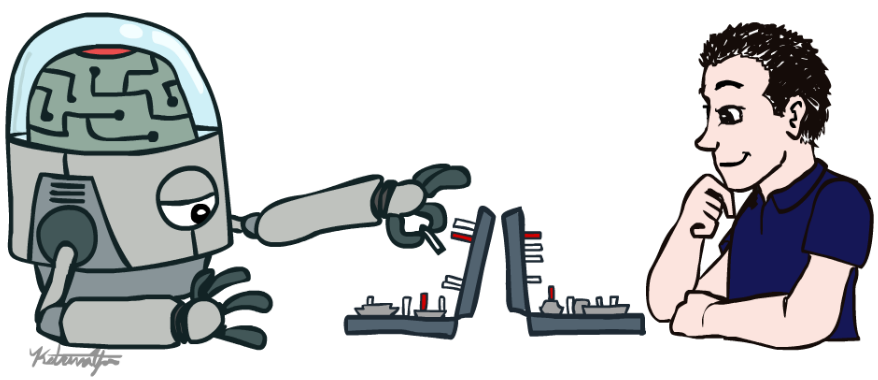
CSE 3521:
Introduction to Artificial Intelligence



[Many slides are adapted from the UC Berkeley. CS188 Intro to AI at UC Berkeley and previous CSE 3521 course at OSU.]

The Ohio State University

Supervised learning

- Data type: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$

- Goal: Build a model so that given a future data instance $\boldsymbol{x}$, it can tell the label y
  - Example: Nearest neighbors

- The "label" in $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ provides supervision of how to give each data instance a label
- The label can be **"numerical" (regression)** or **"categorical" (classification)**

The general representation of data instances

- Definition:
  - A categorical value takes one category from a set of categories.
    - There is no intrinsic ordering of the categories
  - A numerical value is a real number.
    - There is a clear ordering and space between values.

- Examples:
  - Coin instance: weight = 2.5 (g), size = 16.5 (mm)
  - Car instance: brand = Mazda, year = 2015, color = blue

The general representation of data instances

- Mathematical representations:
  - **Numerical values:** vectors

weight	2.5
size	16.5

or $\begin{bmatrix} 2.5 \\ 16.5 \end{bmatrix}$

  - **Categorical values:** a <u>one-hot vector</u> for each feature variable (each has exactly one **1**)

Mazda	1
Toyota	0
Honda	0

blue	1
red	0
gray	0

concatenation

Mazda	1
Toyota	0
Honda	0
blue	1
red	0
gray	0
year	2015

Properties
- A vector element: index & value
- Every two one-hot vectors of different categories are of <u>the same distance</u>

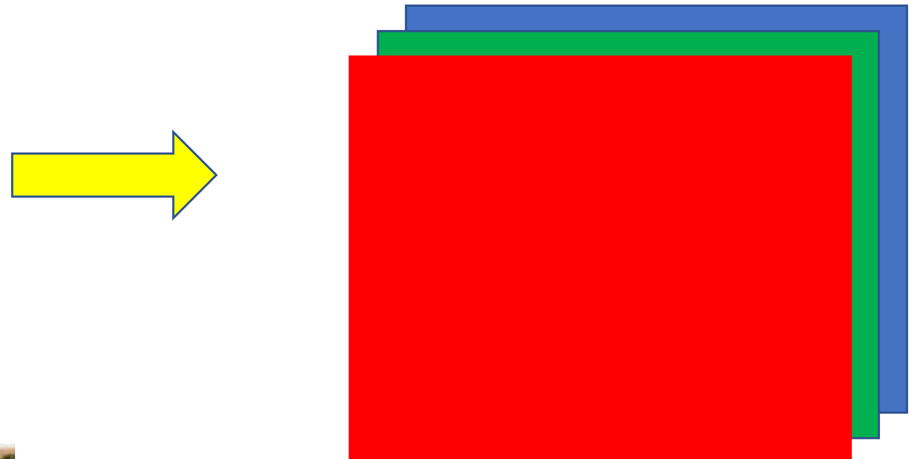
Popular data instances in applications

- **Computer vision:** image & video
- **Natural language processing:** sentence & document
- **Speech:** utterance
- **Robotics:** LiDAR point cloud
- **Health care:** electronic health record (EHR)

Computer vision

Image (s)

Video (s) = sequence of images



RGB image (s): Three matrices

Gray images (s): One matrix

Natural language processing

Character → Word → **Sentence** → **Document** → Doc. collections → ...

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals. Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving"

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Natural language processing

Character → Word → Sentence → Document → Doc. collections → …

h	a	p	p	y
104	97	112	112	121

ASCII Code (symbol):
{0, 1, ……, 255}

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Are these numbers "numerical values" or "categorical values" (indices)?

Is "a" (97) semantically closer to "b" (98) than "p" (112)?

If we change the order in the codebook, does it matter?

