Applied Machine Learning (AML)

Exploratory Data Analysis

Oisin Mac Aodha • Siddharth N.

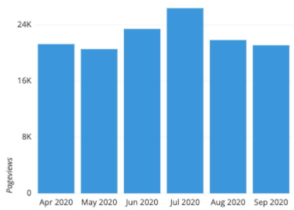
Data Visualisation

Plotting Data

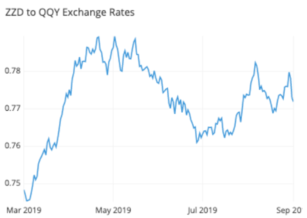
Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations

Bar chart



Line chart



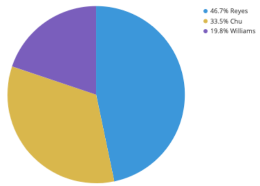
Figures: chartio.com

Plotting Data

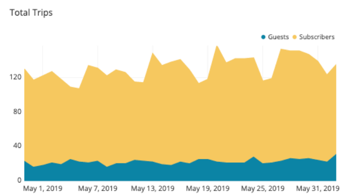
Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations

Pie chart



Stacked area chart



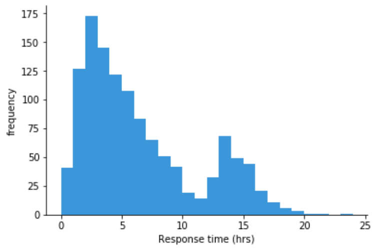
Figures: chartio.com

Plotting Data

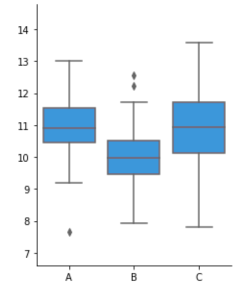
Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations

Histogram



Box plot



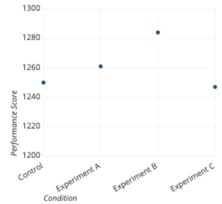
Figures: chartio.com

Plotting Data

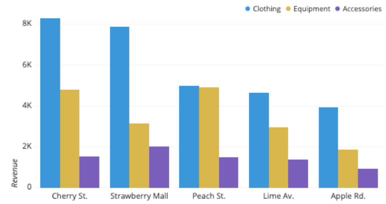
Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations

Point plot



Grouped bar chart



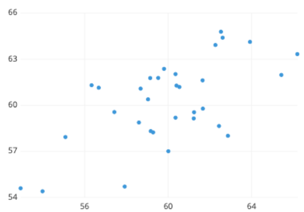
Figures: chartio.com

Plotting Data

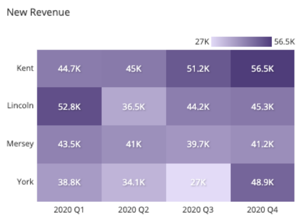
Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations

Scatter plot



Heat map



Figures: chartio.com

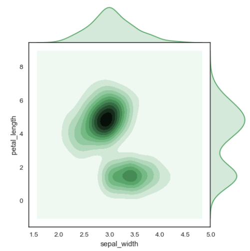
Plotting Data

Plot Types

- temporal change
- part-to-whole composition
- distribution
- group comparison
- inter-variable relations
- and more …

Wordcloud

Joint plot



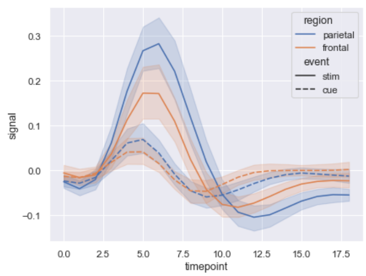
Figures: chartio.com

Features of a good plot

- title
- labelled axes
- axes ranges and ticks
- clarity (colour/thickness)
- legend

