CMU Portugal Academy

CMU Portugal
Advanced Training Program

Artificial Intelligence, Machine Learning and Data Science Bootcamp

Chrysoula Zerva

NOVA SCHOOL OF SCIENCE AND TECHNOLOGY

Carnegie Mellon Portugal · TÉCNICO LISBOA · Ciências ULisboa · NOVA SCHOOL OF SCIENCE & TECHNOLOGY

email: jmag@fct.unl.pt

Chrysoula – Short bio

chrysoula.zerva@tecnico.ulisboa.pt

Assistant Professor, Instituto Superior Tecnico

Associated Researcher, Instituto de Telecomunicações

Member of ELLIS and LUMLIS

PhD, University of Manchester, 2019

EPSRC Doctoral Prize Fellowship in 2019

My research is focused on machine learning (ML) and more specifically natural language processing (NLP), with emphasis on multilinguality and multimodality.

I am particularly keen on topics of uncertainty, explainability, fairness and quality estimation.

Currently I am involved in the Centre for Responsible AI project and the UTTER project

Sweta – Short bio

swetaagrawal20@gmail.com

Postdoctoral Researcher, Instituto de Telecomunicações

PhD, University of Maryland, 2023

My research focuses on developing adaptable, controllable, multilingual, and efficient natural language generation models that can incorporate contextual information.

Also interested in contributing new automatic and human evaluation methods to assess whether the AI generated text fulfills a desideratum or appropriately handles contextual information.

Module Topics

1. Regression vs classification;
2. Data preparation, validation and evaluation;
3. Supervised vs unsupervised models;
4. Neural Networks;
5. Reinforcement learning;
6. Advanced topics; and
7. Your own projects!!

Visit to download materials.

Lecture 01 - topics

1. Introduction to Machine Learning

1. Project structure presentation

1. Math Recap

1. Linear regression

1. Linear classifiers

1. Practical: Building a regressor from scratch

slido

Join at https://app.sli.do/event/sqmxHcQvQQ85Wo9kbo49PY

01

What is Machine Learning

What is machine learning?

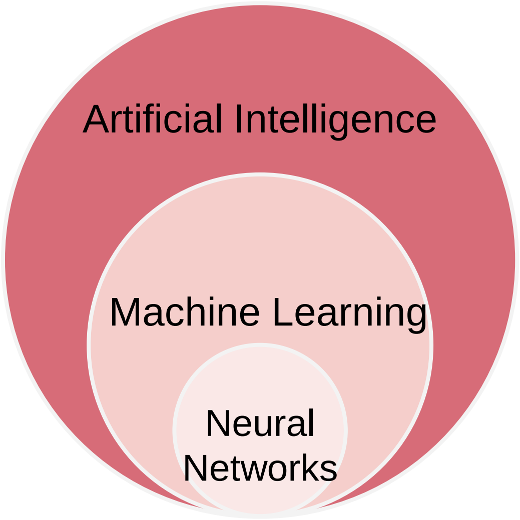
slido

What is machine learning?

Machine Learning

- The study of algorithms and processes that allow us to learn patterns from data in order to make predictions (*inference*)

- An area of Artificial Intelligence (AI)
  - Contrast to rule-based decision making



Machine Learning

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience.”

(Tom Mitchell, 1998)

What do we want to learn?

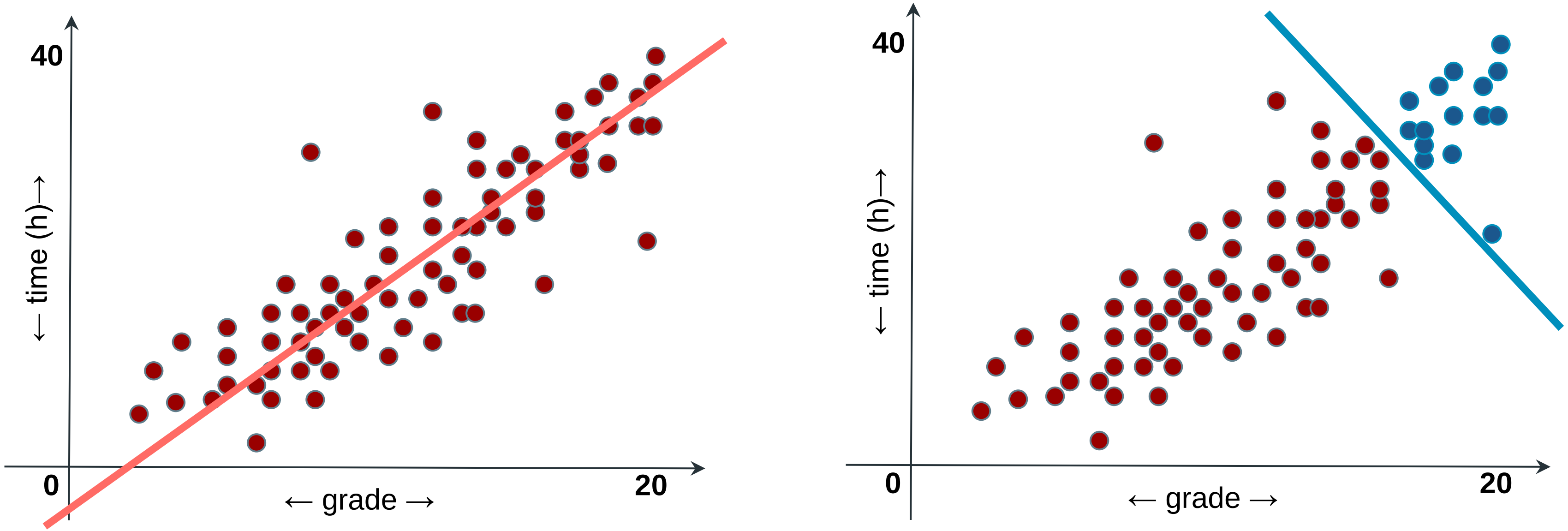
slido

What do we want to learn?

