Data Preparation for Data Mining

Lesson 5

Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Handling Non-numerical Variables

- Remapping
- State space

Handling Non-numerical Variables

Remapping

- One-of-*n*
- *m*-of-*n*
- Ordering
- Ill-formed problems (one-to-many patterns)
- Circular discontinuity

Handling Non-numerical Variables

State Space

- Basic properties
- Locations and points
- Density
- Topography
- Phase space
- Mapping alphas

Remapping overview

- Non-numeric (alpha) variables are remapped to numerical values
  - numerical to non-numerical remapping is of course also possible
- The form of remapping depends on the modeling tool used
  - Neural networks vs. decision trees

Usage

- Remapping can be useful if:
  - a remapped pseudo-variable will have a high information density
  - dimensionality is only slightly increased
  - some form of reasoning can be given for remapping
  - model requires that no ordering of alphas is used

One-of-*n* Remapping

- One binary pseudo-variable per alpha label
- Only a single variable "on" for each sample

One-of-*n* Remapping

- **Advantages:**
  - mean of each pseudo-variable is directly proportional to the number of corresponding labels in the sample
    - ➔ useful in prediction
- **Disadvantages:**
  - big increase in dimensionality
  - low pseudo-variable density
  - in prediction, many pseudo-variables will be on for a single output

One-of-*n* Remapping

- **Example:**
  - one variable for each European country

```
            FIN GER ITA POL ...
Finland       1
Germany             1
Italy                     1
Poland                          1
...
```

m-of-*n* Remapping

- Pseudo variables created from alpha label characteristics
- Several pseudo-variables "on" per sample

m-of-*n* Example

- **Example:**
  - countries are divided according to geographic location, population, GNP, etc.

	North	Center	South	East	Area	GNP	...
Finland	1			1		1	
Germany		1			1	1	
Italy			1		1	1	
Poland		1		1	1		
...							

m-of-*n* Remapping

- **Advantages:**
  - dimensionality increased less than with one-to-*n*
    (if less pseudo-variables than labels)
  - useful new information possibly added
- **Disadvantages:**
  - highly dependent on domain knowledge

Ordering

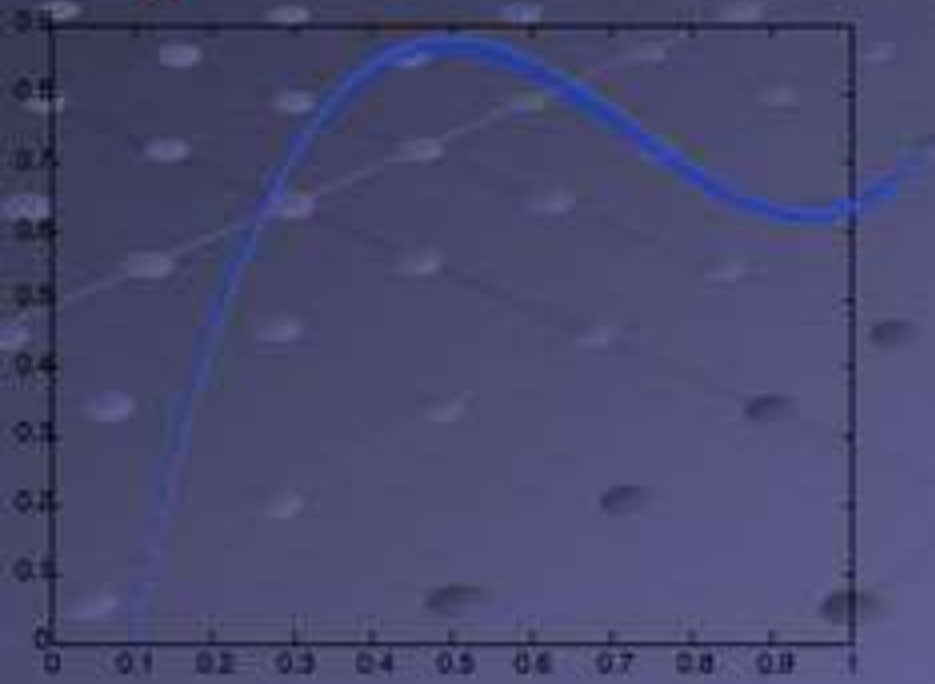
- If the alpha labels to be remapped contain an implicit ordering, it should be preserved
  - Example: labels for lengths of time, sizes etc.
- Remapping can be used to ascertain that there is **no** implication of ordering

