Data Preprocessing

Alipio Jorge

October 2020

Data Preprocessing

- Why transforming data?
  - What can be wrong with data?
  - Real World Data (RWD) is **never** perfect
- Data Quality
  - accuracy, completeness, consistency
  - quality depends on what you want the data for

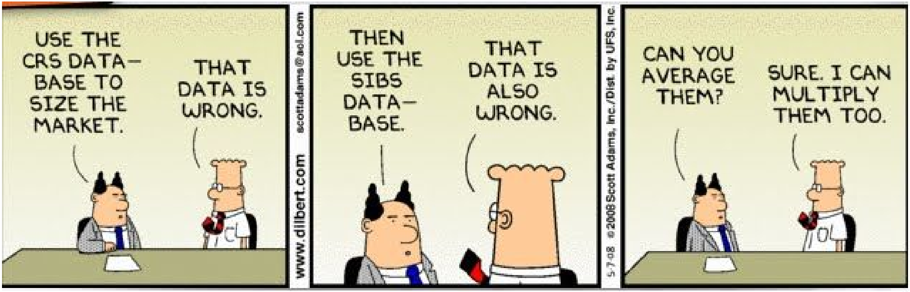
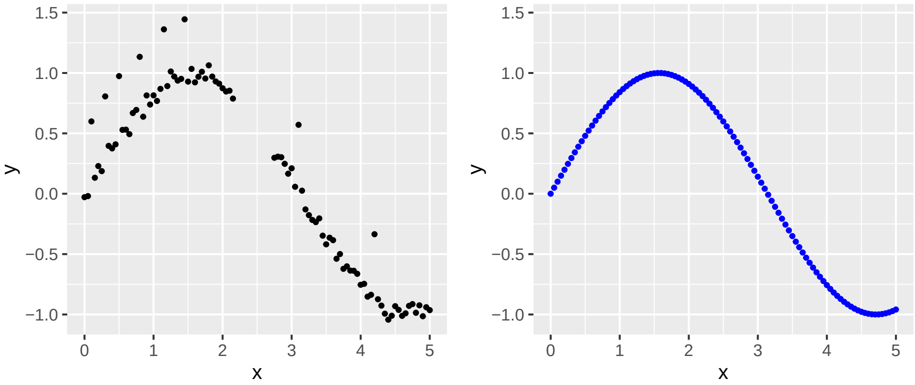


Figure 1: A Dilbert strip

Major tasks

- Data cleaning
  - filling missing values
  - smoothing noise
  - removing outliers
  - resolving inconsistencies



Data integration

- You want to predict your customers preferences
    - customer data
    - products data
    - sales data
    - reviews
    - images from the products
    - posts on facebook
    - weather data

Data integration

- Tasks
  - matching fields: `customer_id` vs `client_id`
  - matching values: `Manuel Joaquim Silva` vs. `manuel j. silva`
  - avoid redundancies (product name may be in sales data and in products data)

Data reduction

- So much data
  - may not improve results
  - may be too much for the resources
  - use **what you need** and **what you can cope with**
- **Dimensionality** reduction
  - less variables (columns)
- **Numerosity** reduction
  - less cases (rows)

Data transformation

- Different **scales** can be a problem
  - normalization, standardization
- e.g., workers described with *age* and *salary*,
  - how to measure a distance between two workers?

