Data

COMP5318/COMP4318 Machine Learning and Data Mining

semester 1, 2023, week 1b

Irena Koprinska

Reference: Tan ch. 2

THE UNIVERSITY OF SYDNEY

Outline

- Nominal and numeric attributes
- Data cleaning
  - Noise
  - Missing values
- Data preprocessing
  - Data aggregation
  - Feature extraction
  - Feature subset selection
  - Converting features from one type to another
  - Normalization of feature values
- Similarity measures
  - Euclidean, Manhattan, Minkowski
  - Hamming, SMC, Jaccard coefficient
  - Cosine similarity
  - Correlation

Data

- Data is collection of *examples* (also called *instances, records, observations, objects)*

- Examples are described with attributes *(features, variables)*

Attributes (features)

Class

Examples

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Nominal and numeric attributes

- Two types of attributes:
  - *nominal (categorical)* - their values belong to a pre-specified, finite set of possibilities
  - *numeric (continuous)* - their values are numbers

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

nominal

sepal length	sepal width	petal length	petal width	iris type
5.1	3.5	1.4	0.2	iris setosa
4.9	3.0	1.4	0.2	iris setosa
4.7	3.2	1.3	0.2	iris setosa
...				
6.4	3.2	4.5	1.5	iris versicolor
6.9	3.1	4.9	1.5	iris versicolor
5.5	2.3	4.0	1.3	iris versicolor
6.5	2.8	4.6	1.5	iris versicolor
6.3	3.3	6.0	2.5	iris virginica
5.8	2.7	5.1	1.9	iris virginica
...				

numeric

Types of data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

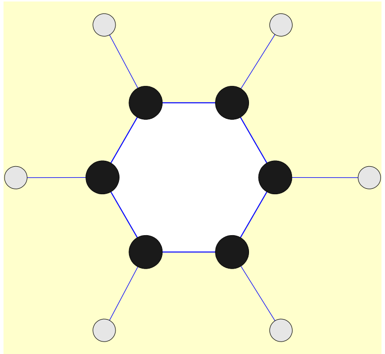
data matrix

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

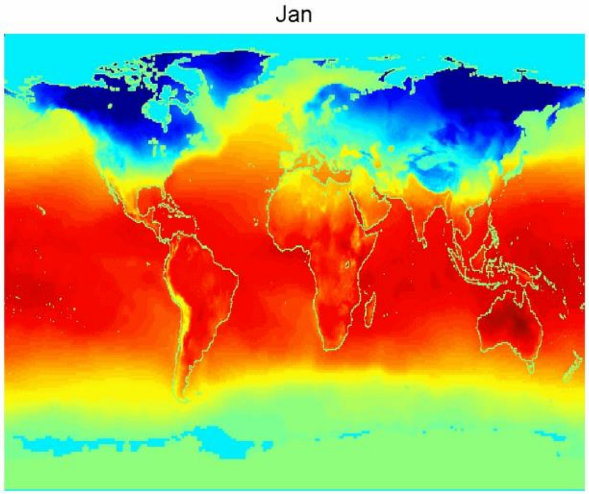
transaction data

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

Sequential
(e.g. genetic sequence)



graph
(e.g. molecular structure)



Average monthly temperature

spatio-temporal

Data cleaning

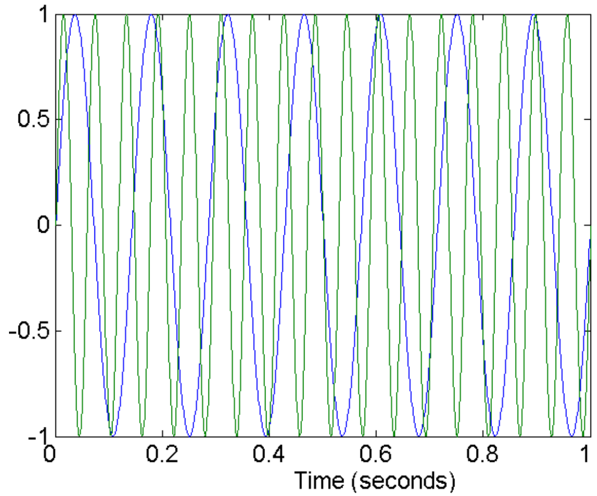
Data cleaning

- Data is not perfect
- Noise due to
  - distortion of values
  - addition of spurious examples
  - inconsistent and duplicate data
- Missing values

