Data Preprocessing

CS 5331 by Rattikorn Hewett
Texas Tech University

Outline

- Motivation
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and hierarchy generation
- Summary

Motivation

- Real-world data are
  - incomplete: missing important attributes, or attribute values, or values giving aggregate data
    e.g., Age = “”
  - noisy: erroneous data or outliers
    e.g., Age = “2000”
  - inconsistent: discrepancies in codes or names or duplicate data
    e.g., Age = “20” Birthday = “03/07/1960”
    Sex = “F” Sex = “Female”

How did this happen?

- Incomplete data
  - Data are not available when collected
  - Data are not considered important to record
  - Errors: forgot to record, delete to eliminate inconsistent, equipment malfunctions
- Noisy data
  - Data collected/entered incorrectly due to faulty equipment, human or computer errors
  - Data transmitted incorrectly due to technology limitations (e.g., buffer size)
- Inconsistent data
  - Different naming conventions in different data sources
  - Functional dependency violation

Relevance to data mining

- Bad data → bad mining results → bad decisions
  - duplicate or missing data may cause incorrect or even misleading statistics.
  - consistent integration of quality data → good warehouses
- Data preprocessing aims to improve
  - Quality of data and thus mining results
  - Efficiency and ease of data mining process

Measures of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness, believability, value added, interpretability
  - Accessibility
- Broad categories:
  - Intrinsic (inherent)
  - Contextual
  - Representational
  - Accessible

Data Preprocessing Techniques

- Data cleaning
  - Fill in missing values, smooth out noise, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integrate data from multiple databases, data cubes, or files
- Data transformation
  - Normalize (scale to certain range)
- Data reduction
  - Obtain reduced representation in volume without sacrificing quality of mining results
  - e.g., **dimension reduction** – remove irrelevant attributes
    discretization – reduce numerical data into discrete data

Outline

- Motivation
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and hierarchy generalization
- Summary

Data Cleaning

"Data cleaning is the number one problem in data warehousing"—DCI survey

- Tasks
  - Fill in missing values
  - Identify outliers and smooth out noises
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

Missing data

- Ignore the tuple with missing values
  e.g., in classification when class label is missing — not effective when the % of missing values per attribute varies considerably.
- Fill in the missing value manually — tedious + infeasible?
- Fill in the missing value automatically with
  - global constant e.g., "unknown" — a new class?
  - attribute mean
  - attribute mean for all samples of the same class
  - most probable value e.g., regression-based or inference-based such as Bayesian formula or decision tree (Ch 7)

Which of these three techniques biases the data?

Noisy data & Smoothing Techniques

Noise is a random error or variance in a measured variable

Data smoothing techniques:

- Binning
  - Sort data and partition into (equi-depth) bins (or buckets)
  - Local smoothing by
    - bin means
    - bin median
    - bin boundaries

Simple Discretization: Binning

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

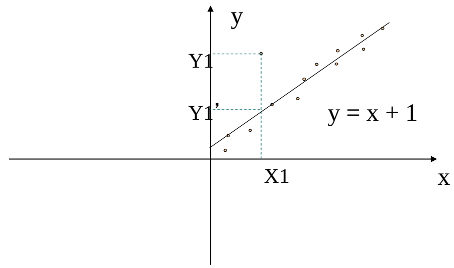
Examples of binning

- Sorted data (e.g., ascending in price)
  - 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
  - N = data set size = 12, B = number of bins/intervals
- Partition into three (equi-depth) bins (say, B = 3 → each bin has N/3 = 4 elements):
  - 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: min and max are boundaries
  Each bin value is replaced by closest boundary
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

For equi-width bins: width = (max-min)/B = (34-4)/3 = 10
I.e., interval range of values is 10
Bin1(0-10): 4,8,9
Bin2(11-20): 15
Bin3(21-30): 21,21,24,25,26,28,29
Bin4(31-40): 34 → outlier can misrepresent the partitioning

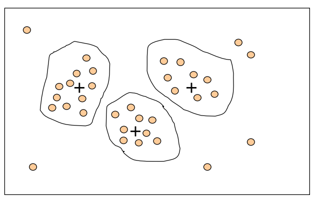
Smoothing Techniques (cont)