$$d(< 21, 27000 >, < 40, 30000 >) =??$$

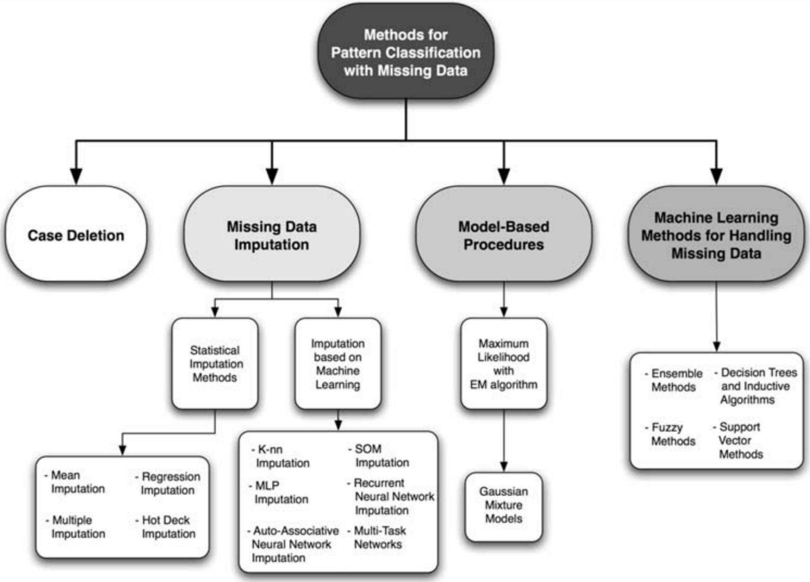
Data transformation

- Shaping data to be used for certain methods
  - Discretization (youth, adult, senior)
  - Concept Hierarchy Generation
  - Binarization
  - Type conversion in general

Missing Values

- **Missing Values** is perhaps the most common problem in RWD
- What to do?
  - Nothing
  - Ignore the attribute
  - Ignore the tuple
  - Impute values (fill in)
    - (a lot to be said)

Missing Values



Missing Values

- **if** the method is robust to missing data and the amount of missing data is not too high
  - do Nothing
- **else**
  - **if** only a few cases have problems
    - ignore the cases
  - **if** the problem is on discardable attributes
    - ignore the attribute
  - **if** missing values persist
    - try value imputation

Always be **very careful** when you transform the data set

Data Imputation

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
NA	36	M	manager	59000
Rui	27	M	NA	27000
Sofia	NA	F	manager	58000
Ana	48	F	CEO	77500

- Do Nothing
  - the Name column
  - Gender?

Data Imputation

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
NA	36	M	manager	59000
Rui	27	M	NA	27000
Sofia	NA	F	manager	58000
Ana	48	F	CEO	77500

- Age?
- Use a global constant
    - **pro**: easy
    - **cons**: data bias, may affect inference

Data Imputation

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
NA	36	M	manager	59000
Rui	27	M	NA	27000
Sofia	NA	F	manager	58000
Ana	48	F	CEO	77500

- Position, Age
- Use a measure of central tendency
  - *mean*, *median*, *mode*
  - **pros**: easy, gets the most likely value
  - **cons**: distorts the distribution
    - e.g.: average keeps average but affects variance

Data Imputation

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
NA	36	M	manager	59000
Rui	27	M	NA	27000
Sofia	NA	F	manager	58000
Ana	48	F	CEO	77500

- Age, Position
- Use a measure of central tendency taken from **same group** or **same class**
  - **pros**: varied values imputed
  - **cons**: may still be too insensitive

Data Imputation

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
NA	36	M	manager	59000
Rui	27	M	NA	27000
Sofia	NA	F	manager	58000
Ana	48	F	CEO	77500

- Age, Position
- Use a measure of central tendency using the **most likely** value for that case
    - e.g.: from *neighbours*, or using **linear regression**
    - **pros**: varied values imputed,
    - **cons**: needs processing, depends on distance measure and parameters

Noisy Data

- **Noise**
  - Random error or variance in a measured variable
- **Smoothing**
  - assume a value is always similar to neighbors
  - you **replace** values (stronger than imputation)
- **Outliers**
  - can be smoothed away if we assume they are noise
- Be very **careful**
  - do not smooth **legitimate** data (unless it helps)

Smoothing

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
José	36	M	manager	59000
Rui	41	M	manager	57000
Sofia	105	F	manager	58000
Ana	28	F	assistant	28500

- Age=105 is an **outlier**
  - Binning: replace each value in group by the group mean
  - average of Age for each Position

Smoothing

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
José	36	M	manager	59000
Rui	41	M	manager	57000
Sofia	105	F	manager	58000
Ana	28	F	assistant	28500

