Università degli Studi di Milano
Master Degree in Computer Science

Information Management course

Teacher: Alberto Ceselli

Lecture 03 : 09/10/2013

Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 3 —

Jiawei Han, Micheline Kamber, and Jian Pei

University of Illinois at Urbana-Champaign &

Simon Fraser University

Chapter 3: 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

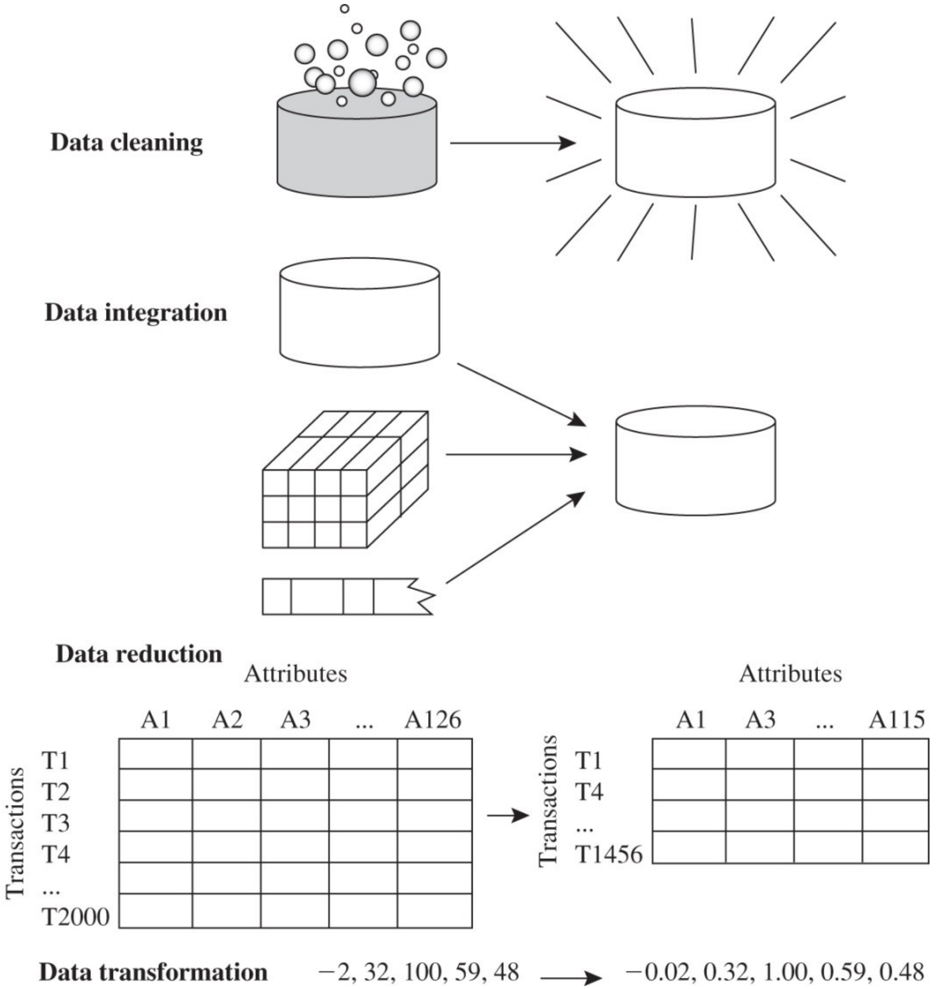
Data Quality: Why Preprocess the Data?

- 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 the data are correct?
  - Interpretability: how easily the data can be understood?

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
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

Major Tasks in Data Preprocessing



Chapter 3: 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 (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*=“03/07/2010”
    - 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., 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

How to Handle Missing Data?

- Ignore the tuple (e.g. when class label is missing and doing classification) → simple, but loss of data
- Fill in the missing value manually → tedious + infeasible?
- Fill in it automatically with
  - global const (e.g., “unknown”) → a new class?!
  - the attribute mean or median
  - 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

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 Cleaning as a Process

- Data discrepancy detection
  - Use knowledge about data → use **metadata** (e.g., domain, range, dependency, distribution) i.e. **know your data!**
  - Check field overloading
  - Check uniqueness rule, consecutive rule and null rule
  - Use commercial tools
    - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
    - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers) → already "data mining"
- Data migration and integration
  - Data migration tools: allow transformations to be specified
  - ETL (Extraction/Transformation/Loading) tools (GUI)
- Integration of the two processes
  - Iterative and interactive (e.g., Potter's Wheels)

