COGS 118A, Winter 2020

Supervised Machine Learning Algorithms

Lecture 1: Basics

Zhuowen Tu

Vector

(probably the most important concept in this class)

Vector: a sequence of elements

(white, red, blue, green)

Important to note: the order DOES matter for vectors

(white, red, blue, green) $\neq$ (red, white, green, blue)

Sometimes, we also use: <white, red, blue, green>

In Python: [white, red, blue, green]

It is of critical importance to under the vector representation in machine learning!

Some notations that we will be using

Input data:

We use x (lower case) to denote a feature value (scalar).

The ith input data sample is represented as a vector using bold $\mathbf{x}$:

$\mathbf{x}_i = (x_{i1}, ..., x_{im}) \in \mathbb{R}^m$: A row vector of m elements.

$\mathbf{x}_i = (22, 1, 0, 160, 180)$

The entire dataset is represented by a set (the sequence in which each data input $\mathbf{x}_i$ usually doesn't matter.

$S = \{\mathbf{x}_i, i = 1..n\}$: A set S with n samples. i goes from 1 to n.

Some notations that we will be using

Output prediction:

We use y (lower case) to denote a binary classification.

$y = -1$ (or sometimes we use $y = 0$) is referred to as the negative class.

$y = +1$ is referred to as the positive class.

Given a data sample $\mathbf{x}_i = (x_{i1}, ..., x_{im})$,

we want to predict $y_i \stackrel{?}{=} -1 \; or \; +1$

Some notations that we will be using

Model parameter:

Model: $\mathbf{w} = (w_1, ..., w_m) \in \mathbb{R}^m$ (in the same dimension of input $\mathbf{x}$)

bias: $b \in \mathbb{R}$ (scalar)

Data sample $\mathbf{x} = (x_1, ..., x_m) \in \mathbb{R}^m$,

$\mathbf{w} \cdot \mathbf{x} + b$ $\qquad$ $y = \begin{cases} +1 & if \ \ \mathbf{w} \cdot \mathbf{x} + b \geq 0 \\ -1 & otherwise \end{cases}$ $\qquad$ $(w_1, w_2, .., w_m) \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ x_m \end{pmatrix} + b$

"$\cdot$" refers to as the dot product between two vectors

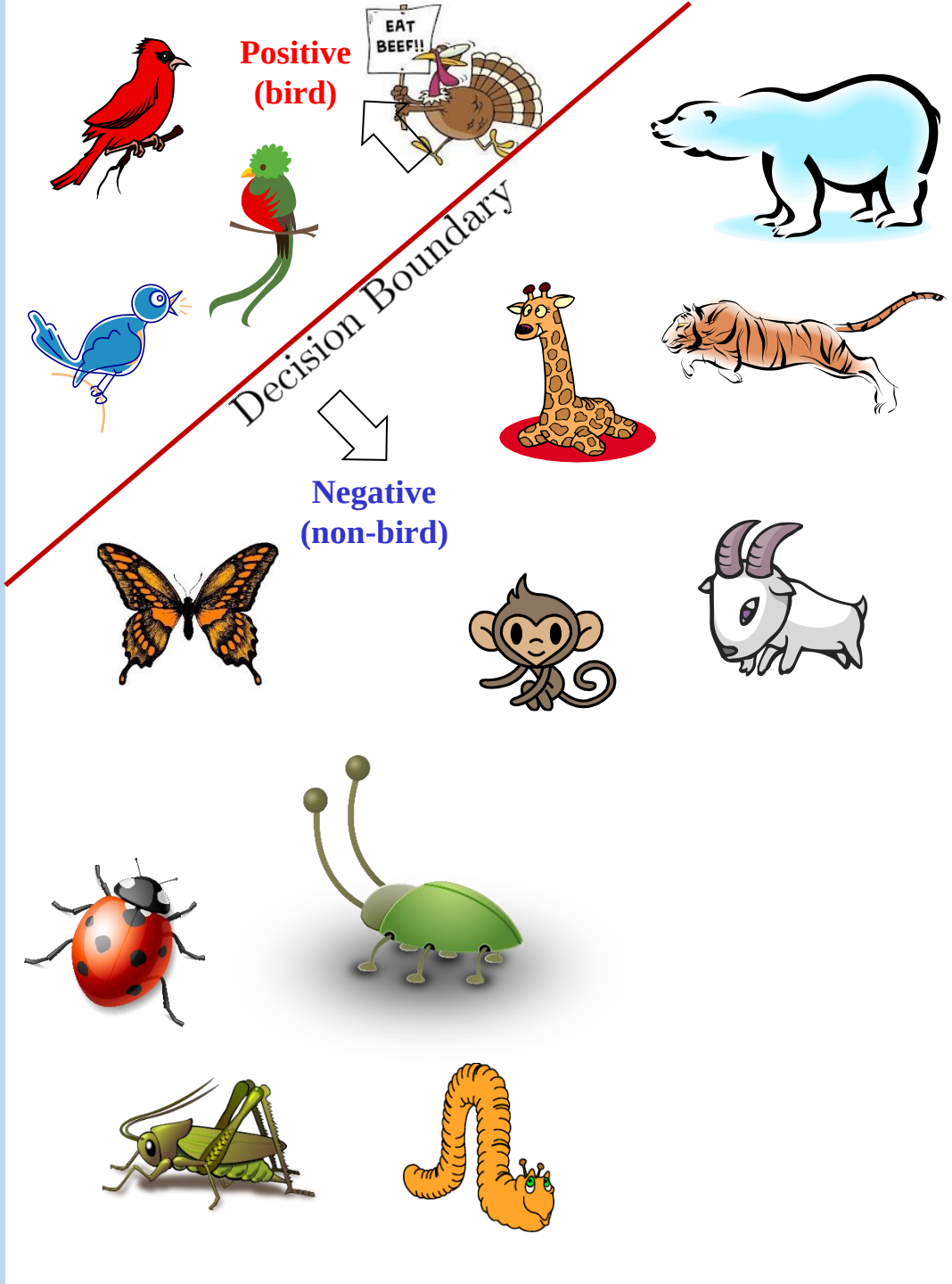
Alternative notation 1: $< \mathbf{w}, \mathbf{x} > + b$

Alternative notation 2: $\mathbf{w}\mathbf{x}^T + b$ ($\mathbf{w}$ and $\mathbf{x}$ are row vectors).

$\mathbf{w}^T\mathbf{x} + b$ ($\mathbf{w}$ and $\mathbf{x}$ are column vectors).

Supervised learning

(Classification)



Summary of the classification problem

$\mathbf{x} = (x_1, x_2, ...)$

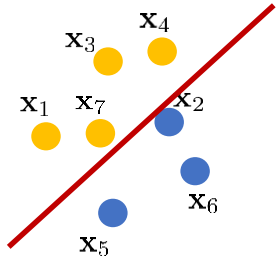
$y = 1(bird)$

x_1: color
x_2 weight
...

$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4), (\mathbf{x}_5, y_5)\}$

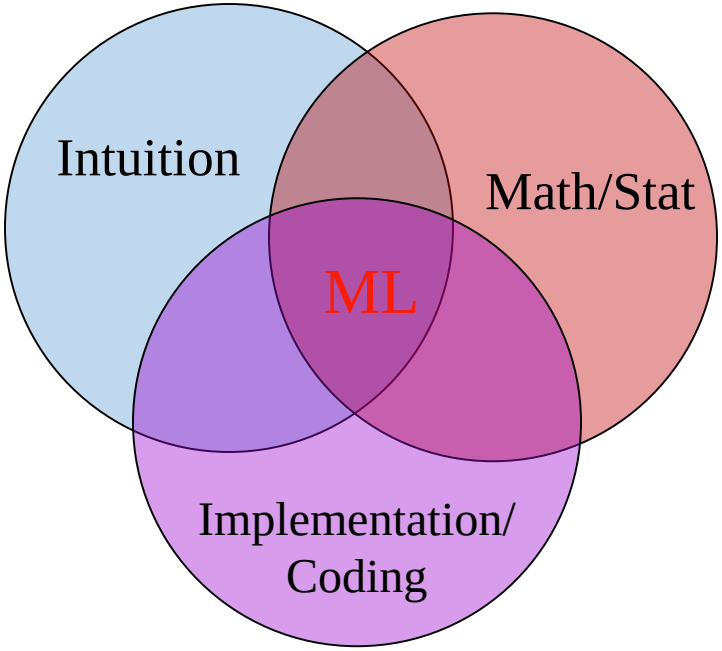
classifier $\mathbf{w} \cdot \mathbf{x} + b$

$\mathbf{w}, b$: model parameters



$\mathbf{w} \cdot \mathbf{x} + b$

A Big Picture



To do well in machine learning:

Intuition + Math/Stat +
Implementation/Coding

Basic concepts (supervised)

$$\mathbf{x}_i = (x_{i1}, ..., x_{im}), x_{ij} \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^m$$

$$y_i \in \mathbb{R}$$

Training (supervised)

$$S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$$

blood presure	age	male or female	weight (lb)	height (cm)
y_1=131	$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
y_2=150	$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
y_3=105	$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

In supervised setting during training, y_i (the solution) to each sample x_i is provided.