- Regression
  - smooth by fitting the data into regression functions



Smoothing Techniques (cont)

- Clustering
  - detect and remove outliers



Smoothing Techniques (cont)

- Combined computer and human inspection
  - Automatically detect suspicious values

    e.g., deviation from known/expected value above threshold
  - Manually select actual "surprise" vs. "garbage"

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and hierarchy generation
- Summary

Data Integration

Data integration

combines data from multiple sources into a coherent store

Issues:

- Schema integration - how to identify matching of entities from multiple data sources → **Entity identification problem**

  e.g., A.customer-id ≡ B.customer-num

  - Use metadata to help avoid integration errors

Data Integration (cont)

Issues:

- Redundancy
  - occurs in an attribute when it can be "derived" from another table
  - can be caused by inconsistent attribute naming
  - can be detected by "correlation analysis"

$$\text{Corr(A, B)} = \frac{\sum (A - \bar{A})(B - \bar{B})}{(n - 1)s_A s_B}$$

+ve → highly correlated → A and B are redundant

0 → independent

-ve → negatively correlated → Are A and B redundant?

  - Redundancy between attributes → detect duplication in tuples

Data Integration (cont)

Issues:

- Data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, coding or scales

    e.g., weight values in: metric vs. British units

    cost values: include tax vs. exclude tax
  - Detection and resolving these conflicts require careful data integration

Data transformation

Change data into forms for mining. May involves:

- Smoothing: remove noise from data → Cleaning
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing → Reduction
- Normalization: scaled to be in a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones e.g., area ← width x length, happy ← rich & healthy

Data normalization

- Transform data to be in a specific range
- Useful in
  - Neural net back propagation – speedup learning
  - Distance-based mining method (e.g., clustering) – prevent attributes with initial large ranges from outweighing those with initial small ranges
- Three techniques: min-max, z-score and decimal scaling

Data normalization (cont)

- Min-max:

  For a given attribute value range, $[min, max] \rightarrow [min', max']$

$$v' = \frac{v - min}{max - min}(max' - min') + min'$$

  - Can detect "out of bound" data
  - Outliers may dominate the normalization

Data normalization (cont)

- Z-score (zero-mean) for value v of attribute A

$$v' = \frac{v - \overline{A}}{s_A}$$

- Useful when
  - Min and max value of A are unknown
  - Outliers dominate the min-max normalization

Z-Score (Example)

v	v'			v	v'		
0.18	-0.84	Avg	0.68	20	-.26	Avg	34.3
0.60	-0.14	sdev	0.59	40	.11	sdev	55.9
0.52	-0.27			5	.55		
0.25	-0.72			70	4		
0.80	0.20			32	-.05		
0.55	-0.22			8	-.48		
0.92	0.40			5	-.53		
0.21	-0.79			15	-.35		
0.64	-0.07			250	3.87		
0.20	-0.80			32	-.05		
0.63	-0.09			18	-.30		
0.70	0.04			10	-.44		
0.67	-0.02			-14	-.87		
0.58	-0.17			22	-.23		
0.98	0.50			45	.20		
0.81	0.22			60	.47		
0.10	-0.97			-5	-.71		
0.82	0.24			7	-.49		
0.50	-0.30			2	-.58		
3.00	3.87			4	-.55		

25

Data normalization (cont)

- Decimal scaling

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

Example: values of A range from –300 to 250
j = 1 → max(|v' |) = max(|v|)/10 = 300/10 > 1
j = 2 → max(|v' |) = max(|v|)/100 = 300/100 > 1
j = 3 → max(|v' |) = max(|v|)/1000 = 300/1000 < 1
Thus, x is normalized to x/1000 (e.g., 99 → .099)

26

Outline

- Motivation
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and hierarchy generation
- Summary

27

Data Reduction

- Obtains a reduced representation of the data set that is much smaller in volume
  and yet produces (almost) the same analytical results
- Data reduction strategies
  - Data cube aggregation
  - Dimension reduction — remove unimportant attributes
  - Data compression
  - Numerosity reduction — replace data by smaller models
  - Discretization and hierarchy generation

28

Data Cube Aggregation

- Aggregation gives summarized data represented in a smaller volume than initial data

  E.g., total monthly sales (12 entries) vs. total annual sales (one entry)
- Each cell of a data cube holds an aggregate data value ~ a point in a multi-dimensional space
- Base cuboid ~ an entity of interest – should be a useful unit
- Aggregate to cuboids at a higher level (of lattice) further reduces the data size
- Should use the *smallest* cuboid relevant to OLAP queries

Dimension Reduction

Goal:

- To detect/remove irrelevant/redundant attributes/dimensions of the data

Example: Mining to find customer's profile for marketing a product

- CD's: age vs. phone number
- Grocery items: Can you name three relevant attributes?

