Week 7-2

Data Preprocessing

Big Data

Prof. Hwanjo Yu
POSTECH

Data quality: Why preprocess the data?

- Measures for data quality
  - **Accuracy**: correct or wrong, accurate or not
  - **Completeness**: not recorded, unavailable, …
  - **Consistency**: some modified but some not, dangling, …
  - **Timeliness**: timely update?
  - **Trustness**: how trustable the data are correct?
  - **Interpretability**: how easily the data can be understood?

Major tasks in data preprocessing

1. **Data cleaning**
   - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
2. **Data integration**
   - Integration of multiple databases, data cubes, or files
   - Need to handle data redundancy (e.g. chi-square test, correlation analysis)
3. **Data reduction**
   - Dimensionality reduction
   - Numerosity reduction (Sampling)
   - Data compression
4. **Data transformation**
   - Normalization
   - Discretization or Binning

Data cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g. instrument faulty, human or computer error, transmission error
  a. incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
     - e.g. *Occupation=" "* (missing data)
  b. noisy: containing noise, errors, or outliers
     - e.g. *Salary="−10"* (an error)
  c. inconsistent: containing discrepancies in codes or names, e.g.,
     - *Age="42", Birthday="03/07/2010"*
     - Was rating "1, 2, 3", now rating "A, B, C"
     - discrepancy between duplicate records
  d. intentional (e.g. disguised missing data)
     - Jan. 1 as everyone's birthday?

Data cleaning: Incomplete (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

Data cleaning: How to handle missing data?

- Ignore the tuple: usually done when class label is missing (when doing classification)
  - not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g. "unknown", a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula, decision tree, or matrix factorization

Data cleaning: Noisy data

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

Data cleaning: How to handle noisy data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g. deal with possible outliers)
- Do nothing
  - Noise is also valuable.

Data integration

- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust\text{-}id \equiv B.cust\text{-}\#$
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g. Bill Clinton = William Clinton
- 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

Data integration: Handling redundancy

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

Data integration: Correlation analysis (nominal data)

- χ^2 (chi-square) test

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the χ^2 value, the more likely the variables are related
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count
- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

Data integration: Chi-square calculation: An example

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- χ^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group.
- p-value is computed using χ^2 and degree of freedom.

Data integration: Correlation analysis (Numeric data)

- Correlation coefficient (also called Pearson's product moment coefficient)

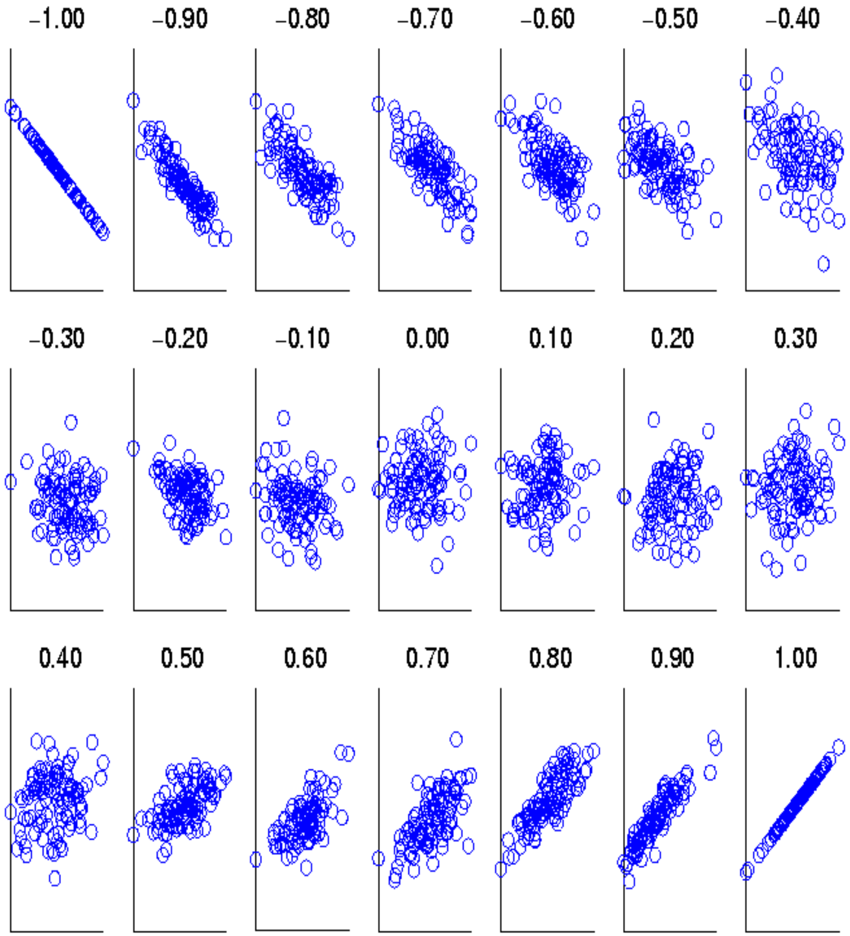
$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_a\sigma_b} = \frac{\sum_{i=1}^{n}(a_i b_i - n\bar{A}\bar{B})}{(n-1)\sigma_a\sigma_b}$$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B,

σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

Data integration: Visually evaluating correlation



Scatter plots showing the similarity from –1 to 1.

Data integration: Correlation (viewed as linear relationship)

- Correlation measures the linear relationship between objects
- To compute correlation, we *standardize* data objects, A and B, and then take their dot product

$$a'_k = (\mathrm{a_k} - \mathrm{mean(A)})/\mathrm{std(A)}$$

$$b'_k = (\mathrm{b_k} - \mathrm{mean(B)})/\mathrm{std(B)}$$

$$\mathrm{Correlation(A,B)} = \mathrm{A}' \bullet B'$$

Data integration: Covariance (Numeric data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $r_{A,B} = \frac{Cov(A,B)}{\sigma_a \sigma_b}$

  - where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- **Positive covariance**: If $\mathrm{Cov}_{A,B} > 0$, then A and B both tend to be larger than their expected values.

- **Negative covariance**: If $\mathrm{Cov}_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.

Data integration: Co-variance: An example

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

- It can be simplified in computation as

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

- Suppose two stocks A and B have the following values in one week: $(2, 5), (3, 8), (5, 10), (4, 11), (6, 14)$.
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - $\mathrm{E(A)} = (2+3+5+4+6)/\ 5 = 20/5 = 4$
  - $\mathrm{E(B)} = (5+8+10+11+14)\ /5 = 48/5 = 9.6$
  - $\mathrm{Cov(A,B)} = (2\times5+3\times8+5\times10+4\times11+6\times14)/5 - 4\times9.6 = 4$
- Thus, A and B rise together since $\mathrm{Cov(A,\ B)} > 0$.

