KDDM1 - Visual Preprocessing

Roman Kern <rkern@tugraz.at>

Version 1.0.1

> **Motivation**:
The human has great pattern recognition, far exceeding machine learning approaches. This can be used to analyse and explore a given dataset.
> **Goal**:
Unterstand visual exploration tools and be able to apply these to unseen datasets.

Outline

Introduction

Why is data inspection important?

Motivation

- Before analysing the data
  - ... one needs to gain a data understanding
- ➔ **Visual inspection** can greatly help here
  - ... for small enough datasets

> For large datasets, a sampling approach (taking the subset) might be an option.

> But for a large amount of features, an automatic filtering process might be needed.

Introduction

Objectives

What to look out for?

1. Distribution of the data
   - e.g., skewed distribution
2. Outliers, missing values, artefacts, etc.
3. Dependencies (Correlation)
   - e.g., between the independent variables and the target
4. Groups (clusters)
5. Relevant features

> In this context outliers and anomalies are synonymous.

Distributions

Guess an underlying distribution

Distributions

Overview of Distributions

- In (finite, observational) data
  - The distribution is an assumption of the data generation process
- Given the distribution
  - Knows, if certain methods are appropriate
  - e.g., If two variables are bivariate normal, the Pearson's correlation describes the association completely

Distributions

Types of Analysis

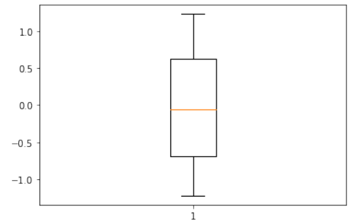
- **Univariate analysis**
  - Just on a single features
- **Multivariate analysis**
  - Combination of multiple features

> Bivariate, of 2 features are considered.
> First start with univariate analysis.

Distributions
Boxplot

- Gives an overview of the range
- ... and key statistics
  - Median (and/or mean)
  - Interquartile ranges (IQR)
  - ... and outliers
- Whiskers
  - May indicate range
  - or the 1.5 IQR

> The IQR indicates where 50% of the values are found (between Q1 = 0.25, and Q3 = 0.75).



Distributions
Histograms

- Shows the frequency of the values
- Works directly for categorical variables
- For continuous variables
  - Binning can be used

> Note: The frequency is not identical to the probability (one can use the frequency to estimate the probability).

Distributions
KDE

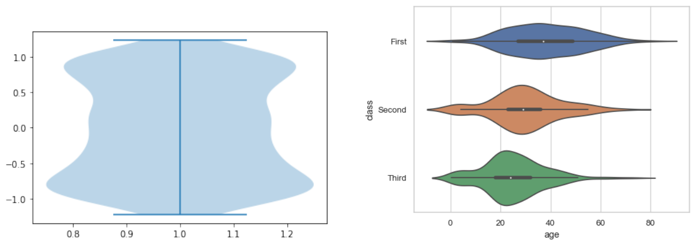
Kernel Density Estimator (KDE)

- Takes discrete data (finite)
- Produces a continuous output
  - Similar to a probability density function (PDF)
- Requires a kernel
  - Acts a smoothing of the data
  - The normal kernel is a popular choice, are requiring a parameter, controlling the bandwidth

>

Distributions
Violin plots

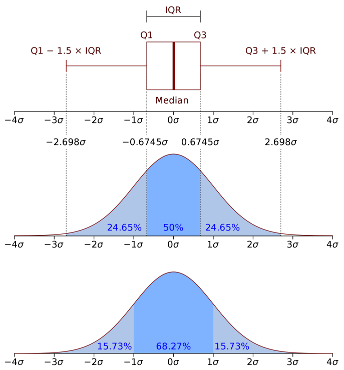
- Boxplots cannot be used to infer the distribution
- **Violin plots** combine box plots with PDF plots



Taken from https://seaborn.pydata.org/generated/seaborn.violinplot.html

> Violin plots are not very common and are generally assumed to be hard to read and interpret.

Distributions
Boxplot and PDF

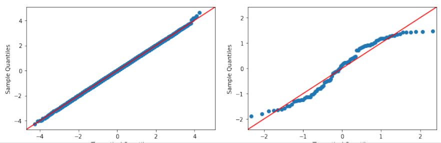


> Taken from https://en.wikipedia.org/wiki/Probability_density_function.