Ill-formed Problems

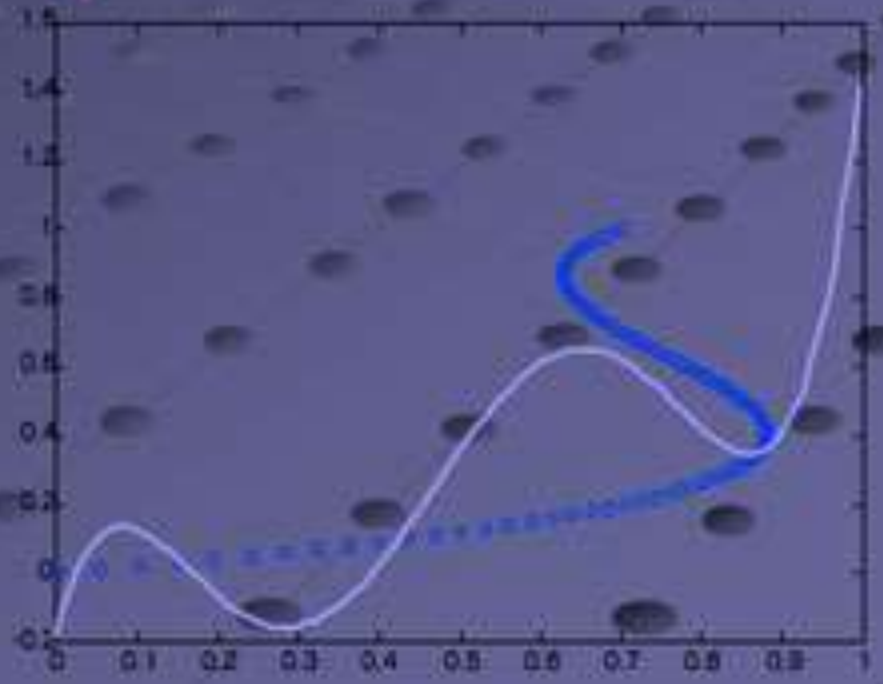
- The one-to-many pattern: several input values indicate the same output
- Modeling tools that try to find a function fitting the data fail

Ill-formed Problems

- Profit curve:
  x = price, *y* = profit



- Same profit curve, axes reversed:
- x = profit, y = price



Remapping Ill-formed Problems

- Areas of multi-valued output hard to detect, easiest in data survey
- If one-to-many situation is known, easiest to correct by data preparation
  - Additional information (more dimensions) must be added to distinguish between the situations of identical output