Natural language processing

Character → Word → Sentence → Document → Doc. collections → …

h	a	p	p	y

104	97	112	112	121

1	0	0	0	0	0
	…	…	…	…	…
97	0	**1**	0	0	0
	…	…	…	…	…
112	0	0	**1**	**1**	0
	…	…	…	…	…
256	0	0	0	0	0

256

One-hot representation:

- 256-dimensional {0, 1} vector
- Each column has exactly one **1**

Properties

- A vector element: index & value
- Every two one-hot vectors (columns) are of <u>the same distance</u> if they are not the same

Natural language processing

Character → **Word** → Sentence → Document → Doc. collections → …

- A word can be represented as a sequence of character indices (a sequence of one-hot vectors)
- A word can be also represented just by a one-hot vector
- More: “happy”: 0001235, “pleased”: 0128736, “sad”: 0059875, …… (from a dictionary)
- What is the vector dimension?

Natural language processing

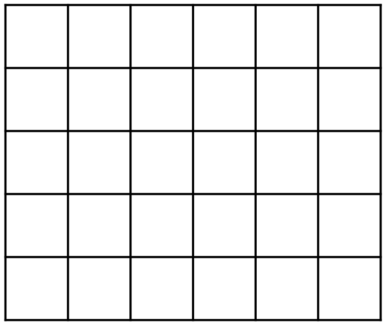
Character → Word → **Sentence → Document** → Doc. collections → …

- A sequence of words OR a unique index of each sentence?
- If there are 10K unique words, and each sentence is of length 10, how many indices?

tokenization

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. …

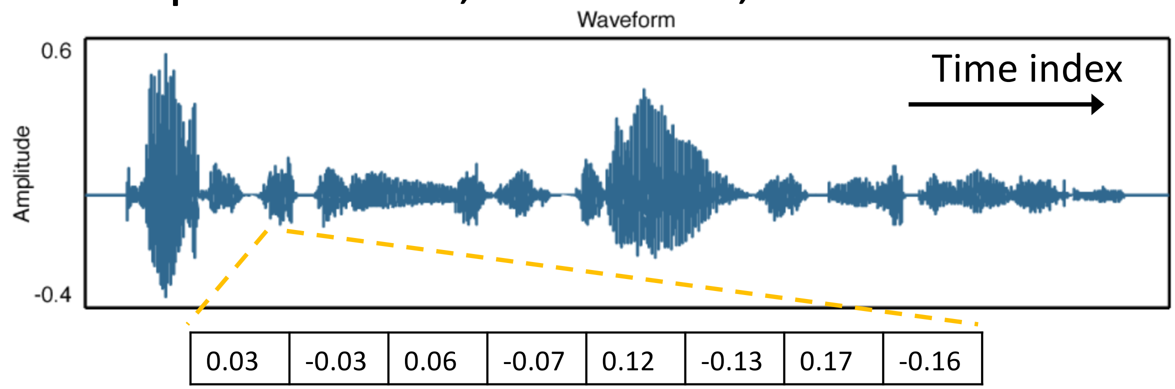
"<Start>" "machine" "learning" "(" "<UNK>" ")" "is" "the" "study" "of" "computer" "algorithms" "that" "improve" "automatically" "through experience" "." "<End>" "<Start>" "it" "is" "seen" "as" "a" "subset" "of" "artificial" "intelligence" "." "<End>" …



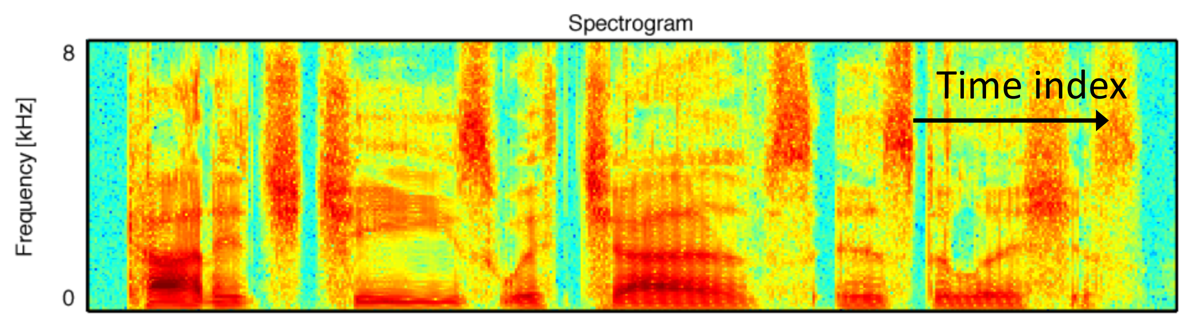
- Sequences of indices/one-hot vectors of words
- <UNK>: out-of-vocabulary (OOV) words

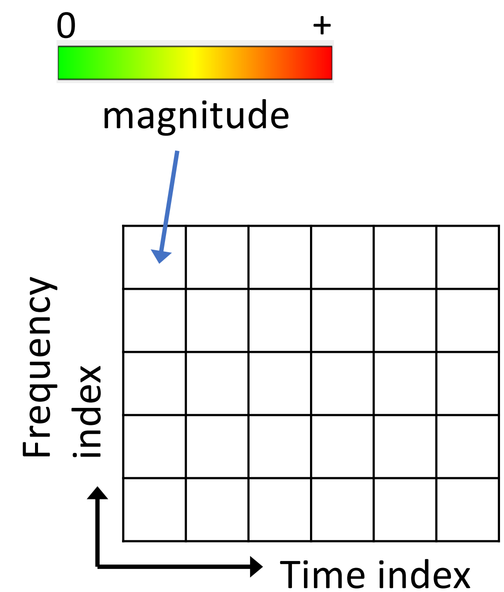
Speech

- Utterance: a spoken word, statement, or vocal sound



- Time-frequency representation:

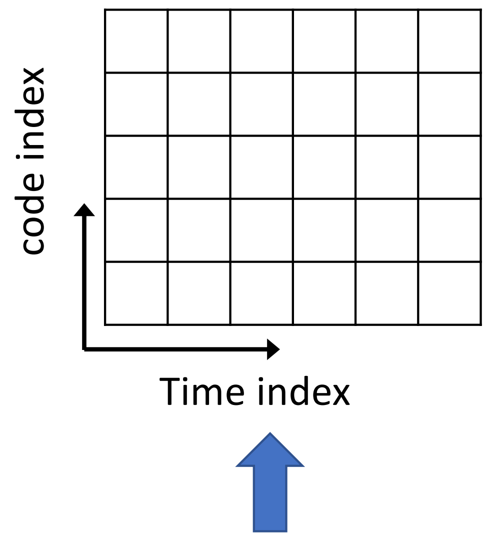




Health care

- Electronic health/medical record (HER/EMR)
  - ICD-9/ ICD-10 codes

Diagnosis	ICD-9	ICD-10
Cervical Sprain, initial encounter	847.0	**S13.4xxA**
Thoracic Sprain, initial encounter	847.1	**S23.3xxA**
Lumbar Sprain, initial encounter	847.2	**S33.5xxA**
Cervical Degenerative Disc Disease	722.4	**M50**
Thoracic Degenerative Disc Disease	722.51	**M51**
Lumbar Degenerative Disc Disease	722.52	**M51.2**



Properties

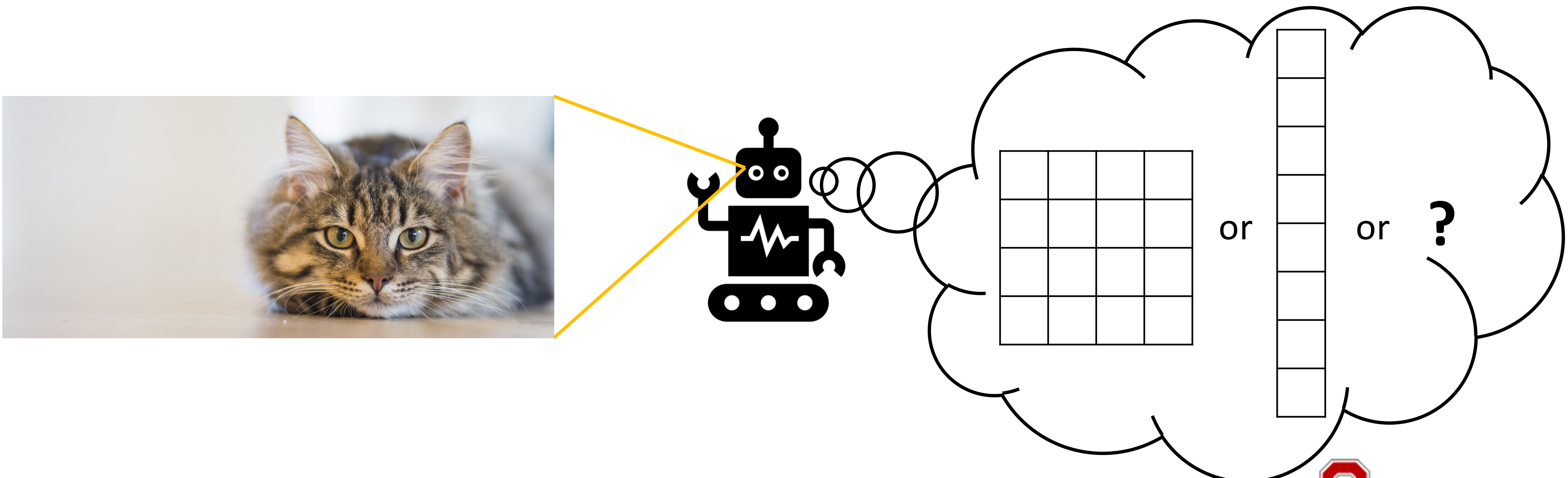
- Multi-hot
- One-hot
- All-zero

  - Every time you see a doctor, some diagnoses are made (with codes)

In summary

- Data instances from many different applications can be represented by vectors or matrices!
- Linear algebra is a very efficient and effective way to perform computation (e.g., discovering patterns) on them!
- There are many other ways to represent data instances!

Data representations (aka features)



THE OHIO STATE UNIVERSITY

Machine learning (ML): detect patterns in data

From the perspective of first-order logic:

- ○ A coin with weight = 5.65 g and diameter = 23.6 mm ⟹ 10 cent
- ○ A coin with weight * 0.2 + diameter * 0.04 < 2 ⟹ 10 cent

