STAT5003

Week 4 : High dimensional visualization and analytics

Dr. Justin Wishart

Semester 2, 2020

THE UNIVERSITY OF SYDNEY

Readings

- In James, Witten, Hastie, and Tibshirani (2013)
  - PCA Dimension reduction, see Section 10.2
  - Clustering, see Section 10.3
- In Hastie, Tibshirani, and Friedman (2017)
  - MDS, see Section 14.8

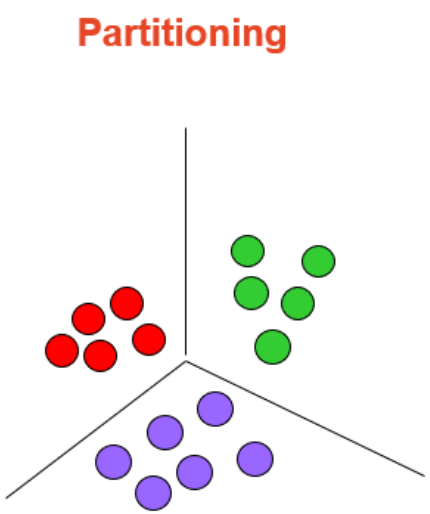
Clustering

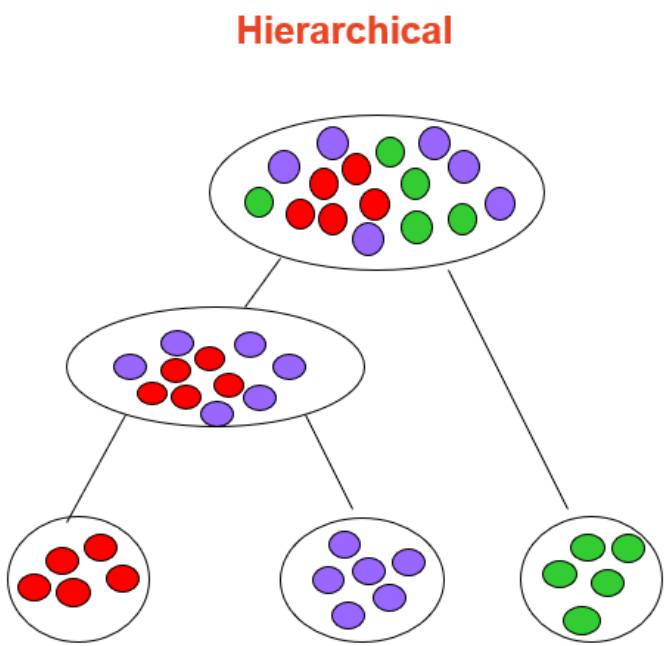
THE UNIVERSITY OF SYDNEY

Clustering basics

- Group observations that are similar based on predefined criteria.
- Requires a similarity or dissimilarity measure
- Goals of clustering:
  - We want clusters to be compact.
  - Small distance between observations within a cluster
  - Large distance between observations between different clusters
- Example algorithms:
  - Hierarchical clustering
  - k-means clustering
  - Gaussian mixture model

Typical methods





- Partitioning
  - Pre-specified number K of mutually exclusive and exhaustive groups.
  - Iterate until criteria is met.
- Hierarchical methods. Two paradigms
  - Agglomerative: Bottom up, more popular
  - Divisive: Top down, less popular
  - Display results with dendrogram

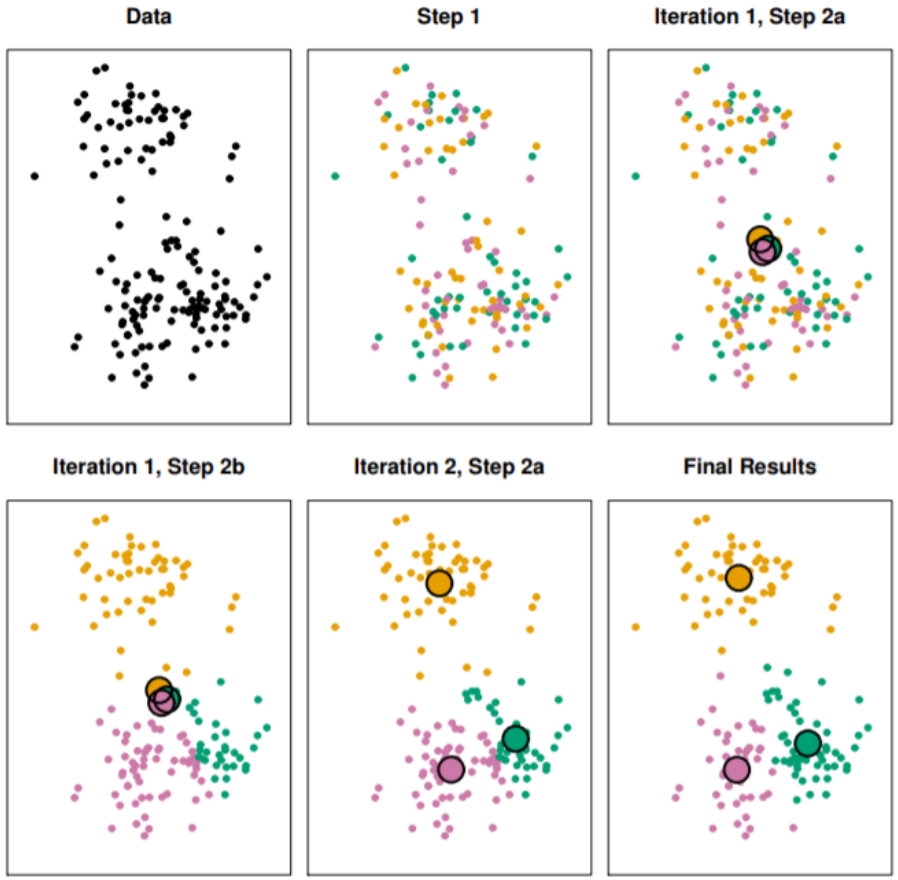
k-means approach

- Initialize each observation at random to a cluster.
- Iterate the following until convergence.
  1. Find cluster means with cluster memberships fixed

$$\widehat{\overline{x}_j} = \text{argmin}_m \sum_{cluster(i)=j} ||x_i - m||^2$$

  2. Find cluster memberships with cluster means fixed

$$\widehat{cluster}(i) = \text{argmin}_k ||x_i - \widehat{\overline{x}_k}||^2$$



k-means properties

- The number of clusters K needs to be specified.
- Local solution and not necessarily global solution.
- Depends on starting values (the random starting values).
- Best for compact, spherical clusters.
- Does not work well when cluster sizes are different.

