Advanced Data Science

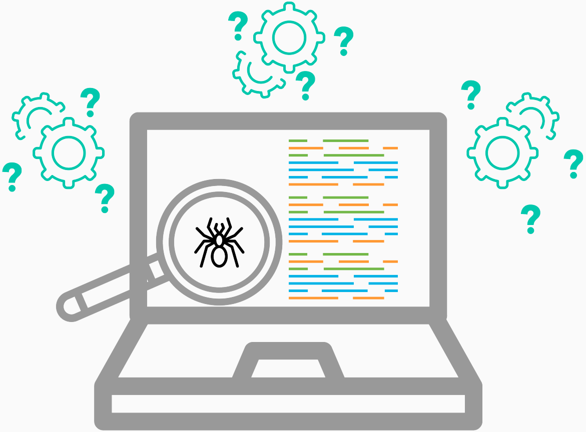
Lecture 8 : Visualisation II

Carl Henrik Ek - che29@cam.ac.uk

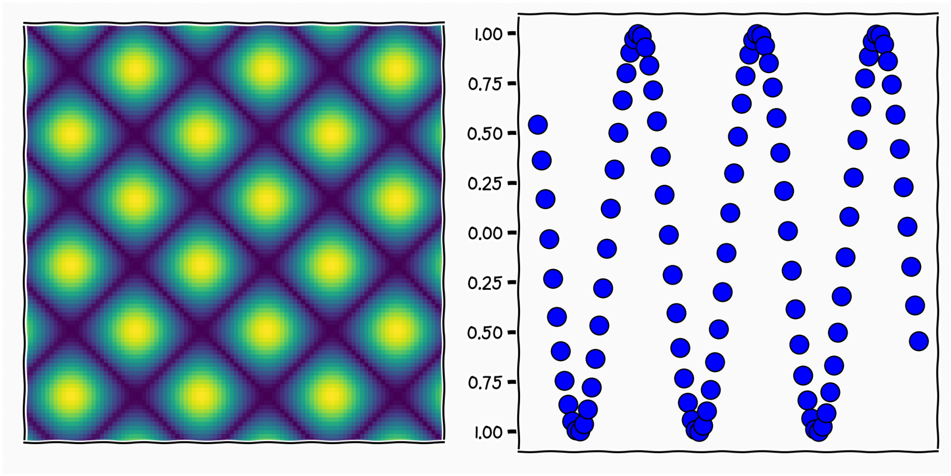
14th of November, 2022

http://carlhenrik.com

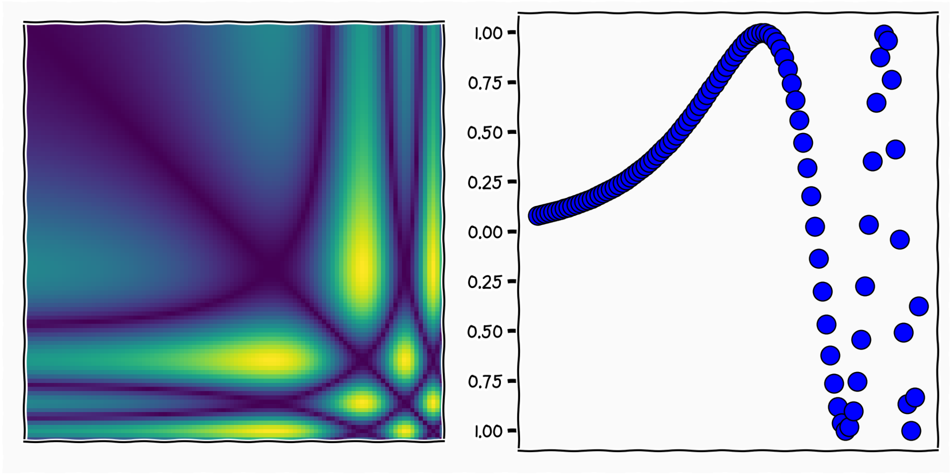
Data Science is Debugging



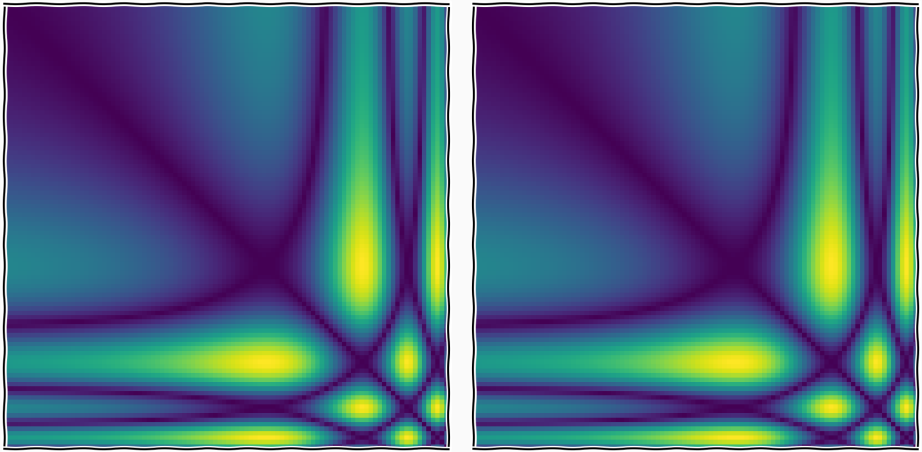
Distance Matrix



Distance Matrix



Distance Matrix



Code

```
scipy.spatial.distance.cdist(XA, XB, metric='euclidean',
                             *, out=None, **kwargs)[source])
```

Compute distance between each pair of the two collections of inputs.

metricstr or callable, optional The distance metric to use. If a string, the distance function can be ‘braycurtis’, ‘canberra’, ‘chebyshev’, ‘cityblock’, ‘correlation’, ‘cosine’, ‘dice’, ‘euclidean’, ‘hamming’, ‘jaccard’, ‘jensenshannon’, ‘kulczynski1’, ‘mahalanobis’, ‘matching’, ‘minkowski’, ‘rogerstanimoto’, ‘russellrao’, ‘seuclidean’, ‘sokalmichener’, ‘sokalsneath’, ‘sqeuclidean’, ‘yule’.