> For the normal distribution, the 1.5 IQR covers about 99.3% of the density.
> And 1 sigma (standard deviation) is 66.27%

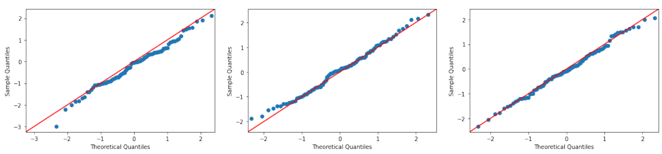
Distributions
QQ Plot

- Visual check for a given distribution, i.e., confirm or reject an assumed distribution
  - With a expected distribution on the x-axis
  - ... and the observed distribution on the y-axis
- Where the data points should be aligned on the main diagonal
- Often used for the normal distribution



> Right: Example of random 100,000 data points from a normal distribution (1,0).
> Left: Example of a 100 random points for a Beta (.9, .9) distribution - one can see that the data points deviate from the line.

Distributions
QQ Plot



> Three samples of 100 data points from a normal distribution (1,0)
> One can see that for each random sample, the plot look different, and is might not be clear if the data truly follow a normal distribution.

Dependencies

Identify relationship between variables

Introduction Dependencies

- The goal is to identify
  - Systematic dependencies between variables
    - Correlation between independent variables
    - Correlation with the dependent variable
  - Groups (or clusters) in the data
  - Partitions in the data

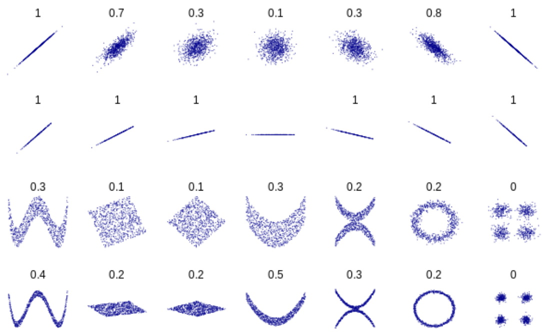
Note: Correlation can be observed/measured in data and might be a result of dependencies between variables.

> If we assume a dependency between a feature (independent variable) and the target feature (dependent variable), then the feature can be considered a candidate if the task is to predict the target feature.

Scatterplot

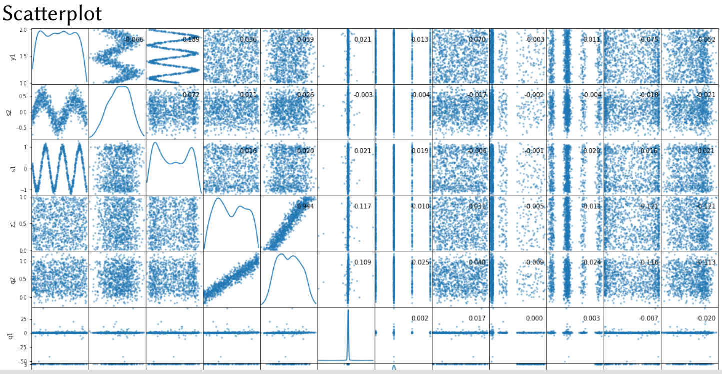
- For bivariate dependencies
  - Of continuous variables
- Each variable is assigned an axis
  - Each data point is represented by a dot (or similar)
  - (Visual) patterns could indicate a dependency
- Hard to estimate the density
  - Sometimes transparency is used
  - Sometimes noise is added (points moved a bit randomly)

Scatterplot



> Examples showing various patterns.
> The number at the top is the distance correlation.
> Top row: classical bivariate normal distributions with varying degree of covariance - in the middle the two normal distributions appear to be independent.
> 2nd row: linear relatinship (with exception of the middle, the knowing one variable determines the value of the other) > 3rd and 4th row: Includes cases, where for one value of one variable there are multiple possible values for the other.

Scatterplot



> Scatter matrix of multiple pairwise scatter plots, great tool to get an overview of a (small) dataset.
> In this example, the KDE is shown in the main diagonal.

Dependencies
Scatterplot

> Still, density might be hard to judge.

- Scatter plots are useful to detect
  - Noise in the data
    - e.g., dots all over the place
  - Outliers
    - e.g., dots in far away
    - e.g., dots in low density areas
  - Missing data
    - e.g., empty patches

Dependencies
Heatmap

> .

