Vector Norms and Proximity Measures

- Vector norms and Unit Circles
- Proximity Measures
- Distance between non-numeric vectors
- Distance between mixed feature vectors
- Distance between non-vectors data objects

Imdad ullah Khan

Vector Norms and Unit Circles

Vector Norms: Error Measurements

- Let $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T \in \mathbb{R}^n$
- compare its competing estimates $\mathbf{y} = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^T$ and $\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix}^T$
- Error vectors $\mathbf{e}_y = \mathbf{x} - \mathbf{y} = \begin{bmatrix} x_1 - y_1 \\ \vdots \\ x_n - y_n \end{bmatrix}$ and $\mathbf{e}_z = \begin{bmatrix} x_1 - z_1 \\ \vdots \\ x_n - z_n \end{bmatrix}$
- e.g. $\mathbf{e}_y = \begin{bmatrix} 10 & -10 & 10 & 20 \end{bmatrix}$ and $\mathbf{e}_z = \begin{bmatrix} 20 & -5 & 0 & 20 \end{bmatrix}$
- Need to map $\mathbf{e}_y$ and $\mathbf{e}_z$ to real numbers and compare
- Compare lengths $\|\mathbf{e}_y\| = \sqrt{10^2 + (-10)^2 + 10^2 + 20^2} = 26.45, \; \|\mathbf{e}_z\| = 28.72$
- Since smaller are better, $\mathbf{y}$ is a better estimate of $\mathbf{x}$
- One can argue that with a different mapping, $\mathbf{z}$ is better
  $\|\mathbf{e}_y\|_1 = |10| + |-10| + |10| + |20| = 50, \; \|\mathbf{e}_z\|_1 = |20| + |-5| + |0| + |20| = 45$
- Note the absolute value sign $\because$ error on either side is bad

Vector Norms

- No universally good mapping of vectors to numbers
- A norm is an operation or a function mapping vectors to real numbers
- Denote the norm of a vector $\mathbf{v}$ by $\|\mathbf{v}\|$
- Norms measure the "size" or "length" or "magnitude" of vectors
- Norms satisfy **3 axioms** ▷ Expected from any *'magnitude'*

1. $\|\mathbf{v}\| \geq 0$, $\|\mathbf{v}\| = 0 \leftrightarrow \mathbf{v} = 0$ ▷ non-negativity
2. $\|c\mathbf{v}\| = c\|\mathbf{v}\|$ ▷ scaling
   - length of 2 times a vector should be double, also works for -2
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ ▷ triangle inequality
   - length of a triangle side should be less than sum of other two

Vector Norms

- ℓ_1**-norm:** $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ ▷ one/mean norm

- ℓ_2**-norm:** $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ ▷ Euclidean/mean-square norm

- ℓ_p**-norm:** $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$ ▷ p-norm

- ℓ_∞**-norm:** $\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,n}\{|x_i|\}$ ▷ infinity/max norm

Vector Norms

- **ℓ_1-norm:** $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ ▷ one/mean norm
- **ℓ_2-norm:** $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$ ▷ Euclidean/mean-squares norm
- **ℓ_p-norm:** $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$ ▷ p-norm
- **ℓ_∞-norm:** $\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,n}\{|x_i|\}$ ▷ infinity/max norm

$$\mathbf{x} = \begin{bmatrix} 10 \\ -5 \\ 0 \\ 20 \end{bmatrix}$$

$$\begin{aligned}
\|\mathbf{x}\|_1 &= 10 + 5 + 0 + 20 &&= 35 \\
\|\mathbf{x}\|_2 &= \sqrt[2]{100 + 25 + 0 + 400} &&= 22.91 \\
\|\mathbf{x}\|_3 &= \sqrt[3]{10^3 + 5^3 + 0^3 + 20^3} &&= 20.896 \\
\|\mathbf{x}\|_4 &= \sqrt[4]{10^4 + 5^4 + 0^4 + 20^4} &&= 20.324 \\
\|\mathbf{x}\|_5 &= \sqrt[5]{10^5 + 5^5 + 0^5 + 20^5} &&= 20.1272 \\
\|\mathbf{x}\|_6 &= \sqrt[6]{10^6 + 5^6 + 0^6 + 20^6} &&= 20.0525 \\
\|\mathbf{x}\|_7 &= \sqrt[7]{10^7 + 5^7 + 0^7 + 20^7} &&= 20.022 \\
\|\mathbf{x}\|_\infty &= \max\left\{10, 5, 0, 20\right\} &&= 20
\end{aligned}$$

Vector Norms

- ℓ_1**-norm:** $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ $\quad\triangleright$ one/mean norm
- ℓ_2**-norm:** $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ $\quad\triangleright$ Euclidean/mean-squares norm
- ℓ_p**-norm:** $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$ $\quad\triangleright$ p-norm
- ℓ_∞**-norm:** $\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,n}\{|x_i|\}$ $\quad\triangleright$ infinity/max norm

