

Dimensionality Reduction

CASA0006: Data Science for Spatial Systems

Huanfa Chen

Recap

What we already know

We can handle data

Using a database accessed through SQL, and tools such as Pandas we can take raw, unstructured data through to something useful

We can analyse data

Clustering, Regression, Classification

Today we extend our skills in data analysis, exploring the use of **dimensionality reduction** methods

Data Mining

The toolbox

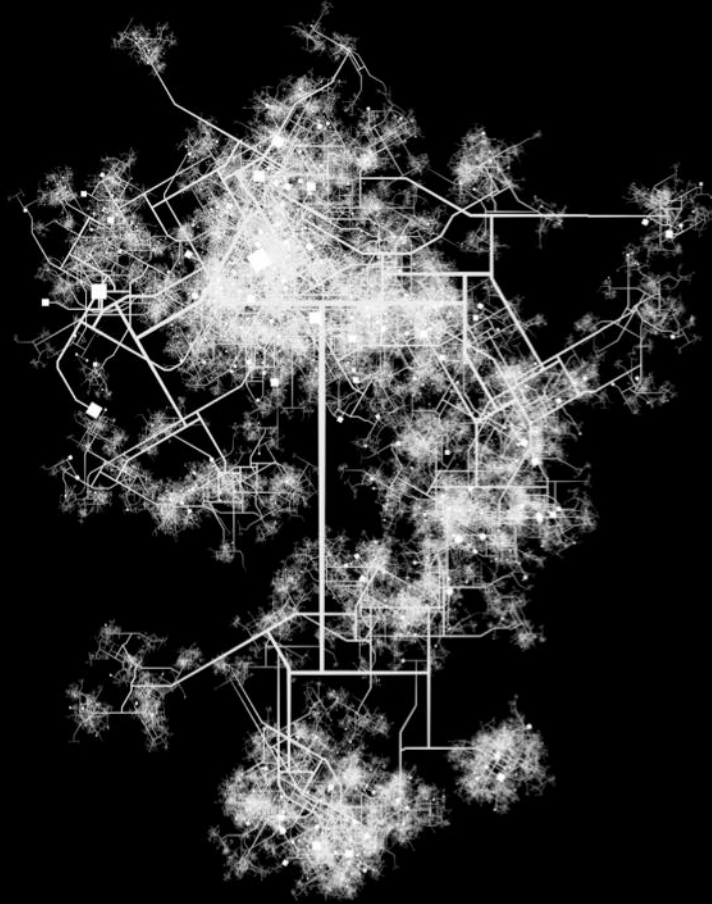
The approach to take towards mining your data depends on what you want to understand from it



Unsupervised = Unlabelled

Supervised = Labelled

Outline



1. Dimensionality Reduction
2. Methods
 - a. Principal Component Analysis
 - b. t-SNE
3. Summary

Dimensionality Reduction

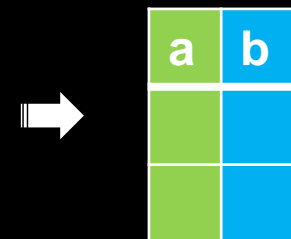
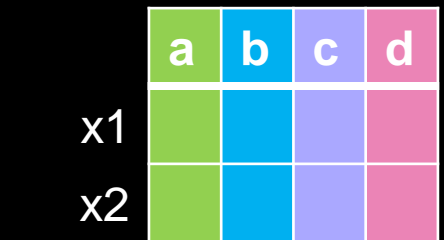
Understanding influential factors



- The process of reducing the number of variables under considerations by obtaining a set of relevant factors
- It is *unsupervised learning*, meaning there is no ground truth to validate the result