Motivation:

- Irrelevant attributes → poor mining results & larger volume of data → slower mining process

Dimension Reduction (cont)

Feature selection (i.e., attribute subset selection)

- Goal: To find a minimum set of features such that the resulting probability distribution of the data classes is close to the original distribution obtained using all features
- Additional Benefit: reduced number of features → resulting patterns are easier to understand
- **Issue:**
  - Computational complexity – 2^d possible subsets for *d* features
  - Solution → **Heuristic search**, e.g., greedy search

Feature selection

Heuristic search

- Evaluation functions to guide search direction can be
  - Test for *statistically significance* of attributes
    - What's the drawback of this method?
  - Information-theoretic measures, e.g., *information gain* (keep attribute that has high information gain to the mining task) as in *decision tree induction*
- Search strategies can be
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - Optimal branch and bound

Search strategies

- Stepwise forward
  - Starts with empty set of attribute
  - Iteratively select the best attribute
- Stepwise backward
  - Starts with a full set of attributes
  - Iteratively remove the worst
- Combined
  - Each iteration step, add the best and remove the worst attribute
- Optimal branch and bound
  - Use feature elimination and backtracking

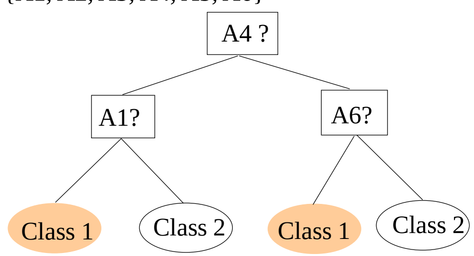
Feature selection (cont)

- Decision tree induction (e.g., ID3, C4.5 – see later)
  - Input: data
  - Output: A decision tree that best represents the data
  - Attributes that do not appear on the tree are assumed to be irrelevant

  (See example next slide)
- Wrapper approach [Kohavi and John 97]
  - Greedy search to select set of attributes for classification
  - Evaluation function is based on errors obtained from using a mining algorithm (e.g., decision tree induction) for classification

Example: Decision Tree Induction

Initial attribute set:
{A1, A2, A3, A4, A5, A6}



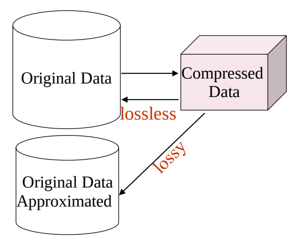
------> Reduced attribute set: {A1, A4, A6}

Data compression

Encoding/transformations are applied to obtain "compressed" representation of the original data

Two types:

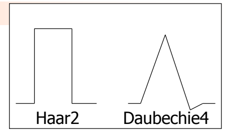
- Lossless compression: can reconstruct original data from the compressed data
- Lossy compression: can reconstruct only approximation of the original data



Data Compression (cont)

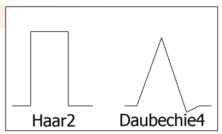
- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically <u>lossless</u>
  - Allow limited data manipulation
- Audio/video compression
  - Typically <u>lossy</u> compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio – think of data collection
  - Typically short and vary slowly with time
- Two important lossy data compression techniques:
  - Wavelet
  - Principal components