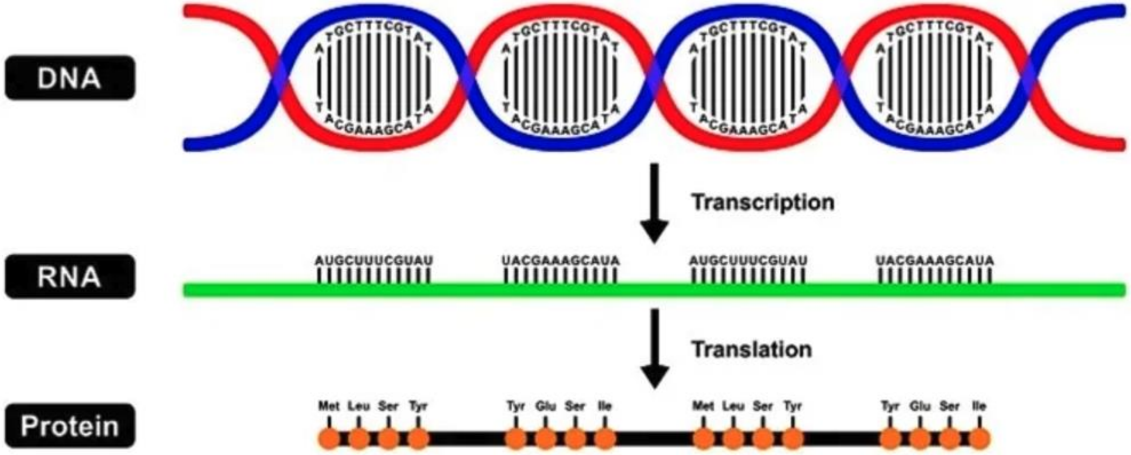
Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

Regression vs classification

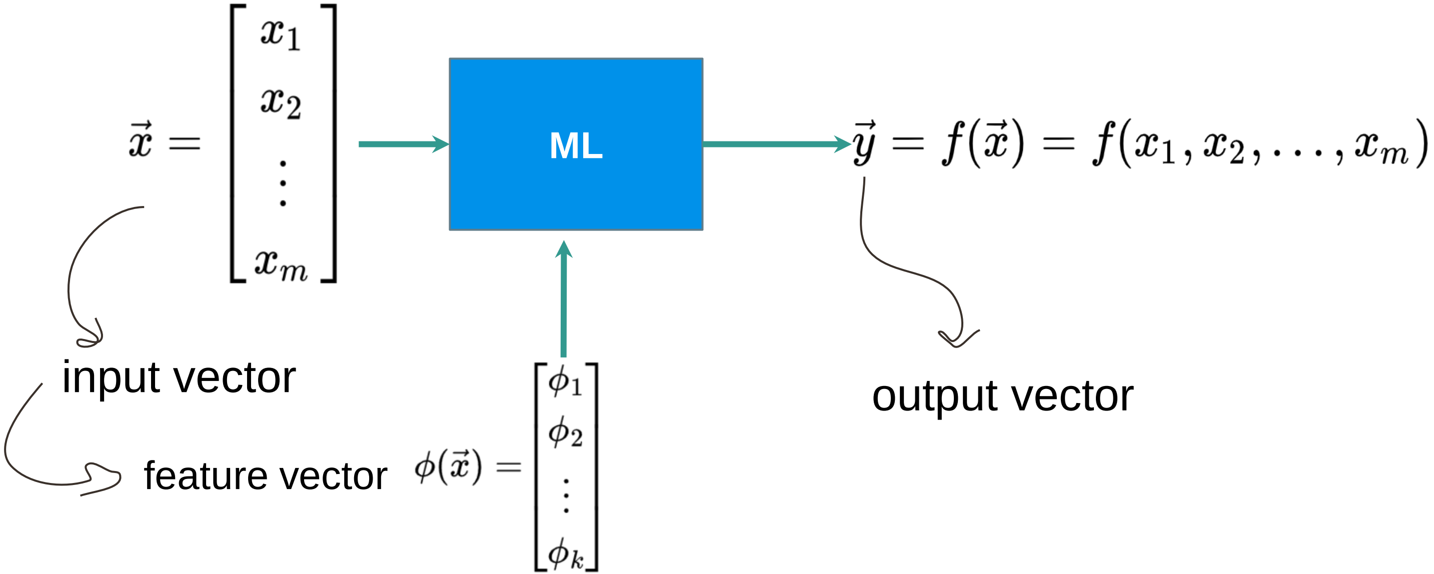
Linear and binary cases



Structured prediction tasks



So what do we want to learn?



$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

ML

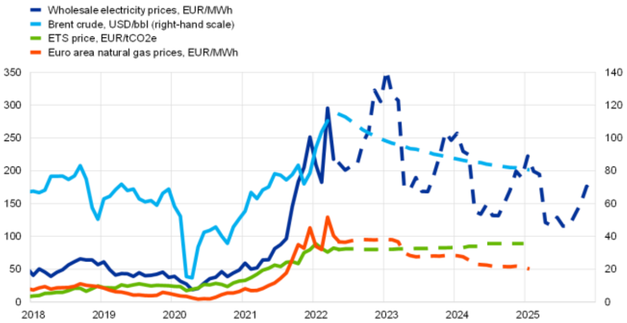
$$\vec{y} = f(\vec{x}) = f(x_1, x_2, \ldots, x_m)$$

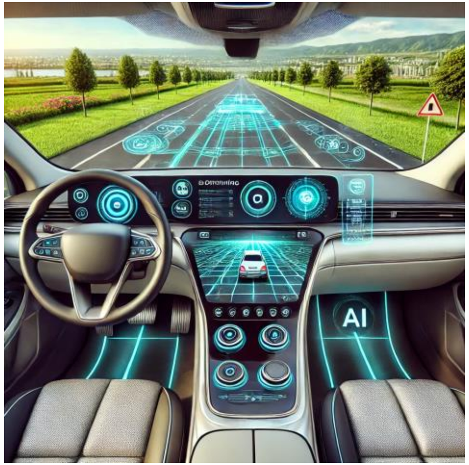
input vector

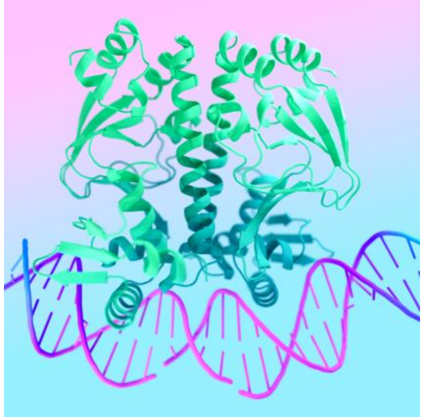
feature vector $\phi(\vec{x}) = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \end{bmatrix}$

output vector

Real use cases







Google DeepMind

02

Introduction to projects

Introduction to Projects

- Consumer Credit Risk Prediction
  - Goal classify credit applicants into two classes: the “good credit” class that is liable to reimburse the financial obligation and the “bad credit” class that should be denied credit due to the high probability of defaulting on the financial obligation.
- COVID-19 Prediction
  - Goal: predict the number of confirmed and death cases for the next ten days in different areas in the world.

Introduction to Projects

- Euro 2024 Final Prediction
  - Goal: predict match outcomes using past matches datasets.
- Bigmart Sales Prediction
  - Goal: The data scientists at BigMart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store.

Project Grading

- Data Preparation/Analysis - 10%
- Method - 30%
- Analysis on outcomes - 10%
- Report - 5%
- Code + Implementation - 5%
- Presentation - 40%

We will select top 3 projects.