- As p grows the effect of smaller terms diminishes
- When $p \to \infty$, we just get the maximum coordinate of the vector
- If many small errors can be tolerated but any significant error is bad, try to minimize the higher norms of errors
- If all errors are bad (in critical apps), then minimize ℓ_1 norms of error
- If all variables/coordinates are not in same scale and unit, then some coordinates dominate values of norms

$$\text{For} \quad p < q \qquad \|\mathbf{x}\|_p \geq \|\mathbf{x}\|_q$$

Unit Circles w.r.t Vector Norms

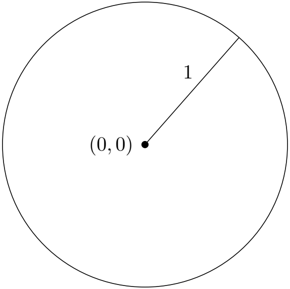
Understanding the geometric implications of L_p norms through unit circles allows us to appreciate the diversity of distance measures in various contexts

Applications

- **Data Science:** Different norms are used for various clustering, regression, and classification techniques
- **Optimization:** Norms influence the behavior of optimization algorithms by defining how distances are measured

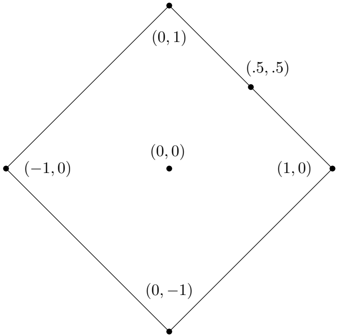
Unit Circles w.r.t Vector Norms

- **Unit Circle:** Set of all points with distance from origin equal to 1
- The (infinite) set of 2-d vectors with ℓ_2 norm equal to 1
- $\left\{\mathbf{x} \in \mathbb{R}^2 \ : \ \|\mathbf{x}\|_2 = 1\right\}$
- Plot of this set is a perfect circle



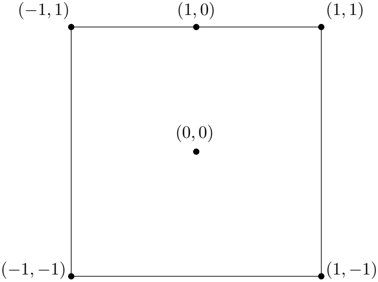
Unit Circles w.r.t Vector Norms

- The (infinite) set of 2-d vectors with ℓ_1 norm equal to 1
- $\left\{\mathbf{x} \in \mathbb{R}^2 \;:\; \|\mathbf{x}\|_1 \;=\; 1\right\}$
- $\|\mathbf{x}\|_1 \;=\; 1$ for all points $\left[z \;\; (1-|z|)\right]^T$
- e.g. $\left[1\ 0\right]^T, \left[0\ 1\right]^T, \left[.5\ .5\right]^T, \left[-.5\ -.5\right]^T, \left[-.8\ .2\right]^T \left[.7\ .3\right]^T$
- Plot of this set is a axis-aligned unit-square rotated by 45°



Unit Circles w.r.t Vector Norms

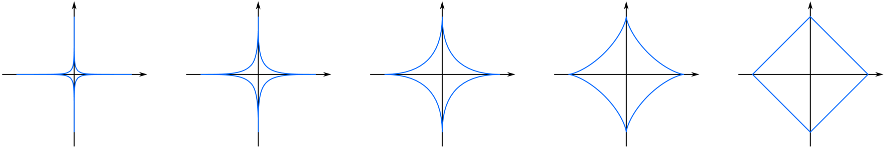
- The (infinite) set of 2-d vectors with ℓ_∞ norm equal to 1
- $\{\mathbf{x} \in \mathbb{R}^2 \; : \; \|\mathbf{x}\|_\infty = 1\}$
- $\|\mathbf{x}\|_\infty = 1$ for all points $\begin{bmatrix}1 \; z\end{bmatrix}^T$ and $\begin{bmatrix}z \; 1\end{bmatrix}^T$
- e.g. $\begin{bmatrix}1 \; 0\end{bmatrix}^T, \begin{bmatrix}1 \; .2\end{bmatrix}^T, \begin{bmatrix}1 \; .5\end{bmatrix}^T, \begin{bmatrix}-1 \; -.5\end{bmatrix}^T, \begin{bmatrix}-.8 \; 1\end{bmatrix}^T$
- Plot of this set is an axis-aligned unit square