Wavelet Transformation



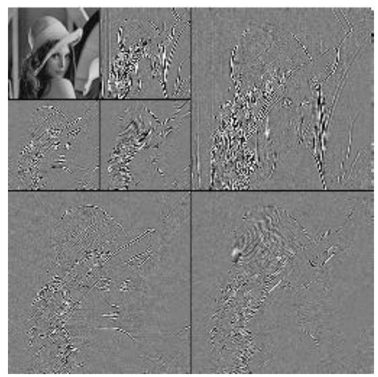
- Discrete wavelet transform (DWT) – a linear signal processing technique that transforms,
  vector of data → vector of coeffs (of the same length)
- Popular wavelet transforms: Haar2, Daubechie4
  (the number is associated to properties of coeffs)
- Approximation of data can be retained by storing only a small fraction of the strongest of the wavelet coefficients
  - Approximated data – noises removed without losing features
- Similar to discrete Fourier transform (DFT), however
  - DWT more accurate (for the same number of coefficients)
  - DWT requires less space

Wavelet Transformation



- Method (sketched):
  - Data vector length, L, must be an integer power of 2 (padding with 0s, when necessary)
  - Each transform has 2 functions: <u>smoothing, difference</u>
  - Apply the transform to pairs of data (low and high frequency contents), resulting in two set of data of length L/2 – repeat recursively, until reach the desired length
  - Select values from data sets from the above iterations to be the wavelet coefficients of the transformed data
- Apply the *inverse* of the DWT used to a set of wavelet coefficients to reconstruct approximation of the original data
- Good results on sparse, skewed or ordered attribute data
  – better results than JPEG compression

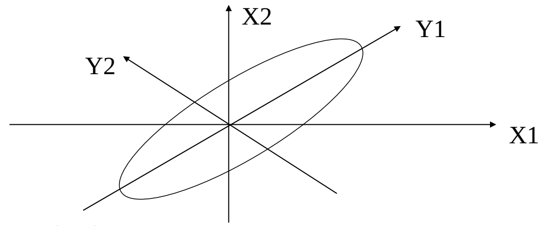
DWT for Image Compression



Principal Component Analysis

- The original data set (N k-dimensional vectors) is reduced to data set of N vectors on c *principal components* (k-dimensional orthogonal vectors that can be best used to represent the data) i.e., N × k → N × c, where c ≤ k
  - Each data vector is a linear combination of the c principal component vectors (not necessary a subset of initial attribute set)
- Works for numeric data only
- Inexpensive computation - used when the number of dimensions is large

Principal Component Analysis



- The principle components
  - serve as a new set of axes for the data
  - are ordered by its degree of variance of data

E.g., Y1, Y2 are first two PCs for the data on the plane X1X2

Variance of data based on Y1 axis is higher than those of Y2

Numerosity Reduction

- Parametric methods
  - Assume the data fits some model
    - estimate model parameters
    - store only the parameters
    - discard the actual data (except possible outliers)
  - Examples: regression and Log-linear models
- Non-parametric methods
  - Do not assume models
    - Store data in reduced representations
    - More amendable to hierarchical structures than parametric method
  - Major families: histograms, clustering, sampling

Parametric methods

- Linear regression
  - Approximates a straight line model: $Y = \alpha + \beta X$
  - Use the least-square method to min error based on distances between the actual data and the model
- Multiple regression
  - Approximates a linear model with multiple predictors: $Y = \alpha + \beta_1 X_1 + \ldots.. + \beta_n X_n$
  - Can be used to approximate non-linear regression model via transformations (Ch 7)
- Log-linear model
  - Approximates a model (or probability distribution) of discrete data

Non-parametric method: Histogram

- Histograms use binning to approximate data distribution
  - For a given attribute, data are partitioned into buckets
  - Each bucket represents a single-value data and its frequency of occurrences
- Partitioning methods:
  - Equiwidth: each bucket has the same value range
  - Equidepth: each bucket has the same frequency of data occurrences
  - V-optimal: the histogram with the number of buckets that has the least "variance" (see text for "variance")
  - MaxDiff: use difference between each pair of adjacent values to determine a bucket boundary

Histograms (cont)

- Example of Max-diff: 1,1,2,2,2,3,5,5,7,8,8,9
  - A bucket boundary is established between each pair of pairs having 3 largest differences:

    1,1,2,2,2,3 | 5,5,7,8,8 | 9
- Histograms are effective for approximating
  - Sparse vs. dense data
  - Uniform vs. skewed data
- For multi-dimensional data, histograms are typically effective up to five dimensions