Remapping Ill-formed Problems

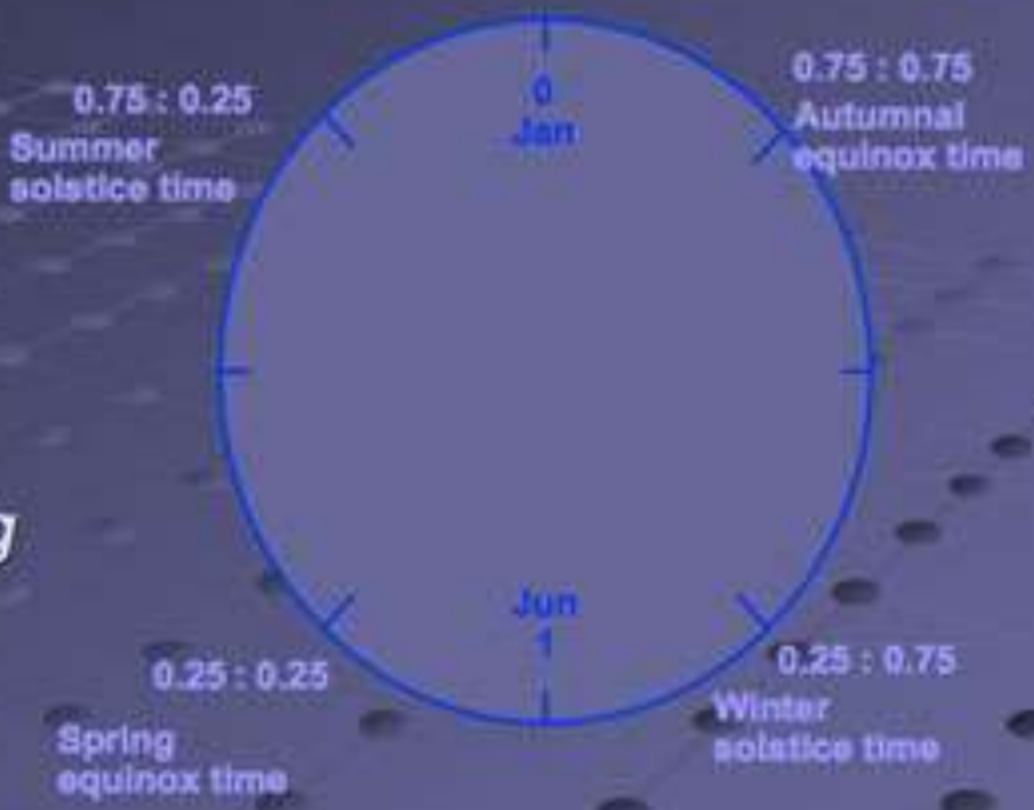
- Other ways to correct one-to-may problem mentioned:
  - "Reverse the axes" - reflect the data in an appropriate state space
  - Use a local distortion to "untwist"
    - Risky
  - Use modeling that can deal with one-to-many

Remapping Circular Discontinuity

- Annual cycles: months, days of month, weeks …
- Also other cycles: weeks to a chosen annual event
- Discontinuity in labeling (from 12 to 1, 31 to 1, 52 to 1), prevents most modeling tools from finding cyclical information

Remapping Circular Discontinuity

- Better labeling: weeks start from 0 in January, rise to 1 in June, then decrease to 0.
- The week number indicator is called *lead variable*. In addition a *lag* is used to indicate the lead variable value quarter of a year ago.
- Two variables are needed to be able to unambiguously define the time (two dimensions - two coordinates)



State Space Overview

- N-dimensional space, variables of the data set as dimensions
- Variable ranges limited, often normalized to *unit state space*
  - modeling tools cannot cope with monotonicity

State Space Overview

- Each point represents a particular state of the system
- Distances between points calculated with Pythagorean theorem
  - $d^2 = \Sigma (d_1^2 + d_2^2 + \ldots + d_n^2)$
  - → distance increases as number of dimensions (*n*) increase
  - measured distance can be normalized in unit state space, since $d_{max}^2 = n$

State Space Overview

- Points close together are called neighbors
  - Neighboring states are more likely to share common features
  - Nature of neighborhoods may change from place to place

