Data Mining: Data Preprocessing

Lecture Notes for Chapter 2

Data Mining

by

Zhaonian Zou

2.5 Data Quality

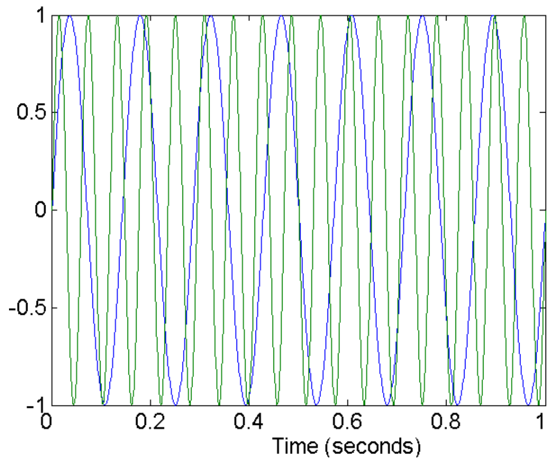
Data Quality

- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

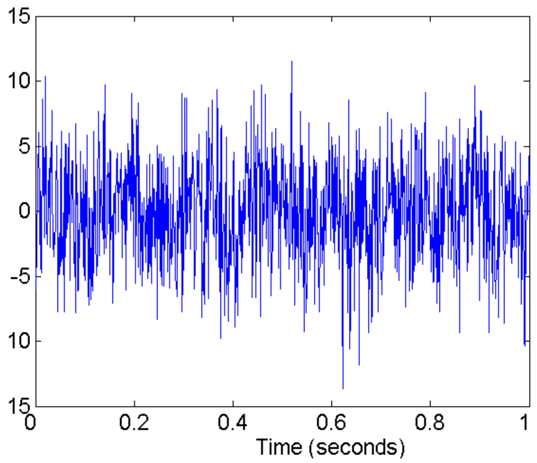
- Examples of data quality problems:
  - Noise
  - Outliers
  - Missing values
  - Duplicate data

Noise

- Noise refers to modification of original values
  - Examples: distortion of a person's voice when talking on a poor phone and "snow" on television screen



Two Sine Waves



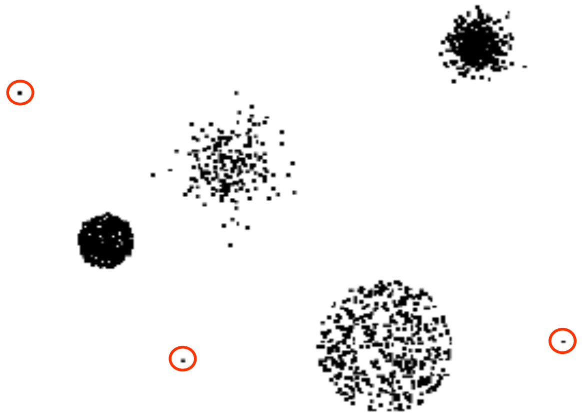
Two Sine Waves + Noise

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

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



Missing Values

- Reasons for missing values
  - Information is not collected (e.g., people decline to give their age and weight)
  - Attributes may not be applicable to all cases (e.g., annual income is not applicable to children)

Duplicate Data

- Data set may include data objects that are duplicates, or almost duplicates of one another
  - Major issue when merging data from heterogenous sources
- Examples:
  - Same person with multiple email addresses

Measures of Data Quality

- 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, eliminate duplicate, and resolve inconsistencies
- **Data Integration (do not introduced in this course)**
  - 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

2.6 Data Cleaning

Data Cleaning

- Data in the real world is dirty: lots of potentially incorrect data, 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*="03/07/2010"
    - 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?

Cleaning 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 or decision tree
  - Replace with all possible values (weighted by their probabilities)

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

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, 25, 25
- Bin 3: 26, 26, 26, 34

Cleaning Inconsistent Data

- Cleaning inconsistent data is more complicated
- Use dependencies, such as functional dependency
- Use human-compiled rules

2.7 Data Reduction

Data Reduction

- **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?
  - 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.

Data Reduction Strategies

- Dimensionality reduction, e.g., remove unimportant attributes
  - Principal Components Analysis (PCA)
  - Feature subset selection, feature creation
  - Wavelet transforms
- Numerosity reduction (some simply call it: Data Reduction)
  - Regression and Log-Linear Models
  - Histograms, clustering, sampling
  - Data cube aggregation
- Data compression

2.7 Data Reduction

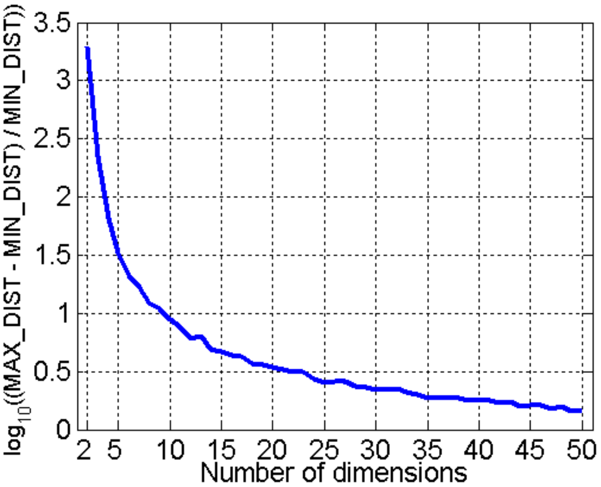
Dimensionality Reduction

Curse of Dimensionality

- If you pick a random point in a unit square, it will have only about a 0.4% chance of being located less than 0.001 from a border
- In a 1000-dimensional unit hypercube, this probability is greater than 99.999999%
- Most points in a high-dimensional hypercube are very close to the border
- Anyone you know is probably an extremist in at least one dimension, if you consider enough dimensions