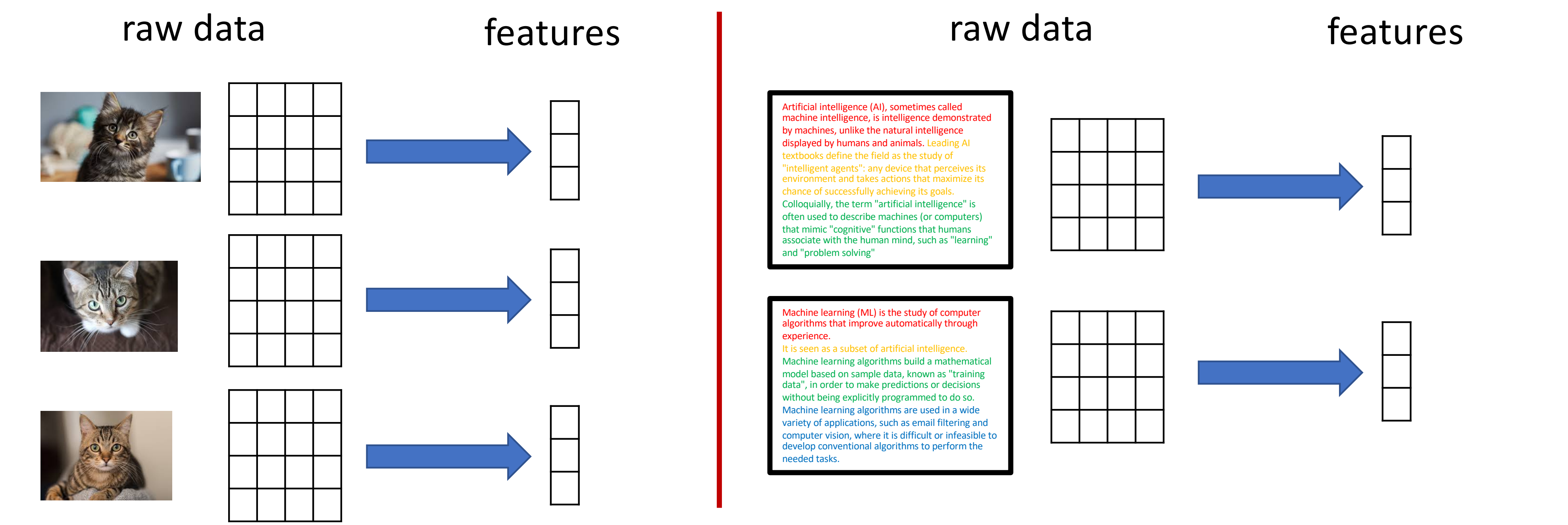
Patterns:

- to be detected by ML
- Built upon features

Feature variables:

- Values (facts) are to be extracted from the data
- What feature variables should we use/define?

Data vs. features



- Can we use raw data representation as features?
- If so, why bother further extracting features from the raw data?
  - *simplify the data, remove unrelated information, domain knowledge, ……*

Feature extraction

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be:

- Informative
- non-redundant
- facilitating the subsequent learning and generalization steps
- leading to better human interpretations

Feature extraction

- **Bag-of-words (BOW) representation**
  - BOW for natural language processing and computer vision
  - Feature normalization: L1 and L2 normalized

- **Dataset representation**
  - Histogram and Parzen window
  - Feature correlation
  - Feature normalization: z-score, whitening

- **Dimensionality reduction**
  - Principal component analysis (PCA)

Bag of Words (BOW)

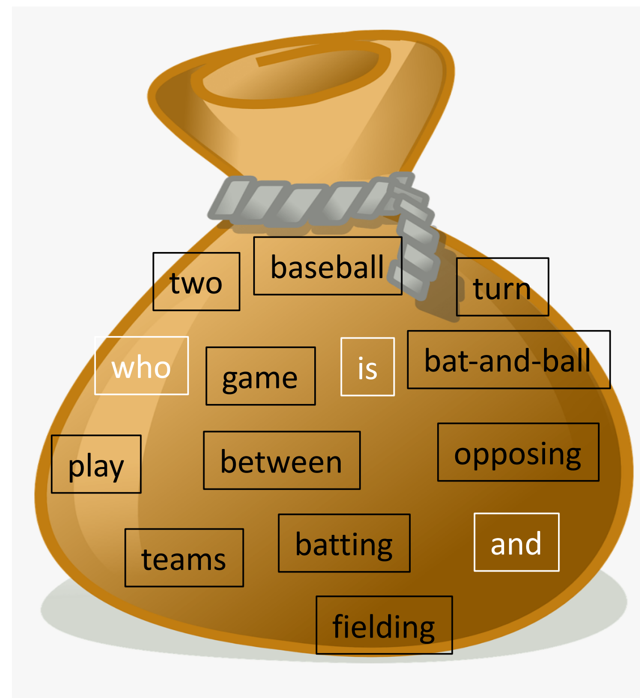
Bag of Words (BOW)

A simplified representation of sentences (documents) for classification or retrieval by counting the occurrences of each unique word (or phrase)

- ignore the grammar
- ignore the word order
- keep the word counts (frequencies)
- lead to a “fixed”-size vector representation

“Baseball” “is” “a” “bat-and-ball”
“game” “played” “between”
“two” “opposing” “teams”
“who” “take” “turns” “batting”
“and” “fielding” “.”