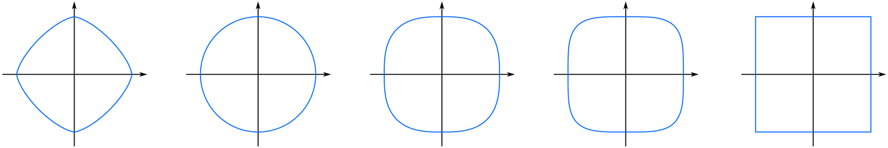
Unit Circles w.r.t Vector Norms

Play around with unit circles w.r.t. ℓ_p norms

image source: Wikipedia



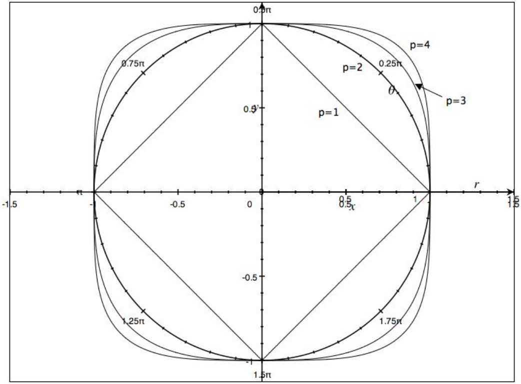
$p = 2^{-2} = 0.25$ | $p = 2^{-1.5} = 0.354$ | $p = 2^{-1} = 0.5$ | $p = 2^{-0.5} = 0.707$ | $p = 2^{0} = 1$



$p = 2^{0.5} = 1.414$ | $p = 2^{1} = 2$ | $p = 2^{1.5} = 2.828$ | $p = 2^{2} = 4$ | $p = 2^{\infty} = \infty$

Unit Circles w.r.t Vector Norms

Play around with unit circles w.r.t. ℓ_p norms



Unit Circles w.r.t Vector Norms

Different values of p lead to distinct shapes of unit circles, reflecting how the norm measures distances

Observations

- L_1 norm (Manhattan): Forms a diamond shape, emphasizing direct paths along axes.
- L_2 norm (Euclidean): The familiar circular shape, measuring straight-line distance.
- L_∞ norm (Max Norm): Forms a square, where the maximum coordinate value dictates the distance.

These shapes provide insights into how different norms penalize deviations from the origin.

Matrix Norms

Matrix Norms map matrices to real number so one can compare matrices

Matrix norms are tools used to measure the size or length of matrices

Key Properties: Non-negativity, scalability, and triangle inequality.

Matrix norms are widely used in various applications, including:

- **Numerical Stability:** Determining the stability of algorithms, particularly in solving systems of linear equations
- Fundamental in numerical analysis, particularly numerical linear algebra and optimization
- **Error Analysis:** Measuring how perturbations in input affect the solution
- **Control Theory:** Analyzing and designing control systems

Commonly Used Matrix Norms

Matrix norms can be broadly categorized into vector norms applied to matrices and norms based on singular values

Examples of Matrix Norms

- **Frobenius Norm:** Square root of the sum of the absolute squares of its elements.
- **Spectral Norm (or L_2 Norm):** Largest singular value of the matrix.
- **Induced L_1 and L_∞ Norms:** Maximum absolute column sum and maximum absolute row sum, respectively.

Comparing Distributions

Comparing distributions is fundamental in statistics and data science, helps understand how a dataset differs from another or from a theoretical model

Motivation and Applications

- Identify differences in populations, assess model fit, or track changes over time
- ML model evaluation, hypothesis testing, and feature selection

Some commonly used norms and distributions are

- K-L divergence (and its variations)
- entropy
- diversity index
- moments

Search the literature on this! I mentioned it so you are not scared if you see something like this in your papers or need it in your projects

Proximity Measures

Proximity Measures

Proximity measures help determine how close or similar data objects are to each other, which is fundamental in many machine learning algorithms

- **Distance Measures:** Defines the dissimilarity between objects
- **Similarity Measures:** Quantify how similar two data objects are
- Examining data for similar items is fundamental to data analytics
- Note that finding the same items is easy
- Finding almost same is hard and is more frequently needed
  - Plagiarism detection (whole documents might not be same)
  - Finding mirror pages (headers could be different)
  - Establishing articles from same source (different news outlet might add additional things such as web address, different accompanying ads)
- Data analytics require a notion of similarity and then deal with its computational issues

Analytics Require Proximity Measures

Notion of proximity is **fundamental** to data analytics

Almost all other topics cannot even be discussed without a notion and understanding of similarity/distance

- Classification/Clustering Group similar items
- Recommendation Systems: recommend item j to user i if users alert"similar" to i like items "similar" to j
- Outlier Detection: identify dissimilar items
- Nearest Neighbor Search: find the most similar objects to the query object
- Locality Sensitive Hashing: "similar" items go to same bucket
- Reduce dimensionality while preserving "similarities"

Distance, Similarity and Proximity

- Similarity
  - Quantitative measure of how similar/alike are two objects
  - Usually falls in the range $[0, 1]$ or is scaled to this range
  - Higher values means objects are more similar
  - Maximum value for the same object
- Dissimilarity (e.g. distance)
  - Quantitative measure of how dissimilar/different are two objects
  - Minimum value is usually 0
  - Lower values means objects are more similar
  - Minimum value for the same object
  - The "inverse" of similarity
- Proximity
  - Refers to similarity or dissimilarity