Choosing K

- For cluster C_k can define within-group sum of squares as:

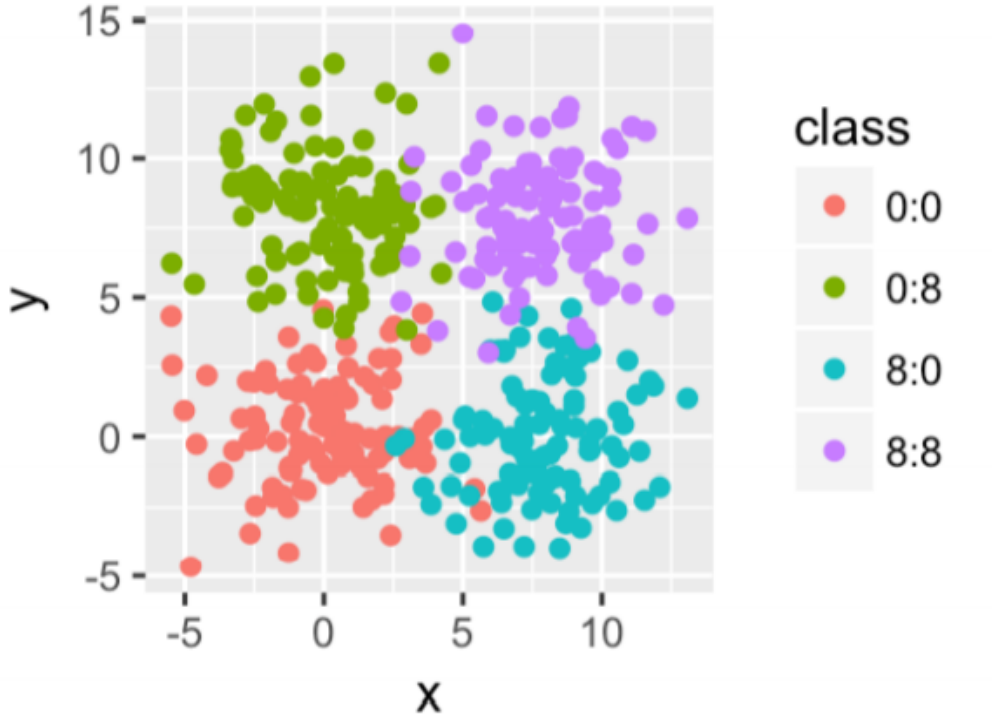
$$WSS_k = \frac{1}{|C_k|} \sum_{i,j \in C_k} ||x_i - x_j||^2$$

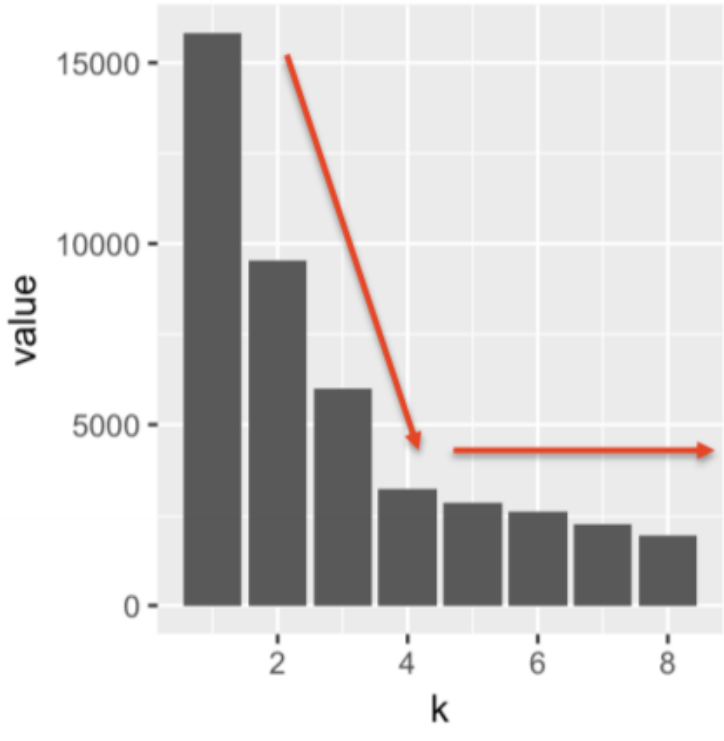
- This is the sum of all the pairwise squared Euclidean distances between observations in the k^{th} cluster, divided by total number of observations in the k^{th} cluster.
- The total within sum of squares criterion aggregates this metric across

$$WSS_{Total} = \sum_{k=1}^{K} WSS_k$$

- The total within sum of square criterion will decrease as k in increases.
- Rule of thumb: Look for the elbow

Elbow plot

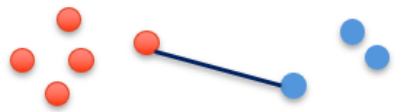

15
10
5
0
-5
y
-5
0
5
10
x
class
0:0
0:8
8:0
8:8


15000
10000
5000
0
value
2
4
6
8
k

Hierarchical Clustering

- Begin with every observation representing a single cluster.
- At each iteration, merge the two closest clusters into one cluster.
  - Needs a measure of similarity/dissimilarity between two clusters
  - These measures are called linkages.

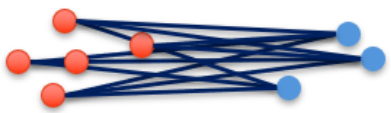
- Linkages - Measure of dissimilarity between two sets of objects that determine how two set of objects are merged.
  - Single linkage.
  - Complete linkage.
  - Average Linkage.



