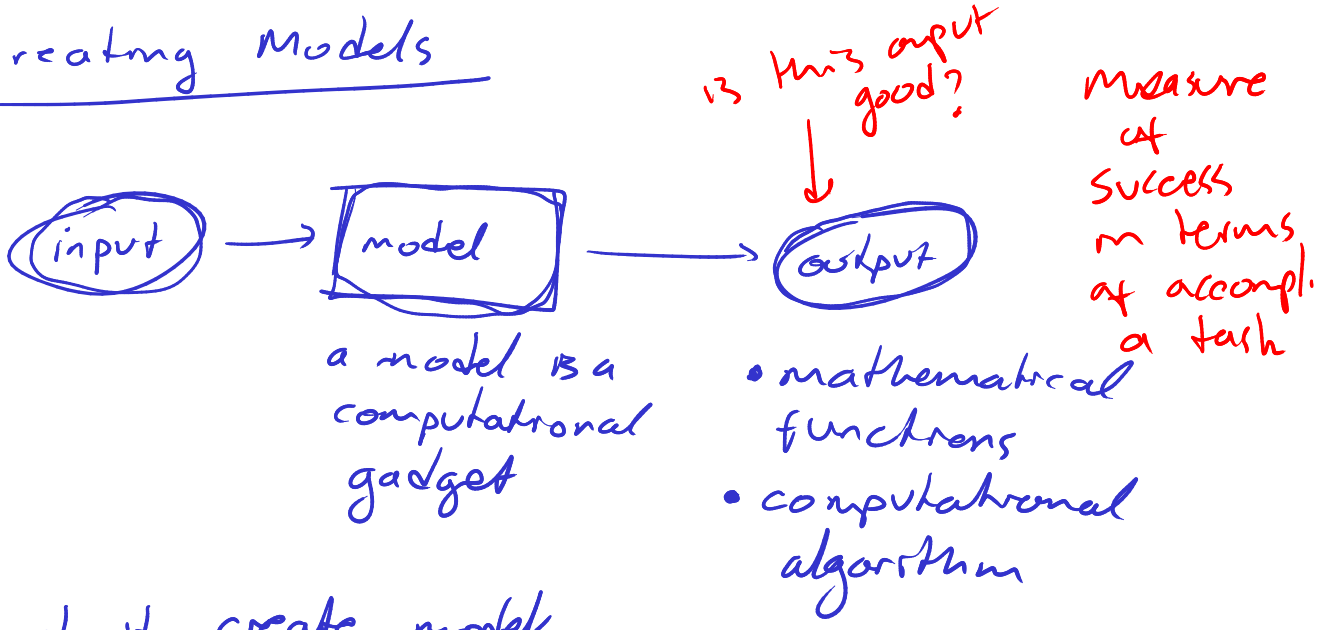
MAT 555

Review

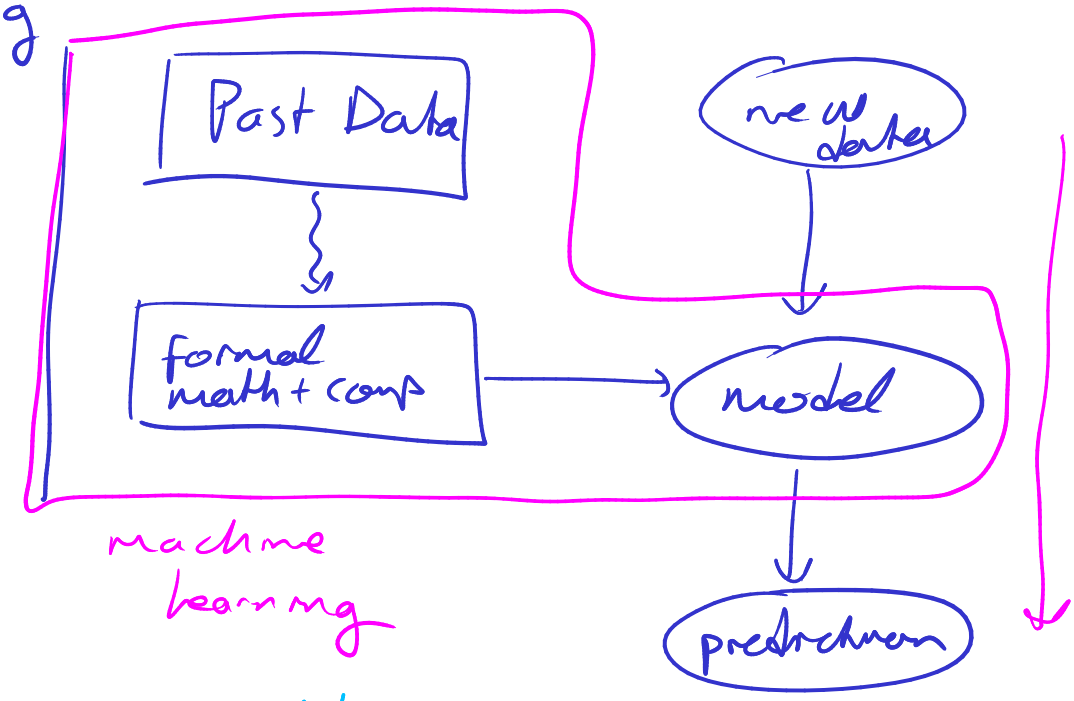
Creating Models



a model is a computational gadget

- mathematical functions
- computational algorithm

We don't create models out of nothing



Machine learning models have to be evaluated in order to be useful

CROSS-VALIDATION is the collection of practices that help you evaluate a given model

Two Main Types of ML

Supervised (Output-Driven)

Data consists of (input, output) pairs

Since we have outputs

The most natural way of measuring a model is to measure the diff. btw output & prediction

we statistically measure how much of a difference "in the large" exists btw prediction & output.

Unsupervised (Task-Driven)