Distance Measures for Nominal Vectors

Distance Measures and Distance Metric

- We mainly discuss distance measures (similarity is its "inverse")
- A **distance function** takes two objects and returns a real number
- A distance function is a **distance metric** if it satisfies 4 axioms

1. $d(u, v) \geq 0$ ▷ non-negativity
   - it doesn't make sense to have distance of -3
2. $d(u, v) = 0 \iff u = v$ ▷ indiscernibility
3. $d(u, v) = d(v, u)$ ▷ symmetry
4. $d(u, w) \leq d(u, v) + d(v, w)$ ▷ triangle inequality
   - direct distance is shorter than the distance via an intermediate point

Vectors in Non-Euclidean Space

- We discuss in detail proximity between vectors
- First we discuss when data items are nominal/ordinal feature vectors
- Then we talk about distance between two data items/objects described by their numeric attributes (considered vectors in $\mathbb{R}^n$)
- Then we discuss when data items are mixed feature vectors
- Then we discuss when data items are not vectors at all
  - e.g. sets, bags, strings, sequences
- We cover how to covert non-vector data to vector forms later

Distance Measures for Nominal Vectors

Distance measures for nominal vectors enable quantitative analysis of categorical data in clustering, classification, and association rule mining

Used in market basket analysis, genetic data processing, and text clustering

Some distance measures for handling nominal data are

- **Simple Matching Distance (SMD):** Counts the proportion of mismatches between two vectors.
- **Hamming Distance:** Measures the number of positions at which the corresponding entries are different.
- **Jaccard Distance:** Based on the number of common attributes divided by the number of attributes that appear in at least one of the two objects.

Simple Matching Distance

Let $\mathbf{o}_i$ and $\mathbf{o}_j$ be two objects with n nominal/categorical attributes

If m out of the n attributes have equal values for $\mathbf{o}_i$ and $\mathbf{o}_j$, then

$$d(\mathbf{o}_i, \mathbf{o}_j) := d(i,j) = \frac{n-m}{n} \qquad \text{and} \qquad sim(i,j) = \frac{m}{n}$$

St.ID	Gender	City	School
$\mathbf{s}_1$	M	PSH	SDSB
$\mathbf{s}_2$	F	LHR	SS
$\mathbf{s}_3$	M	KCH	LAW
$\mathbf{s}_4$	M	KCH	SSE
$\mathbf{s}_5$	F	LHR	SDSB
$\mathbf{s}_6$	F	MTN	LAW
$\mathbf{s}_7$	M	KCH	SSE
$\mathbf{s}_8$	F	FSD	SSE

- $d(1,2) = {}^{3-0}/_{3} = 1$
- $d(1,3) = {}^{3-1}/_{3} = {}^{2}/_{3}$
- $d(2,7) = {}^{3-0}/_{3} = {}^{2}/_{3}$
- $d(3,4) = {}^{3-2}/_{3} = {}^{1}/_{3}$
- $d(1,1) = {}^{3-3}/_{3} = 0$
- $d(2,2) = {}^{3-3}/_{3} = 0$
- $d(3,3) = {}^{3-3}/_{3} = 0$

Symmetric Binary Feature Vectors

Symmetric binary values are equally valuable and carry the same weight

Let contingency table of $\mathbf{o}_i$ and $\mathbf{o}_j$ with n symmetric binary attributes be

		$\mathbf{o}_j$		
		TRUE	FALSE	
$\mathbf{o}_i$	TRUE	p	q	$p+q$
	FALSE	r	s	$r+s$
		$p+r$	$q+s$	$p+q+r+s=n$

- $d(i,j) = \dfrac{q+r}{p+q+r+s} = \dfrac{q+r}{n}$ (fraction of variables with different values)
- $sim(i,j) = 1 - d(i,j) = \dfrac{p+s}{p+q+r+s}$ (fraction of variables with same values)

Asymmetric Binary Feature Vectors

Asymmetric binary values are not of equal importance, e.g. Positive and negative outcomes for any medical test (Positive - 1, Negative - 0)

Let contingency table of $\mathbf{o}_i$ and $\mathbf{o}_j$ with n asymmetric binary attributes be

		$\mathbf{o}_j$		
		TRUE	FALSE	
$\mathbf{o}_i$	TRUE	p	q	$p+q$
	FALSE	r	s	$r+s$
		$p+r$	$q+s$	$p+q+r+s=n$

- Then typically distance is defined as $d(i,j) \;=\; \dfrac{q+r}{p+q+r}$
- Both FALSE (agreement on false) does not matter

Symmetric/Asymmetric Binary Feature Vectors Example

- Consider LUMS RO data
- Pass/fail status of students with different majors in different courses
- Similarity captures similarity of the students' majors

