

Wstępna obróbka danych

Nguyen Hung Son

preprocessing

1

Dzisiejsze tematy

- Wstęp
- Metody czyszczenia danych
- Transformacja i integracja (scalanie) danych
- Redukcja danych
- Dyskretyzacja i utworzenie hierarchii pojęć
- Podsumowanie

preprocessing 2

Dlaczego wstępna obróbka danych?

- Dane w realnym świecie są „brudne”
 - Są **niepełne**: Brak niektórych wartości w kolumnie, brak pewnych ważnych atrybutów, zawierają zagregowane informacje, a brakuje bezpośredniego dostępu do danych.
 - **Szumy**: zawierają błędy lub wyjątki
 - **Sprzeczne**: containing discrepancies in codes or names
- Nie ma wartościowych wyników analizy bez danych dobrej jakości!
 - Trafna decyzja biznesowa musi być oparta o dobrej jakości dane
 - Hurtownie danych integrują niesprzeczne i dobrej jakości danych

preprocessing 3

Zrozumienie danych i ich powiązania

- Jakie dane są dostępne dla zadania?
- Czy dane są odpowiednie?
- Czy istnieją dodatkowe odpowiednie dane?
- Jakie dane historyczne są dostępne?
- Kto jest ekspertem tych danych?

preprocessing 4

Data Understanding: Quantity

- Number of instances (records, objects)
 - *Rule of thumb: 5,000 or more desired*
 - if less, results are less reliable; use special methods (boosting, ...)
- Number of attributes (fields)
 - *Rule of thumb: for each attribute, 10 or more instances*
 - If more fields, use feature reduction and selection
- Number of targets
 - *Rule of thumb: >100 for each class*
 - if very unbalanced, use stratified sampling

preprocessing 5

Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
 - Accuracy
 - Completeness
 - Consistency
 - Timeliness
 - Believability
 - Value added
 - Interpretability
 - Accessibility
- Broad categories:
 - intrinsic, contextual, representational, and accessibility.

preprocessing 6

Data Cleaning: Metadata

- **Field types:**
 - **binary, nominal (categorical), ordinal, numeric, ...**
 - **For nominal fields: tables translating codes to full descriptions**
- **Field role:**
 - input : inputs for modeling
 - target : output
 - id/auxiliary : keep, but not use for modeling
 - ignore : don't use for modeling
 - weight : instance weight
 - ...
- **Field descriptions**

preprocessing 13

Data Cleaning: Reformatting

Convert data to a standard format (e.g. arff or csv)

- Missing values
- Unified date format
- Binning of numeric data
- Fix errors and outliers
- Convert nominal fields whose values have order to numeric.
 - **Q: Why?** A: to be able to use ">" and "<" comparisons on these fields)

preprocessing 14

Missing Data

- Data is not always available
 - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred.

preprocessing 15

How to Handle Missing Data?

- **Ignore the tuple:** usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably).
- **Fill in the missing value manually:** tedious + infeasible?
- **Use a global constant to fill in the missing value:** e.g., "unknown", a new class?!
- **Imputation:** Use the attribute mean to fill in the missing value, or use the attribute mean for all samples belonging to the same class to fill in the missing value: smarter
- **Use the most probable value to fill in the missing value: inference-based**
such as Bayesian formula or decision tree

preprocessing 16

Data Cleaning: Unified Date Format

- We want to transform all dates to the same format internally
- Some systems accept dates in many formats
 - e.g. "Sep 24, 2003", 9/24/03, 24.09.03, etc
 - dates are transformed internally to a standard value
- Frequently, just the year (YYYY) is sufficient
- For more details, we may need the month, the day, the hour, etc
- Representing date as YYYYMM or YYYYMMDD can be OK, but has problems
- **Q: What are the problems with YYYYMMDD dates?**
 - A: Ignoring for now the Looming Y10K (year 10,000 crisis ...)
 - YYYYMMDD does not preserve intervals:
 - 20040201 - 20040131 ≠ 20040131 - 20040130
 - This can introduce bias into models

preprocessing 17

Unified Date Format Options

- To preserve intervals, we can use
 - Unix system date: Number of seconds since 1970
 - Number of days since Jan 1, 1960 (SAS)
- **Problem:**
 - values are non-obvious
 - don't help intuition and knowledge discovery
 - harder to verify, easier to make an error

preprocessing 18

KSP Date Format

$$\text{KSP Date} = \text{YYYY} + \frac{\text{days_starting_Jan_1} - 0.5}{365 + 1_{\text{if_leap_year}}}$$