Single (minimum)

Complete (maximum)

Distance between centroids



Average (mean) linkage

Dimension reduction: Principal Components Analysis (PCA)

THE UNIVERSITY OF SYDNEY

High dimensional data

- High-dimensional data refers to data set with more features p than observations n
  - Examples: in genetic data, we can easily measure 500k individual DNA mutations (human genome have ~3 billion base pairs of DNA), but experiments generally have < 1000 people e.g. $p \sim 500k$, $n \sim 1000$
- It is very hard to visualize high-dimensional data
  - Only have 2 (sometimes 3 or 4) dimensional canvas to create plots.
- Many algorithms and methods have been designed for low dimensional data and would not work well for high-dimensional data
- To build a linear regression model data with $500k$ features will result in $500k$ parameters. This problem is underdetermined if we only have 1000 observations.

Dimension reduction

- Dimension reduction can be a pre-processing step, do it before applying clustering, classification and/or regression
- Data with small number of dimensions are easier to visualize and plot
- Dimension reduction can be a useful exploratory data analysis tool.

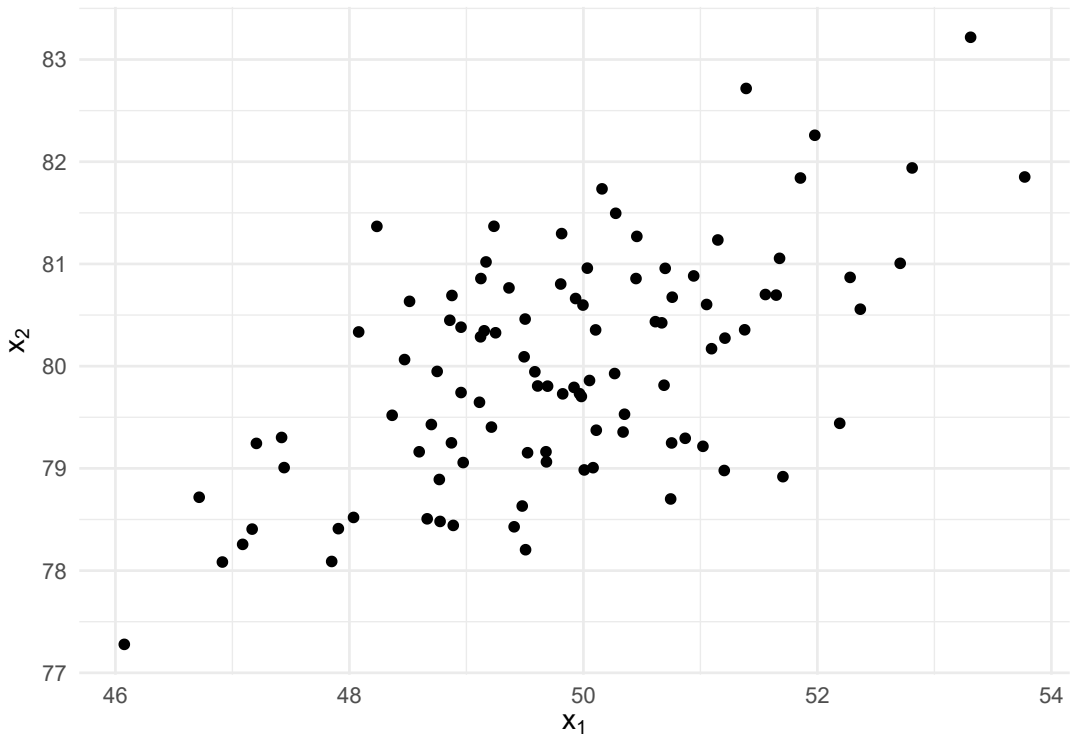
Dimension reduction strategies

- Eliminate or remove features
  - Need to decide which features to be eliminated? Keep ones with high variance?
- Select features
  - E.g. Lasso and ridge regression (coming soon in later module)
- Build or construct new features from existing ones
  - Replace many existing features with a single one.
  - PCA and t-SNE

PCA

- Suppose we have a data matrix X with n observations and p features.
  - Can we plot the data in a 2-dimensional plot?
- Naively, we can do all pairwise combinations (1 vs 2, 1 vs 3, . . . , (p vs $(p-1)$))
  - $\binom{p}{2} = \frac{p(p-1)}{2} = \mathcal{O}(p^2)$ different plots!
- Principal components analysis (PCA) finds a way to represent the data in a different space
  - It is still p dimensional, albeit a different coordinate space.
  - Aims to explain most variation in the first few dimensions.

Best way to represent 2d in 1d?



- Could use a single variable? x_1 say?
- Or could remap x_1 and x_2 to a single variable
  - $z = \phi_1 x_1 + \phi_2 x_2$
- Can generalize this to many dimensions.
  - $z = \sum_{i=1}^{p} \phi_i x_i$
- Pick the transformation that maximises the variance!

Principal Components

Start with a data matrix $\boldsymbol{X}$, assume it has mean zero.

$$\boldsymbol{X} = \begin{pmatrix} X_1 & X_2 & \dots & X_p \end{pmatrix}$$

The first principal component is the normalised linear combination of the features that maximises the variance in the new component.

$$Z = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p = \sum_{i=1}^{p} \phi_{i1} X_i = \boldsymbol{\phi}_1^T \boldsymbol{X}$$

The elements ϕ_{i1} are known as the loadings of the first principal component

- By normalised, we mean the squared loadings have to sum to 1, i.e. $\sum_{i=1}^{p} \phi_{i1}^2 = 1 \Leftrightarrow \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 = 1$
- Also, it is desired to maximise

$$\mathbb{V}\text{ar}(Z_1) = \mathbb{V}\text{ar}(\boldsymbol{\phi}_1^T \boldsymbol{X}) = \sum_{i=1}^{p} \phi_{i1}^2 \mathbb{V}\text{ar}(X_i) + \sum_{i \neq j} \phi_{i1} \phi_{j1} \mathbb{C}\text{ov}(X_i, C_j)$$

Solving the first principal component (not assessable)

- To find the first principal component, solve the following optimization problem

$$\max_{\phi_{11},\ldots\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \quad \text{such that} \quad \sum_{i=1}\phi_{i1}^2 = 1 \tag{1}$$

