C | A | U Christian-Albrechts-Universität zu Kiel Technische Fakultät

INF-KDDM: Knowledge Discovery and Data Mining

Winter Term 2019/20

Lecture 2: Data preprocessing and feature spaces

Lectures: Prof. Dr. Matthias Renz

Exercises: Christian Beth

Recap from previous lecture

- KDD definition
- KDD process
- DM step
- Supervised (or predictive) vs Unsupervised (or descriptive) learning
- Main DM tasks
  - Clustering: partitioning in groups of similar objects
  - Classification: predict class attribute from input attributes, class is categorical
  - Regression: predict class attribute from input attributes, class is continuous
  - Association rules mining: find associations between attributes
  - Outlier detection: identify non-typical data

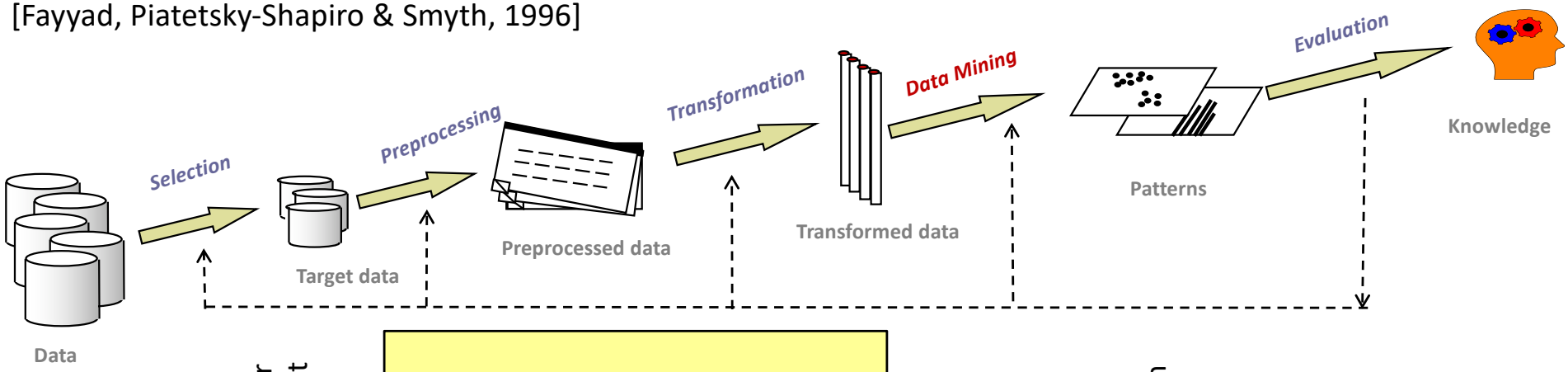
How data mining differs from database querying?

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data
- Homework/ Tutorial
- Things you should know from this lecture

Recap: The KDD process

[Fayyad, Piatetsky-Shapiro & Smyth, 1996]



Selection:
- Select a relevant dataset or focus on a subset of a dataset
- File / DB/

Preprocessing/Cleaning:
- Integration of data from different data sources
- Noise removal
- Missing values

Transformation:
- Select useful features
- Feature transformation/ discretization
- Dimensionality reduction

Data Mining:
- Search for patterns of interest

Evaluation:
- Evaluate patterns based on interestingness measures
- Statistical validation of the Models
- Visualization
- Descriptive Statistics

Why data preprocessing?

- Real world data are noisy, incomplete and inconsistent:
  - Noisy: errors/ outliers
    - erroneous values : e.g. salary = -10K
    - unexpected values: e.g. salary=100K when the rest dataset lies in [30K-50K]
  - Incomplete: missing data
    - missing values: e.g., occupation=“ ”
    - missing attributes of interest: e.g. no information on occupation
  - Inconsistent: discrepancies in the data
    - e.g. student grade ranges between different universities might differ, in DE [1-5], in GR [0-10]
- “Dirty” data → poor mining results
- Data preprocessing is necessary for improving the quality of the mining results !
- Not a focus of this class!

Know your 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 (Entity identification, Value resolution)
- Data transformation:
  - Normalization in a given range, e.g., [0-1]
  - Generalization through some concept hierarchy, e.g. "*milk 1.5% brand x*"→"*milk 1.5%*" or "*milk*"
- Data reduction:
  - Aggregation, e.g., from 12 monthly salaries to month's average salary.
  - Dimensionality reduction, through e.g., PCA
  - Duplicate elimination

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data
- Homework/ Tutorial
- Things you should know from this lecture

Datasets = instances + features

- Datasets consists of instances (also known as examples or objects)
  - e.g., in a university database: students, professors, courses, grades,…
  - e.g., in a library database: books, users, loans, publishers, ….
  - e.g., in a movie database: movies, actors, director,…
- Instances are described through features (also known as attributes or variables)
  - E.g. a course is described in terms of a title, description, lecturer, teaching frequency etc.
  - An easy to visualize example: if our data are in a database table, the rows are the instances and the columns are the features.