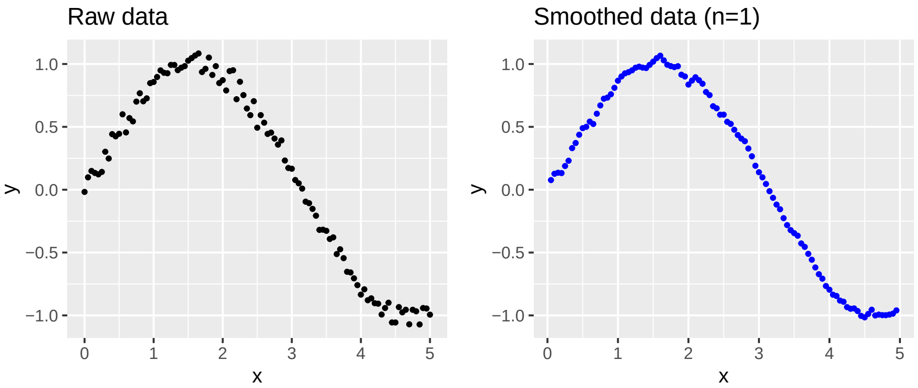
- Regression
  - try to predict ‘Age’ from the other attributes
  - replace the original values with the predicted ones
  - may lose **too much** information

Smoothing

Name	Age	Gender	Position	Salary
Manuel	25	M	assistant	23000
José	36	M	manager	59000
Rui	41	M	manager	57000
Sofia	105	F	manager	58000
Ana	28	F	assistant	28500

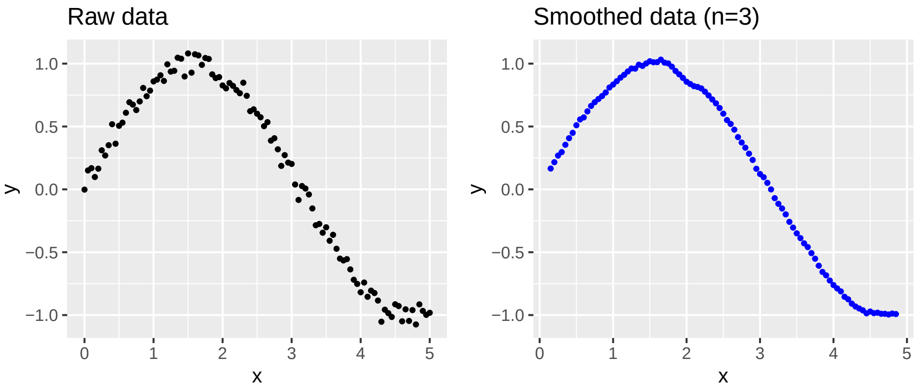
- Age=105 is an **outlier**
  - can be detected with clustering or using the IQR rule
  - can be replaced by the mean age of *manager*
  - i.e., detect outliers and replace them by a sensible mean

Smoothing



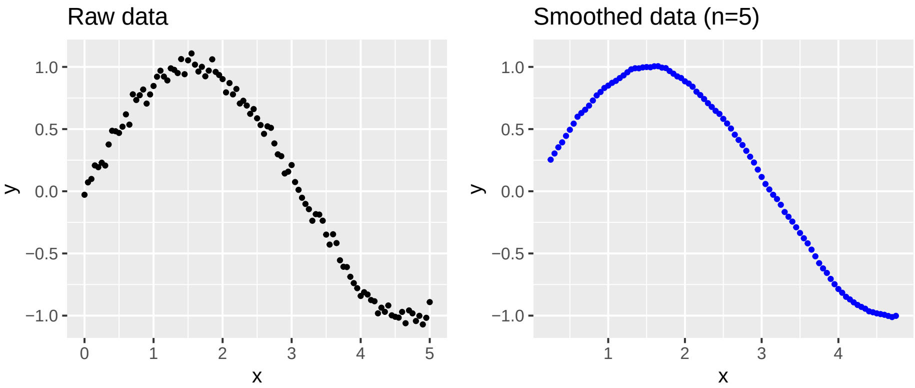
- Smoothing with **moving average**
  - replace each value y_i with $average_{j-n \geq j \leq j-n} y_j$
  - the larger the n, the smoother the line
  - Above $n = 1$

Smoothing



- Smoothing with **moving average**
  - replace each value y_i with $average_{j-n \geq j \leq j-n} y_j$
  - the larger the n, the smoother the line
  - Above $n = 3$

Smoothing



- Smoothing with **moving average**
  - replace each value y_i with $average_{j-n \geq j \leq j-n} y_j$
  - the larger the n, the smoother the line
  - Above $n = 5$

Data integration

- The same object can have different representations
  - customer in social network an in sales data
  - two companies merging
  - **entity identification problem**
- There may be **redundant variables**
  - **detect** redundancy
  - **remove** redundant variables