- Preserves intervals (almost)
- The year and quarter are obvious
 - Sep 24, 2003 is $2003 + (267-0.5)/365 = 2003.7301$ (round to 4 digits)
- Consistent with date starting at noon
- Can be extended to include time

preprocessing 19

Y2K issues: 2 digit Year

- 2-digit year in old data – legacy of Y2K
- E.g. Q: Year 02 – is it 1902 or 2002 ?
 - A: Depends on context (e.g. child birthday or year of house construction)
 - Typical approach: CUTOFF year, e.g. 30
 - if $YY < \text{CUTOFF}$, then 20YY, else 19YY

preprocessing 20

Conversion: Nominal to Numeric

- Some tools can deal with nominal values internally
- Other methods (neural nets, regression, nearest neighbor) require only numeric inputs
- To use nominal fields in such methods need to convert them to a numeric value
 - Q: Why not ignore nominal fields altogether?
 - A: They may contain valuable information
- Different strategies for binary, ordered, multi-valued nominal fields

preprocessing 21

Conversion: Binary to Numeric

- Binary fields
 - E.g. Gender=M, F
- Convert to Field_0_1 with 0, 1 values
 - e.g. Gender = M → Gender_0_1 = 0
 - Gender = F → Gender_0_1 = 1

preprocessing 22

Conversion: Ordered to Numeric

- Ordered attributes (e.g. Grade) can be converted to numbers preserving *natural* order, e.g.
 - A → 4.0
 - A- → 3.7
 - B+ → 3.3
 - B → 3.0
- Q: Why is it important to preserve *natural* order?
- A: To allow meaningful comparisons, e.g. Grade > 3.5

preprocessing 23

Conversion: Nominal, Few Values

- Multi-valued, unordered attributes with small (*rule of thumb* < 20) no. of values
 - e.g. Color=Red, Orange, Yellow, ..., Violet
 - for each value v create a binary “flag” variable C_v , which is 1 if Color= v , 0 otherwise

ID	Color	...	ID	C_re	C_orang	C_yello	...
371	red		371	1	0	0	
433	yellow		433	0	0	1	

preprocessing 24

Conversion: Nominal, Many Values

- Examples:
 - US State Code (50 values)
 - Profession Code (7,000 values, but only few frequent)
- Q: How to deal with such fields ?
- A: Ignore ID-like fields whose values are unique for each record
- For other fields, group values “naturally”:
 - e.g. 50 US States → 3 or 5 regions
 - Profession – select most frequent ones, group the rest
- Create binary flag-fields for selected values

preprocessing 25

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may due to
 - faulty data collection instruments
 - data entry problems
 - data transmission problems
 - technology limitation
 - inconsistency in naming convention
- Other data problems which requires data cleaning
 - duplicate records
 - incomplete data
 - inconsistent data

preprocessing 26

How to Handle Noisy Data?

- Binning method:
 - first sort data and partition into (equi-depth) bins
 - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
 - detect and remove outliers
- Combined computer and human inspection
 - detect suspicious values and check by human
- Regression
 - smooth by fitting the data into regression functions

preprocessing 27

Simple Discretization Methods: Binning

- Equal-width (distance) partitioning:
 - It divides the range into N intervals of equal size: uniform grid
 - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B-A)/N$.
 - The most straightforward
 - But outliers may dominate presentation
 - Skewed data is not handled well.
- Equal-depth (frequency) partitioning:
 - It divides the range into N intervals, each containing approximately same number of samples
 - Good data scaling
 - Managing categorical attributes can be tricky.