A sequence of tokens (and their indices)



Bag of Words (BOW)

Given:

- A <u>dictionary, vocabulary, or codebook</u>: f(token) → index ∈ $\{1, \cdots, D\}$ or N/A
- An all-0 vector: $\boldsymbol{x}$ = $\begin{bmatrix} x[1] \\ \vdots \\ x[D] \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

Input: A sequence of word tokens w[1], w[2], ….. w[M]

for m = 1 : M

  if f(w[m]) ~= N/A

    x[f(w[m])] += 1

  end

end

Return: $\boldsymbol{x}$

- **Out-of-vocabulary**
- **"Stop" words:** too frequent in all sentences/documents and less informative in differentiating them (e.g., "is", "are", "and", ……)

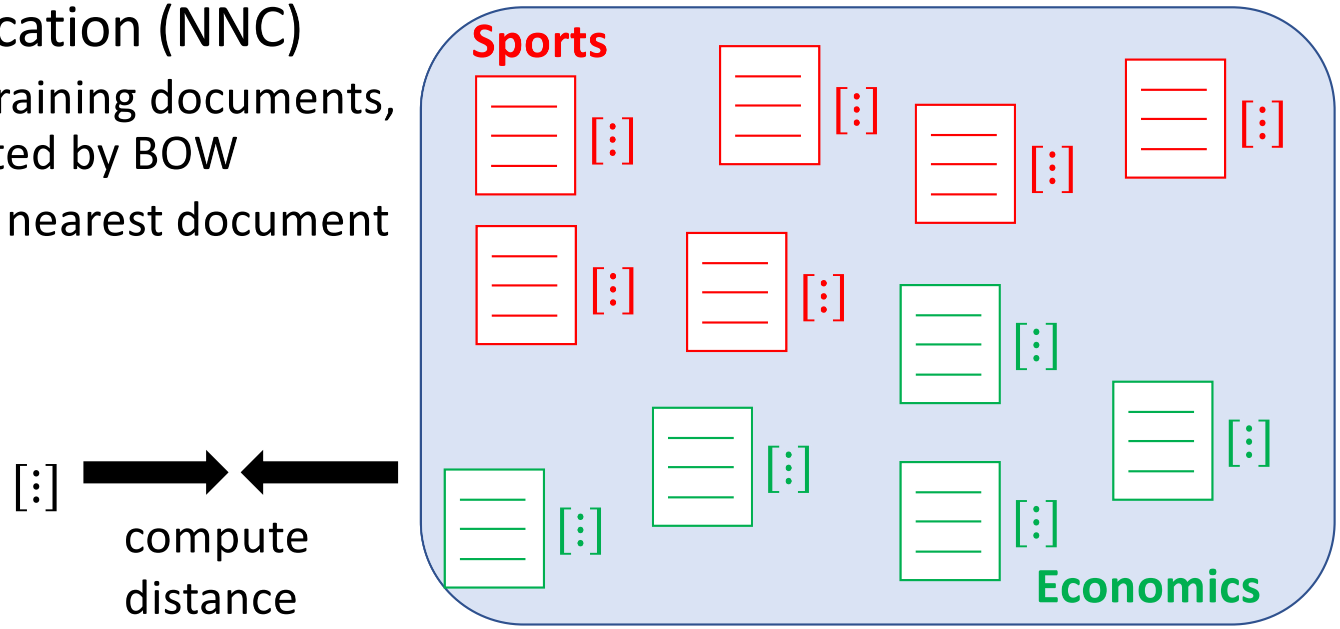
Word token counts

Example: BOW for classification

- Consider a document classification:
  - {sports, economics}
- Nearest neighbor classification (NNC)
  - Compute distance to all training documents, each of them is represented by BOW
  - Output the “label” of the nearest document

Baseball is a bat-and-ball game played between two opposing teams who take turns batting and fielding.

Training documents



Example: BOW for classification

Distance

- Euclidean (L_2) distance:

$$\|\boldsymbol{x} - \boldsymbol{x_n}\|_2 = \left(\sum_{d=1}^{D} (x[d] - x_n[d])^2\right)^{1/2}$$

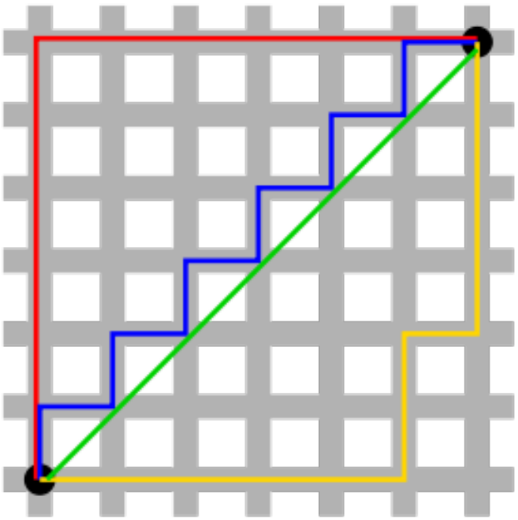
- L_1 distance:

$$\|\boldsymbol{x} - \boldsymbol{x_n}\|_1 = \sum_{d=1}^{D} |x[d] - x_n[d]|$$

- L_p norm:

$$\|\boldsymbol{x} - \boldsymbol{x_n}\|_p = \left(\sum_{d=1}^{D} |x[d] - x_n[d]|^p\right)^{1/p}$$

$\boldsymbol{x}$: 2-dimensional vectors



Green line is Euclidean distance. Red, Blue, and Yellow lines are L_1 distance

Example: BOW for classification

Given: a distance metric **dis**, a training set = $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $t = \infty$ (min distance), y (predicted label)