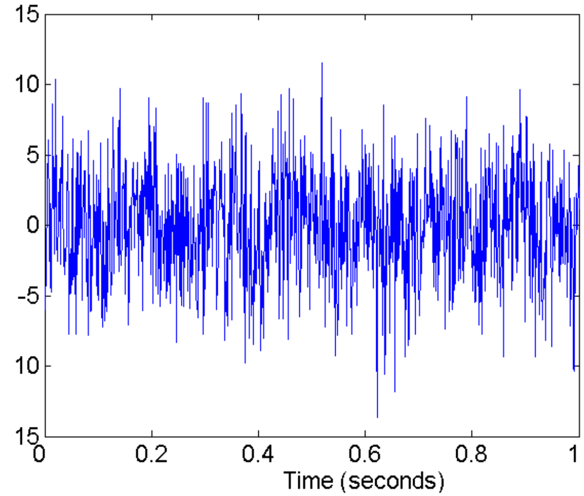
Noise

- Human errors when entering data or limitations of measuring instruments, flaws in the data collection process

- 1) Noise - distortion of values
  - Ex: distortion of human voice when talking on a poor phone line
    - Higher distortion => the shape of the signal may be lost



voice



voice + noise

Noise (2)

- 2) Noise – addition of spurious examples
  - Some are far from the other examples (are outliers), some are mixed with the non-noisy data

(a) Three groups of points.

(b) With noise points (+) added.

Noise (3)

- 3) Noise – inconsistent and duplicate data
  - E.g. negative weight and height values, non-existing zip codes, 2 records for the same person – need to be detected and corrected
  - Typically easier to detect and correct than the other two types of noise

- Reducing noise types 1) and 2):
  - Using signal and image processing and outlier detection techniques before DM
  - Using ML algorithms that are more robust to noise – give acceptable results in presence of noise

Dealing with missing values

- Various methods, e.g.:

1) Ignore all examples with missing values

- Can be done if small % missing values

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
?	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	?	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	?	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

2) Estimate the missing values by using the remaining values

- Nominal attributes - replace the missing values for attribute A with
  - the most common value for A or
  - the most common value among the examples with the same class (if supervised learning)
- Numerical – replace with the average value of the nearest neighbors (the most similar examples)

Data preprocessing

Data preprocessing

- Data aggregation
- Dimensionality reduction
- Feature extraction
- Feature subset selection
- Converting attributes from one type to another
- Normalization

Data aggregation

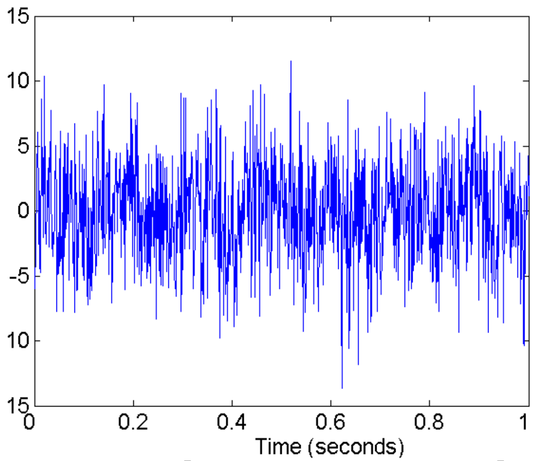
- Combining two or more attributes into one – purpose:
  - Data reduction - less memory and computation time; may allow the use of computationally more expensive ML algorithms
  - Change of scale - provides high-level view
    - E.g. cities aggregated into states or countries
  - More stable data - aggregated data is less variable than non-aggregated
    - E.g. consumed daily food (food_day1, food_day2, etc.) aggregated into weekly food to get a more reliable understanding of the diet (carbohydrates, fat, protein, etc.)
  - Disadvantage – potential loss of interesting details