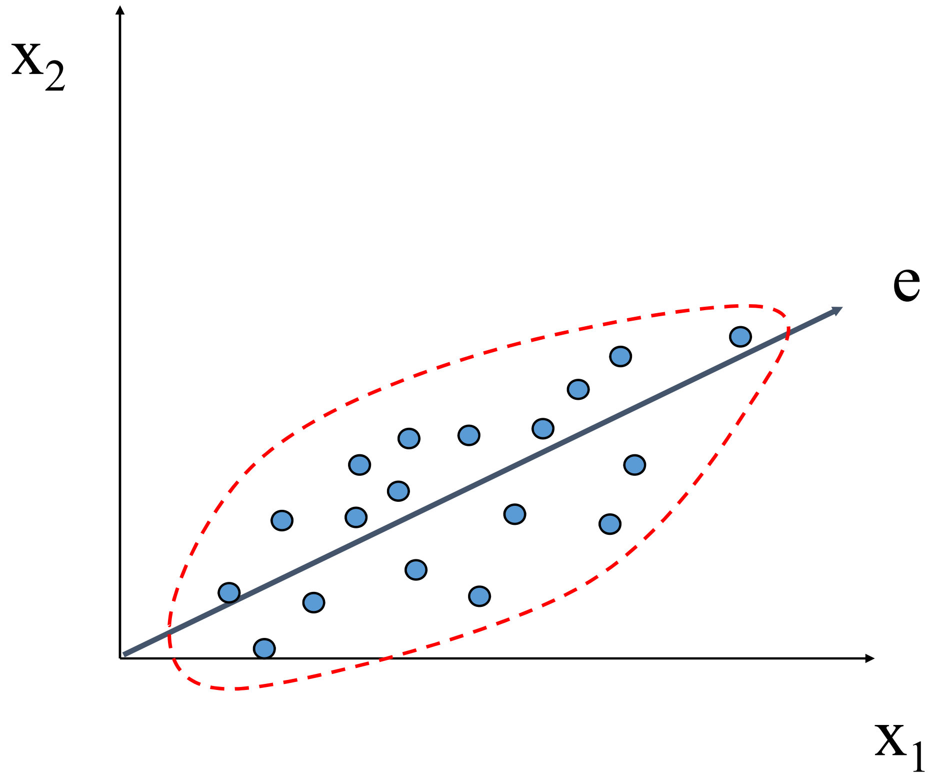
POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Data reduction: Strategies

- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces almost the same (or better) analytical results
- Why data reduction?
  - to reduce running time or to improve the (clustering or classification) accuracy
- Data reduction strategies
  - Dimensionality reduction, e.g. remove unimportant attributes
    - Feature transformation: Principal Components Analysis (PCA), …
    - Feature subset selection: (1) Wrapper method, (2) Filter method
  - Numerosity reduction (sampling)
  - Data compression

Data reduction: Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Data reduction: Principal Component Analysis (steps)

- Given *N* data vectors from *n*-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e. principal components
  - Each input data (vector) is a linear combination of the k principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e. those with low variance (i.e. using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

Data reduction: Attribute subset selection

- Another way to reduce dimensionality of data
- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
  - e.g. purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
  - e.g. students' ID is often irrelevant to the task of predicting students' GPA

Data reduction: Heuristic search in attribute selection

- There are 2^{d} possible attribute combinations of d attributes
- Typical heuristic attribute selection methods:
  1. Select attributes under the attribute independence assumption: choose by significance tests
  2. Step-wise attribute selection (forward):
     - The best single-attribute is picked first
     - Then next best attribute condition to the first, ...
  3. Step-wise attribute elimination (backward):
     - Repeatedly eliminate the worst attribute
  4. Combined attribute selection and elimination