Dimensionality Reduction

High Dimensional

0.98177005	-0.99053874	-0.01683981	-0.3994665	0.12133672
1.16342824	-0.99520027	0.90381171	0.27386304	-1.06091985
-1.90577283	0.91220641	1.74809035	1.66393916	-0.54346161
-0.56907458	0.89406555	-0.17182898	1.81980444	1.8713991
1.53380634	1.20296216	-0.26604579	0.48691598	-1.3871063
-0.95765954	-0.61907303	-1.33657998	0.71134795	1.01014797
1.32466764	0.53453037	-1.55772646	1.55236474	0.84368406
-0.6207868	0.25005863	-0.90101442	0.07198261	0.92843713
0.89584615	0.20860728	0.56883429	0.2793335	0.32354156
0.10053249	-1.01930463	0.71546593	-1.87660674	-1.03507809
-0.54741634	1.42964806	-1.84004808	-0.94952952	-0.31223371

Eigen-decomposition

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$$

$$\boldsymbol{\Lambda} = \begin{cases} 0 & i \neq j \\ \lambda_i & i = j \end{cases}$$

$$\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{I} \quad \Rightarrow \quad \mathbf{V}^{-1} = \mathbf{V}^{\mathrm{T}}.$$

Eigen-decomposition

$$\mathbf{M} = \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^{\mathrm{T}}.$$

- the eigen decomposition means we can write a matrix as a sum of rank one matrices
- all symmetric real matrices have a diagonal matrix that they are similar to

Rank-Nullity Theorem

$$\mathrm{Rank}(T) + \mathsf{Nullity}(T) = \mathsf{dim}(A)$$

- $T : A \to B$ is a map between two vector spaces
- $\mathrm{Rank}(T)$ is the dimensionality of the *image* of T
- $\mathrm{Nullity}(T)$ is the dimensionality of the *kernel* of T

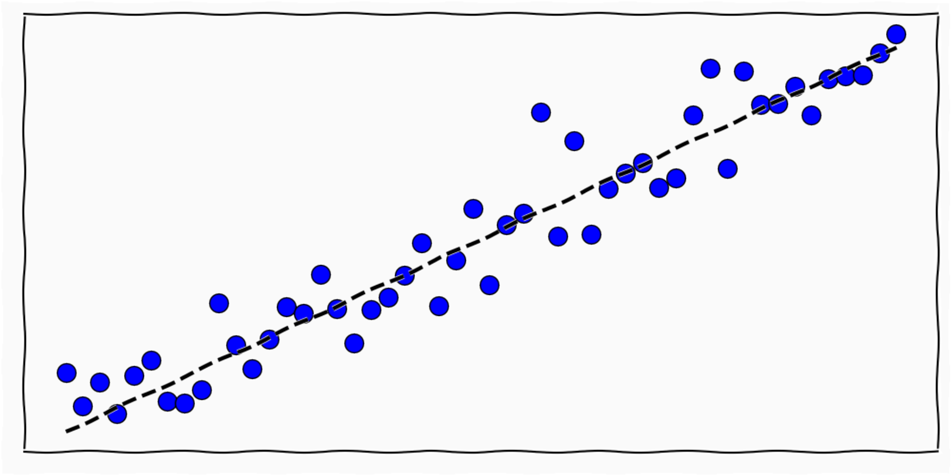
Rank-Nullity Theorem

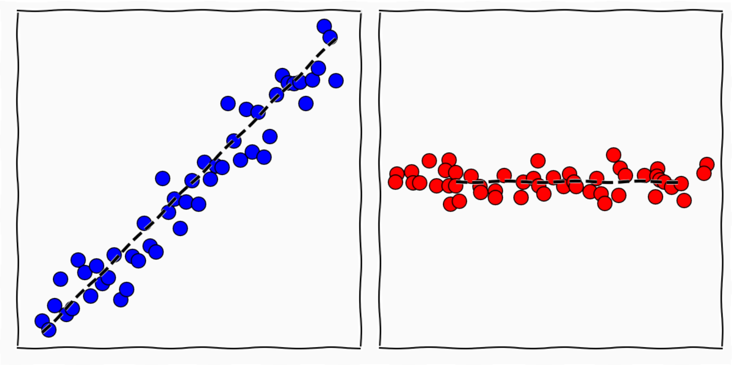
$$\mathrm{Rank}(T) + \mathsf{Nullity}(T) = \mathsf{dim}(A)$$

Task Can we find a map T such that kernel of the map is the subspace where the data have no variations?

Task Can we find a map T such that the dimensions are ordered in decreasing order of how much variations the data has?

Rank-Nullity





$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

- Compute Empirical Covariance Matrix of the data

$$\mathbf{C} = \mathbf{Y}^T\mathbf{Y}$$

- Compute Empirical Covariance Matrix of the data

$$\mathbf{C} = \mathbf{Y}^T\mathbf{Y}$$

- Diagonalise C

$$\mathbf{C} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

Principal Component Analysis

- Compute Empirical Covariance Matrix of the data

$$\mathbf{C} = \mathbf{Y}^T \mathbf{Y}$$

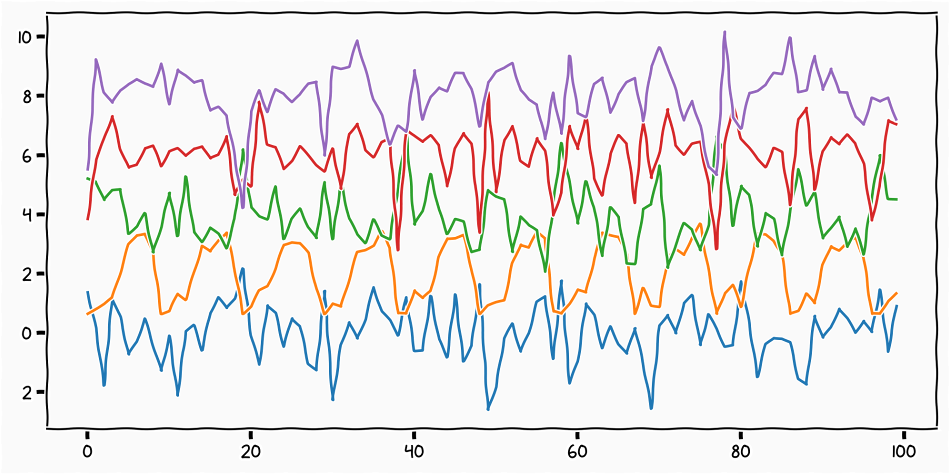
- Diagonalise C

$$\mathbf{C} = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^{\mathrm{T}}$$

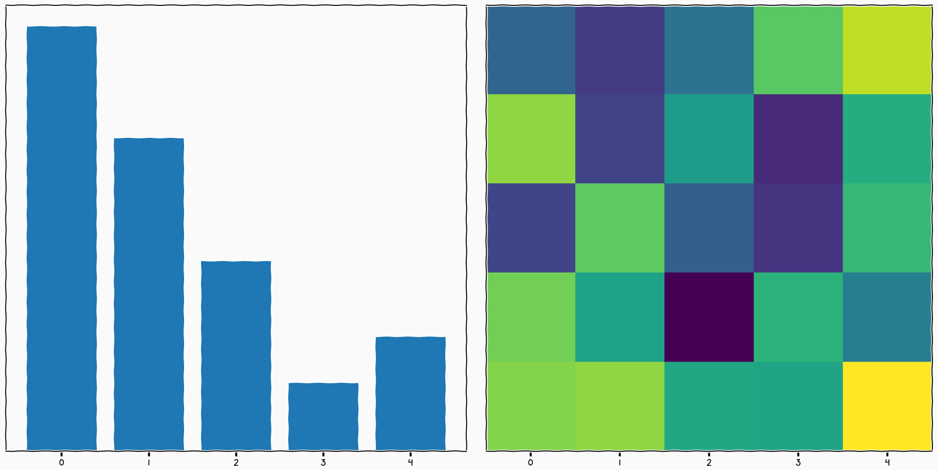
- Project Data onto eigenvectors that corresponds to highest variance

