Data Mining

Lecture 3
Preprocessing Methods

http://www.informatik.uni-hamburg.de/WTM/

Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Quality: why preprocess the Data?

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

Why We should Clean Dirty Data

Major Tasks in Data Preprocessing

- ***Data cleaning***
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- ***Data integration***
  - Integration of multiple databases, data cubes, or files
- ***Data reduction***
  - Dimensionality reduction
  - Data compression
- ***Data transformation*** and ***data discretization***
  - Normalization
  - Concept hierarchy generation

Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Cleaning

- Data in the real world is "dirty" or incorrect, e.g., instrument faulty, human or computer error, transmission error
- ***Incomplete***: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - e.g., *Occupation*=" " (missing data)
- ***Noisy***: containing noise, errors, or outliers
  - e.g., *Salary*="−10" (an error)
- ***Inconsistent***: containing discrepancies in codes or names, e.g.,
  - *Age*="42", *Birthday*="03/07/2012"
  - Was rating "1, 2, 3", now rating "A, B, C"
  - discrepancy between duplicate records
- ***Intentional*** (e.g., *disguised missing* data)
  - Jan. 1 as everyone's birthday?

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

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 it in 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 or decision tree

Missing Data

- One possible interpretation of missing values – ***"don't care"*** values:

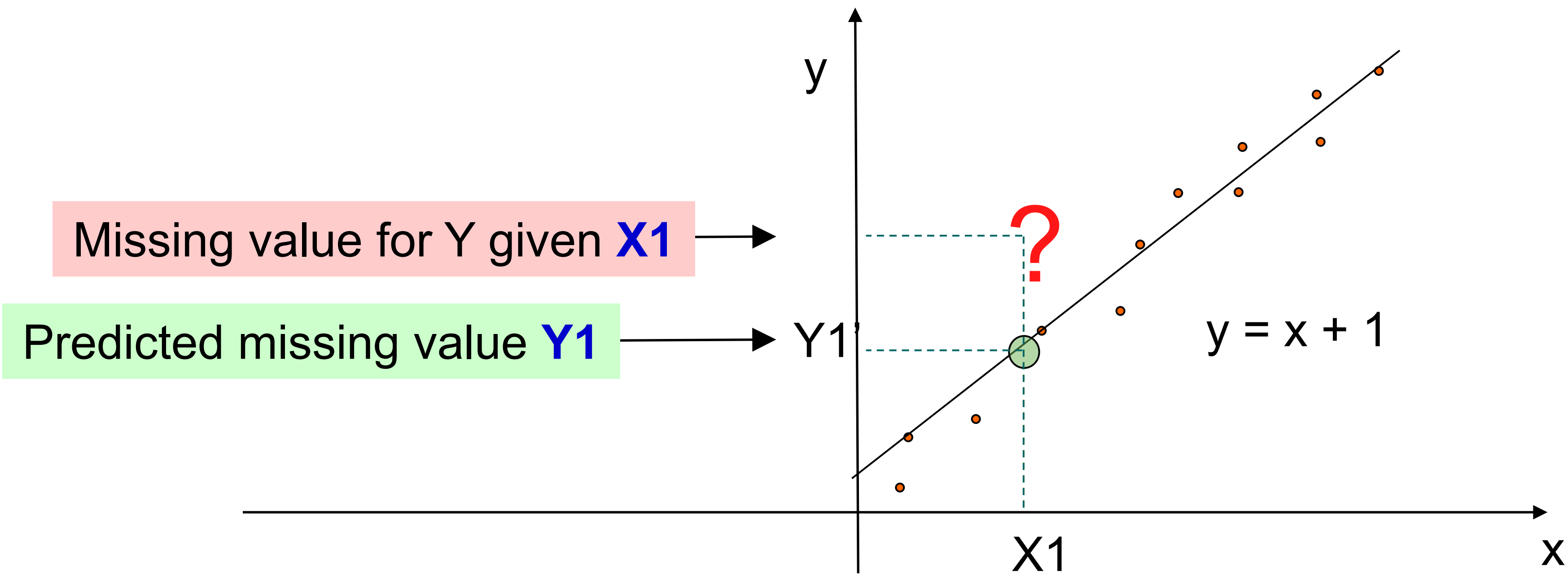
```
X = {1,?,3}
```

  → for the second feature the domain is `[0,1,2,3,4]`:

```
X1 = {1,0,3}, X2 = {1,1,3}, X3 = {1,2,3},
  X4 = {1,3,3}, X5 = {1,4,3}
```

- *Data miner* can generate model of ***correlation between features***.
  - Different techniques possible: regression, Bayesian formalism, clustering, or decision tree induction.

Missing Data Replacement with Regression Analysis



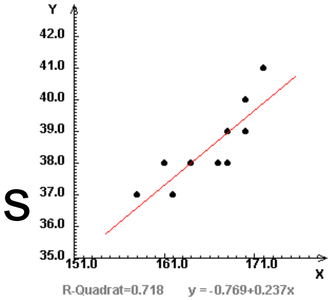
- In general, replacement of missing values is *speculative and often misleading* to replace missing values using a simple, artificial schema of data preparation.
- It is best to generate multiple solutions of data mining ***with and without features*** that have missing values, and then make comparison, analysis and interpretation.

Noisy Data

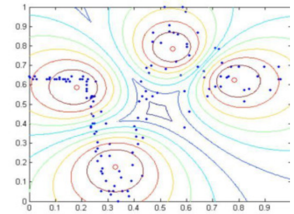
- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning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

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



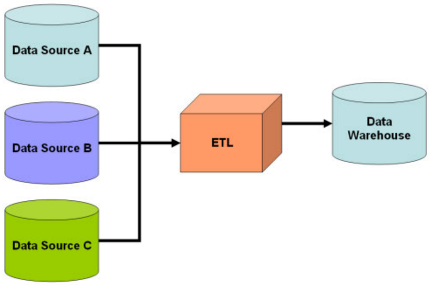
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Quality: why preprocess the Data?

Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Integration



- Data integration:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id $\equiv$ B.cust-#
  - 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

Handling Redundancy in Data Integration

- Redundant data occur often when integrating 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***
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis (nominal Data)

- ***X^2 (chi-square) test***

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

- The cells that contribute the most to the X^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

Expected Frequency

$$e_{ij} = \frac{count\ (A = a_i) \text{ x } count\ (B = b_j)}{N}$$

where N is the number of tuples and $count\ (A = a_i)$ is the number of tuples having value a_i for A

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

- X^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 prefered_reading and game_favour are correlated in the group (since X^2 larger than 10.828, from X^2 table – a statistical measure for significance of 2x2 table)

Values Reduction

ChiMerge Technique

1. ***Sort*** the data for the given feature in ascending order
2. ***Define initial intervals*** so that every value of the feature is in a separate interval
3. ***Repeat until*** no X^2 test of any two adjacent intervals is less than threshold value:

   3.1 Compute X^2 tests for each pair of adjacent intervals

   3.2 Merge two adjacent intervals with the lowest X^2 value, if calculated X^2 is less than threshold

Values Reduction – Contingency Table

- A ChiMerge requires computation of X^2 test for the contingency table 2 x 2 of categorical data:

	Class 1	Class 2	Σ
Interval-1	A_{11}	A_{12}	R_1
Interval-2	A_{21}	A_{22}	R_2
Σ	C_1	C_2	N

X^2 test is:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}}$$

where:

k = number of classes,

A_{ij} = number of instances in the i-th interval, j-th class,

E_{ij} = ***expected frequency*** of A_{ij}, which is computed as $(R_i \cdot C_j) / N$,

R_i = number of instances in the i-th interval = $\sum A_{ij}$, j = 1,…k,

C_j = number of instances in the j-th class = $\sum A_{ij}$, i = 1,2,

N = total number of instances = $\sum R_i$, i = 1,2.

Values Reduction – ChiMerge Technique Example

Data Set

Sample:	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Initial interval points

0
2
5
7.5
8.5
.
.
.