Label:
$y \in$ {sports, economics}
$y \in$ {-1, 1} or {0, 1}
$y \in \{1, \dots, C\}$

Input: a test data instance $\boldsymbol{x}$

for n = 1 : N

    if dis($\boldsymbol{x}$, $\boldsymbol{x}_n$)< t

        y= y_n

        t = dis($\boldsymbol{x}$, $\boldsymbol{x}_n$)

    end

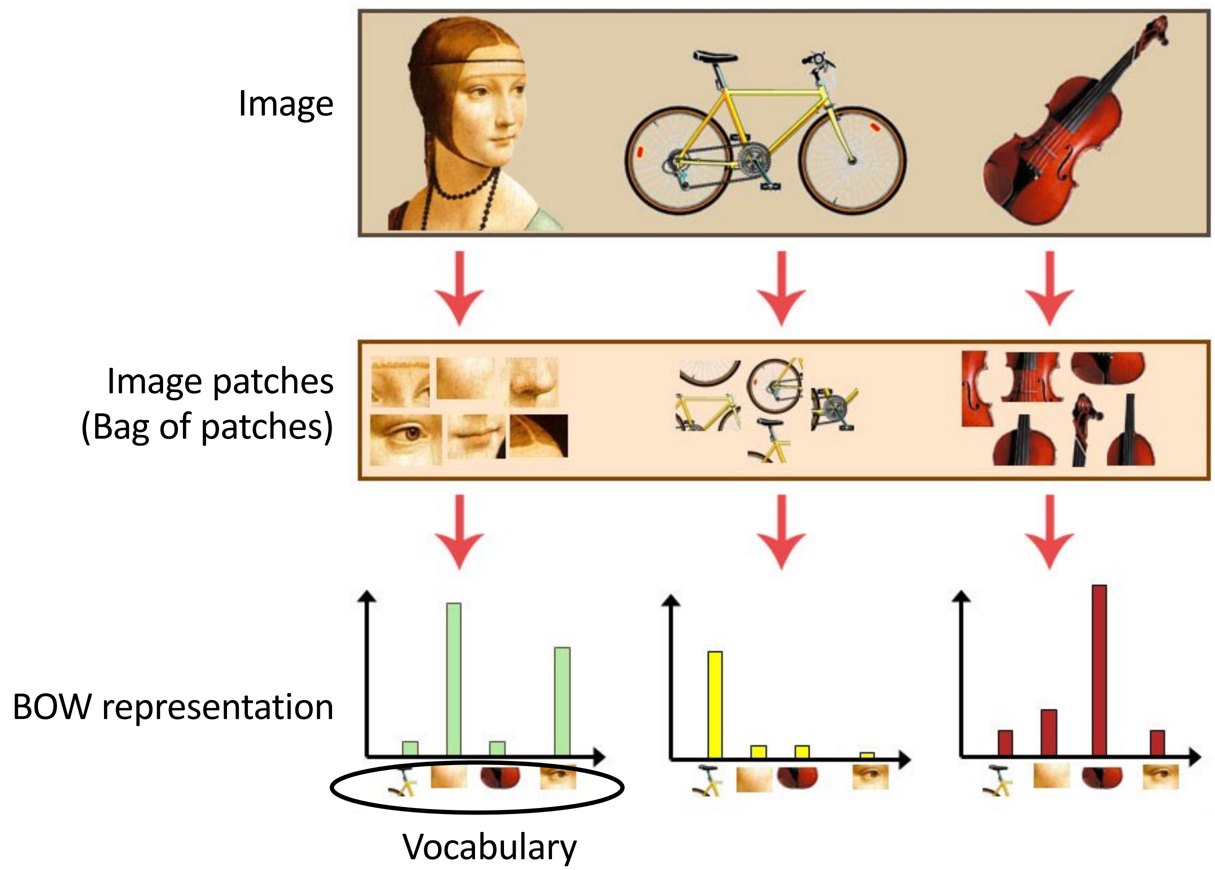
end

Return: y

Bag of Words (BOW) for images

BOW can be applied to vision

- Vocabulary: *D* image patches
- Each image: a bag of patches
- BOW representation: *D*-dim
  - For each image patch I[m], find the "nearest" patch in the vocabulary and use its index ∈ {1, …, D} to represent I[m]
  - Count the time that each index shows up in the image

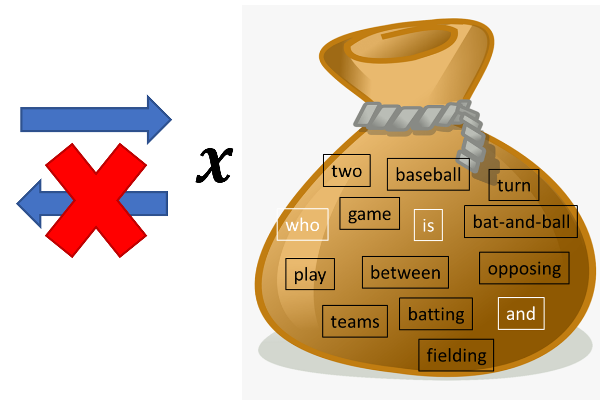


Bag of Words (BOW)

What do we lose?

- What the original sequence was

Baseball is a bat-and-ball game played between two opposing teams who take turns batting and fielding.