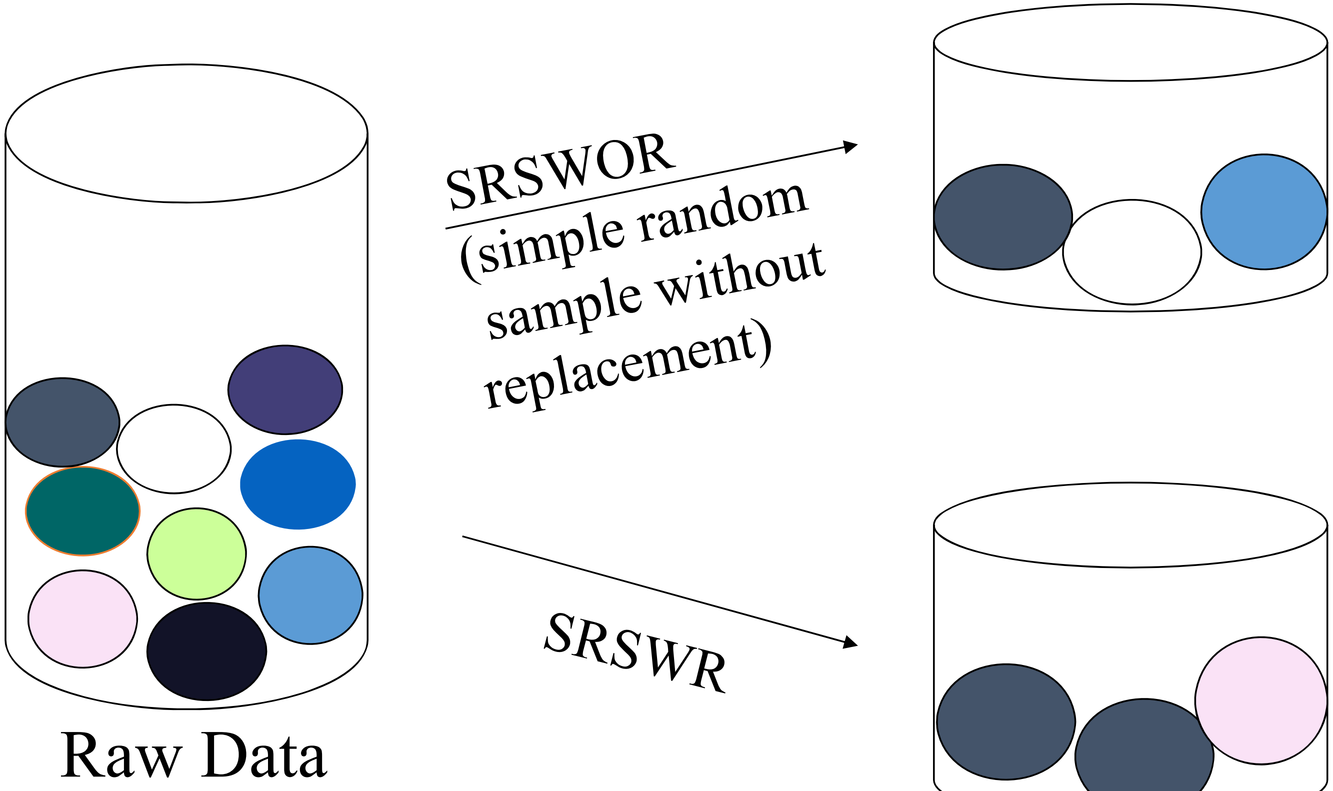
Data reduction: Attribute subset selection

1. Wrapper method (Scheme-dependent)
   - User learning method to select attributes (use heuristic search)
   - Slower than filter method
   - e.g. RFE (Recursive Feature Elimination) with SVM
2. Filter method (Scheme-independent)
   - Select attributes before learning
   1. Use a single-attribute evaluator, with ranking
      - Can eliminate <u>*irrelevant*</u> attributes
   2. Combine an attribute subset evaluator with a heuristic search method
      - Can eliminate <u>*irrelevant*</u> and <u>*redundant*</u> attributes as well
      - A subset of attributes is good if they are highly correlated with the class attribute and not strongly correlated with one another
      - Goodness of an attribute subset = $\frac{\sum_{all\ attributes\ x} Corr(x, class)}{\sqrt{\sum_{all\ attributes\ x} \sum_{all\ attributes\ y} Corr(x,y)}}$