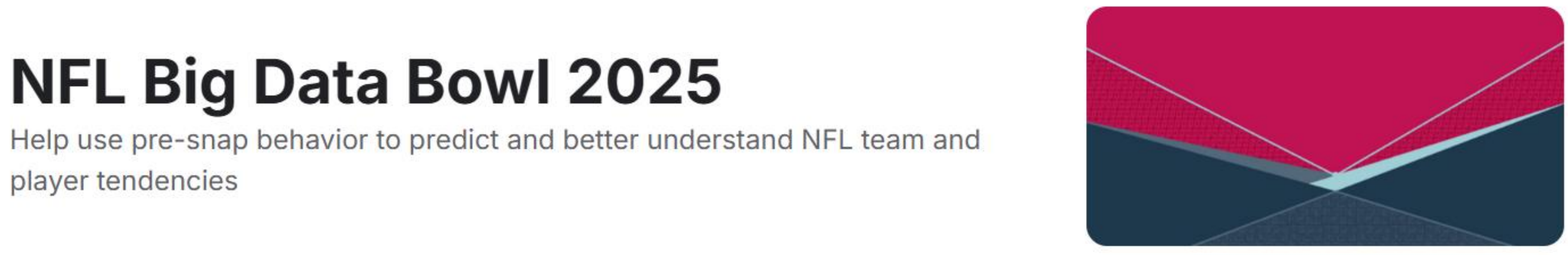
Ongoing Kaggle Competitions

Jane Street Real-Time Market Data Forecasting

Predict financial market responders using real-world data.

The competition dataset comprises a set of timeseries with 79 features and 9 responders, anonymized but representing real market data. The goal of the competition is to forecast one of these responders, i.e., responder_6, for up to six months in the future.

Ongoing Kaggle Competitions



NFL Big Data Bowl 2025

Help use pre-snap behavior to predict and better understand NFL team and player tendencies

The challenge is generating actionable, practical, and novel insights from player tracking data corresponding to pre-snap team and player tendencies. Examples include, but are not limited to:

- Play prediction (run v pass)
- Scheme prediction (blitzes, run fits, route combinations, etc)
- Player prediction (pass patterns, blocking assignments, etc)

Ongoing Kaggle Competitions

CZII - CryoET Object Identification

Find small biological structures in large 3D volumes

This competition challenges you to create ML algorithms that automatically annotate five classes of protein complexes within a curated real-world cryoET dataset.

Ongoing Kaggle Competitions

Child Mind Institute — Problematic Internet Use

Relating Physical Activity to Problematic Internet Use

This competition challenges you to develop a predictive model capable of analyzing children's physical activity data to detect early indicators of problematic internet and technology use.

Ongoing Kaggle Competitions

Google - Unlock Global Communication with Gemma

Create Gemma model variants for a specific language or unique cultural aspect

Check out more at

https://www.kaggle.com/competitions

03

Math and Notation Recap

Recap of useful notions: Matrices and Vectors

$$A \in \mathbb{R}^{m \times n}$$

$$A = \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{bmatrix}$$

Transpose: A^T such that $(A^T)_{i,j} = A_{j,i}$

Matrix multiplication:

$C = AB \in \mathbb{R}^{m \times p}$, where $C_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}$

↪ Associative but **not** commutative!

$$x \in \mathbb{R}^n$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Inner product:

$$\langle x, y \rangle = x^T y = y^T x = \sum_{i=1}^{n} x_i y_i$$

Outer product:

$$x\, y^T \in \mathbb{R}^{n \times m}, \quad \text{where } (x\, y^T)_{i,j} = x_i\, y_j$$

Recap of useful notions: Matrices II

Identity Matrix:

$I \in \mathbb{R}^{n\times n}$ is a square matrix such that

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Neutral element of matrix product: *AI* = *IA* =*A*

Recap of useful notions: Matrices III

Inverse:

Matrix $A \in \mathbb{R}^{n\times n}$ is invertible if there is $A^{-1} \in \mathbb{R}^{n\times n}$ s.t. $A^{-1}A = A\,A^{-1} = I$

Matrix A $\in \mathbb{R}^{n\times n}$ is invertible ⇔ det(A) ≠ 0.

$$\det(A^{-1}) = \frac{1}{\det(A)}$$

Solving system $Ax = b$, if A is invertible: $x = A^{-1}b$

$$(A^{-1})^{-1} = A, \quad (A^{-1})^{T} = (A^{T})^{-1}, \quad (A\,B)^{-1} = B^{-1}A^{-1}$$

Recap of useful notions: Probabilities

Classical definition $\mathbb{P}(A) = \frac{N_A}{N}$

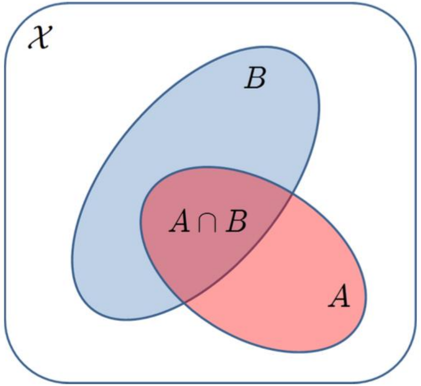
N mutually exclusive equally likely outcomes, N_A of which result in the occurrence of event A.

Frequentist definition $\mathbb{P}(A) = \lim_{N \to \infty} \frac{N_A}{N}$

relative frequency of occurrence of A in infinite number of trials

Recap of useful notions: Conditional Probabilities

If $\mathbb{P}(B) > 0$, $\qquad \mathbb{P}(A|B) = \dfrac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$

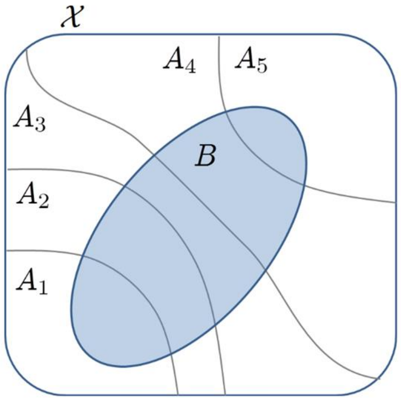


$$(A \perp\!\!\!\perp B) \;\Leftrightarrow\; \mathbb{P}(A \cap B) = \mathbb{P}(A)\,\mathbb{P}(B)$$

$$A \perp\!\!\!\perp B \;\Leftrightarrow\; \mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(A)\,\mathbb{P}(B)}{\mathbb{P}(B)} = \mathbb{P}(A)$$

Recap of useful notions: Conditional Probabilities

$$\mathbb{P}(B) = \sum_i \mathbb{P}(B|A_i)\mathbb{P}(A_i)$$
$$= \sum_i \mathbb{P}(B \cap A_i)$$



