Vector Norms and Proximity Measures

- Imdadullah Khan

Vector Norms: Error Measurements

- Let $x = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T \in \mathbb{R}^n$
- compare its competing estimates $y = \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^T$ and $z = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix}^T$
- Error vectors $e_y = x - y = \begin{bmatrix} x_1 - y_1 \\ \vdots \\ x_n - y_n \end{bmatrix}$ and $e_z = \begin{bmatrix} x_1 - z_1 \\ \vdots \\ x_n - z_n \end{bmatrix}$
- e.g. $e_y = \begin{bmatrix} 10 & -10\ 10\ 20 \end{bmatrix}$ and $e_z = \begin{bmatrix} 20 & -5\ 0\ 20 \end{bmatrix}$
- Need to map e_y and e_z to real numbers and compare
- Compare lengths $\|e_y\| = \sqrt{10^2 + (-10)^2 + 10^2 + 20^2} = 26.45$, $\|e_z\| = 28.72$
- Since smaller are better, y is a better estimate of x
- One can argue that with a different mapping, z is better
  $\|e_y\|_1 = |10| + |-10| + |10| + |20| = 50$, $\|e_z\|_1 = |20| + |-5| + |0| + |20| = 45$
- Note the absolute value sign $\because$ error on either side is bad
- No universally good mapping of vectors to numbers

Vector Norms

- A norm is an operation or a function on a single vector
- Denote the norm of a vector v by $\|v\|$
- Informally, they are interpreted as magnitude or length of the vector
- It maps vectors to real numbers
- A norm must satisfy the following **three axioms**
- These are reasonable things expected of anything defining length

1. $\|v\| \geq 0$, $\|v\| = 0 \leftrightarrow v = 0$ (non-negativity)
2. $\|cv\| = c\|v\|$
    - if you take 2 times a vector its length should double, also works for -2
3. $\|u + v\| \leq \|u\| + \|v\|$ (triangle inequality)
    - in a triangle length of a side should be less than the sum of two

Vector Norms

- L_1 **Norm,** $\|x\|_1$**:** one-norm, mean norm $\|x\|_1 = \sum_{i=1}^n |x_i|$
- L_2 **Norm,** $\|x\|_2$**:** Euclidean, mean-squares norm $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$
- L_p **Norm,** $\|x\|_p$**:** p-norm $\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$
- L_∞ **Norm,** $\|x\|_\infty$**:** infinity norm or max norm $\|x\|_\infty = \max_{i=1,\ldots,n}\{|x_i|\}$

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

- $\|x\|_1 = 10 + 5 + 0 + 20 = 35$
- $\|x\|_2 = \sqrt{100 + 25 + 0 + 400} = 22.91$
- $\|x\|_3 = \sqrt[3]{1000 + 125 + 0 + 8000} = 20.896$
- $\|x\|_4 = \sqrt[4]{10000 + 625 + 0 + 160000} = 20.324$
- $\|x\|_5 = \sqrt[5]{100000 + 3125 + 0 + 3200000} = 20.1272$
- $\|x\|_6 = \sqrt[6]{1000000 + 15625 + 0 + 64000000} = 20.0525$
- $\|x\|_7 = \sqrt[7]{10000000 + 78125 + 0 + 1280000000} = 20.022$
- $\|x\|_\infty = 20$

Vector Norms

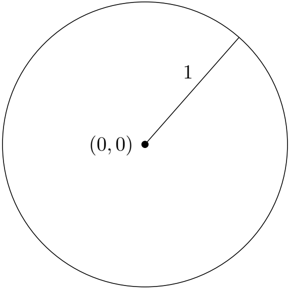
- L_1 **Norm,** $\|x\|_1$**:** one-norm, mean norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- L_2 **Norm,** $\|x\|_2$**:** Euclidean, mean-squares norm $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$
- L_p **Norm,** $\|x\|_p$**:** p-norm $\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$
- L_∞ **Norm,** $\|x\|_\infty$**:** infinity norm or max norm $\|x\|_\infty = \max_{i=1,\ldots,n}\{|x_i|\}$