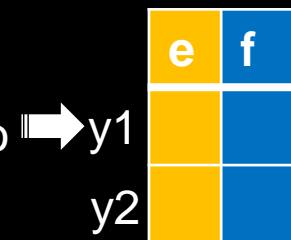
1. Feature selection

- Stepwise regression
- LASSO



2. Feature extraction

- Transform the data in the high-dimensional space to fewer dimensions
- Data transformation: linear or non-linear



$$e = f(a, b, c, d)$$

$$f = f(a, b, c, d)$$

Dimensionality Reduction

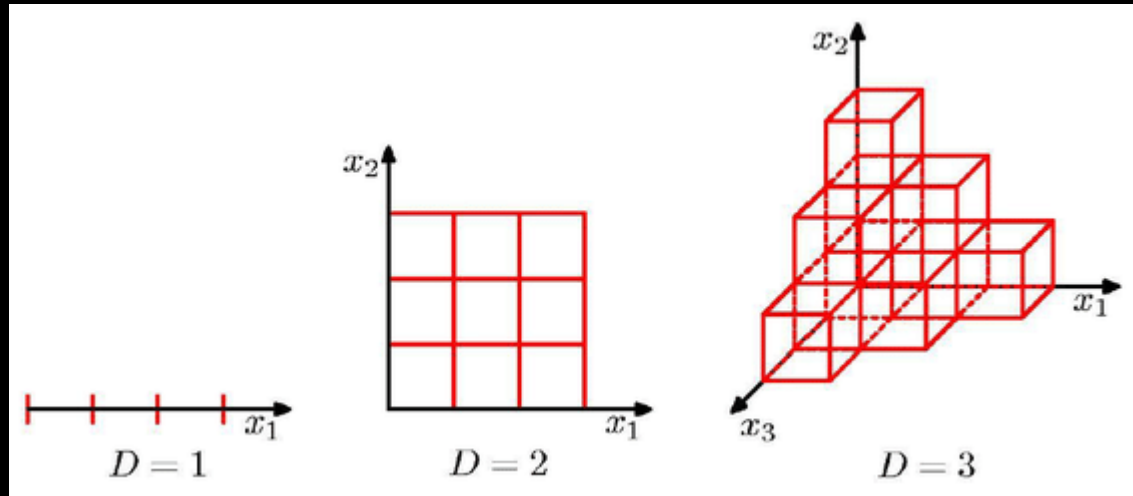
Motivations

Perspective	Details
Visualisation	To visualise the data when reduced to low dimensions such as 2D or 3D
Computation	To reduce the time and storage space
Modelling	To remove multi-collinearity; to improve the interpretation of model parameters
Others	To avoid the curse of dimensionality

Curse of Dimensionality

A large number of possible values

1. Possibilities are exponential in the dimensionalities



Possible values

3^1

3^2

3^3

Curse of Dimensionality

Distance functions

2. Distance functions become meaningless in high dimensions

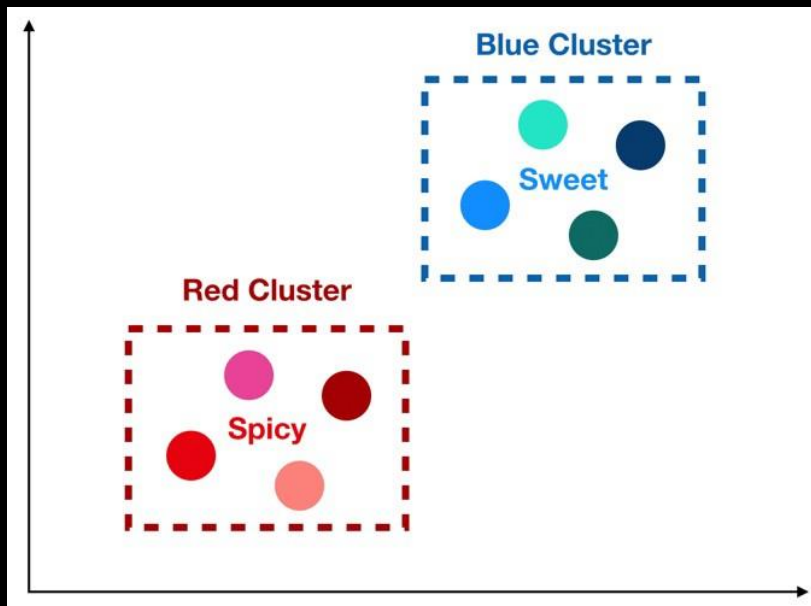
Example

Clustering of candies from colours

(Flavours: sweet, spicy)

Colour definition 1: 8 colours

- The Euclidean distance between each pair?
- How many clusters?