Supervised learning (training)

1. We are provided with a list of data samples in training.

2. In addition, each sample is associated with a label (-1 or +1 in binary classification), which is often manually (automatically, sometimes) delineated.

3. Our job is to use the provided training data to train to obtain a machine learning model for the future prediction.

4. In modern machine learning, the training process is often guided with principled mathematical formulations and algorithms.

Basic concepts (supervised)

$$\mathbf{x}_i = (x_{i1}, ..., x_{im}), x_{ij} \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^m$$

$$y_i \in \mathbb{R}$$

Testing:

$$S_{testing} = \{(\mathbf{x}_i), i = 1..u\}, what\ is\ y_i?$$

blood presure	age	male or female	weight (lb)	height (cm)
y_1=?	$x_{11} = 32$	$x_{12} = F$	$x_{13} = 130$	$x_{14} = 180$
y_2=?	$x_{21} = 11$	$x_{22} = M$	$x_{23} = 52$	$x_{24} = 135$

Supervised learning (testing)

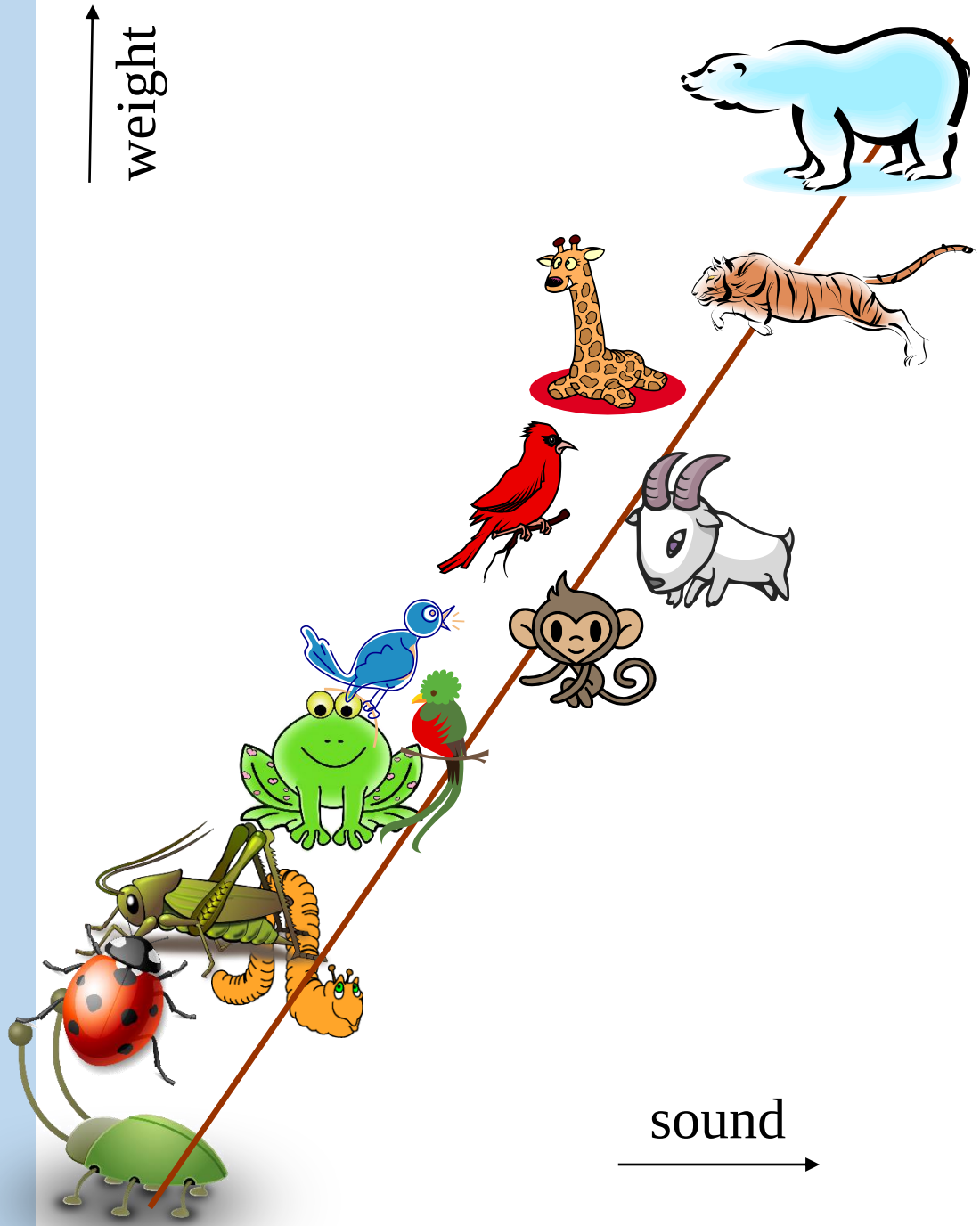
1. Given a model learned in the training process, we can make prediction in the “future” situations.

2. To be able to simulate and evaluate the performance of the learned model, we establish a “testing” dataset whose “ground-truth” labels are unknown for the model.

3. However, we still provide the “ground-truth” labels for the evaluation purposes.

Supervised learning

(Regression: weight prediction)



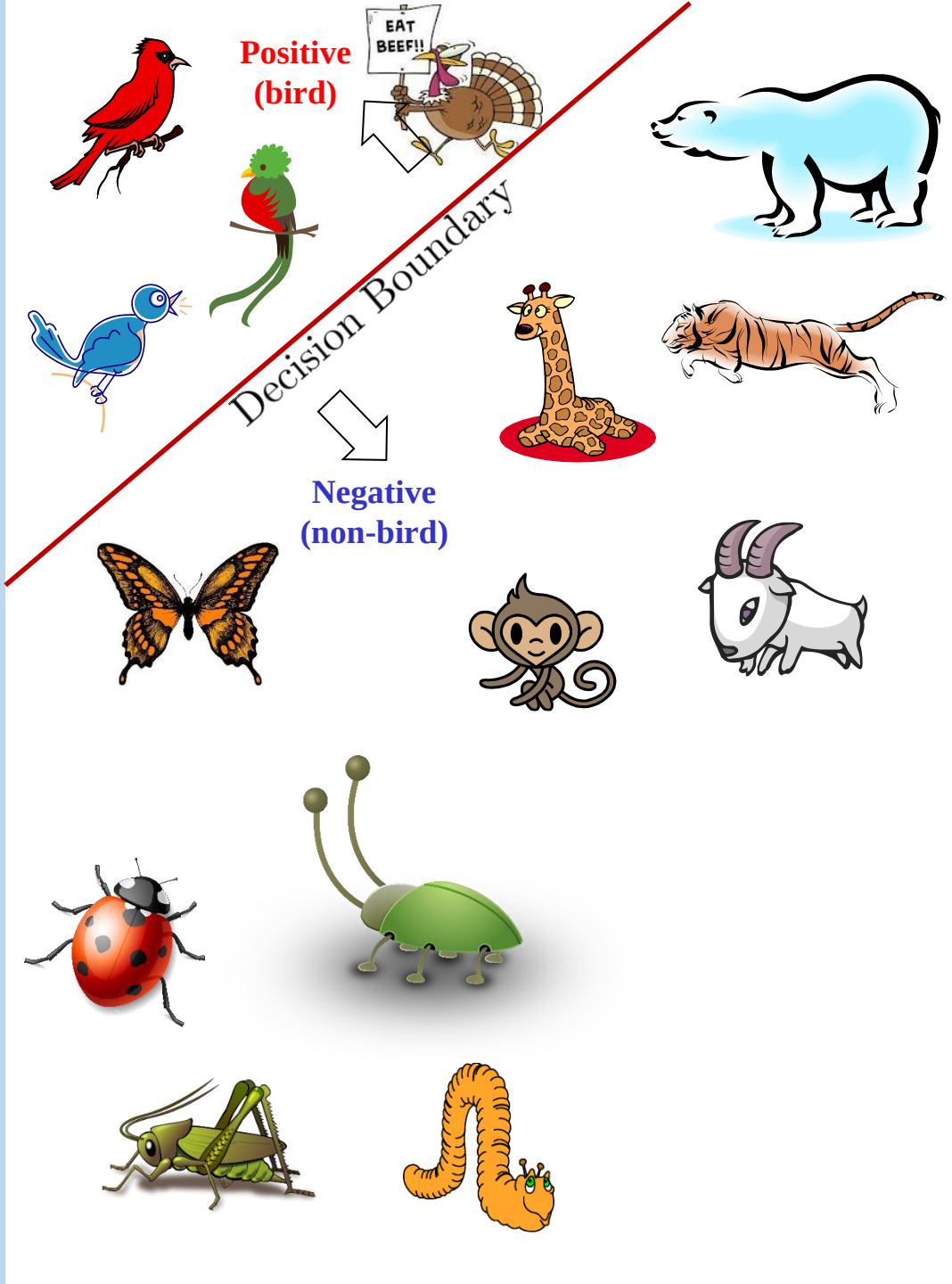
Regression

In regression, the “ground-truth” labels are continues values that are directly comparable.

That is, you can compare $y_1 = -1$ and $y_2 = 1$ to state for example $y_1 < y_2$ $(-1 < 1)$.