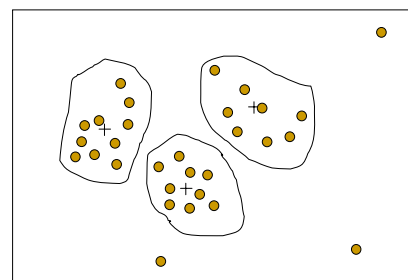
preprocessing 28

Binning Methods for Data Smoothing

- * Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- * Partition into (equi-depth) bins:
 - Bin 1: 4, 8, 9, 15
 - Bin 2: 21, 21, 24, 25
 - Bin 3: 26, 28, 29, 34
- * Smoothing by bin means:
 - Bin 1: 9, 9, 9, 9
 - Bin 2: 23, 23, 23, 23
 - Bin 3: 29, 29, 29, 29
- * Smoothing by bin boundaries:
 - Bin 1: 4, 4, 4, 15
 - Bin 2: 21, 21, 25, 25
 - Bin 3: 26, 26, 26, 34

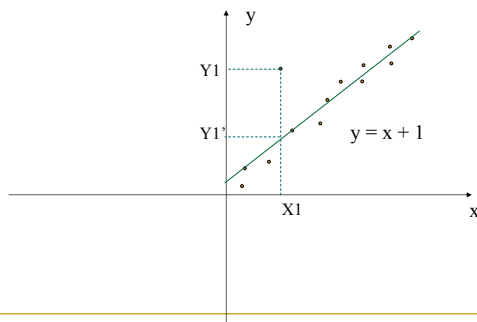
preprocessing 29

Cluster Analysis



preprocessing 30

Regression



preprocessing 31

Dzisiejsze tematy

- Wstęp
- Metody czyszczenia danych
- Transformacja i integracja (scalanie) danych
- Redukcja danych
- Dyskretyzacja i utworzenie hierarchii pojęć
- Podsumowanie

preprocessing 32

Data Integration

- Data integration:
 - combines data from multiple sources into a coherent store
- Schema integration
 - integrate metadata from different sources
 - Entity identification problem: identify real world entities from multiple data sources, e.g., A.cust-id ≡ B.cust-#
- Detecting and resolving data value conflicts
 - for the same real world entity, attribute values from different sources are different
 - possible reasons: different representations, different scales, e.g., metric vs. British units

preprocessing 33

Handling Redundant Data in Data Integration

- Redundant data occur often when integration of multiple databases
 - The same attribute may have different names in different databases
 - One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant data may be able to be detected by correlational analysis
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

preprocessing 34

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
- Attribute/feature construction
 - New attributes constructed from the given ones

preprocessing 35

Data Transformation: Normalization

- min-max normalization

$$v' = \frac{v - \min_s}{\max_s - \min_s} (\text{new_max}_s - \text{new_min}_s) + \text{new_min}_s$$

- z-score normalization

$$v' = \frac{v - \text{mean}_A}{\text{stand_dev}_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

preprocessing 36

Unbalanced Target Distribution

- Sometimes, classes have very unequal frequency
 - Attrition prediction: 97% stay, 3% attrite (in a month)
 - medical diagnosis: 90% healthy, 10% disease
 - eCommerce: 99% don't buy, 1% buy
 - Security: >99.99% of Americans are not terrorists
- Similar situation with multiple classes
- Majority class classifier can be 97% correct, but useless