we only have data no "expected output"

Our data is coll. for a specific purpose and we have a task to accomplish

we also have a measure of "success" for this task.

to what degree ~~how~~ does this output help for us to accomplish our task

Cross-validation

Kmeans vs KNN

unsupervised supervised

Data $\subseteq \mathbb{R}^n$ in KNN $c : \text{Data} \longrightarrow \{1, \dots, l\}$

pt in $\mathbb{R}^n$ c is given a finite set of labels

$$\text{Data} = \{ (x, \underbrace{c(x)}_{\text{label}}) \mid x \in \text{Data} \}$$

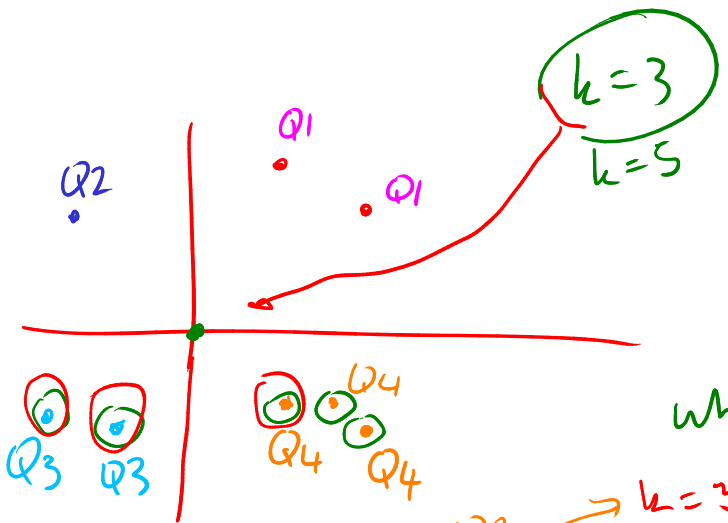
Create a computational gadget so that if we have a new pt $x \in \mathbb{R}^n$ make a prediction $c(x)$ based on the Data and $c : \text{Data} \longrightarrow \{1, \dots, l\}$ we have.

