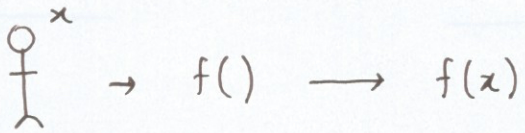
WEEK 1

THINKING ABOUT DATA

ATTRIBUTE-VALUE REPRESENTATION



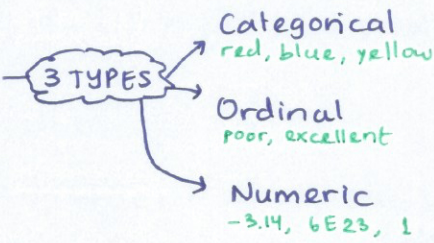
→ How do we represent 🧍 mathematically?

- Depends on what we're trying to do
  - Deciding to loan money?
  - Predicting gender
- Represent 🧍 as a set of **attribute-value pairs**

  i.e. x = [height = 180cm, eyes = "blue", job = "student"]

  ↳ NO ordering/structure!

→ If structure is essential, embed it in the attributes.



1) Categorical Attributes

- Each instance falls into one of a set of categories
  - Categories are mutually exclusive!
- Categories usually encoded as numbers
  - No natural ordering to categories
  - Only equality testing (=, ≠) is meaningful
- Synonymy is a major challenge for real datasets:

  i.e. country == folk ? house == techno ?

2) Ordinal Attributes

- Like categorical, but there is a natural ordering to categories

  i.e. Likert scale: [disagree, neutral, agree, strongly agree]
- Encoded as numbers to preserve ordering
  - Meaningful to compare values (<, =, >)
  - Should not do arithmetics w/ the numbers
- Sometimes hard to differentiate w/ categorical

  i.e. Does [single, married, divorced] have a natural ordering?

3) Numeric Attributes

- Integers / Real no.
- Usually want to **normalize** values
  - If not, learning algos will get confused; all projection methods will do poorly
  - Bring data to the same scale so all units are roughly comparable across all the numeric attributes
    - ↳ zero mean & unit variance: $x' = \frac{(x - \text{mean})}{\text{std. deviation}}$
    - ↳ Sometimes we want [0, 1]: $x' = \frac{(x - \min)}{(\max - \min)}$
- Sensitive to **outliers** (extreme values)
  - ↳ Must handle this before normalization!

HOWEVER, sometimes these extreme values are NOT outliers (i.e. Personal Wealth)

↓ ISSUES W/ NUMERIC ATTRIBUTES

a) Skewed Distributions

→ systematic extreme values

→ affects regression, naive bayes, kNN; but NOT decision trees

→ How to fix? → log(x), arctan(x), cumulative distribution fn } then normalize

b) Non-monotonic effect of attributes

→ affects regression, naive bayes, DTs; but less less important for kNN

i.e. There is a monotonic relationship between net worth & lending risk

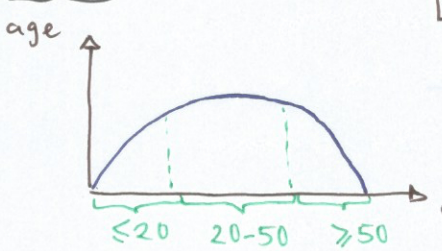
↑ Net worth — ↓ lending risk

There is a non-monotonic relationship between age & winning a marathon

PROBLEMATIC! (for learning algos)

→ How to fix? → Quantization

↳ can be unsupervised



↳ Use categories!; Convert the values to a variety of overlapping numerical ranges

HOW TO REPRESENT IMAGES

→ Previous examples have obvious attributes, but how do we represent bitmap images?