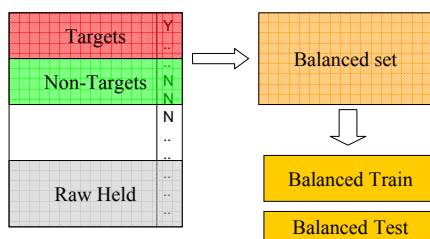
preprocessing 37

Handling Unbalanced Data

- With two classes: let positive targets be a minority
- Separate raw held-aside set (e.g. 30% of data) and raw train
 - put aside raw held-aside and don't use it till the final model
- Select remaining positive targets (e.g. 70% of all targets) from raw train
- Join with equal number of negative targets from raw train, and randomly sort it.
- Separate randomized balanced set into balanced train and balanced test

preprocessing 38

Building Balanced Train Sets



preprocessing 39

Learning with Unbalanced Data

- Build models on balanced train/test sets
- Estimate the final results (lift curve) on the raw held set
- Can generalize “balancing” to multiple classes
 - stratified sampling
 - Ensure that each class is represented with approximately equal proportions in train and test

preprocessing 40

Dzisiejsze tematy

- Wstęp
- Metody czyszczenia danych
- Transformacja i integracja (scalanie) danych
- Redukcja danych
- Dyskretyzacja i utworzenie hierarchii pojęć
- Podsumowanie

preprocessing 41

Data Reduction Strategies

- Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
 - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Data reduction strategies
 - Data cube aggregation
 - Dimensionality reduction
 - Numerosity reduction
 - Discretization and concept hierarchy generation

preprocessing 42

Data Cube Aggregation

- The lowest level of a data cube
 - the aggregated data for an **individual entity of interest**
 - e.g., a customer in a phone calling data warehouse.
- Multiple levels of aggregation in data cubes
 - Further reduce the size of data to deal with
- Reference appropriate levels
 - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

preprocessing 43

Dimensionality Reduction

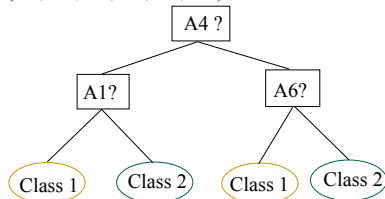
- Feature selection (i.e., attribute subset selection):
 - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
 - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
 - step-wise forward selection
 - step-wise backward elimination
 - combining forward selection and backward elimination
 - **decision-tree induction**

preprocessing 44

Example of Decision Tree Induction

Initial attribute set:

{A1, A2, A3, A4, A5, A6}



.....> Reduced attribute set: {A1, A4, A6}

preprocessing 45

Attribute Selection

- There are 2^d possible sub-features of d features
- First: Remove attributes with no or little variability
 - Examine the number of distinct field values
 - *Rule of thumb: remove a field where almost all values are the same (e.g. null), except possibly in minp % or less of all records.*
 - *minp* could be 0.5% or more generally less than 5% of the number of targets of the smallest class
- What is good N?
 - Rule of thumb -- keep top 50 fields

preprocessing 46

Heuristic Feature Selection Methods

- Several heuristic feature selection methods:
 - Best single features under the feature independence assumption: choose by significance tests.
 - Best step-wise feature selection:
 - The best single-feature is picked first
 - Then next best feature condition to the first, ...
 - Step-wise feature elimination:
 - Repeatedly eliminate the worst feature
 - Best combined feature selection and elimination:
 - Optimal branch and bound:
 - Use feature elimination and backtracking

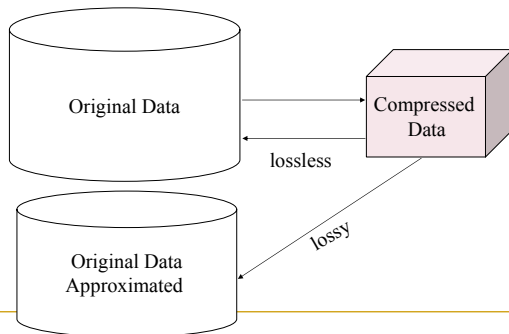
preprocessing 47

Data Compression

- String compression
 - There are extensive theories and well-tuned algorithms
 - Typically lossless
 - But only limited manipulation is possible without expansion
- Audio/video compression
 - Typically lossy compression, with progressive refinement
 - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
 - Typically short and vary slowly with time

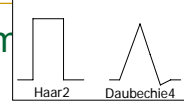
preprocessing 48

Data Compression



preprocessing 49

Wavelet Transform



- Discrete wavelet transform (DWT): linear signal processing
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
 - Length, L , must be an integer power of 2 (padding with 0s, when necessary)
 - Each transform has 2 functions: smoothing, difference
 - Applies to pairs of data, resulting in two set of data of length $L/2$
 - Applies two functions recursively, until reaches the desired length