Chapter 3: 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 ≡ 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 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*

Correlation Analysis (Nominal Data)

- **X^2 (chi-square) test**
  - Attribute A has c values (a_1 … a_c)
  - Attribute B has r values (b_1 … b_r)
  - Build a contingency table [o_{ij}], having 1 row for each a_i, one col for each b_j
  - o_{ij} is the observed frequency (number of tuples having value ai for A and bj for B)

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{num.\ data\ tuples}$$

$$\chi^2 = \sum_i \sum_j \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

Correlation Analysis (Nominal Data)

- The larger the X^2 value, the more likely the variables are related
- 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

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

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 e_{ij})

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

- 2x2 table = 1 degree of freedom
- From chi-square distribution, the value for rejecting hypotesis of independency at 0.001 significance level is 10.828 → **strong correlation**

Deg. freedom											
1	0.00	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.86	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
p-val	0.95	0.9	0.8	0.7	0.5	0.3	0.2	0.1	0.05	0.01	0.001

1 – Cum. Distr. Funct. = significance level

Covariance (Numeric Data)

- Covariance:
  - Attributes A and B
  - n → number of tuples
  - $\overline{A}$ and $\overline{B}$ → respective means of A and B
  - σ_A and σ_B → the respective standard deviation of A and B

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

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

Covariance (Numeric Data)

- Covariance:

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

- **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**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) a covariance of 0 does imply independence

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

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

- 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.

Correlation Analysis (Numeric Data)

- Correlation coefficient (also called Pearson's product moment coefficient)
  - Attributes A and B
  - n → number of tuples
  - $\overline{A}$ and $\overline{B}$ → respective means of A and B
  - σ_A and σ_B → the respective standard deviation of A and B

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

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

Correlation Analysis (Numeric Data)

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

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

- 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

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

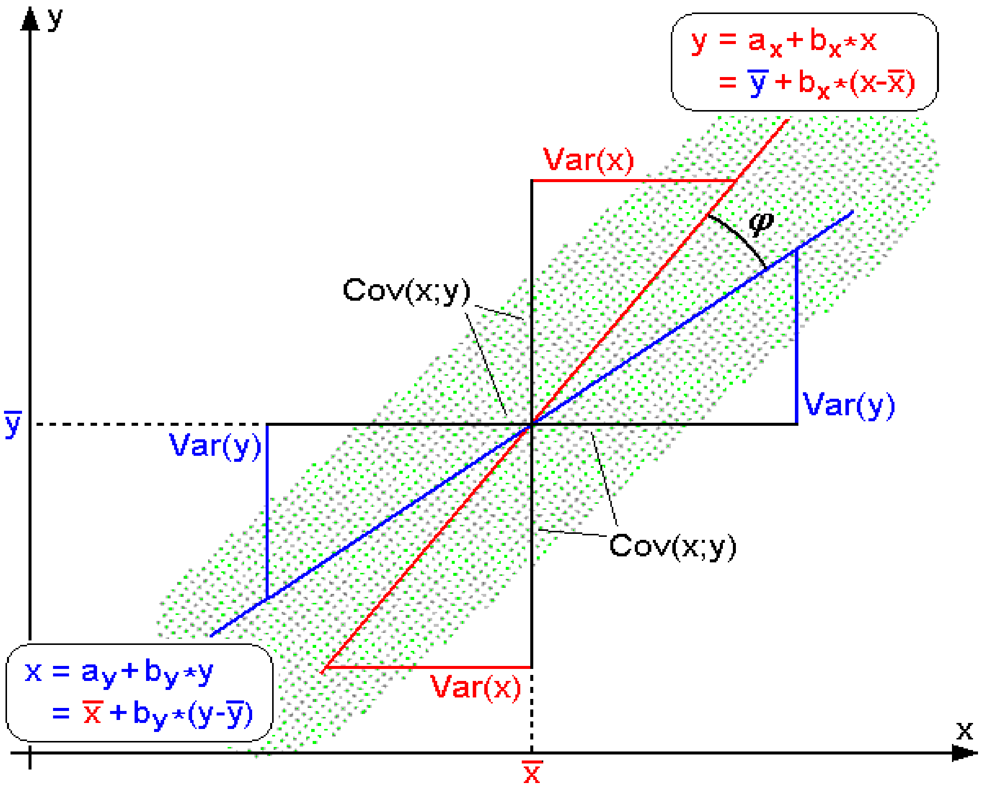
$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