In KNN we take a vote

- x new pt $z_1, \dots, z_k$
- Find k-closest pts in Data (k = odd number)
- Take a vote in z's and the highest number of labels wins

$c(z_1), c(z_2), \dots, c(z_k)$

Ex



Label set

Q1, Q2, Q3, Q4

we have 8 pts in $\mathbb{R}^2$

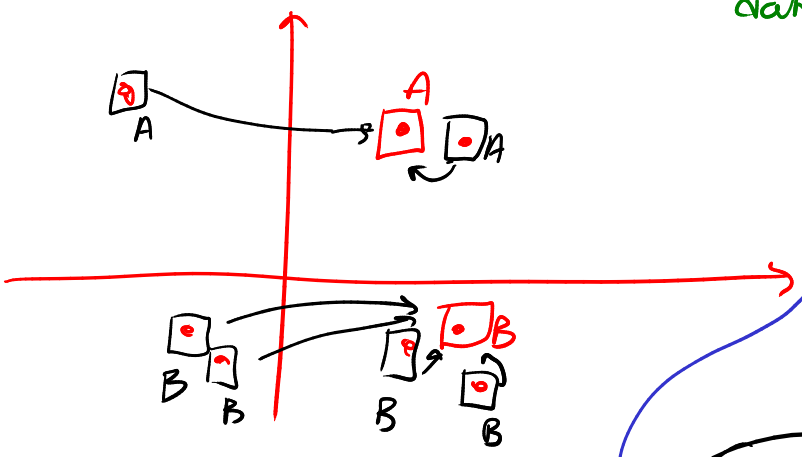
what is the Label of (Q0)?

$Q3 \to k=3$ Q3 Q3 Q4

$Q4 \to k=5$ Q3 Q3 Q4 Q4 Q4

For K=5 Q4 wins!

In the K-means algorithm, we don't have outputs! (in the data)



Task: Split this data into disjoint pieces.

We usually determine at the beginning the number of pieces.

k=2 → two disjoint subsets.

The red pts are chosen as the center of each group randomly then pts are assigned to a group depending on the distances

Next Find the center of group A group B and repeat the process until group centers stabilize!

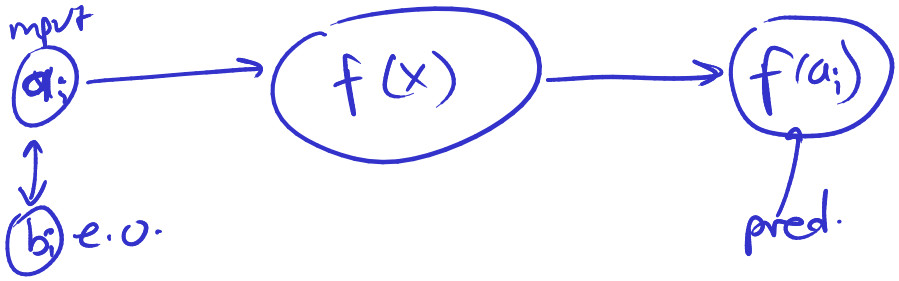
In Supervised learning

(input, expected output)

model gives you (input, prediction pairs)

If we have a [similarity / distance] measure

then we can evaluate the model in terms of success.



$(a_1, b_1) \ldots (a_N, b_N)$ Data

$$Sim(D) = \frac{1}{N} \sum_{i=1}^{N} Sim(b_i, f(a_i))$$ average sim!

$$Cost(D) = \frac{1}{N} \sum_{i=1}^{N} Dist(b_i, f(a_i))$$ Average cost.

