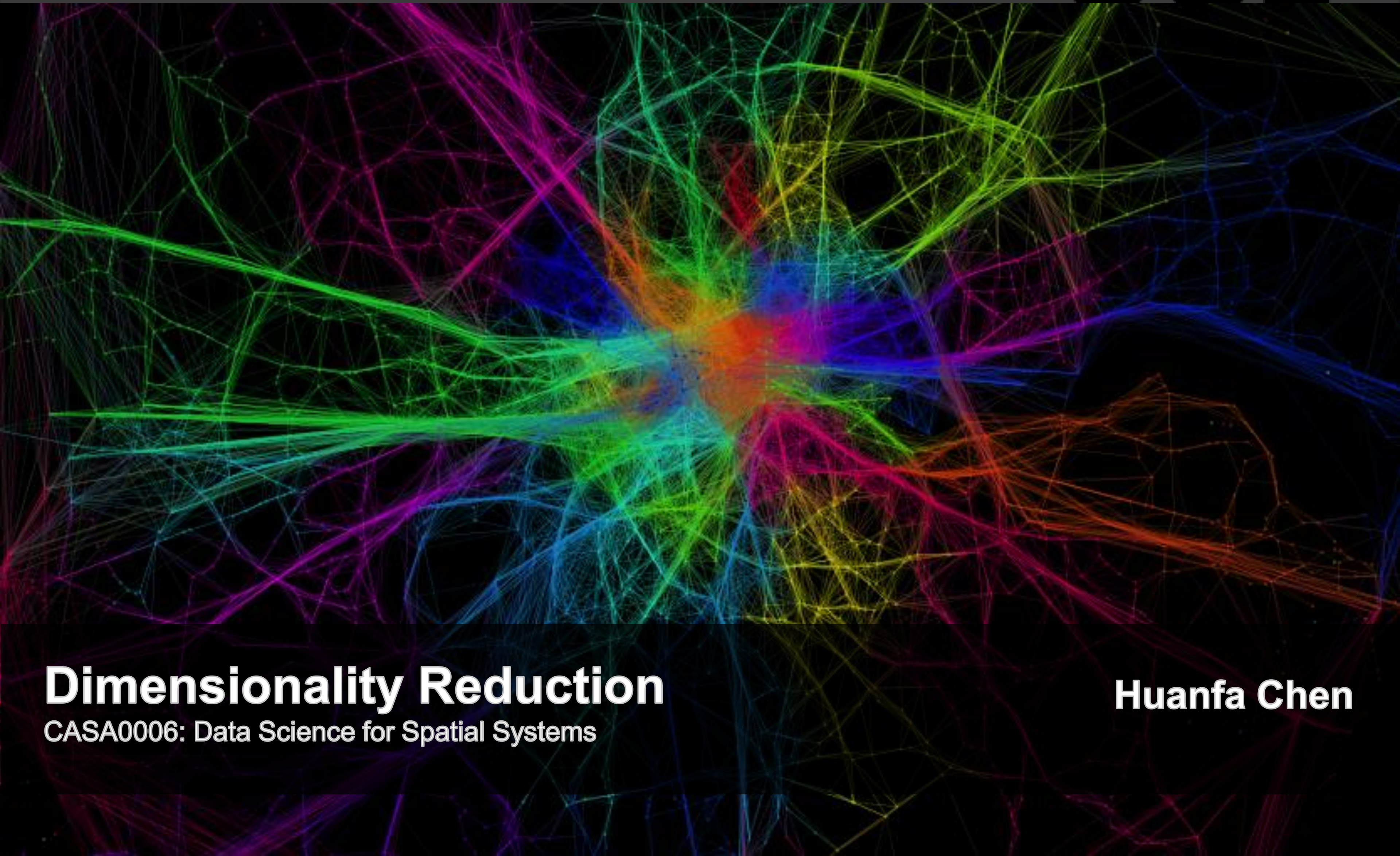


Dimensionality Reduction

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 explore the use of **dimension reduction** methods

Data Mining
The toolbox

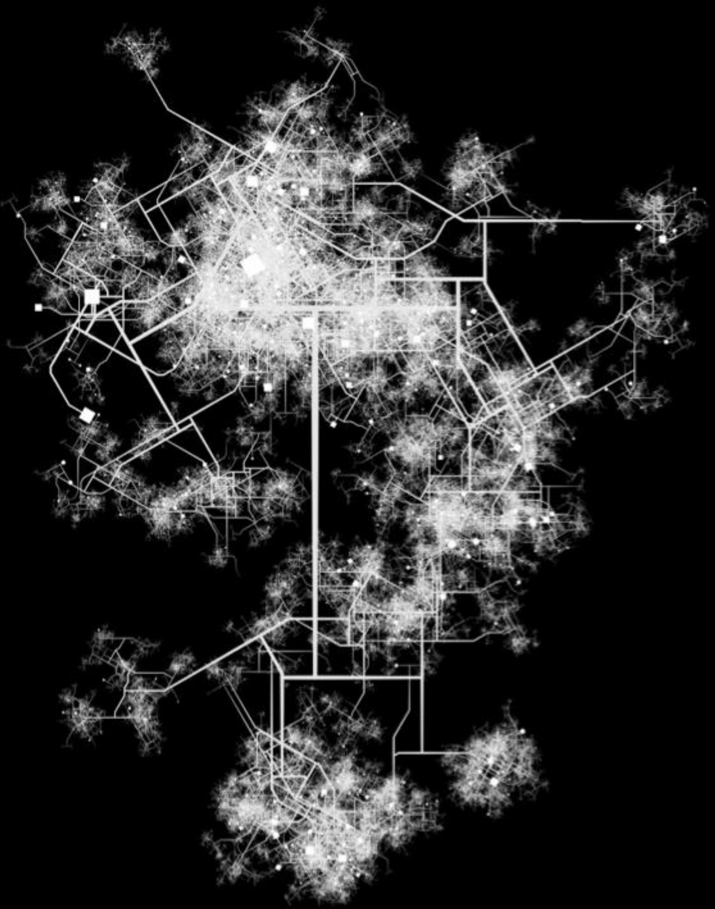
The approach to take towards mining your data depends on what you want to understand from it

Input Dataset ➡

Method		Output
Clustering	⟶	Creation of Groupings
Regression	⟶	Identify Data Relationships
Classification	⟶	Identify Discrete Class
Dimensionality Reduction	⟶	New representations in low-dim space
Association Rule Mining	⟶	Identify Dependencies
Anomaly Detection	⟶	Identify Outliers

Unsupervised = Unlabelled
Supervised = Labelled

Outline

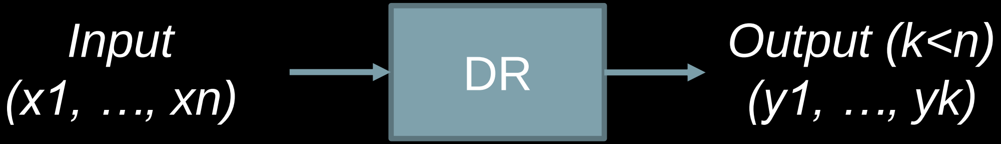


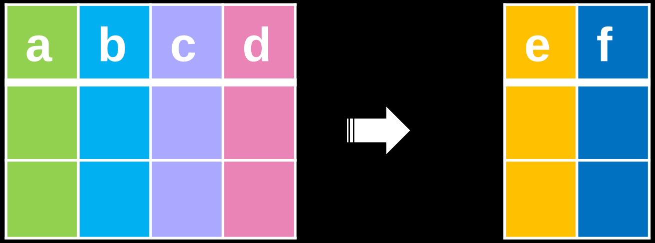
Dimensionality reduction

Dimensionality Reduction

New representations in lower-dim space

- 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





Dimensionality Reduction

New representations in lower-dim space

Two classes of DR

1. Linear DR
    1. each new dimension is a linear function of the original dimensions
    2. Example: PCA

2. Non-linear DR
    1. Example: kernel PCA, LLE

Dimensionality Reduction

New representations in lower-dim space

Key questions for a DR algorithm:

1. Is it linear or non-linear?
2. What is the optimization objective of this algorithm?
3. What is the application of the outputs?
4. What are the hyperparameters? How to tune them?