Bayes Theorem: $$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(B \cap A_i)}{\mathbb{P}(B)}$$

04

Linear Regression

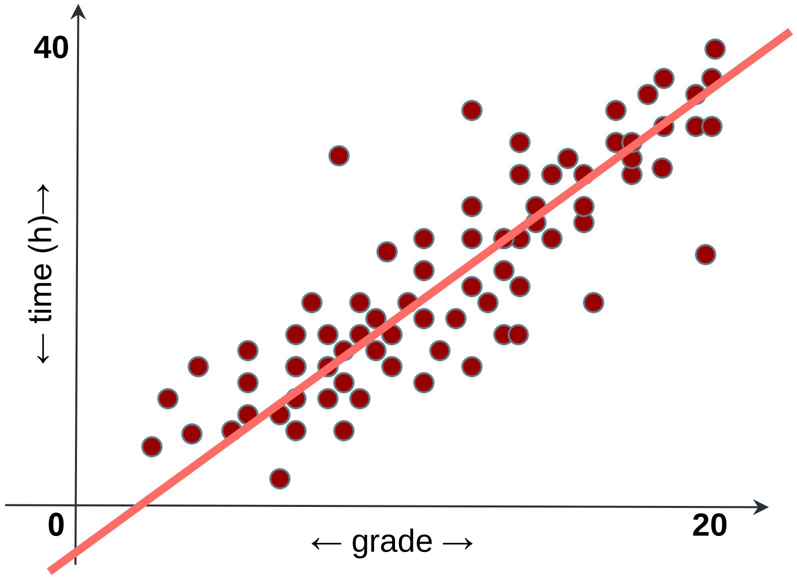
Linear regression

Model:

$$\hat{y} = w\,x + b$$

For training data:

$$\mathcal{D}\{(x_n, y_n)\}_{n=1}^{N}$$



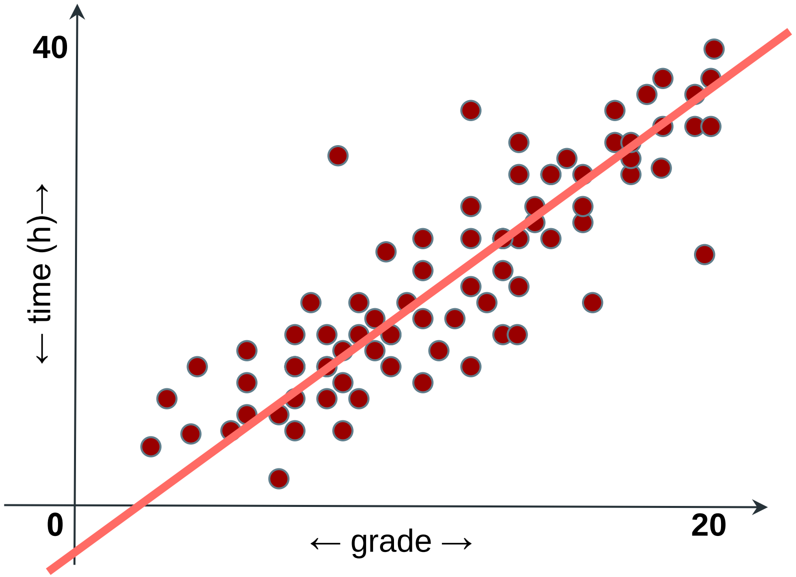
Recap of linear regression

Model:

$$\hat{y} = \overset{\text{slope}}{w}\, x + \underset{\text{intercept}}{b}$$

For training data:

$$\mathcal{D}\{(x_n, y_n)\}_{n=1}^{N}$$

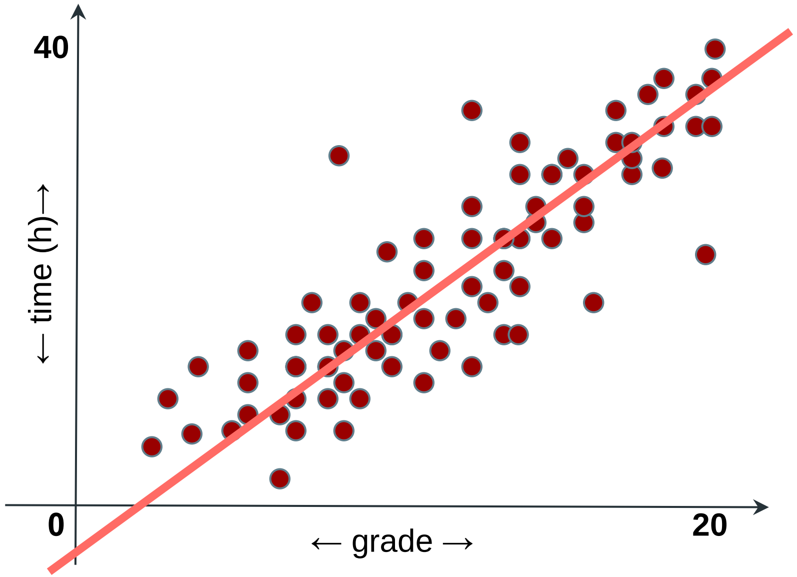


Recap of linear regression

How do we find the optimal line?

➔ Minimise the squared error:

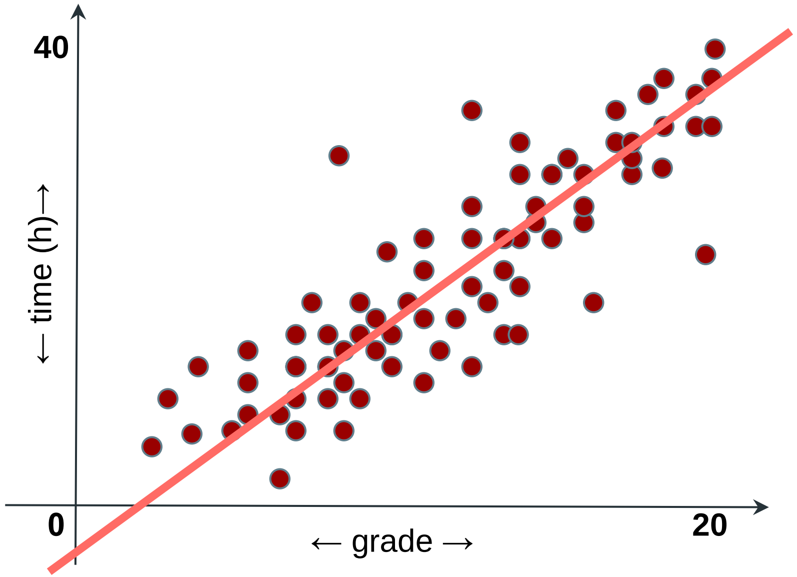
$$\min_{w,\, b} \sum_{n=1}^{N} \left(y_n - \underbrace{(w\, x_n + b)}_{\hat{y}_n}\right)^2$$