Redundancy analysis

- We can measure the “similarity” of two variables
  - Nominal: χ^2 statistical test
    - if the null hypothesis is accepted, one of the variables is redundant
    - if rejected the variables are independent

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

- o is the observed frequencies, e are the expected - high values of the χ_2 statistic mean **independence**

$$e_{ij} = \frac{\#(A = a_i) \times \#(B = b_j)}{n}$$

Redundancy analysis

- We can measure the “similarity” of two variables
  - Numerical: Pearson **correlation**

Other operations in data integration

- Eliminate **duplicate tuples**
  - the same customer appears in the DB (from two different sources)
- Detect **conflicting values**
  - different representations, units, encondings
  - e.g. sales in Euros and in Dollars
  - e.g. sales per day and sales per week

Data reduction: Dimensionality reduction

- reduce the number of variables
- **Principal Components Analysis** (PCA)
  - finds new variables that
    - are much **fewer** than original ones
    - each is a **linear combination** of the original ones
    - explain *most* but not all of what is observed
  - **cons**: new variables may not be interpretable

Data Reduction: Dimensionality reduction

- reduce the number of variables
- **Feature selection**
  - e.g. we want to predict if a customer is leaving a mobile operator (churn)
  - not all features are relevant for **this problem**
  - a **good feature** is correlated with the target variable
- **Techniques**
  - Eliminate features with low correlation
    - does not consider joint effects of variables
  - **Stepwise forward selection**
    - start with zero features, add the best feature, keep adding
    - stop when improvement stops
  - **Stepwise backward elimination**
    - start with all the features, ...
  - (among others)

Data Reduction: Numerosity reduction

- **Sampling**
  - very important technique
- Types of sampling
  - **random without replacement**
  - **random with replacement**
  - **stratified**
    - data is divided in groups (e.g. by gender and age)
    - random sampling is done in the groups
    - warrants representaion of groups
    - important when groups have different sizes (e.g. do you often go to the stadium?)

Sampling

- Easy to **control**
  - we can reduce the data size almost arbitrarily
- We can **determine the minimum size** of the sample
  - under certain conditions
- We must be careful with the sampling **methodology**
  - avoid bias
  - e.g. asking about smoking habits to people at the door of buildings

Data Transformation

- **Normalization**
  - Some methods are **sensitive to the range** of variables
    - distance/similarity measures
    - neural networks
  - **Solution**: scale all variables to the **same range**
  - Typical *normalized* ranges
    - $[0, 1]$ and $[-1, 1]$

Data Transformation

- **Min-max normalization**

$$x'_i = \frac{x_i - min_x}{max_x - min_x}$$

- May have *out of bound* future values
  - in that case, clip if needed

Data Transformation

- **Standardization**
  - is a kind of normalization
    - but without a closed boundary

$$x_i' = \frac{x_i - mean_x}{std_x}$$

Data Transformation

- **Discretization**
  - transform a numerical variable into a categorical one
    - e.g.: salary $\rightarrow \{low, medium, high\}$
  - necessary for some methods (e.g. association rules)
  - may improve interpretability
- Techniques
  - domain expert
  - **binning**: divide data in bins of equal width or equal frequency

$$Age =< 20, 21, 21, 24, 25, 25, 27, 27, 28, 29, 35, 35 >$$

- Equal width

$$Junior = [20, 25],\ Advanced = [26, 30],\ Senior = [31, 35]$$

- Equal depth (frequency)

$$Junior = [20, 24],\ Advanced = [25, 27],\ Senior = [28, 35]$$

Data Transformation

- **Binarization**
  - transform a multi valued categorical variable into binary variables
  - usually increase the number of veriables
  - important (necessary) for some techniques
- Use a categorical variable in linear regression
  - $Age = Junior, Advanced, Senior$
  - solution: create three (dummy) binary variables:
    - *Junior*, *Advanced*, *Senior*

References

- Han, Kamber & Pei, Data Mining Concepts and Techniques, Morgan Kaufman.
- García-Laencina, P.J., Sancho-Gómez, J. & Figueiras-Vidal, A.R. Pattern classification with missing data: a review. Neural Comput & Applic 19, 263–282 (2010).
- Pavel Horbonos, A brief guide to data imputation with Python and R: Make the data clean, Towards Data Science (2020)