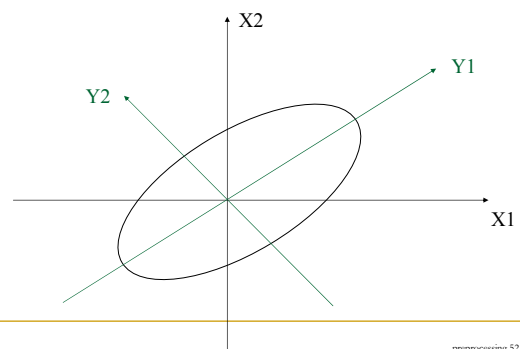
preprocessing 50

Principal Component Analysis

- Given N data vectors from k -dimensions, find $c \leq k$ orthogonal vectors that can be best used to represent data
 - The original data set is reduced to one consisting of N data vectors on c principal components (reduced dimensions)
- Each data vector is a linear combination of the c principal component vectors
- Works for numeric data only
- Used when the number of dimensions is large

preprocessing 51

Principal Component Analysis



preprocessing 52

Numerosity Reduction

- Parametric methods
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Log-linear models: obtain value at a point in m -D space as the product on appropriate marginal subspaces
- Non-parametric methods
 - Do not assume models
 - Major families: histograms, clustering, sampling

preprocessing 53

Regression and Log-Linear Models

- Linear regression: Data are modeled to fit a straight line
 - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability distributions

preprocessing 54

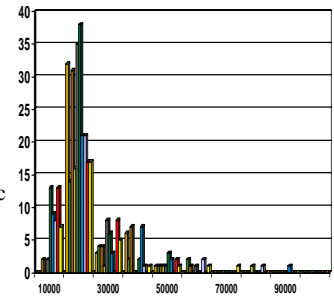
Regress Analysis and Log-Linear Models

- **Linear regression:** $Y = \alpha + \beta X$
 - Two parameters, α and β specify the line and are to be estimated by using the data at hand.
 - using the least squares criterion to the known values of $Y_1, Y_2, \dots, X_1, X_2, \dots$
- **Multiple regression:** $Y = b_0 + b_1 X_1 + b_2 X_2$.
 - Many nonlinear functions can be transformed into the above.
- **Log-linear models:**
 - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
 - Probability: $p(a, b, c, d) = \alpha ab \beta ac \gamma ad \delta bcd$

preprocessing 55

Histograms

- A popular data reduction technique
- Divide data into buckets and store average (sum) for each bucket
- Can be constructed optimally in one dimension using dynamic programming
- Related to quantization problems.



preprocessing 56

Clustering

- Partition data set into clusters, and one can store cluster representation only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms, further detailed in Chapter 8

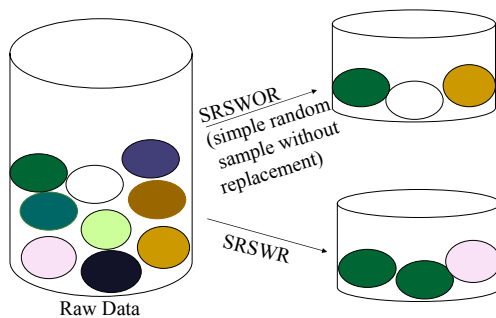
preprocessing 57

Sampling

- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a **representative** subset of the data
 - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
 - Stratified sampling:
 - Approximate the percentage of each class (or subpopulation of interest) in the overall database
 - Used in conjunction with skewed data
- Sampling may not reduce database I/Os (page at a time).