Recap of linear regression

✓ We have a closed-form solution:

$$\hat{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$



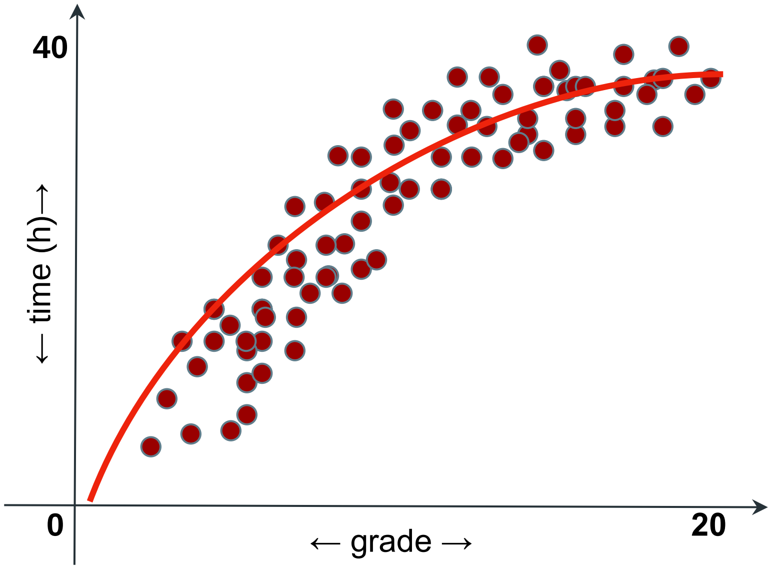
Linear regression for non-linear data

Can we transform our input?

$$\varphi(x) = [\ 1,\ x,\ x^2\ ]$$

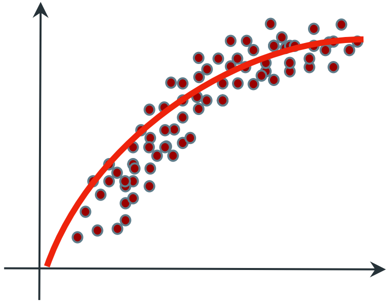
$$\boldsymbol{X} = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_N \end{bmatrix}$$

$$\boxed{\hat{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}}$$



$\varphi(x) = [\ 1,\ x,\ x^2\ ]$

Is this model linear?



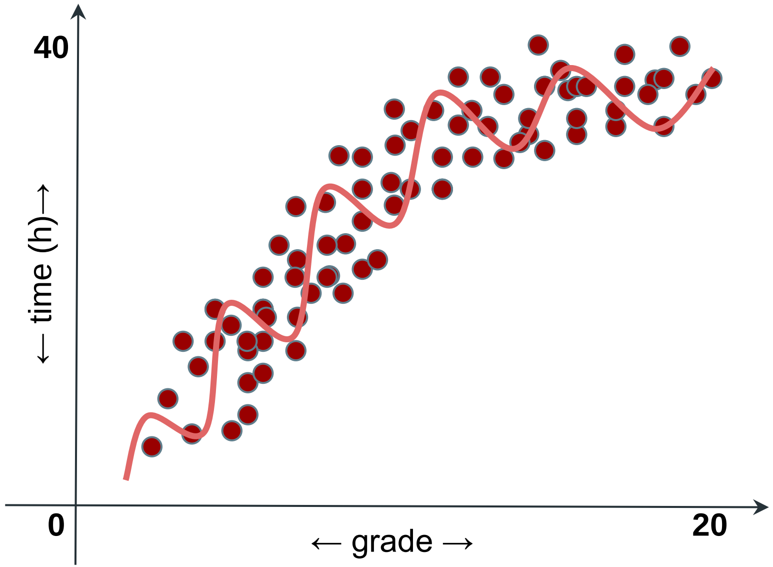
slido

Linear regression for non-linear data

Can we transform our input?

$$\varphi(x) = [\ 1, x, x^2, x^3, \ldots, x^K]$$

What if we overfit?



Fitting a linear model

- We want to **minimize** the least squares criterion $$\min_{w,\, b} \sum_{n=1}^{N} \Big(y_n - \underbrace{(w\, x_n + b)}_{\hat{y}_n}\Big)^2$$

- Corresponds to minimizing the squared **loss function**

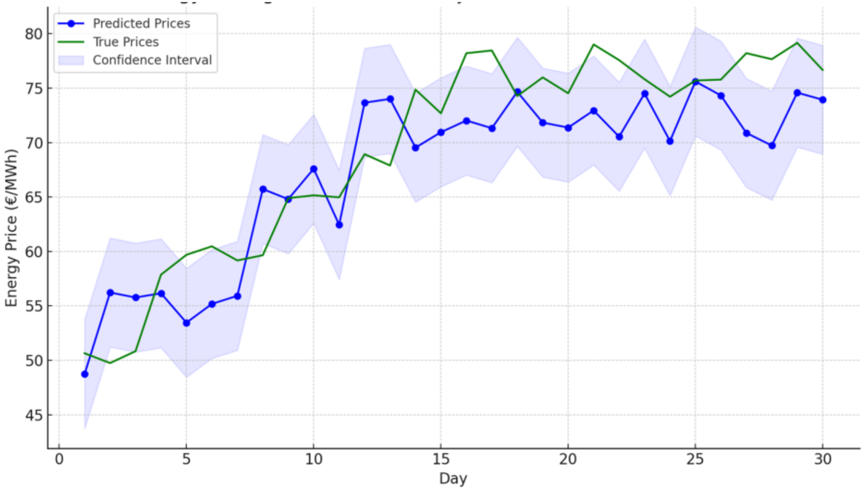
$$L(y, \widehat{y}) = \frac{1}{2}(y - \widehat{y})^2, \quad \text{where } \widehat{y} = w^T \phi(x)$$

$$\hat{w} = \arg\min_{w} \sum_{n=1}^{N} L(y_n - \hat{y}_n) = \arg\min_{w} \frac{1}{2} \sum_{n=1}^{N} \big(y_n - w^T \phi(x)\big)^2$$

Multiple (linear) regression

Usually we want to consider **multiple** (independent) variables:

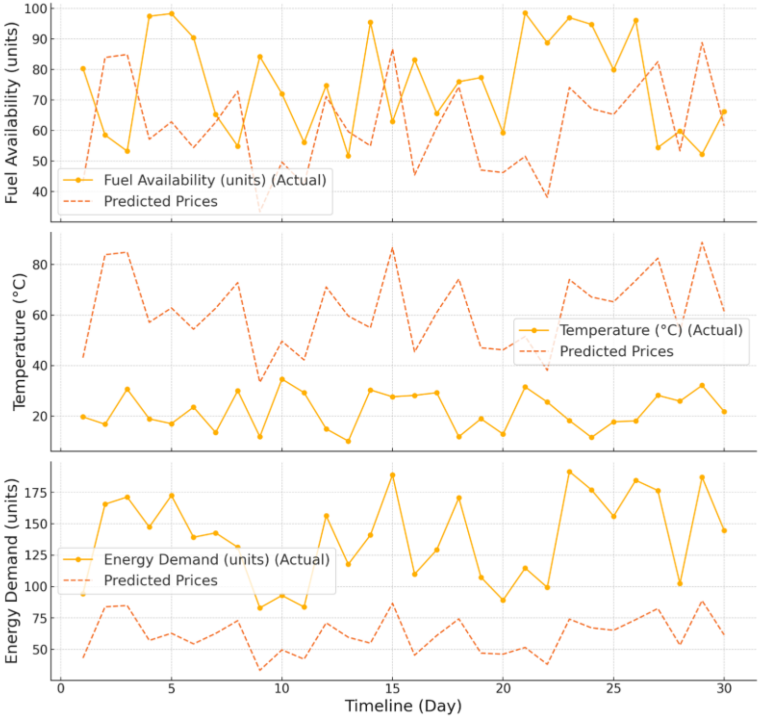
Short term energy pricing forecast



Multiple (linear) regression

Usually we want to consider **multiple** (independent) variables:

x_1: fuel availability
x_2: temperature
x_3: demand

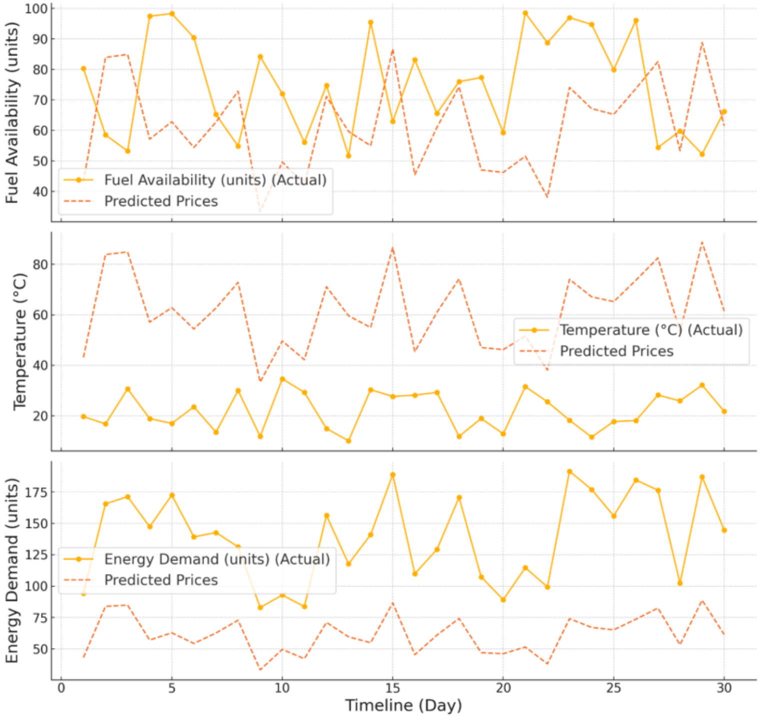


Multiple (linear) regression

Usually we want to consider **multiple** (independent) variables:

x_1: fuel availability
x_2: temperature
x_3: demand

$$\hat{y} = w_1\, x_1 + w_2\, x_2 + w_3\, x_3 + b$$



Multiple (linear) regression

Usually we want to consider **multiple** (independent) variables:

x_1: fuel availability
x_2: temperature
x_3: demand

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Nm} \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix}^T \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_N \end{bmatrix}$$

$$\hat{y} = w_1\, x_1 + w_2\, x_2 + w_3\, x_3 + b$$

$$\hat{y} = w_1\, x_1 + w_2\, x_2 + w_3\, x_3 + w_0$$

✓ The closed form solution still holds

$$\hat{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Multivariate Multiple regression

What if we also have multiple variables in the output?

Multivariate Multiple regression

What if we also have multiple variables in the output?

Assume k dependent variables —>
we can consider a matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times k}$

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nk} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nk} \end{bmatrix}$$

✓ The closed form solution still holds

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$$

✓ The analytical solution minimising the squared error loss holds as well!

How do we minimize a loss function?

$$\hat{w} = \arg\min_{w} \sum_{n=1}^{N} L(y_n - \hat{y}_n) = \arg\min_{w} \frac{1}{2} \sum_{n=1}^{N} (y_n - w^T \phi(x))^2$$

- ➔ Can we always find a global minimum?

$$\text{for any } x \in \mathbb{R}^n,\ f(x^*) \le f(x)$$

- ➔ Does a local minimum suffice?

$$\text{for any } \|x - x^*\| \le \delta \Rightarrow f(x^*) \le f(x)$$

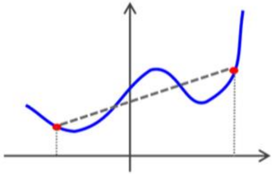
Minimization – convexity

f is a convex function, if, for any $\lambda \in [0, 1]$, and any x, x',

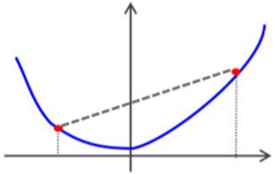
$$f\left(\lambda x + (1-\lambda)x'\right) \leq \lambda f(x) + (1-\lambda) f(x')$$

f is a strictly convex function, if, for any $\lambda \in]0, 1[$, and any x, x',

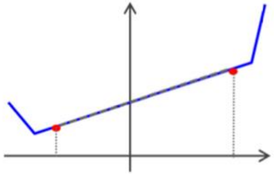
$$f\left(\lambda x + (1-\lambda)x'\right) < \lambda f(x) + (1-\lambda) f(x')$$



non-convex



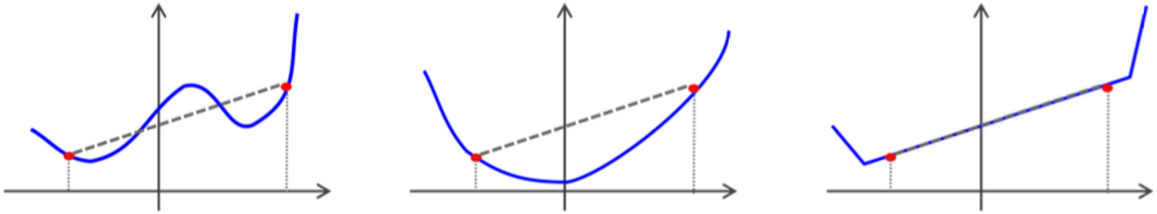
convex
strictly convex



convex, not strictly

Minimization – convexity

Assume we want to find ***w*** to minimize function *f*



non-convex

convex

strictly convex

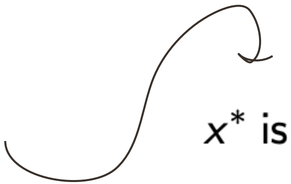
convex, not strictly

- ➔ If *f* is convex and *x** is a local minimizer, then it is also a global minimizer.
- ➔ If *f* is strictly convex and *x** is a local minimizer, then it is also the unique global minimizer.