- As p grows the effect of smaller terms diminish
- 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

$$\text{For } p < q \quad \|x\|_p \geq \|x_q\|$$

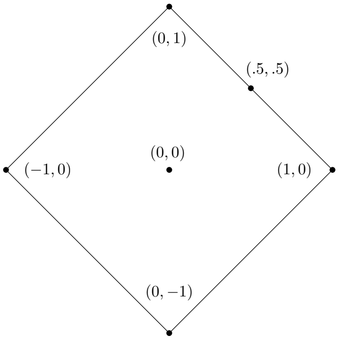
Unit Circles w.r.t Vector Norms

- **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
- $\{x \in \mathbb{R}^2 : \|x\|_2 = 1\}$
- We can draw this set, and it looks like a perfect circle



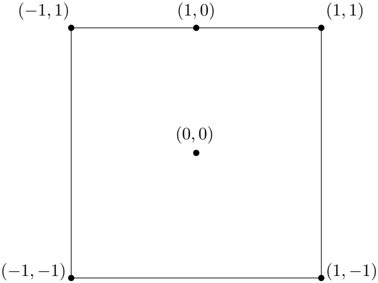
Unit Circles w.r.t Vector Norms

- The (infinite) set of 2-d vectors with L_1 norm equal to 1
- $\{x \in \mathbb{R}^2 : \|x\|_1 = 1\}$
- $\|x\|_1 = 1$ for all points $\left[z \;\; (1-z)\right]^T$
- e.g. $\left[1 \; 0\right]^T, \left[0 \; 1\right]^T, \left[.5 \; .5\right]^T, \left[-.5 \; -.5\right]^T, \left[-.8 \; .2\right]^T \left[.7 \; .3\right]^T$
- We can plot this set, it looks like a square rotated by 45 deg



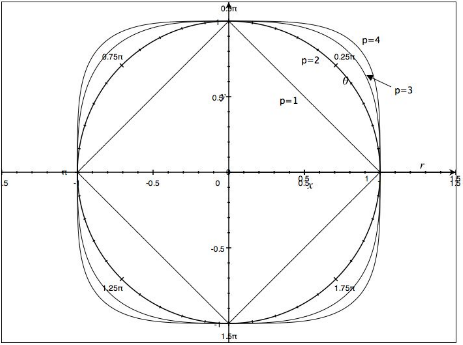
Unit Circles w.r.t Vector Norms

- The (infinite) set of 2-d vectors with L_∞ norm equal to 1
- For $p = \infty$ we get the perfect square
- $\|x\|_\infty = 1$ for all points $\begin{bmatrix}1 & z\end{bmatrix}^T$ and $\begin{bmatrix}z & 1\end{bmatrix}^T$
- e.g. $\begin{bmatrix}1 & 0\end{bmatrix}^T, \begin{bmatrix}1 & .2\end{bmatrix}^T, \begin{bmatrix}1 & .5\end{bmatrix}^T, \begin{bmatrix}-1 & -.5\end{bmatrix}^T, \begin{bmatrix}-.8 & 1\end{bmatrix}^T$
- We can plot this set, it looks like an axis aligned unit square



Unit Circles w.r.t Vector Norms

- Play around with unit circles w.r.t. other norms



Matrix Norms

- **Matrix Norms** are defined so as one can compare two matrices
- You are encouraged to search the literature for this
- I mentioned it so you are not scared if you see something like this in your papers or need it in your projects. The concept is quite simple
- Similar concept for probability distributions (histograms/densities for discrete/continuous probability distributions)
- Some commonly used norms of matrices and distributions are
- K-L divergence (and its variations)
- entropy
- entropy (infrmation theory)
- diversity index
- moments

Proximity Measures

- 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 document might not be same)
- Finding mirror pages (headers could be different)
- Establishing articles from same source (different news outlet might add a few additional things such as web address, accompanying ads might be different)
- 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