Pros:

- Simplified, easily-understandable, fixed-size

Bag of Words (BOW)

Cons:

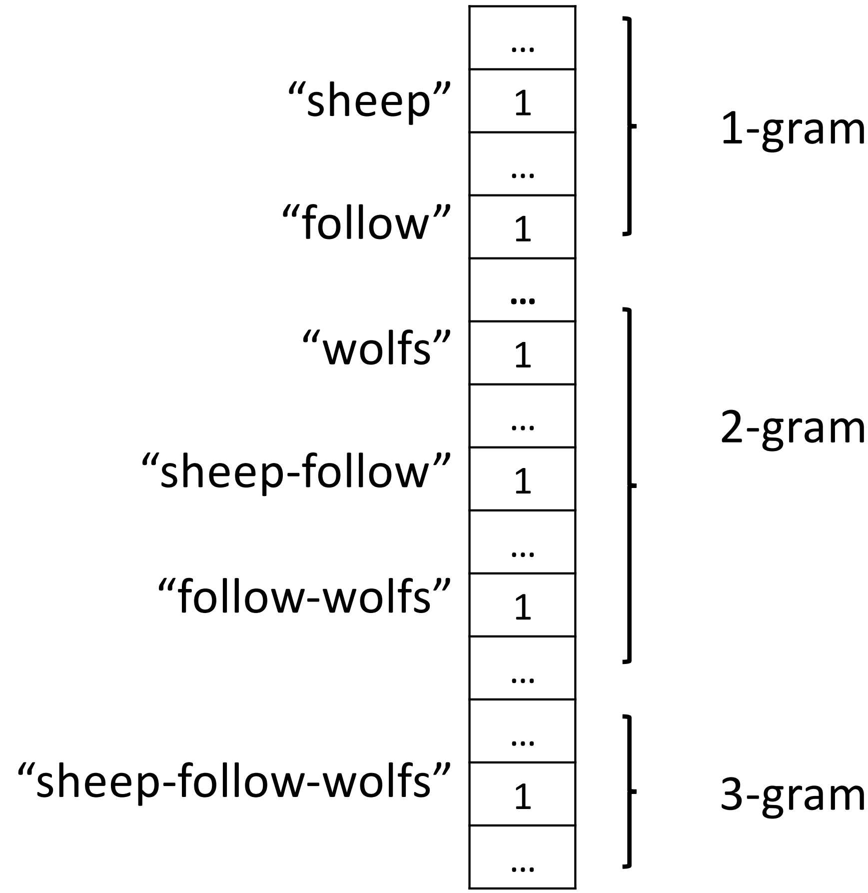
- missing the sequential information (e.g., “sheep follow wolfs” vs. “wolfs follow sheep”)
- not normalized (e.g., comparing long vs. short documents)
- words treated as independent (e.g., no synonyms and antonyms)
- highly sparse, high dimensional

N-gram vocabulary

Cons 1: Missing the sequential information

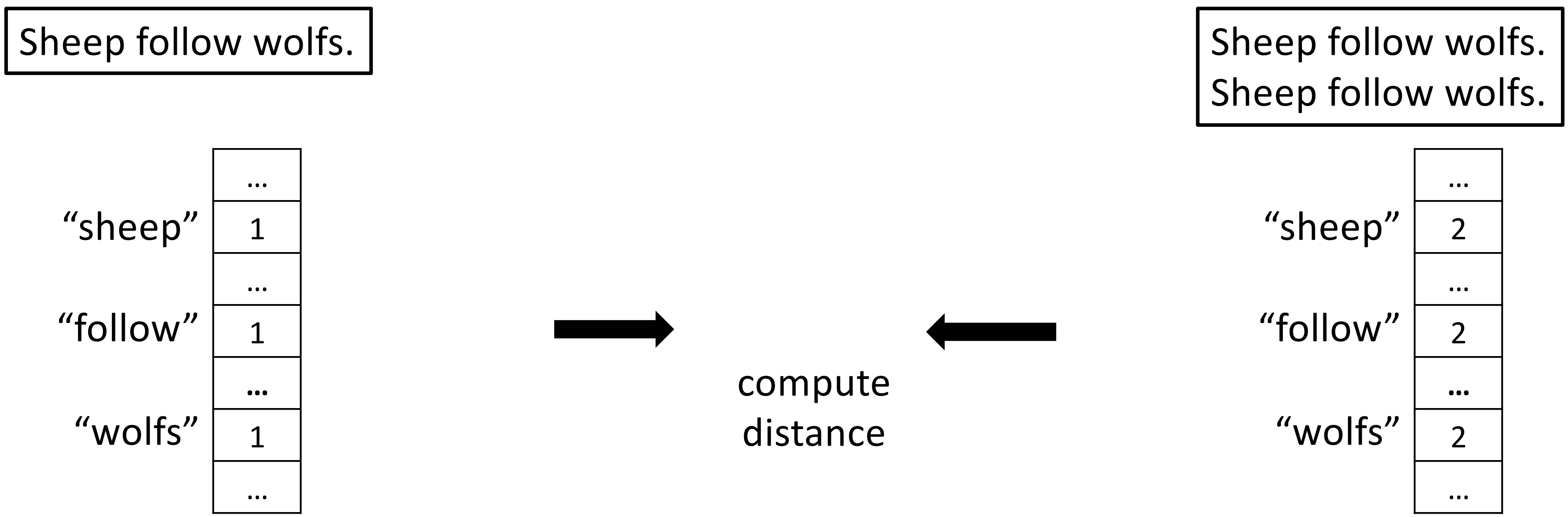
- One solution: **N-gram vocabulary**
- 1-gram (unigram): “sheep”, “follow”, “wolfs”
- 2-gram (bigram): “sheep-follow”, “follow-wolfs”
- 3-gram: “sheep-follow-wolfs”
- …
- Size of the vocabulary: $D + D^2 + D^3$