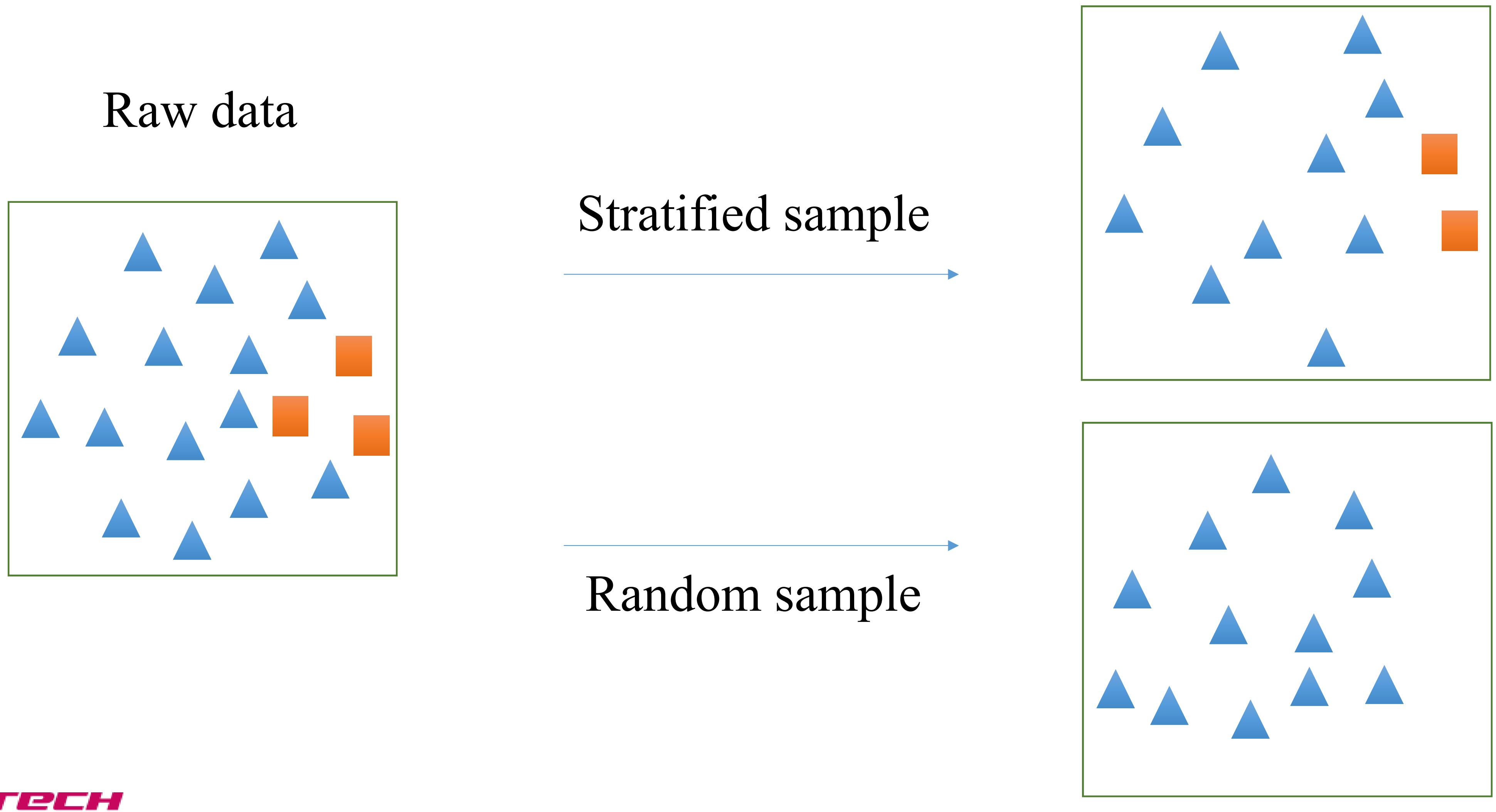
Data reduction: Sampling

- Sampling: select a *representative* subset of data
- Type of Sampling
  - **Simple random sampling**
    - There is an equal probability of selecting any particular item
  - **Sampling without replacement**
    - Once an object is selected, it is removed from the population
  - **Sampling with replacement**
    - A selected object is not removed from the population
  - **Stratified sampling:**
    - Partition the data set, and draw samples from each partition (proportionally, i.e. approximately the same percentage of the data)
    - Used in conjunction with skewed data

Data reduction: Sampling with or without replacement



Data reduction: Random sampling vs. Stratified sampling



Data transformation

- Transform the entire set of values of a given attribute to a new set of replacement
- Normalization: Scaled to fall within a smaller, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Discretization

Data transformation: Normalization

- **Min-max normalization**: to [new_minA, new_maxA]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

e.g. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0].