- Locality Sensitive Hashing: "similar" items go to same bucket
- Reduce dimensionality while preserving "similarities"
- Clustering: group "similar" items
- Recommendation Systems: recommend item j to user i if users "similar" to i like items "similar" to j
- "Nearest neighbors" classifier

Distance, Similarity and Proximity

- Similarity
  - Quantitative measure of how similar/alike are two objects
  - Usually falls in the range $[0, 1]$, can be 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 and Distance Metric

- We mainly discuss distance measures (similarity is its “inverse”)
- Think of distance as a function that takes two objects and returns a real number
- A distance $d(u, v)$ is a **distance metric** if it satisfies these 4 axioms

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

Vectors in Non-Euclidean Space

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

Distance Between Nominal/Categorical Feature Vectors

Let o_i and o_j be two objects with n nominal/categorical attributes

- If m out of n attributes have equal values for o_i and o_j, then
- distance $d(o_i, o_j) := d(i, j) = \dfrac{n - m}{n}$ and $sim(i, j) = \dfrac{m}{n}$

St.ID	Gender	City	School
s_1	M	PSH	SDSB
s_2	F	LHR	SS
s_3	M	KCH	LAW
s_4	M	KCH	SSE
s_5	F	LHR	SDSB
s_6	F	MTN	LAW
s_7	M	KCH	SSE
s_8	F	FSD	SSE

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

Symmetric Binary Feature Vectors

Symmetric binary values are equally valuable and carry the same weight

Let contingency table of o_i and o_j with n symmetric binary attributes be

		o_j		
		true	**false**	
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 o_i and o_j with n asymmetric binary attributes be

		o_j		
		true	**false**	
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 $d(i,j) = \dfrac{q+r}{p+q+r}$
- Having both false might not matter (their agreement on false carries no weight)

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'
s_1	CS	1	0	1	1	1	0	0	0	1	0	0
s_2	CS	1	1	1	1	1	0	1	0	0	0	0
s_3	CS	0	1	1	1	1	1	1	0	0	0	0
s_4	EE	1	0	1	0	0	1	1	1	0	0	0
s_5	EE	1	1	0	0	0	1	1	1	0	0	0
s_6	EE	1	1	0	1	0	1	1	1	1	0	0
s_7	EC	1	1	1	0	0	0	0	0	1	1	1
s_8	EC	1	1	0	0	0	0	0	0	1	1	1
s_9	EC	1	0	0	0	0	0	1	0	1	1	1

Symmetric/Asymmetric Binary Feature Vectors Example

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
s_1	11	8	6	5	3	5	6	5	5
s_2	8	11	9	6	6	6	5	4	4
s_3	6	9	11	6	6	6	3	2	2
s_4	5	6	6	11	9	7	4	3	5
s_5	3	6	6	9	11	9	4	5	5
s_6	5	6	6	7	9	11	4	5	5
s_7	6	5	3	4	4	4	11	10	8
s_8	5	4	2	3	5	5	10	11	9
s_9	5	4	2	5	5	5	8	9	11

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
s_1	5	4	3	2	1	3	3	2	2
s_2	4	6	5	3	3	4	3	2	2
s_3	3	5	6	3	3	4	2	1	1
s_4	2	3	3	5	4	4	2	1	2
s_5	1	3	3	4	5	5	2	2	2
s_6	3	4	4	4	5	7	3	3	3
s_7	3	3	2	2	2	3	6	5	4
s_8	2	2	1	1	2	3	5	5	4
s_9	2	2	1	2	2	3	4	4	5

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

- Consider s_1 and s_7 with different majors i.e CS and EC
- $sim(s_1, s_7) = 6$ in symmetric binary case because courses not taken by both are also adding value to similarity (it shouldn't be the case)
- $sim(s_1, s_7) = 3$ in asymmetric binary case (false to false match is not adding any value)