→ Think about what we're trying to accomplish:

- We're learning a predictor: $f(\underline{x}) \rightarrow y$ — we pick xs that give some info. on what we're trying to predict

- One idea is SIMILARITY
  - ↳ If x_1, x_2 in the same class, then similar values

HANDWRITTEN DIGITS

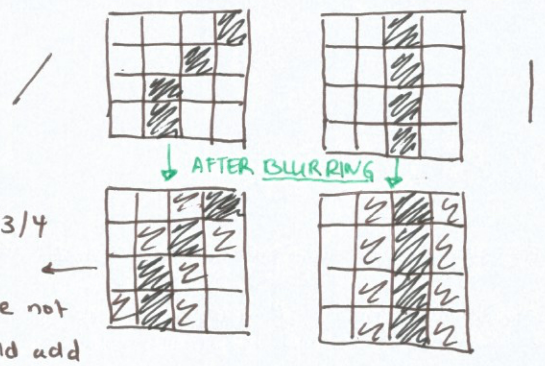
- Challenges: varying style, slant, pressure, pen type, etc.
- One way is to represent each pixel as a separate attribute
  - i.e. 20 x 20 bitmap → 400 attributes $X = x_1, x_2, x_3 \ldots x_{400}$
    - Each attribute is a real no. (degree of 'blackness' of a pixel)

↓ OR to save space...

- Each attribute is a binary (0, 1) — 0 (white) if $x_i < t$, else 1 (black)
  - — BUT accuracy will suffer

↓ So, we 'blur' the image

- Why does blurring work? (An example of handwriting '1')



Now this may look like a 2/3/4 but on avg., it's going to be more similar to 1s bc. we're not adding blur where a 2 would add a blur

↳ Similarity / overlap between the two bitmaps will be much higher

- Image pixels as attributes work bc. we can isolate each digit, rescale and de-slant

i.e. 20 x 20 bitmap

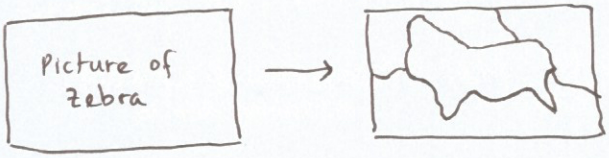
OBJECT RECOGNITION [Recognize objects in an image, i.e. faces]

→ Challenges: position in a photo, orientation, scale, lighting differences, obstructions

→ Using pixels as attributes will NOT work

- A single pixel no longer carries any identifying information

→ A way is to segment the image into 'regions' and compute features describing the region.



POSSIBLE ATTRIBUTES
- position (x, y)
- relative area
- circumference } tells you how circular objects are
- color frequencies
- texture filters

bc. they are invariant to irrelevant differences

→ Segmentation is inaccurate

- but we hope these errors are systematic (i.e. same for all zebras)

HOW TO REPRESENT TEXTS

→ i.e. Spam mails

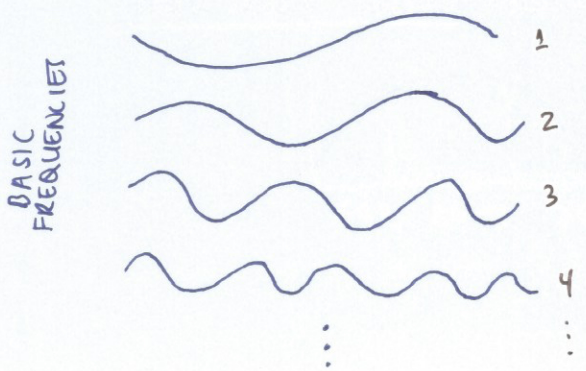
- We use words as numeric attributes
- One attribute for every possible word in the language (instead of every possible word in the email)
  - ↳ 1 if word was observed, 0 otherwise
  - ↳ Note that we'll have $10^5 - 10^6$ attributes but 99.99% will be zeros
- Robust; if we delete a word in the email, we'll only have to turn 1 to 0 and the rest is going to be the same.

HOW TO REPRESENT MUSIC

→ Fourier transform: decompose music into base frequencies f and find 'weight' of each f to get a signal similar to the time series input

→ X_f = weight of frequency f

↳ insensitive to volume, shift, etc.