$$\mathbf{X} = \mathbf{Y} \mathbf{V}^{\mathrm{T}}$$

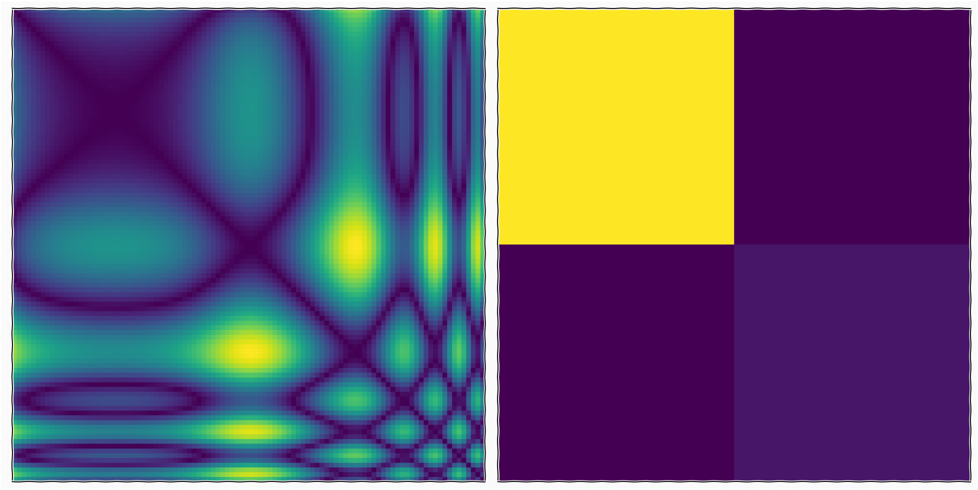
MoCap



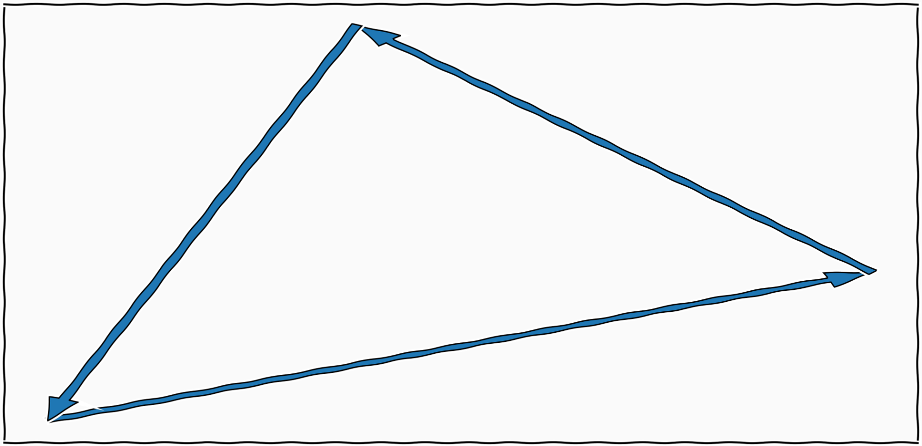
Eigenvectors and Eigenvalues



Matrices



Distances and Inner Products



Distances and Inner Products

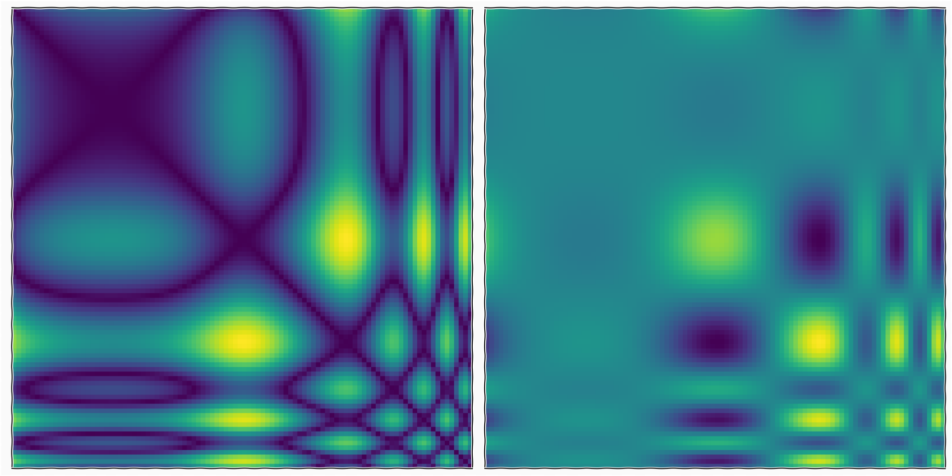
$$\mathbf{D}_{ij}^2 = d_{ij}^2 = \sum_{k=1}^{d} (y_{ki} - y_{kj})^2 = \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i + \mathbf{y}_j^{\mathrm{T}}\mathbf{y}_j - 2\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j$$

$$\mathbf{G}_{ij} = g_{ij} = \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j$$

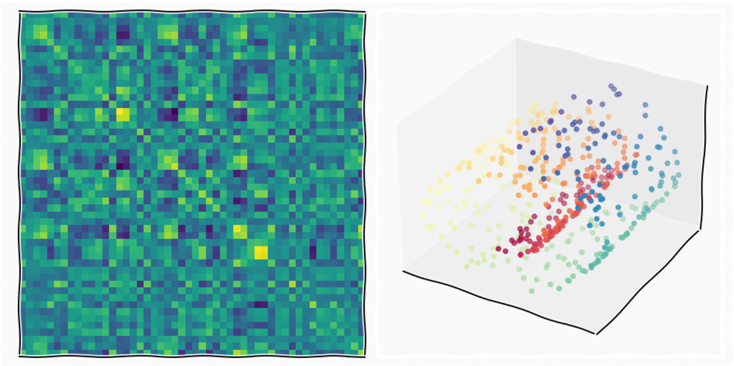
$$d_{ij}^2 = g_{ii} + g_{jj} - 2g_{ij}$$

- if we assume that the data is centred we can write the Gram matrix as a function of the distance matrix

Distances and Inner Products



Multi Dimensional Scaling [Cox et al., 2008]



- Given a similarity matrix $\mathbf{\Delta}$ can we find a vectorial representation such that,

$$\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j = \mathbf{\Delta}_{ij}$$

Multi Dimensional Scaling

$$\mathbf{\Delta} = \begin{bmatrix} \delta_{00} & \delta_{01} & \cdots & \delta_{0N} \\ \delta_{10} & \delta_{11} & \cdots & \delta_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{N0} & \delta_{N1} & \cdots & \delta_{NN} \end{bmatrix}$$

Multi Dimensional Scaling

- MDS Objective,

$$\hat{\mathbf{Y}} = \text{argmin}_{\mathbf{Y}} \|\mathbf{D} - \boldsymbol{\Delta}\|_F.$$