Then \$73,000 is mapped to $\frac{73,600-12.000}{98,000-12,000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

e.g. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600-54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

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

Data transformation: Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g. color, profession
  - Ordinal—values from an ordered set, e.g. military or academic rank
  - Numeric—real numbers, e.g. integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Prepare for further analysis, e.g. classification
  - Interval labels can then be used to replace actual data values
  - Supervised vs. unsupervised

Data transformation: Simple discretization: Binning

- Equal-width (distance) partitioning
  - 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
  - Divides the range into N intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

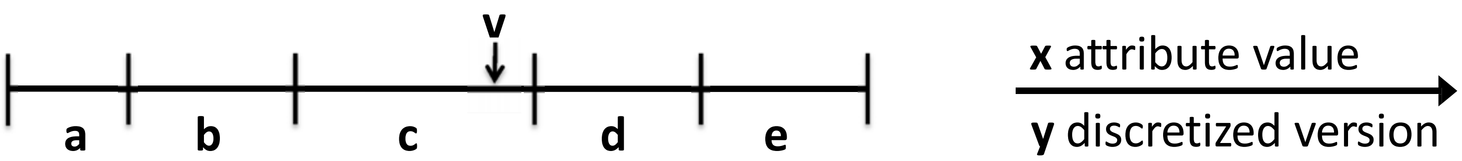
Data transformation: Simple discretization: Binning

e.g. 4, 8, 9, 15, 16, 19, 21, 22, 23, 24, 29, 34

- Equal-width (distance) partitioning
  - Bin 1: 4, 8, 9
  - Bin 2: 15, 16, 19, 21, 22, 23, 24
  - Bin 3: 29, 34
- Equal-depth (frequency) partitioning
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 16, 19, 21, 22
  - Bin 3: 23, 24, 29, 34

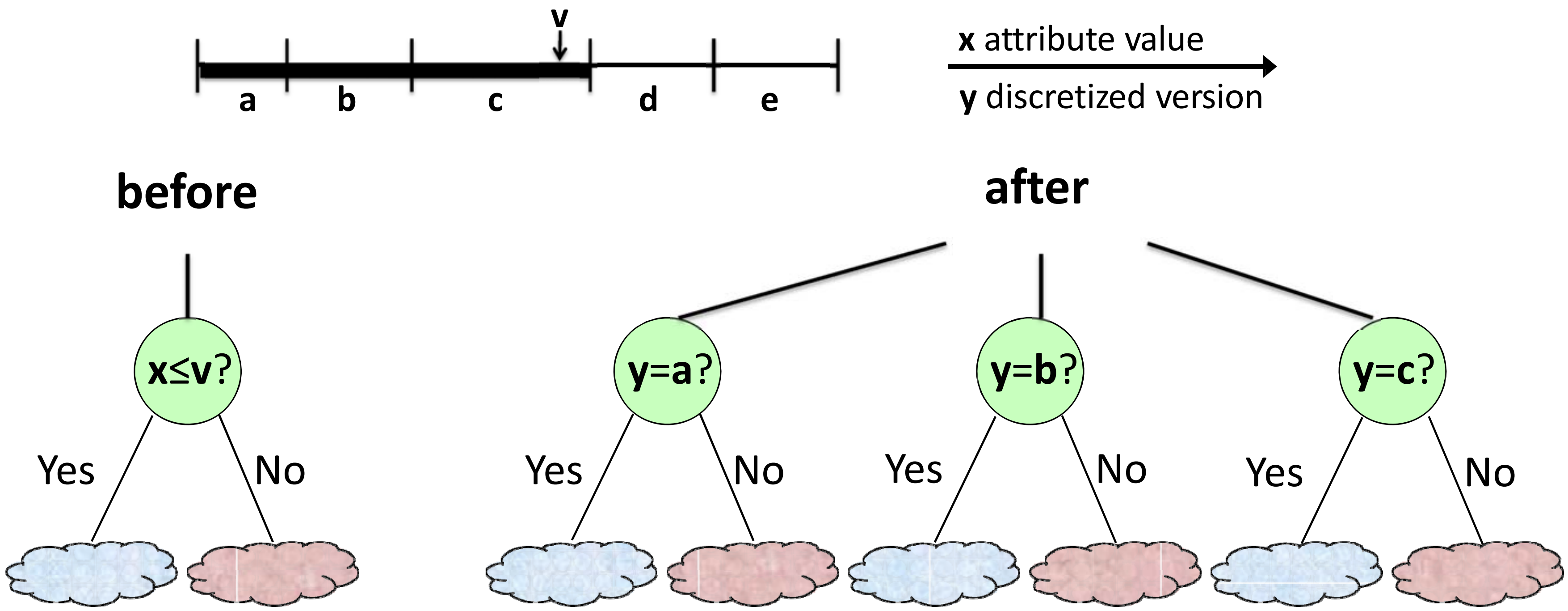
Data transformation: A numeric => Multiple binaries