Values Reduction – ChiMerge Technique Example

- X^2 was minimum for intervals: `[7.5,8.5]` and `[8.5,10]`

Sample:	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

	Class 1	Class 2	Σ
Interval `[7.5,8.5]`	$A_{11}=1$	$A_{12}=0$	$R_1=1$
Interval `[8.5,10 ]`	$A_{21}=1$	$A_{22}=0$	$R_2=1$
Σ	$C_1=2$	$C_2=0$	$N=2$

Based on the table's values, we can calculate expected values:

$E11 = 2/2 = 1$, $E12 = 0/2 \approx 0.1$,

$E21 = 2/2 = 1$, & $E22 = 0/2 \approx 0.1$

and corresponding X^2 test:

$$X^2 = (1-1)^2/1 + (0-0.1)^2/0.1 + (1-1)^2/1 + (0-0.1)^2/0.1 = \mathbf{0.2}$$

For the degree of freedom $d=1$, and $X^2 = 0.2 < 2.706$ **(MERGE !)**

Values Reduction – ChiMerge Technique Example

- … One of the additional iterations:

Sample:	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

	Class 1	Class 2	Σ
Interval [**0.0**,**7.5**]	$A_{11}=2$	$A_{12}=1$	$R_1=3$
Interval [7.5,10]	$A_{21}=2$	$A_{22}=0$	$R_2=2$
Σ	$C_1=4$	$C_2=1$	$N=5$

E11 = 12/5 = 2.4, E12 = 3/5 = 0.6,
E21 = 8/5 = 1.6, & E22 = 2/5 = 0.4

$$X^2 = (2-2.4)^2/2.4 + (1-0.6)^2/0.6 + (2-1.6)^2/1.6 + (0-0.4)^2/0.4 = \mathbf{0.834}$$

For the degree of freedom d=1, and X^2 = **0.834 < 2.706** **(MERGE !)**

Values Reduction – ChiMerge Technique Example

- … One of the additional iterations:

Sample:	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

	Class 1	Class 2	Σ
Interval [0.0,7.5]	A_{11}=4	A_{12}=1	R_1=5
Interval [**10** ,**42**]	A_{21}=1	A_{22}=3	R_2=4
Σ	C_1=5	C_2=4	N=9

E11 = 2.78, E12 = 2.22,
E21 = 2.22, E22 = 1.78

X^2 = 2.72 > 2.706 **(NO MERGE !)**

Final discretization: [0, 10], [10, 42], and [42, 60]

↓ ↓ ↓

Interval representatives: 5 (low) 26 (medium) 51 (high)

Values Reduction – ChiMerge Technique Example

Final data set with reduced set of values for the future F:

Sample:	F	K
1	5	1
2	5	2
3	5	1
4	5	1
5	5	1
6	26	2
7	26	2
8	26	1
9	26	2
10	51	1
11	51	1
12	51	1

Correlation Analysis (numeric Data)

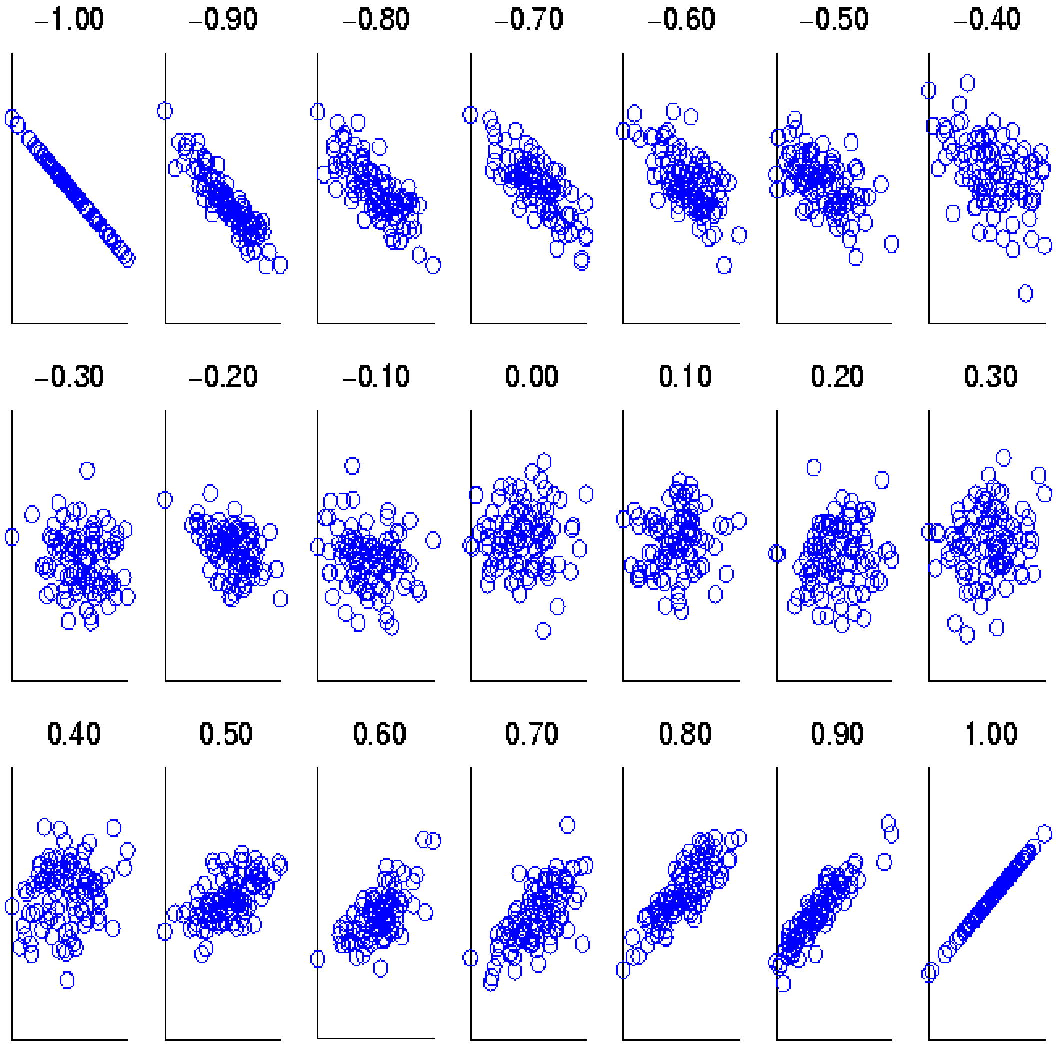
- Correlation coefficient (also called ***Pearson's product moment coefficient***)

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \overline{A})(b_i - \overline{B})}{N\sigma_A\sigma_B}$$

  where N is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of attributes A and B, σ_A and σ_B are the respective standard deviation of A and B

- 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

Visually evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

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, $\overline{A}$ and $\overline{B}$ 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 $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- ***Negative covariance***: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- ***Independence***: $Cov_{A,B} = 0$

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?
  - $E(A)$ = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - $E(B)$ = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - $Cov(A,B)$ = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since $Cov(A, B)$ > 0.

Data Reduction Strategies

- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Obtain a reduced representation of the data set that is much smaller in volume but yet produces the (almost) same analytical results
- Data reduction strategies
  - ***Dimensionality reduction***, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - ***Numerosity*** reduction (some simply call it: Data Reduction)
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
    - Data cube aggregation
  - ***Data compression***

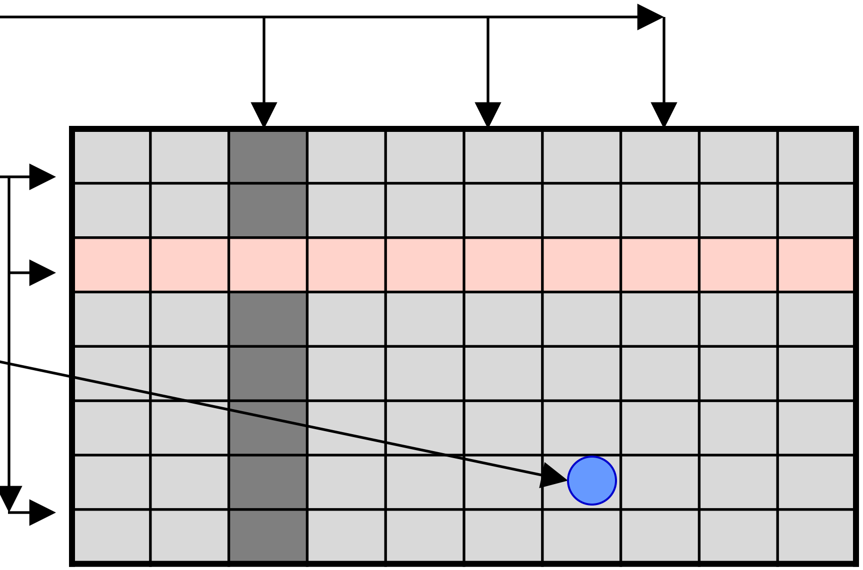
Data Reduction: Dimensionality Reduction

- ***Curse of dimensionality***
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- ***Dimensionality reduction***
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

Dimensions Reduction of Large Data Sets

Main dimensions:

- ***columns*** (features),
- ***rows*** (cases or samples),
- ***values*** of the features for the given sample



Feature Reduction

- Which features to select, and how?