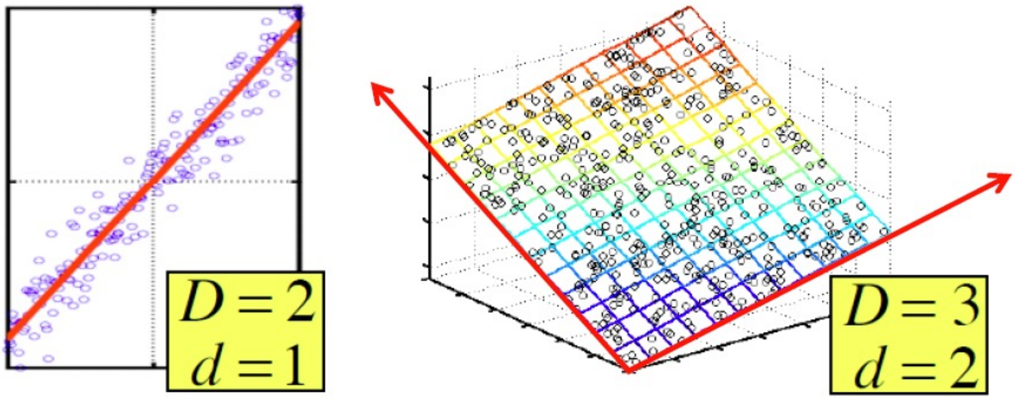
Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



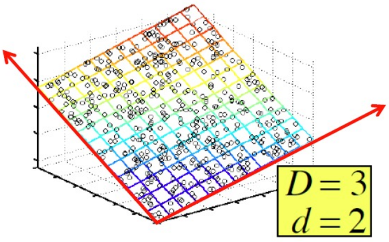
- **Randomly generate 500 points**
- **Compute difference between max and min distance between any pair of points**

Dimensionality Reduction



- **Assumption:** Data lies on or near a low *d*-dimensional subspace
- **Axes of this subspace are effective representation of the data**

Rank is "Dimensionality"



- **Q:** What is **rank** of a matrix **A**?
- **A:** Number of **linearly independent** columns of **A**
- **For example:**
  - Matrix **A** = $\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$ has rank **r = 2**
    - **Why?** The first two rows are linearly independent, so the rank is at least 2, but all three rows are linearly dependent (the first is equal to the sum of the second and third) so the rank must be less than 3.
- **Why do we care about low rank?**
  - We can write **A** as two "basis" vectors: [1 2 1] [-2 -3 1]
  - And new coordinates of : [1 0] [0 1] [1 -1]

Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise

- Techniques
  - Principle Component Analysis (PCA)
  - Singular Value Decomposition (SVD)
  - Others: supervised and non-linear techniques

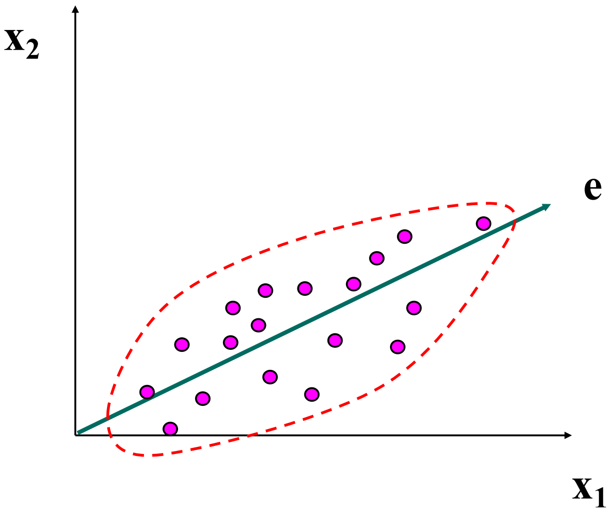
2.7 Data Reduction

Dimensionality Reduction

Principal Component Analysis (PCA)

Idea: Projecting & Preserving Variance

- 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 Components

- PCA identifies the axis (c_1) that accounts for the largest amount of variance in the dataset
  - c_1 is called the first principal component