Dimensionality Reduction

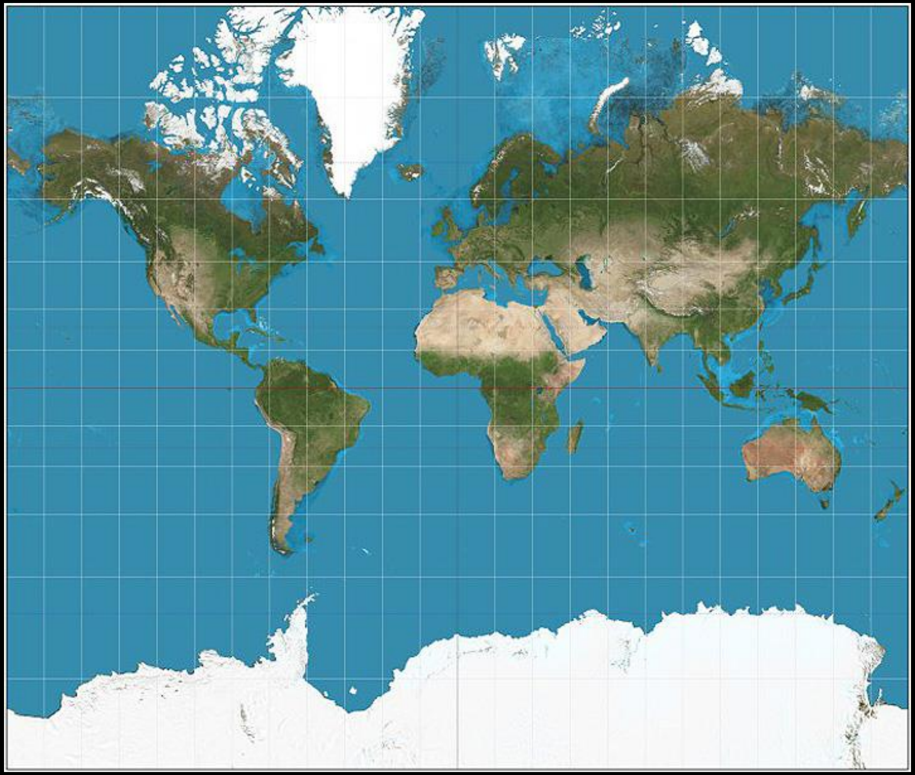


Image Credit

DR is similar with Map projection (e.g. Mercator)

1. Both are reduction of dimensions
2. There are various methods, depending on applications
3. The DR would lead to some Information loss

Dimensionality Reduction

Motivations

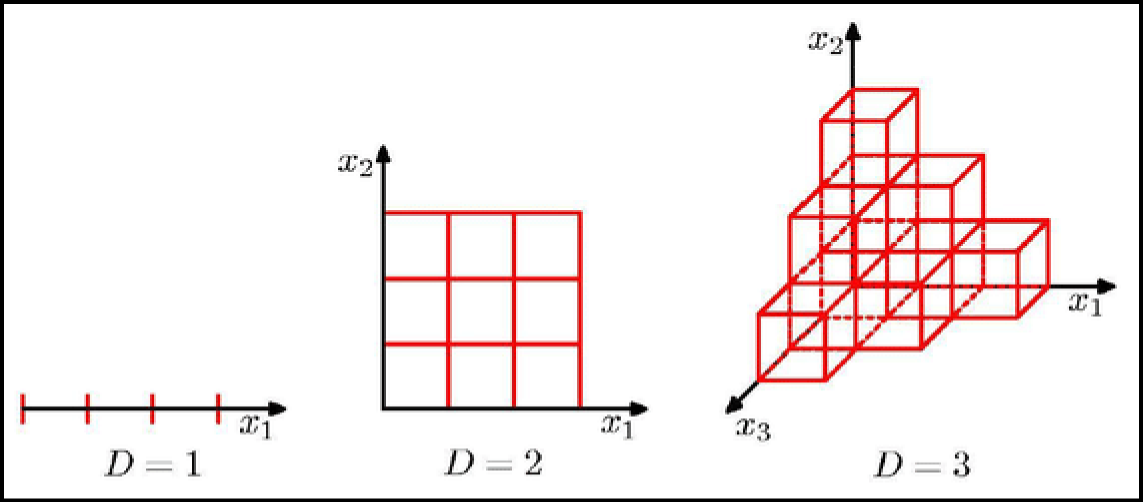
Perspective	Details
Visualisation	To visualise the data when reduced to low dimensions such as 2D or 3D
Computation	To reduce the time and storage space
Modelling	To reduce number of features and avoid overfitting
Others	To avoid the curse of dimensionality

Curse of dimensionality

Curse of Dimensionality

Difficulty of high dimensions

1. Possibilities are exponential in the dimensions



Possible values 3^1 3^2 3^3 3^k

Each additional dimension triples the effort to grid search all combinations.

https://www.researchgate.net/publication/278786166_Some_Contributions_to_Supervised_Classification_of_Hyperspectral_Data

Curse of Dimensionality
Difficulty of high dimensions

2. High dimensions cause overfitting in machine learning

- An enormous amount of training data is required to ensure that there are several samples with each combination of values.