Locations, points and density

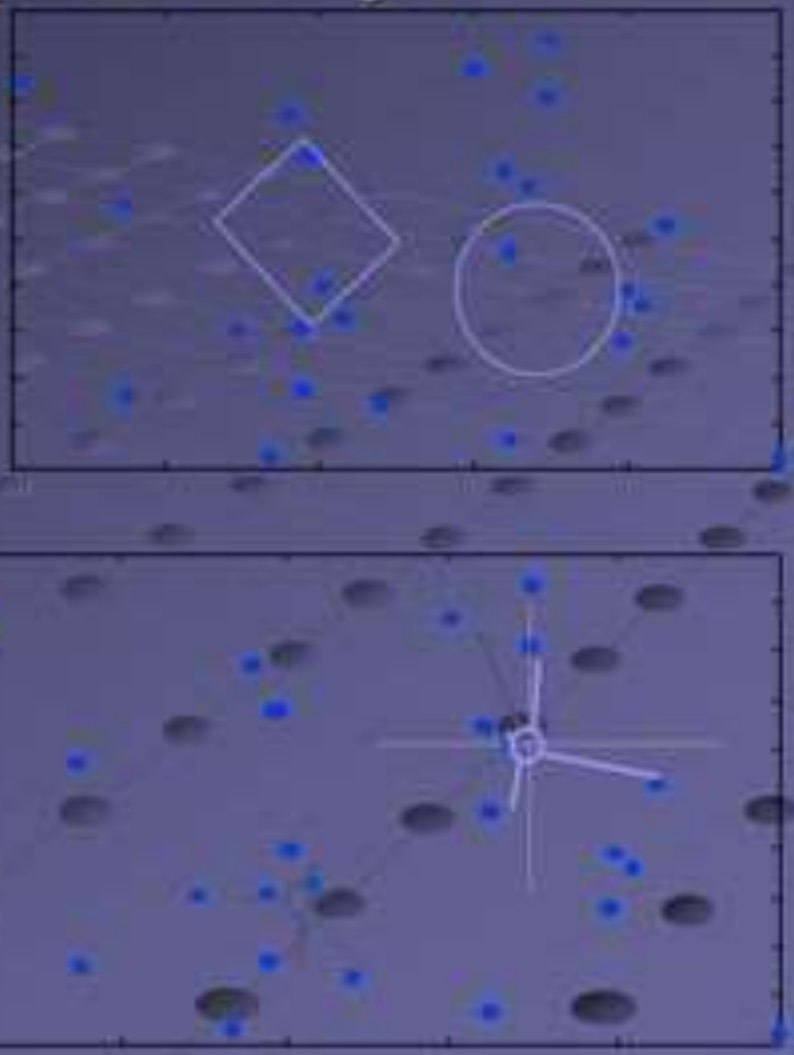
- Location or position indicates specific place in state space
- Point or data point indicates a location which represents a measured system state
- Density measured as number of points in specific volume

Locations, points and density

- State space volume is fixed, but number of points depends on the size of the data set
- ➔ *Relative density* most useful to examine
  - Relative density = specific area density / mean density
  - Unaffected by changing data set size
  - Not usually normalized

Estimating density

- By number of points in an area (volume)
  - depends on shape of area
  - rotation and translation affect result
- By distance to *n* nearest neighbors (NN)
  - value of *n*?
  - closest neighbors may be "biased" to one dir.
  - better estimate when area divided and NN found in each division



State space topography

- Values can be smoothed between the points to get a continuous density gradient
- Density values can be represented as height on the map (high density down, low density up)

State space topography

- Contours of constant "elevation" can be drawn
- Contours point out natural clusters in the data - the valleys of high density

- Data points can be thought to form geometric objects
  - higher-dimensional objects can be projected ("cast shadows") to a lower-dimensional space

Phase space and mapping alphas

- Phase space is used to represent features of objects or systems *other* than their state
- Alpha labels are positioned into phase space each with specific distance and direction from neighboring labels

Phase space and mapping alphas

- Once the appropriate places for the labels (in phase space) are known, the appropriate label values (in state space) can be found
- The alpha labels are associated with some particular area on the state space map
- There is no absolute value associated with each label, but the order and distance of labels is preserved in the numeration

Example: Montreal Canadians

- **Example 1**
  - two-dimensional state space consisting of player height and weight
  - arbitrary labels are assigned for player weights
  - the labels are given values according to the normalized height of the player
  - the correlation of original and recovered weights is quite good (0.85), *which indicates that taller hockey players tend also to weigh more than short ones*

Example 2

- **Example 2**
  - three-dimensional state space consisting of player height, weight and position
  - player positions (defense, forward, goal, reserve) are inherently labeled
  - the labels are given (two-dimensional) values by calculating the mean height and weight of all players represented by that label
  - the labels fall nearly on a straight line in (height-weight) state space, so a single numerical label (which represents the normalized position on the line) is sufficient