ACCURACY & IMBALANCED CLASSES

→ i.e. Predict if scientific publication will lead to a nobel prize

Claim: have a classifier that will be at least 99.99% accurate

↳ Always predict 'no' → correct for almost all publications

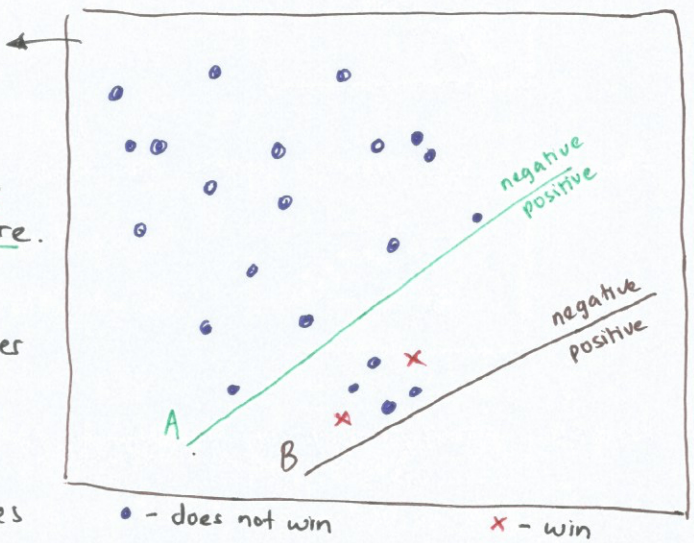
Accuracy is higher for B than A but it is better to have A as our classifier. Hence accuracy is a poor metric here.

↓

We want relative cost of false (+)ves or false (-)ves

~~ie cost (MIS)~~

Give more weight to the false (-)ves than to the false (+)ves.

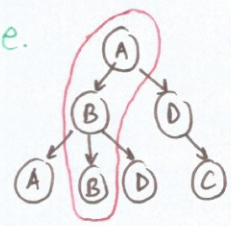


STRUCTURED OBJECTS

1) Structured Input

→ Embed in attributes

→ i.e.



Attributes = root-to-leaf path

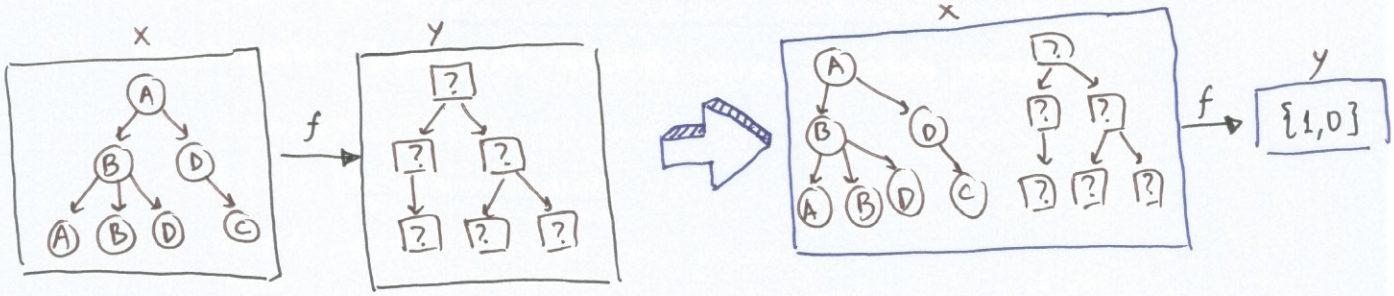
0	AAA
1	ABA
1	ABB
0	ABC
	⋮

2) Structured Output

→ Embed in input

→ Predict 1/0: Output does/doesn't go w/ input

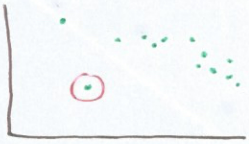
→ Search over possible outputs becomes main focus



DETECTING OUTLIERS

→ Detect through confidence interval (CI)

→ However, CI won't work for cases like:



→ Hence, always try to visualize data to detect irregularities

GENERATIVE vs. DISCRIMINATIVE CLASSIFIERS

- models the classes
- probabilistic 'model' for each class
- can use unlabeled data

- models the decision boundary
- may/may not be probabilistic
- can't use unlabeled data