		Courses										
ID	Major	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c'
$\mathbf{s}_1$	CS	1	0	1	1	1	0	0	0	1	0	0
$\mathbf{s}_2$	CS	1	1	1	1	1	0	1	0	0	0	0
$\mathbf{s}_3$	CS	0	1	1	1	1	1	1	0	0	0	0
$\mathbf{s}_4$	EE	1	0	1	0	0	1	1	1	0	0	0
$\mathbf{s}_5$	EE	1	1	0	0	0	1	1	1	0	0	0
$\mathbf{s}_6$	EE	1	1	0	1	0	1	1	1	1	0	0
$\mathbf{s}_7$	EC	1	1	1	0	0	0	0	0	1	1	1
$\mathbf{s}_8$	EE	1	1	0	0	0	0	0	0	1	1	1
$\mathbf{s}_9$	EE	1	0	0	0	0	0	1	0	1	1	1

Symmetric/Asymmetric Binary Feature Vectors Example

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$	$\mathbf{s}_6$	$\mathbf{s}_7$	$\mathbf{s}_8$	$\mathbf{s}_9$
$\mathbf{s}_1$	11	8	6	5	3	5	6	5	5
$\mathbf{s}_2$	8	11	9	6	6	6	5	4	4
$\mathbf{s}_3$	6	9	11	6	6	6	3	2	2
$\mathbf{s}_4$	5	6	6	11	9	7	4	3	5
$\mathbf{s}_5$	3	6	6	9	11	9	4	5	5
$\mathbf{s}_6$	5	6	6	7	9	11	4	5	5
$\mathbf{s}_7$	6	5	3	4	4	4	11	10	8
$\mathbf{s}_8$	5	4	2	3	5	5	10	11	9
$\mathbf{s}_9$	5	4	2	5	5	5	8	9	11

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$	$\mathbf{s}_6$	$\mathbf{s}_7$	$\mathbf{s}_8$	$\mathbf{s}_9$
$\mathbf{s}_1$	5	4	3	2	1	3	3	2	2
$\mathbf{s}_2$	4	6	5	3	3	4	3	2	2
$\mathbf{s}_3$	3	5	6	3	3	4	2	1	1
$\mathbf{s}_4$	2	3	3	5	4	4	2	1	2
$\mathbf{s}_5$	1	3	3	4	5	5	2	2	2
$\mathbf{s}_6$	3	4	4	4	5	7	3	3	3
$\mathbf{s}_7$	3	3	2	2	2	3	6	5	4
$\mathbf{s}_8$	2	2	1	1	2	3	5	5	4
$\mathbf{s}_9$	2	2	1	2	2	3	4	4	5

Pairwise similarity matrix with binary attributes symmetric(left) and asymmetric (right)

- Consider $\mathbf{s}_1$ and $\mathbf{s}_7$ with different majors i.e CS and EC
- $sim(\mathbf{s}_1, \mathbf{s}_7) = 6/9$ in symmetric binary case because courses not taken by both are also adding value to similarity (it shouldn't be the case)
- $sim(\mathbf{s}_1, \mathbf{s}_7) = 3/9$ in asymmetric binary case (FALSE to FALSE match is not adding any value)

Encoding is used to convert nominal/categorical variable to numeric

- Nominal/Categorical variables are converted into numeric because most algorithm work on numeric values
- For example, the car company datasets with their prices and names

Name	**Price**
VW	20000
Acura	10011
Honda	50000
Honda	10000

Car Dataset

LABEL-ENCODING

Encoding is used to convert nominal/categorical variable to numeric

Cannot give '*numerical values*' to categorical attributes

Name	Numeric Value	Price
VW	1	20000
Acura	2	10011
Honda	3	50000
Honda	3	10000

- This organization presupposes that categorical values are *VW* $<$ *Acura* $<$ *Honda* (higher the numeric value better the category)
- Say your algorithm internally calculates average
- This implies that: Average of VW and Honda is Acura
- Analytics on this data would be meaningless

One-Hot-Encoding

One-hot-encoding "binarizes" each category (each level of value)

- 0 indicates non existing, 1 indicates existing

Name	**Price**
VW	20000
Acura	10011
Honda	50000
Honda	10000

VW	**Acura**	**Honda**	**Price**
1	0	0	20000
0	1	0	10011
0	0	1	50000
0	0	1	10000

- A non-binary nominal attribute can be converted into many binary variables and the distance measure is applied
- Each category has equal weight
- This works well if the number of levels (categories) is not very large

Distance Measures for Numeric Vectors

Numeric Feature Vectors

Distance and similarity measures are crucial for many applications in data science, including clustering, classification, and anomaly detection

Objective: Understand how different measures capture the notion of distance and similarity between numeric vectors

- Many distance measures for objects described by n numeric attributes
- **Key Measures:** Manhattan, Euclidean, Minkowski, Supremum, and Cosine.
- We use one or the other measure depending on the applications and after a certain amount of trial and error

Numeric Feature Vectors: Euclidean distance

Euclidean distance: most well-known and common distance measure

It has several nice qualities that make it very useful

- Generally applicable
- Nice geometric interpretation (straight line distance between points)
- Can do many geometric and algebraic operations on Euclidean vectors