ENCODING

- 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

Encoding is used to convert nominal/categorical variable to numeric

label-Encoding

- One solution is to give ‘*numerical value*’ 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 categorical value, better the category)
- Say your algorithm internally calculates average
- This implies that: Average of VW and Honda is Acura
- The model’s prediction would have a lot of errors

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

Numeric Feature Vectors

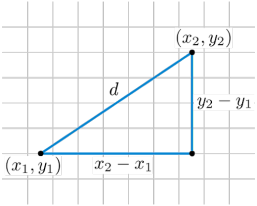
- Many distance/similarity measures for objects described by *n* numeric attributes
- We use one or the other measure depending on the applications and after a certain amount of trial and error

Numeric Feature Vectors: Euclidean distance

$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ are numeric vectors

- The most natural and commonly used distance measure is the
- Straight line distance, Euclidean Distance, or ℓ_2 distance
- Since x_i and x_j are vectors (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

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{jn} - y_{jn})^2} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$



Numeric Feature Vectors: Manhattan distance

- **Manhattan distance** is sum of coordinate wise absolute differences

$$d(x_i, x_j) = \sum_{k=1}^{n} |x_{ik} - x_{jk}|$$

- Correspond to the number of blocks one has to travel while moving from x_i to 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
- The ℓ_p norm of the difference vector

$$d(x_i, x_j) = \sqrt[p]{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^p}$$

Numeric Feature Vectors: Supremum Distance

- Also called Chebychev distance
- This is a special case of the Minkowski distance when p approaches ∞

$$d(x_i, x_j) = \lim_{p \to \infty} \sqrt[p]{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^p}$$

- One can easily see that

$$d(x_i, x_j) = \lim_{p \to \infty} \sqrt[p]{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^p} = \max_k \{|x_{ik} - x_{jk}|\}$$

Numeric Feature Vectors: Cosine Similarity

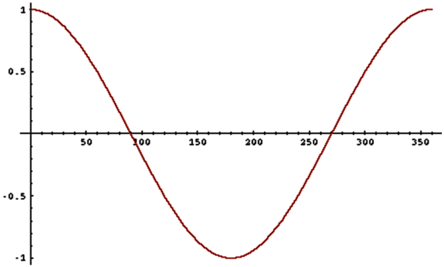
$$CS(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

- Calculates similarity by measuring cosine of the angle between the two vectors
- Based on orientation/directions and not magnitudes of the vectors
- $u \cdot v$ is dot product and $\|u\|$, $\|v\|$ are magnitudes of u and v

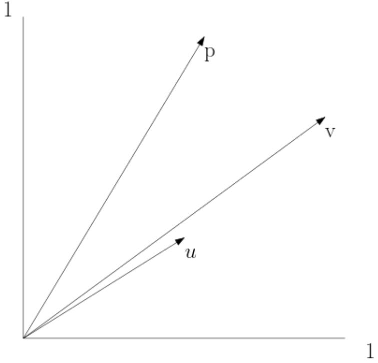
- Let $\vec{u}$ = [1 3 2] and $\vec{v}$ = [5 0 -3]
- $u \cdot v = -1$, $\|u\| = \sqrt{14}$, $\|v\| = \sqrt{34}$
- $CS(u, v) \ = \ -1/(\sqrt{14} * \sqrt{34}) = 0.99$

From Angle to Cosine

- Cosine is monotonically decreasing for the interval [0°, 180°]
- $\cos(0^\circ) = 1$ (Overlapping/parallel vectors)
- $\cos(90^\circ) = 0$ (perpendicular/orthogonal vectors)
- $\cos(> 90^\circ) < 1$
- So $CS(u, v) \to 1$ means more similar