Multi Dimensional Scaling

- MDS Objective,

$$\hat{\mathbf{Y}} = \text{argmin}_{\mathbf{Y}} \|\mathbf{D} - \mathbf{\Delta}\|_F.$$

- Element-Wise Matrix norm,

$$\|\mathbf{M}\|_{p,q} = \left(\sum_{j=1}^{n} \left(\sum_{i=1}^{m} |m_{ij}|^p \right)^{\frac{p}{q}} \right)^{\frac{1}{q}}$$

Multi Dimensional Scaling

$$\text{argmin}_{\mathbf{D}} \|\mathbf{D} - \mathbf{\Delta}\|_F^2 = \text{argmin}_{\mathbf{D}} \text{trace} \left(\mathbf{D} - \mathbf{\Delta}\right)^2$$

Multi Dimensional Scaling

$$\begin{aligned}\mathrm{argmin}_{\mathbf{D}}\|\mathbf{D}-\boldsymbol{\Delta}\|_F^2 &= \mathrm{argmin}_{\mathbf{D}}\mathrm{trace}\left(\mathbf{D}-\boldsymbol{\Delta}\right)^2\\ &= \mathrm{argmin}_{\mathbf{Q},\hat{\boldsymbol{\Lambda}}}\mathrm{trace}\left(\mathbf{Q}\hat{\boldsymbol{\Lambda}}\mathbf{Q}^{\mathrm{T}}-\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}\right)^2\end{aligned}$$

Multi Dimensional Scaling

$$\begin{aligned}
\operatorname{argmin}_{\mathbf{D}} \|\mathbf{D} - \boldsymbol{\Delta}\|_F^2 &= \operatorname{argmin}_{\mathbf{D}} \operatorname{trace} (\mathbf{D} - \boldsymbol{\Delta})^2 \\
&= \operatorname{argmin}_{\mathbf{Q}, \hat{\boldsymbol{\Lambda}}} \operatorname{trace} \left(\mathbf{Q} \hat{\boldsymbol{\Lambda}} \mathbf{Q}^{\mathrm{T}} - \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^{\mathrm{T}} \right)^2 \\
&= \operatorname{argmin}_{\mathbf{Q}, \hat{\boldsymbol{\Lambda}}} \operatorname{trace} \left(\mathbf{V}^{\mathrm{T}} \left(\mathbf{Q} \hat{\boldsymbol{\Lambda}} \mathbf{Q}^{\mathrm{T}} - \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^{\mathrm{T}} \right) \mathbf{V} \right)^2
\end{aligned}$$

Multi Dimensional Scaling

$$
\begin{aligned}
\mathrm{argmin}_{\mathbf{D}} \|\mathbf{D} - \boldsymbol{\Delta}\|_F^2 &= \mathrm{argmin}_{\mathbf{D}} \mathrm{trace} \left(\mathbf{D} - \boldsymbol{\Delta}\right)^2 \\
&= \mathrm{argmin}_{\mathbf{Q}, \hat{\boldsymbol{\Lambda}}} \mathrm{trace} \left(\mathbf{Q}\hat{\boldsymbol{\Lambda}}\mathbf{Q}^{\mathrm{T}} - \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}\right)^2 \\
&= \mathrm{argmin}_{\mathbf{Q}, \hat{\boldsymbol{\Lambda}}} \mathrm{trace} \left(\mathbf{V}^{\mathrm{T}} \left(\mathbf{Q}\hat{\boldsymbol{\Lambda}}\mathbf{Q}^{\mathrm{T}} - \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}\right) \mathbf{V}\right)^2 \\
&= \mathrm{argmin}_{\mathbf{Q}, \hat{\boldsymbol{\Lambda}}} \mathrm{trace} \left(\mathbf{V}^{\mathrm{T}}\mathbf{Q}\hat{\boldsymbol{\Lambda}}\mathbf{Q}^{\mathrm{T}}\mathbf{V} - \mathbf{V}^{\mathrm{T}}\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^2
\end{aligned}
$$

Multi Dimensional Scaling

$$
\begin{aligned}
\operatorname{argmin}_{\mathbf{D}} \|\mathbf{D} - \mathbf{\Delta}\|_F^2 &= \operatorname{argmin}_{\mathbf{D}} \operatorname{trace} \left(\mathbf{D} - \mathbf{\Delta}\right)^2 \\
&= \operatorname{argmin}_{\mathbf{Q},\hat{\mathbf{\Lambda}}} \operatorname{trace} \left(\mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^{\mathrm{T}} - \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\right)^2 \\
&= \operatorname{argmin}_{\mathbf{Q},\hat{\mathbf{\Lambda}}} \operatorname{trace} \left(\mathbf{V}^{\mathrm{T}} \left(\mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^{\mathrm{T}} - \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\right) \mathbf{V}\right)^2 \\
&= \operatorname{argmin}_{\mathbf{Q},\hat{\mathbf{\Lambda}}} \operatorname{trace} \left(\mathbf{V}^{\mathrm{T}}\mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^{\mathrm{T}}\mathbf{V} - \mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^2 \\
&= \operatorname{argmin}_{\mathbf{Q},\hat{\mathbf{\Lambda}}} \operatorname{trace} \left(\mathbf{V}^{\mathrm{T}}\mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^{\mathrm{T}}\mathbf{V} - \mathbf{\Lambda}\right)^2 .
\end{aligned}
$$

Multi Dimensional Scaling

$$\mathbf{D} = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^{\mathrm{T}},$$

$$\|\mathbf{D} - \boldsymbol{\Delta}\|_F = \sqrt{\sum_{i=d+1}^{N} \lambda_i^2}$$

- To get the best d dimensional solution we pick the top d eigenvalues

Multi Dimensional Scaling

$$\mathbf{D} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

Multi Dimensional Scaling

$$
\begin{aligned}
\mathbf{D} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \\
&= \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\right)
\end{aligned}
$$

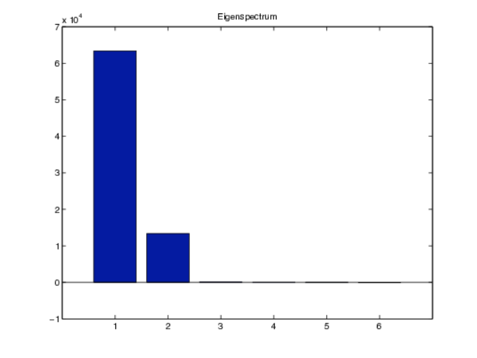
$$
\begin{aligned}
\mathbf{D} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \\
&= \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\right) \\
&= \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)^{\mathrm{T}}
\end{aligned}
$$

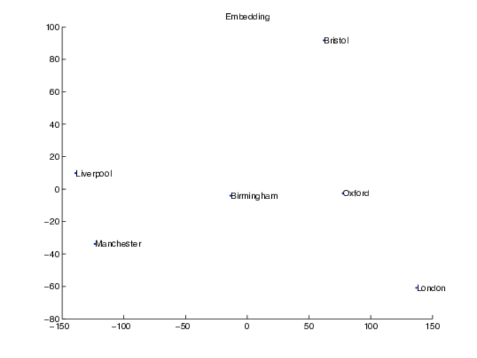
$$
\begin{aligned}
\mathbf{D} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \\
&= \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\right) \\
&= \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)^{\mathrm{T}} \\
&\Rightarrow \mathbf{Y} = \mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}
\end{aligned}
$$

