Machine Learning

Proximity Measures

Claudio Sartori

DISI
Department of Computer Science and Engineering – University of Bologna, Italy
claudio.sartori@unibo.it

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 and Dissimilarity by Attribute type

p and q are the values of an attribute 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 Values mapped to integers 0 to V-1	$d = \frac{\lvert p-q \rvert}{V-1}$	$s = 1 - \frac{\lvert p-q \rvert}{V-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = \frac{1}{1+d}$ or $s = 1 - \frac{d - \min(d)}{\max(d) - \min(d)}$

Euclidean distance – L_2

$$\text{dist} = \sqrt{\sum_{d=1}^{D} (p_d - q_d)^2}$$

- Where D is the number of dimensions (attributes) and p_d and q_d are, respectively, the d-th attributes (components) of data objects p and q
- Standardization/Rescaling is necessary if scales differ

Minkowski distance – L_r

$$\text{dist} = \left(\sum_{d=1}^{D} |p_d - q_d|^r \right)^{\frac{1}{r}}$$

- Where D is the number of dimensions (attributes) and p_d and q_d are, respectively, the d-th attributes (components) of data objects p and q
- Standardization/Rescaling is necessary if scales differ
- r is a parameter which is chosen depending on the data set and the application

Minkowski distance – Cases

$r = 1$ also named city block, Manhattan, L_1 norm

- it is the best way to discriminate between zero distance and near zero distance
- a ϵ change on any coordinate causes a ϵ change in the distance
- works better than euclidean in very high dimensional spaces

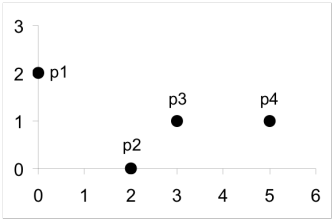
$r = 2$ euclidean, L_2 norm

$r = \infty$ also named Chebyshev, supremum, L_{max} norm, L_∞ norm

- considers only the dimension where the difference is maximum
- provides a simplified evaluation, disregarding the dimensions with lower differences

$$\text{dist}_\infty = \lim_{r \to \infty} \left(\sum_{d=1}^{D} |p_d - q_d|^r \right)^{\frac{1}{r}} = \max_d |p_d - q_d|$$

Minkowski distances – Example



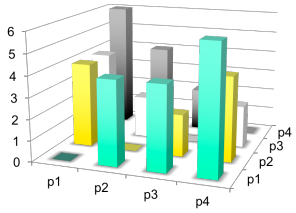
point	*x*	*y*
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L_1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

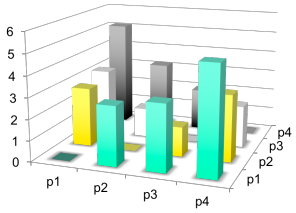
L_2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

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

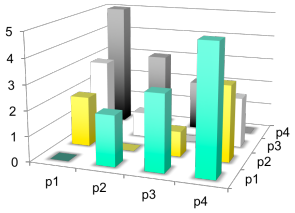
Comparison



L_1



L_2



L_∞

Mahalanobis Distance

- Considers data distribution
- The Mahlanobis distance between two points p and q decreases if, keeping the same euclidean distance, the segment connecting the points is stretched along a direction of greater variation of data
- The distribution is described by the covariance matrix of the data set

$$\Sigma_{ij} = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

$$\text{dist}_m = \sqrt{(p-q)\Sigma^{-1}(p-q)^T}$$

Mahalanobis Distance – Example

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

$$A = (0.5, 0.5)$$

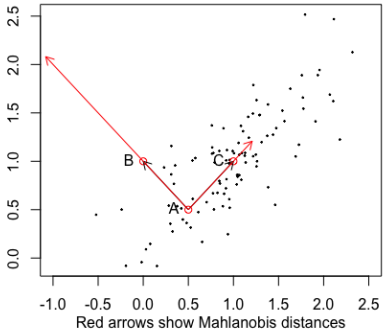
$$B = (0, 1)$$

$$C = (1, 1)$$

The euclidean distances AB and AC are the same

$$\text{dist}_m(A, B) = 2.236068$$

$$\text{dist}_m(A, C) = 1$$



Covariance matrix

- Variation of pairs of random variables
- The summation is over all the observations
- The main diagonal contains the variances
- The values are positive if the two variables grow together
- If the matrix is diagonal the variables are non–correlated
- If the variables are standardised the diagonal contains “one”
- If the variables are standardised and non correlated, the matrix is the identity and the Mahalanobis distance is the same as the euclidean

Common properties of a distance

1. Positive definiteness: $\text{Dist}(p, q) \geqslant 0 \ \forall p, q$ and $\text{Dist}(p, q) = 0$ if and only if $p = q$
2. Symmetry: $\text{Dist}(p, q) = \text{Dist}(q, p)$
3. Triangle inequality: $\text{Dist}(p, q) \leqslant \text{Dist}(p, r) + \text{Dist}(r, q) \forall p, q, r$