ID	Gender	Height(cm)	Weight (kg)	Hair Color	Blood Group	Glasses	Smoker	GGS 787 Grade
67	Female	175	60	brown	A	no	frequent	A+
68	Female	176	52	blond	AB	yes	frequent	A
69	Female	176	63	black	A	yes	casual	A+
70	Female	179	65	brown	0	yes	no	B

Basic feature types

- Binary/ Dichotomous variables
- Categorical (qualitative)
  - Nominal variables
  - Ordinal variables
- Numeric variables (quantitative)
  - Interval-scale variables
  - Ratio-scaled variables

Binary/ Dichotomous variables

- The attribute can take two values, {0,1} or {true,false}
  - usually, 0 means absence, 1 means presence
  - e.g., smoker variable: 1→ smoker, 0→ non-smoker
  - e.g., true (1), false (0)
- Symmetric binary: both outcomes equally important:
  - e.g., gender (male, female)
- Asymmetric binary: outcomes not equally important.
  - e.g., medical tests (positive vs. negative)
  - Convention: assign 1 to most important outcome (e.g., HIV positive)

Person	isSmoker
Eirini	0
Erich	1
Kostas	0
Jane	0
Emily	1
Markus	0

Give me some examples of binary variables!

ID	Gender	Height(cm)	Weight (kg)	Hair Color	Blood Group	Glasses	Smoker	GGS 787 Grade
67	Female	175	60	brown	A	no	frequent	A+
68	Female	176	52	blond	AB	yes	frequent	A
69	Female	176	63	black	A	yes	casual	A+
70	Female	179	65	brown	0	yes	no	B

Categorical: Nominal variables

- The attribute can take values within a set of *M* categories/ states.
  - *No ordering* in the categories/ states.
  - Only *distinctness relationships*, i.e., *equal (==)* and *different* (!=), apply.
  - Examples:
    - Colors = {brown, green, blue,…,gray},
    - Occupation = {engineer, doctor, teacher, …, driver}
    - Gender = {male, female]

Person	gender	occupation
Eirini	female	professor
Erich	male	engineer
Kostas	male	doctor
Jane	female	engineer
Emily	female	teacher
Markus	male	driver

Give me some examples of nominal variables!

ID	Gender	Height(cm)	Weight (kg)	Hair Color	Blood Group	Glasses	Smoker	GGS 787 Grade
67	Female	175	60	brown	A	no	frequent	A+
68	Female	176	52	blond	AB	yes	frequent	A
69	Female	176	63	black	A	yes	casual	A+
70	Female	179	65	brown	0	yes	no	B

Categorical: Ordinal variables

- Similar to categorical variables, but the *M* states are ordered/ ranked in a meaningful way.
  - There is an *ordering* between the values.
  - Allows to apply *order relationships*, i.e., $>, \geq, <, \leq$
  - However, the difference and ratio between these values has no (quantitative) meaning.
  - Examples:
    - School grades: *{A,B,C,D,F}*
    - Movie ratings: {hate, dislike, indifferent, like, love}
      - Also, movie ratings: {*, **, ***, ****, *****}
      - Also, movie ratings: {1, 2, 3, 4, 5}
    - Medals = {bronze, silver, gold}

Person	A beautiful mind	Titanic
Eirini	5	3
Erich	5	1
Kostas	3	3
Jane	1	5
Emily	1	5
Markus	4	3

Give me some examples of ordinal variables!

ID	Gender	Height(cm)	Weight (kg)	Hair Color	Blood Group	Glasses	Smoker	GGS 787 Grade
67	Female	175	60	brown	A	no	frequent	A+
68	Female	176	52	blond	AB	yes	frequent	A
69	Female	176	63	black	A	yes	casual	A+
70	Female	179	65	brown	0	yes	no	B

Numeric: Interval-scale variables

- Measured on a scale of equal-sized units
  - It is assumed that the intervals keep the same importance throughout the scale.
- Differences between values are meaningful
  - The difference between 90^o and 100^o temperature is the same as the difference between 40^o and 50^o temperature.
- Ratio still has no meaning
  - A temperature of 2^o Celsius is not much different than a temperature of 1^o Celsius.
  - The issue is that the 0^o point of the Celsius scale is in a physical sense arbitrary and therefore the ratio of two Celsius temperatures is not physically meaningful.
- No meaningful (unique and non-arbitrary) zero value
- Examples:
  - Temperature in Farenheit or Celsius
  - Calendar dates

Give me some examples of interval-scale variables!

Numeric: Ratio-scale variables

- Both differences and ratios have a meaning
  - E.g., a 100 Kgs person is twice heavy as a 50 Kgs person.
  - E.g., a 50 years old person is twice old as a 25 years old person.
- Meaningful (unique and non-arbitrary) zero value
- Examples:
  - age, weight, length, number of sales
  - temperature in Kelvin
    - When measured on the Kelvin scale, a temperature of 2^{o} is, in a physical meaningful way, twice that of a 1^{o}.

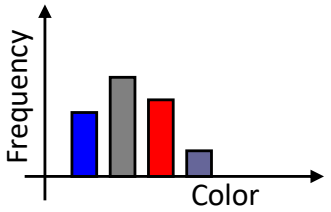
Give me some examples of ratio-scale variables!