	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
Red	1	0	0	0	0	0	0	0
Maroon	0	1	0	0	0	0	0	0
Pink	0	0	1	0	0	0	0	0
Flamingo	0	0	0	1	0	0	0	0
Blue	0	0	0	0	1	0	0	0
Turquoise	0	0	0	0	0	1	0	0
Seaweed	0	0	0	0	0	0	1	0
Ocean	0	0	0	0	0	0	0	1









Curse of Dimensionality

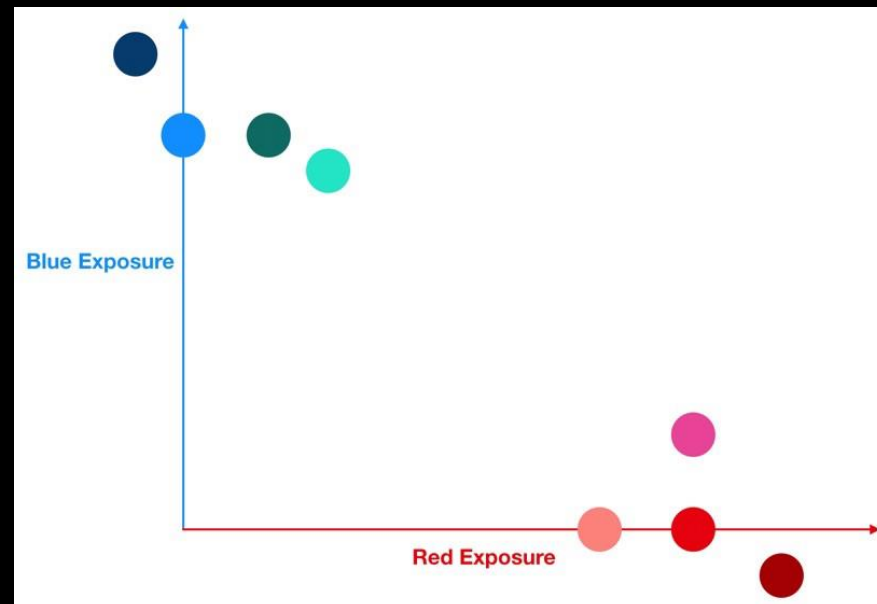
Colour definition 1: 8 colours

- 8 clusters, 4 as spicy and 4 as sweet
- Zero insight into how colours are related with flavour

Colour definition 2: red-blue features (dim reduction)

- Given a colour, break it into a red-blue combination
- Clustering using Euclidean distance

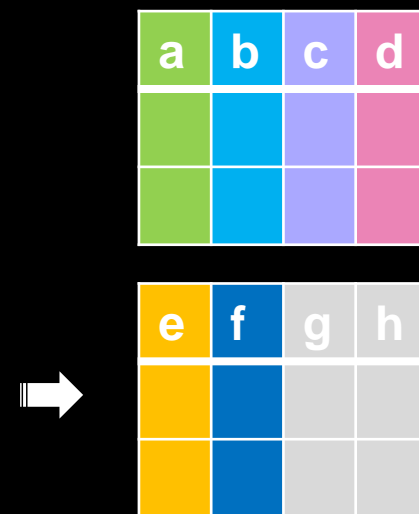
	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
	1	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0
	0	0	1	0	0	0	0	0
	0	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0
	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	1



Principal Component Analysis

Linear combination of features

- Principles (“*Keep the variance*”)
 - Find a new set of dimensions, such that each new dim is a linear combination of the original, and all dims are linearly independent (reduce multicollinearity)
 - Rank all new dimensions according to the variance of data. The more variance (or *spreading out*), the higher importance.
 - Keep the first k new dimensions. (usually $k=2,3$)