Supervised learning

(Classification)



Classification

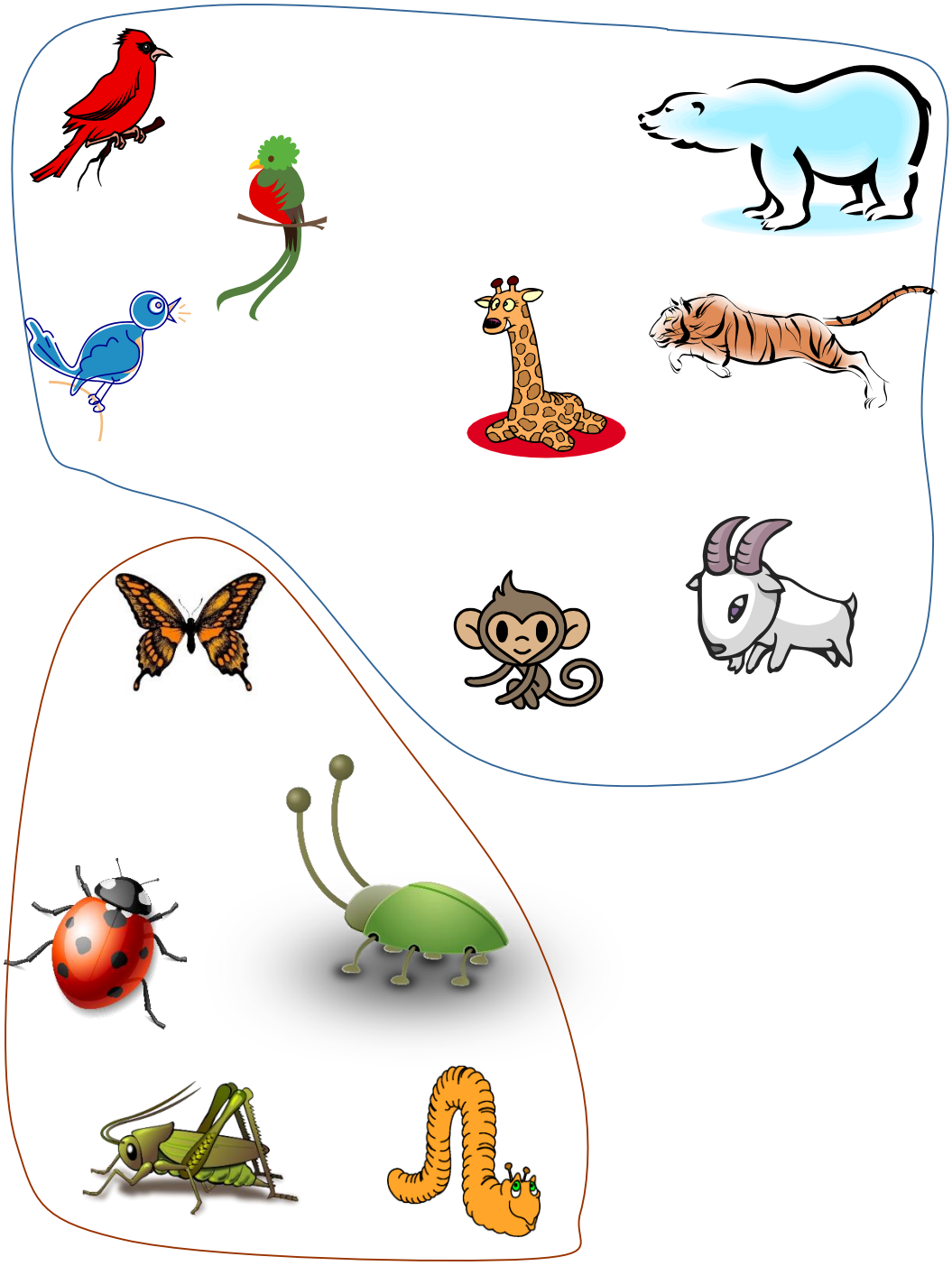
In classification, the "ground-truth" labels are categorical labels that are NOT directly comparable.

That is, you can not compare $y_1 =$ "$bird''$ and $y_2 =$ "$non - bird''$) to state for example:

"bird" $<$ "non-bird" (even we use $+1$ and -1 to denote their labels respectively).

Unsupervised learning

(Clustering)



Clustering (unsupervised)

In clustering, no “ground-truth” labels are given and our task is the separate the given samples into a number of ``groups”.

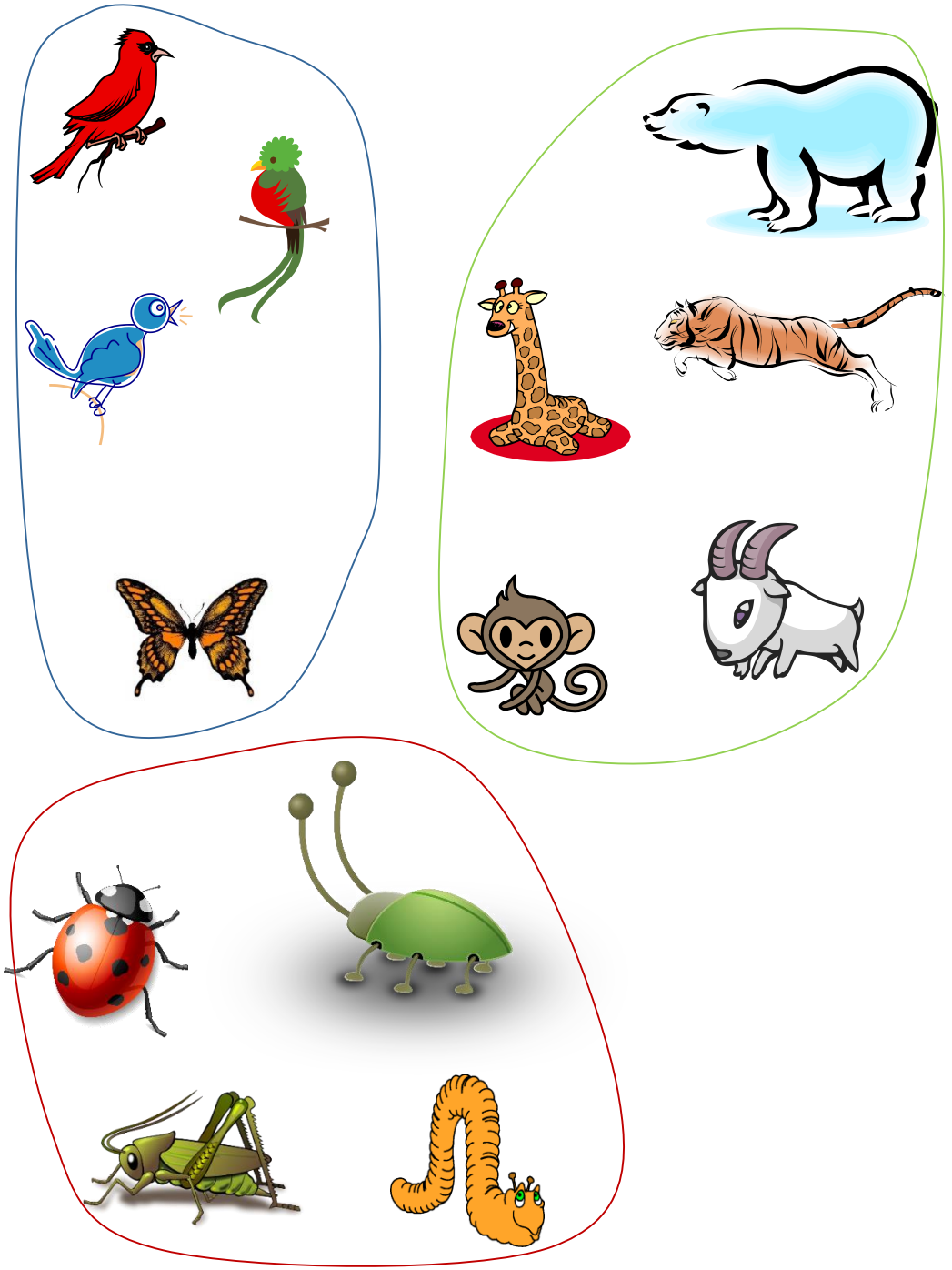
You need to decide:

1. How many groups are there.
2. How to group/divide them.

The clustering problem is unsupervised and consists of greater level of uncertainty, when compared with the supervised learning scenario.

Unsupervised learning

(Clustering results are not UNIQUE)



Features

What defines features?

How are they given?