How to exploit ordering information? – what's the problem?



Data transformation: A numeric => Multiple binaries

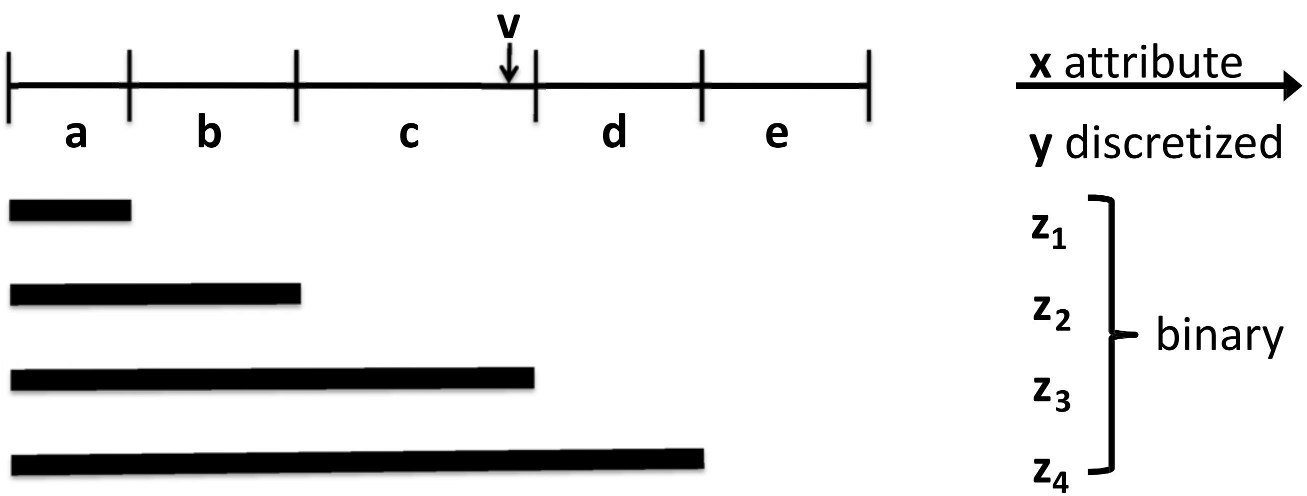
How to exploit ordering information? – what's the problem?



Data transformation: A numeric => Multiple binaries

How to exploit ordering information? – a solution

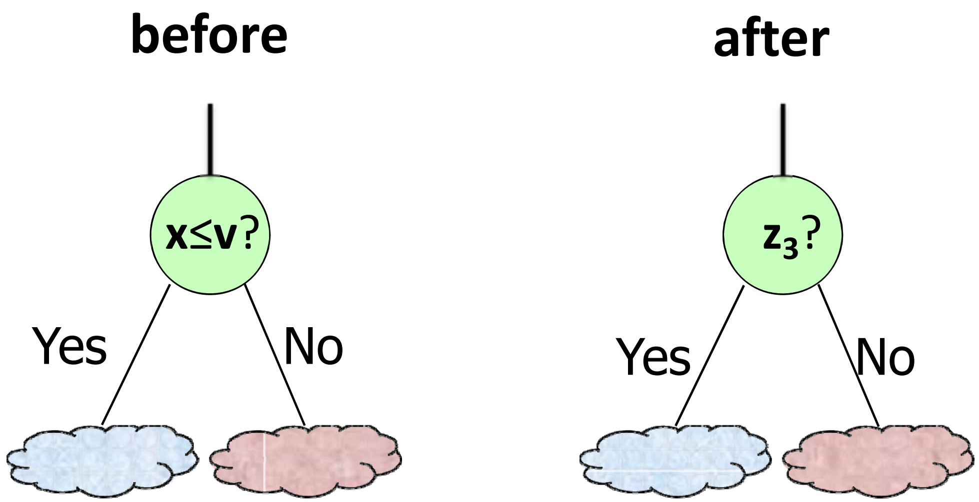
- Transform a discretized attribute with *k* values into *k–1* binary attributes
- If the original attribute's value is i for a particular instance, set the first *i-1* binary attributes to *false* and the remainder to *true*