preprocessing 58

Sampling

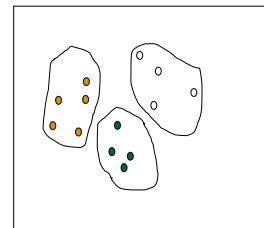
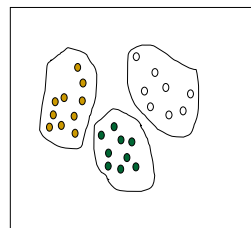


preprocessing 59

Sampling

Raw Data

Cluster/Stratified Sample



preprocessing 60

Hierarchical Reduction

- Use multi-resolution structure with different degrees of reduction
- Hierarchical clustering is often performed but tends to define partitions of data sets rather than “clusters”
- Parametric methods are usually not amenable to hierarchical representation
- Hierarchical aggregation
 - An index tree hierarchically divides a data set into partitions by value range of some attributes
 - Each partition can be considered as a bucket
 - Thus an index tree with aggregates stored at each node is a hierarchical histogram

preprocessing 61

Dzisiejsze tematy

- Wstęp
- Metody czyszczenia danych
- Transformacja i integracja (scalanie) danych
- Redukcja danych
- **Dyskretyzacja i utworzenie hierarchii pojęć**
- Podsumowanie

preprocessing 62

Discretization

- Three types of attributes:
 - Nominal — values from an unordered set
 - Ordinal — values from an ordered set
 - Continuous — real numbers
- Discretization:
 - divide the range of a continuous attribute into intervals
 - Some classification algorithms only accept categorical attributes.
 - Reduce data size by discretization
 - Prepare for further analysis

preprocessing 63

Discretization and Concept hierarchy

- Discretization
 - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.
- Concept hierarchies
 - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

preprocessing 64

Discretization and concept hierarchy generation for numeric data.

- Binning (see sections before)
- Histogram analysis (see sections before)
- Clustering analysis (see sections before)
- Entropy-based discretization
- Segmentation by natural partitioning

preprocessing 65

Entropy-Based Discretization

- Given a set of samples S , if S is partitioned into two intervals S_1 and S_2 using boundary T , the entropy after partitioning is
$$E(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$
- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,
$$Ent(S) - E(T,S) > \delta$$
- Experiments show that it may reduce data size and improve classification accuracy

preprocessing 66

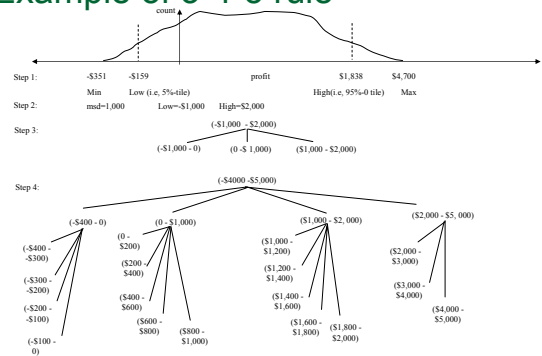
Segmentation by natural partitioning

3-4-5 rule can be used to segment numeric data into relatively uniform, "natural" intervals.

- * If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals
- * If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals
- * If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

preprocessing 67

Example of 3-4-5 rule



preprocessing 68

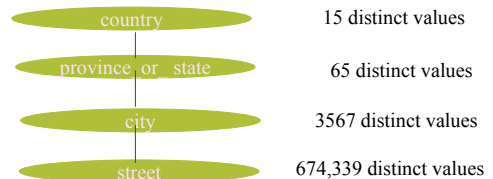
Concept hierarchy generation for categorical data

- Specification of a partial ordering of attributes explicitly at the schema level by users or experts
- Specification of a portion of a hierarchy by explicit data grouping
- Specification of a set of attributes, but not of their partial ordering
- Specification of only a partial set of attributes

preprocessing 69

Specification of a set of attributes

Concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set. The attribute with the most distinct values is placed at the lowest level of the hierarchy.



preprocessing 70

Dzisiejsze tematy

- Wstęp
- Metody czyszczenia danych
- Transformacja i integracja (scalanie) danych
- Redukcja danych
- Dyskretyzacja i utworzenie hierarchii pojęć
- Podsumowanie

preprocessing 71

Outline

- Introduction
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

preprocessing 72