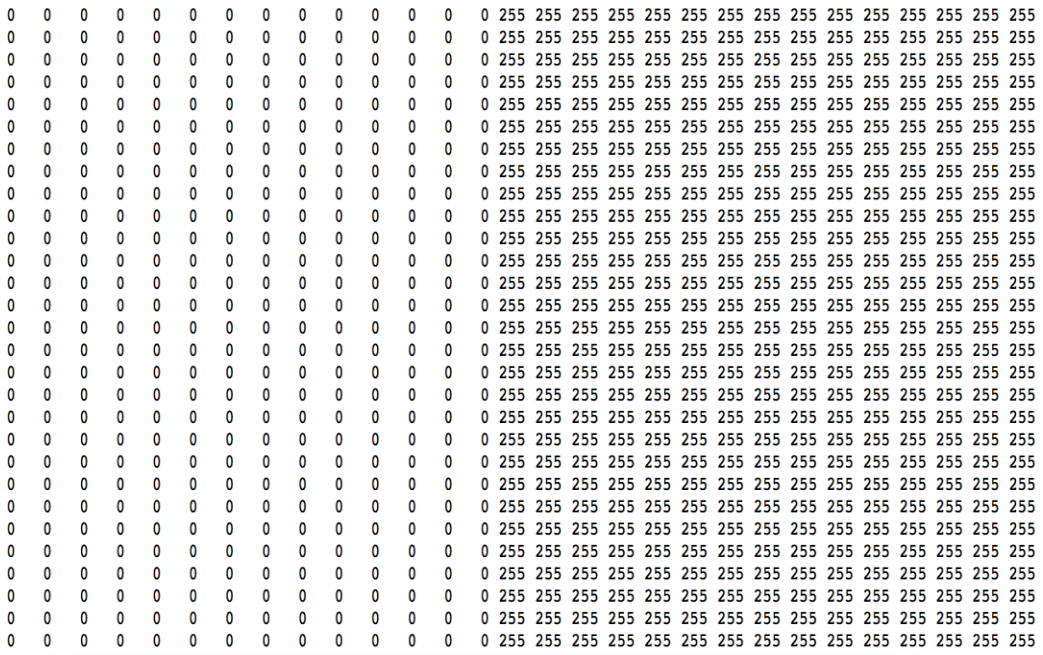
What is the difference between features and the input?

How to compute them?

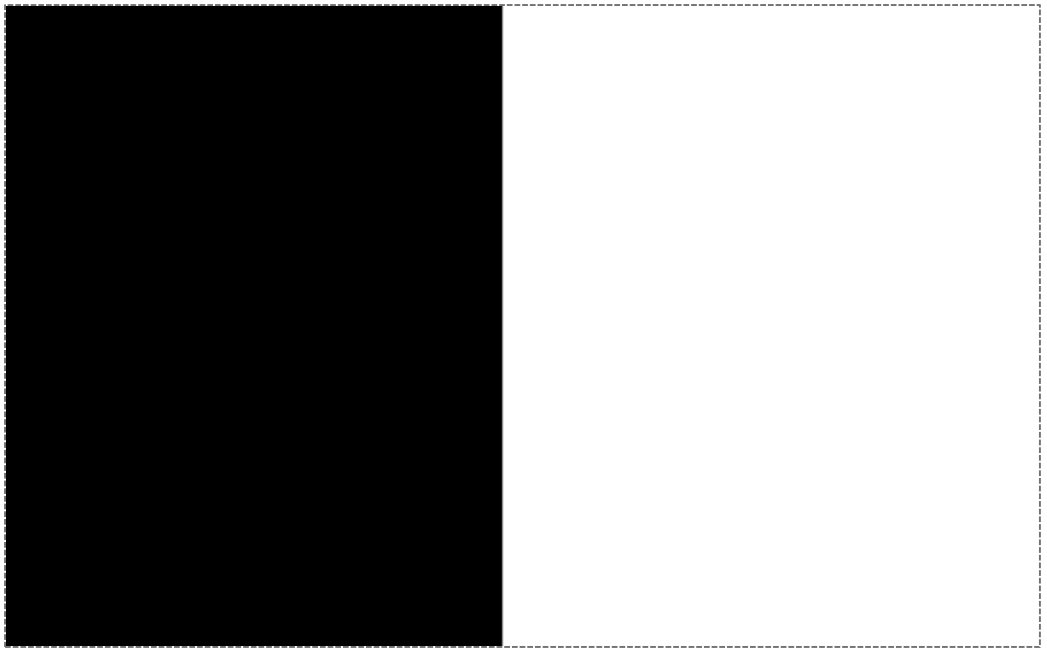
How to evaluate them?

How to use them?

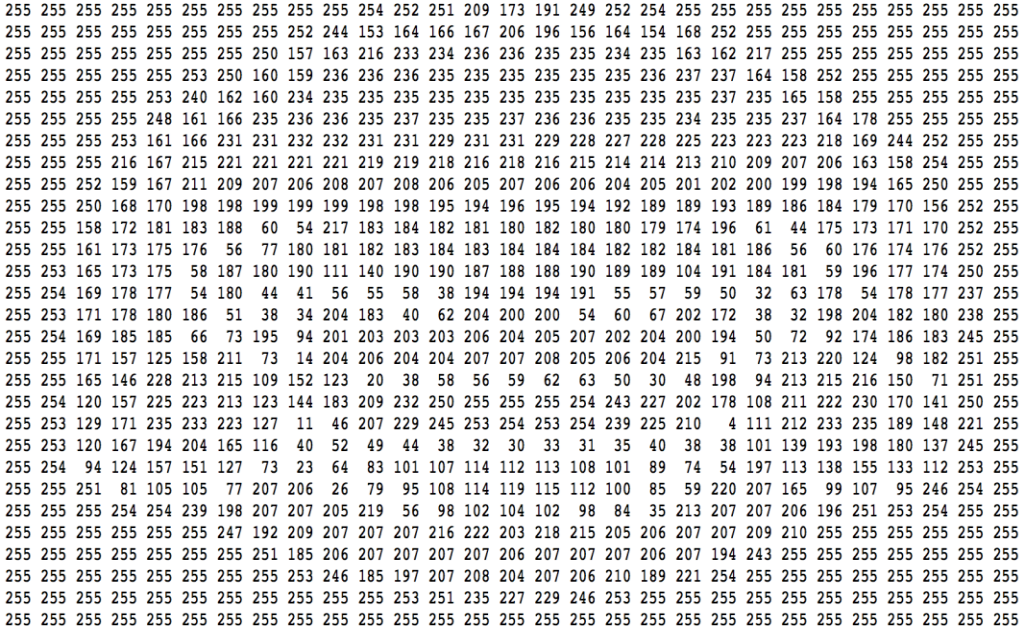
Why are features important



Why are features important



What is it?



A. Dog?

B. Chicken?

C. Car?

D. Elephant?

What is it?

255 255 255 255 255 255 255 255 255 255 254 252 251 209 173 191 249 252 254 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 252 244 153 164 166 167 206 196 156 164 154 168 252 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 250 157 163 216 233 234 236 236 235 235 234 235 163 162 217 255 255 255 255 255 255 255 255
255 255 255 255 255 253 250 160 159 236 236 236 235 235 235 235 235 235 236 237 237 164 158 252 255 255 255 255 255 255
255 255 255 255 253 240 162 160 234 235 235 235 235 235 235 235 235 235 235 235 237 235 165 158 255 255 255 255 255 255
255 255 255 255 248 161 166 235 236 236 235 237 235 235 237 236 236 235 235 234 235 235 237 164 178 255 255 255 255 255
255 255 255 253 161 166 231 231 232 232 231 231 229 231 231 229 228 227 228 225 223 223 223 218 169 244 252 255 255 255
255 255 255 216 167 215 221 221 221 221 219 219 218 216 218 216 215 214 214 213 210 209 207 206 163 158 254 255 255 255
255 255 252 159 167 211 209 207 206 208 207 208 206 205 207 206 206 204 205 201 202 200 199 198 194 165 250 255 255 255
255 255 250 168 170 198 198 199 199 199 198 198 195 194 196 195 194 192 189 189 193 189 186 184 179 170 156 252 255 255
255 255 158 172 181 183 188 60 54 217 183 184 182 181 180 182 180 180 179 174 196 61 44 175 173 171 170 252 255 255
255 255 161 173 175 176 56 77 180 181 182 183 184 183 184 184 184 182 182 184 181 186 56 60 176 174 176 252 255 255
255 253 165 173 175 58 187 180 190 111 140 190 190 187 188 188 190 189 189 104 191 184 181 59 196 177 174 250 255 255
255 254 169 178 177 54 180 44 41 56 55 58 38 194 194 194 191 55 57 59 50 32 63 178 54 178 177 237 255 255
255 253 171 178 180 186 51 38 34 204 183 40 62 204 200 200 54 60 67 202 172 38 32 198 204 182 180 238 255 255
255 254 169 185 185 66 73 195 94 201 203 203 203 206 204 205 207 202 204 200 194 50 72 92 174 186 183 245 255 255
255 255 171 157 125 158 211 73 14 204 206 204 204 207 207 208 205 206 204 215 91 73 213 220 124 98 182 251 255 255
255 255 165 146 228 213 215 109 152 123 20 38 58 56 59 62 63 50 30 48 198 94 213 215 216 150 71 251 255 255
255 254 120 157 225 223 213 123 144 183 209 232 250 255 255 255 254 243 227 202 178 108 211 222 230 170 141 250 255 255
255 253 129 171 235 233 223 127 11 46 207 229 245 253 254 253 254 239 225 210 4 111 212 233 235 189 148 221 255 255
255 253 120 167 194 204 165 116 40 52 49 44 38 32 30 33 31 35 40 38 38 101 139 193 198 180 137 245 255 255
255 254 94 124 157 151 127 73 23 64 83 101 107 114 112 113 108 101 89 74 54 197 113 138 155 133 112 253 255 255
255 255 251 81 105 105 77 207 206 26 79 95 108 114 119 115 112 100 85 59 220 207 165 99 107 95 246 254 255 255
255 255 255 254 254 239 198 207 207 205 219 56 98 102 104 102 98 84 35 213 207 207 206 196 251 253 254 255 255 255
255 255 255 255 255 255 247 192 209 207 207 207 216 222 203 218 215 205 206 207 207 209 210 255 255 255 255 255 255 255
255 255 255 255 255 255 255 251 185 206 207 207 207 207 206 207 207 207 206 207 194 243 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 253 246 185 197 207 208 204 207 206 210 189 221 254 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 253 251 235 227 229 246 253 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255

How to represent a raw image?