Feature extraction

- Feature extraction is the creation of features from raw data – very important task
  - Requires domain expertise
  - Ex: classifying images into outdoors or indoors

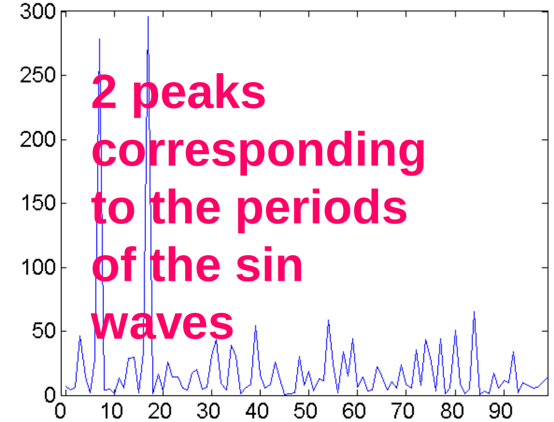
    raw data: color value for each pixel

    extracted features: color histogram, dominant color, edge histogram, etc.
- May require mapping data to a new space, then extract features
  - The new space may reveal important characteristics



2 sin waves + noise

Fourier transform →



power spectrum

Feature subset selection

- The process of removing irrelevant and redundant features and selecting a small set of features that are necessary and sufficient for good classification
- Very important for successful classification
- Good feature selection typically improves accuracy
- Using less features also means:
  - Faster building of the classifiers, i.e. reduces computational cost
  - Often more compact and easier to interpret classification rule

Useful references:

Kohavi, R., John, H.: Wrappers for feature subset selection, Artificial Intelligence, vol. 97, issue 1-2 (1997), pp. 273 – 324

Hall, M.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. 17th Int. Conf. on Machine Learning (ICML). Morgan Kaufmann (2000) 359-366

Feature subset selection methods

- Brute force – try all possible combinations of features as input to a ML algorithm, select the best one (rarely possible in practice – too many combinations of features)
- Embedded - some ML algorithms can automatically select features (e.g. decision trees)
- Filter – select features before the ML algorithm is run; the feature selection is independent of the ML algorithm
  - Based on statistical measures, e.g. information gain, mutual information, odds ratio, etc.
  - Correlation-based feature selection, Relief
- Wrapper – select the best subset for a given ML algorithm; it uses the ML algorithm as a black box to evaluate different subsets and select the best

Feature selection is well studied in ML and there are many excellent methods!

Feature weighting

- Can be used instead of feature reduction or in conjunction with it
- The more important features are assigned a higher weight, the less important – lower
  - manually - based on domain knowledge
  - automatically – some classification algorithms do it (e.g. boosting) or may do it if this option is selected (k-nearest neighbor)
- Key idea: features with higher weights play more important role in the construction of the ML model

Converting attributes from one type to another

- Converting numeric attributes to nominal (discretization)
- Converting numeric and nominal attributes to binary attributes (binarization)

- Needed as some ML algorithms work only with numeric, nominal or binary attributes

- Converting categorical and numeric attributes into binary
  - There is no best method; the best one is the one that works best for a given ML algorithm but all possibilities cannot be evaluated
- Simple technique
  - categorical attribute -> integer -> binary
  - numeric attribute -> categorical -> integer -> binary

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

categorical -> binary

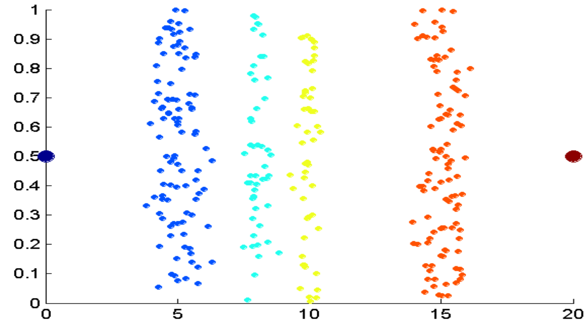
Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Discretization