Non-parametric method: Clustering

- Partition data objects into clusters so data objects within the same cluster are "similar"
- "quality" of a cluster can be measured by
  - Max distance between two objects in the cluster
  - Centroid distance – average distance of each cluster object from the centroid of the cluster
- Actual data are reduced to be represented by clusters
- Some data can't be effectively clustered e.g., smeared data
- Can have hierarchical clusters
- Further detailed techniques and definitions in Ch 8

Non-parametric methods: Sampling

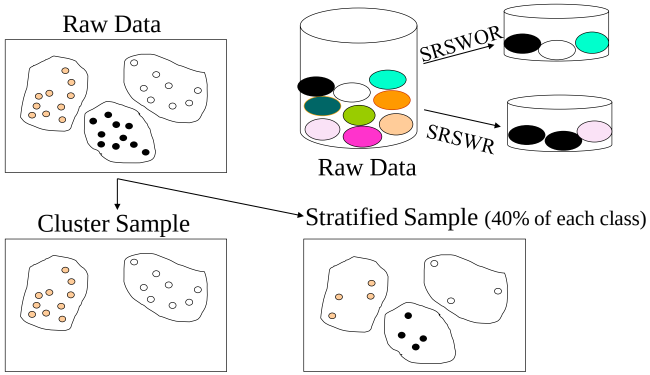
Data reduction by finding a representative data sample

Sampling techniques:

- Simple Random Sampling (SRS)
  - WithOut Replacement (SRSWOR)
  - With Replacement (SRSWR)
- Cluster Sample:
  - Data set are clustered into M clusters
  - Apply SRS to randomly select m of the M clusters
- Stratified sample - adaptive sampling method
  - Apply SRS to each class (or stratum) of data to ensure that a sample will have representative data from each class

When should we use stratified sampling?

Sampling



Non-parametric methods: Sampling

- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the original data
- Let N be the data set size
  n be a sample size
  - Cost of obtaining a sample is proportional to ______
- Sampling complexity increases linearly as the number of dimensions increase

Outline

- Motivation
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and hierarchy generation
- Summary

Discretization

- Three types of attribute values:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divides the range of values of a continuous attribute into intervals
  - Why?
    - Prepare data for analysis by mining algorithms that only accept categorical attributes.
    - Reduce data size

Discretization & Concept hierarchy

- Discretization
  - reduces the number of values for a given continuous attribute by dividing the range of the attribute into intervals

    E.g., age group: [1,10], [11,19], [20,40], [41,59], [60,100]
- Concept hierarchies
  - reduces the data by collecting and replacing low level concepts by higher level concepts

    E.g., orange, grapefruit, apple, banana

    → citrus, non-citrus → fruit → produce

53

Entropy

- Shannon's information theoretic measure - approx. information captured from $m_1, \ldots, m_n$

$$Ent(\{m_1,\ldots,m_n\}) = -\sum_i p(m_i)\log_2(p(m_i))$$

- For a r.v. X, $Ent(X) = -\sum_x p(x)\log_2 p(x)$

Example: Toss a balanced coin: H H T H T T H ….

X = {H, T}

P(H) = P(T) = ½

Ent (X) = – ½ $\log_2(½)$ – ½ $\log_2(½)$ = – $\log_2(½)$ = – (0 – 1) = 1

What if the coin is a two-headed coin?

Ent(X) = 0 ~ information captured from X is certain

54

Entropy-based discretization

- For an attribute value set S, each labeled with a class in C and p_i is a probability that class i is in S, then

$$Ent(S) = -\sum_{i \in C} p_i \log_2 p_i$$

Example: Form of element: (Data value, class in C), where C = {A, B}

S = {(1, A), (1, B), (3, A), (5, B), (5, B)}

Ent(S) = – 2/5 $\log_2(2/5)$ – 3/5 $\log_2(3/5)$ ~

Information (i.e., classification) captured by data values of S

55

Entropy-based discretization (cont)

Goal: to discretize an attribute value set S in ways that it maximize information captured by S to classify classes in C