Each image is turned into a vector of size: $n \times m$

m: width

n : height

255	255	255	255	255	255	255	255	255	255	254	252	251	209	
255	255	255	255	255	255	255	255	255	252	244	153	164	166	167
255	255	255	255	255	255	255	250	157	163	216	233	234	236	
255	255	255	255	255	253	250	160	159	236	236	236	235	235	
255	255	255	255	253	240	162	160	234	235	235	235	235	235	
255	255	255	255	248	161	166	235	236	236	235	237	235	235	
255	255	255	253	161	166	231	231	232	232	231	231	229	231	
255	255	255	216	167	215	221	221	221	221	219	219	218	216	
255	255	252	159	167	211	209	207	206	208	207	208	206	205	
255	255	250	168	170	198	198	199	199	199	198	198	195	194	
255	255	158	172	181	183	188	60	54	217	183	184	182	181	
255	255	161	173	175	176	56	77	180	181	182	183	184	183	
255	253	165	173	175	58	187	180	190	111	140	190	190	187	
255	254	169	178	177	54	180	44	41	56	55	58	38	194	
255	253	171	178	180	186	51	38	34	204	183	40	62	204	
255	254	169	185	185	66	73	195	94	201	203	203	203	206	
255	255	171	157	125	158	211	73	14	204	206	204	204	207	
255	255	165	146	228	213	215	109	152	123	20	38	58	56	

Image Matrix

Image Vector

How to represent a raw image?

Each image is turned into a vector of size: $n \times m$

m: width

n : height

255	255	255	255	255	255	255	255	255	255	254	252	251	209
255	255	255	255	255	255	255	255	255	252	244	153	164	166	167
255	255	255	255	255	255	255	255	250	157	163	216	233	234	236
255	255	255	255	255	255	253	250	160	159	236	236	236	235	235
255	255	255	255	255	253	240	162	160	234	235	235	235	235	235
255	255	255	255	248	161	166	235	236	236	235	237	235	235
255	255	255	253	161	166	231	231	232	232	231	231	229	231
255	255	255	216	167	215	221	221	221	221	219	219	218	216
255	255	252	159	167	211	209	207	206	208	207	208	206	205
255	255	250	168	170	198	198	199	199	199	198	198	195	194
255	255	158	172	181	183	188	60	54	217	183	184	182	181
255	255	161	173	175	176	56	77	180	181	182	183	184	183
255	253	165	173	175	58	187	180	190	111	140	190	190	187
255	254	169	178	177	54	180	44	41	56	55	58	38	194
255	253	171	178	180	186	51	38	34	204	183	40	62	204
255	254	169	185	185	66	73	195	94	201	203	203	203	206
255	255	171	157	125	158	211	73	14	204	206	204	204	207
255	255	165	146	228	213	215	109	152	123	20	38	58	56

Image Matrix

Image Vector

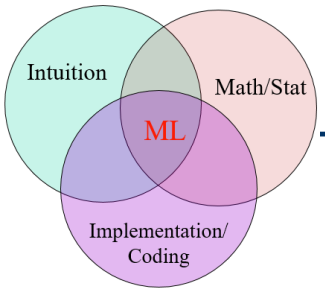
255 255 255 255 255 255 255 255 255 255 254 252 251 209 255 255 255 255 255 255 255 255 252 244 153 164 166 167 255 255 255 255 255 255 255 250 157 163 216 233 234 236

Machine learning

A major difference between the “modern machine learning” and “traditional artificial intelligence (AI)” fields is in the difference of data representations.

Machine learning: canonical numerical values. $\mathbb{R}$

Traditional AI: mixed raw inputs including continues, discrete, and categorical.



Recap: Supervised Learning

Intuition: A prediction task with a clear objective (e.g. a yes or no decision, which school to go to, a price to estimate, etc.) in which some history data for training can be acquired with the known prediction results already.

Math:

Training: $S_{training} = \{(\mathbf{x}_i, y_i), i = 1..n\}$

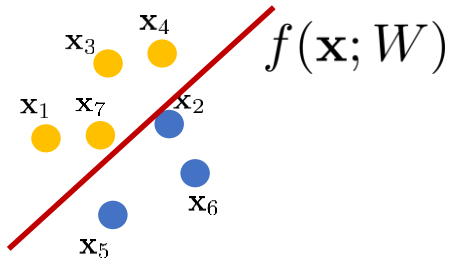
Testing: $S_{testing} = \{(\mathbf{x}_i), i = 1..u\}, what\ is\ y_i?$

Three key variables we are dealing with

Input: $\mathbf{x} = (x_1, x_2, ...)$

Label: $y \in 0, 1$

Model parameter: W



Features

Each data sample is often described by a set of features.

$\mathbf{x} = (x_1, x_2, ...)$

x_1: color
x_2 weight
...

Features are often multi-modality.

Three typical types:

1. Real number (continues), $x \in \mathbb{R}$.

   e.g. weight

2. Integer (continues), $x \in \mathbb{Z}$.

   e.g. age

3. Categorical (disjoint), $x \in \{red, blue, green, ...\}$.

   e.g. color

Feature representation

A major difference between the "modern machine learning" and "traditional artificial intelligence (AI)" fields is in the difference of data representations.

Machine learning: canonical numerical values. $\mathbb{R}$

Why do we care about creating a canonical feature representation?

Which of the following statements is FALSE?

A. To build generic mathematical formulations.
B. To scale up the learning/training process.
C. To build transparent machine learning algorithms.
D. To make different machine learning algorithms directly comparable.
E. To make a faster prediction in the testing time.

Feature representation

Why do we care about creating a canonical feature representation?

Which of the following statements is FALSE?

A. To build generic mathematical formulations.
B. To scale up the learning/training process.
C. To build transparent machine learning algorithms.
D. To make different machine learning algorithms directly comparable.
E. To make a faster prediction in the testing time.

Mathematical representation for features

$$S = \{(\mathbf{x}_i), i = 1..n\} \qquad \mathbf{x}_i = (x_{i1}, ..., x_{im})$$

What if it is a city: $x_{i2} \in \{LosAngeles, SanDiego, Irvine\}$?

We ue a coding strategy by expanding the features.
For N number of possible states, we expand the features into N-dimenstional.

One-hot encoding:

	coded values
Los Angeles	1, 0, 0
San Diego	0, 1, 0
Irvine	0, 0, 1

Pros: we can naturally deal with any type of input (can associate confidence directly).

Cons: the feature dimension has become much larger.

One-hot encoding

- "One-hot encoding" is a standard technique that turns categorical features into general real numbers that can be used alone/together with other features in continues or discrete values.
- Feature values after one-hot encoding can take standard mathematical operations, just like standard real numbers.
- One-hot encoding can also generate soft values with a probabilistic interpretation, that is measurable and comparable.
- One-hot encoding gains in its convenience in a canonical mathematical representation by sacrificing in the space complexity: one category of k classes is turned into k real numbers in [0, 1].
- There are other encoding strategies, such as Word2Vec, that can turn categorical features into real numbers.

One-hot encoding

What if we simply:

	coded values
Los Angeles	1
San Diego	2
Irvine	3

This is wrong since it implies:

Los Angeles (1) + San Diego (2) = Irvine (3)

Irvin (3) > San Diego (2)

One-hot encoding with soft values

What if it is a city: $x_{i2} \in \{LosAngeles, SanDiego, Irvine\}$?