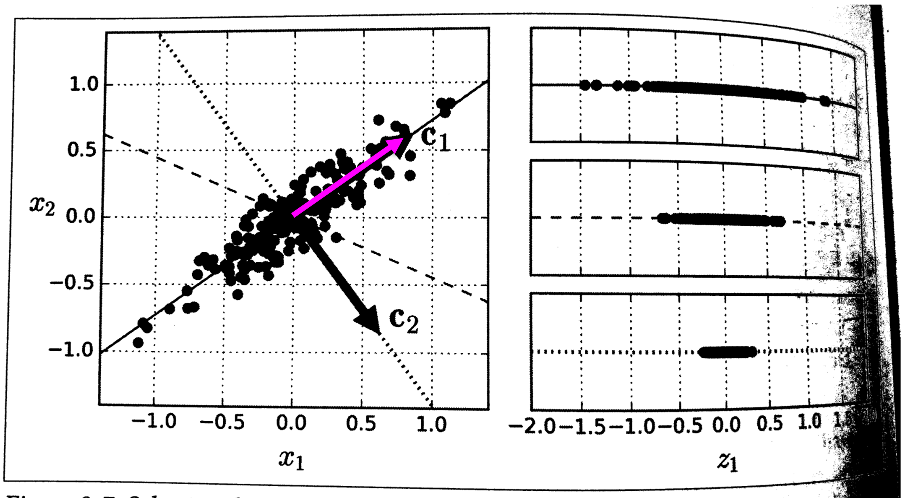


Figure 8-7. Selecting the subspace onto which to project

Principal Components

- PCA also find a second axis (c_2), orthogonal to the first one (c_1), that accounts for the largest amount of remaining variance
  - c_2 is called the second principal component

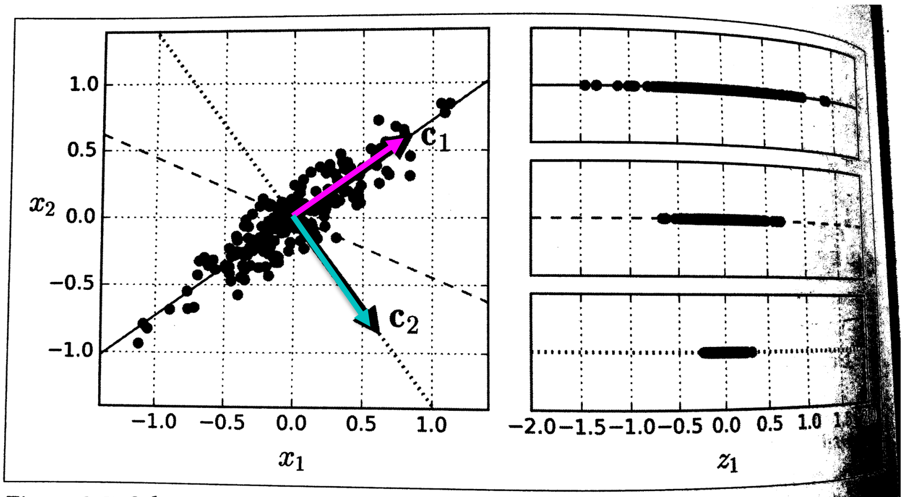


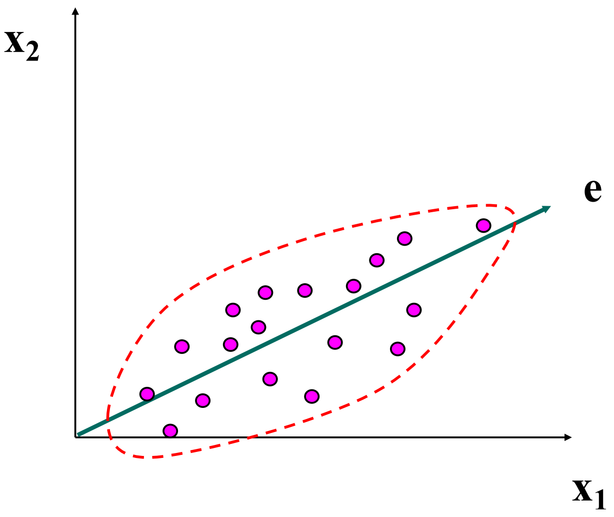
Figure 8-7. Selecting the subspace onto which to project

Principal Components

- If it were a higher-dimensional dataset, PCA would also find a third axis (c_3), orthogonal to both previous axes (c_1 and c_2), that accounts for the largest amount of remaining variance
  - c_3 is called the second principal component
- …
- The number of principal components is equal to the number of dimensions in the dataset

Principal Component Analysis (PCA)

- Find the *eigenvectors* of the *covariance matrix*
- The eigenvectors define the new space



Principal Component Analysis (PCA)

- Covariance matrix
  - **X** =

ID	X_1	X_2	…	X_n
1				
2				
…				
m				

$$\mu_i = \mathrm{E}(X_i)$$

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$

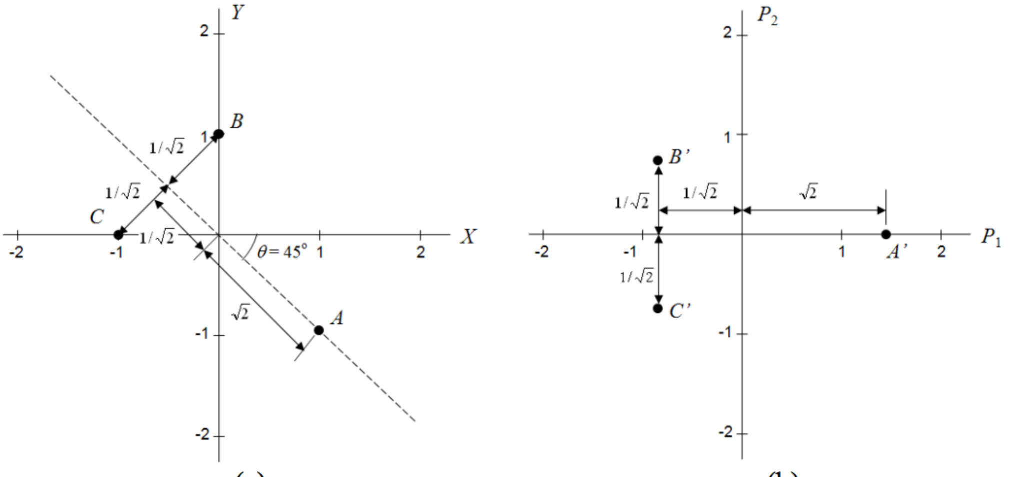
  - $\mathbf{X}^T\mathbf{X}$ can be recognized as proportional to **Σ**