TRS_DT	TRS_TYP_CD	REF_DT	REF_NUM	CO_CD	GDS_CD	QTY	UT_CD	UT_PRIC
21/05/93	00001	04/05/93	25119	10002J	001M	10	CTN	22.000
21/05/93	00001	05/05/93	25124	10002J	032J	200	DOZ	1.370
21/05/93	00001	05/05/93	25124	10002J	033Q	500	DOZ	1.000
21/05/93	00001	13/05/93	25217	10002J	024K	5	CTN	21.000
21/05/93	00001	13/05/93	25216	10026H	006C	20	CTN	69.000
21/05/93	00001	13/05/93	25216	10026H	008Q	10	CTN	114.000
21/05/93	00001	14/05/93	25232	10026H	006C	10	CTN	69.000
21/05/93	00001	14/05/93	25235	10027E	003A	5	CTN	24.000
21/05/93	00001	14/05/93	25235	10027E	001M	5	CTN	24.000
21/05/93	00001	22/04/93	24974	10035E	009F	50	CTN	118.000
21/05/93	00001	27/04/93	25033	10035E	015A	375	GRS	72.000
21/05/93	00001	20/05/93	25313	10041Q	010F	10	CTN	26.000
21/05/93	00001	12/05/93	25197	10054R	002E	25	CTN	24.000

Features Reduction

Two standard approaches:

- ***Feature selection***: A process that chooses an optimal subset of features according to an objective function:
  - feature ranking algorithms, and
  - minimum subset algorithms.

- ***Feature extraction***: refers to the mapping of the original high-dimensional data onto a lower-dimensional space. Criterion for :
  - Descriptive setting: minimize the information loss
  - Predictive setting: maximize the class discrimination

Feature selection – Example for Optimal Features' Subset

F_1	F_2	F_3	F_4	F_5	C
0	0	1	0	1	0
0	1	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1

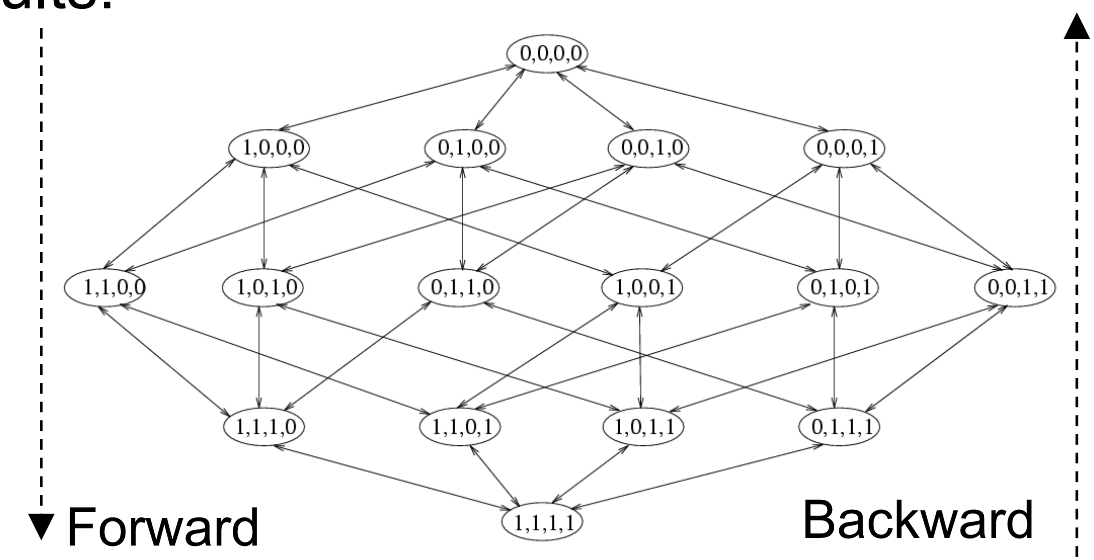
- Data set (whole set)
  - Five Boolean features
  - $C = F_1 \vee F_2$
  - $F_3 = \neg F_2$, $F_5 = \neg F_4$
  - Optimal subset:

    $\{F_1, F_2\}$ or $\{F_1, F_3\}$
- Combinatorial nature of searching for an optimal subset

Feature Selection – Complexity

- **Feature selection** in general can be viewed as a search problem (2^N).

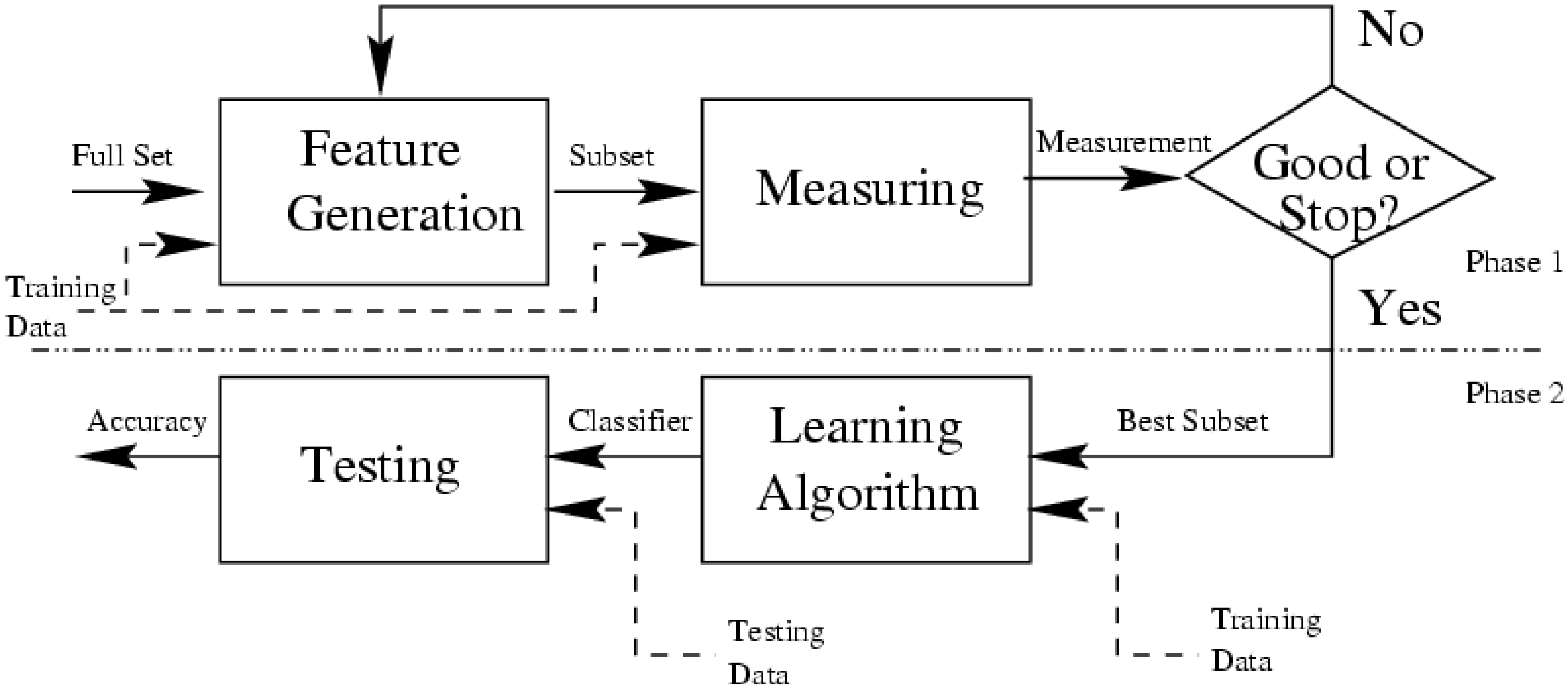
- For practical methods, an optimal search is not feasible, and simplifications are made to produce acceptable and timely reasonable results:
  - heuristic criteria
  - bottom-up approach
  - top-down approach