- Define the Covariance matrix

$$\boldsymbol{\Sigma} = \mathbb{V}\text{ar}(\boldsymbol{X}) = \begin{pmatrix} \mathbb{V}\text{ar}(X_1) & \mathbb{C}\text{ov}(X_1, X_2) & \cdots & \mathbb{C}\text{ov}(X_1, X_p) \\ \mathbb{C}\text{ov}(X_1, X_2) & \mathbb{V}\text{ar}(X_2) & \cdots & \mathbb{C}\text{ov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}\text{ov}(X_p, X_1) & \cdots & \cdots & \mathbb{V}\text{ar}(X_p) \end{pmatrix}$$

- Also, $\mathbb{V}\text{ar}(Z_1) = \mathbb{V}\text{ar}(\boldsymbol{\phi}_1^T\boldsymbol{X}) = \boldsymbol{\phi}_1^T\boldsymbol{\Sigma}\boldsymbol{\phi}_1$ and the above optimization is equivalent to

$$\max_{\boldsymbol{\phi}_1} \boldsymbol{\phi}_1^T\boldsymbol{\Sigma}\boldsymbol{\phi}_1 \quad \text{such that} \quad \boldsymbol{\phi}_1^T\boldsymbol{\phi}_1 = 1$$

Solving via multivariable calculus (not assessable)

- Can solve this with multivariable calculus! The Lagrangian.

$$L(\boldsymbol{\phi}, \lambda) = \boldsymbol{\phi}_1^T \boldsymbol{\Sigma} \boldsymbol{\phi}_1 + \lambda(1 - \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1)$$

- Computing partial derivatives and solving

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\phi}_1} &= 2\boldsymbol{\Sigma}\boldsymbol{\phi}_1 - 2\lambda\boldsymbol{\phi}_1 = \mathbf{0}, & \frac{\partial L}{\partial \lambda} &= 1 - \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 = 0 \\ &\Updownarrow & &\Updownarrow \\ \boldsymbol{\Sigma}\boldsymbol{\phi}_1 &= \lambda\boldsymbol{\phi}_1 & \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 &= 1 \end{aligned}$$

This is the eigenvalue equation. The eigenvector of $\boldsymbol{\Sigma}$ gives the loadings.

Solving the second principal component (not assessable)

- Can repeat the process to get the next principal component.
- Find $\boldsymbol{\phi}_2$ to optimise

$$\max_{\phi_{12},\ldots\phi_{p2}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j2}x_{ij}\right)^2 \quad \text{such that} \quad \sum_{i=1}\phi_{i2}^2 = 1 \quad \text{and} \quad \sum_{i=1}\phi_{i1}\phi_{i2} = 0$$

- Or using vector notation

$$\max_{\boldsymbol{\phi}_2} \boldsymbol{\phi}_2^T\boldsymbol{\Sigma}\boldsymbol{\phi}_2 \quad \text{such that} \quad \boldsymbol{\phi}_2^T\boldsymbol{\phi}_2 = 1 \quad \text{and} \quad \boldsymbol{\phi}_2^T\boldsymbol{\phi}_1 = 0$$

Principal Component Scores

- Given the principal component loadings, we can project our data matrix $\boldsymbol{X}$ onto the principal component space.
  - The projection is a linear combination of the sample feature values:

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots \phi_{p1} x_{ip}$$

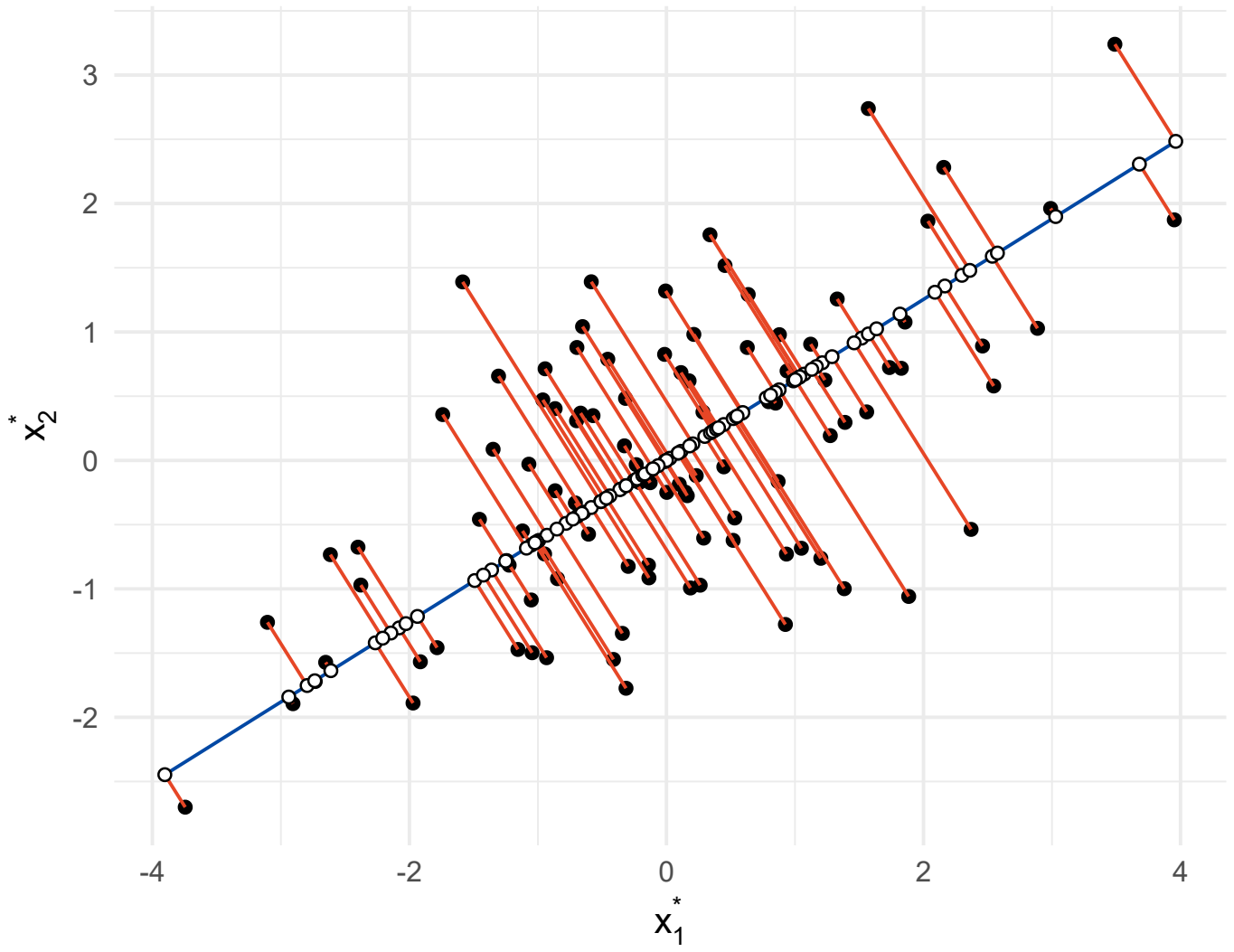
    This is known as the principal component score.