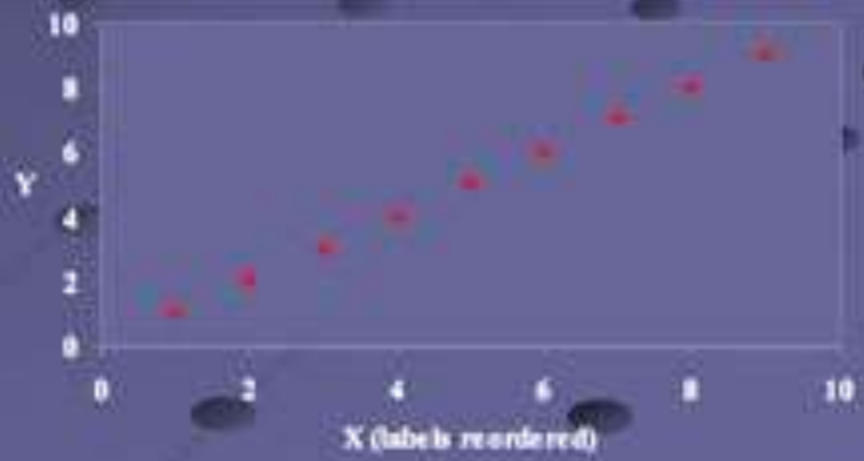
Handling non-numerical variables

- Ordering alpha labels:
  - joint distribution tables
  - when no numerical variables are available
- Dimension reduction
  - multidimensional scaling (MDS)

Ordering alpha labels

- Trivial ordering always possible (e.g., alphabetical)
- Better idea: Order labels of different variables so that correlations show

- How?

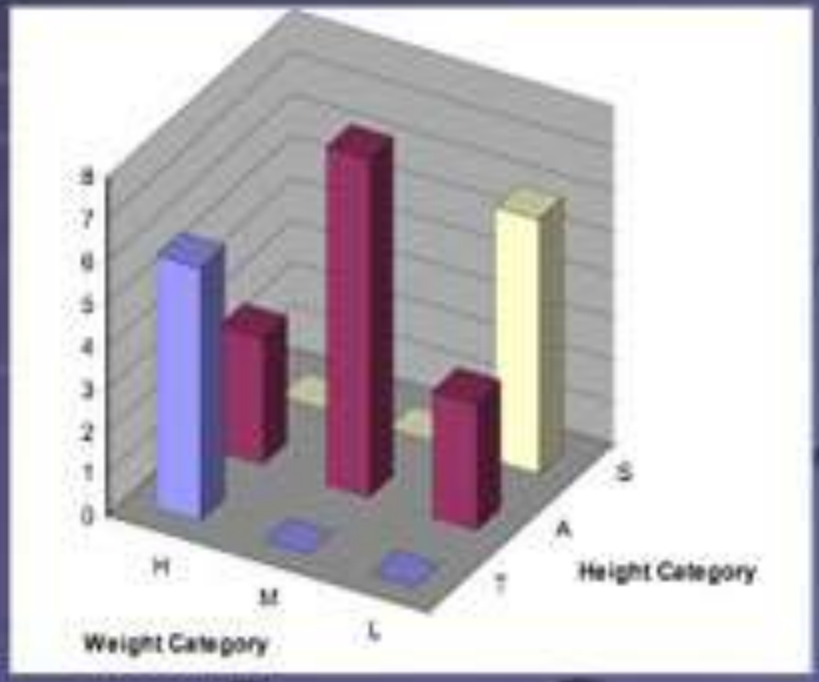


Ordering alpha labels

- Two-way tables:

Joint frequency listing of two alpha values

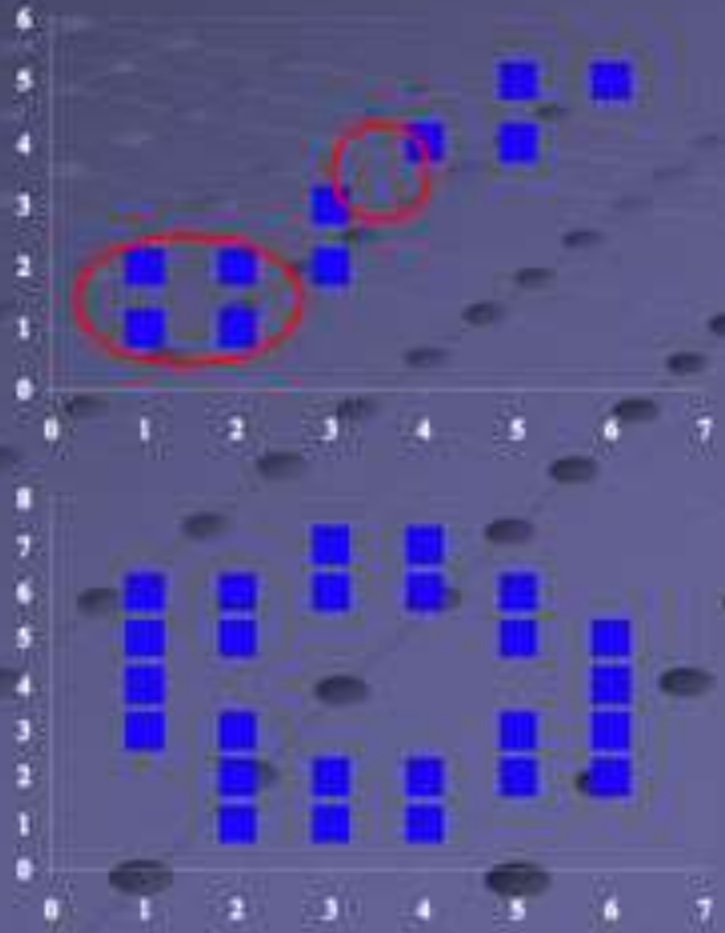
	H	M	L
T	6	0	0
A	3	8	3
S	0	0	6



- Tall, Average, Short vs. Heavy, Medium, Light
- nine entries – one for each combination of labels
- The graph illustrates the distribution graphically

Ordering alpha labels

- Two-way tables
- Jigsaw puzzle
- Problems
  - too much overlap
  - too little overlap
  - nonlinear relationships



Ordering alpha labels

- Two-way tables
- Jigsaw puzzle
- Problems
- Implementation
  - straightforward
  - always possible to numerate some of the variables (?)
  - continue with mixed alpha-numeric methods
- More vars (dimensions): same procedure

Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Normalization

All variables are assumed to have a numerical representation.

Two topics:

- Normalizing the range of a variable
- Normalizing the distribution of a variable (redistribution)

Part I: Normalizing variables

- Variable normalization requires taking values that span a specific range and representing them in another range.
- The standard method is to normalize variables to [0,1].
- This may introduce various distortions or biases into the data.
- Therefore, the properties and possible weaknesses of the used method must be understood.
- Depending on the modeling tool, normalizing variable ranges can be beneficial or sometimes even required.

Linear scaling transform

- First task in normalizing is to determine the minimum and maximum values of variables.
- Then, the simplest method to normalize values is the linear scaling transform:

$$y = (x - \min\{x_1, x_N\}) / (\max\{x_1, x_N\} - \min\{x_1, x_N\})$$

- Introduces no distortion to the variable distribution.
- Has a one-to-one relationship between the original and normalized values.

Out-of-range values

- In data preparation, the data used is only a sample of the population.
- Therefore, it is not certain that the actual minimum and maximum values of the variable have been discovered when normalizing the ranges
- If some values that turn up later in the mining process are outside of the limits discovered in the sample, they are called *out-of-range* values.

Dealing with out-of range values

- After range normalization, all variables should be in the range of [0,1].
- Out-of-range values, however, have values like -0.2 or 1.1 which can cause unwanted behavior.

Solution 1. Ignore that the range has been exceeded.

- Most modeling tools have (at least) some capacity to handle numbers outside the normalized range.
- Does this affect the quality of the model?

Dealing with out-of range values

Solution 2. Ignore the out-of-range instances.

- Used in many commercial modeling tools.
- One problem is that reducing the number of instances reduces the confidence that the sample represents the population.
- Another, and potentially more severe problem is that this approach introduces bias. Out-of-range values occur with a certain pattern and ignoring these instances removes samples according to a pattern introducing distortion to the sample.

Dealing with out-of range values

Solution 3. Clip the out-of-range values.

- If the value is greater than 1, assign 1 to it. If less than 0, assign 0.
- This approach assumes that out-of-range values are somehow equivalent with range limit values.
- Therefore, the information content on the limits is distorted by projecting multiple values into a single value.
- Has the same problem with bias as Solution 2.