- Converting numeric attributes into nominal
- 2 types: unsupervised and supervised
  - Unsupervised – class information is not used
  - Supervised - class information is used
- Decisions to be taken
  - How many categories (intervals)?
  - Where should the splits be?

numeric -> nominal

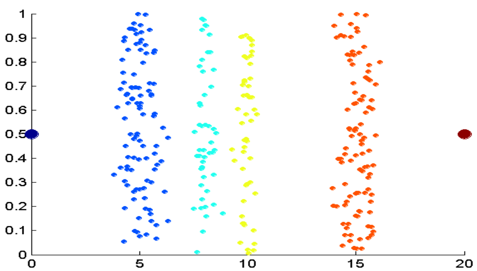


2-dim data; x and y are numeric attributes

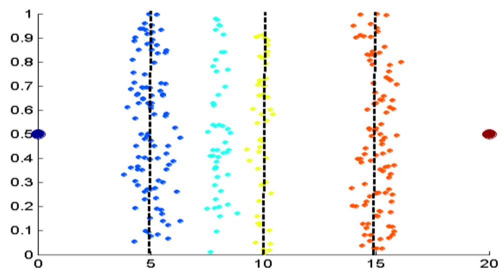
Goal: convert x from numeric to nominal

Unsupervised discretization

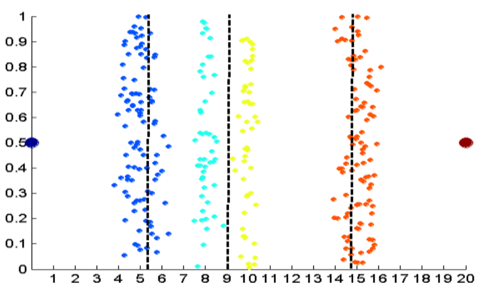
- How many intervals?
  - The user specifies them, e.g. 4
- Where should the splits be?
  - 3 methods



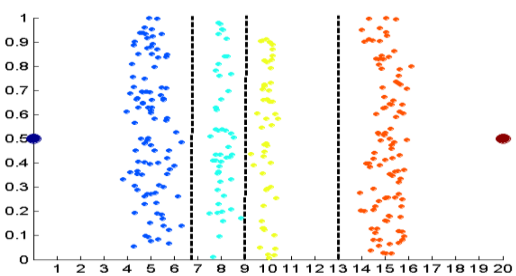
Data



Equal width



Equal frequency



K-means

- *equal width* – 4 intervals with the same width - [0,5), [5, 10), [11,15), [15,20)
- *equal frequency* – 4 intervals with the same number of points in each of them
- *clustering (e.g. k-means)* – 4 intervals determined by a clustering method

Supervised discretization – entropy-based

- Splits are placed so that they maximizes the *purity* of the intervals
- *Entropy* is a measure of the purity of a dataset (interval) S
- The higher the entropy, the lower the purity <u>of the dataset</u>

$$entropy(S) = -\sum_i P_i . \log_2 P_i$$

Pi - proportion of examples from class i

- Ex.: Consider a split between 70 and 71. What is the entropy of the left and right datasets (intervals)?
- values of temperature:

64	65	68	69	70 |	71	72	73	74	75	80	81	83	85
yes	no	yes	yes	yes |	no	no	no	yes	yes	no	yes	yes	no

$$entropy(S_{left}) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.722\, bits$$

$$entropy(S_{right}) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.991\, bits$$

Entropy-based discretization - example

- Total entropy of the split = weighted average of the interval entropies

$$totalEntropy = \sum_{i}^{n} w_i \, entropy(S_i)$$

w_i – proportion of values in interval *i*, *n* – number of intervals

- Algorithm: evaluate all possible splits and choose the best one (with the lowest total entropy); repeat recursively until stopping criteria are satisfied (e.g. user specified number of splits is reached)

Entropy-based discretization – example (2)

-attribute *temperature*

64	65	68	69	70	71	72	73	74	75	80	81	83	85
yes	no	yes	yes	yes	no	no	no	yes	yes	no	yes	yes	no

