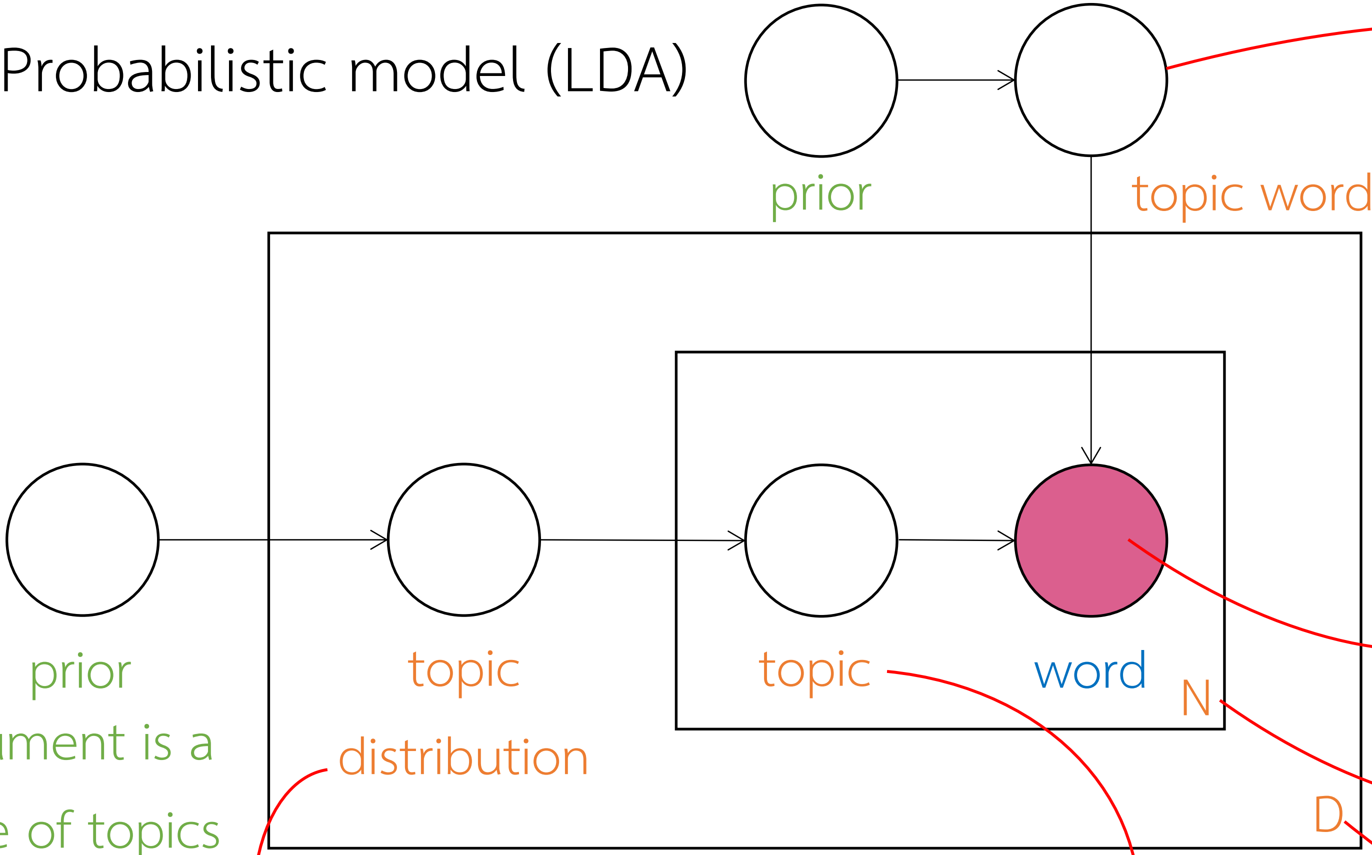
Topic e.g. sports

Words related to the topic have high relevance score in this topic

Topic model


d0
d14
d15
d8
d10
d11
d2
d1
d16
d17
d12
d5
d9
d7
d4
d18
d6
d13
d3
d19
air
pollution
power
environmental
illegal
immigration
amnesty
aliens
shuttle
space
launch
booster
health
drug
blood
study
defense
nuclear
arms
treaty

2. Probabilistic model (LDA)



e.g.