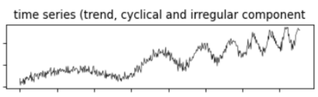
- Scatter plots work the best for continuous variables
  - ... and are not good to provide information about density
  - ... or other information, e.g., a third variable
- Heatmaps
  - Combine characteristics of scatter plots with histograms

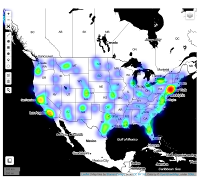
Dependencies
Higher Order Dependencies

- Finding dependencies in up to 3 dimensions works well
- But higher order dependencies are hard to visualise
- Often the data is preprocessed
  - For example, via dimensionality reduction
  - Where a high dimensional dataset is reduced to lower (2-3) dimensions
  - Each resulting dimension is then a combination of the original variables

Dependencies
Specific datasets

> Heatmap taken from: https://geospatial.streamlit.app/Heatmap

- For specific datasets
  - Dedicates visualisations can be done
  - e.g., spatial data
    - Maps
  - e.g., time series
    - 



Dependencies
Tools

- Number of advanced tools for visual inspection
  - Specialised visualisation components
    - e.g., parallel coordinates
  - Changes and selections visible in all components
    - Coordinated views

> Examples: Tableau, Power BI, AI Visualiser

> Build own apps via Streamlit, e.g., `https://tdenzl-bulian-bulian-ifeiih.streamlit.app/`

Feature Extraction

From raw data to initial features

Feature Extraction
Data vs. Information

- **Raw data** is often useless
  - i.e., cannot be directly fed to automatic methods (e.g., machine learning)
- Need techniques to (automatically) extract **information** from it
- Data: recorded (collected, crawled) facts
- Information: (novel, informative, implicit, useful, ...) patterns within the data

Feature Extraction
Description of Features

What are features?

- An individual measurable property of a phenomenon being observed
- The items, that represent knowledge suitable for Data Mining algorithms
- A piece of information that is potentially useful for prediction

> They are sometimes also called *attributes* (Machine Learning) or *variables* (statistics).

Feature Extraction
Example of Features

- **Images** → colours, textures, contours, gradients, ...
- **Signals** → frequency, phase, samples, peaks, spectrum, ...
- **Time series** → ticks, trends, self-similarities, seasonality, ...
- **Biomed** → DNA sequence, response to intervention, ...
- **Text** → words, POS tags, grammatical dependencies, ...
- **Qualitative** → questionnaire, subjective rating, ...

> Features encode these properties in a way suitable for a chosen algorithm

Feature Extraction
Types of Features [Stevens 1946]

- **Numeric** (for quantitative data)
  - Continuous, e.g., height, time, ...
    - Interval, if intervals are equally split, e.g., date
    - Ratio, for intervals with a defined zero point, e.g., temperature, age
  - Discrete, e.g., counts
- **Categorical** (often for qualitative data)
  - Nominal
    - Two or more categories
    - e.g., gender, colour
  - Ordinal
    - There is an ordering within the values, e.g., ranking

> Binary features are quite common - what are they?
> Continuous features are often transformed to categorical variables.
> What is a Likert scale?

Feature Extraction
Categories of Features

- **Contextual features**
  - e.g., position information, browsing history
- **Structural features**
  - e.g., structural markups, DOM elements
- **Linguistic features**
  - e.g., POS tags, noun phrases
- ...

Feature Extraction
Example for feature extraction

- Handwriting recognition
  - ... popular introductory example in textbooks about machine learning, e.g. Machine Learning in Action [Harrington 2012]

Feature Extraction
Example for feature extraction

- **Input**: A collection of scanned in handwritten digits
- **Preprocessing**:
  - Remove noise
  - Adapt saturation changes, due to differences in pressure when writing
  - Normalise to the same size
  - Center the images, e.g., center of mass or bounding box
- **Feature extraction**:
  - Pixels as binary features

> Depending on the algorithm to center the images, some algorithm improve in performance, e.g. SVM according to the authors of the MNIST data set

Preprocessing

Practical Considerations

Preprocessing
Task of importing data

- Data concerns, e.g., CSV files
  - Encoding
  - Separator and escape character
- Assign feature types
  - Parse the raw data
  - e.g., comma or dot for comma separator
  - e.g., consistent handling of umlauts

> See also https://statisquo.de/2018/08/27/csv-dateien-in-python-importieren-mit-pandas/

Preprocessing
Task of preprocessing

- Identify quality issues in the data
  - Unnecessary data
  - Missing values
  - Noise
  - Incorrect data
  - Inconsistent data
  - Formatting issues
  - Duplicate information
  - Disguised data

> Garbage in, garbage out

> All quality impairments might negatively affect the data mining task.

Preprocessing

Task of preprocessing

- Split the dataset into coherent parts
  - e.g., one dataset for each user group
- Remove not needed data
  - Rows (instances) or columns (features)
- Transform input data
  - Achieve consistency
  - Transform distribution
- Add additional data from external sources

> The goal it to help the algorithm pick up the relevant information in a way suitable.
> For example, some algorithms prefer data to be centred, or normalised.

> Further considerations: privacy (personal data), fairness (sensitive data)
> Bias in, bias out

Thank You!

... for your attention