Example

	Man	Ox	Lon	Bri	Liv	Birm
Man	0	203	262	224	46	114
Ox	203	0	83	95	217	91
Lon	262	83	0	170	285	161
Bri	224	95	170	0	217	122
Liv	46	217	285	217	0	126
Birm	114	91	161	122	126	0

Example


Eigenspectrum
x 10^4
7
6
5
4
3
2
1
0
−1
1
2
3
4
5
6


Embedding
100
80
60
40
20
0
−20
−40
−60
−80
−150
−100
−50
0
50
100
150
Bristol
Liverpool
Birmingham
Oxford
Manchester
London

PCA Equvivalence [1]

- In MDS we diagonalise a $N \times N$ matrix

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$$

[1] see attached notes

PCA Equvivalence [1]

- In MDS we diagonalise a $N \times N$ matrix

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$$

- In PCA we diagonalise a $D \times D$ matrix

$$\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$$

[1] see attached notes

PCA Equvivalence [1]

- In MDS we diagonalise a $N \times N$ matrix

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$$

- In PCA we diagonalise a $D \times D$ matrix

$$\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$$

- Rank

$$\mathrm{Rank}\left(\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\right) = \mathrm{Rank}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right).$$

[1] see attached notes

Proximity Graph

- We have a method to find a geometrical embedding from a similarity relationship

Proximity Graph

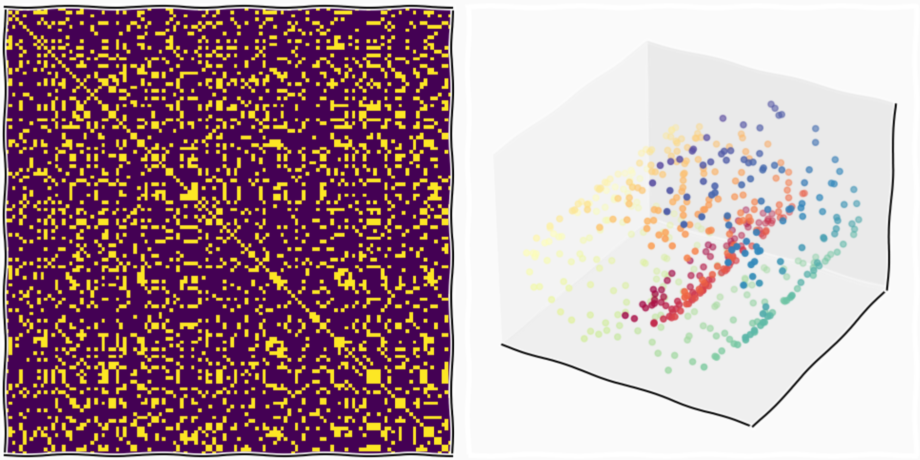
- We have a method to find a geometrical embedding from a similarity relationship
- *a manifold is a topological space that near each point resembles Euclidean space*

Proximity Graph

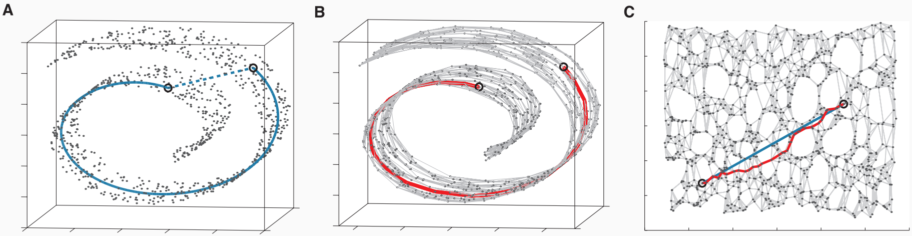
- We have a method to find a geometrical embedding from a similarity relationship
- *a manifold is a topological space that near each point resembles Euclidean space*
- $\Rightarrow$ we can *measure* local distances faithfully

Proximity Graph

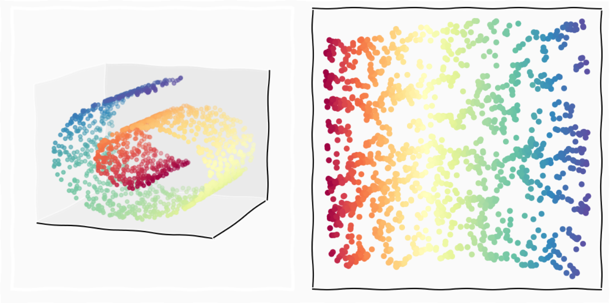
- We have a method to find a geometrical embedding from a similarity relationship
- *a manifold is a topological space that near each point resembles Euclidean space*
- $\Rightarrow$ we can *measure* local distances faithfully
- Learning manifold implies completing similarity relationship



Isomap [Tenenbaum et al., 2000]



1. Compute local similarity
2. Compute shortest path in graph
3. Apply MDS



Multi-Dimensional Scaling

- Compute a distance matrix D

Multi-Dimensional Scaling

- Compute a distance matrix D
- Convert distance matrix to inner-product (Gram matrix)

Multi-Dimensional Scaling

- Compute a distance matrix D
- Convert distance matrix to inner-product (Gram matrix)
- Diagonalise inner-produce matrix

Multi-Dimensional Scaling

- Compute a distance matrix D
- Convert distance matrix to inner-product (Gram matrix)
- Diagonalise inner-produce matrix
- Recover *relative* spatial structure that reflect distance

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}}$$

Summary

- Learn how to read distance matrices

Summary

- Learn how to read distance matrices
- PCA is your first `fprintf(stderr, ... )`

Summary

- Learn how to read distance matrices
- PCA is your first `fprintf(stderr, ... )`
- PCA diagonalises the covariance matrix $D \times D$

Summary

- Learn how to read distance matrices
- PCA is your first `fprintf(stderr, ... )`
- PCA diagonalises the covariance matrix $D \times D$
- MDS diagonalises the distance matrix $N \times N$