Tdpic / Word	Soccer	Foul	Parliament	Win
Sports	0.4	0.3	0.1	0.2
politics	0.1	0.1	0.6	0.2

Word $w^{(i,j)}$ seen in document $d^{(i)}$

e.g. win

Each document $d^{(i)}$ has $n^{(i)}$ words

We have D document

A document is a mixture of topics

Mixture of topics in document $d^{(i)}$

e.g. sports politics

0.4 0.6

Topic of word $w^{(i,j)}$

e.g. sports

Project

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Finding data insight

Customer segmentation/ads target

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

Supervised vs unsupervised learning

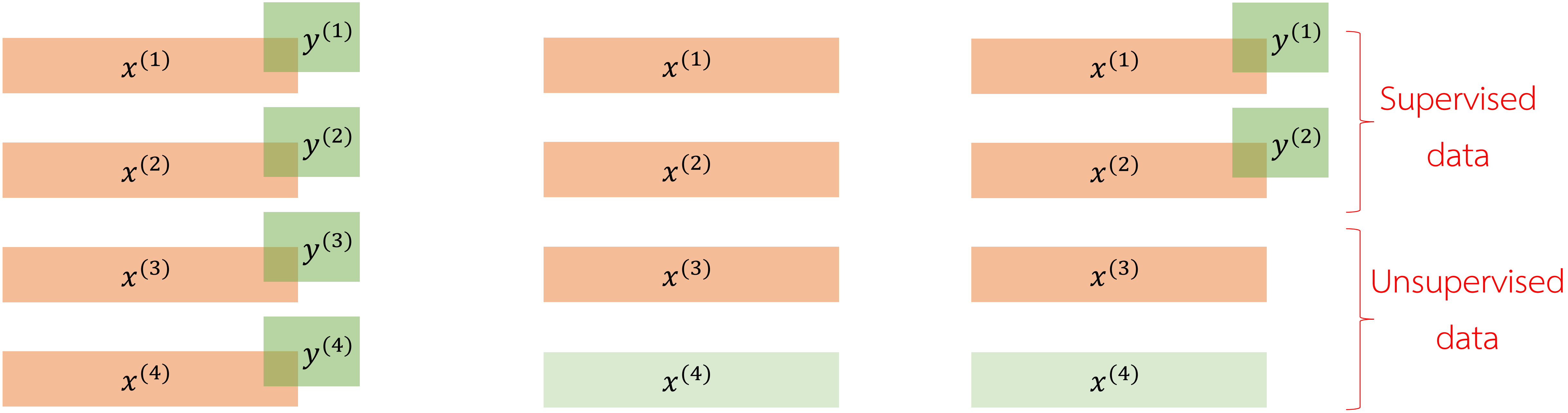
Supervised learning

$$y = \hat{f}(x)$$

Unsupervised learning

$$y = ?$$

Semi-supervised learning



Labeled data

Unlabeled data

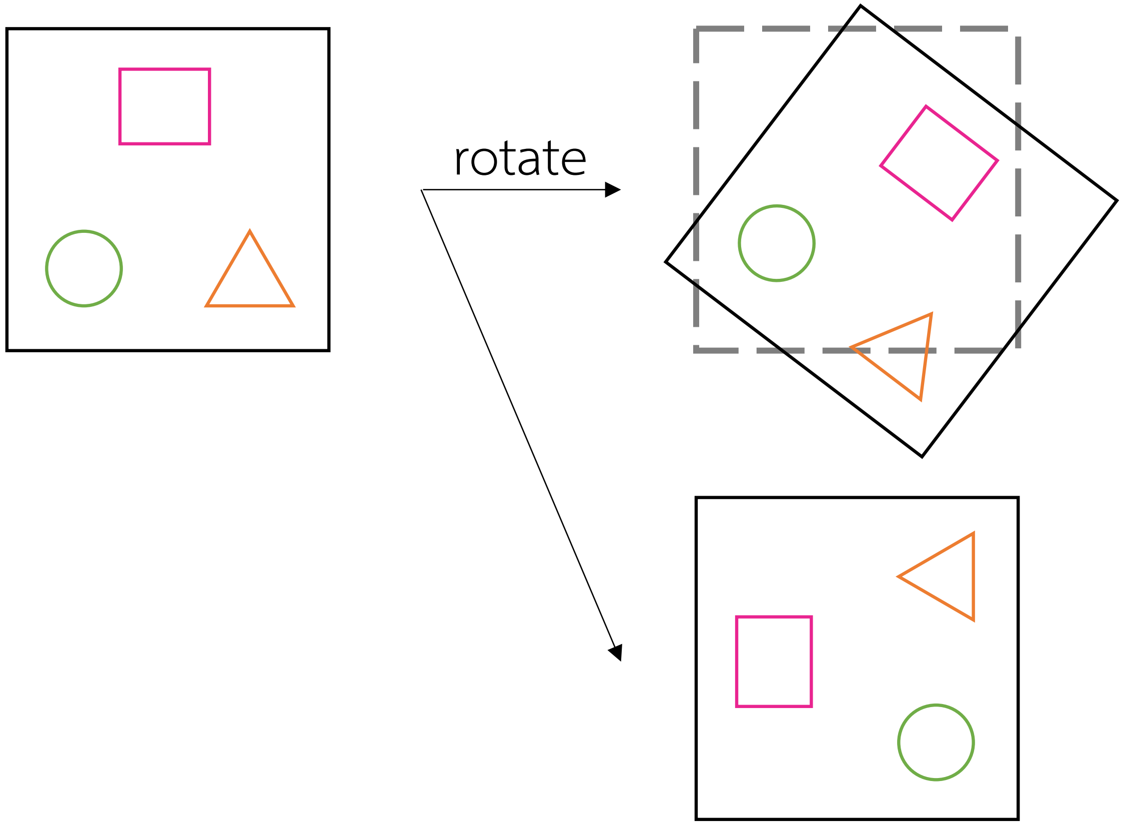
Semi-supervised learning

- Labeled data is expensive
- Unlabeled data is cheap