One-hot encoding:

	coded values		
70% Los Angeles and 30% San Diego	0.7,	0.3,	0
50% San Diego and 50% Irvine	0,	0.5,	0.5
Irvine	0,	0,	1

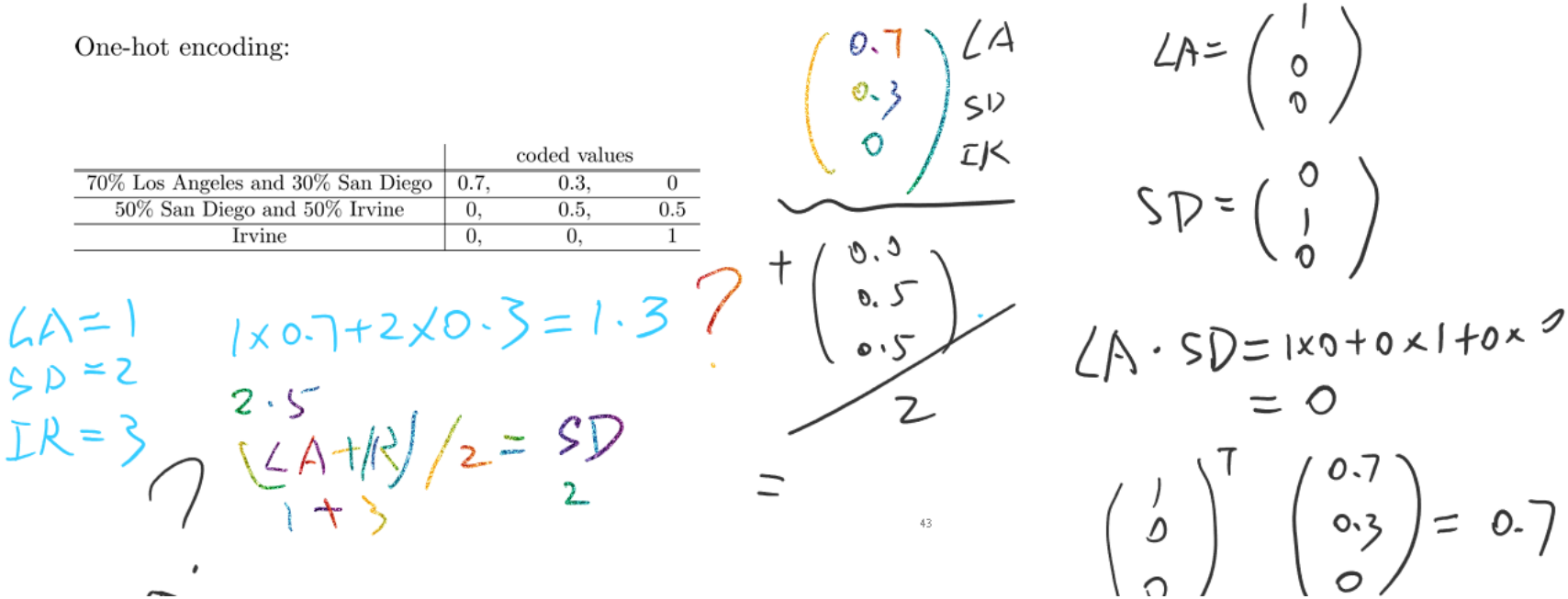
One-hot encoding

One-hot encoding with soft values

What if it is a city: $x_{i2} \in \{LosAngeles, SanDiego, Irvine\}$?

One-hot encoding:

	coded values		
70% Los Angeles and 30% San Diego	0.7,	0.3,	0
50% San Diego and 50% Irvine	0,	0.5,	0.5
Irvine	0,	0,	1



Data representation

$$S = \{\mathbf{x}_i, i = 1..n\} \qquad \mathbf{x}_i = (x_{i1}, ..., x_{im})$$

age	male or female	weight (lb)	height (cm)
$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

If we write each sample as a row vector:

$$\mathbf{x}_1 = (22, 0, 1, 160, 180)$$

$$X = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix} \quad X \in \mathbb{R}^{n \times m}$$

$$X = \begin{pmatrix} 22 & 0 & 1 & 160 & 180 \\ 51 & 0 & 1 & 190 & 175 \\ 43 & 1 & 0 & 120 & 165 \end{pmatrix}$$

Data representation

$$S = \{\mathbf{x}_i, i = 1..n\} \qquad \mathbf{x}_i = (x_{i1}, ..., x_{im})^T$$

age	male or female	weight (lb)	height (cm)
$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

More often we write each sample as a COLUMN vector:

$$\mathbf{x}_1 = \begin{pmatrix} 22 \\ 0 \\ 1 \\ 160 \\ 180 \end{pmatrix}$$

$$X = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \quad X \in R^{m \times n}$$

$$X = \begin{pmatrix} 22 & 51 & 43 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 160 & 190 & 120 \\ 180 & 175 & 165 \end{pmatrix}$$

Row or column vector?

$$\mathbf{x} = (22, 0, 1, 160, 180)$$

$$\mathbf{x} = \begin{pmatrix} 22 \\ 0 \\ 1 \\ 160 \\ 180 \end{pmatrix}$$

row vector

column vector

$$\mathbf{w} = (w_1, w_2, w_3, w_4, w_5)$$

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}$$

$$\mathbf{w} \cdot \mathbf{x} \equiv < \mathbf{w}, \mathbf{x} > \equiv \mathbf{w}\mathbf{x}^T$$

$$\mathbf{w} \cdot \mathbf{x} \equiv < \mathbf{w}, \mathbf{x} > \equiv \mathbf{w}^T\mathbf{x}$$

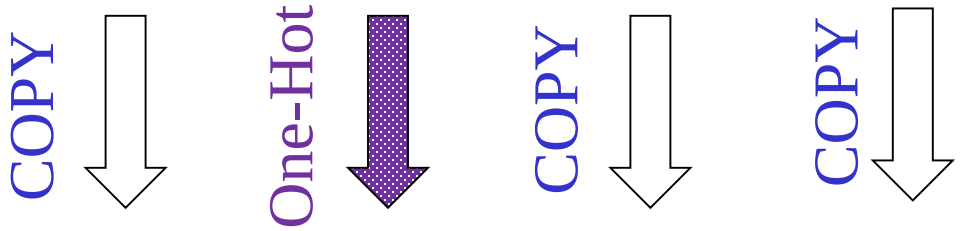
Row vector or column vector can be alternatively used, but make sure you stick to the same definition/setting in your problem formulations.

It is slightly more common to use the column vector representation:

Building a canonical mathematical representation

age	male or female	weight (lb)	height (cm)
$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

discrete categorical continues continues



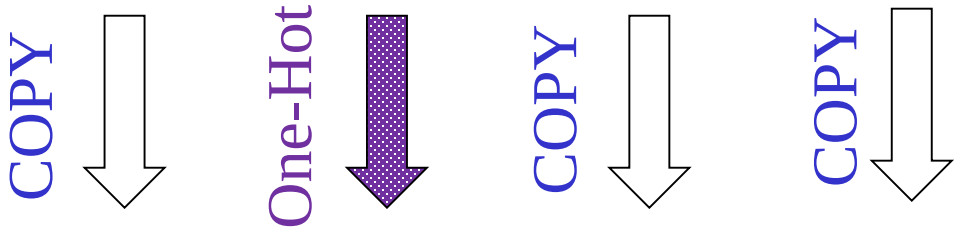
$$\begin{pmatrix} 22 & 0 & 1 & 160 & 180 \\ 51 & 0 & 1 & 190 & 175 \\ 43 & 1 & 0 & 120 & 165 \end{pmatrix}$$

Building a canonical mathematical representation

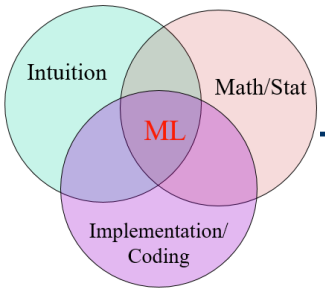
For a categorical feature that has only two possible categories, we sometimes use only one entry to reduce the encoded feature size.

age	male or female	weight (lb)	height (cm)
$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

discrete categorical continues continues



$$\begin{pmatrix} 22 & 0 & 160 & 180 \\ 51 & 0 & 190 & 175 \\ 43 & 1 & 120 & 165 \end{pmatrix}$$



Recap: One-hot Encoding

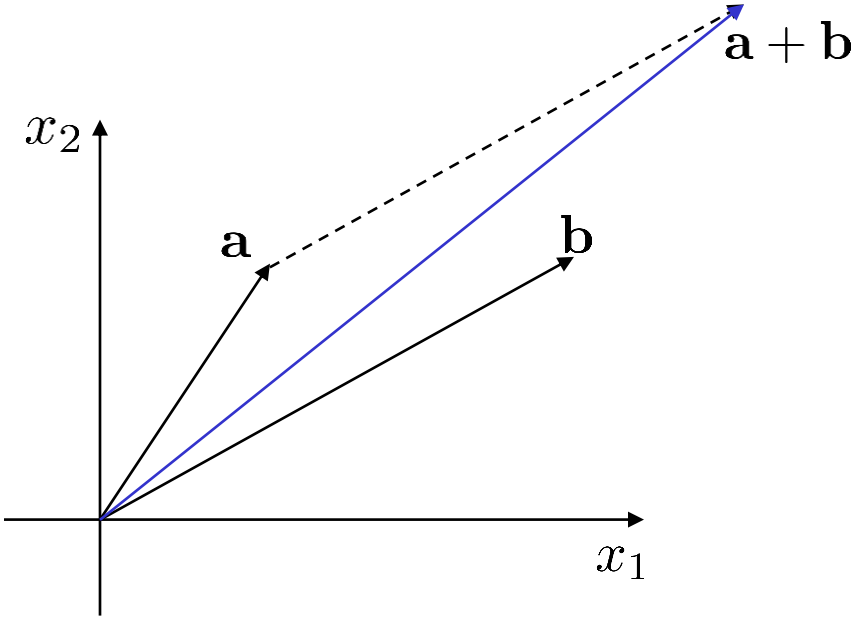
Intuition: Turn categorical features (e.g. city, school, name, etc.) to which arithmetic operations cannot be applied into real numbers for direct mathematical manipulations. After the transformation, the input features originally described as categories have no difference to those in real numbers. They will be treated canonically in the later mathematical and statistical functions.