Methods of Feature Selection

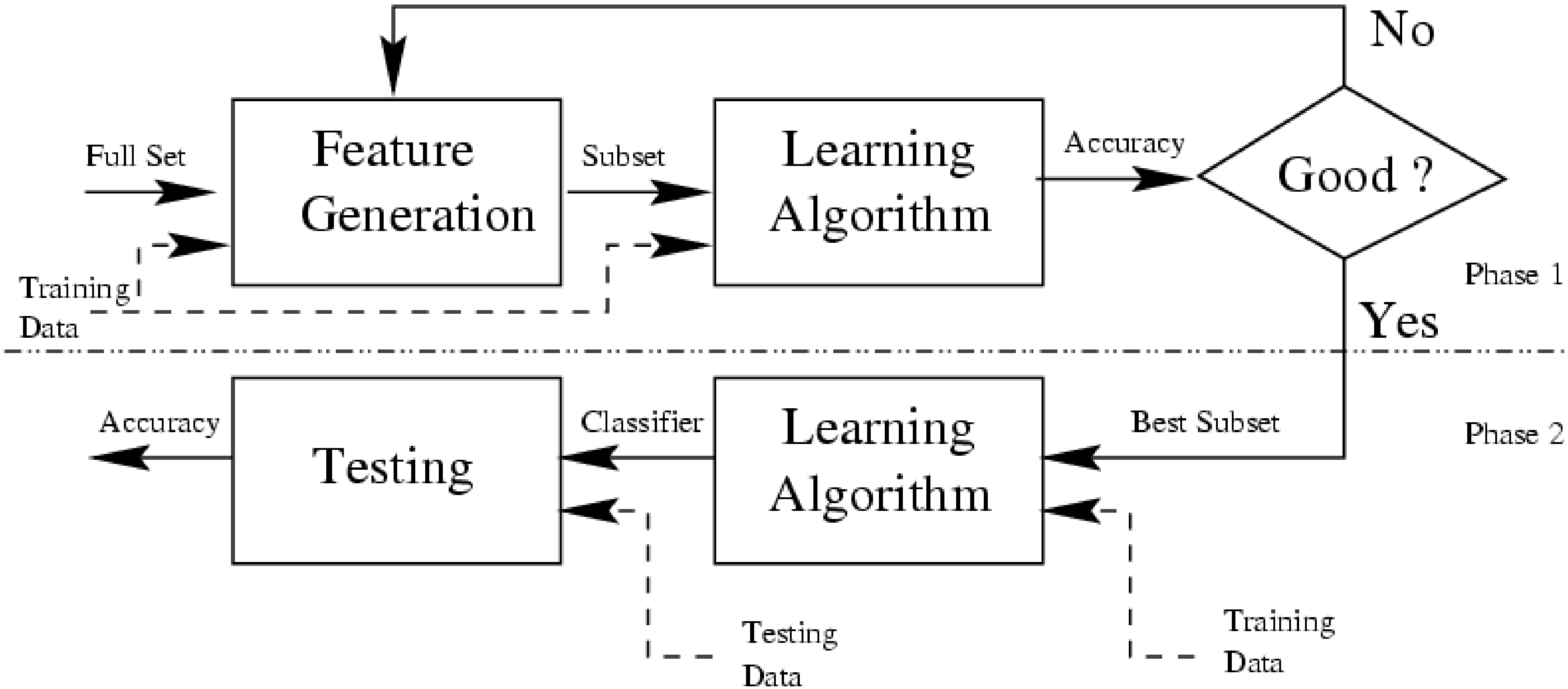
- Univariate methods
  - Considers one variable (feature) at a time.
- Filter methods
  - Separating feature selection from classifier learning
  - Relying on general characteristics of data (information, distance, dependence, consistency)
  - No bias toward any learning algorithm, fast
- Wrapper methods
  - Relying on a predetermined classification algorithm.
  - Using predictive accuracy as goodness measure
  - High accuracy, computationally expensive
- Embedded methods
  - Combine Filter and Wrapper approaches

Filter Model



- Example filter algorithm for Feature Selection:
  - ***Relief*** (Kira & Rendell 1992)

Wrapper Model



- Example wrapper algorithm for Feature Selection:
  - ***SVM***

Features Selection: Univariate Methods

Comparison of means and variances:

- Samples of two classes (A and B) can be examined:

$$\mathrm{SE}(A-B) = \sqrt{\left(\frac{\mathrm{var}(A)}{n_1} + \frac{\mathrm{var}(B)}{n_2}\right)}$$

- TEST:

$$\frac{\left|\mathrm{mean}(A) - \mathrm{mean}(B)\right|}{\mathrm{SE}(A-B)} > \textit{threshold-value}$$

where n_1 and n_2 are the corresponding number of samples for classes A and B.

Features Selection: Univariate Methods

- Comparison of *means* and *variances* – **Example**:

X	Y	C
0.3	0.7	A
0.2	0.9	B
0.6	0.6	A
0.5	0.5	A
0.7	0.7	B
0.4	0.9	B

Threshold value is 0.5

$X_A = \{0.3, 0.6, 0.5\}$, $X_B = \{0.2, 0.7, 0.4\}$,

$Y_A = \{0.7, 0.6, 0.5\}$, and $Y_B = \{0.9, 0.7, 0.9\}$

Features Selection: Univariate Methods

- Comparison of *means* and *variances* – **Example**:

$$\mathrm{SE}(X_A - X_B) = \sqrt{\left(\frac{\mathrm{var}(X_A)}{n_1} + \frac{\mathrm{var}(X_B)}{n_2}\right)} = \sqrt{\frac{0.0233}{3} + \frac{0.6333}{3}} = 0.4678$$

$$\mathrm{SE}(Y_A - Y_B) = \sqrt{\left(\frac{\mathrm{var}(Y_A)}{n_1} + \frac{\mathrm{var}(Y_B)}{n_2}\right)} = \sqrt{\frac{0.01}{3} + \frac{0.0133}{3}} = 0.0875$$

Tests:

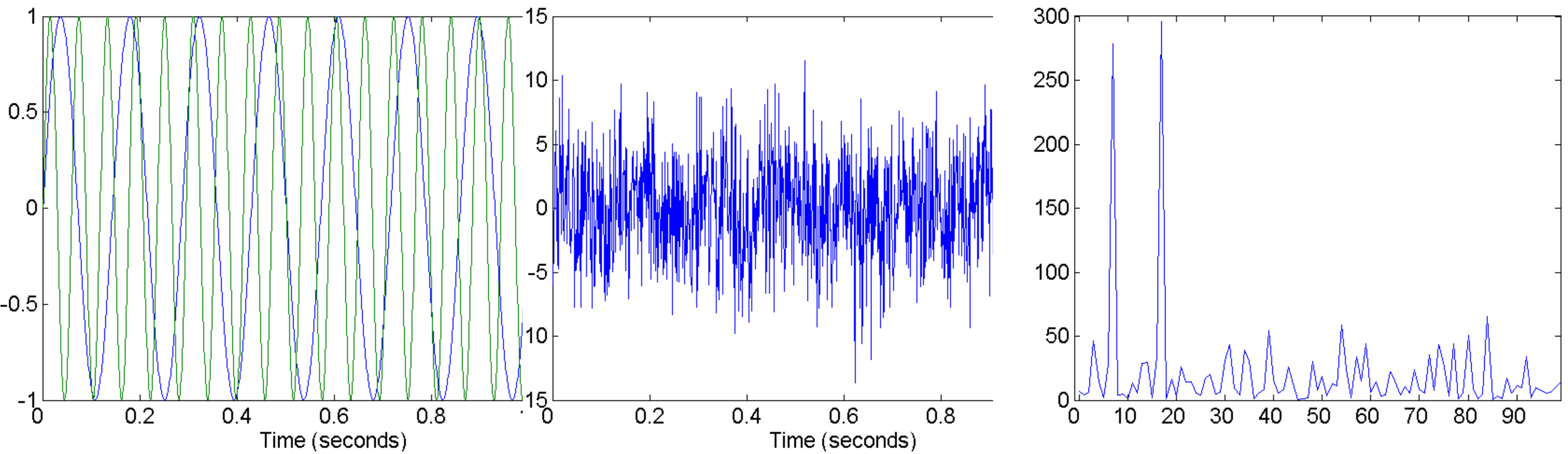
$$\frac{|\mathrm{mean}(A) - \mathrm{mean}(B)|}{\mathrm{SE}(A-B)} = \frac{|0.4667 - 0.4333|}{0.4678} < 0.5$$

$$\frac{|\mathrm{mean}(A) - \mathrm{mean}(B)|}{\mathrm{SE}(A-B)} = \frac{|0.6 - 0.8333|}{0.0875} > 0.5$$

X is a candidate feature for reduction because its mean values are close, and therefore the final test is below threshold value.