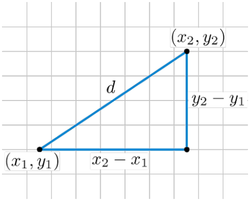
Scale of all coordinate should be the same and they should have the same units (or unitless)

Numeric Feature Vectors: Euclidean distance

The most natural and commonly used distance measure is the Straight line distance, Euclidean Distance, or ℓ_2 distance

For Numeric vectors: $\mathbf{x}_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ and $\mathbf{x}_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_n})$ (points in $\mathbb{R}^n$), their distance is length of the line segment joining them (shortest distance between them) ▷ The ℓ_2 norm of the difference vector

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i_1} - x_{j_1})^2 + \ldots + (x_{j_n} - x_{j_n})^2} = \sqrt{\sum_{k=1}^{n} (x_{i_k} - x_{j_k})^2}$$

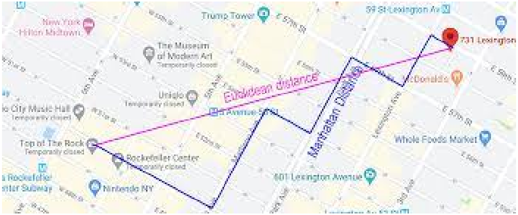


Manhattan distance is sum of coordinate wise absolute differences

$$d(\mathbf{x}_i, \mathbf{x}_j) \; = \; \sum_{k=1}^{n} |x_{i_k} - x_{j_k}|$$

Corresponds to the number of blocks one has to travel while moving from $\mathbf{x}_i$ to $\mathbf{x}_j$ in a city like Manhattan (a grid)

Also called ℓ_1 distance, as it is ℓ_1 norm of the difference vector



Numeric Feature Vectors: Minkowski distance

Minkowski or ℓ_p distance generalizes the above two measures.

The ℓ_p norm of the difference vector

$$d(\mathbf{x}_i, \mathbf{x}_j) \; = \; \sqrt[p]{\sum_{k=1}^{n} |x_{i_k} - x_{j_k}|^p}$$

For different values of p, this distance behaves differently. Commonly,

- $p = 1$ (Manhattan)
- $p = 2$ (Euclidean)
- $p \to \infty$ (Supremum)

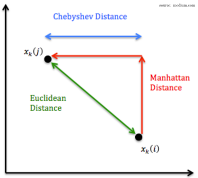
Numeric Feature Vectors: Supremum Distance

A special case of the Minkowski distance when p approaches ∞.

▷ Also called Chebychev distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \lim_{p \to \infty} \sqrt[p]{\sum_{k=1}^{n} |x_{i_k} - x_{j_k}|^p}$$

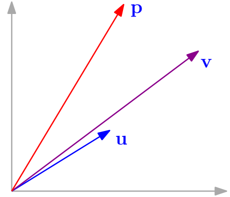
$$d(\mathbf{x}_i, \mathbf{x}_j) = \lim_{p \to \infty} \sqrt[p]{\sum_{k=1}^{n} |x_{i_k} - x_{j_k}|^p} = \max_k \{|x_{i_k} - x_{j_k}|\}$$

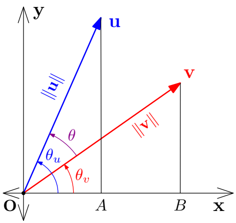


Useful in infinity norm scenarios where greatest difference dominates others

- Cosine distance is good for discrete versions of Euclidean space
- Ignores vector magnitudes; distance is based on their directions
  - A vector is the same as a unit vector in its direction
- Cosine distance is the angle between two vectors in range $[0^\circ - 180^\circ]$
- Calculate by first computing the cosine of angle between two vectors
- Then compute the arc-cosine to get the angle

$$\cos\theta = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\|\mathbf{u}\|\|\mathbf{v}\|} = \left(\frac{\mathbf{u}}{\|\mathbf{u}\|}\right)^T \left(\frac{\mathbf{v}}{\|\mathbf{v}\|}\right)$$



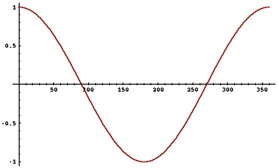


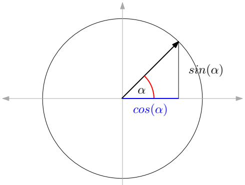
Numeric Feature Vectors: Cosine Distance

- $\mathbf{x} = [7, 2, 3], \qquad \mathbf{y} = [5, 0, -2]$
- $cos(\theta) \;=\; {}^{\mathbf{x}\cdot\mathbf{y}}/_{\|\mathbf{x}\|\|\mathbf{y}\|} \;=\; {}^{(7*5)+(2*0)+(3*(-2))}/_{\|\mathbf{x}\|\|\mathbf{y}\|} = {}^{29}/_{\sqrt{62}*\sqrt{29}}$
- $d(\mathbf{x}, \mathbf{y}) \;=\; \text{arccos}(0.684) \;=\; 46.8^{\circ}$
- 2 vectors in any space define a plane in which the angle is measured
- Vectors taking only $+ve$ values (e.g. TF-IDF) $\implies d_{cos} \in [0^{\circ} - 90^{\circ}]$
- Cosine distance is a distance metric (if we only consider unit vectors)