informative:
convey as much as necessary

clean:
avoid overfilling & redundancy



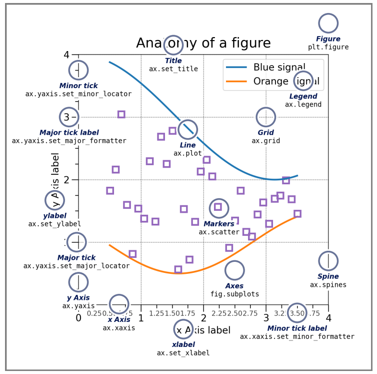
Figures: Matplotlib—anatomy of a figure

Features of a good plot

- title
- labelled axes
- axes ranges and ticks
- clarity (colour/thickness)
- legend

informative:
convey as much as necessary

clean:
avoid overfilling & redundancy



Figures: Matplotlib—anatomy of a figure

Features of a good plot

- title
- labelled axes
- axes ranges and ticks
- clarity (colour/thickness)
- legend

informative:
convey as much as necessary

clean:
avoid overfilling & redundancy

Relatively easy to think about when data is low dimensional

What do we do when data is high dimensional?

Figures: Matplotlib—anatomy of a figure

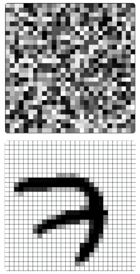
Dimensionality Reduction

Curse of Dimensionality

Manifold Hypothesis

High-dimensional data in the real world really lies on low-dimensional manifolds within that high-dimensional space.

- Data is typically high dimensional
  vision: 10^4 pixels, text: 10^6 words
- Example: handwritten digits (MNIST)
  - 28×28 pixels $\longrightarrow \{0,1\}^{784}$ possible "images"
  - only a very small number of these images are actually real
  - true dimensionality: actual variation of pen strokes!



Dealing with high dimensionality

Statistics

- ML involves some form of "counting" observations and features
  - count within some regions
    e.g. constructing histograms
  - use counts to construct predictors
    e.g. decision trees
- As dimensionality grows, fewer observations per region

Mitigation

- domain knowledge / feature engineering
- modelling assumptions about features
  independence, smoothness, symmetry
- reduce data dimensionality
  construct a new set of dimensions / variables

Dimensionality Reduction

Goal: Represent data using a "few" variables

- compression: preserve as much information/structure as possible
- discrimination: only keep information that enables task (e.g. classification)

Selection

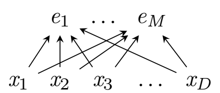
- subset of all features

$$x_1, x_2, x_3, \ldots, x_{D-1}, x_D$$

- relevant to task
  e.g. 'credit history' $\longrightarrow$ loan?

Transformation

- construct a new set of dimensions



$M \ll D$

- transformation of original
  e.g. linear $F \implies \boldsymbol{e} = F\boldsymbol{x}$

Dimensionality Reduction

PCA

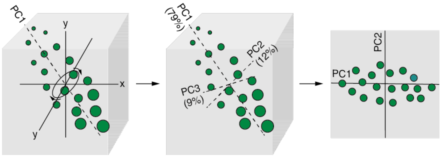
Principal Components Analysis (PCA)

Define principal components (PCs)

- 1^{st} PC: direction of *greatest* variation in the data
- 2^{st} PC: $\perp$ 1^{st} PC; greatest *remaining* variation

  …and so on until D, for $\boldsymbol{x} \in \mathbb{R}^D$.