Mapping Data to a New Space

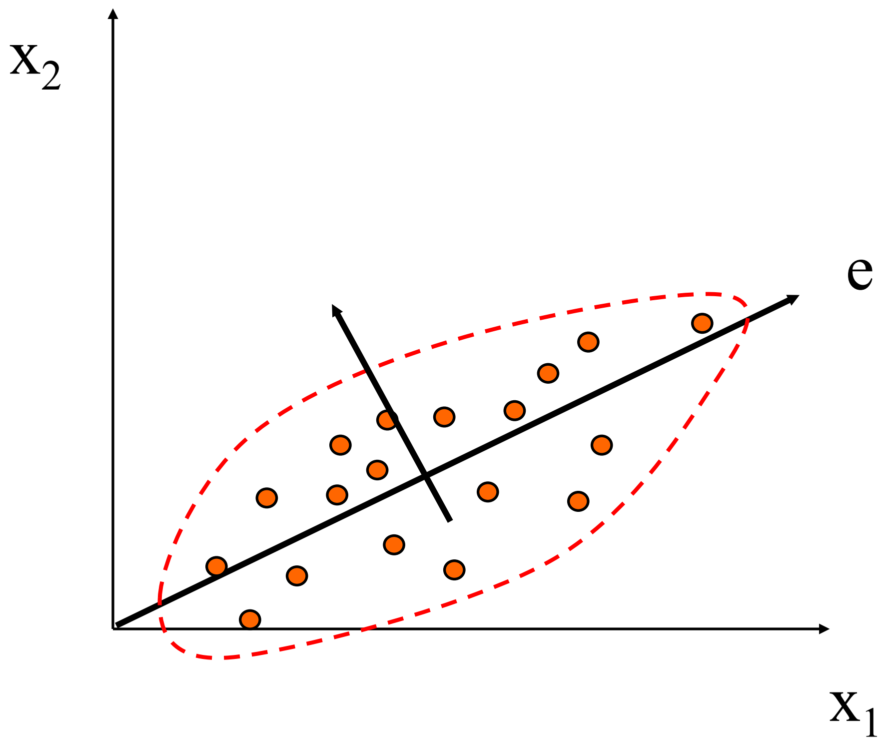
- Fourier transform: mapping from time to frequency domain
- Wavelet transform



Two Sine Waves **Two Sine Waves + Noise** **Frequency**

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



Principal Component Analysis (Steps)

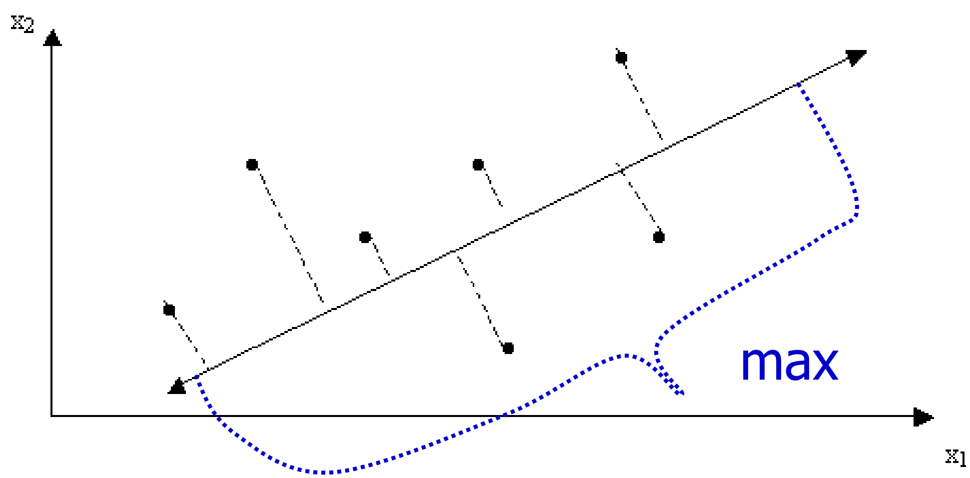
- Given N data vectors from n-dimensions, find $k \leq n$ orthogonal vectors (***principal components***) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute k orthonormal (unit) vectors, i.e., *principal components*
  - Each input 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; reduction of higher dimensions to lower

Principal Components Analysis

- The features are examined collectively, merged and transformed into a new set of features that hopefully retain the original information content in a reduced form.
- Given m features, they can be transformed into a single new feature F', by the simple application of weights w:

$$F' = \sum_{j=1}^{m} w(j) \cdot f(j)$$

The first principal component is an axis in the direction of maximum variance.



Principal Components Analysis

- Most likely a single set of weights ***w(j)*** will not be adequate transformation.
- Up to m transformations are generated, where each vector of m weights is called a ***principal component*** and it generate a new feature.
- Eliminating the bottom ranked transformation will cause dimensions reduction.

Principal Components Analysis Algorithm

- We use ***covariance matrix*** S computation, as a first step in features transformation.

$$S_{n\times n} = \frac{1}{n-1}\cdot\sum_{j=1}^{n}\left(x_j - x'\right)^T\cdot\left(x_j - x'\right) \qquad \text{where} \qquad x' = \frac{1}{n-1}\cdot\sum_{j=1}^{n} x_j$$

- The ***eigenvalues*** of the covariance matrix S for the given data should be calculated in the next step and the eigenvalues of $S_{n\times n}$ are sorted: $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$.
- The ***eigenvectors*** $e_1, e_2, \ldots, e_n$ correspond to eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, and they are called the ***principal axes***.
- The criterion for features selection is based on the ratio R of the sum of the m largest eigenvalues of S to the trace of S (for example R>90%):

$$R = \sum_{i=1}^{m}\lambda_i \Big/ \sum_{i=1}^{n}\lambda_i$$

Principal Components Analysis – IRIS Data

	Feature 1	Feature 2	Feature 3	Feature 4
Feature 1	1.0000	-0.1094	0.8718	0.8180
Feature 2	-0.1094	1.0000	-0.4205	-0.3565
Feature 3	0.8718	-0.4205	1.0000	0.9628
Feature 4	0.8180	-0.3565	0.9628	1.0000

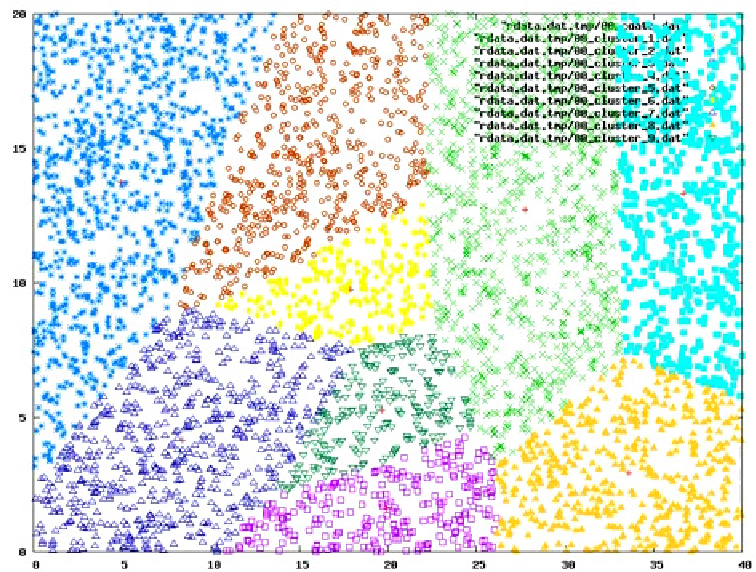
The correlation matrix for Iris data

Features	Eigenvalues
Feature 1 *	2.91082
Feature 2 *	0.92122
Feature 3 *	0.14735
Feature 4 *	0.02061

The eigenvalues for Iris data

Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms



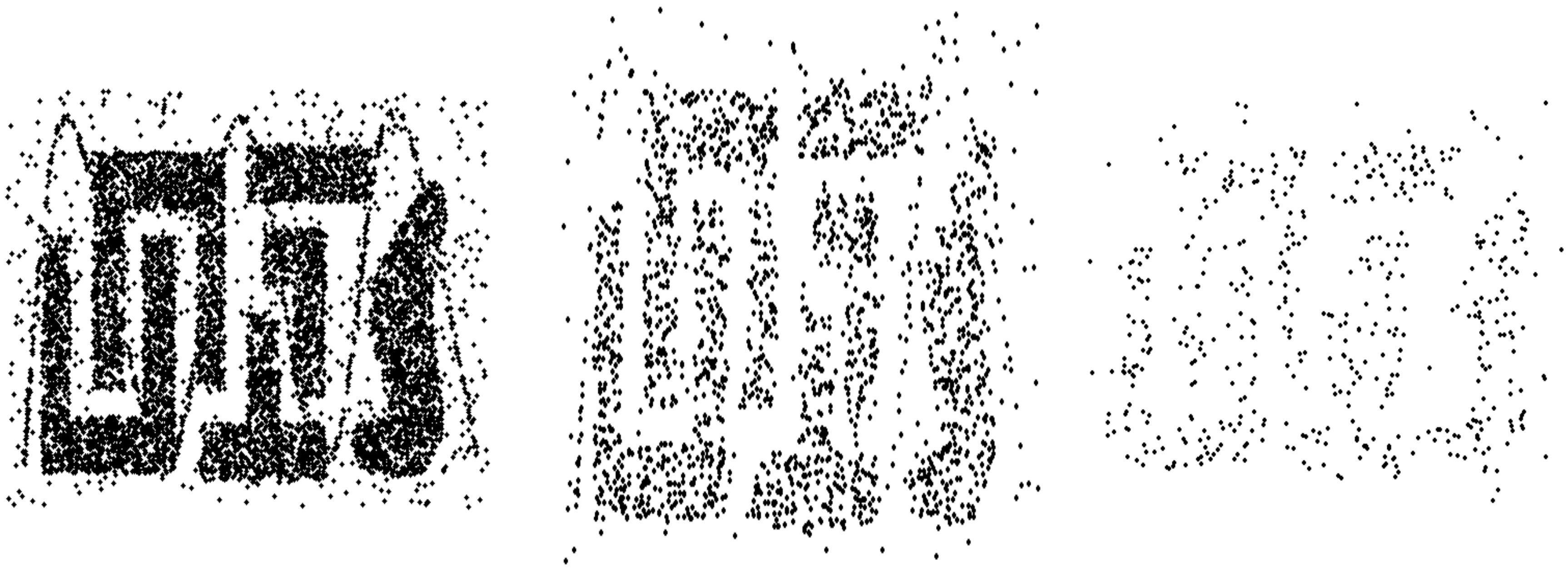
Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a ***representative*** subset of the data
  - Simple random sampling may have very poor performance in the presence of skewed data
  - Develop adaptive sampling methods, e.g., stratified sampling:

Types 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

Cases Reduction: Sample Size



8000 points **2000 Points** **500 Points**

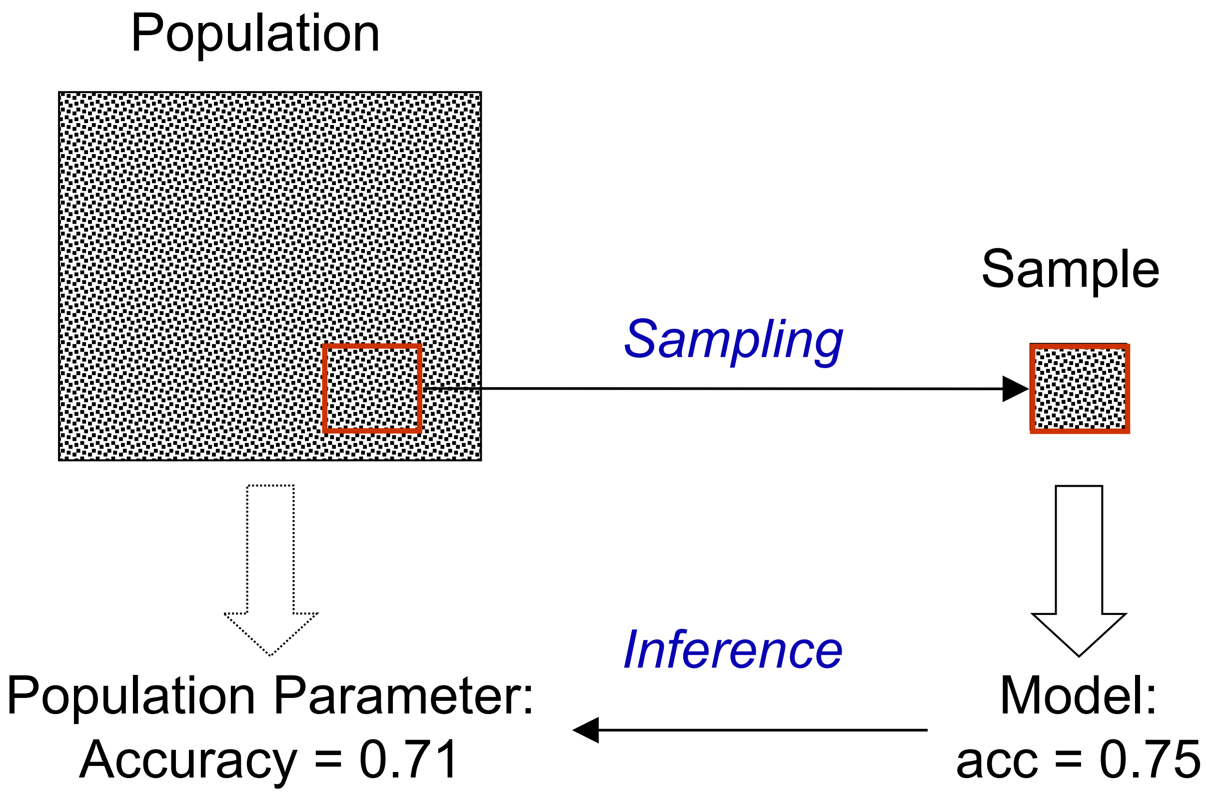
Cases Reduction: Sampling …

Key principle for effective sampling:

- Using a sample will work almost as well as using the entire data sets, if the sample is ***representative***.
- A sample is representative if it has approximately the same property (of interest) as the original set of data.

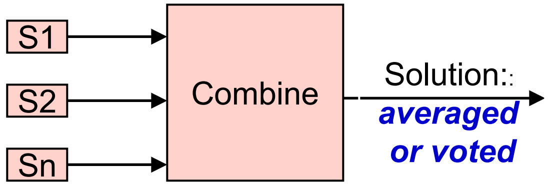
Cases Reduction: Accuracy Parameter Estimation

- ***Challenging task***: Infer the value of a population parameter based on a sample model.



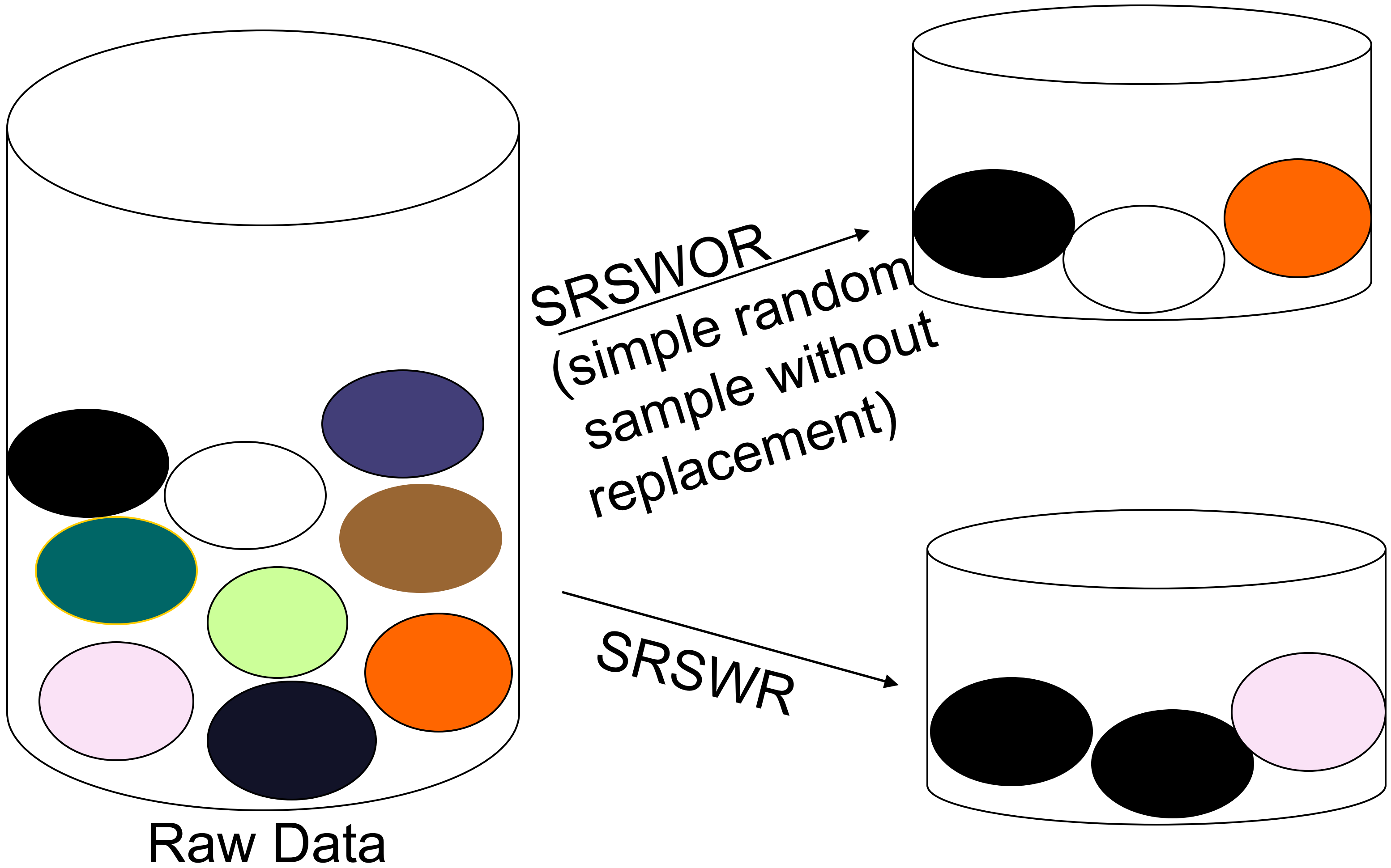
Cases Reduction: General-purpose sampling methods

- ***Systematic sampling***:
  - Simplest
  - For example 50% of a data set (every second sample)
  - Built in most of Data Mining tools
  - Problem: regularities in data set!
- ***Random sampling***
  - Random sampling without replacement,
  - Random sampling with replacement.
  - Average sampling: Combined solution from several subsets (randomly selected).