ID	Gender	Height(cm)	Weight (kg)	Hair Color	Blood Group	Glasses	Smoker	GGS 787 Grade
67	Female	175	60	brown	A	no	frequent	A+
68	Female	176	52	blond	AB	yes	frequent	A
69	Female	176	63	black	A	yes	casual	A+
70	Female	179	65	brown	0	yes	no	B

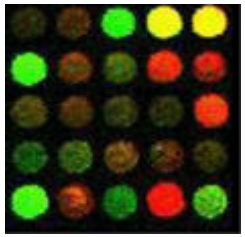
Feature extraction

- Feature extraction depends on the application

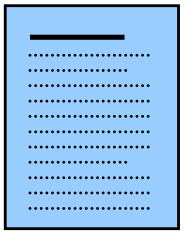
Image databases:
Color histograms



Gene databases:
gene expression level



Text databases:
Word frequencies



Data	25
Mining	15
Feature	12
Object	7
...	

- But, the feature-approach allows uniform treatment of instances from different applications.

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data
- Homework/ Tutorial
- Things you should know from this lecture

Univariate descriptors 1/5

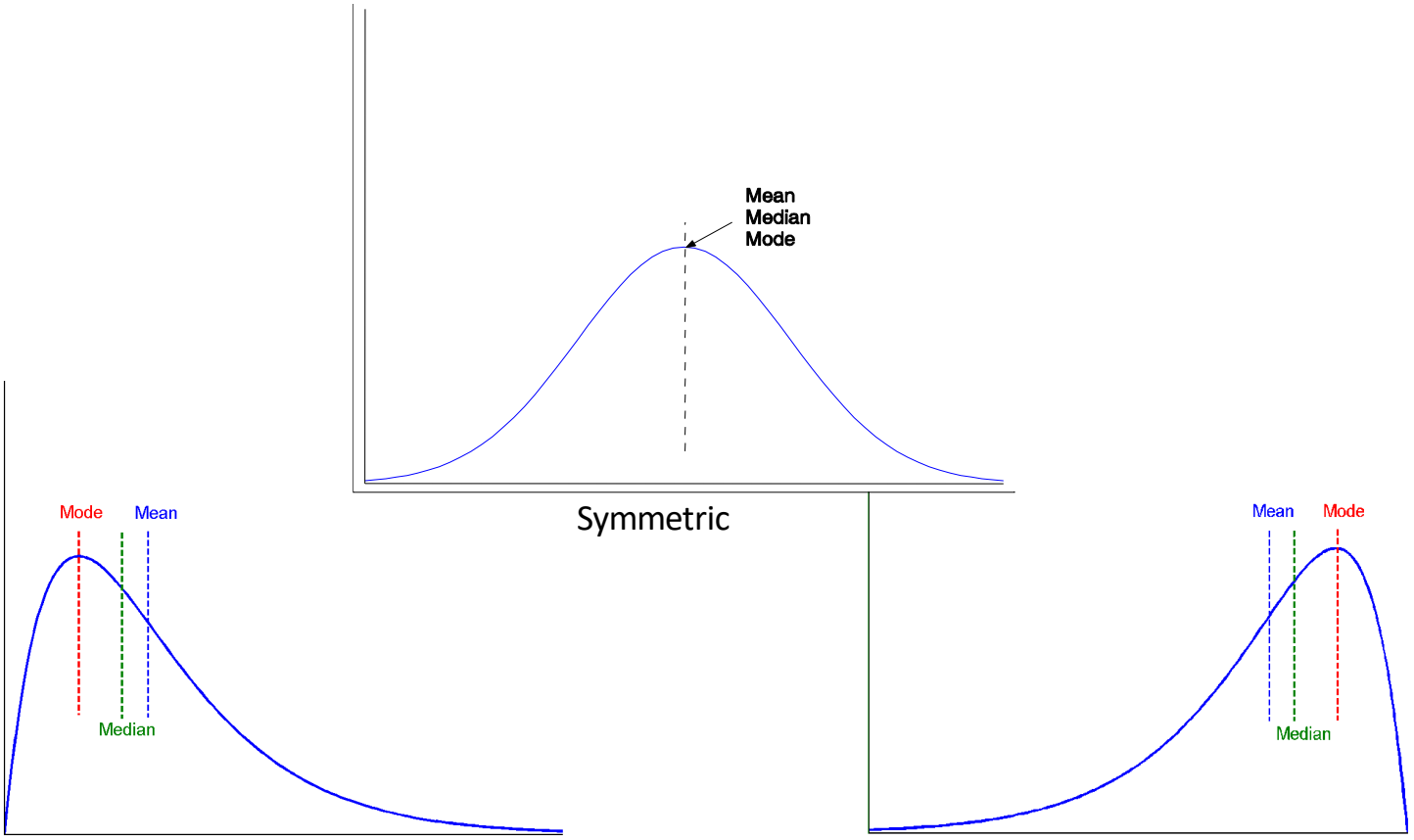
Let $x_1,\ldots,x_n$ be a random sample of an attribute X. Measures of central tendency of X include:

- (Arithmetic) mean/ center/ average:

$$\bar{x} = \frac{1}{n}\sum_{1}^{n} x_i$$

- Weighted average: $\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
- Median: the central element in ascending ordering
  - Middle value if odd number of values, or average of the middle two values otherwise
- Mode: Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal

Univariate descriptors 2/5



Univariate descriptors 3/5

Let $x_1,\ldots,x_n$ be a random sample of an attribute *X*. The degree to which X values tend to spread is called dispersion or variance of X :

- Range: max value – min value
  - Q_1 (25th percentile), Q_3 (75th percentile)
  - Median is the 50th percentile
- 5 number summary: min, Q_1, median, Q_3, max
  - Boxplots to visualize them
- Variance σ^2:

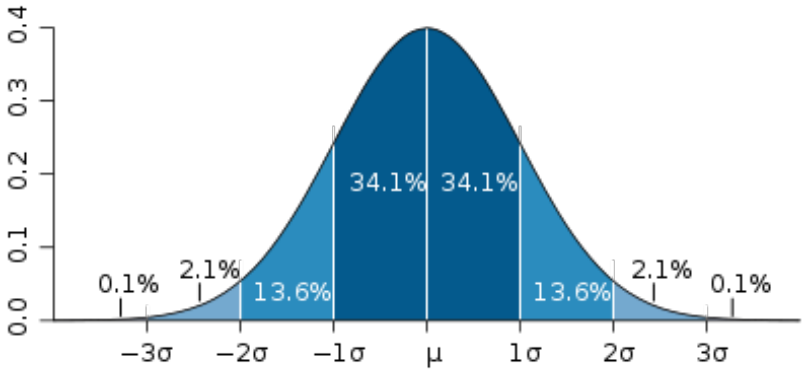
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2]$$

- Standard deviation σ: $\sigma = \sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Univariate descriptors 4/5

Example: The normal distribution curve

- ~68% of values drawn from a normal distribution are from μ-σ to μ+σ
- ~95% of the values lie from μ-2σ to μ+2σ
- ~99.7% of the values are from μ-3σ to μ+3σ

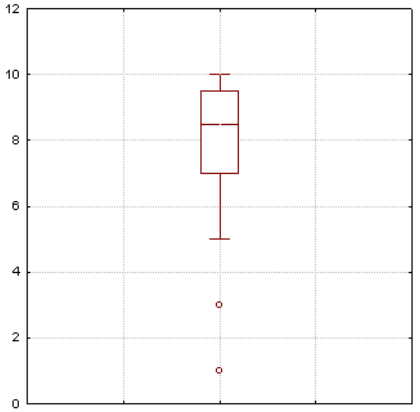


Source: http://en.wikipedia.org/wiki/Normal_distribution

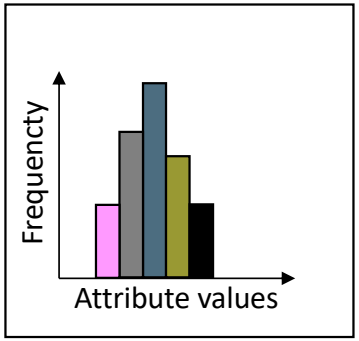
Univariate descriptors 5/5

Let $x_1,\ldots,x_n$ be a random sample of an attribute X. For visual inspection of X, several types of charts are useful, e.g.:

- Boxplots
  - 5 number summary
- Histograms:
  - Summarizes the distribution of X
  - X axis: attribute values, Y axis: frequencies
  - Absolute frequency: for each value a, # occurrences of a in the sample
  - Relative frequency: f(a) = h(a)/n
- Different types of histograms, e.g.:
  - Equal width:
    - It divides the range into *N* intervals of equal size
  - Equal frequency/ depth:
    - It divides the range into *N* intervals, each containing approximately same number of samples



Source: http://de.wikipedia.org/wiki/Boxplot



Bivariate descriptors 1/5

- Given two attributes X, Y one can measure how strongly they are correlated
  - For numerical data → correlation coefficient
  - For categorical data → χ^2 (chi-square)

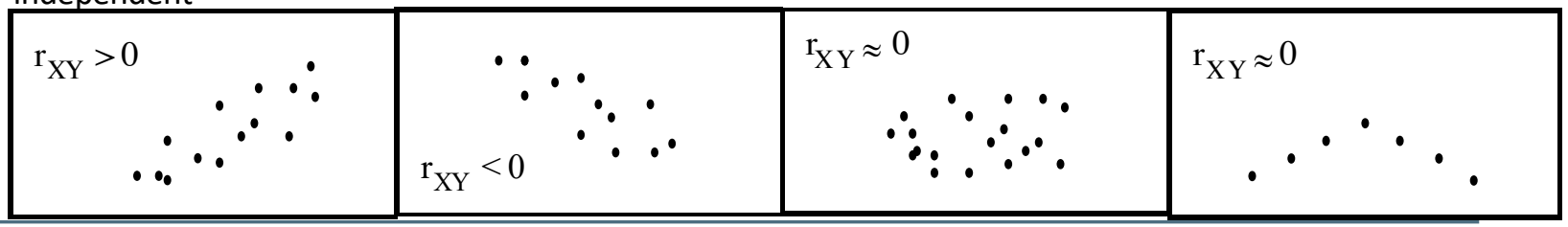
Bivariate descriptors 2/5: for numerical features