A distance function satisfying all the properties above is called a metric

Common properties of a Similarity

1. $Sim(p, q) = 1$ only if $p = q$
2. $Sim(p, q) = Sim(q, p)$

Similarity between binary vectors

- Consider the counts below
  - M_{00} the number of attributes where p is 0 and q is 0
  - M_{01} the number of attributes where p is 0 and q is 1
  - M_{10} the number of attributes where p is 1 and q is 0
  - M_{11} the number of attributes where p is 1 and q is 1
- Simple Matching Coefficient

$$\text{SMC} = \frac{\text{number of matches}}{\text{number of attributes}} = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

- Jaccard Coefficient

$$\text{JC} = \frac{\text{number of 11 matches}}{\text{number of non–both–zero attributes}} = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

Cosine similarity

- It is the cosine of the angle between two vectors

$$\cos(p, q) = \frac{p \cdot q}{\|p\| \|q\|}$$

- Example

$$p = 3\ 2\ 0\ 5\ 0\ 0\ 0\ 2\ 0\ 0$$
$$q = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 2$$
$$p \cdot q = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$
$$\|p\| = \sqrt{3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0} = 6.481$$
$$\|q\| = \sqrt{1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2} = 2.245$$
$$\cos(p, q) = .3150$$

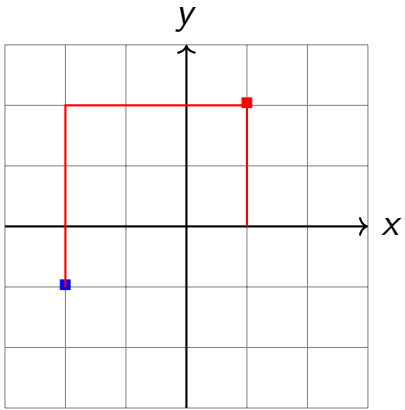
Extended Jaccard Coefficient (Tanimoto)

- Variation of Jaccard for continuous or count attributes
  - reduces to Jaccard for binary attributes

$$\mathrm{T}(p, q) = \frac{p \cdot q}{\|p\|^2 + \|q\|^2 - p \cdot q}$$

Manhattan Distance – Use cases

- Sparse High-Dimensional Data
  - Useful when features are not densely populated and dimensions are independent.
- Grid-Based Systems
  - Applied in grid-like systems, such as in urban planning or robotics (e.g., pathfinding in grid maps).
- Lasso Regression
  - Emphasizes feature selection by shrinking coefficients of less important features to zero.



When to Use Manhattan Distance

- Independent dimensions: When features represent truly separate, uncorrelated dimensions (e.g., urban grid coordinates where diagonal movement is not possible)
- Presence of outliers: Manhattan distance is more robust to outliers than Euclidean distance, as it does not square the differences
- High-dimensional data: In high-dimensional spaces, Euclidean distance may suffer from the curse of dimensionality
- Features with different scales: When variables are on very different scales and you want each dimension to contribute more linearly
- Sparse data: In applications like text mining or recommendation systems with sparse vector representations

Recommendation System Example: Movie Ratings

	Inception	Titanic	Matrix	Avengers	Notebook
User A	5	-	4	-	-
User B	4	2	-	5	-
User C	-	5	-	-	4
User D	5	-	3	4	-

Why Manhattan works better:

- Sparse matrix: users rate only a few movies
- Better handling of missing values
- More robust to extreme ratings

Distance Calculation Example

Comparing User A and User D (common movies: Inception, Matrix):
Manhattan distance:

$$|5-5| + |4-3| = 0 + 1 = 1$$

Euclidean distance:

$$\sqrt{(5-5)^2 + (4-3)^2} = \sqrt{0+1} = 1$$

For vectors [5,1,5,1] and [4,2,4,2]:

- Manhattan: $|5-4| + |1-2| + |5-4| + |1-2| = 4$
- Euclidean: $\sqrt{1^2 + 1^2 + 1^2 + 1^2} = 2$

Manhattan better captures total distance between preferences

Practical Application: Collaborative Filtering

Algorithm:

1. Calculate Manhattan distance from User A to all other users
2. Identify the k nearest neighbors (e.g., k=3)
3. Recommend to A the movies appreciated by neighbors that A hasn't seen yet

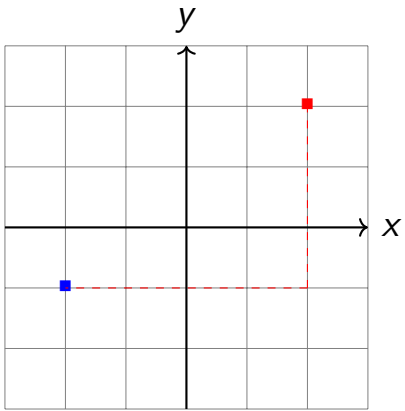
Example:

- If User D is most similar to User A
- Recommend "Avengers" to A (D rated it 4 stars)

This approach works particularly well with Manhattan distance for sparse datasets (Netflix, Spotify, Amazon, etc.)