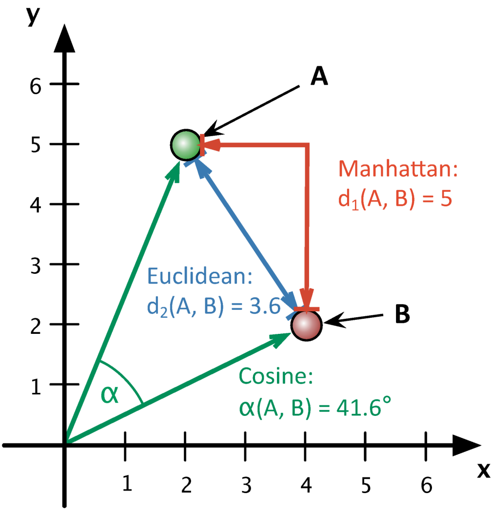


Numeric Feature Vectors: Cosine Similarity



- ℓ_2-distance between *v* and *u* is larger than that between *v* and *p*
- But in terms of direction *v* is more similar to *u* than it is to *p*

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 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 (that are needed for example for grouping students by majors or selecting the top students)

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_{ik} is mapped to corresponding rank of the value of x_{ik}.
- 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
- Group attributes by their types and compute group wise distances, sum these distances and normalize it
- A_1, A_2, A_3 are sets of nominal, ordinal and numeric attributes
- Let o_i^{nom} : o_i as described by nominal attributes (columns of A_1)
- Assuming there are no missing values, then

$$d(i,j) = \frac{|A_1| \cdot d(o_i^{nom}, o_j^{nom}) + \sum_{k \in A_2 \cup A_3} |o_{ik} - o_{jk}|}{n}$$

- 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

Non-Vector Data

- Some data is not directly described by values of attributes
- i.e. not given as feature vectors
- We first convert data into some kind of feature vectors
- Then apply methods and models developed for vectors
- Many types and flavors of data such as text, sequences, time-series data, streams are not given as feature vectors
- Important step is **Feature Selection, Feature description, Feature Mapping**

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

- Measures number of positions at which the corresponding characters are different in two string
- The Hamming distance between
  - “karolin” and “kathrin” is **3**
  - 1011101 and 1001001 is **2**
- Common applications of Hamming distance are in
  - error correction codes, communication, information theory
  - computational biology, bioinformatics
- Limitations
  - Only works for sequences of equal length
  - Count all mismatches as equal
  - Cannot perform any edit operation on sequences
- Similarity generally 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 string into other string
- The Levenshtein distance between
  - "cats" and "rats" is 1, since you need to substitue the "c" with the "r"
  - "house" and "host" is 2 (remove "u" and substitute "e" with "t")
- Applications of Levenshtein distance are:
  - Spelling correction (find the closest word from the vocabulary)
  - Auto suggestions of words
  - RNA/DNA sequencing and others in bioinformatics

Similarity between sets

- 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
- Sets: unordered collections (repetition and order do not matter)
- 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

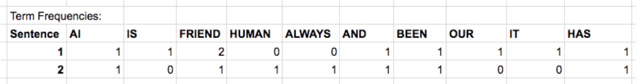
- Any sets similarity notion takes into account intersections
- $sim(S_i, S_j) = |S_i \cap S_j|$
- Doesn't take into account places where they mismatch
- Let $A = \{1, 3, 5, 7\}, B = \{1, 3, 5, 7\}, C = \{1, 3\}$
- Let D be the set of all odd numbers
- Let E be the set of the first 20 positive integers
- Let F be the set of the first 10 odd integers
- Their intersection sizes do not really capture their similarities
- $sim(A, B) = sim(A, E) = sim(A, D) = sim(A, F)$, while clearly there should be some difference
- $sim(S_i, S_j) := |S_i \cap S_j|/(|S_i| + |S_j|)$ and various other similarities
- Though the problem is mitigated, but could construct examples where this is not very good notion of similarity

Jaccard Similarity

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

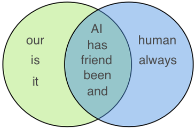
$$JS(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

- The **Jaccard distance** is defined as $1 - JS(S_i, S_j)$
- Sentence 1: AI is our friend and it has been friendly
- Sentence 2: AI and humans have always been friendly
- Sometime need to lemmatize
- Jaccard similarity of $5/(5+3+2) = 0.5$



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



Text Data: Feature Extraction

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