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

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{n\sigma_X\sigma_Y} = \frac{\sum_{i=1}^{n}(x_i y_i) - n\bar{x}\bar{y}}{n\sigma_X\sigma_Y}$$

  - n: # tuples; x_i, y_i: the values in the i^{th} tuple for X, Y

- value range: $-1 \leq r_{XY} \leq 1$
- the higher r_{XY} the stronger the correlation
  - $r_{XY} > 0$ positive correlation
  - $r_{XY} < 0$ negative correlation
  - $r_{XY} \sim 0$ no correlation/ independent



Bivariate descriptors 3/5: for numerical features

- Visual inspection of correlation

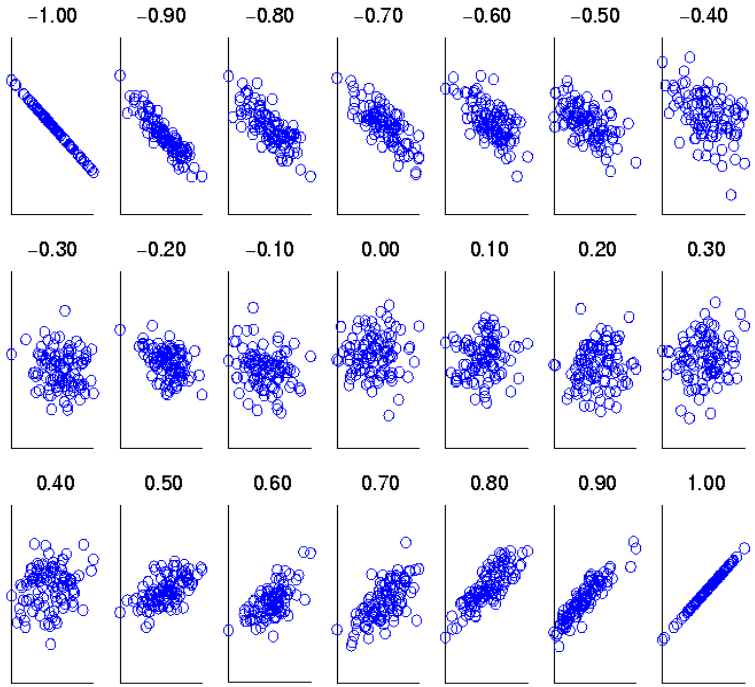


Figure 5.11. Scatter plots illustrating correlations from -1 to 1.

Bivariate descriptors 4/5: for categorical features

- Contingency table
  - For categorical/ nominal features X={x_1, …, x_c }, Y={y_1, …, y_r}
  - Represents the absolute frequency h_{ij} of each combination of values (x_i, y_J) and the marginal frequencies h_i, h_j of X, Y.

Attribute Y

Attribute X	Medium-term unemployment	Long-term unemployment	Total
No education	19	18	37
Teaching	43	20	63
Total	62	38	100

- Chi-square χ^2 test

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

o_{ij}:observed frequency
e_{ij}: expected frequency

$$e_{ij} = \frac{h_i h_j}{n}$$

Bivariate descriptors 4/5: for categorical features

- Chi-square 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 like_science_fiction and play_chess are correlated in the group

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data
- Homework/ Tutorial
- Things you should know from this lecture

Feature spaces and proximity measures

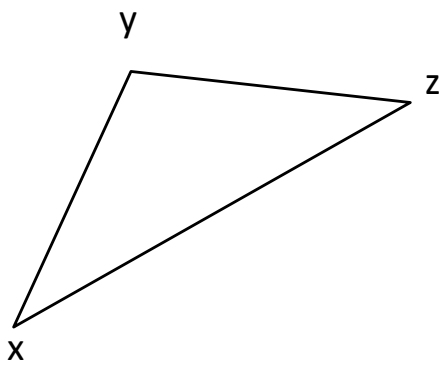
Feature space

- Intuitively: a domain with a distance function
- Formally: feature space **F** = (*Dom*, *dist*):
  - *Dom* is a set of attributes / features
  - *dist*: a numerical measure of the degree to which the two compared objects differ
    - $dist : Dom \times Dom \rightarrow R^{+}_{0}$
- For all *x*, *y* in *Dom*, $x \neq y$, *dist* is required to satisfy the following properties:
  - $dist(x,y) > 0$ (non-negativity)
  - $dist(x,x) = 0$ (reflexivity)

Feature spaces and proximity measures

Metric space

- Formally: Metric space $M = \{Dom, dist\}$:
  - M is a feature space
    - i.e, *dist*(*x*,*y*) > 0 (non-negativity) and,
    - *dist*(*x*,*x*) = 0 (reflexivity)
  - $dist(x, y) = 0 \Rightarrow x = y$ (strictness)
  - $\forall x, y \in Dom: dist(x, y) = dist(y, x)$ *(symmetry)*
  - $\forall x,y,z \in Dom$: $dist(x,z) \leq dist(x,y) + dist(y,z)$ (triangle inequality)



- Measures that satisfy all the above properties are called metrics.