Data transformation: A numeric => Multiple binaries

How to exploit ordering information? – a solution

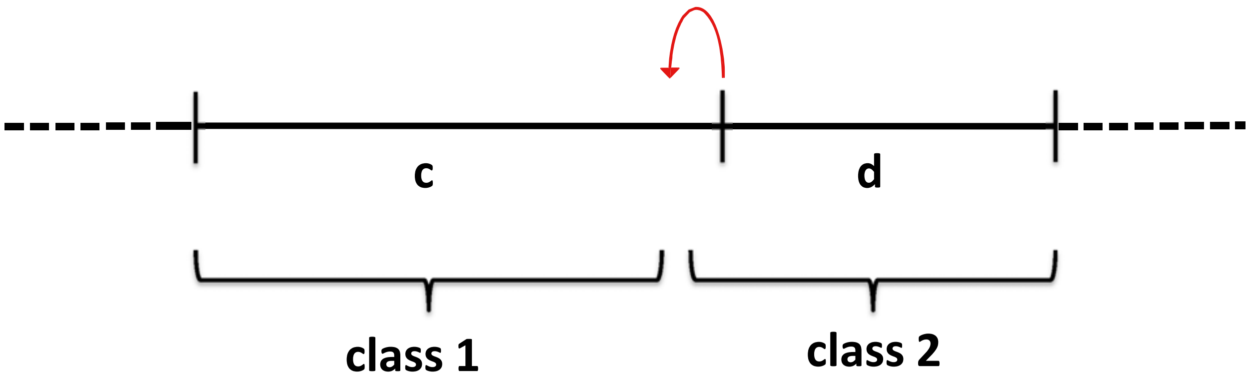
- Transform a discretized attribute with *k* values into *k–1* binary attributes
- If the original attribute's value is i for a particular instance, set the first *i-1* binary attributes to *false* and the remainder to *true*



Data transformation: Supervised discretization

Transforming numeric attributes to nominal

- What if all instances in a bin have one class, and all instances in the next higher bin have another class except for the first, which has the original class?



- Take the class values into account – supervised discretization

Data transformation: Supervised discretization

Transforming numeric attributes to nominal

- Use the entropy heuristic (pioneered by C4.5 – J48 in Weka)
- e.g. temperature attribute of weather.numeric.arff dataset

```
64    65    68    69    70  | 71    72    75    80    81    83    85
yes   no    yes   yes   yes | no    no    yes   no    yes   yes   no
                            |       yes   yes
                            |
```

4 yes, 1 no   5 yes, 4 no
entropy = 0.934 bits

amount of information required to specify the individual values of *yes* and *no* given the split

- Choose split point with smallest entropy (largest information gain)
- Repeat recursively until some stopping criterion is met

```
64    65  | 68    69    70  | 71    72  | 75    80  | 81    83  | 85
yes   no  | yes   yes   yes | no    no  | yes   no  | yes   yes | no
          |                 |       yes | yes       |           |
```

Summary in data preprocessing

1. **Data cleaning**
    - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
2. **Data integration**
    - Integration of multiple databases, data cubes, or files
    - Need to handle data redundancy (e.g. chi-square test, correlation analysis)
3. **Data reduction**
    - Dimensionality reduction
    - Numerosity reduction (Sampling)
    - Data compression
4. **Data transformation**
    - Normalization
    - Discretization or Binning