- When we have more features than training records, we run the risk of massively overfitting our model

Curse of Dimensionality

Distance functions

3. Distance functions become meaningless in high dimensions

- Every observation in the dataset appears ‘equidistant’ from all the others

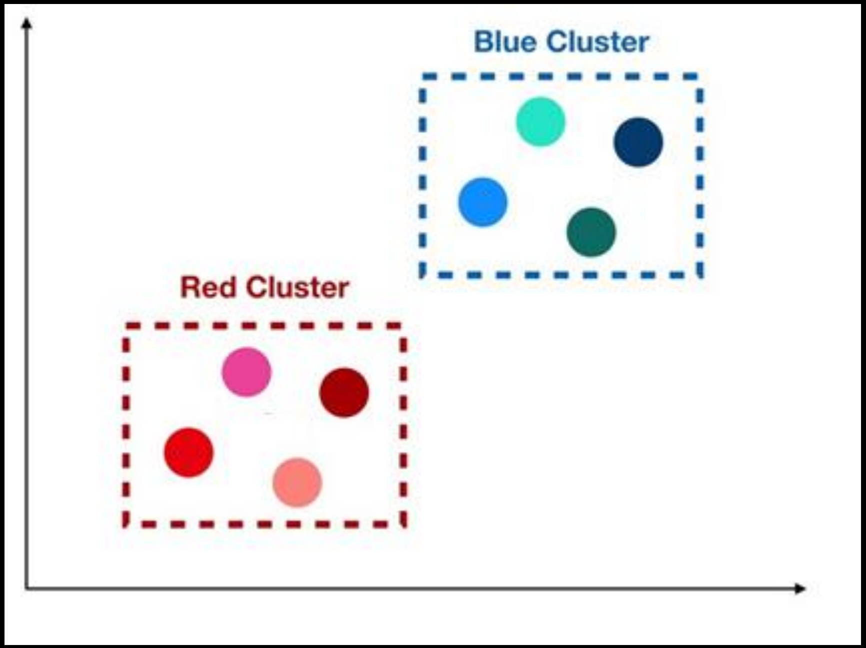
- No meaningful clusters can be found.

https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e

Curse of Dimensionality

Distance functions

Example: Clustering of candies from colours



Visual observations:
there are two clusters of candies

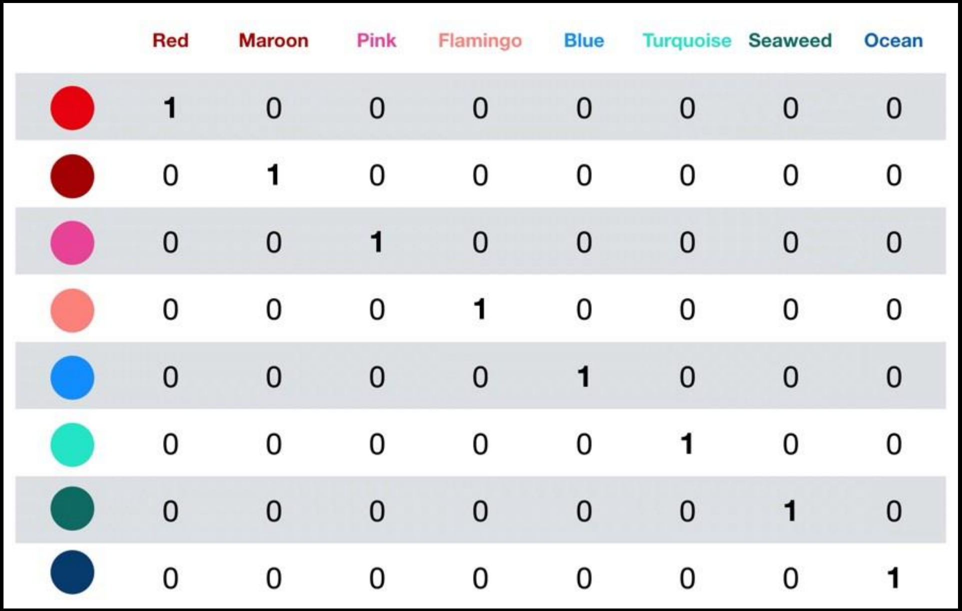
Example credit (Adjusted)

Curse of Dimensionality

Distance functions

Colour definition using 8 colours

- What is the Euclidean distance between each pair?
- How many clusters?

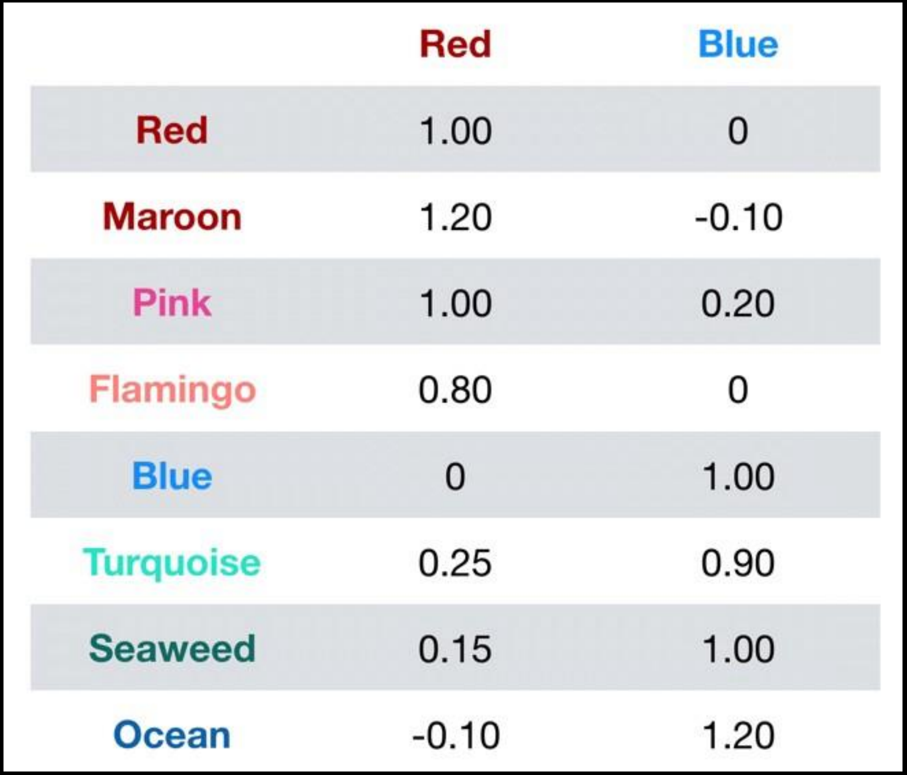


	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
●	1	0	0	0	0	0	0	0
●	0	1	0	0	0	0	0	0
●	0	0	1	0	0	0	0	0
●	0	0	0	1	0	0	0	0
●	0	0	0	0	1	0	0	0
●	0	0	0	0	0	1	0	0
●	0	0	0	0	0	0	1	0
●	0	0	0	0	0	0	0	1

Example credit (Adjusted)