Minimization

If *f* is differentiable then we can compute the gradient at *x*:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n$$



$$x^* \text{ is local minimizer } \substack{\Rightarrow \\ \not\Leftarrow} \nabla f(x^*) = 0$$

Gradient Descent

For differentiable functions:

We can take small steps in the negative gradient direction until we meet some stopping criterion

$$x^{(t+1)} \leftarrow x^{(t)} - \eta_{(t)} \nabla f(x^{(t)})$$

Learning rate

stepsize

Criteria:

- Number of epochs
- Avg error

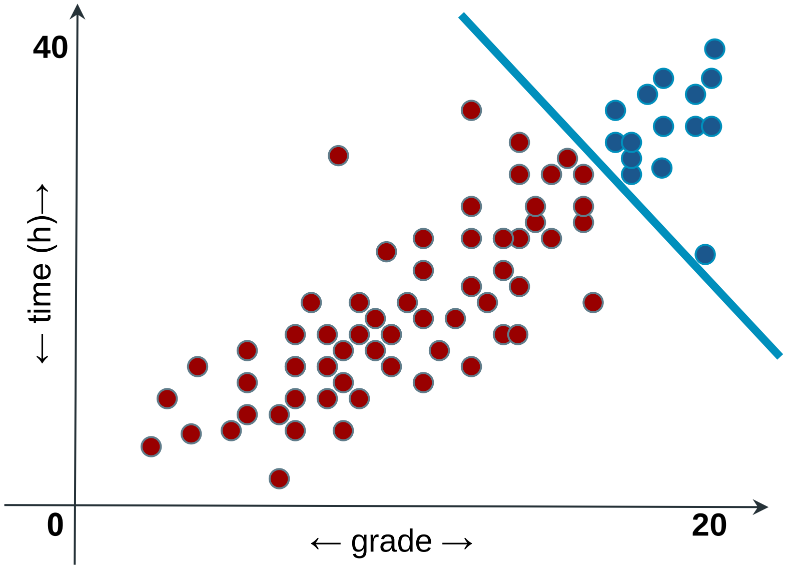
05

Linear Classification

Linear Classification

- ➔ (almost) linearly separable data
  - ◆ For some feature vector $\varphi(x)$
- ➔ We want to predict the correct label y^* over a set of possible labels (classes) $\mathcal{Y} = [y_1, y_2, \ldots, y_N]$
- ➔ We can use 1-hot encoding to represent each class

How do we predict the separation hyperplane?



Linear (binary) classifier

$$\hat{y} = \mathrm{sign}(w^T\phi(x) + b) = \begin{cases} +1, & \text{if } w^T\phi(x) + b \geq 0 \\ -1, & \text{if } w^T\phi(x) + b < 0. \end{cases}$$

move from continuous to discrete

score of the positive class

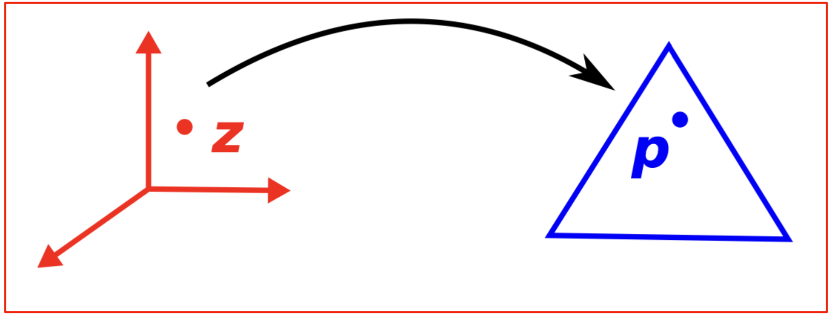
Classification decision (label)

The decision boundary is a hyperplane: $w^T\phi(x) + b = 0$

Logistic Regression - general case

➔ A linear model gives a score for each class $y:\ w_y^\top \phi(x)$

Should we just keep the class with the highest score?



Logistic Regression - general case

- ➔ A linear model gives a score for each class $y\colon\ w_y^\top \phi(x)$
- ➔ We can now compute the conditional posterior probability

$$P(y|x) = \underbrace{\frac{\exp\left(w_y^\top \phi(x)\right)}{Z_x}}_{\text{Softmax transformation}}, \qquad \text{where } Z_x = \sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^\top \phi(x)\right)$$

Softmax transformation

Logistic Regression – general case

- ➔ Is this still a linear model?

$$
\begin{aligned}
\arg\max_y \; P(y|x) &= \arg\max_y \; \frac{\exp\left(w_y^\top \phi(x)\right)}{Z_x} \\
&= \arg\max_y \; \exp\left(w_y^\top \phi(x)\right) \\
&= \arg\max_y \; w_y^\top \phi(x)
\end{aligned}
$$

Logistic Regression – Binary case

- ➔ $\mathcal{Y}$ = {-1,+1}:
- ➔ Adding a constant c to all scores will not affect the probabilities
- ➔ We set the score to 0 for the -1 class and to $w^T\varphi(x)$ for the +1 class

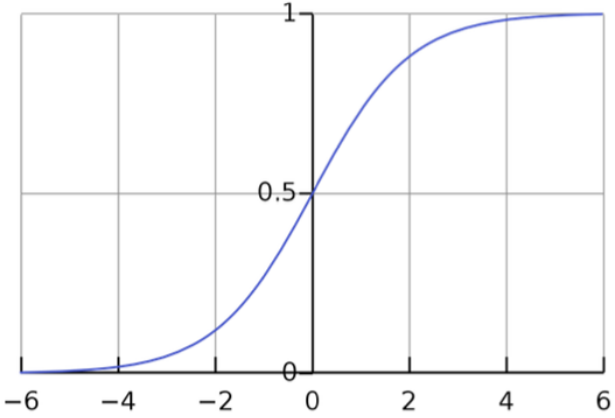
$$\begin{aligned} P(y=+1 \mid x) &= \frac{\exp\left(w^\top \phi(x)\right)}{1+\exp\left(w^\top \phi(x)\right)} \\ &= \frac{1}{1+\exp\left(-w^\top \phi(x)\right)} \\ &\equiv \sigma\left(w^\top \phi(x)\right). \end{aligned}$$

Sigmoid transformation

Sigmoid

- ✓ Widely used in neural networks.
- ✓ Maps R into [0, 1].
- ✓ The output can be interpreted as a probability.
- ✓ Positive, bounded, strictly increasing.

$$\boxed{\sigma(u) = \frac{e^u}{1 + e^u}}$$



Logistic Regression

What do we want to optimise?