- First $M \ll D$ components $\longrightarrow$ new basis dimensions
- …transform coordinates of each data point to new basis



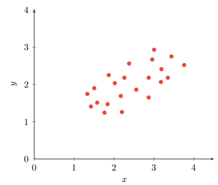
Rationale

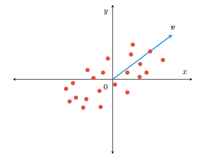
- variation along direction = *information*
- transform basis $\longrightarrow$ fit maximum information into M dimensions

PCA: Basics

$$X = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{bmatrix} \qquad X \in \mathbb{R}^{N \times D},\ \boldsymbol{x}_i \in \mathbb{R}^D \quad \text{(data)}$$

$$S = \frac{1}{N} X^\top X \qquad S \in \mathbb{R}^{D \times D} \quad \text{(covariance, assuming 0-mean)}$$





Intuition

Repeated transformation using the covariance (S) turns towards direction of maximum variance (example)

$$S\boldsymbol{v} = \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.2 \\ 0.2 \end{bmatrix} \overset{S}{=} \dots \overset{S}{=} \begin{bmatrix} -14.1 \\ -6.4 \end{bmatrix} \overset{S}{=} \begin{bmatrix} -33.3 \\ -15.1 \end{bmatrix}$$

where the *slope* converges to 0.454

Goal: Find $\boldsymbol{v}$ such that

$$S\boldsymbol{v} = \lambda \boldsymbol{v}$$

PCA: Maximising Variance

Recall $X\boldsymbol{v}$ projects X onto $\boldsymbol{v}$

$$\begin{aligned} \mathrm{Var}[X\boldsymbol{v}] &= \frac{1}{N}(X\boldsymbol{v})^\top (X\boldsymbol{v}) \\ &= \frac{1}{N}\boldsymbol{v}^\top X^\top X \boldsymbol{v} \\ &= \boldsymbol{v}^\top \frac{X^\top X}{N} \boldsymbol{v} \\ &= \boldsymbol{v}^\top S \boldsymbol{v} \end{aligned}$$

$max\ \boldsymbol{v}^\top S\boldsymbol{v}$, s.t. $\boldsymbol{v}^\top \boldsymbol{v} = 1$

solved using *Lagrange multipliers* as

$$\underbrace{max\ \boldsymbol{v}^\top S\boldsymbol{v} - \lambda\left(\boldsymbol{v}^\top \boldsymbol{v} - 1\right)}_{\mathcal{L}}$$

computing derivative w.r.t $\boldsymbol{v}$ and setting $= 0$

$$\frac{d\mathcal{L}}{d\boldsymbol{v}} = 2S\boldsymbol{v} - 2\lambda\boldsymbol{v} = 0$$

$$S\boldsymbol{v} = \lambda\boldsymbol{v} \quad \square$$

$\boldsymbol{v} \longrightarrow$ direction of max variance

$$S\boldsymbol{v} = \lambda\boldsymbol{v}$$

left multiply by $\boldsymbol{v}^\top$

$$\begin{aligned} \boldsymbol{v}^\top S\boldsymbol{v} &= \boldsymbol{v}^\top \lambda \boldsymbol{v} \\ &= \lambda \boldsymbol{v}^\top \boldsymbol{v} \\ &= \lambda \quad \square \end{aligned}$$

$\lambda \longrightarrow$ max variance

PCA: Finding Principal Components

More generally, solve for $SV = \Lambda V$ using Eigen decomposition

$$V = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_D],\ \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_D \end{bmatrix} \qquad \boldsymbol{v}_i \in \mathbb{R}^D,\ V \in \mathbb{R}^{D \times D},\ \Lambda^{D \times D}$$

Eigenvalues

Solve $|S - \lambda I| = 0$

$$\begin{vmatrix} 2.0 - \lambda & 0.8 \\ 0.8 & 0.6 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 2.6\lambda + 0.56 = 0$$

$$\implies \{\lambda_1, \lambda_2\} = \{2.36, 0.23\}$$

Eigenvectors

Find i^{th} eigenvector by solving $S\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = 2.36 \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} \implies \boldsymbol{v}_1 = \begin{bmatrix} 2.2 \\ 1 \end{bmatrix}$$

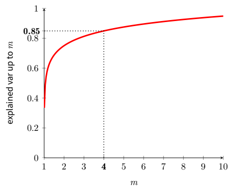
$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} = 0.23 \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} \implies \boldsymbol{v}_2 = \begin{bmatrix} -0.41 \\ 0.91 \end{bmatrix}$$