- The first principal component score vector is

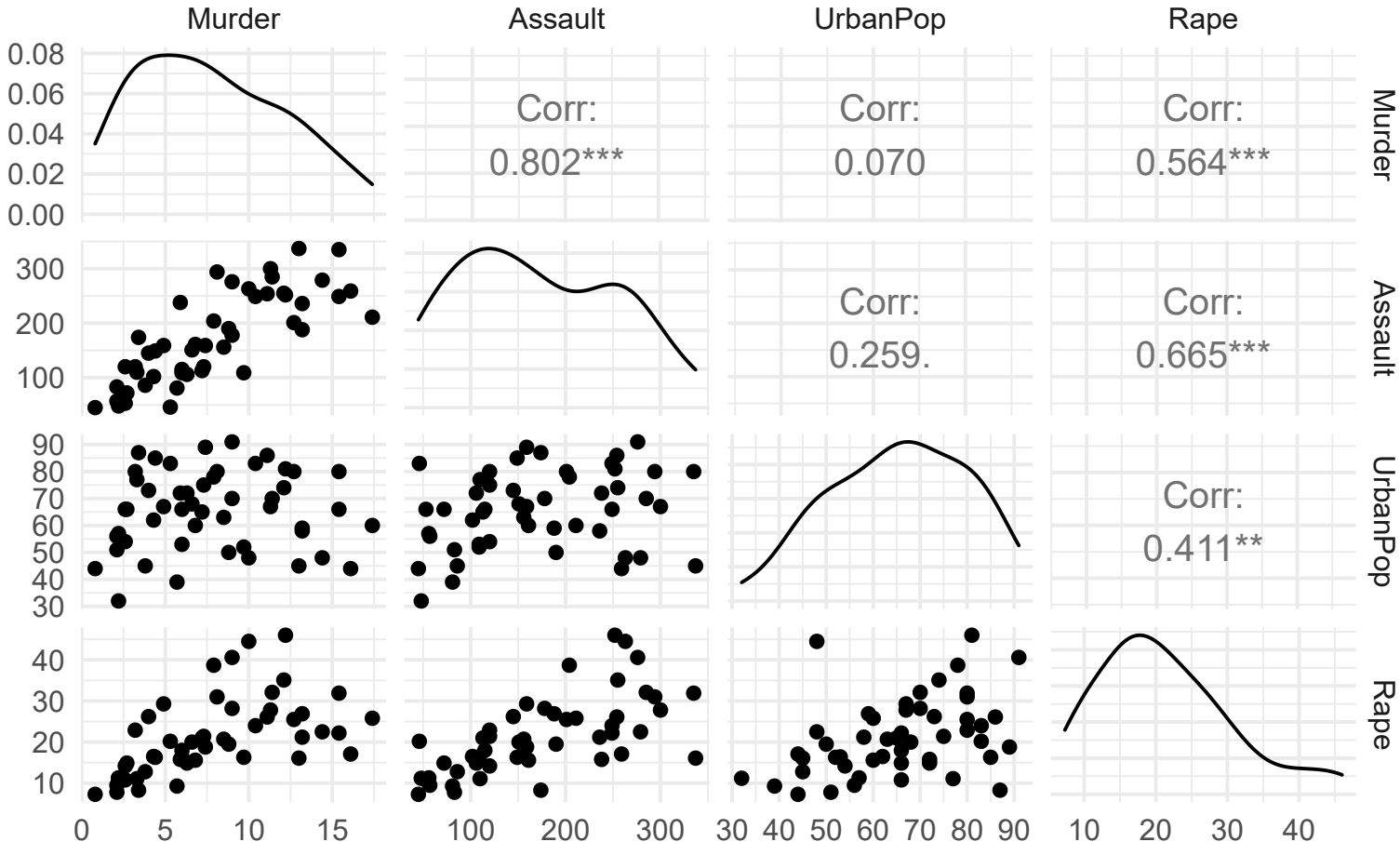
$$\boldsymbol{Z}_1 = (z_{11}, z_{21}, \ldots, z_{n1})$$

  The principal component score vectors are uncorrelated.