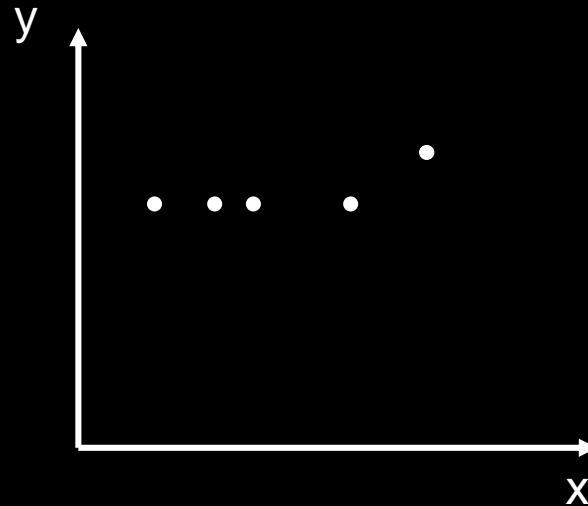
Name	New dim	combination	Var
1 st component	e	$0.5a + 0.6b + 0.1c + 0.3d$	0.8
2 nd component	f	$0.6a + 0.1b + 0.2c + 0.5d$	0.1
	g		0.05
	h		0.05

Principal Component Analysis

Linear combination of features

- Variance: quantifying spread, or the difference between points.

$$Variance(x) = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n}$$

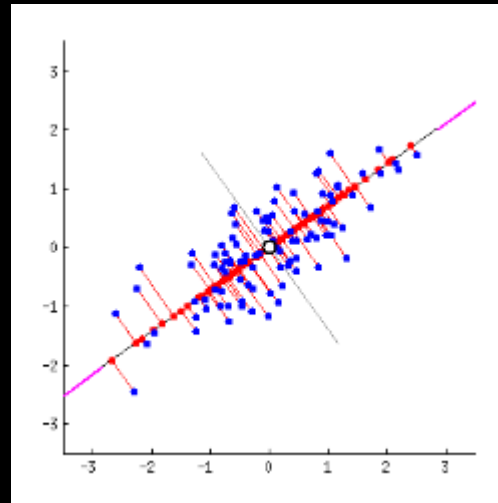
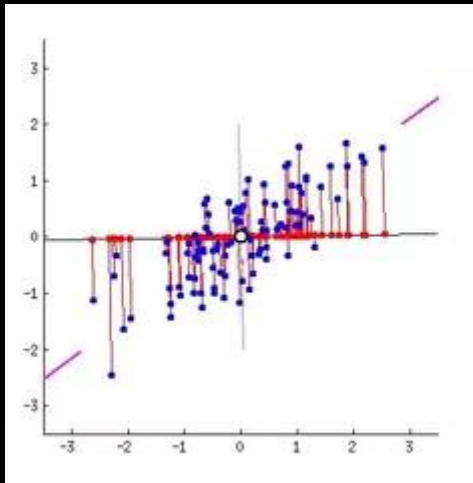


Which dimension has a larger variance? x or y?

Principal Component Analysis

Linear combination of features

- Projecting points to new dimensions while keeping variance



Project 2-D data to 1-D.

Which projection leads to a smaller loss of variance?

How many factors of PCA to retain?

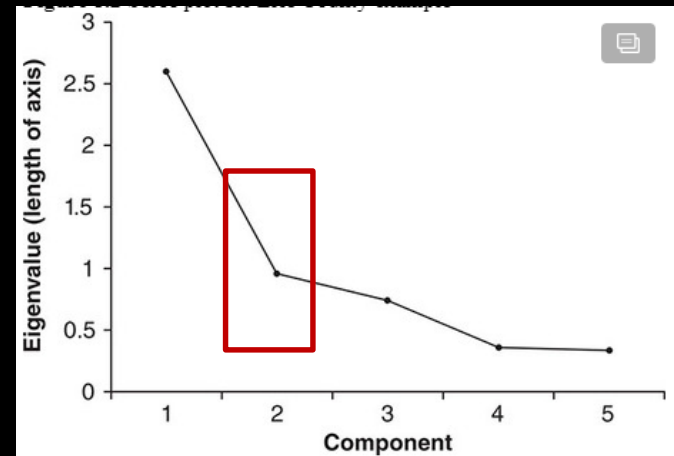
Three 'rules of thumb'

1. [For visualisation] Using two or three factors
2. To retain components with eigenvalues greater than one (would fail if all/most eigenvalues are smaller than one)
3. To plot the eigenvalues on the y axis and the factor number on the x axis of a graph (termed a *scree plot*), and then locate a point just before the graph flattens out (like the elbow method)