- “Sheep follow wolfs” becomes:



Dataset-independent normalization

Cons 2: not normalized



Dataset-independent normalization

Cons 2: not normalized

- One solution: **feature normalization**
- Vector norm (i.e., length or magnitude):

$$\|\boldsymbol{x}\|_p = \left(\sum_{d=1}^{D} |x[d]|^p\right)^{1/p}$$

- **L_p normalization:**

$$\boldsymbol{z} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_p} = \begin{bmatrix} \frac{x[1]}{\|\boldsymbol{x}\|_p} \\ \vdots \\ \frac{x[D]}{\|\boldsymbol{x}\|_p} \end{bmatrix}$$

Properties

- After **L_p normalization**, $\|\boldsymbol{z}\|_p = 1$
- Proof:

$$\left\|\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_p}\right\|_p = \left(\sum_{d=1}^{D} \left|\frac{x[d]}{\|\boldsymbol{x}\|_p}\right|^p\right)^{1/p}$$

$$= \left(\left(\frac{1}{\|\boldsymbol{x}\|_p}\right)^p \sum_{d=1}^{D} |x[d]|^p\right)^{1/p}$$

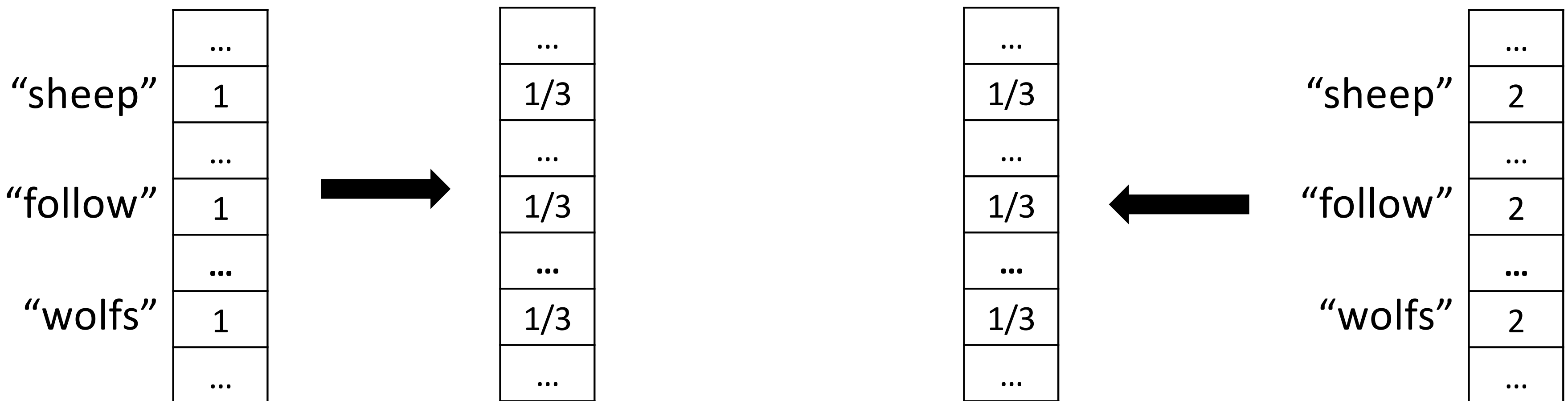
$$= \frac{1}{\|\boldsymbol{x}\|_p}\left(\sum_{d=1}^{D} |x[d]|^p\right)^{1/p}$$

$$= 1$$

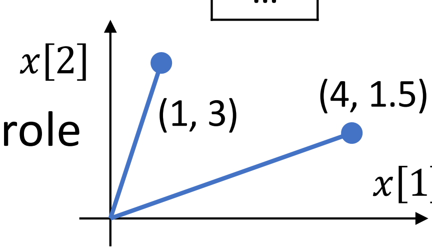
Dataset-independent normalization

Cons 2: not normalized

- One solution: **feature normalization**
- **L_1 normalization:** popular for counts (frequency, probability)



- **L_2 normalization:** widely used if the vector angle plays an important role



Bag of Words (BOW)

Cons:

- missing the sequential information (e.g., "sheep follow wolfs" vs. "wolfs follow sheep")
- not normalized (e.g., comparing long vs. short documents)
- words treated as independent (e.g., no synonyms and antonyms)
- highly sparse, high dimensional

Summary

- Data and data representation (features)
  - Numerical vs. categorical variables
  - Feature extraction: from raw data to simplified, informative, non-redundant, or more interpretable representations

- Bag-of-words (BoW) representation
  - Fixed-sized representations for sentences and documents (and images)
  - Nearest neighbor classification based on distance metrics