- 7 initial possible splits
- For each of the 7 splits:
  - Compute the entropy of the 2 intervals
  - Compute the total entropy of the split
- Choose the best split (the one with minimum total entropy)
- Repeat for the remaining splits until the desired number of splits is reached

Normalization and standardization

- Attribute transformation to a new range, e.g. [0,1]
- Used to avoid the dominance of attributes with large values over attributes with small values
- Required for distance-based ML algorithms; some other algorithms also work better with normalized data
  - E.g. *age* (in years) and annual *income* (in dollars) have different scales

    A=[20, 40 000]

    B=[40, 60 000]

    D(A,B)=|20-40| + |40 000-60 000|=20 020

    Difference in *income* dominates, *age* doesn't contribute

- Solution: first normalize or standartize then calculate distance

Normalization and standardization (2)

- Performed for <u>each</u> attribute

Normalization
(also called min-max scaling):

$$x' = \frac{x - \min(\mathrm{x})}{\max(\mathrm{x}) - \min(\mathrm{x})}$$

Standardization:

$$x' = \frac{x - \mu(\mathrm{x})}{\sigma(\mathrm{x})}$$

x – original value
x' – new value

x – all values of the attribute; a vector
min(x) and max(x) – min and max values of the attribute (of the vector x)
μ(x) - mean value of the attribute
σ(x) - standard deviation of the attribute

Normalization - example

Examples with 2 attributes: *age* and *income*:

A=[20, 40 000]

B=[40, 60 000]

C=[25, 30 000]

…

Suppose that:

for *age*: min = 0, max=100

for *income*: min=0, max=100 000

After normalization:

A=[0.2, 0.4]

B=[0.4, 0.6]

C=[0.25, 0.3]

…

D(A,B)=|0.2-0.4| + |0.4-0.6|=0.4, i.e. *income* and *age* contribute equally

Similarity measures

Measuring similarity

- Many ML algorithms require to measure the similarity between 2 examples
- Two main types of measures
  - Distance
  - Correlation

Euclidean and Manhattan distance

- Distance measures for numeric attributes
  - A, B – examples with attribute values $a_1, a_2, \ldots, a_n$ & $b_1, b_2, \ldots, b_n$
  - E.g. A= [1, 3, 5], B=[1, 6, 9]
- Euclidean distance (L2 norm) – most frequently used

$$D(A,B)=\sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\ldots+(a_n-b_n)^2}$$

D(A,B) = sqrt $((1-1)^2+(3-6)^2+(5-9)^2)=5$

- Manhattan distance (L1 norm)

$$D(A,B)=|a_1-b_1|+|a_2-b_2|+\ldots+|a_n-b_n|$$

D(A,B)=|1-1|+|3-6|+|5-9|=7

Minkowski distance

- Minkowski distance – generalization of Euclidean & Manhattan

$$D(A,B) = (|a_1 - b_1|^q + |a_2 - b_2|^q + ... + |a_n - b_n|^q)^{1/q}$$

q – positive integer

- Weighted distance – each attribute is assigned a weight according to its importance (requires domain knowledge)
  - Weighted Euclidean:

$$D(A,B) = \sqrt{w_1|a_1 - b_1|^2 + w_2|a_2 - b_2|^2 + ... + w_n|a_n - b_n|^2}$$

Similarity between binary vectors

- Hamming distance = Manhattan for binary vectors
  - Counts the number of different bits

$$D(A,B) = |a_1 - b_1| + |a_2 - b_2| + ... + |a_n - b_n|$$

A = [1 0 0 0 0 0 0 0 0 0]

B = [0 0 0 0 0 0 1 0 0 1]

D(A,B) = 3

- Similarity coefficients

  f00: number of matching 0-0 bits

  f01: number of matching 0-1 bits

  f10: number of matching 1-0 bits

  f11: number of matching 1-1 bits

- Calculate these coefficients for the example above!