Example

Table 8.2 Variance explained by each component

Component	Total variance explained					
	Extraction sums of squared loadings			Rotation sums of squared loadings		
	Total	% of variance	Cumulative %	Total	% of variance	Cumulative %
1	2.602	52.032	52.032	1.035	20.707	20.707
2	.957	19.149	71.181	1.032	20.637	41.344
3	.741	14.826	86.007	1.018	20.358	61.702
4	.362	7.244	93.251	1.005	20.110	81.812
5	.337	6.749	100.000	.909	18.188	100.000



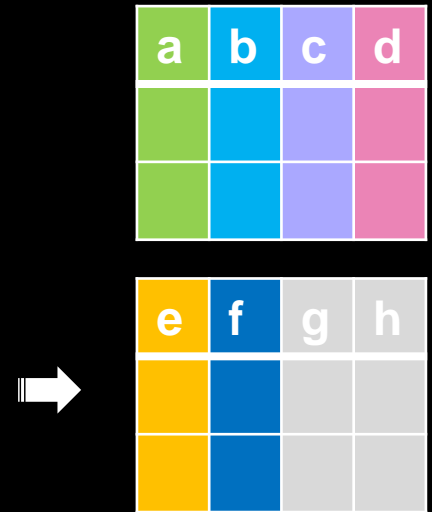
Rule 2: choose $k = 1$;

Rule 3: choose $k = 2$;

Principal Component Analysis

Some notes

- PCA is sensitive to the relative scaling of the original variables (Z-score normalisation needed)
- Good interpretation: each component is a linear combination of features
- It is guaranteed that the new features are uncorrelated – no more multicollinearity concerns.
- Can be used for visualising the data, or checking the clustering results
- Can be used as an input to clustering/classification/regression.



t-SNE

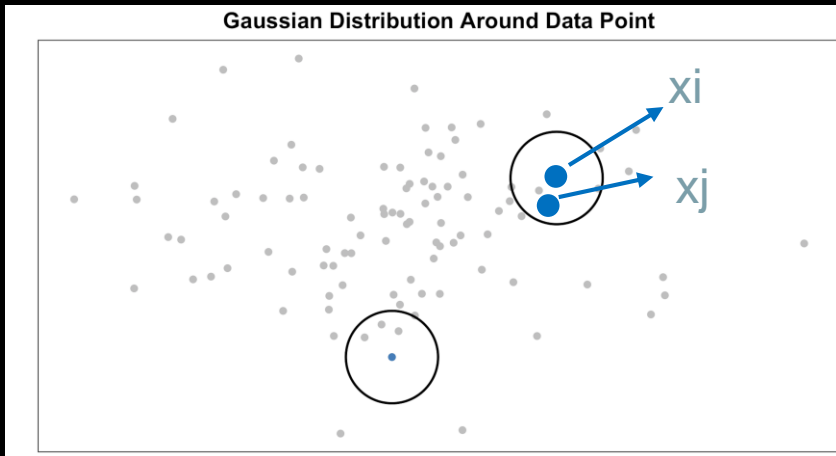
Non-linear transformation of features

	PCA	t-SNE
Principle	Preserving large pairwise distance	Preserving small pairwise distance or local pattern
Meaning of new dimensions	Linear combination of original dims (Clear and important)	Unclear and unimportant
Flexibility	The result is deterministic and can't be adjusted	The result is probabilistic and can be adjusted using the parameter. There is a trade-off between local or global patterns.
As input to other analysis?	Yes	No. Mainly used for exploration and visualisation
Computation cost	Lower than t-SNE	High

t-SNE

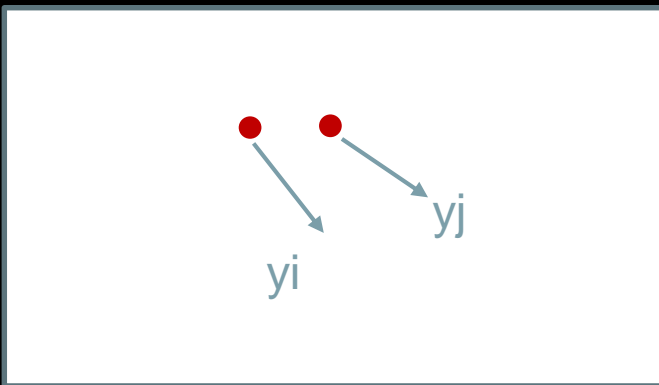
Non-linear transformation of features

Original data
(N dim)