Summary

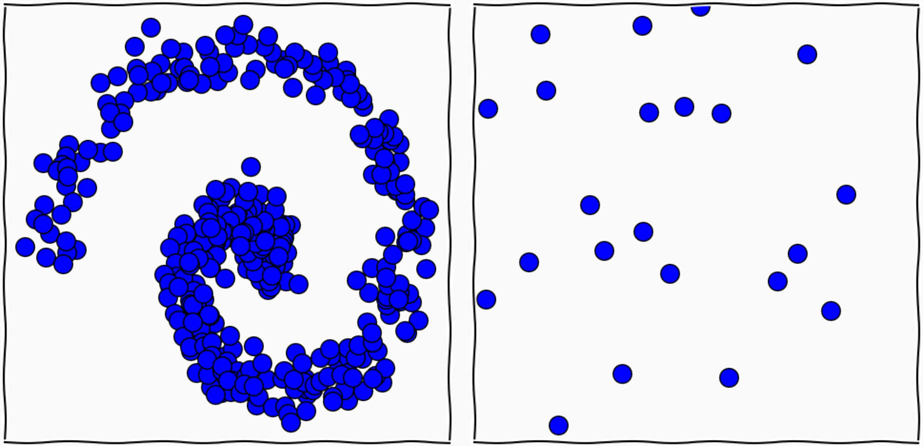
- Learn how to read distance matrices
- PCA is your first `fprintf(stderr, ... )`
- PCA diagonalises the covariance matrix $D \times D$
- MDS diagonalises the distance matrix $N \times N$
- You can non-linearise MDS with a non-linear distance measure

Latent Variable Models

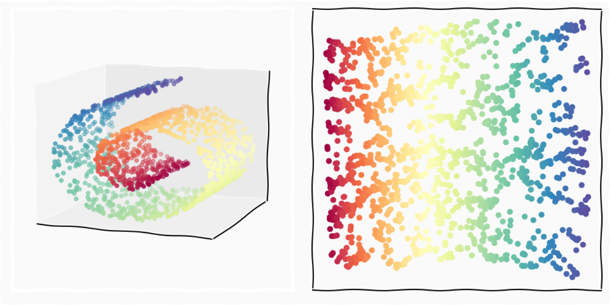
PCA vs MDS

- PCA is a global/linear method
- MDS allows for non-linearisation through localised measure

Locality



Generative Model

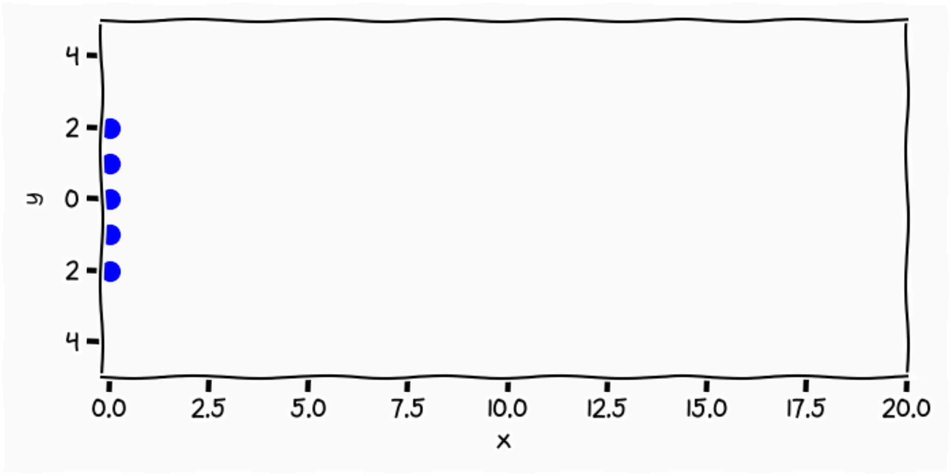


$$\mathbf{y}_i = f(\mathbf{x}_i)$$

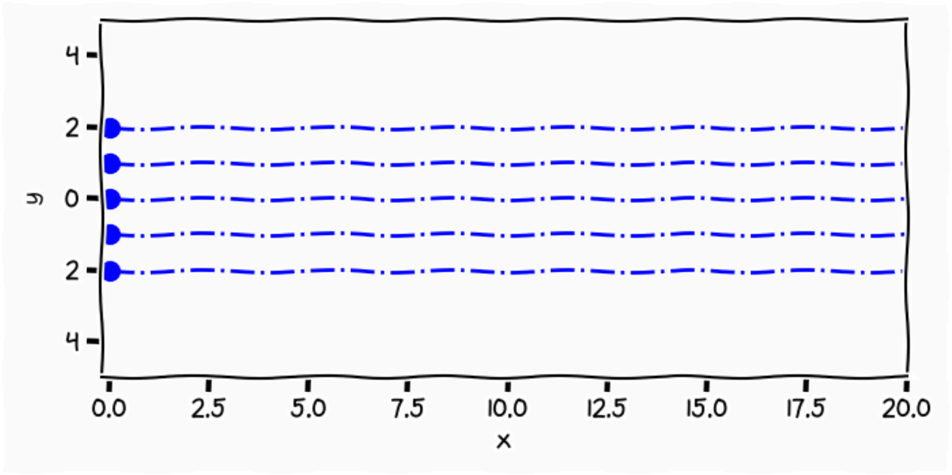
Unsupervised learning

$$y = f(x)$$

- In unsupervised learning we are given only output
- Task: recover both f and x


4
2
0
2
4
y
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
x

Unsupervised Learning

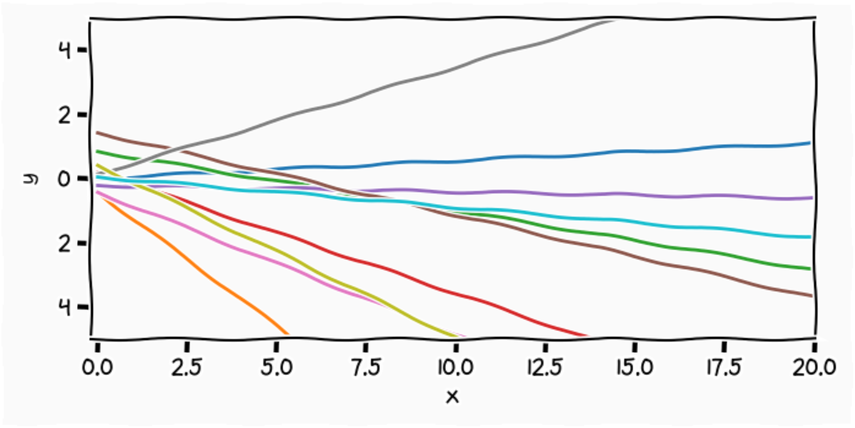


Solution bias

- This problem is very ill-posed
- We have to encode a preference towards the solution that we want
- Remember the `GLM`

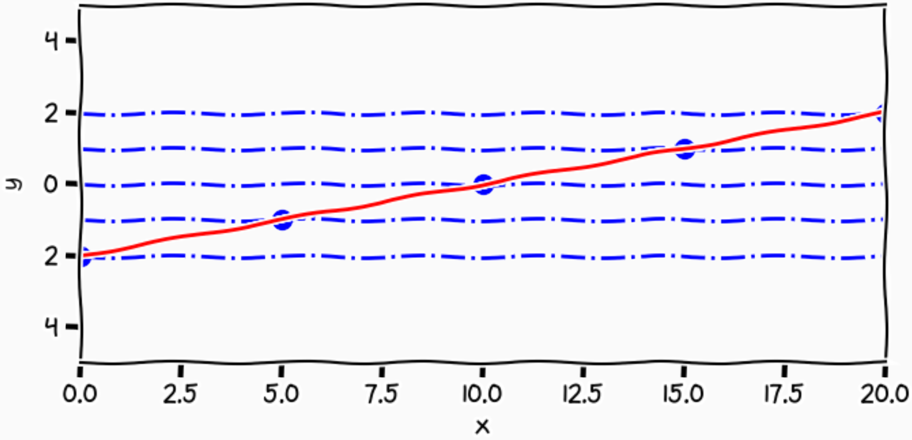
$$\hat{\boldsymbol{\beta}} = \operatorname*{argmax}_{\boldsymbol{\beta}} \prod_{i=1}^{N} p(y_i \mid \boldsymbol{\beta}, \mathbf{x}_i) + \lambda \left(\sum_{j=1}^{d} \beta_j^p \right)^{\frac{1}{p}}$$