PCA: Picking number of dimensions

Given: eigenvectors $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D]$; **Require:** $M \ll D$
Known: eigenvalue λ_i = variance along $\boldsymbol{v}_i$

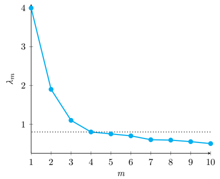
Explained variance

- sort eigenvectors s.t. $\lambda_1 \geq \ldots \geq \lambda_D$
- choose top M eigenvectors that explain "most" variance (typically 85%, 90%, or 95%)



Elbow plot

- plot eigenvalues in descending order $\lambda_1 \geq \ldots \geq \lambda_D$
- choose point at which curve "bends" most (i.e. elbow)



PCA: Dimensionality Reduction

Let $V_M = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M] \in \mathbb{R}^{D \times M}$ denote the *truncated* eigenvector matrix for $M \ll D$

Reduction

Dimensionality reduction on data $\boldsymbol{x}_i$

$$\boldsymbol{e}_i^\top = \boldsymbol{x}_i^\top V_M \quad \in \mathbb{R}^M$$

More generally, projected data E

$$\begin{aligned} E &= \left[\boldsymbol{e}_1^\top, \ldots, \boldsymbol{e}_N^\top\right] \\ &= \left[\boldsymbol{x}_1^\top V_M, \ldots, \boldsymbol{x}_N^\top V_M\right] \\ &= X V_M \quad \in \mathbb{R}^{N \times M} \end{aligned}$$

Reconstruction

Recover data $\hat{\boldsymbol{x}}_i$ from $\boldsymbol{e}_i$ using $V_M^\top$

$$\hat{\boldsymbol{x}}_i^\top = \boldsymbol{e}_i^\top V_M^\top = (\boldsymbol{x}_i^\top V_M)\, V_M^\top \quad \in \mathbb{R}^D$$

More generally, reconstructed data $\hat{X}$

$$\begin{aligned} \hat{X} &= \left[\hat{\boldsymbol{x}}_1^\top, \ldots, \hat{\boldsymbol{x}}_N^\top\right] \\ &= X V_M V_M^\top \quad \in \mathbb{R}^{N \times D} \end{aligned}$$

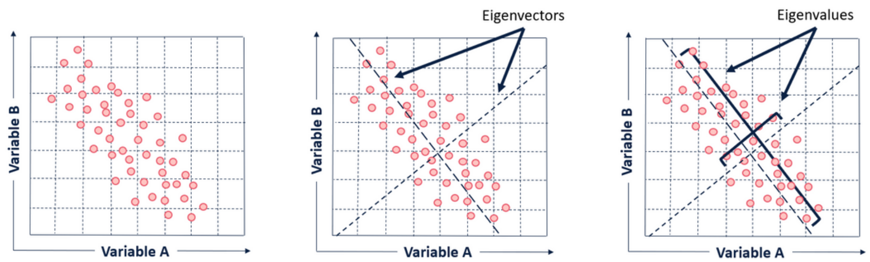
$V_M V_M^\top \in \mathbb{R}^{D \times D}$ is the data *projection* matrix

Dimensionality Reduction

PCA: Examples

PCA: Overview and Use

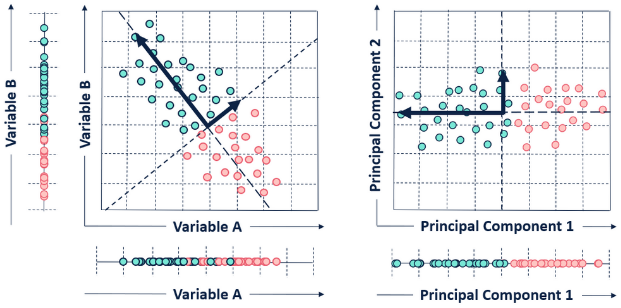
Characteristics



Figures: Sydney Firmin @ towardsdatascience.com

PCA: Overview and Use

Use: Classification



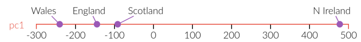
Figures: Sydney Firmin @ towardsdatascience.com