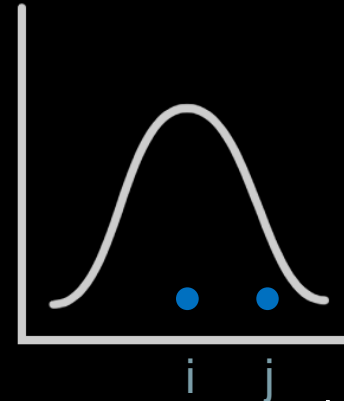


NOTE: x_i is fixed, and y_i is what to learn from t-SNE.

Projected data
(1 D)

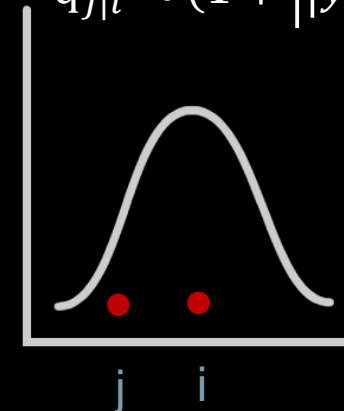


Conditional probability: given point i , how likely is to observe point j ? Depending on the distance between i and j



$$p_{j|i} \propto \exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)$$

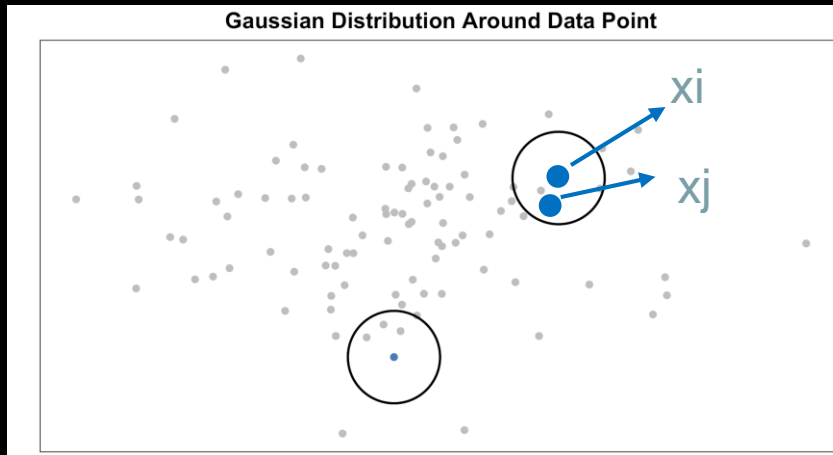
$$q_{j|i} \propto (1 + \|y_i - y_j\|^2)^{-1}$$



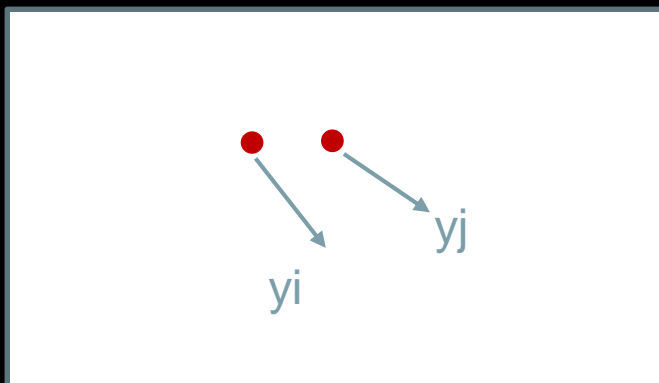
t-SNE

Non-linear transformation of features

Original data
(N dim)



Projected data
(2 D)



We can calculate all pairwise similarity in the original and projected space.

Denoted as P and Q.