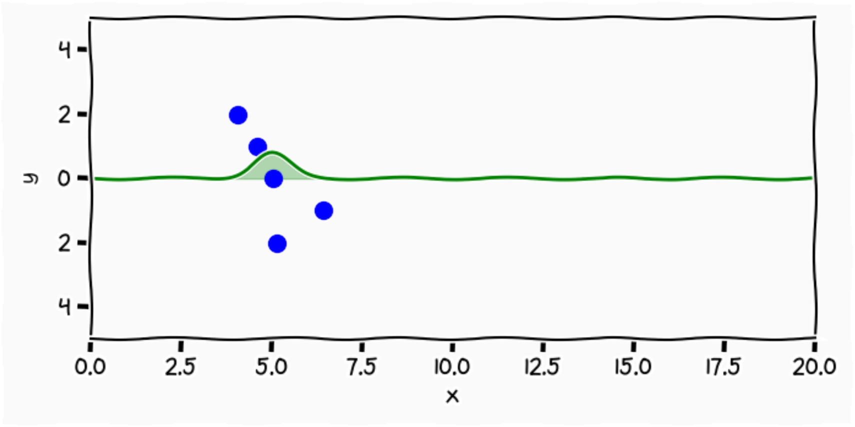
Unsupervised Learning



$$p(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$$

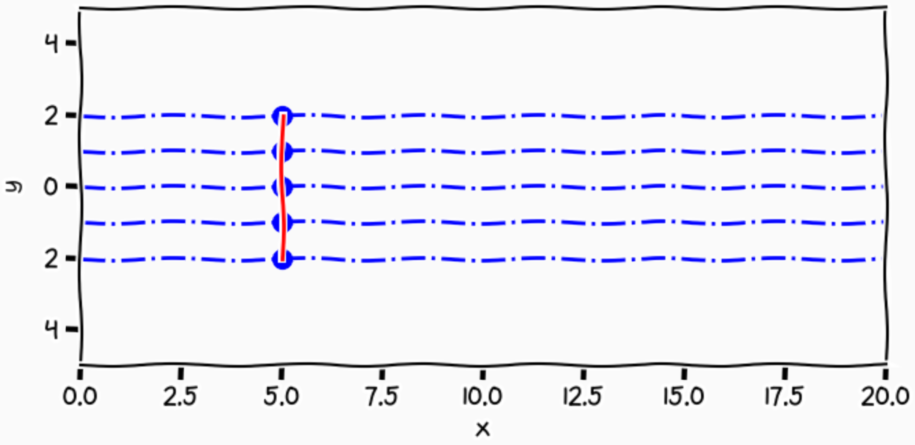
Unsupervised Learning



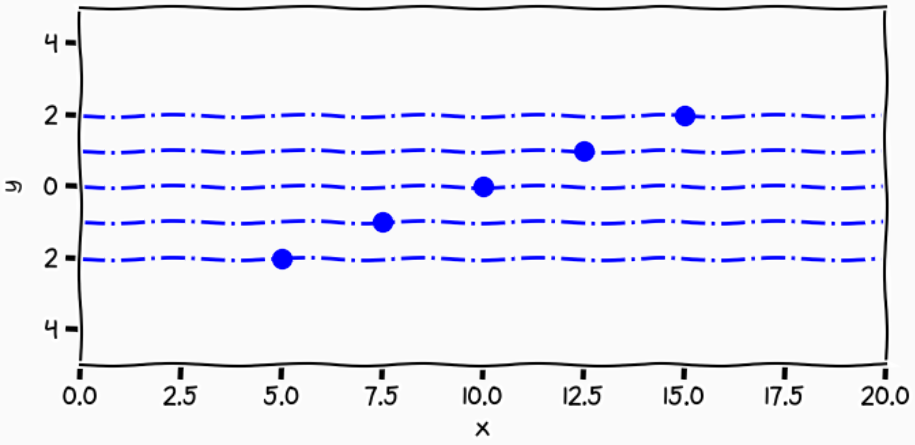


$$p(\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \alpha_2 \mathbf{I})$$

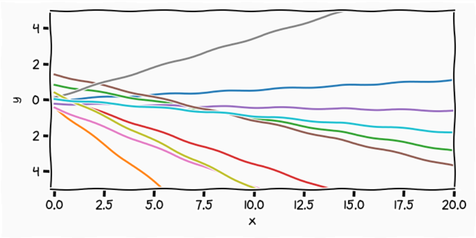
Unsupervised Learning

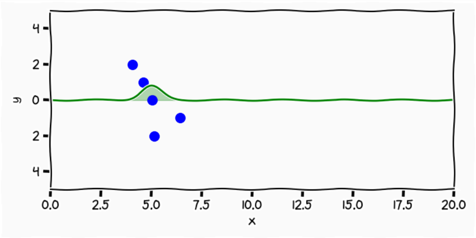


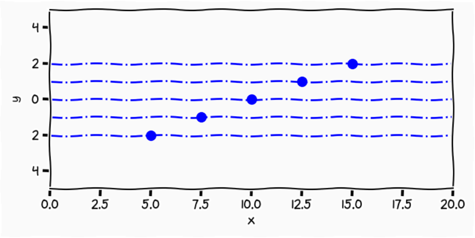
Unsupervised Learning



Unsupervised Learning







Principled Incorporation of Bias

- Bayes' Rule

$$p(f, \mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid f, \mathbf{X})p(f)p(\mathbf{X})}{p(\mathbf{Y})}$$

Principled Incorporation of Bias

- Bayes' Rule

$$p(f, \mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid f, \mathbf{X}) p(f) p(\mathbf{X})}{p(\mathbf{Y})}$$

- Maximum a posteriori estimate (MAP)

$$\{\hat{f}, \hat{\mathbf{X}}\} = \underset{f, \mathbf{X}}{\operatorname{argmax}} \log p(\mathbf{Y} \mid f, \mathbf{X}) + \underbrace{\log p(f) + \log p(\mathbf{X})}_{\text{regularisers}}$$

Principled Incorporation of Bias

- Bayes' Rule

$$p(f, \mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid f, \mathbf{X}) p(f) p(\mathbf{X})}{p(\mathbf{Y})}$$