exact, pred

Success of f can be measured using $Sim(D)$ $Cost(D)$

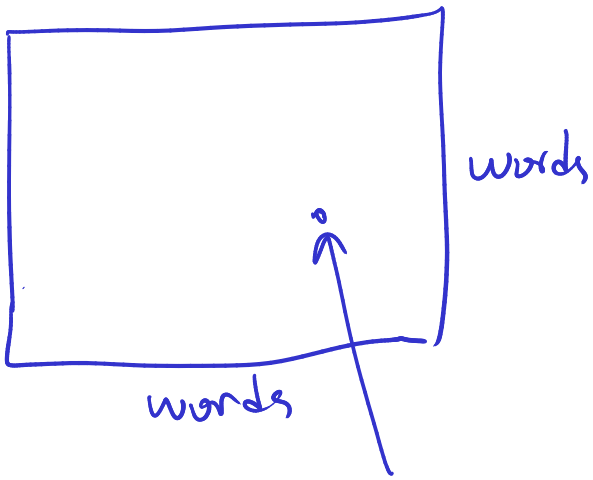
Idea / Problem / Task

Text ⇝ Create a meaningful useful dataset in $\mathbb{R}^n$

Word Embedding

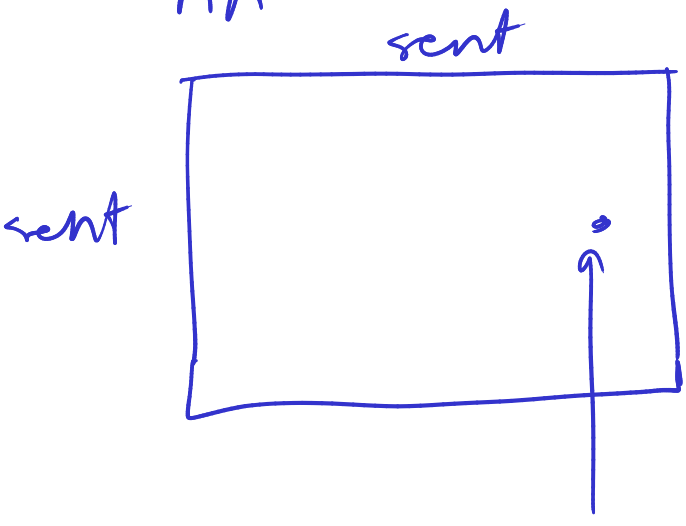
matrix

$A^t A$



# of times each pair of words appear in the same sentence

AA^t



# of words in common.

$A^t A$ and AA^t give us similarity measure

how similar words are

how similar sentences are

em: Words $\longrightarrow$ $\mathbb{R}^n$

Regression Models

Data $\subseteq \mathbb{R}^n$ $\quad$ (expected outputs) $\subseteq \mathbb{R}$

$(x^{(i)}, y^{(i)})$ $\quad$ $i = 1, \ldots, N$ ← # of data pts

$x^{(i)} \in \mathbb{R}^n$ $\quad$ $y^{(i)} \in \mathbb{R}$

Regression model

$$y^{(i)} \approx \beta \cdot x^{(i)} + \alpha$$

inner product

$\beta \in \mathbb{R}^n \qquad \alpha \in \mathbb{R}$

Find best fitting "linear" model

β, α unknown.

Abs. Error$_i$ $= |\, y^{(i)} - \beta \cdot x^{(i)} - \alpha \,|$

$y^{(i)}$: exp. out

$\beta \cdot x^{(i)} - \alpha$: pred

Total Abs Error $= \sum_{i=1}^{N} |\, y^{(i)} - \beta \cdot x^{(i)} - \alpha \,| = \text{Cost}(\beta, \alpha)$

Trick: instead of $\mathbb{R}^n$ we use $\mathbb{P}^n \subseteq \mathbb{R}^{n+1}$ proj

$x^{(i)} \rightsquigarrow (x^{(i)}; 1) = z^{(i)}$

model $y^{(i)} = \beta' \cdot z^{(i)}$

$\beta' = (\beta; \alpha)$ vector

TAE $= \sum_i \| y^{(i)} - \beta' \cdot z^{(i)} \| = \text{Cost}(\beta')$

$$\sum_{i=1} \left(y^{(i)} - \beta' \cdot z^{(i)} \right)^2 = \text{Cost}(\beta')$$

OLS

REGRESSION

known

unkn.

known

optimum wrt β'

MAIN IDEA

You make a choice $\longrightarrow$ $f(-, \theta)$

I prefer these models with an unknown θ as my parameter

Then depending on the context (unsupervised vs supervised)

$$\text{Cost}(\theta) = \underbrace{\text{Fit}(\theta)}_{\text{supervised}} + \underbrace{\gamma(\theta)}_{\text{sometimes we learn}} \sum_{x_i \in D} \text{Dist}\big(\underbrace{y_i}_{\text{exp}}, f(\underbrace{x_i}_{\text{pred}}, \theta)\big)$$

sometimes we learn add penalty to simplify $f(-, \theta)$

$\text{Cost}(\theta)$ depends on the task in the unsupervised case

Next Step Optimize $\text{Cost}(\theta)$, $\text{Fit}(\theta)$

Most popular choice for optimization

GRADIENT DESCENT

$$\theta_{n+1} = \theta_n + \lambda \nabla \text{Cost}(\theta_n)$$

θ_n's are approximations

$\nabla \text{Cost}(\theta)$ is the gradient

λ = fixed "learning rate"

θ_∞ is the $\lim_{n \to \infty} \theta_n$

best fitting model

$f(-, \theta_\infty)$