The objective here is to minimize the divergence of two distributions (we want two distributions to be close)

The problem becomes:
Choose y_i to minimize $KL(P||Q)$

Kullback–Leibler divergence

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

t-SNE

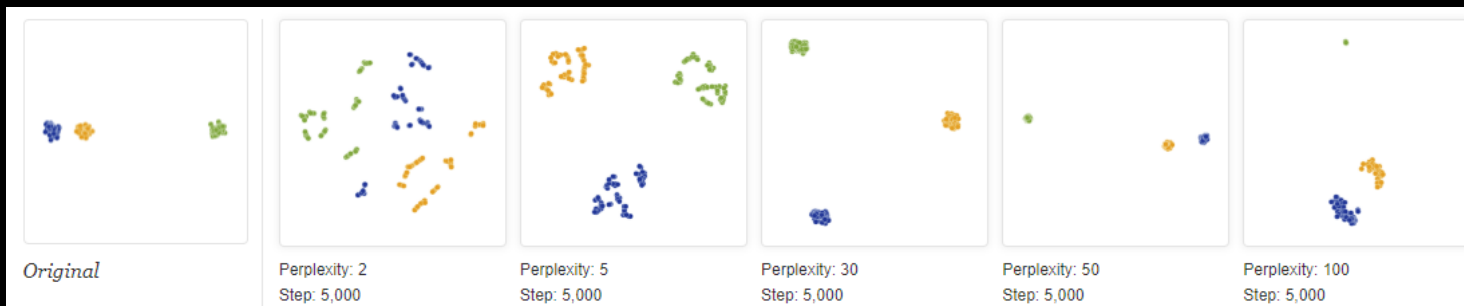
Notes

- When using t-SNE, the only parameter to adjust is perplexity.
- It is an estimate of the number of close neighbours each point has. It roughly regulates the balance between local and global patterns. Typical values are between 5 and 50.
- To get the most from t-SNE, analyse multiple plots using different perplexity.
- t-SNE is used to visualise high-dimensional data. Don't use the t-SNE results for further analysis (clustering, classification, or regression).

t-SNE

Notes

- Distances between clusters might not mean anything.



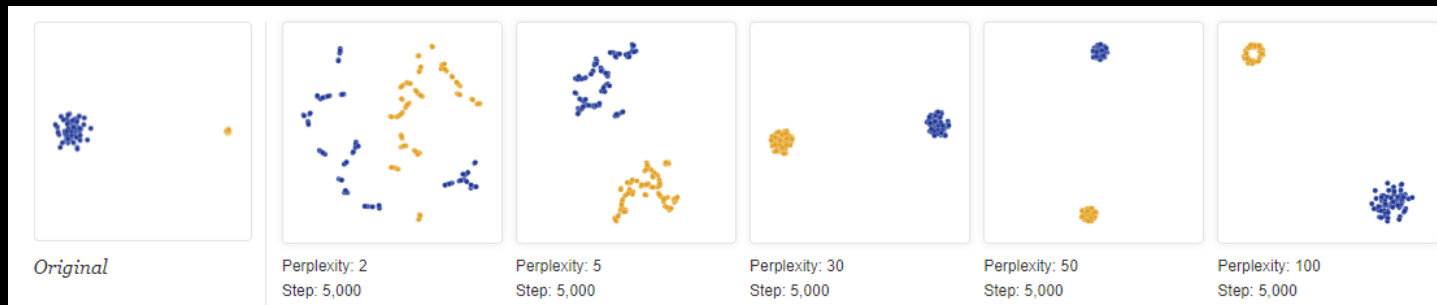
How to Use t-SNE Effectively

<https://distill.pub/2016/misread-tsne/>

t-SNE

Notes

- The extent of a cluster in t-SNE plot mean nothing.
- It may expand dense clusters and contract sparse ones.



How to Use t-SNE Effectively

<https://distill.pub/2016/misread-tsne/>

Summary

- Dimensionality Reduction



- PCA: linear transformation of x . Used for visualisation and as input to other analysis.
- t-SNE: non-linear transformation of x . Mainly used for visualisation.



Thank You
Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk

Workshop

Classification

- In this workshop you will extend your skills in data mining by learning PCA and t-SNE
- Once again you'll be using the Python sklearn library
- Again, you're not expected to understand all of the maths and computation, only the usefulness and application of these approaches.
- **Download this week's Python Notebook from Moodle, open it in Anaconda and work through.**