Answer: f01 = 2, f10 = 1, f00 = 7 , f11 = 0

Similarity between binary vectors (2)

- Simple Matching Coefficient (SMC) - matching 1-1 and 0-0 / num. attributes

  SMC = (f11+f00)/(f01+f10+f11+f00)

  Ex.: A = [1 0 0 0 0 0 0 0 0 0]

  B = [0 0 0 0 0 0 1 0 0 1]

  f01 = 2, f10 = 1, f00 = 7 , f11 = 0

  SMC = (0+7) / (2+1+0+7) = 0.7

- Task: Suppose that A and B are the supermarket bills of 2 customers. Each product in the supermarket corresponds to a different attribute.
  - attribute value = 1 – product was purchased
  - attribute value = 0 - product was not purchased
- SMC is used to calculate the similarity between A and B. Is there any problem using SMC?

Similarity between binary vectors (2)

- Yes, SMC will find all customer transactions (bills) to be similar
- Reason: The number of products that are not purchased in a transaction is much bigger than the number of products that are purchased
- => f00 will be very high (not purchased products)
  - f11 will be low (purchased products)
  - f00 will be much higher than f11 and its effect will be lost

SMC = (f11+f00)/(f01+f10+f11+f00)

- => More generally, the problem is that the 2 vectors A and B contain many 0s, i.e. are very *sparse* => SMC is not suitable for sparse data

A = [1 0 0 0 0 0 0 0 0 0]
B = [0 0 0 0 0 0 1 0 0 1]

SMC vs Jaccard

- An alternative: Jaccard coefficient
  - counts matching 1-1 and ignores matching 0-0

    J=f11/(f01+f10+f11)

    A = [1 0 0 0 0 0 0 0 0 0]
    B = [0 0 0 0 0 0 1 0 0 1]
    f01 = 2, f10 = 1, f00 = 7 , f11 = 0
    J = 0 / (2 + 1 + 0) = 0 (A and B are dissimilar)

- Compare with SMC:

    SMC= (0+7) / (2+1+0+7) = 0.7 (A and B are highly similar - incorrect)

Cosine similarity

- Useful for sparse data (both binary and non-binary)
- Widely used for classification of text documents:

$$\cos(A, B) = \frac{A.B}{||A||\,||B||}$$

  - . - vector dot product, ||*A*|| - length of vector *A*

- Geometric representation: measures the angle between A and B
  - Cosine similarity=1 => angle(A,B)=0º
  - Cosine similarity =0 => angle (A,B)=90º

Cosine similarity - example

- Two document vectors:

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

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

$$\cos(d_1, d_2) = \frac{d_1 . d_2}{||d_1|| \, ||d_2||}$$

$d_1 . d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1||$ = $(3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{1/2}$ = $(42)^{1/2}$ = 6.481

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

=> cos(d_1, d_2) = 0.3150

Correlation

- Measures *linear* relationship between numeric attributes
- *Pearson correlation coefficient* between vectors x and y with dimensionality *n*

$$\text{corr}(\mathbf{x},\mathbf{y}) = \frac{\text{covar}(\mathbf{x},\mathbf{y})}{\text{std}(\mathbf{x})\,\text{std}(\mathbf{y})}$$

where:

$$mean(\mathbf{x}) = \frac{\sum_{k=1}^{n} x_k}{n}$$

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{n}\left(x_k - mean(\mathbf{x})\right)^2}{n-1}}$$

$$\text{co}\,\text{var}(\mathbf{x},\mathbf{y}) = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - mean(x))(y_k - mean(y))$$

- Range: [-1, 1]
  - -1: perfect negative correlation
  - +1: perfect positive correlation
  - 0: no correlation

Correlation - examples

- Ex1: corr(x,y)=?
  x=(-3, 6, 0, 3, -6)
  y=(1,-2, 0,-1, 2)

- Ex2: corr(x,y)=?
  x=(3, 6, 0, 3, 6)
  y=(1, 2, 0, 1, 2)

- Ex3: corr(x,y)=?
  x=(-3, -2, -1, 0, 1, 2, 3)
  y=(9, 4, 1, 0, 1, 4, 9)