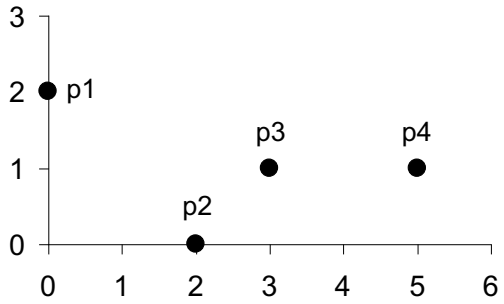
Feature spaces and proximity measures

- Famous example: Euclidean vector space *E*=(*Dom*, *dist*)
  - (*Dom*, *dist*) is a metric space
  - $Dom = \mathbb{R}^d$
  - $dist(x, y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$
- Notation:
  - Euclidean vector space =: “Feature space”
  - Vectors (Objects in the Euclidean feature space) =: “Feature vectors”
  - The *d* dimensions of the vector space =: “Features”
- Standardization is necessary, if scales differ!
  - e.g., age (e.g., range [0-100] vs salary (e.g., range: 10000-100000))

We will come back to this in a few slides

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

Point coordinates



	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

Feature spaces and proximity measures

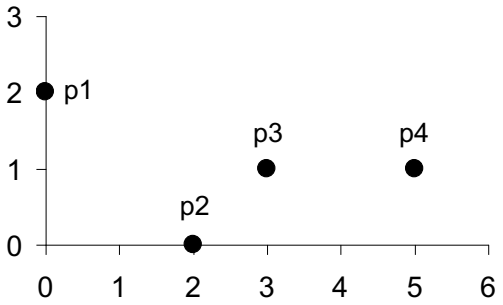
- Manhattan distance or City-block distance (L_1 norm)
  - $dist_1 = |p_1-q_1|+|p_2-q_2|+...+|p_d-q_d|$
  - The sum of the absolute differences of the *p,q* coordinates
- Euclidean distance (L_2 norm)
  - $dist_2 = ((p_1-q_1)^2+(p_2-q_2)^2+...+(p_d-q_d)^2)^{1/2}$
  - The length of the line segment connecting p and q
- Supremum distance (L_{max} norm or L_∞ norm)
  - $dist_\infty = \max\{|p_1-q_1|, |p_2-q_2|,...,|p_d-q_d|\}$
  - The max difference between any attributes of the objects.
- Minkowski Distance (Generalization of L_p-distance)
  - $dist_p = (|p_1-q_1|^p + |p_2-q_2|^p + ...+ |p_d-q_d|^p)^{1/p}$

Feature spaces and proximity measures

- Example

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

Point coordinates



L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L1 distance matrix

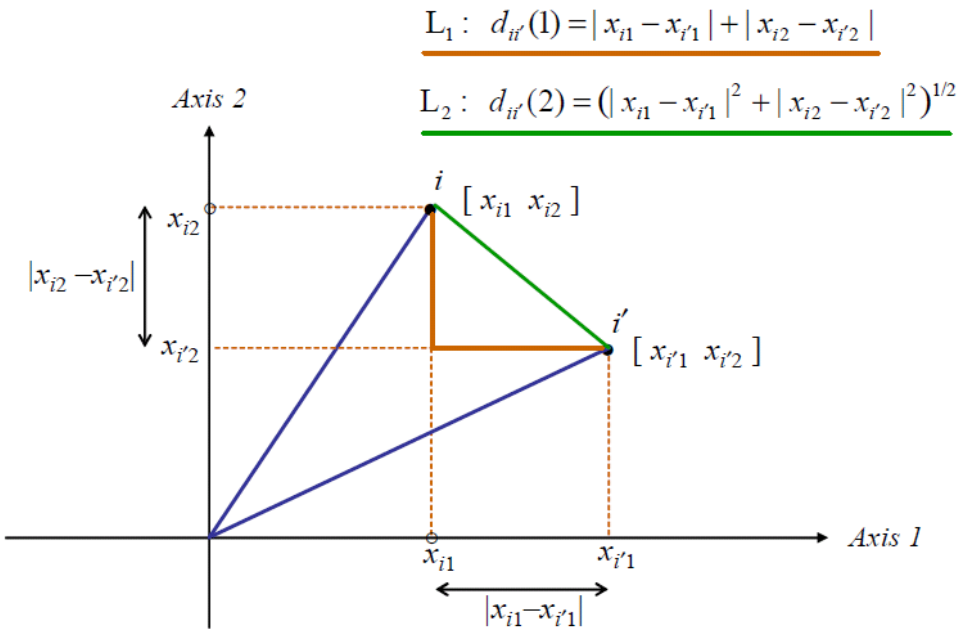
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L2 distance matrix