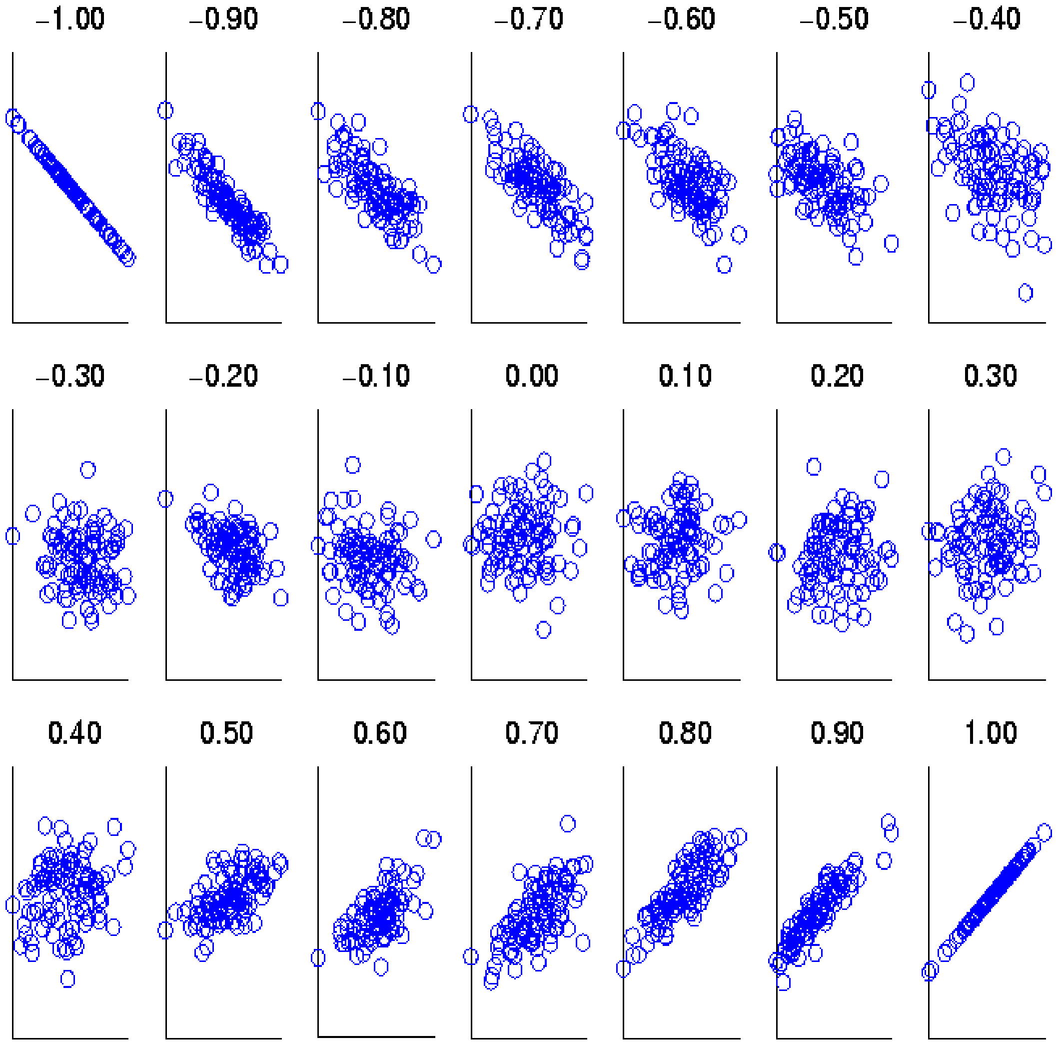
$$correlation(A, B) = A' \bullet B'$$

Correlation Analysis (Numeric Data)

- Geometrically: the cosine of the angle between the two vectors, after centering (or possible regression lines)



Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Chapter 3: 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 Reduction Strategies