Math: $x_{raw} \in \{Los\ Angeles, San\ Diego, Irvine\}$

category	$\mathbf{x}_{one-hot}$
Los Angeles	1, 0, 0
San Diego	0, 1, 0
Irvine	0, 0, 1

Vector

Addition:



$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\mathbf{a}+\mathbf{b} = \begin{pmatrix} a_1+b_1 \\ a_2+b_2 \\ a_3+b_3 \end{pmatrix}$$

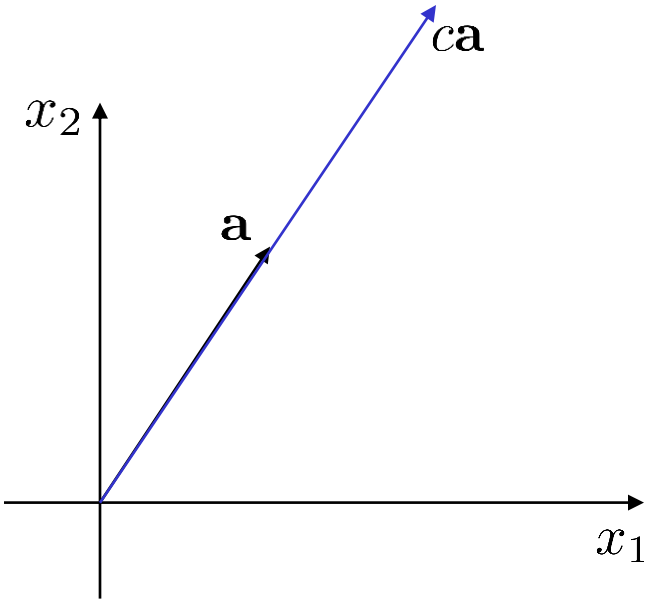
It's still a vector in the same space as a and b.

Visual illustration by 3Blue1Brown

https://www.youtube.com/watch?v=fNk_zzaMoSs

Vector

Scaling:



$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$c \in R$$

$$c\mathbf{a} = \begin{pmatrix} c \times a_1 \\ c \times a_2 \\ c \times a_3 \end{pmatrix}$$

It's still a vector in the same space as a.

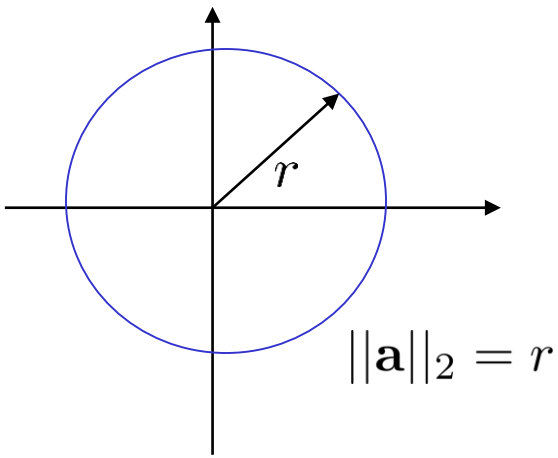
Norm

$\mathbf{a} = (a_1, a_2, ..., a_n),\ a_i \in \mathbb{R}$

L2 Norm:

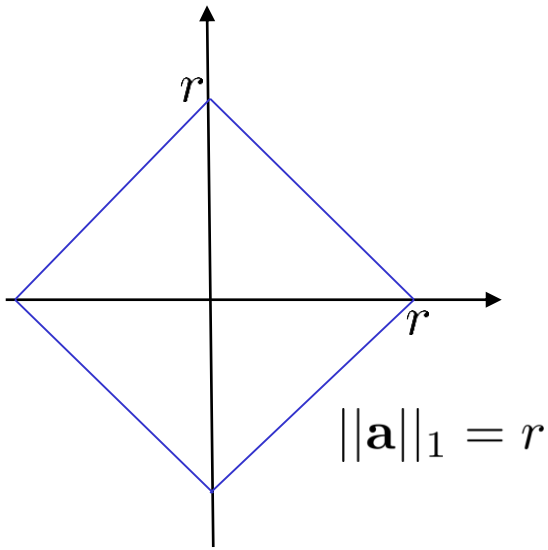
$$||\mathbf{a}||_2 = \sqrt{\sum_{i=1}^{n} a_i^2}$$

$$||\mathbf{a}||_2^2 = \sum_{i=1}^{n} a_i^2$$



L1 Norm:

$$||\mathbf{a}||_1 = \sum_{i=1}^{n} |a_i|$$



Norm

$$||\mathbf{a}||_2 = \sqrt{\sum_{i=1}^{n} a_i^2}$$

We use norm (typically L2) to measure the "length" (magnitude) of a vector.

Norm is a non-negative real number (scalar).

In machine learning,

1. we sometimes use norm to normalize an in put vector,

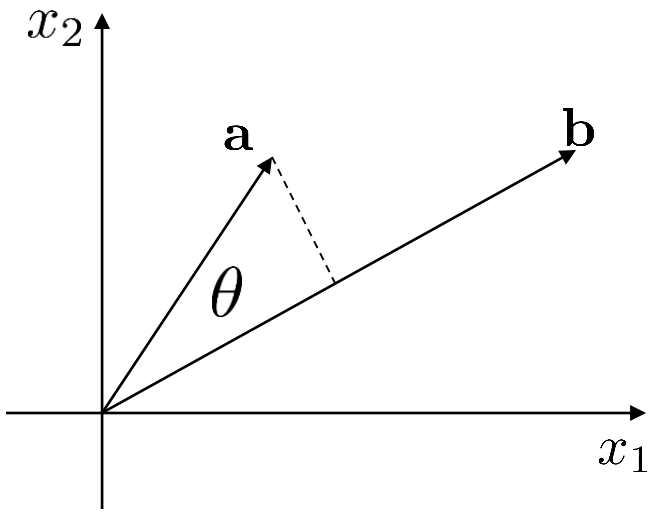
   $\mathbf{a}/||\mathbf{a}||_2$

2. or to use norm as a regularization to prevent overfitting for the model parameter

   $||\mathbf{w}||_2$

Vector: Projection (inner product)

(one of the most important concepts in machine learning)



$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$< \mathbf{a}, \mathbf{b} > \equiv \mathbf{a} \cdot \mathbf{b} \equiv \mathbf{a}^T \mathbf{b} \equiv a_1 b_1 + a_2 b_2 + a_3 b_3$$

It's a scalar!

$$cos(\theta) = \frac{<\mathbf{a},\mathbf{b}>}{||\mathbf{a}||_2 \times ||\mathbf{b}||_2}$$

Significance of the dot product between two vectors

"Dot product" outputs a scalar value and it is arguably the most important mathematical operation in machine learning.

$$< \mathbf{a}, \mathbf{b} > \equiv \mathbf{a} \cdot \mathbf{b} \equiv \mathbf{a}^T \mathbf{b}$$
$$\equiv < \mathbf{b}, \mathbf{a} > \equiv \mathbf{b} \cdot \mathbf{a} \equiv \mathbf{b}^T \mathbf{a}$$

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Why?

"Dot product" computes for the magnitude for the projection from one vector to the other, which essentially measures the "similarity" between two vectors.

For two unit vectors (L2 norm being 1), their dot product outputs the largest value 1 when they well aligned (same), and otherwise 0 when they are orthogonal (different) to each other.

Significance of the dot product between two vectors

	fly?	laying eggs?	weight (lb)
sparrow	yes	yes	0.087
chipmunk	no	no	0.19
bat	yes	no	0.09

Feature representation (one-hot encoded).

$$\text{sparrow} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0.087 \end{pmatrix} \qquad \text{chipmunk} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0.19 \end{pmatrix} \qquad \text{bat} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0.09 \end{pmatrix}$$

$\text{sparrow} \cdot \text{chipmunk} = 0.01653$ very different!