An Example for PCA

- Step 1: center the dataset around the origin

ID	X	Y
A	2	0
B	1	2
C	0	1

$$S = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}$$



(a) (b)

An Example for PCA

- Step 2: compute S^TS

$$C = S^T S = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

- Step 3: find eigenvalues and eigenvectors of C

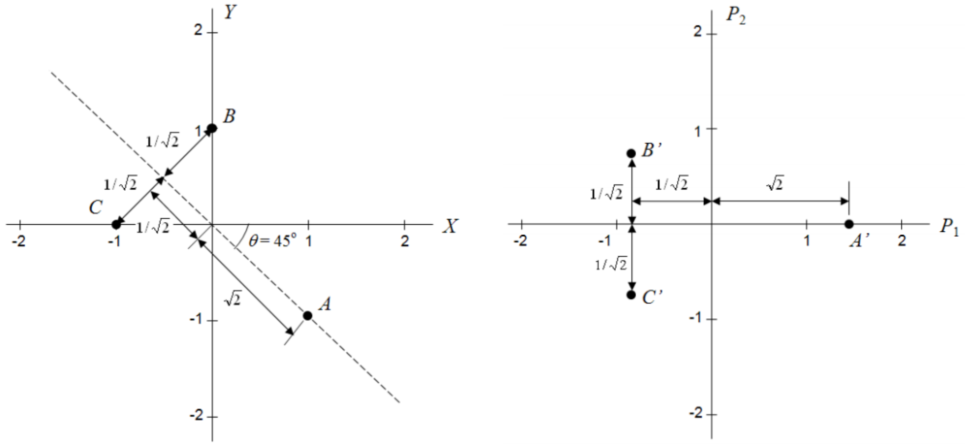
$$\begin{vmatrix} 2-\lambda & -1 \\ -1 & 2-\lambda \end{vmatrix} = 0 \Rightarrow \lambda^2 - 4\lambda + 3 = 0 \Rightarrow \lambda_1 = 3, \text{and } \lambda_2 = 1$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

An Example for PCA

- Step 4: get principle components $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- Step 5: project the dataset into a new space

$$P = SU = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & 0 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}$$



(a)

(b)

2.7 Data Reduction

Dimensionality Reduction

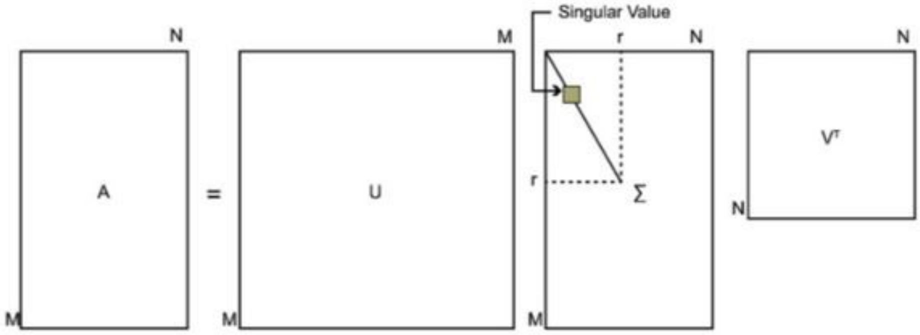
PCA via Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD)

- SVD factorizes a m x n matrix A as follows

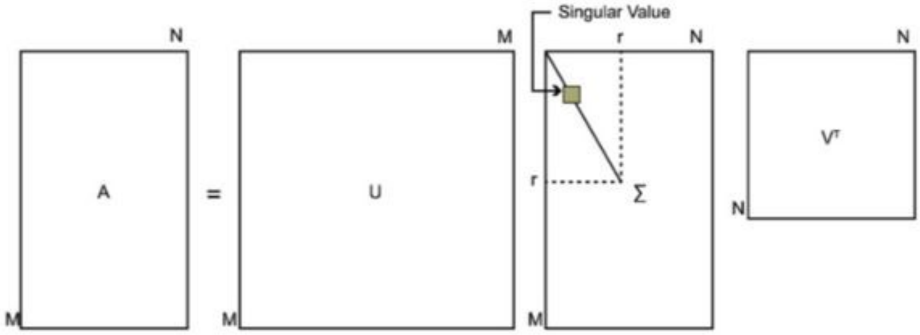
$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\mathbf{V}^T_{n\times n}$$

  - U is a m x m matrix, and $U^T = U^{-1}$
  - V is a n x n matrix, and $V^T = V^{-1}$
  - **Σ** is a m x n diagonal matrix