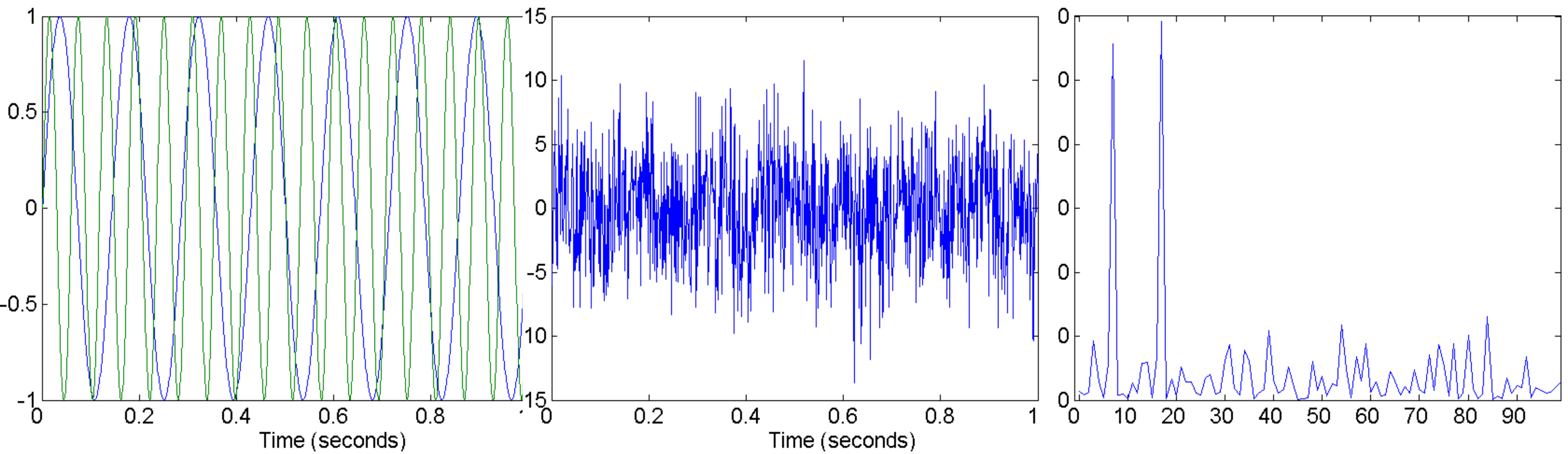
- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? Computational issues in big data!
- Data reduction strategies
  - Dimensionality reduction, e.g., remove unimportant 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 1: 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
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

Mapping Data to a New Space

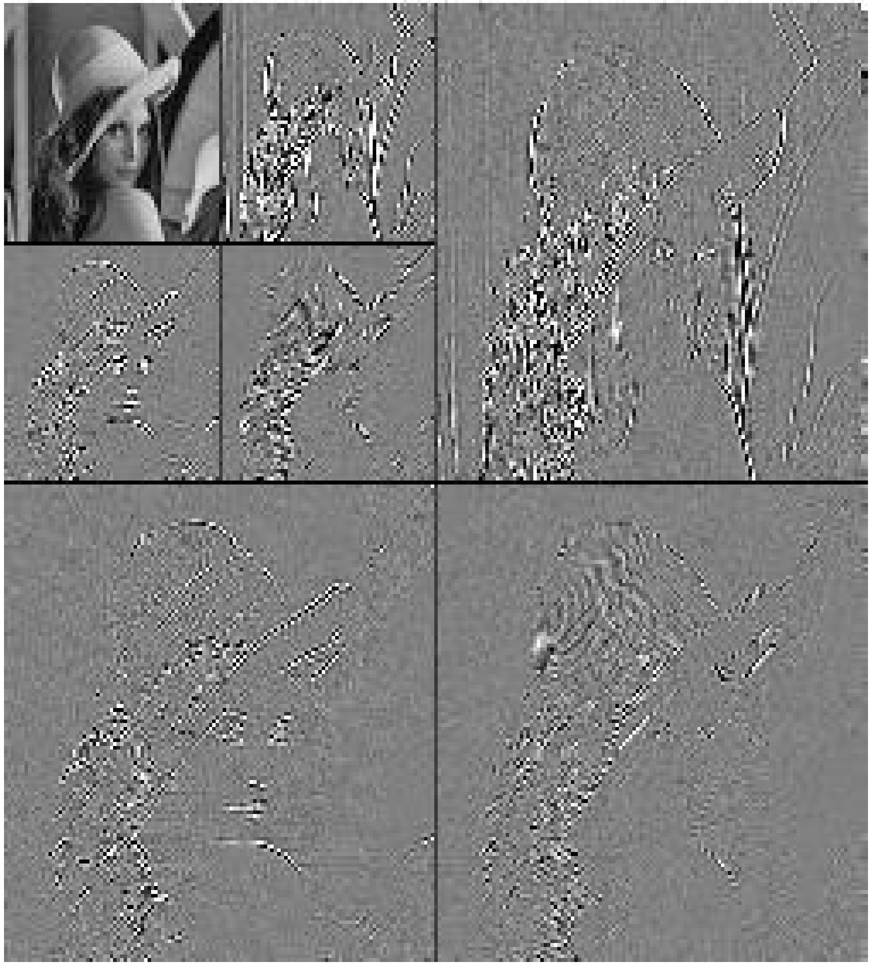
- **Fourier transform**
- **Wavelet transform**