$\text{sparrow} \cdot \text{bat} = 1.00783$

$\text{chipmunk} \cdot \text{bat} = 1.0171$

More illustrations about vector operations

https://www.youtube.com/watch?v=fNk_zzaMoSs&t=3s

3Blue1Brown

Basics about data and linear algebra operations

$$S = \{(\mathbf{x}_i, y_i), i = 1..n\} \qquad y_i \in \{-1, +1\}$$

	age	male or female	weight (lb)	height (cm)
$y_1 = -1$ (negative)	$x_{11} = 22$	$x_{12} = M$	$x_{13} = 160$	$x_{14} = 180$
$y_2 = +1$ (positive)	$x_{21} = 51$	$x_{22} = M$	$x_{23} = 190$	$x_{24} = 175$
$y_3 = +1$ (positive)	$x_{31} = 43$	$x_{32} = F$	$x_{33} = 120$	$x_{34} = 165$

$$X = \begin{pmatrix} 22 & 1 & 0 & 160 & 180 \\ 51 & 1 & 0 & 190 & 175 \\ 43 & 0 & 1 & 120 & 165 \end{pmatrix} \qquad Y = \begin{pmatrix} -1 \\ +1 \\ +1 \end{pmatrix}$$

$$W = \begin{pmatrix} 0.075 \\ 0 \\ 0 \\ -0.007 \\ -0.008 \end{pmatrix} \qquad \hat{Y} = XW = \begin{pmatrix} -0.91 \\ 1.095 \\ 1.065 \end{pmatrix}$$

Matrix multiplication

Vector:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$AB = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3$$

$$AB \neq BA$$

$$BA = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} b_1 a_1 & b_1 a_2 & b_1 a_3 \\ b_2 a_1 & b_2 a_2 & b_2 a_3 \\ b_3 a_1 & b_3 a_2 & b_3 a_3 \end{pmatrix}$$

Matrix multiplication

Matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}$$

$$AB = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}$$

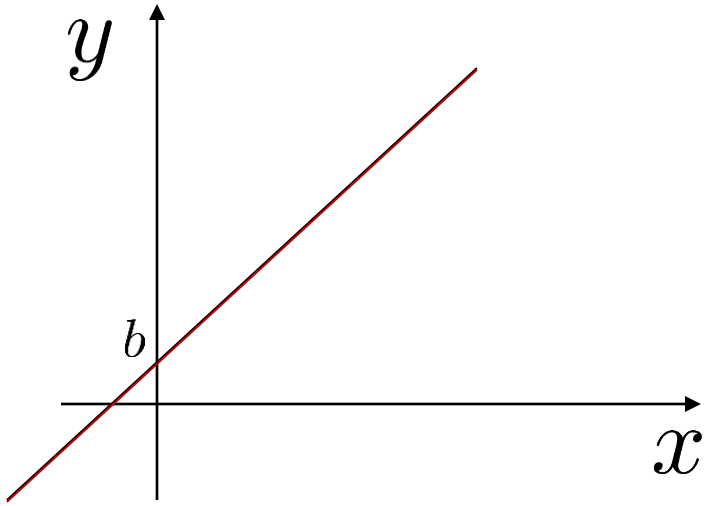
$$= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{pmatrix}$$

Calculus

Scalar:

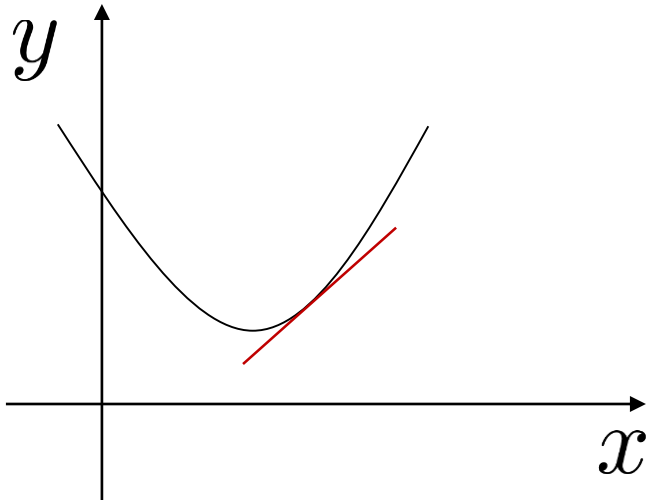
$$y = ax + b$$

$$\frac{dy}{dx} = a$$



$$y = ax^2 + bx + c$$

$$\frac{dy}{dx} = 2ax + b$$



Vector calculus

Vector derivatives

Vector-by-scalar

$$\mathbf{y}(x) = \begin{pmatrix} y_1(x) & y_2(x) & y_3(x) \end{pmatrix}$$

$$\frac{d\mathbf{y}(x)}{dx} = \begin{pmatrix} \frac{dy_1(x)}{dx} & \frac{dy_2(x)}{dx} & \frac{dy_3(x)}{dx} \end{pmatrix}$$

Vector-by-vector

$$\mathbf{y}(\mathbf{x}) = \begin{pmatrix} y_1(\mathbf{x}) & , \ldots, & y_m(\mathbf{x}) \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_1 & , \ldots, & x_n \end{pmatrix}$$

$$\frac{d\mathbf{y}(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{dy_1(\mathbf{x})}{dx_1} & , \ldots, & \frac{dy_m(\mathbf{x})}{dx_1} \\ . & . & . \\ \frac{dy_1(\mathbf{x})}{dx_n} & , \ldots, & \frac{dy_m(\mathbf{x})}{dx_n} \end{pmatrix}$$

Vector calculus

Vector-by-vector

$$A = \begin{pmatrix} a_{11} & , \ldots, & a_{1m} \\ . & . & . \\ a_{n1} & , \ldots, & a_{nm} \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ . \\ x_m \end{pmatrix}$$

$$\frac{\partial A\mathbf{x}}{\partial \mathbf{x}} = A$$

numerator form

$$\frac{\partial \mathbf{x}^T A^T}{\partial \mathbf{x}} = A$$

denominator form

Vector calculus

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by y and $\mathbf{x}^\mathsf{T}$	Denominator layout, i.e. by $\mathbf{y}^\mathsf{T}$ and x
a is not a function of **x**	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
A is not a function of **x**	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
A is not a function of **x**	$\frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
a = *a*(**x**), **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial a}{\partial \mathbf{x}}$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial a}{\partial \mathbf{x}}\mathbf{u}^\top$
A is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial (\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
u = **u**(**x**)	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$

https://en.wikipedia.org/wiki/Matrix_calculus

Why is computing the vector derivatives so important?

Which statement is NOT true?

A. Help us better understand the machine learning problem.

B. Can always be accurately computed.

C. Gives a principled and general way for training a machine learning algorithm.

D. Scales up to big high-dimensional data.

Why is computing the vector derivatives so important?

Which statement is NOT true?

A. Help us better understand the machine learning problem.

☆B. Can always be accurately computed.

C. Gives a principled and general way for training a machine learning algorithm.

D. Scales up to big high-dimensional data.

Matrix calculus

Matrix-by-scalar

$$Y(x) = \begin{pmatrix} y_{11}(x) & , \ldots, & y_{1m}(x) \\ . & . & . \\ y_{n1}(x) & , \ldots, & y_{nm}(x) \end{pmatrix}$$

$$\frac{dY(x)}{dx} = \begin{pmatrix} \frac{dy_{11}(x)}{dx} & , \ldots, & \frac{dy_{1m}(x)}{dx} \\ \frac{dy_{n1}(x)}{dx} & , \ldots, & \frac{dy_{nm}(x)}{dx} \end{pmatrix}$$

Matrix calculus

Scalar-by-vector

$$A = \begin{pmatrix} a_{11} & , \ldots, & a_{1n} \\ . & . & . \\ a_{n1} & , \ldots, & a_{nn} \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ . \\ x_n \end{pmatrix}$$

$$\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{x}^T A$$

Matrix calculus

Condition	Expression	Numerator layout, i.e. by x^T; result is row vector	Denominator layout, i.e. by x; result is column vector
a is not a function of **x**	$\frac{\partial(\mathbf{a}\cdot\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial(\mathbf{x}\cdot\mathbf{a})}{\partial\mathbf{x}} =$ $\frac{\partial\mathbf{a}^\top\mathbf{x}}{\partial\mathbf{x}} = \frac{\partial\mathbf{x}^\top\mathbf{a}}{\partial\mathbf{x}} =$	$\mathbf{a}^\top$	$\mathbf{a}$
A is not a function of **x** **b** is not a function of **x**	$\frac{\partial\mathbf{b}^\top\mathbf{A}\mathbf{x}}{\partial\mathbf{x}} =$	$\mathbf{b}^\top\mathbf{A}$	$\mathbf{A}^\top\mathbf{b}$
A is not a function of **x**	$\frac{\partial\mathbf{x}^\top\mathbf{A}\mathbf{x}}{\partial\mathbf{x}} =$	$\mathbf{x}^\top(\mathbf{A}+\mathbf{A}^\top)$	$(\mathbf{A}+\mathbf{A}^\top)\mathbf{x}$
A is not a function of **x** **A** is symmetric	$\frac{\partial\mathbf{x}^\top\mathbf{A}\mathbf{x}}{\partial\mathbf{x}} =$	$2\mathbf{x}^\top\mathbf{A}$	$2\mathbf{A}\mathbf{x}$
A is not a function of **x**	$\frac{\partial^2\mathbf{x}^\top\mathbf{A}\mathbf{x}}{\partial\mathbf{x}^2} =$	$\mathbf{A}+\mathbf{A}^\top$	
A is not a function of **x** **A** is symmetric	$\frac{\partial^2\mathbf{x}^\top\mathbf{A}\mathbf{x}}{\partial\mathbf{x}^2} =$	$2\mathbf{A}$	

https://en.wikipedia.org/wiki/Matrix_calculus

Vector and Matrix Calculus

The main reason to study/use vector calculus is to formulate machine learning problems using canonical mathematical representations that can be accepted by the generic machine learning algorithms including neural networks, decision tree, nearest neighborhood, boosting, logistic regression classifier etc.

Using vector representations with vector calculus significantly facilitates the understanding, training, evaluating, scaling up, and transferring of the machine learning algorithms with significantly reduced overhead and customization.

Vector and Matrix Calculus

One way to master the concept of vector representation and calculus is by simplifying (conceptually) the formulation into scalar cases, which can be understood more easily.

Taking the derivatives w.r.t. scalar and vector gives us a strong leverage in using vector calculus for performing optimization to train the various machine learning algorithms.