Curse of Dimensionality

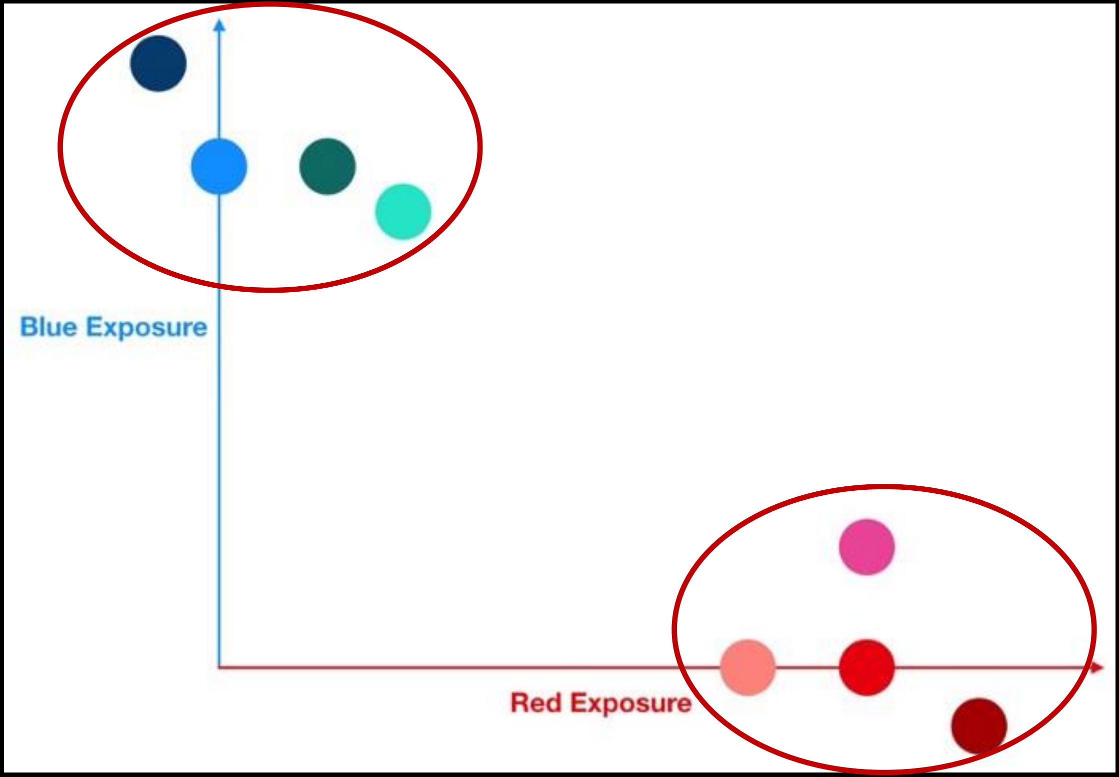
Using DR to learn the new dimensions



	Red	Blue
Red	1.00	0
Maroon	1.20	-0.10
Pink	1.00	0.20
Flamingo	0.80	0
Blue	0	1.00
Turquoise	0.25	0.90
Seaweed	0.15	1.00
Ocean	-0.10	1.20

Curse of Dimensionality

Transform the candies using the new dimensions



Two clusters are identified, which is consistent with the visual judgement

Curse of Dimensionality

Implications

In a high dimension space, it is (very) likely

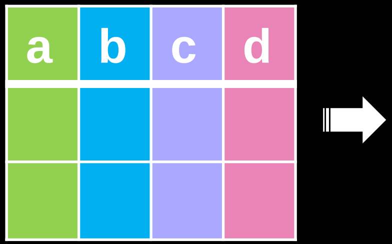
1. Many features are almost constant and some are highly correlated.
2. Most observations actually lie within (or close to) a much lower-dimensional subspace
3. DR algorithms aim to learn the low-dimensional subspace

Principal Component Analysis

Principal Component Analysis

Linear combination of features

- Steps (“*Keep the largest variance*”)
  1. Find a new set of dimensions (called principal components, or PC). Each PC is a linear combination of the original dims
  2. Rank all PC according to the variance of data. The larger variance, the higher importance.
  3. Keep the first *k* PC. (using rules to select)
  4. Project the data into the new space.

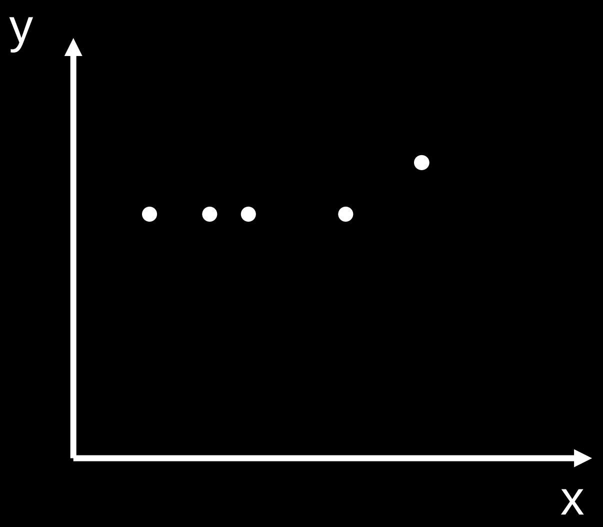


Name	New dim	combination	Variance
1st component	e	0.5a + 0.6b + 0.1c + 0.3d	0.8
2nd component	f	0.6a + 0.1b + 0.2c + 0.5d	0.1
	g		0.05
	h		0.05

Principal Component Analysis
Linear combination of features

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

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



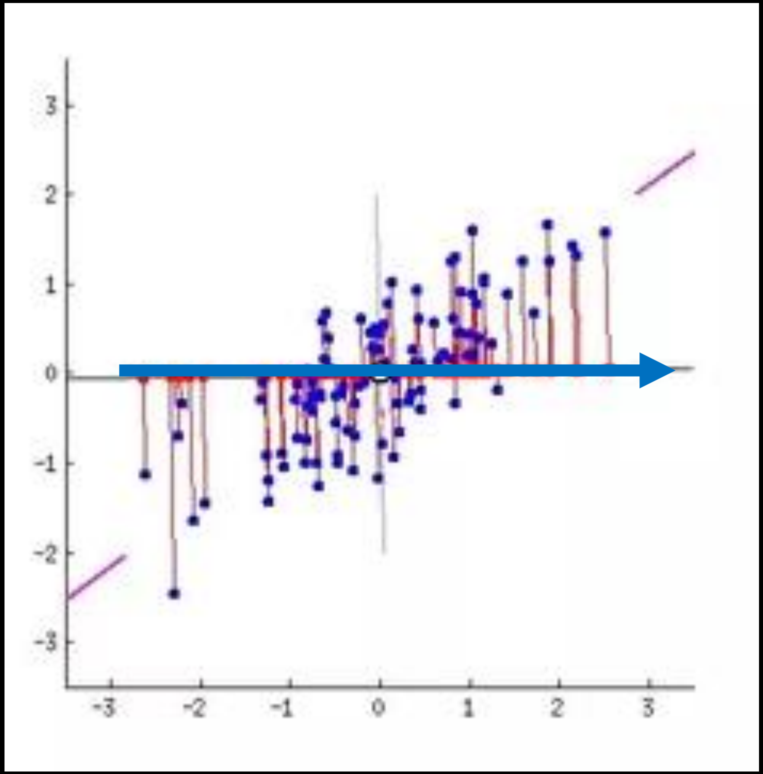
Which dimension has a larger variance? x or y?