Making room for out-of-range values

- The linear scaling transform provides an undistorted normalization but suffers from out-of-range values.
- Therefore, we should modify it to somehow include also values that are out of range.
- Most of the population is inside the range so for these values the normalization should be linear.
- The solution is to reserve some part of the range for the out-of-range values.
- Reserved amount of space depends on the confidence level of the sample:
  - 98% confidence → linear part is [0.01, 0.99]

Squashing the out-of-range values

- Now the problem is to fit the out-of-range values into the space left for them.
- The greater the difference between a value and the range limit, the less likely any such value is found.
- Therefore, the transformation should be such that as the distance to the range grows, the smaller the increase towards one or decrease towards zero.
- One possibility is to use functions of the form $y = 1/x$ and attach them to the ends of the linear part.

Softmax scaling

- Carrying out the normalization in pieces is tedious so one function with equal properties would be useful.
- This functionality is achieved with *softmax scaling*.
- The extent of the linear part can be controlled by one parameter.
- The space assigned for out-of-range values can be controlled by the level of uncertainty in the sample.
- Nonidentical values have always different normalized values.

The logistic function

- Softmax scaling is based on the logistic function:

$$y = 1 / (1 + e^{-x})$$

where y is the normalized value and x is the original value.

- The logistic function transforms the original range of [-∞, ∞] to [0,1] and also has a linear part on the transform.
- Due to finite wordlength in computers, very large positive and negative numbers are not mapped to unique normalized values.

Modifying the linear part of the logistic function range

- The values of the variables must be modified before using the logistic function in order to get a desired response.
- This is achieved by using the following transform

$$x' = (x - \underline{x})/(\lambda(\sigma/2\pi))$$

where $\underline{x}$ is the mean of x , σ is the standard deviation, and λ is the size of the desired linear response.

- The linear part of the curve is described in terms of how many normally distributed standard deviations are to have a linear response.

Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Redistributing variable values

- (Linear) range normalization does not alter the distribution of the variables.
- The existing distribution may also cause problems or difficulties for the modeling tools.
  - Outlying values
  - Outlying clusters
- Many modeling tools assume that the distributions are normal (or uniform).
- Varying densities in distribution may cause difficulties.

Adjusting distributions

- Easiest way adjust distributions is to “spread” high-density areas until the mean density is reached.
  - Results in uniform distribution
  - Can only be fully performed if none of the instance values is duplicated
- Every point in the distribution is displaced in a particular direction and distance.
- The required movement for different points can be illustrated in a displacement graph.

Modified distributions

- What changes if a distribution of a variable is adjusted?
  - Median values move closer to point 0.5
  - Quartile ranges locate closer to their appropriate locations in a uniform distribution
  - "Skewness" decreases
  - May cause distortions e.g. with monotonic variables