Numeric Feature Vectors: Cosine Distance

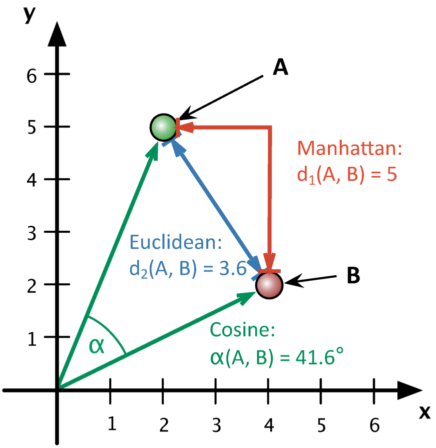
- In some texts the cosine of the angle is taken as a similarity measure
- i.e. $sim(\mathbf{u}, \mathbf{v}) \ = \ cos(\theta_{\mathbf{uv}}) \ = \ {}^{\mathbf{u} \cdot \mathbf{v}}/_{\|\mathbf{u}\|\|\mathbf{v}\|}$
- In general case range of similarity is $[-1, 1]$
- If coordinate values are only positive, then range is $[0, 1]$
  - $sim(\mathbf{u}, \mathbf{v})$ is monotonically decreasing in the interval $[0^\circ, 180^\circ]$
  - $sim(\mathbf{u}, \mathbf{v}) = 1$, if $\mathbf{u}$ and $\mathbf{v}$ are co-linear
  - $sim(\mathbf{u}, \mathbf{v}) = 0$, if $\mathbf{u}$ and $\mathbf{v}$ are orthogonal
  - $sim(\mathbf{u}, \mathbf{v}) < 0$, if $\theta_{\mathbf{uv}} > 90^\circ$
- In this case the cosine distance is defined to be $1 - sim_{cos}(\mathbf{x}, \mathbf{y})$





Observe that the length of base ($cos(\alpha)$) ranges from 1 to -1 when we rotate the unit vector

Numeric Feature Vectors: Comparison



Scales of attributes/coordinates

- Attributes with wider range or smaller units will dominate
- Attributes should be on the same scale (or same units)
- Students with their performance in two courses

  If for one course we report score out of 100 and for the other course we report scores out of 10

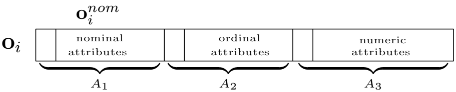
  The effect of the latter course would be negligible in distances (needed for example for grouping students by majors or selecting the top student)

Ordinal Feature Vectors

- Values of ordinal attributes have meaningful order among themselves, but cannot quantify the difference between values
- A common approach is to map values of ordinal attributes to numeric (discrete) values and then use any the above distances
- Here x_{i_k} is mapped to corresponding rank of the value of x_{i_k}.
- Some examples include mapping Freshman, Sophomore, Junior, and Senior to $1, 2, 3$ and 4, respectively
- Grades A, B, C, D, F to $5, 4, 3, 2, 1$, respectively
- Since different attributes can have different levels, normalize the mapped values to a common scale of say $[0, 1]$

Mixed Feature Vectors

- Objects described by n attributes of different types
- Compute distances by groups of attributes, distance is their "average"
- Let A_1, A_2, A_3 be sets of nominal, ordinal and numeric attributes



- Let $\mathbf{o}_i^{nom}$: $\mathbf{o}_i$ as described by nominal attributes (columns of A_1)
- Assuming there are no missing values, then

$$d(i,j) \;=\; \frac{1}{n}\left[|A_1| \cdot d(\mathbf{o}_i^{nom}, \mathbf{o}_j^{nom}) + \sum_{k \in A_2 \cup A_3} |o_{i_k} - o_{j_k}| \right]$$

- Assumed ordinal attributes are converted into numeric
- Numeric and ordinal attributes should be scaled to the interval $[0,1]$, otherwise they will dominate this distance

Distance Measures for Non-Vector Data

Non-Vector Data

- Some data is not directly described by values of attributes
- i.e. not given as feature vectors
  - Text, sequences, images, videos, time-series data, streams
- We first convert data into some kind of feature vectors
  - Feature Engineering, Feature Mapping, Feature Extraction
- Then apply methods and models developed for vectors

Strings and Sequences

- In many cases data is in the form of string or sequences
- Distance between strings/sequences helps infer the similarity, which is needed for all kind of analytics

Hamming Distance

The number of positions with different characters in two strings

The Hamming distance between

- “karolin” and “kathrin” is **3**
- 1011101 and 1001001 is **2**

Hamming distance $= 3$

0	0	0	1	0	1	0	1	1	0	1
0	1	0	1	0	0	0	1	1	1	1

Applications

- Error correcting codes, communication, information theory
- Computational biology, bioinformatics