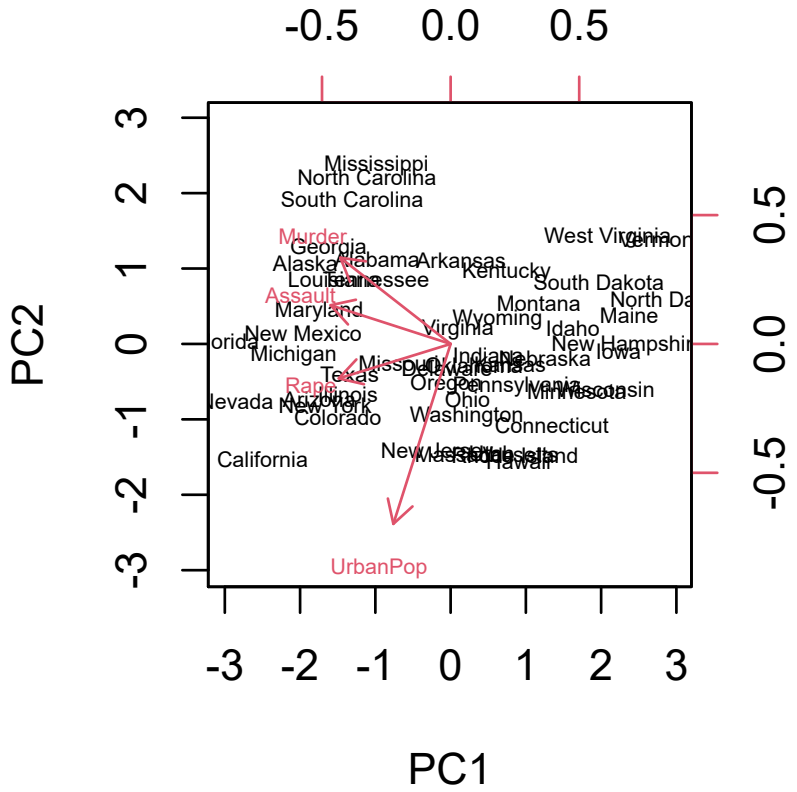
Geometric interpretation



USArrests Example



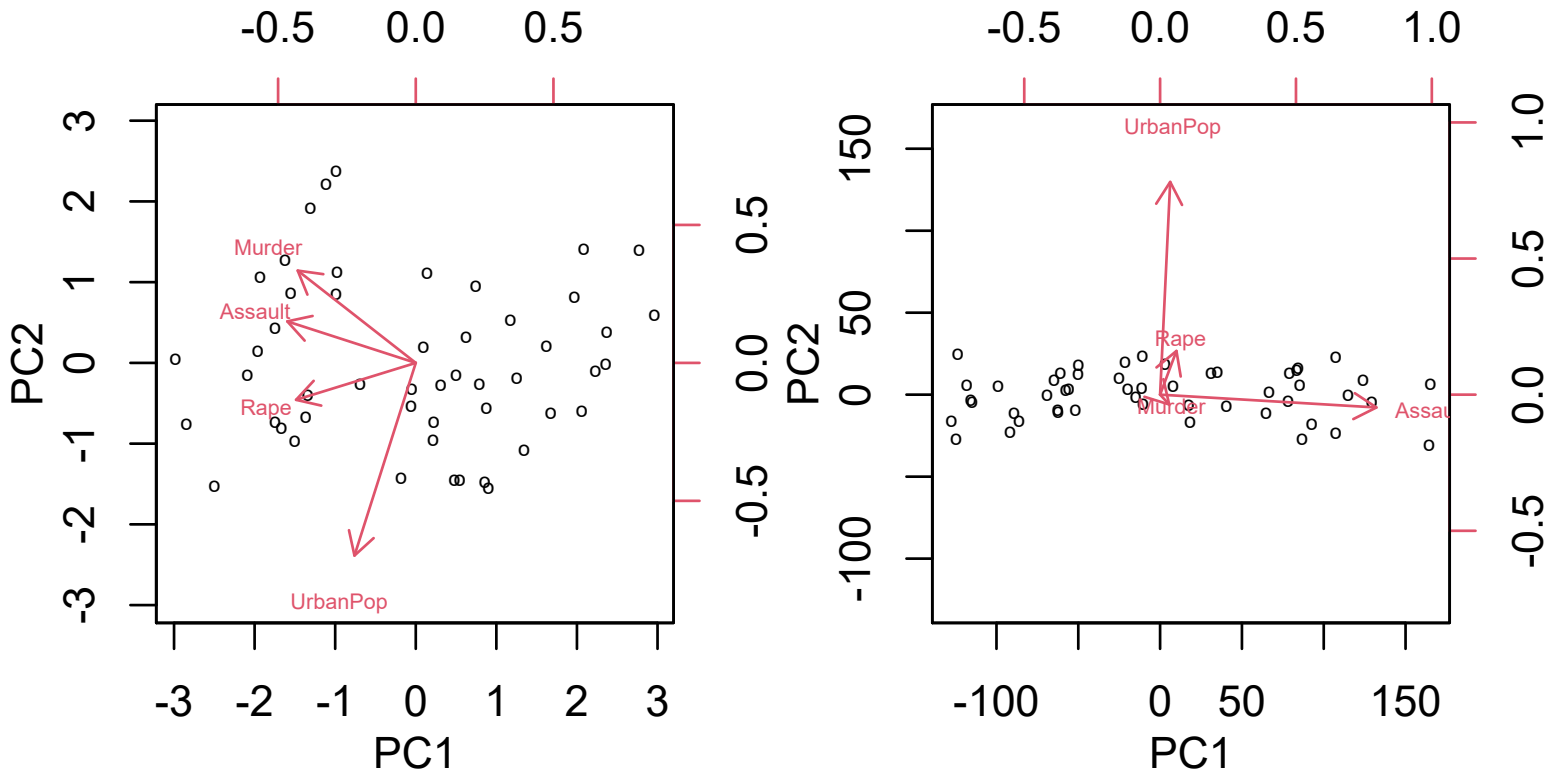
Biplot of the USArrests



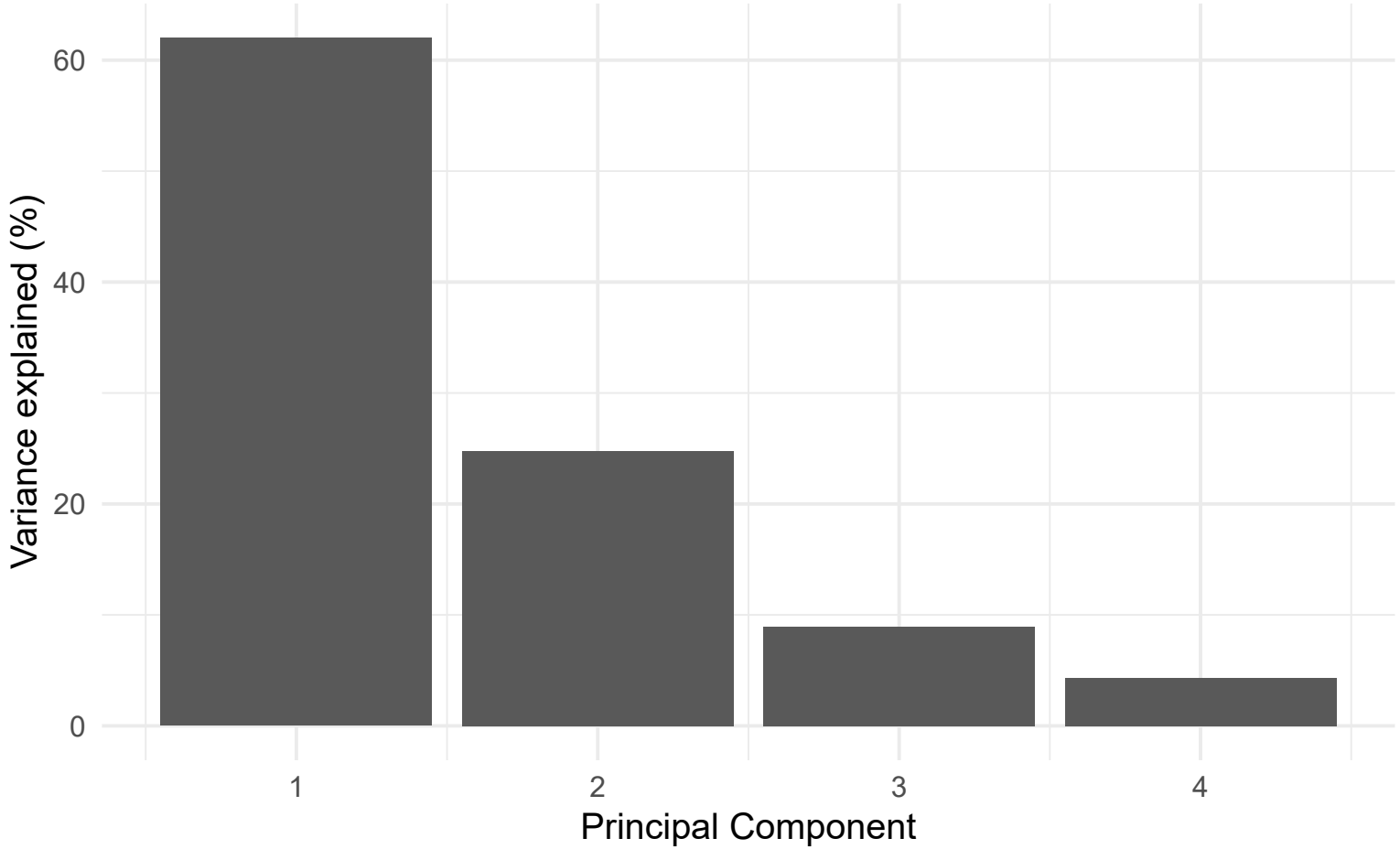
Scaling the variables

- In a PCA analysis, it is common to centre the variable by removing the mean
- You can also standardise the data to make all the variables have a standard deviation of 1
- If the variables have different units (e.g. in the USArrests dataset, Murder is measured as number per 100,000 people, but UrbanPop is the percentage of population that lives in urban area), the variance would be very different
- The loadings will put more weight on variables with higher variance
    - this may not be what you want!
- However, if all the variables share the same unit, then standardisation may not be necessary

Effect of scaling (left) vs unscaled (right)



Scree plot



Properties of PCA

- **Unique** and **Global** solution!
- Ordered components
- Best low rank approximation to the data

$$\min_{\widehat{X}} ||X - \widehat{X}||_F^2 \quad \text{such that } rank(\widehat{X}) = p$$

- Best linear dimension reduction possible
- Is not the best for non-linear relationships

PCA with K-means

- Very common approach to deal with high dimensional data
- Use the first M principal component scores as inputs into the kmeans algorithm (M << p)
- Can help improve the clustering model if the signal in the data can be captured in a few principal components

PCA with regression

- Use the first M principal component scores as the predictors in a linear regression model
- We are assuming that a small number of principal components can explain most of the variability in the data as well as the response
- PCR is useful when variables in the data are highly correlated (i.e. collinear)

