Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data

Feature spaces and proximity measures

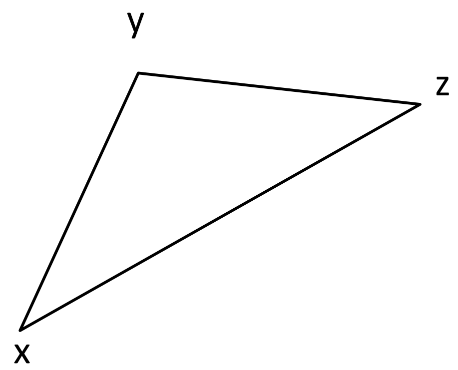
Feature space

- 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≠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.

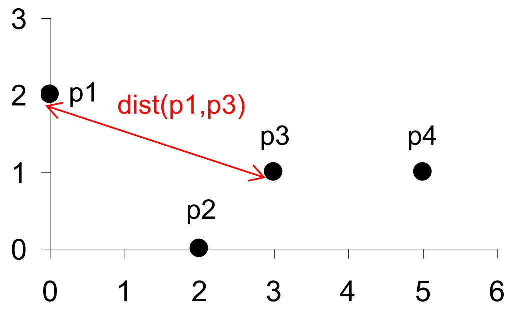
Feature spaces and proximity measures

- 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 (normalization)!
  - 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

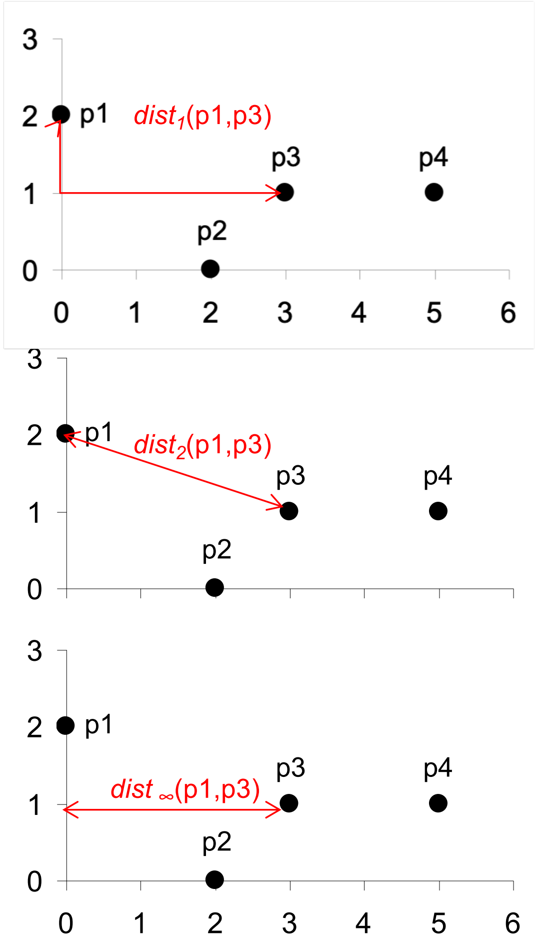


	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance matrix

Feature spaces and proximity measures

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

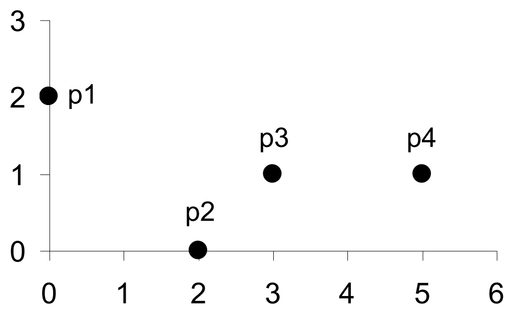


Feature spaces and proximity measures

- 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

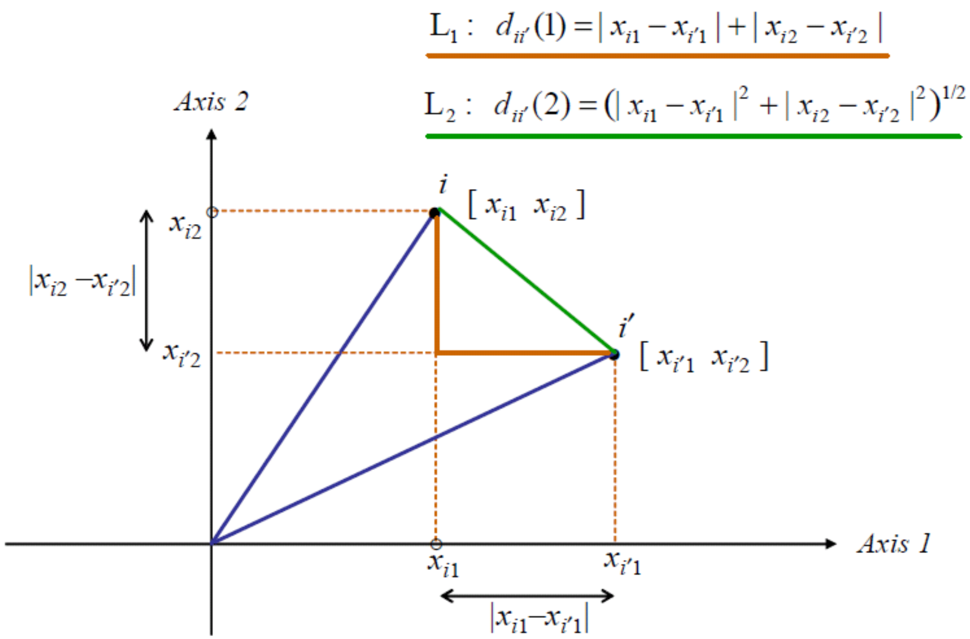
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

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

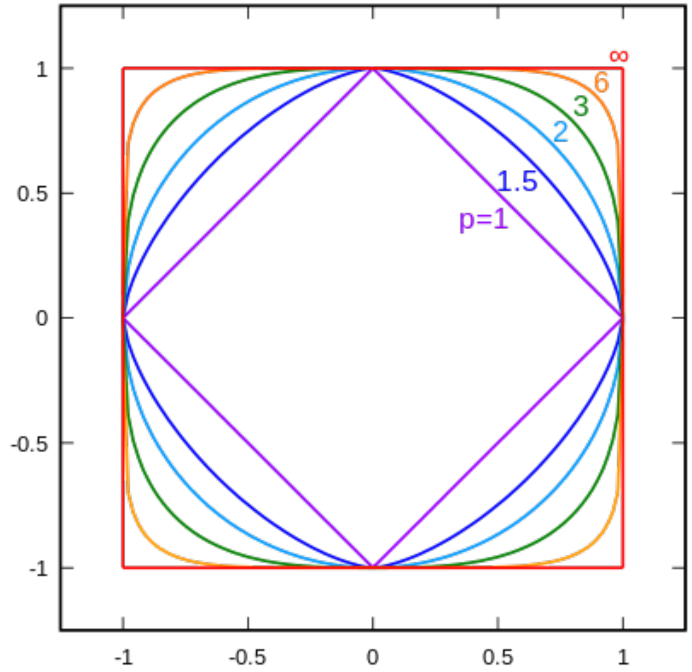
Feature spaces and proximity measures



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 j

Instance i

	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

- <u>Method 2</u>: 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 categorical 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 … but research on this issue is still ongoing.

Outline

- Data preprocessing
- Decomposing a dataset: instances and features
- Basic data descriptors
- Feature spaces and proximity (similarity, distance) measures
- Feature transformation for text data