- ➔ We want to choose **W** to maximise the probability of true labels
- ➔ Or maximise the conditional log likelihood of the training data
- ➔ Strictly convex
- ➔ No closed form solution

$$
\begin{aligned}
\widehat{\boldsymbol{W}} &= \arg\max_{\boldsymbol{W}} \log\left(\prod_{t=1}^{N} P_{\boldsymbol{W}}(y_t|x_t)\right) \\
&= \arg\min_{\boldsymbol{W}} -\sum_{t=1}^{N} \log P_{\boldsymbol{W}}(y_t|x_t) \\
&= \arg\min_{\boldsymbol{W}} \sum_{t=1}^{N} \left(\log \underbrace{\sum_{y'} \exp(w_{y'}^\top \phi(x_t))}_{Z_{x_t}} - w_{y_t}^\top \phi(x_t)\right)
\end{aligned}
$$

Logistic Regression - Gradient Descent

Cross entropy loss function:

$$L(\boldsymbol{W};(x,y)) = \log \sum_{y'} \exp\left(w_{y'}^{\top}\phi(x)\right) - w_y^{\top}\phi(x)$$

- ➔ We want to find the weights that minimise the loss: $\arg\min_{\boldsymbol{W}} \sum_{t=1}^{N} L(\boldsymbol{W};(x_t,y_t))$
- ➔ Initialise weights to 0
- ➔ Iterate until convergence
- ➔ Use suitable learning rate

$$\boxed{\begin{aligned} \boldsymbol{W}^{(k+1)} &= \boldsymbol{W}^{(k)} - \eta_k \nabla_{\boldsymbol{W}} \left(\sum_{t=1}^{N} L(\boldsymbol{W}^{(k)};(x_t,y_t)) \right) \\ &= \boldsymbol{W}^{(k)} - \eta_k \sum_{t=1}^{N} \nabla_{\boldsymbol{W}} L(\boldsymbol{W}^{(k)};(x_t,y_t)) \end{aligned}}$$

Batch GD

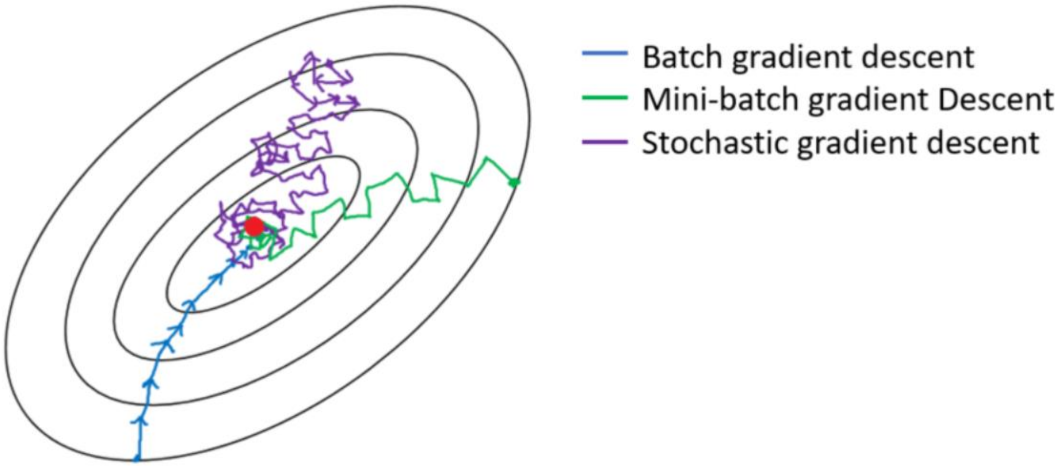
Logistic Regression - Stochastic Gradient Descent

- Iterate over single data samples and update
- Iterate until convergence
  - Or until reaching some stopping criterion
- Initialise weights to 0
- Pick (x_t,y_t) randomly

$$\boxed{\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \eta_k \nabla_{\boldsymbol{w}} L(\boldsymbol{w}^{(k)}; (x_t, y_t))}$$

- In between variant: mini-batch SGD

Convergence for GD variants



Gradient computation

Notes:

$$\nabla \log F(\boldsymbol{W}) = \frac{\nabla F(\boldsymbol{W})}{F(\boldsymbol{W})}$$

$$\nabla \exp F(\boldsymbol{W}) = \exp(F(\boldsymbol{W}))\nabla F(\boldsymbol{W})$$

1hot vector representation of class y

$$\boldsymbol{e}_y = [0, \ldots, 0, 1, 0, \ldots, 0]^\top.$$

y-th position

$$\begin{aligned}
\nabla L(\boldsymbol{W}; (x, y)) &= \nabla\left(\log \sum_{y'} \exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x)) - \boldsymbol{w}_y^\top \boldsymbol{\phi}(x)\right) \\
&= \nabla \log \sum_{y'} \exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x)) - \nabla \boldsymbol{w}_y^\top \boldsymbol{\phi}(x) \\
&= \frac{1}{\sum_{y'} \exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x))} \sum_{y'} \nabla \exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x)) - \boldsymbol{e}_y \boldsymbol{\phi}(x)^\top \\
&= \frac{1}{Z_x} \sum_{y'} \exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x)) \nabla \boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x) - \boldsymbol{e}_y \boldsymbol{\phi}(x)^\top \\
&= \sum_{y'} \frac{\exp(\boldsymbol{w}_{y'}^\top \boldsymbol{\phi}(x))}{Z_x} \boldsymbol{e}_{y'} \boldsymbol{\phi}(x)^\top - \boldsymbol{e}_y \boldsymbol{\phi}(x)^\top \\
&= \sum_{y'} P_{\boldsymbol{W}}(y'|x) \boldsymbol{e}_{y'} \boldsymbol{\phi}(x)^\top - \boldsymbol{e}_y \boldsymbol{\phi}(x)^\top \\
&= \left(\begin{bmatrix} P_{\boldsymbol{W}}(1|x) \\ \vdots \\ P_{\boldsymbol{W}}(|\mathcal{Y}| \mid x) \end{bmatrix} - \boldsymbol{e}_y\right) \boldsymbol{\phi}(x)^\top.
\end{aligned}$$

Link to Materials

Slides

Practicals

Project Form

Thank you!