PCA Example 1: UK Food Consumption

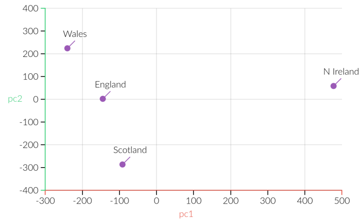
$X \in \mathbb{R}^{4\times17}$

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

Projecting to 1 component (V_1)



Projecting to 2 components (V_2)



Figures: setosa.io Data: Mark Richardson

PCA Example 2: Eigenfaces

Data $X \in \mathbb{R}^{300\times4096}$

Image $\boldsymbol{x} \in \mathbb{R}^{64\times64}$ is flattened to $\mathbb{R}^{4096}$

...

Mean face:

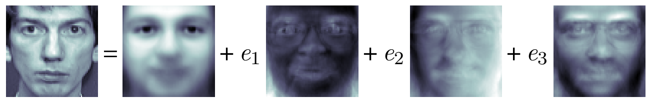
Principal Component Faces:

...

PCA Example 2: Eigenfaces

Projection

Projecting face $\boldsymbol{x}_i$ onto $\boldsymbol{e}_i = [e_{i1}, \ldots, e_{iM}]$



$= \; + e_1 \; + e_2 \; + e_3 \; + \cdots$

Reconstruction

Reconstructing face $\hat{\boldsymbol{x}}_i$ using M components

$\rightarrow$ $M = 10$ $M = 30$ $M = 50$ $M = 70$ $M = 90$

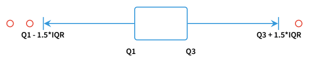
$(90 \ll 4096!)$

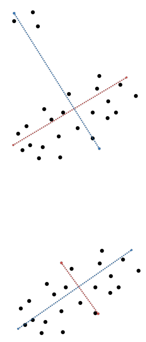
PCA: Limitations

Sensitivity

- outliers or scaling dimensions
  - changes variance along dimension
  - changes principal components
- **fix:** normalise—zero mean unit variance

  $$x' = \frac{x-\mu}{\sigma}$$
- find outliers using interquartile range (IQR)
  - 'spread' of middle 50% of values
  - median(upper quartile) - median(lower quartile)
  - define 'outlier' as values > 1.5*IQR



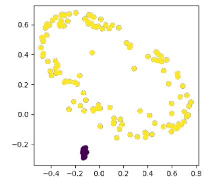


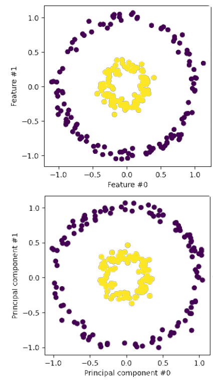
Removing outliers

PCA: Limitations

Linearity

- 1D: line; 2D: plane
- transform to handle non-linearity



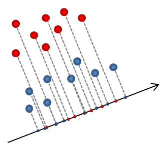


PCA: Limitations

Unsupervised

- maximises data variance along few directions
- ignorant of class labels
- could be hard to separate classes

Data



Projection on PC1

EDA: Summary

- Broad range of visualisation types
- Need to think about what information goes into a visualisation
- Actual data dimensionality $\ll$ observed dimensionality
- For high-dimensional data
  - domain knowledge / feature engineering
  - modelling assumption: independence / smoothness / symmetry etc.
  - dimensionality reduction: selection / transformation
- Principal Components Analysis (PCA)
  - choose directions that maximise variation (eigenvectors)
  - for smaller number of components M, pack information
  - examples: UK food consumption, Eigenfaces