- If S is partitioned by T into two intervals S1 (–∞, T) and S2 [T, ∞), the expected class information entropy induced by T is

$$I(S,T) = \frac{|S_1|}{|S|}Ent(S_1) + \frac{|S_2|}{|S|}Ent(S_2)$$

- Information gain: $Gain(S, T) = Ent(S) – I(S, T)$

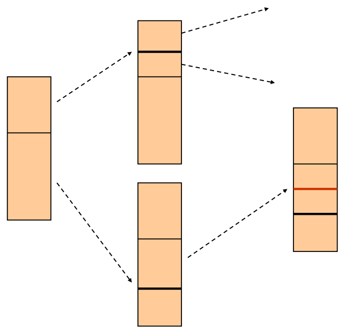
Idea:

- Find T (among possible data points) that minimizes $I(S, T)$ (i.e., max information gain)
- Recursively find new T to the partitions obtained until some stopping criterion is met, e.g., $Gain(S, T) > \delta$

→ may reduce data size and improve classification accuracy

56

Entropy-based discretization (cont)



Terminate when
$Gain(S, T) > \delta$

E.g., in [Fayyad & Irani, 1993]

$$\delta = \frac{\log_2(N-1)}{N} + \frac{f(S,T)}{N}$$

$$f(S, T) = \log_2(3^k - 2) - [k\,Ent(S) - k_1\,Ent(S_1) - k_2(Ent(S_2))]$$

N = number of data points in S

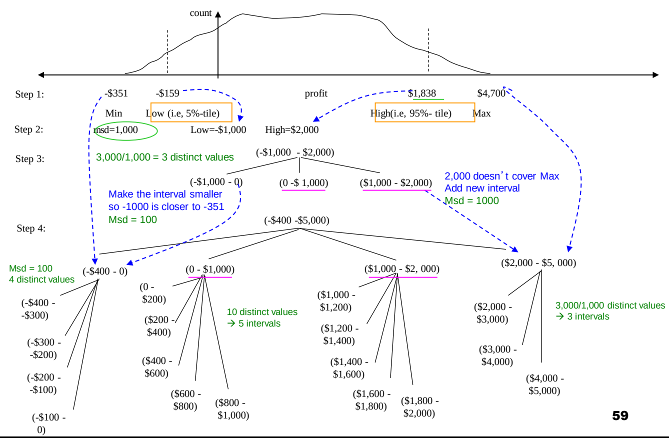
k_i (or k) is the number of class labels represented in set S_i (or S)

Segmentation by Natural Partitioning

Idea: want boundaries of range to be intuitive (or natural)
E.g., 50 vs. 52.7

- A **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

Example of 3-4-5 Rule



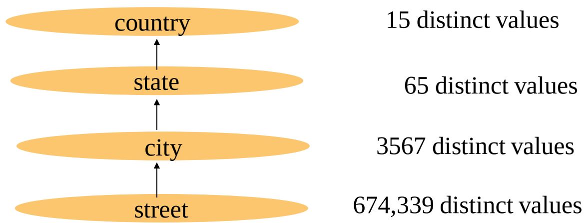
Hierarchy Generation

- Categorical Data are
  - Discrete
  - Finite but possibly large (e.g., city, name)
  - No ordering
- How to create concept hierarchy of categorical data
  - User specified total/partial ordering of attributes explicitly
    E.g., street<city<state<country)
  - Specify a portion of a hierarchy (by data groupings)
    E.g., {Texas, Alabama} ⊂ Southern_US as part of state<country
  - Specify a partial set of attributes
    E.g., only street < city, not others in dimension "location", say
  - Automatically generate partial ordering by analysis of the number of distinct values

    Heuristic: top level hierarchy (most general) has smallest number of distinct values

Automatic Hierarchy Generation

- Some concept hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the given data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Use this with care ! E.g., weekday (7), month(12), year(20, say)



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Summary

- Data preparation is a big issue for both warehousing and mining
- Data preparation includes
  - Data cleaning and data integration
  - Data transformation and normalization
  - Data reduction - feature selection,discretization
- A lot a methods have been developed but still an active area of research

References

- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4
- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997.
- A. Maydanchik, Challenges of Efficient Data Cleansing (DM Review - Data Quality resource portal)
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999.
- D. Quass. A Framework for research in Data Cleaning. (Draft 1999)
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001.
- T. Redman. Data Quality: Management and Technology. Bantam Books, New York, 1992.
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996.
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995.
- http://www.cs.ucla.edu/classes/spring01/cs240b/notes/data-integration1.pdf