Computing SVD

- The columns of V (right singular vectors) are the eigenvectors of A^TA
- The columns of U (left singular vectors) are the eigenvectors of AA^T
- The elements on the diagonal of **Σ** (singular values) are square roots of eigenvalues of A^TA (or AA^T)



Computing SVD

- Some algorithms can compute the SVD of a matrix without computing the eigenvectors of A^TA
- The `scikit-learn` tool actually implements PCA by SVD

```
X_centered = X - X.mean(axis=0)
U, s, V = np.linalg.svd(X_centered)
c1 = V.T[:, 0]
c2 = V.T[:, 1]
```

2.7 Data Reduction

Dimensionality Reduction

Attribute Subset Selection

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

Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - ◆The best single-attribute is picked first
    - ◆Then next best attribute condition to the first, ...
  - Step-wise attribute elimination:
    - ◆Repeatedly eliminate the worst attribute
  - Best combined attribute selection and elimination
  - Optimal branch and bound:
    - ◆Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space (see: data reduction)
    - E.g., Fourier transformation, wavelet transformation (not covered)
  - Attribute construction
    - Combining features (see: discriminative frequent patterns)
    - Data discretization

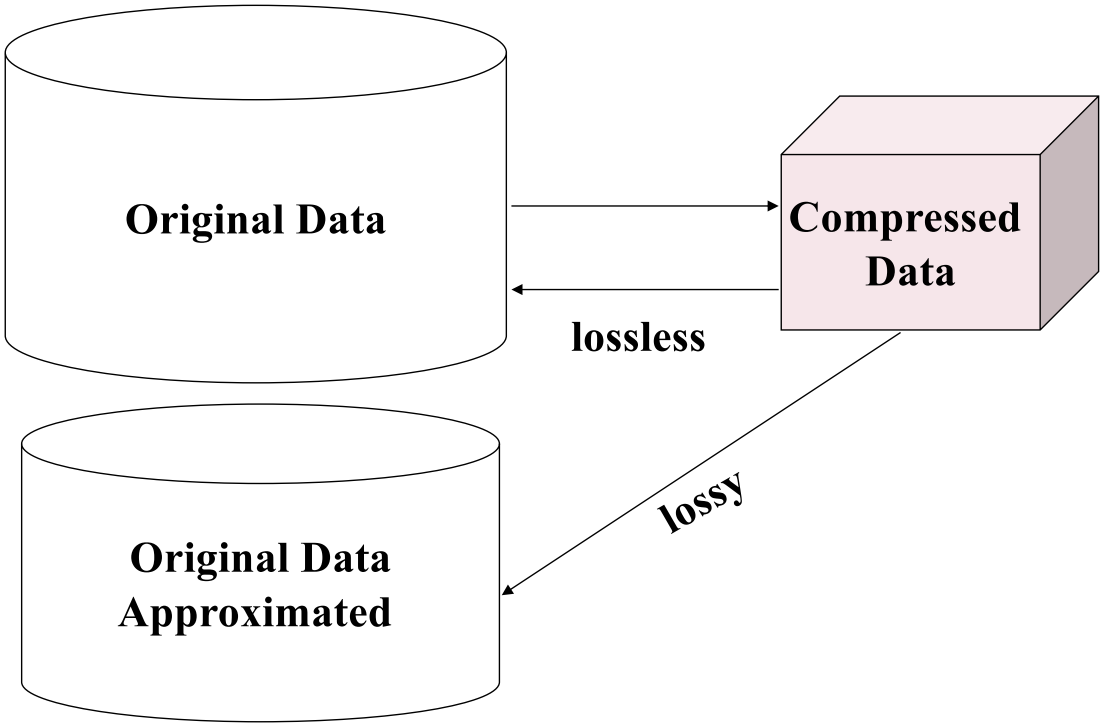
2.7 Data Reduction

Data Compression

Data Compression

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

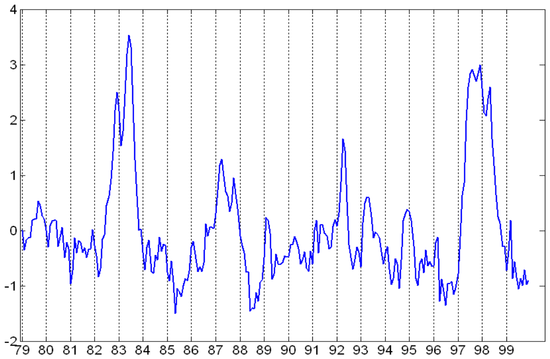
Data Compression



2.8 Data Transformation

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
  - Simple functions: x^k, $\log(x)$, e^x, $|x|$
  - Standardization and Normalization



Data Transformation Methods

- **Smoothing**: Remove noise from data
- **Attribute/feature construction**
  - New attributes constructed from given ones
- **Aggregation**: Summarization, data cube construction
- **Normalization**: Scaled to fall within a smaller, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- **Discretization**: Concept hierarchy climbing

2.8 Data Transformation

Normalization

Normalization

- Min-Max Normalization
- Z-Score Normalization
- Normalization by Decimal Scaling

Min-Max Normalization

- Normalize the attribute values from the current range [v_{min}, v_{max}] to a new range [v'_{min}, v'_{max}]
- Attribute value v is normalized to v' as follows