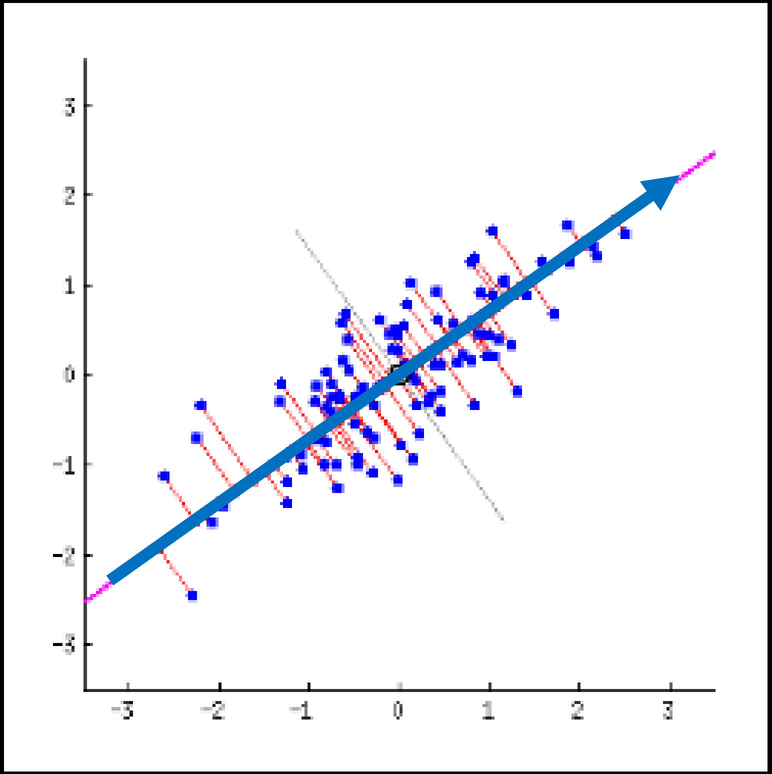
Principal Component Analysis

Linear combination of features

- Example: 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

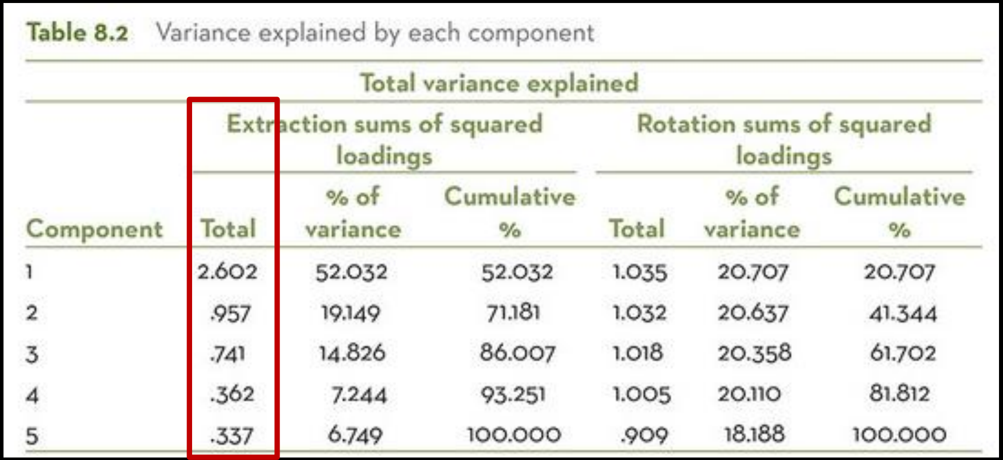
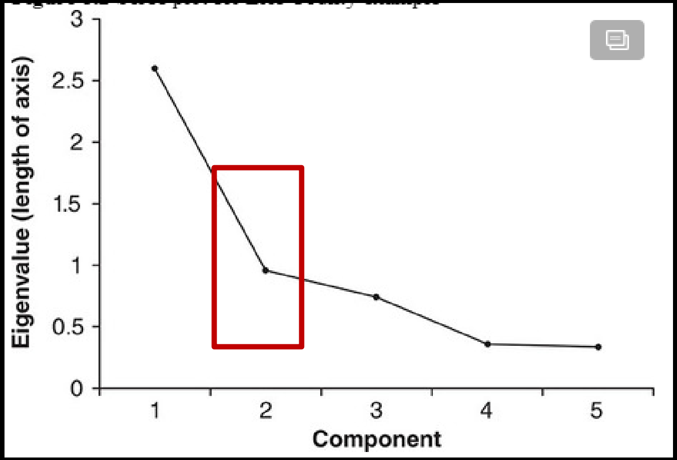


Table 8.2 Variance explained by each component

	Total variance explained					
	Extraction sums of squared loadings			Rotation sums of squared loadings		
Component	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

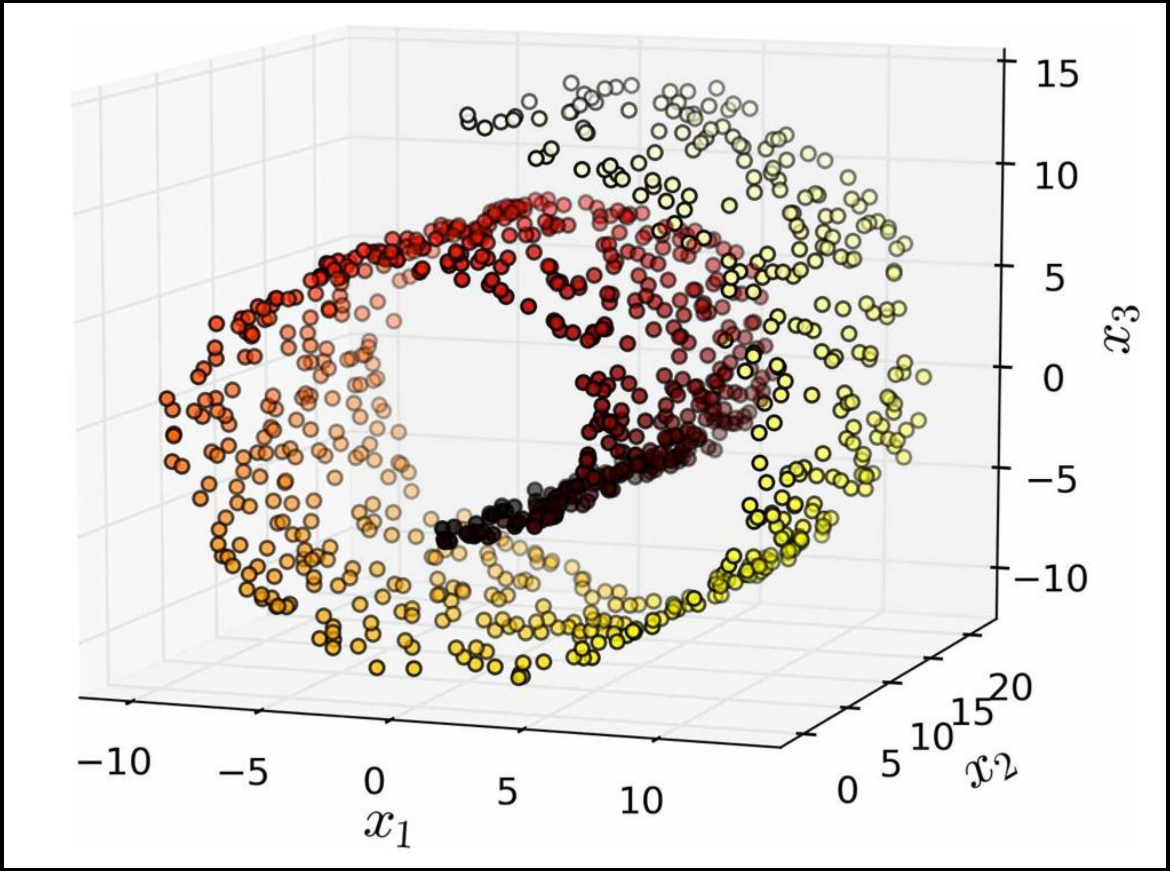
Some notes

- PCA requires data standardisation, as it is sensitive to the relative scales of the original variables.
- Good interpretation: each component is a linear combination of features
- It is guaranteed that the new features are uncorrelated – no more multicollinearity concerns.
- Common use: visualising the data; checking the clustering results
- The PCA outputs can be used as an input to clustering/classification/regression.