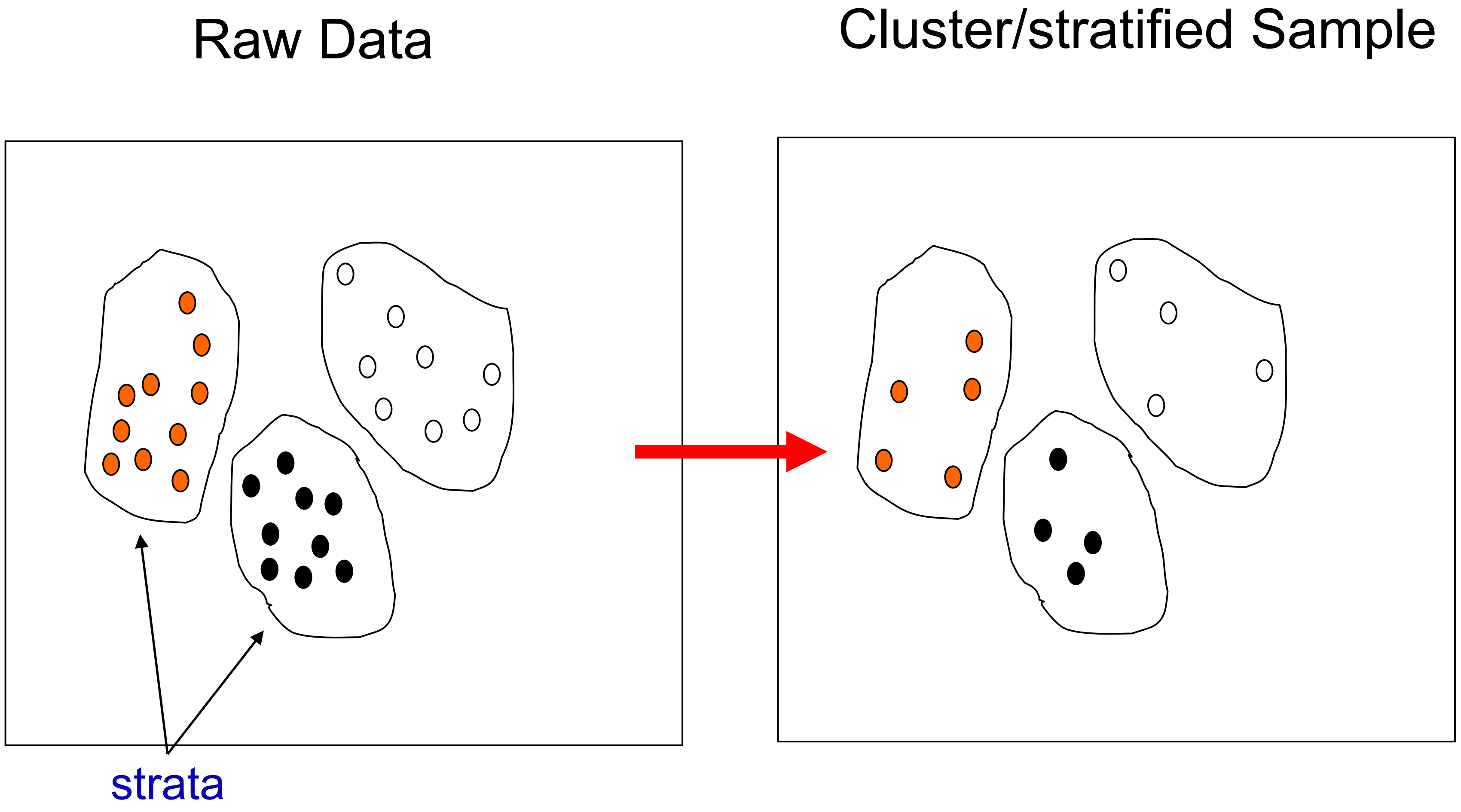


  - Stratified sampling:
    - Split data set into non-overlapping ⇨ subsets = strata.
    - Combine strata results.

Sampling: with or without Replacement



Sampling: Cluster or stratified Sampling



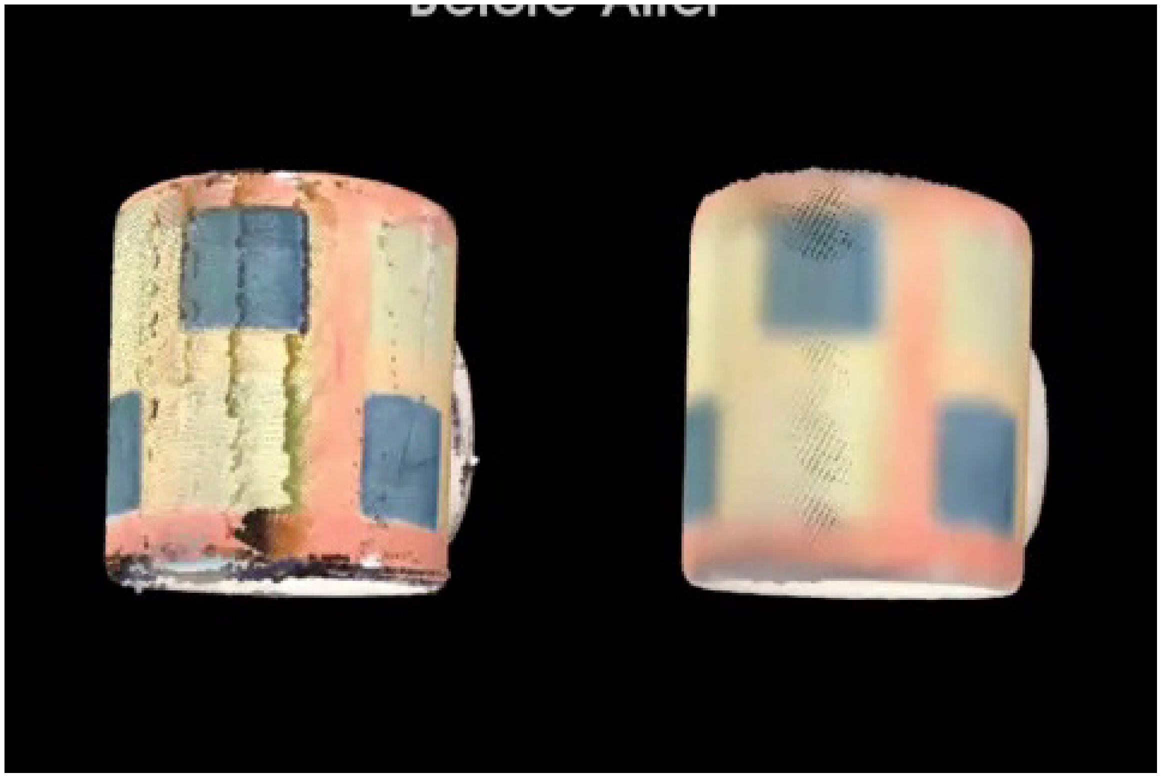
Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values so that each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization: Concept hierarchy climbing

Example: Data Resampling and Smoothing in Point Cloud Application



Normalization

- ***Min-max normalization***: to [new_min$_A$, new_max$_A$]

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

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,600 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):

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

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- ***Normalization by decimal scaling***

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

Transformation of Raw Data

- ***Data smoothing***

  F = {0.93, 1.01, 1.001, 3.02, 2.99, 5.03, 5.01, 4.98},

  F smoothed = {1.0, 1.0, 1.0, 3.0, 3.0, 5.0, 5.0, 5.0}.

- ***Differences and ratios***

  $s(t+1)-s(t)$ $\qquad$ $s(t+1)/s(t)$

- ***Composing new features***

  For example:

  Body mass index **BMI= k F(Weight, Height)**

Time-dependent Data

- The ***time series*** of values can be expressed as a list:

  X = $\{t(1), t(2), t(3), \ldots, t(n)\}$,

  where $t(n)$ is the most recent value.

- For many problems based on time series the goal is to:
  - ***forecast*** $t(n+1)$ from previous n values of the feature (or more general forecast $t(n+j)$), where these values are directly related to the predicted value, or
  - ***find patterns*** in time series.