Answers

- Ex1: corr(x,y)=?
  x=(-3, 6, 0, 3, -6)
  y=(1,-2, 0,-1, 2)

corr(x,y) = -1

perfect negative linear correlation

- Ex2: corr(x,y)=?
  x=(3, 6, 0, 3, 6)
  y=(1, 2, 0, 1, 2)

corr(x,y) = +1

perfect positive linear correlation

- Ex3: corr(x,y)=?
  x=(-3, -2, -1, 0, 1, 2, 3)
  y=(9, 4, 1, 0, 1, 4, 9)

corr(x,y) = 0

no linear correlation

However, there is a non-linear relationship: $y=x^2$

Correlation – visual evaluation

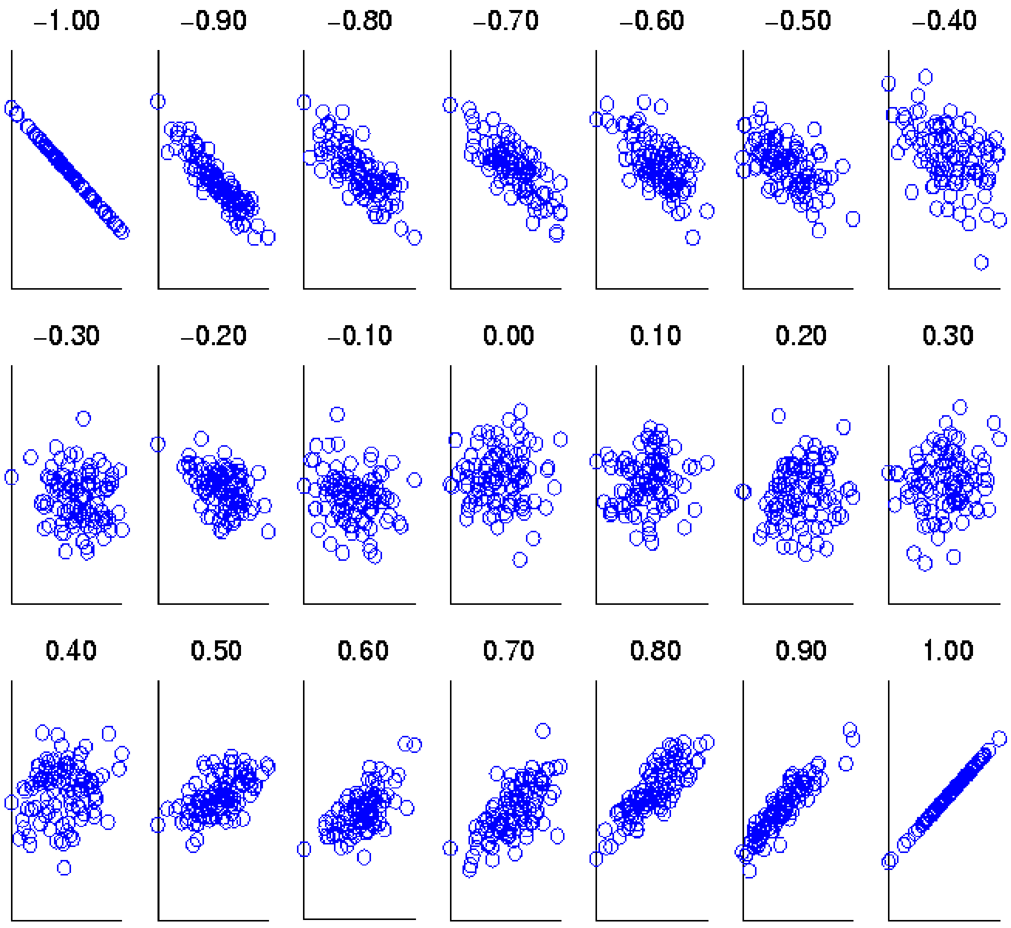


Figure 5.11. Scatter plots illustrating correlations from -1 to 1.

Distance measures for nominal attributes

- Various options depending on the task and type of data type; requires domain expertise
- E.g.:
  - difference =0 if attribute values are the same
  - difference =1 if they are not

  - Example: 2 attributes = *temperature* and *windy*

    temperature values: low and high

    windy values: yes and no

    A={high, no}

    B={high, yes}

    $d(A,B) = (0+1)^{1/2} = 1$ (Euclidean distance)