Principal Component Analysis

Problems

- PCA does not work well for ‘twisted’ dataset. In this case, non-linear DR or manifold learning is useful.



Example: a 3-D twisted data (a swiss roll)

This data is systhesised to illustrate the DR methods.

Colour is used to represent clusters.

Image Credit

Manifold learning

- It relies on the manifold assumption: most real-world high-dimensional datasets lie close to a much lower-dimensional manifold.
- This assumption is very often empirically observed.

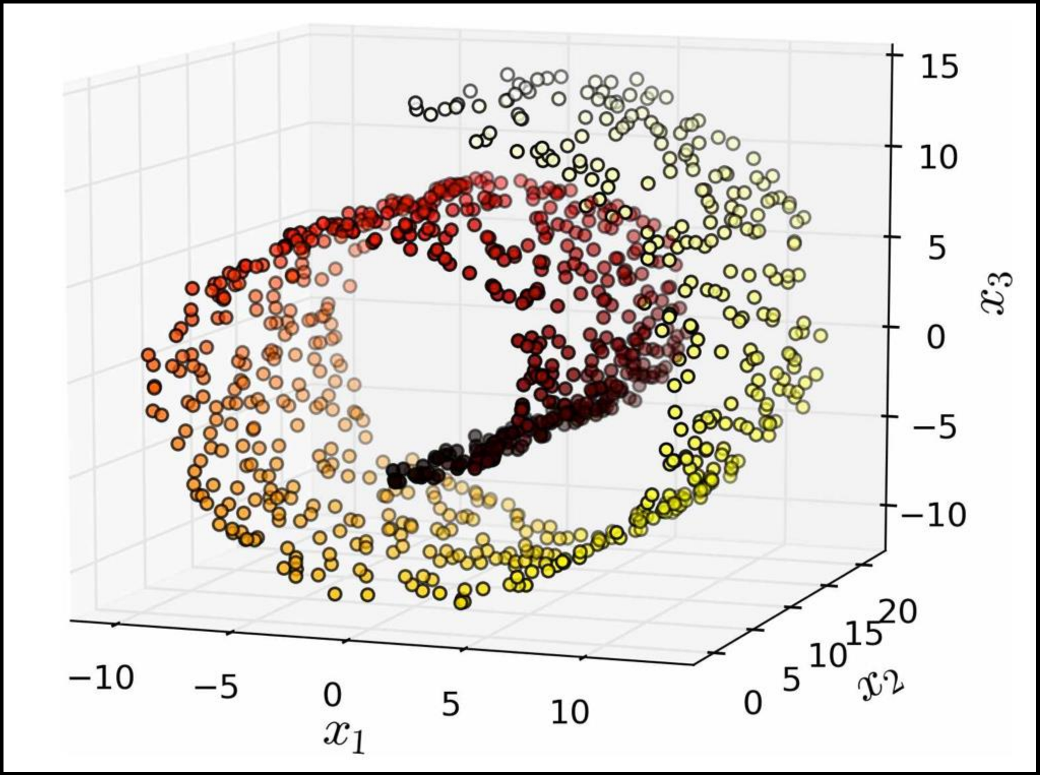


Image Credit

PCA vs. Manifold learning

- PCA (left): fail to identify point clusters in the original space
- Manifold learning (right)

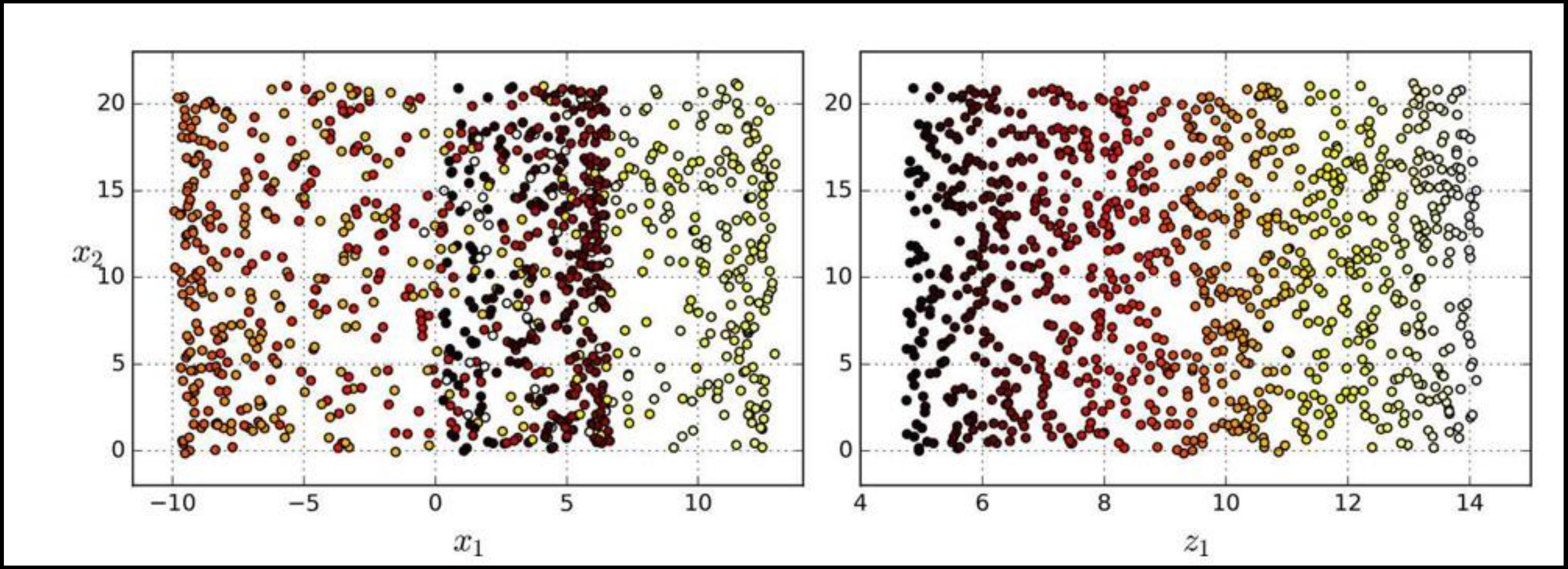


Image Credit

Kernel PCA

kPCA

- One type of manifold learning or non-linear DR
- Kernel trick: a mathematical technique that implicitly maps instances into a very high-dimensional space, enabling non-linear classification/regression
- kPCA uses this idea: first uses a kernel function to map observations into a higher dimension, then do DR in the higher dimension space.
- The benefits are that it would find patterns that are not identified using linear DR.