- The most important step in preprocessing of row time-dependent data is specification of a window or a time lag

Time-dependent Data

- For example, if the time series consists of eleven measurements:

X = {t(0), t(1), t(2), t(3), t(4), t(5), t(6), t(7), t(8), t(9), t(10)}

- **1.)**
  - window size: **w=5,**
  - next value: **j=1**

Sample	WINDOW					Next Value
	M1	M2	M3	M4	M5	
1	t(0)	t(1)	t(2)	t(3)	t(4)	t(5)
2	t(1)	t(2)	t(3)	t(4)	t(5)	t(6)
3	t(2)	t(3)	t(4)	t(5)	t(6)	t(7)
4	t(3)	t(4)	t(5)	t(6)	t(7)	t(8)
5	t(4)	t(5)	t(6)	t(7)	t(8)	t(9)
6	t(5)	t(6)	t(7)	t(8)	t(9)	t(10)

Time-dependent Data

- For example, if the time series consists of eleven measurements:

X = {t(0), t(1), t(2), t(3), t(4), t(5), t(6), t(7), t(8), t(9), t(10)}

- **2.)**
  - window size: **w=5,**
  - next value: **j=3**

Sample	WINDOW					Next Value
	M1	M2	M3	M4	M5	
1	t(0)	t(1)	t(2)	t(3)	t(4)	t(7)
2	t(1)	t(2)	t(3)	t(4)	t(5)	t(8)
3	t(2)	t(3)	t(4)	t(5)	t(6)	t(9)
4	t(3)	t(4)	t(5)	t(6)	t(7)	t(10)

Time-dependent Data

Time-dependent ***2D*** data

Time	a	b
1	5	117
2	8	113
3	4	116
4	9	118
5	10	119
6	12	120

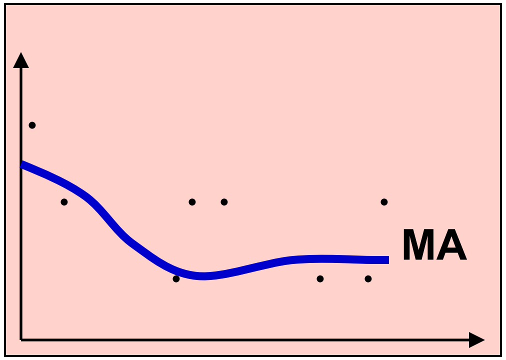
Samples prepared for ***window*** w = 3

Sample	a (n-2)	a (n-1)	a(n)	b (n-2)	b (n-1)	b(n)
1	5	8	4	117	113	116
2	8	4	9	113	116	118
3	4	9	8	116	118	119
4	9	10	12	118	119	120

Time-dependent Data

- One way of ***summarizing*** features in the data set is to average them producing so called "***moving averages***" (MA):

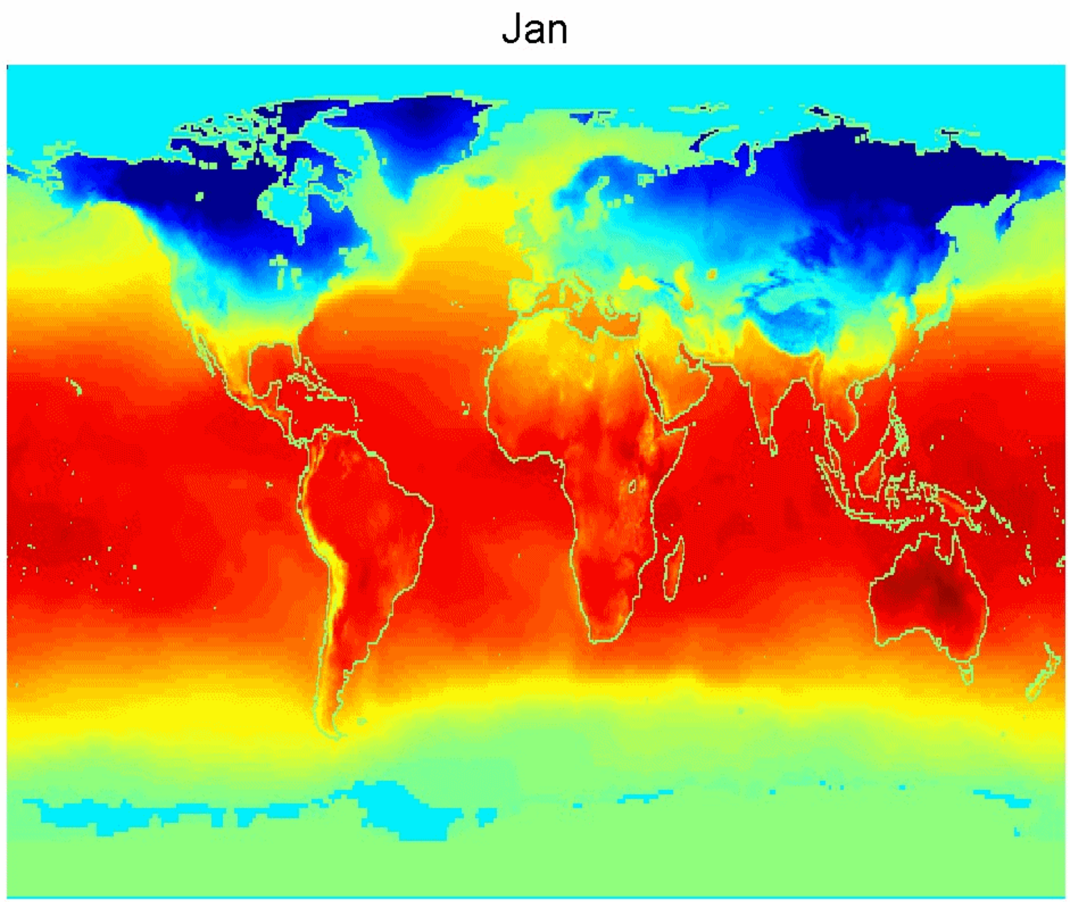
$$\mathrm{MA}(i,m) = \frac{1}{m} \cdot \sum_{j=i-m+1}^{i} t(j)$$



- The objective is to ***smooth*** neighboring time points by a moving average to reduce the random variation and noise components:

$$\mathrm{MA}(i,m) = t(i) = \mathrm{mean}(i) + error$$

Spatial-Temporal Data



Average Monthly Temperature of land and ocean

New disciplines: Temporal, Spatial, and Streaming Data Mining

Data Discretization Methods

- Reduce number of values for given continuous attribute by dividing into intervals
  - ***Binning***: equal width binning and replacing bin by mean
    - Top-down split, unsupervised, no class information used
  - ***Histogram analysis***
    - Top-down split, unsupervised, no class information used
  - ***Clustering analysis*** (unsupervised, top-down split or bottom-up merge)
  - ***Decision-tree analysis*** (supervised, top-down split)
  - ***Correlation*** (e.g., χ^2) ***analysis*** (unsupervised, bottom-up merge)

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

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 equal-frequency (***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***:

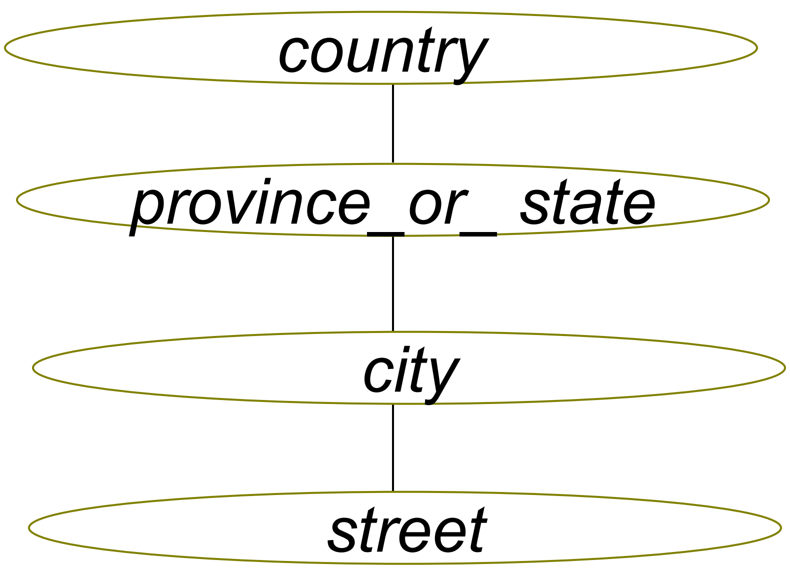
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

Smoothing by ***bin boundaries***:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 21, 25
- Bin 3: 26, 26, 26, 34

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year



15 distinct values

365 distinct values

3567 distinct values

674,339 distinct values

Noise example in real world from the WTM lab www.informatik.uni-hamburg.de/WTM or www.knowledge-technology.info

Summary

- ***Data quality***: accuracy, completeness, consistency, timeliness, believability, interpretability
- ***Data cleaning***: e.g. missing/noisy values, outliers
- ***Data integration*** from multiple sources:
  - Entity identification problem
  - Remove redundancies
  - Detect inconsistencies
- ***Data reduction***
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- ***Data transformation*** and ***data discretization***
  - Normalization
  - Concept hierarchy generation