- Maximum a posteriori estimate (MAP)

$$\{\hat{f}, \hat{\mathbf{X}}\} = \underset{f, \mathbf{X}}{\operatorname{argmax}} \log p(\mathbf{Y} \mid f, \mathbf{X}) + \underbrace{\log p(f) + \log p(\mathbf{X})}_{\text{regularisers}}$$

- GLM

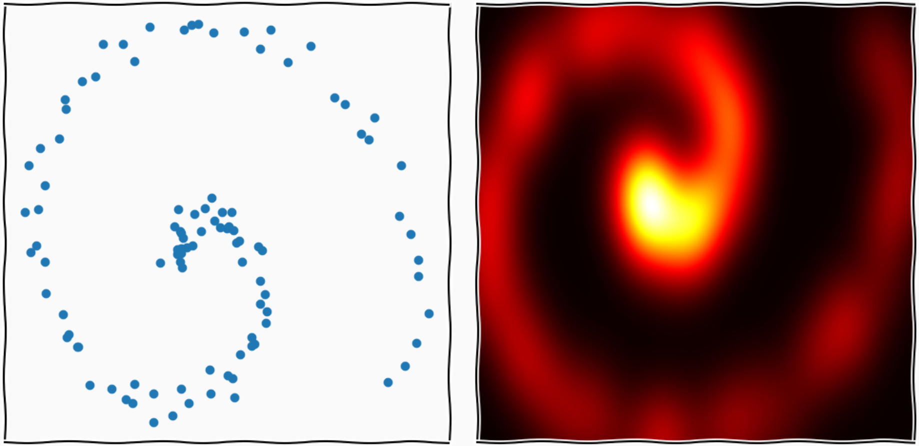
$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid \boldsymbol{\beta}, \mathbf{x}_i) + \lambda \left(\sum_{j=1}^{d} \beta_j^p \right)^{\frac{1}{p}}$$

Linear Model

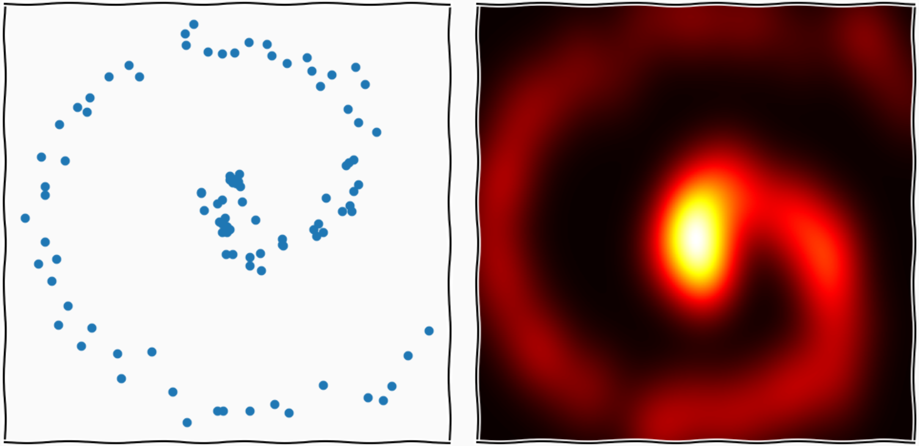
$$p(\mathbf{Y}, \mathbf{W}, \mathbf{X}) = p(\mathbf{Y}|\mathbf{W}, \mathbf{X})p(\mathbf{X})p(\mathbf{W})$$
$$p(\mathbf{Y}|\mathbf{W}, \mathbf{X}) = \mathcal{N}(\mathbf{XW} + \mu, \beta^{-1}\mathbf{I}),$$

- we assume the data is corrupted by Gaussian noise we get a likelihood
- we assume the mapping to be linear such that $\mathbf{Y} = \mathbf{XW}$

Example



Example II



Principal Component Analysis [2]

$$\mathbf{V}\Lambda\mathbf{V}^{\mathrm{T}} = \mathbf{y}^{\mathrm{T}}\mathbf{y}$$

$$\mathbf{y} = \sum_{i}^{d} \mathbf{y}\mathbf{V}_i$$

- The above is the solution if $\beta \to \infty$

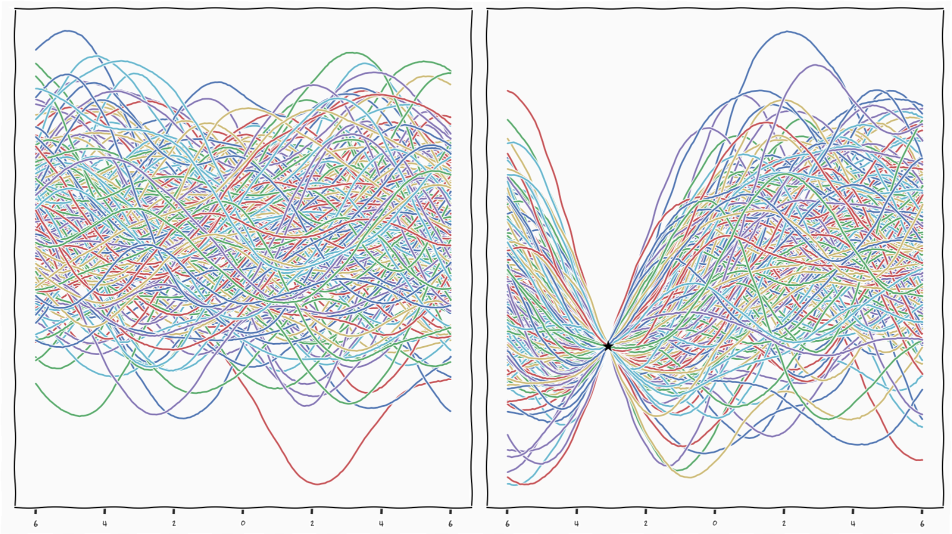
[2] Spearman, 1904

Principal Component Analysis

- You have seen this explained in two different way
  - *Retain variance*
  - *Gaussian priors*
- The statistical model provides a clearer intuition to the assumptions

- You have seen this explained in two different way
  - *Retain variance*
  - *Gaussian priors*
- The statistical model provides a clearer intuition to the assumptions
- *what about non-linearities*

What about non-linear methods



Example

Font Demo

Summary

Summary

- Visualisation is key to get insight into high-dimensional data

Summary

- Visualisation is key to get insight into high-dimensional data
- Unsupervised learning is inherently ill-posed

Summary

- Visualisation is key to get insight into high-dimensional data
- Unsupervised learning is inherently ill-posed
- Solutions can only be interpreted in light of the assumptions/bias that lead to the solution

Summary

- Visualisation is key to get insight into high-dimensional data
- Unsupervised learning is inherently ill-posed
- Solutions can only be interpreted in light of the assumptions/bias that lead to the solution
- PCA is a linear (global) model with a clear underlying statistical interpretation

Summary

- Visualisation is key to get insight into high-dimensional data
- Unsupervised learning is inherently ill-posed
- Solutions can only be interpreted in light of the assumptions/bias that lead to the solution
- PCA is a linear (global) model with a clear underlying statistical interpretation
- Non-linearisation through MDS can be very useful

EOF

eof