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

L_∞ distance matrix

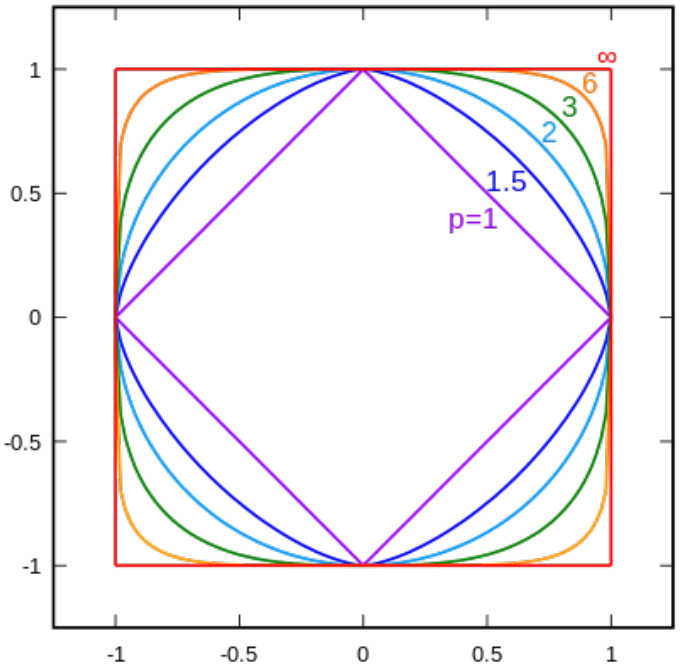
Feature spaces and proximity measures



Source:http://www.econ.upf.edu/~michael/stanford/maeb5.pdf

Feature spaces and proximity measures

- Let *x,y* in [-1,1]
- For L1 norm
  - $|(x,y)|_1=1 \Rightarrow x+y=1$
  - If x=1, y=0
  - If x=0.8, y=0.2
  - …
- For L2 norm
  - $(x^2+y^2)^{1/2}=1$
  - It is circle
- …



Unit Circle for different Lp-distances

Source:https://de.wikipedia.org/wiki/P-Norm

Normalization

- Attributes with large ranges outweigh ones with small ranges
  - e.g. income [10K-100K]; age [10-100]
- To balance the "contribution" of an attribute *A* in the resulting distance, the attributes are scaled to fall within a small, specified range.
- min-max normalization: to [new_min_A , new_max_A]

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

  - e.g. normalize age=30 in [0-1], when min=10,max=100. new_age=((30-10)/(100-10))*(1-0)+0=2/9
- z-score normalization also called zero-mean normalization
  - After zero-mean normalizing each feature will have a mean value of 0

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

e.g. normalize 70,000 iff μ=50,000, σ=15,000.
new_value = (70,000-50,000)/15,000=1.33

Proximity between binary attributes 1/2

- A binary attribute has only two states: 0 (absence), 1 (presence)
- A contingency table for binary data

Instance i \ *Instance j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

q = the number of attributes where i was 1 and j was 1
t = the number of attributes where i was 0 and j was 0

s = the number of attributes where i was 0 and j was 1
r = the number of attributes where i was 1 and j was 0

- Simple matching coefficient

  (for symmetric binary variables)

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

(for *asymmetric* binary variables)

Proximity between binary attributes 2/2

- Example:

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0
Jim	1	1	0	0	0	0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

(from previous slide)

q = the number of attributes where i was 1 and j was 1
t = the number of attributes where i was 0 and j was 0

s = the number of attributes where i was 0 and j was 1
r = the number of attributes where i was 1 and j was 0

$$d(i, j) = \frac{r+s}{q+r+s}$$

Proximity between categorical attributes

- A nominal attribute has >2 states (generalization of a binary attribute)
  - e.g. color={red, blue, green}
- Method 1: Simple matching
  - m: # of matches, p: total # of variables

$$d(i,j) = \frac{p - m}{p}$$

Name	Hair color	Occupation
Jack	Brown	Student
Mary	Blond	Student
Jim	Brown	Architect

- Method 2: Map it to binary variables
  - create a new binary attribute for each of the *M* nominal states of the attribute

Name	Brown hair	Blond hair	IsStudent	IsArchitect
Jack	1	0	1	0
Mary	0	1	1	0
Jim	1	0	0	1

Selecting the right proximity measure

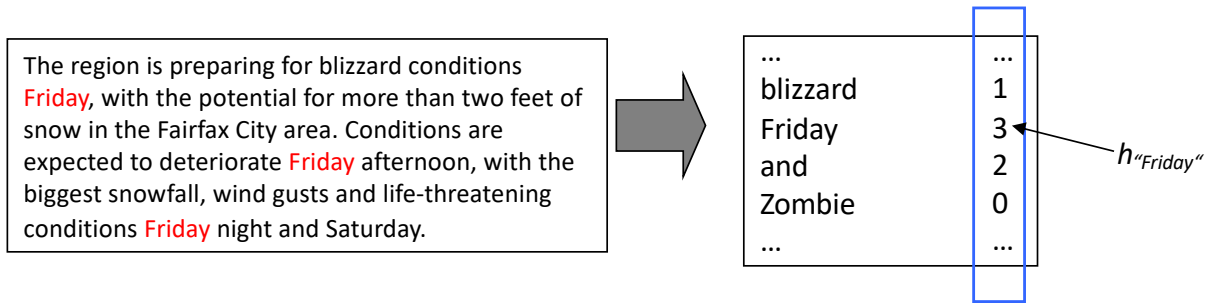
- The proximity function should fit the type of data
  - For dense continuous data, metric distance functions like Euclidean are often used.
  - For sparse data, typically measures that ignore 0-0 matches are employed
    - We care about characteristics that objects share, not about those that both lack
- Domain expertise is important, maybe there is already a state-of-the-art proximity function in a specific domain and we don't need to answer that question again.
- In general, choosing the right proximity measure can be a very time consuming task
- Other important aspects: How to combine proximities for heterogenous attributes (binary and numeric and nominal etc.)
  - e.g., using attribute weights