kPCA

Common kernels

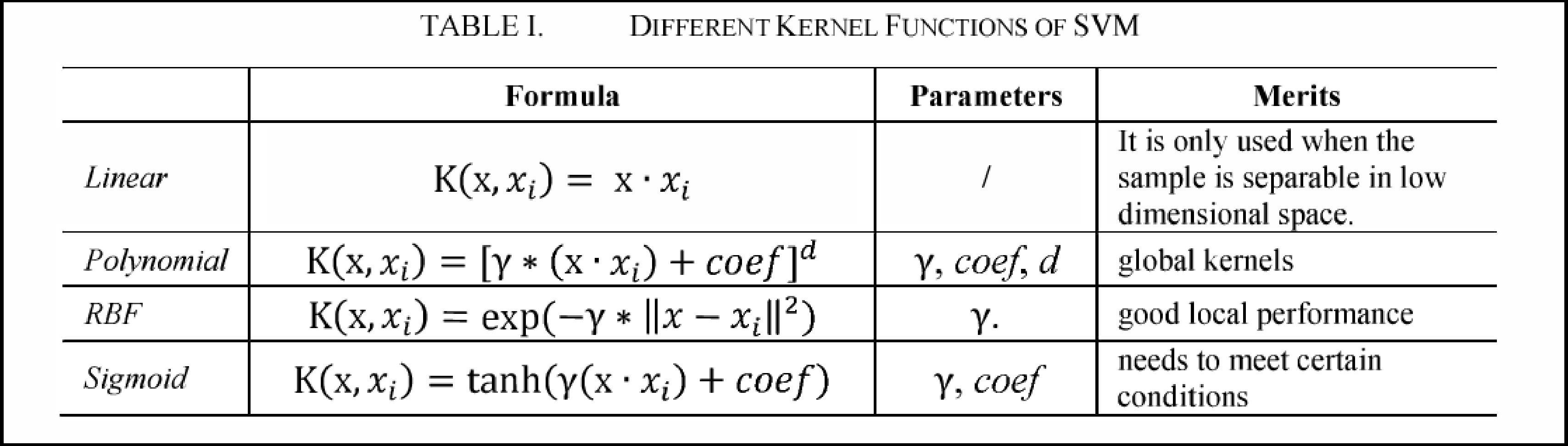


TABLE I. DIFFERENT KERNEL FUNCTIONS OF SVM

	Formula	Parameters	Merits
Linear	$K(x, x_i) = x \cdot x_i$	/	It is only used when the sample is separable in low dimensional space.
Polynomial	$K(x, x_i) = [\gamma * (x \cdot x_i) + coef]^d$	γ, *coef*, *d*	global kernels
RBF	$K(x, x_i) = \exp(-\gamma * \|x - x_i\|^2)$	γ.	good local performance
Sigmoid	$K(x, x_i) = \tanh(\gamma(x \cdot x_i) + coef)$	γ, *coef*	needs to meet certain conditions

Image Credit

kPCA

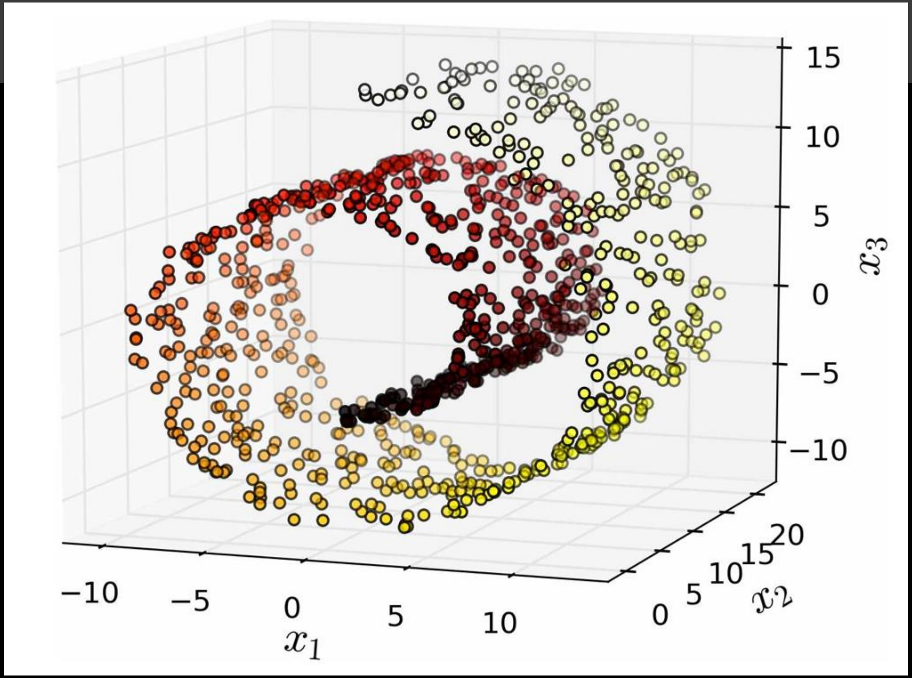
- Hyperparameters of kPCA
  - Which kernel to use: linear kernel, rbf, sigmoid kernel, etc
  - Hyperparameters of kernels: gamma of rbf

```
k_pca = KernelPCA(n_components = 2, kernel="rbf", gamma=0.0433, fit_inverse_transform=True)
```

- Similar problems with Kmeans or random forest
- You can use a performance measure and some ground-truth data (e.g. classification, or MSE of fit) to tune the hyperparameters.

kPCA

- Example: using kPCA to reduce the twisted data from 3D to 2D.
- The clusters are roughly kept in the kPCA results.



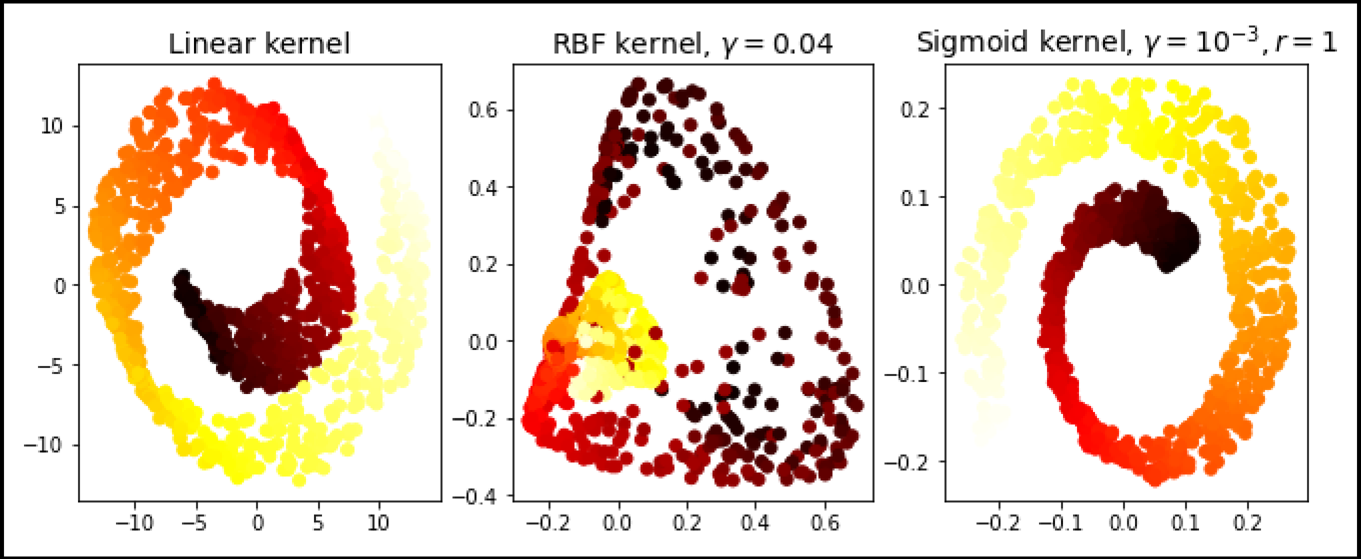


Image Credit

Locally Linear Embedding (LLE)

LLE

- One type of manifold learning or non-linear DR
- The principle is to preserve local relations (contrast to preserving global variance in PCA)

- Two steps (hyperparameter: k or n_neighbors)
  1. Measures how each instance relates to closest neighbors (weighted sum)
  2. Looks for low-dimension representation where local relations are best preserved.