Dimension reduction t-SNE

THE UNIVERSITY OF SYDNEY

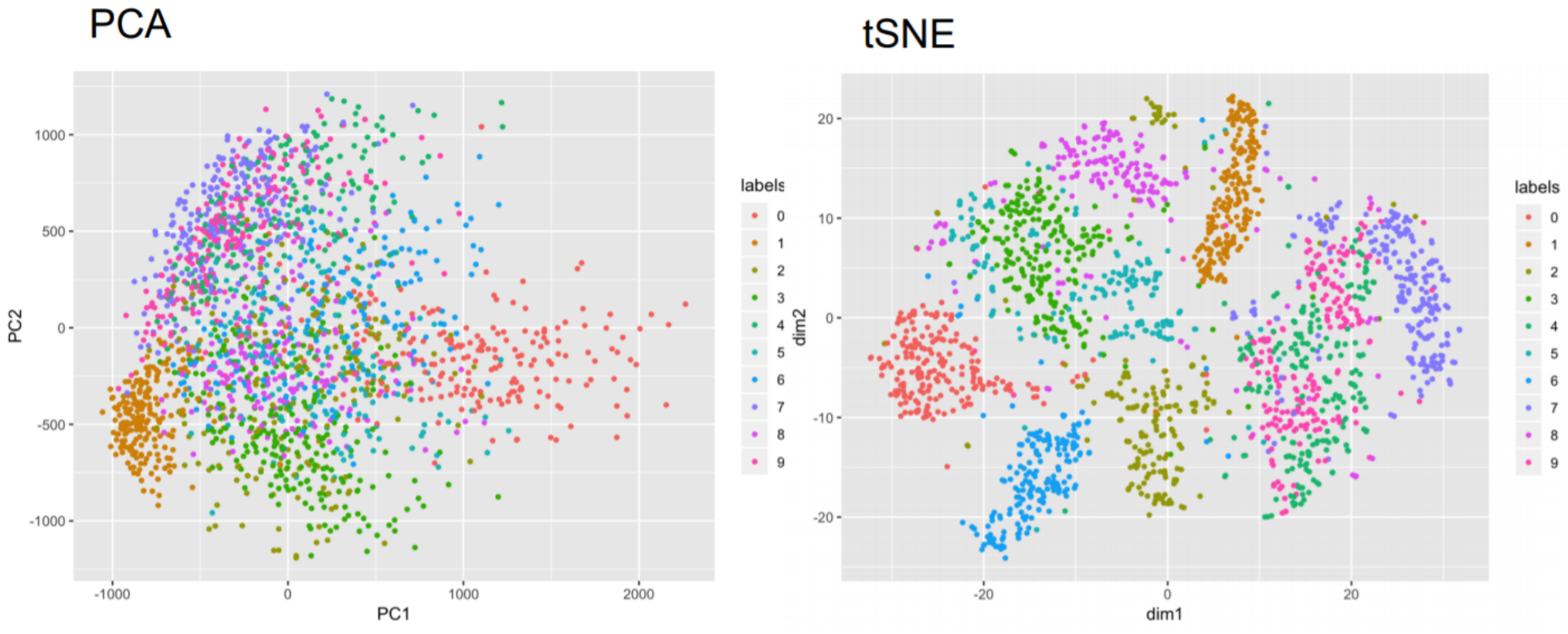
t-SNE

- Non-linear technique developed for visualizing high dimensional datasets
- Uses local structure in the data to find a low dimensional representation
- Applications include computer security research, music analysis, cancer research, bioinformatics, and biomedical signal processing.

MNIST Example

- 28 by 28 pixel images of handwritten digits
- 784 features

MNIST Example



t-SNE vs PCA

- t-SNE is a probabilistic method – it will give you a different representation every time you run it.
- PCA is defined by a mathematical formula
- t-SNE is mostly a visualization method. The PCs from PCA can be interpreted whereas t-SNE representation cannot be used for inference.
- t-SNE is more computationally intensive than PCA
- PCA is a linear method so can only capture linear relationships whereas t-SNE can find more complicated non-linear relationships

Three steps in t-SNE

1. Constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked while dissimilar points have an extremely small probability of being picked.
2. Defines a similar probability distribution over the points in the low-dimensional map.
3. Minimizes the Kullback–Leibler divergence between the two distributions with respect to the locations of the points in the map.

Recommended to view the guide at https://distill.pub/2016/misread-tsne/ for more information on using the t-SNE framework.

More details: Step 1 (Not assessable)

Given a set of n high dimensional objects $x_1, x_2, \ldots, x_n$ in p-dimensional space, t-SNE first computes probabilities p_{ij} that are proportional to the similarities of objects x_i and x_j as follows:

$$p_{j|i}(\sigma_i^2) = \frac{\phi(x_j; x_i, \sigma_i^2)}{\sum_{k \neq i} \phi(x_k; x_i, \sigma_i^2)}$$

- $\phi(x; \mu, \sigma^2)$ denotes the Gaussian density.
- Think of $p_{j|i}(\sigma_i^2)$ as a conditional probability that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i and with variance σ_i^2
- The conditional probability can be made symmetric with

$$pij = \frac{p_{j|i} + p_{i|j}}{2n}$$

More details: Step 2 (Not assessable)

- Learn a lower dimensional representation of the data: $y_1, y_2, \ldots, y_n$ that preserves the similarities p_{ij} as much as possible
- In the low dimensional space, use the heavy-tailed Student t-distribution with one degree of freedom to define similarities.
- Hence define similarities q_{ij} in low dimensional space as:

$$q_{ij} = \frac{(1 + ||y_j - y_i||_2^2)^{-1}}{\sum_{k \neq i}(1 + ||y_j - y_i||_2^2)^{-1}}$$

More details: Step 3 - Kullback-Leibler divergence (Not assessable)

- Find the low-dimensional representation $y_1, y_2, \ldots, y_n$ that minimizes the Kullback-Leibler (KL) divergence of the distribution q_{ij} from p_{ij}.
- KL divergence is a non-symmetric measure of the difference between two probability distributions
- KL divergence is defined as:

$$KL(p||q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

  Think of KL divergence as a measure of how many bits of information is lost when we use q to approximate p

Dimension reduction: Multidimensional Scaling (MDS)

THE UNIVERSITY OF SYDNEY

Multidimensional Scaling (MDS)

- Visually represent proximities (similarities or distances) between objects in a lower dimensional space. (usually 2 or 3d space)
- The objective of MDS is to take a Matrix of similarities or dissimilarities, $\boldsymbol{D}$, and find projections $z_1, \ldots, z_k$ where k is the desired lower dimension.
- The distances are near preserved by optimizing a stress function
- Full data not required

Multidimensional Scaling (MDS) Example

	Adelaide	Alice	Brisbane	Cairns	Canberra	Darwin	Melbourne	Perth	Sydney
Adelaide	0	1533	2044	3143	1204	3042	728	2725	1427
Alice	1533	0	3100	2500	2680	1489	2270	3630	2850
Brisbane	2044	3100	0	1718	1268	3415	1669	4384	1010
Cairns	3143	2500	1718	0	2922	3100	3387	5954	2730
Canberra	1204	2680	1268	2922	0	3917	647	3911	288
Darwin	3042	1489	3415	3100	3917	0	4045	4250	3991
Melbourne	728	2270	1669	3387	647	4045	0	3430	963
Perth	2725	3630	4384	5954	3911	4250	3430	0	4110
Sydney	1427	2850	1010	2730	288	3991	963	4110	0

Multidimensional Scaling (MDS) Example

```
mds <- cmdscale(city.dist, k = 2); colnames(mds) <- c("x", "y")
mds <- data.frame(mds, City = colnames(city.dist))
library(ggrepel)
ggplot(mds, aes(x = x, y = y, label = City)) + geom_point() + geom_text_repel() + theme_minimal()
```

MDS Stress functions

- These functions attempt to force the lower dimensional projections to preserve the distances in the original data.
- Common stress functions
  - Least squares $S_{LS}(z_1, z_2) = \sqrt{\sum_{i \neq j}(d_{ij} - ||z_i - z_j||)^2}$

MDS Benefits/Drawbacks

- Full data not required, only its distance or dissimilarity matrix
- Need to choose K (could use the same elbow in Scree plot technique)
- Can be used as a visualization technique for non-linear data.

Interpretting MDS outputs

- Interpreting MDS maps:
  - Can be rotated (axes and orientation are somewhat arbitrary).
  - Only relative locations important.
  - Typically look for objects close in the MDS map

References

Hastie, T, R. Tibshirani, and J. Friedman (2017). *The elements of statistical learning: data mining, inference, and prediction*. Second Edition, 12th printing. Springer Science & Business Media.

James, G, D. Witten, T. Hastie, et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.