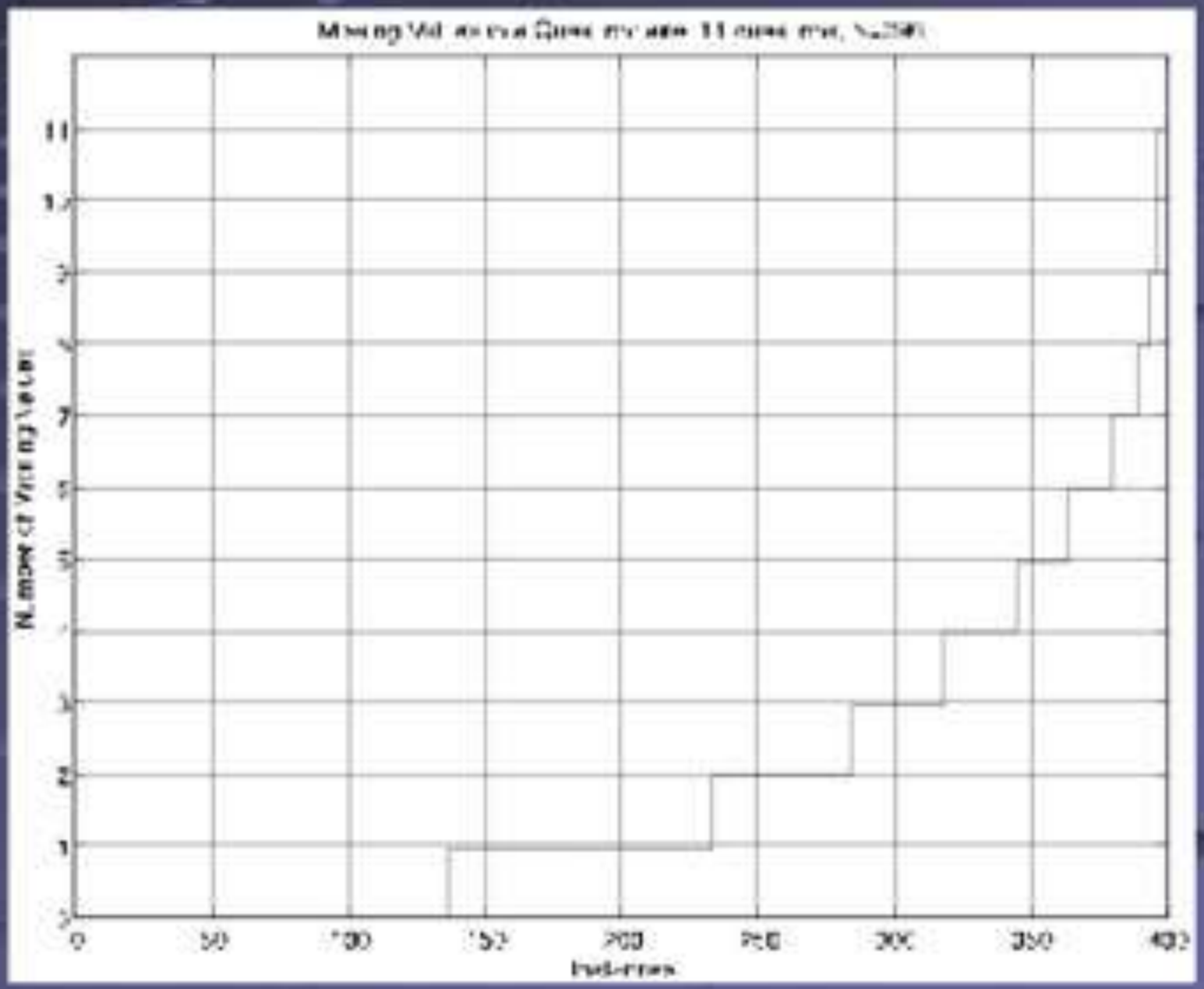
Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Dealing with Missing Values

- Motivation
- Objectives
- Meaning for the conclusions
- Origin of missing values (MV)
- Detection of missing values
- Replacing missing values
- Examples

…missing values?



Motivation

- There are always MVs in a real data set
- MVs may have an impact on modeling, in fact, they can destroy it!
- MVs contain also information!!!
- Hint for the modeler: Avoid-Detect-Replace-Understand

“Definitions”

- Missing value - not captured in the data set: errors in feeding, transmission, ...
- Empty value - no value in the population
- Outlier, out-of-range value

Objectives

- Controlled and understood by the modeler
- “Least harm”, no “new” information into a data set
- statistical estimation of MVs not the primary issue, but DM
- speed and simplicity
- PIE-I/O – training + testing + execution

Origin and Detection

- Missing data process
- Degree of randomness
  - nonrandom
  - missing at random
  - missing completely at random
- Detecting missing value patterns
  - number of MVs in each variable/case
  - compare MVP to complete sets

Replacing missing values

- Randomness of MVs?
- Methods
  - Use the complete data
  - Delete variable(s)/case(s)
  - Imputation methods...
  - Model based (e.g. Bayes)
  - Use robust models

Lesson 5 Overview

- Handling non-numeric variables
- Normalizing variables
- Redistributing variables
- Replacing missing values
- Replacing empty values

Imputation methods

- Process of estimating MVs based on valid values of other variables / cases
- Techniques:
  - distribution characteristics from all available valid values
  - replacing: case, mean substitution, cold deck, regression imputation

Examples

- Polls, Questionnaires
  - planning more than essential
  - human factors!
  - small amounts of data
- Data from steel plant
  - information system
  - errors, default values
  - lots of data
  - select appropriate subset and clean it

Summary

- Replacing missing values very important
- As important to capture the patterns of the values that are missing, as it is to carefully replace them
- Using inappropriate replacement values easily disturbs existing patterns and introduces spurious patterns, called bias or noise