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

- Example: Let income range $12,000 to $98,000 normalized to [0.0, 1.0]. Then, $73,000 is mapped to 0.716.

$$0 + \frac{73000 - 12000}{98000 - 12000}(1 - 0) = 0.716$$

Z-Score Normalization

- Normalize attribute values to the new ones with mean 0 and standard deviation 1
- Attribute value v is normalized to v' as follows

$$v' = \frac{v - \mu}{\sigma}$$

  - μ: the mean of the values
  - σ: the standard deviation of the values
- Example: Let μ = 54,000 and σ = 16,000. Then, $73,000 is mapped to 1.225.

$$\frac{73000 - 54000}{16000} = 1.225$$

Normalization by Decimal Scaling

- Normalize attribute values to be within [-1, 1]
- Attribute value v is normalized to v' as follows

$$v' = \frac{v}{10^s}$$

  - s: the smallest integer such that

$$\max(|v_{\min}|, |v_{\max}|) \leq 10^s$$

- Example: Let income range $12,000 to $98,000. Then, $73,000 is mapped to 0.73.

2.8 Data Transformation

Discretization

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
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

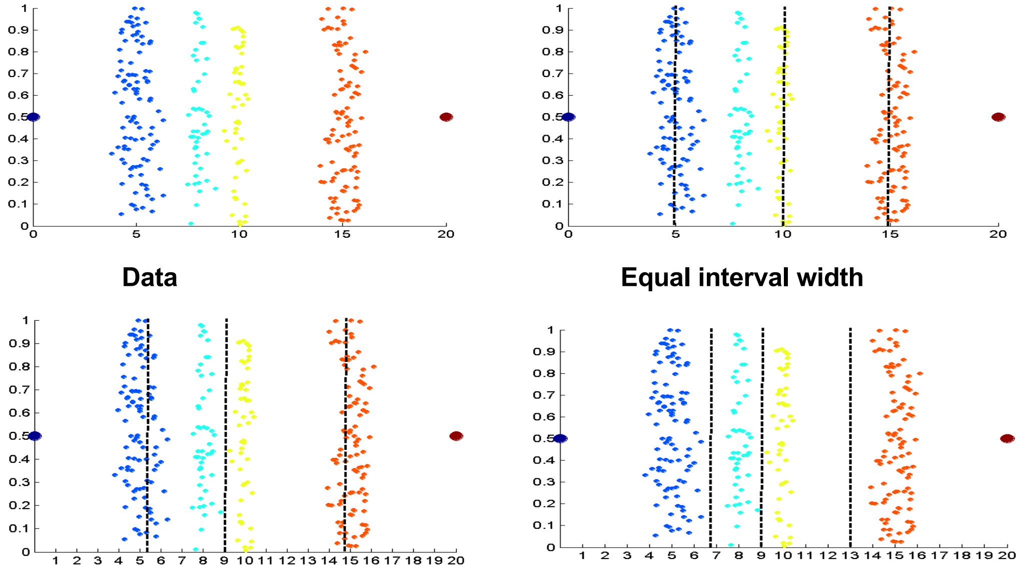
Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - 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
  - Managing categorical attributes can be tricky

Discretization w/o Using Class Labels



Data

Equal interval width

Equal frequency

K-means

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in Chapter 4
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

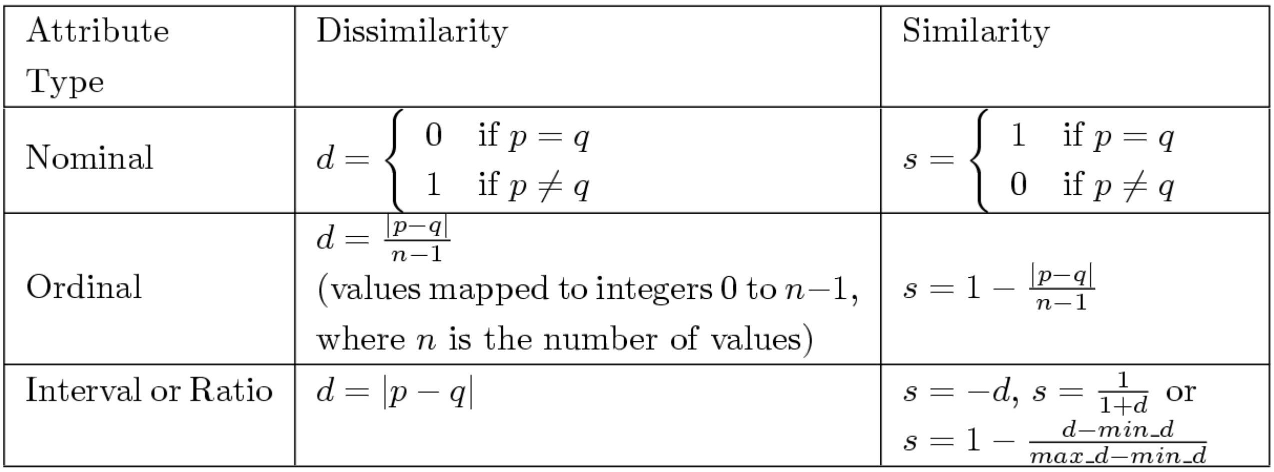
2.9 Similarity and Dissimilarity

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are
  - Is higher when objects are more alike
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Similarity/Dissimilarity for Simple Attributes

p and *q* are the attribute values for two data objects.



Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Euclidean Distance

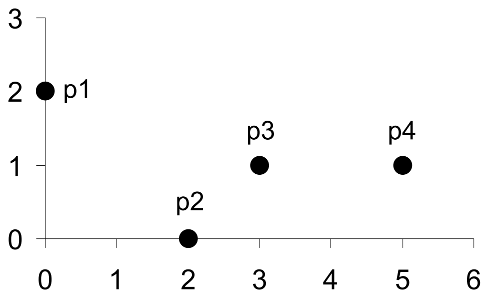
- Euclidean Distance

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

  – *n* is the number of dimensions (attributes) and p_i and q_i are, respectively, the ith attributes (components) or data objects *p* and *q*.

- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(p, q) \geq 0$ for all *p* and *q* and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
2. $d(p, q) = d(q, p)$ for all *p* and *q*. (Symmetry)
3. $\mathrm{d}(p, r) \leq d(p, q) + d(q, r)$ for all points *p*, *q*, and *r*. (Triangle Inequality)

where $d(p, q)$ is the distance (dissimilarity) between points (data objects), *p* and *q*.

- A distance that satisfies these properties is a metric

Common Properties of a Similarity

- Similarities, also have some well known properties.

  1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.
  2. $s(p, q) = s(q, p)$ for all p and q. (Symmetry)

  where $s(p, q)$ is the similarity between points (data objects), p and q.

2.9 Similarity and Dissimilarity

Similarity between Binary Vectors

Similarity Between Binary Vectors

- Common situation is that objects, *p* and *q*, have only binary attributes
- Compute similarities using the following quantities

  M_{01} = the number of attributes where p was 0 and q was 1

  M_{10} = the number of attributes where p was 1 and q was 0

  M_{00} = the number of attributes where p was 0 and q was 0

  M_{11} = the number of attributes where p was 1 and q was 1
- Simple Matching and Jaccard Coefficients

  SMC = number of matches / number of attributes

  = $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

  J = number of 11 matches / number of not-both-zero attributes values

  = $(M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0

q = 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$

J = $(M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$

2.9 Similarity and Dissimilarity

Cosine Similarity

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

  cos(d_1, d_2) = ($d_1 \bullet d_2$) / $||d_1||\ ||d_2||$,

where • indicates vector dot product and || *d* || is the length of vector *d*.

- Example:

  d_1 = **3 2 0 5 0 0 0 2 0 0**

  d_2 = **1 0 0 0 0 0 0 1 0 2**

$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1||$ = $(3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5}$ = $(42)^{0.5}$ = 6.481

$||d_2||$ = $(1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5}$ = $(6)^{0.5}$ = 2.245

cos(d_1, d_2) = .3150

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k, in the range $[0, 1]$.
2. Define an indicator variable, δ_k, for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of } 0\text{, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
  - Use weights w_k which are between 0 and 1 and sum to 1.

$$similarity(p, q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$distance(p, q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}$$

2.10 Correlation

Correlation

- Correlation analysis tests how two attributes are related
- 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
- Techniques
  - Pearson's correlation coefficient
  - Chi-square test

2.10 Correlation

Pearson’s Correlation Coefficient

Pearson’s Correlation Coefficient

- Pearson’s correlation coefficient measures the correlation between two *numeric* attributes

$$\operatorname{corr}(x,y) = \frac{s_{xy}}{s_x s_y} \qquad x = (x_1, x_2, \ldots, x_n)^T \quad y = (y_1, y_2, \ldots, y_n)^T$$

  - s_{xy}: *covariance* of x and y

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

  - s_x: *sample standard deviation* of x

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

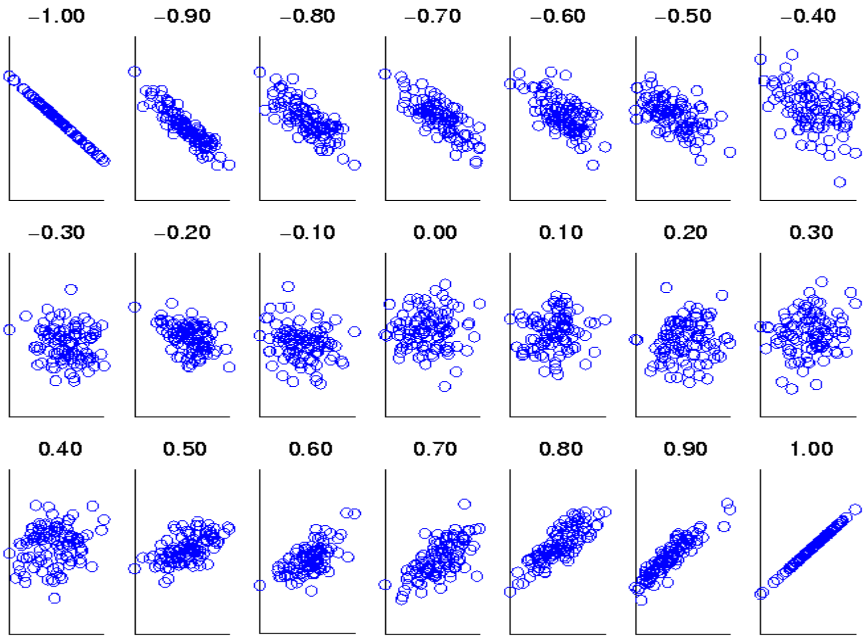
  - s_y: *sample standard deviation* of y