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

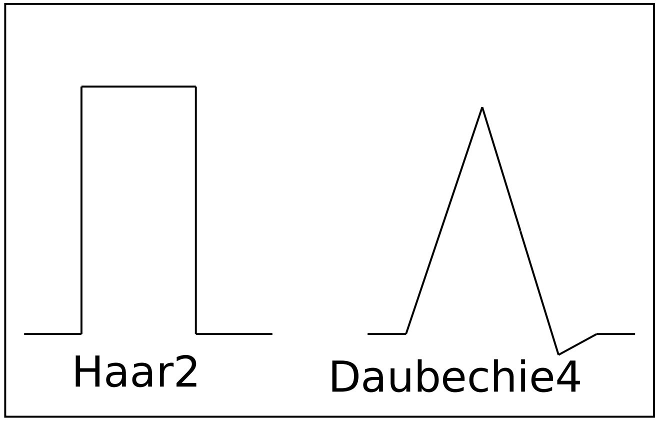
What Is Wavelet Transform?

- Decomposes a signal into different frequency subbands
  - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression



Wavelet Transformation

- Discrete wavelet transform (DWT) for linear signal processing, multi-resolution analysis
- 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



Wavelet Transformation

- DWT Algorithm:
  - Length, L, must be an integer power of 2 (padding with 0’s, when necessary)
  - Each transform needs to apply 2 functions: smoothing (s()), difference (d())
  - Applies s() and d() to pairs of data $(x_{2i}, x_{2i+1}) \rightarrow$
    two sets A and D of length L/2
  - Applies both s() and d() recursively to A
  - Until reaching the desired length (e.g. 2), obtaining L values (1 value in A, L-1 values in D)
  - Select a few values to represent the wavelet coefficients (e.g. the single value in A and k values in D)

Wavelet Decomposition

- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- S = [2, 2, 0, 2, 3, 5, 4, 4] can be transformed to S_w = [$2\frac{3}{4}$, $-1\frac{1}{4}$, $\frac{1}{2}$, 0, 0, -1, -1, 0]
- s() = avg(); d() = diff / 2
- Compression: many small detail coefficients can be replaced by 0's, and only the significant coefficients are retained

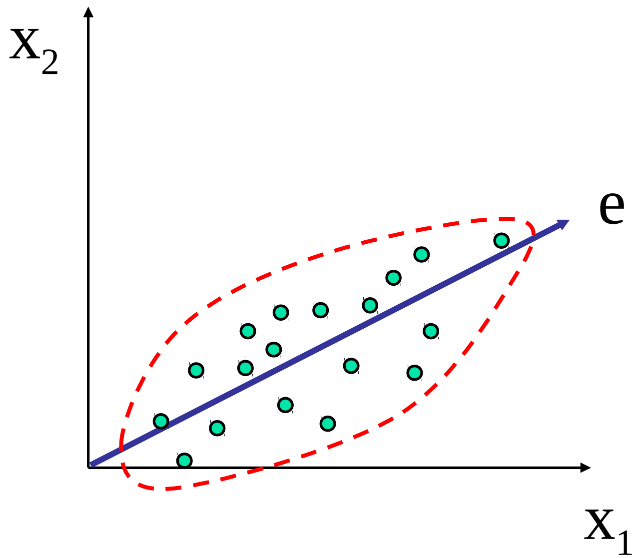
Resolution	Averages	Detail Coefficients
8	[2, 2, 0, 2, 3, 5, 4, 4]	
4	[2, 1, 4, 4]	[0, −1, −1, 0]
2	[$1\frac{1}{2}$, 4]	[$\frac{1}{2}$, 0]
1	[$2\frac{3}{4}$]	[$-1\frac{1}{4}$]

Why Wavelet Transform?

- Use hat-shape filters
  - Emphasize region where points cluster
  - Suppress weaker information in their boundaries
- Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- Multi-resolution
  - Detect arbitrary shaped clusters at different scales
- Efficient
  - Complexity O(N)
- Only applicable to low dimensional data

Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- How?
  - find k (< n) orthogonal vectors that “best” represent data
  - project data into the space defined by these vectors
- Popular choice: eigenvectors



PCA Algorithm (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

PCA Algorithm (remarks)

- Using the strongest principal components, it should be possible to rebuild a good approximation of original data
- Works for numeric data only
- unlike attribute subset selection, **new attributes are found**