LLE

- Step 1
  - For each training instance x_i, LLE identifies its k closest neighbors
  - Then, LLE tries to reconstruct x_i as a linear function of these neighbors $x_{NN_{ij}}$

$$x_i = \sum_{j=1}^{k} w_{ij} x_{NN_{ij}}$$

where NN_{ij} is the index of the j-th neighbour of x_i

  - Weights w_{ij} are optimized such that
  - The squared distance between x_i and $\sum_{j=1}^{k} w_{ij} NN_{ij}$ is minimised.
  - AND weights are normalised $\sum_{j=1}^{k} w_{ij} = 1$ for any x_i

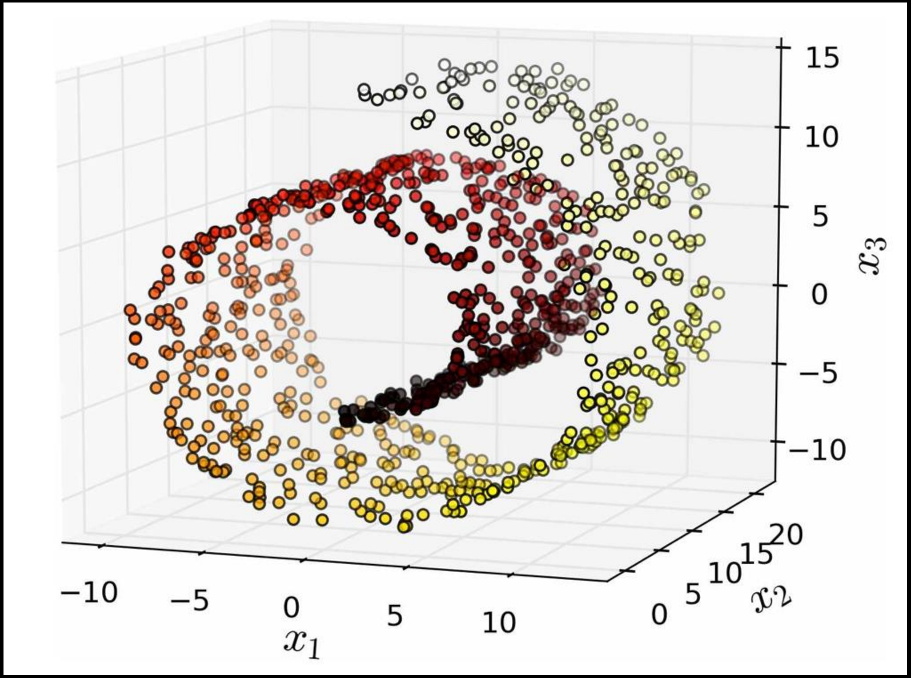
  - The output of Step 1 is the weights w_{ij}

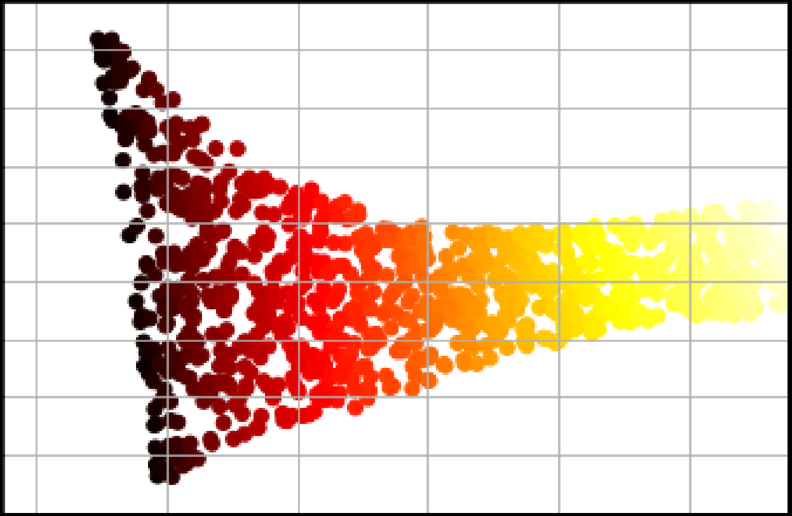
LLE

- Step 2
  - Map the observations into a lower d-dimensional space such that these local relationships are preserved as much as possible.
  - If z_i is the d-space equivalent of x_i, then we want $z_i - \sum_{j=1}^{k} w_{ij} z_{NN_{ij}}$ to be minimized, given the w_{ij} from Step 1
  - The output of Step 2 is the position of observations in the new d-dim space

LLE

- Example (using LLE to unroll the Swiss roll)
  - The neighbour relation (with similar colours) is kept in the new space





Comparison

Comparing PCA, kPCA, LLE

	PCA	kPCA	LLE
Principle	Preserving global variance	Preserving global variance	Preserving local relation (nearest neighbours)
Meaning of new feature	Linear combination of original dims	Unclear	Unclear
Flexibility	Results are deterministic and can’t be adjusted	Can be adjusted using the hyperparameters	Can be adjusted using the hyperparameters. n_neighbors reflects a trade-off between local or global patterns.
As input to other analysis?	Yes	Yes	Yes
Computation cost	Normally low, but high for really high dims	Normally low	Normally low

Suggestions of using DR

- First choice: PCA is the baseline DR algorithm and the first choice.
- If PCA is really slow or does not yield good results (visually or low variation explained by first few PCs), then try kPCA or LLE.

Other DR methods

- t-Distributed Stochastic Neighbour Embedding (tSNE): preserves similar instances that are close and pushes dis-similar instances apartExample
- Multi-D Scaling (MDS): tries to reduce D while keep instance distance the same
- ISOMAP: connects each instance to its neighbours then preserves the geodesic distance between instances

Summary

- Dimensionality Reduction
- Curse of dimensionality
- PCA is a linear DR algorithm. Used for visualisation. The output of PCA can be used as input to other analysis.
- kPCA and t-SNE are non-linear and powerful DR. The new features are not interpretable.

Thank You

Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk

Workshop
Dimension reduction

- **Download this week’s Python Notebook from Moodle, open it in Anaconda and work through.**

- Again, you’re not required to understand all of the maths and computation. The application and interpretation is the key points.