Limitations

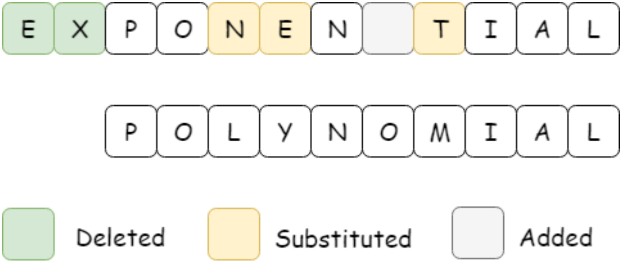
- Only works for sequences of equal length (strings)
- Count all mismatches as equal

Hamming similarity is n - dist (n is length of strings)

Edit Distance or Levenshtein distance

Allows us to compare sequences of different lengths

Minimal number of edit operations (insertions, deletions and substitution) needed to transform a sequence into another sequence



source: Algorithms and Me

Edit Distance or Levenshtein distance

Allows us to compare sequences of different lengths

Minimal number of edit operations (insertions, deletions and substitution) needed to transform a sequence into another sequence

The Levenshtein distance between

- "cats" and "rats" is 1, need to substitue the "c" with the "r"
- "house" and "host" is 2 (remove "u" and substitute "e" with "t")

Applications

- Spelling correction (find closest word in the vocabulary)
- Auto suggestions of words
- RNA/DNA sequencing and others in bioinformatics

Similarity between sets

Sets: unordered collection ▷ repetition and order do not matter

- Data could be sets, such as transactions data (market baskets)
- Documents can be considered as subsets of Σ
  - vocabulary: set of all words, called language lexicon
- Cannot use similarities and distances defined for vectors
- Can represent a set by its characteristic vector (membership-vector) bit-vector of length $|U|$ (the universal set) e.g. Σ
- set complement, intersection and union via bit-wise operations
- They don't really become $|\Sigma|$-dimensional real vectors

Similarity between sets

- Any sets similarity between sets takes into account their intersection
- Intersection similarity: $sim(S_i, S_j) = |S_i \cap S_j|$
- Doesn't take into account places where they mismatch

Limitation

- $A = \{1, 3, 5, 7\}, \quad B = \{1, 3, 5, 7\}, \quad C = \{1, 3\}$
- D : set of all odd numbers
- E : set of the first 20 positive integers
- F : set of the first 10 odd integers
- $sim(A, B) = sim(A, E) = sim(A, D) = sim(A, F)$
- while clearly there should be some difference
- Their intersection sizes do not capture their similarities

- $sim(S_i, S_j) := |S_i \cap S_j| / (|S_i| + |S_j|)$ and various other
- The problem is mitigated, but still not very good notion of similarity

Jaccard Similarity

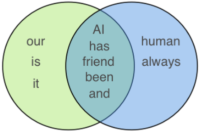
- **Jaccard similarity: intersection relative to union of the two sets**

$$JS(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

- Sentence 1: *"AI is our friend and it has been friendly"*
- Sentence 2: *"AI and humans have always been friendly"*
- Sometime need to lemmatize
- $JS(S_1, S_2) = {}^{5}/_{5+3+2} = 0.5$

Term Frequencies:

Sentence	AI	IS	FRIEND	HUMAN	ALWAYS	AND	BEEN	OUR	IT	HAS
1	1	1	2	0	0	1	1	1	1	1
2	1	0	1	1	1	1	1	0	0	1



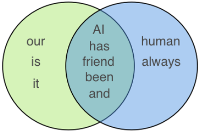
The **Jaccard distance** is defined as $1 - JS(S_i, S_j)$

Jaccard similarity: used for multisets or non-negative vectors

$$J_W(\mathbf{x},\mathbf{y}) = \frac{\sum_i \min\{x_i, y_i\}}{\sum_i \max\{x_i, y_i\}}$$

- Sentence 1: *"AI is our friend and it has been friendly"*
- Sentence 2: *"AI and humans have always been friendly"*
- Sometime need to lemmatize
- $J_W(S_1, S_2) = {}^{1+0+1+0+0+1+1+0+0+1}/_{1+1+2+1+1+1+1+1+1+1} = 0.45$

Term Frequencies:										
Sentence	**AI**	**IS**	**FRIEND**	**HUMAN**	**ALWAYS**	**AND**	**BEEN**	**OUR**	**IT**	**HAS**
1	1	1	2	0	0	1	1	1	1	1
2	1	0	1	1	1	1	1	0	0	1



Text Data: Feature Extraction

- Text include documents, articles, Facebook posts, tweets, messages
- Algorithms cannot directly work on raw text
- Convert them into numeric vectors ▷ Vector Space Modeling
- Vector are derived from textual data, in order to reflect various linguistic properties of the text
- Popular methods of feature extraction from text data are
  - Set-of-Words
  - Bag-of-Words
  - TF-IDF