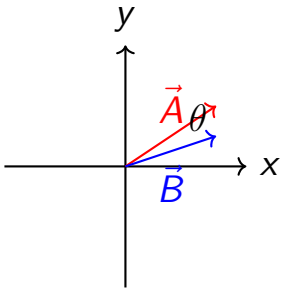
Supremum (Chebyshev) Distance – Use cases

- Anomaly Detection
  - Efficient for detecting extreme deviations across high-dimensional features.
- Chessboard Distance
  - Often used in modeling real-world systems where maximum single-step moves matter (e.g., chessboard).
  - Industrial Applications
    - Applied in quality control, where maximum tolerances are checked for anomalies.



Cosine Similarity – Use cases

- Text Mining and Document Similarity
  - Used for document comparison and recommendation systems to detect contextually similar content.
- Image Similarity
  - Applied in image retrieval systems to match images with similar features.
- Recommendation Systems
  - Collaborative filtering methods often leverage cosine similarity to recommend items.



Manhattan vs Cosine Distance: Key Differences

Manhattan Distance

- Measures absolute differences
- Sensitive to magnitude
- Formula: $\sum_{i=1}^{n} |x_i - y_i|$
- Range: $[0, \infty)$

Cosine Similarity

- Measures angle/direction
- Magnitude-invariant
- Formula: $\frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \cdot ||\vec{y}||}$
- Range: [-1, 1]

Example 1: Movie Ratings - When Magnitude Matters

Three users rate movies on scale 1-5:

	Movie 1	Movie 2	Movie 3	Profile
Alice	5	5	5	Loves everything
Bob	3	3	3	Moderate ratings
Carol	5	5	4	Loves most things

Manhattan Distance:

- Alice-Bob: $|5-3| + |5-3| + |5-3| = 6$
- Alice-Carol: $|5-5| + |5-5| + |5-4| = 1$
- Bob-Carol: $|3-5| + |3-5| + |3-4| = 5$

Result: Alice is closest to Carol (both are enthusiastic raters)

Example 1: Cosine Similarity Perspective

Same data:

Cosine Similarity:

- Alice-Bob: $\frac{5(3)+5(3)+5(3)}{\sqrt{75}\cdot\sqrt{27}} = \frac{45}{45} = 1.0$
- Alice-Carol: $\frac{5(5)+5(5)+5(4)}{\sqrt{75}\cdot\sqrt{66}} = 0.995$
- Bob-Carol: $\frac{3(5)+3(5)+3(4)}{\sqrt{27}\cdot\sqrt{66}} = 0.995$

Result: All users are highly similar! Cosine ignores that Bob rates lower overall, focusing only on *relative preferences*.

Use Manhattan when rating scale/magnitude matters

Example 2: Document Similarity - When Direction Matters

Three documents represented by word frequencies:

	"machine"	"learning"	"algorithm"	"data"
Doc A	10	8	12	15
Doc B	2	1	3	2
Doc C	5	12	3	8

Observation: Doc B is just a shorter version of Doc A (same topic, fewer words)

Manhattan Distance:

- A-B: $|10-2| + |8-1| + |12-3| + |15-2| = 8 + 7 + 9 + 13 = 37$
- A-C: $|10-5| + |8-12| + |12-3| + |15-8| = 5 + 4 + 9 + 7 = 25$

Result: Doc C seems closer to A than B (misleading!)

Example 2: Cosine Captures True Similarity

Cosine Similarity:

- A-B: $\frac{10(2)+8(1)+12(3)+15(2)}{\sqrt{533}\cdot\sqrt{18}} = \frac{94}{98} = 0.959$
- A-C: $\frac{10(5)+8(12)+12(3)+15(8)}{\sqrt{533}\cdot\sqrt{242}} = \frac{282}{359} = 0.785$

Result: Doc B is more similar to A (correctly identifies same topic despite length difference)

Use Cosine when you care about proportional similarity, not absolute counts

Common applications:

- Text classification
- Information retrieval
- Recommending articles/documents

Example 3: User Behavior Vectors

Two users' activity on a website (clicks per section):

	News	Sports	Tech	Entertainment
User X	100	80	60	40
User Y	10	8	6	4

Scenario: User Y is a casual visitor, User X is a power user

- Manhattan distance: 270 (very large - treats them as different)
- Cosine similarity: 1.0 (perfect - same interest distribution!)

Decision:

- Use Manhattan if engagement level matters (e.g., detecting power users)
- Use Cosine if interest pattern matters (e.g., content recommendations)

Quick Decision Guide

Scenario	Manhattan	Cosine
Absolute values matter	✓	
Scale/magnitude important	✓	
Different-length documents		✓
Interest patterns		✓

Rule of thumb:

- Manhattan: "How different are the values?"
- Cosine: "How similar are the patterns?"

Jaccard Similarity

- Definition: Measures overlap between two sets relative to their union.
- Use Cases
  - Document Similarity in Information Retrieval
    - Applied in detecting plagiarism or duplicate documents.
  - Image Processing
    - Measures similarity in object recognition and segmentation.
  - Clustering and Community Detection
    - Useful in social network analysis to find communities based on shared elements.