Pearson's Correlation Coefficient …

- Attributes x and y are independent if corr(x, y) = 0
  - The covariance of x and y is 0
- x and y are positively correlated if corr(x, y) > 0
  - x's values increase as y's increases
  - The higher, the stronger correlation
- x and y are negatively correlated if corr(x, y) < 0
  - x's values decrease as y's increases
  - The lower, the stronger correlation

Visually Evaluating Correlation

- Scatter plots of (x_i, y_i) showing the correlation from –1 to 1

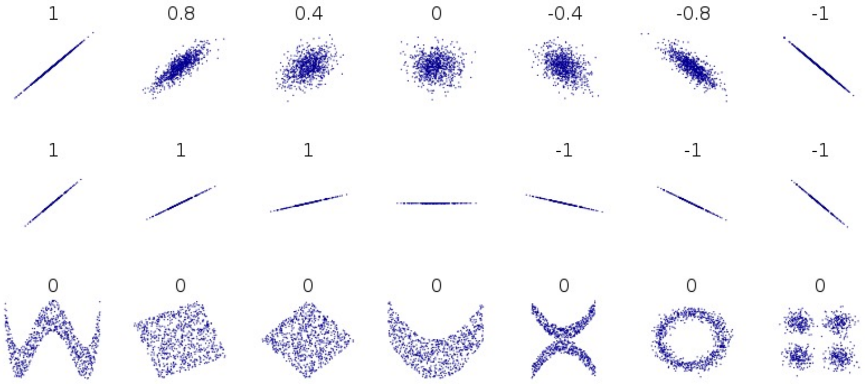


Limitations of Correlation Coefficient

- The correlation coefficient only measures linear correlations

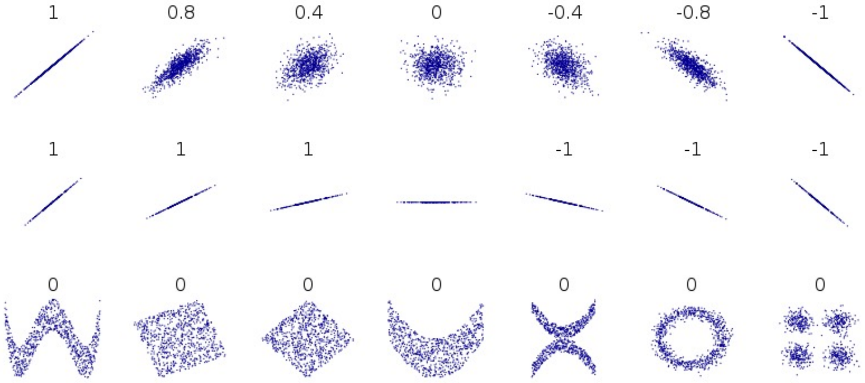
  If x goes up, then y generally goes up/down

- It may completely miss out on nonlinear relationships



Limitations of Correlation Coefficient

- The second row shows examples where the correlation coefficient is equal to 1 or -1
- This has nothing to do with the slope



2.10 Correlation

χ^2 (Chi-Square) Test

Chi-Square Test

- Chi-square test examines the correlation between two nominal attributes
- **Contingency Tables**
  - Attribute x has values $x_1, x_2, \ldots, x_m$
  - Attribute y has values $y_1, y_2, \ldots, y_n$
  - c_{ij}: # of objects with $x = x_i$ and $y = y_j$

	$y = y_1$	$y = y_2$	…	$y = y_n$	Total
$x = x_1$	c_{11}	c_{12}	…	c_{1n}	c_{1*}
$x = x_2$	c_{21}	c_{22}	…	c_{2n}	c_{2*}
…	…	…	…	…	…
$x = x_m$	c_{m1}	c_{m2}	…	c_{mn}	c_{m*}
Total	c_{*1}	c_{*2}	…	c_{*n}	c

Chi-Square Test …

- **X^2 Statistic**: $$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(c_{ij}-e_{ij})^2}{e_{ij}}$$
  - e_{ij}: the *expected* number of objects with x = x_i and y = y_j

$$e_{ij} = \frac{c_{i*}c_{*j}}{c}$$

	y = y_1	y = y_2	…	y = y_n	Total
x = x_1	c_{11}	c_{12}	…	c_{1n}	c_{1*}
x = x_2	c_{21}	c_{22}	…	c_{2n}	c_{2*}
…	…	…	…	…	…
x = x_m	c_{m1}	c_{m2}	…	c_{mn}	c_{m*}
Total	c_{*1}	c_{*2}	…	c_{*n}	c

Chi-Square Test …

- **X^2 Statistic**: $$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(c_{ij}-e_{ij})^2}{e_{ij}}$$
- The larger X^2 is, the more likely x and y are related
- **Null hypnosis**: x and y are independent
- Under the null hypothesis, X^2 has approximately a chi-square distribution whose number of degrees of freedom are (m – 1)(n – 1)

Thanks