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data
- Homework/ Tutorial
- Things you should know from this lecture

Feature transformations for text data 1/6

- Text represented as a set of terms ("Bag-Of-Words" model)
  - Terms:
    - Single words ("cluster", "analysis"..)
      or
    - bigrams, trigrams, …n-grams ("cluster analysis"..)
  - Transformation of a document d in a vector $r(d) = (h_1, ..., h_d)$, $h_i \geq 0$: the frequency of term t_i in d



Feature transformations for text data 2/6

- Challenges/Problems in Text Mining:
  1. Common words (“e.g.”, “the”, “and”, “for”, “me”)
  2. Words with the same root (“fish”, “fisher”, “fishing”,…)
  3. Very high-dimensional space (dimensionality $d > 10.000$)
  4. Not all terms are equally important
  5. Most term frequencies $h_i = 0$ (“sparse feature space“)
- More challenges due to language:
  - Different words have same meaning (synonyms)
    - “freedom” – “liberty”
  - Words have more than one meanings
    - e.g. “java”, “mouse”

Feature transformations for text data 3/6

- Problem 1: Common words (“e.g.”, “the”, “and”, “for”, “me”)
  - Solution: ignore these terms (Stopwords)

    There are stopwords list for all languages in WWW.

- Problem 2: Words with the same root (“fish”, “fisher”, “fishing”,…)
  - Solution: Stemming

    Map the words to their root

    - "fishing", "fished", "fish", and "fisher" to the root word, "fish".

    For English, the Porter stemmer is widely used.
    (Porters Stemming Algorithms: http://tartarus.org/~martin/PorterStemmer/index.html)

    Stemming solutions exist for other languages also.

    The root of the words is the output of stemming.

Feature transformations for text data 4/6

- Problem 3: Too many features/ terms
  - Solution: Select the most important features ("Feature Selection")
  - Example: average document frequency for a term
    - Very frequent items appear in almost all documents
    - Very rare terms appear in only a few documents

Ranking procedure:

1. Compute document frequency for all terms t_i :
2. Sort terms w.r.t. $DF(t_i)$ and get $rank(t_i)$
3. Sort terms by score(t_i)= $DF(t_i)$ · rank(t_i)
   e.g. score(t_{23}) = 0.82 · 1 = 0.82
   score(t_{17}) = 0.65 · 2 = 1.3
4. Select the *k* terms with the largest score(t_i)

$$DF(t_i) = \frac{\#Docs\ containing\ t_i}{\#All\ documents}$$

Rank	Term	DF
1.	t_{23}	0.82
2.	t_{17}	0.65
3.	t_{14}	0.52
4.	…	…

Feature transformations for text data 5/6

- Problem 4: Not all terms are equally important
  - Idea: Very frequent terms are less informative than less frequent words. Define such a term weighting schema.
  - Solution: TF-IDF (Term Frequency · Inverse Document Frequency)

    Consider both the importance of the term in the document and in the whole collection of documents.

$$TF(t,d) = \frac{n(t,d)}{\sum_{w \in d} n(w,d)}$$

The relative frequency of term t in d [n(t,d) = # t in d]

$$IDF(t) = \log\left(\frac{|DB|}{|\{d \mid d \in DB \wedge t \in d\}|}\right)$$

Inverse frequency of term t in all DB

$$TF \times IDF = TF(t,d)\,IDF(t)$$

Feature vector with TF IDF : $r(d) = (TF(t_1,d) \cdot IDF(t_1), \ldots, TF(t_n,d) \cdot IDF(t_n))$

Feature transformations for text data 6/6

- Problem 5: for most of the terms $h_i = 0$
  - Euclidean distance is not a good idea: it is influenced by vectors lengths
  - Idea: use more appropriate distance measures

Jaccard Coefficient: Ignore terms absent in both documents

$$d_{Jaccard}(d_1, d_2) = 1 - \frac{|d_1 \cap d_2|}{|d_1 \cup d_2|} = \frac{|\{t|t \in d_1 \wedge t \in d_2\}|}{|\{t|t \in d_1 \vee t \in d_2\}|}$$

Cosine Coefficient: Consider term values (e.g. TFIDF values)

$$d_{cosinus}(d_1, d_2) = 1 - \frac{\langle d_1, d_2 \rangle}{\|d_1\| \cdot \|d_2\|} = 1 - \frac{\sum_{i=0}^{n} (d_{1,i} \cdot d_{2,i})}{\sqrt{\sum_{i=0}^{n} d_{1,i}^2} \cdot \sqrt{\sum_{i=0}^{n} d_{2,i}^2}}$$