However, it is hard to build a good model without supervision.

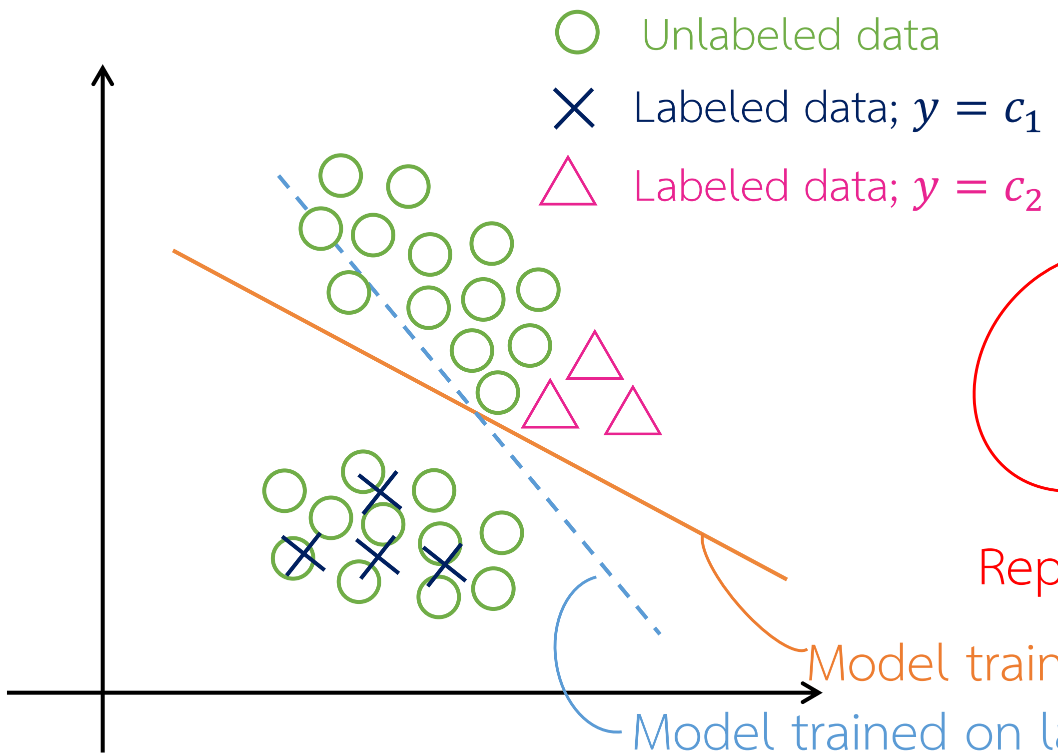
Mix them together

Data augmentation



- Add noise(random, Gaussian)
- Image translation, transformation
- Web crawling/scraping

Getting more pseudo-labeled data



pseudo-label

1. Initialize the model
2. Use the model to update labels of unlabeled data
3. Re-train the model with labeled and pseudo-labeled data

Repeat

Module Outline

- Introduction
- Similarity, dissimilarity and evaluation metrics
- Dimensionality reduction
- Distance-based learning
- Hierarchical clustering
- Density-based clustering
- Association rules
- Topic model
- Try them all
- Semi-supervised learning
- Summary

What, when, why do we need unsupervised learning

- We don’t have labeled data
- We want to find insights in the data (we have some assumptions)