INTRO TO MACHINE LEARNING (ML)

M. A. BENATIA,
MABENATIA@CESI.FR

DEC 2020

Outline for today

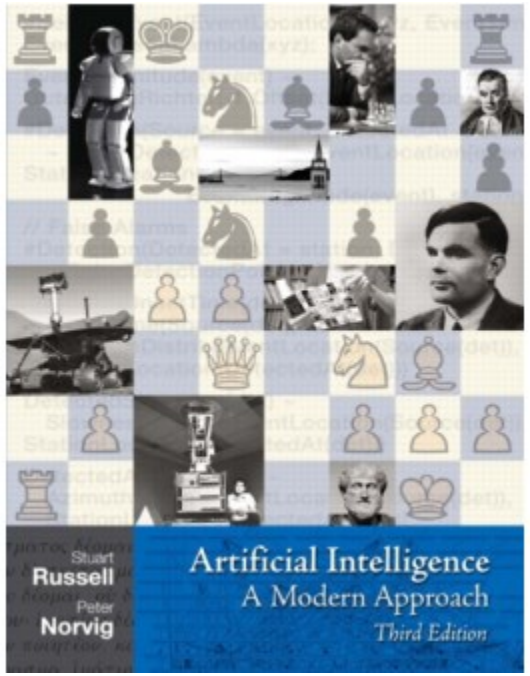
- ❑ Overview of the course
- ❑ Introduction to AI
- ❑ Data Mining processes
- ❑ ML in the context of AI
- ❑ ML methods for regression tasks

Prerequisites

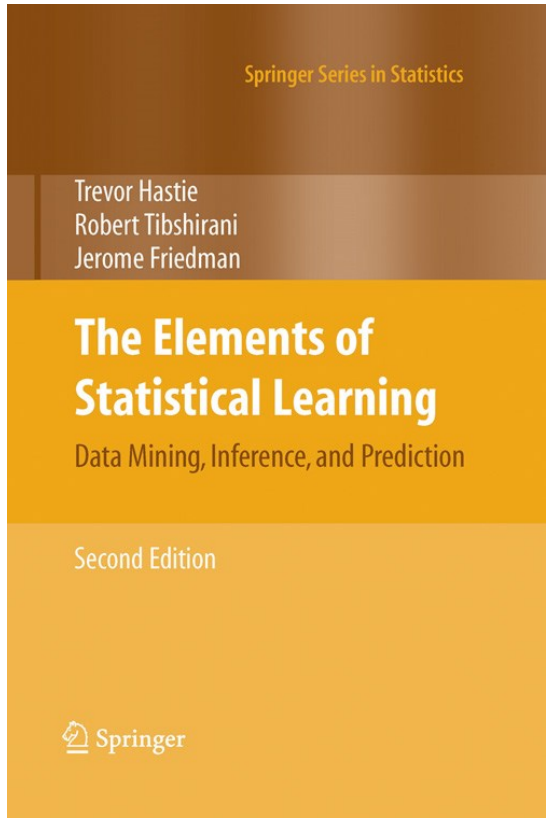
- ❑ Basic probability & Statistics
- ❑ Proofs
- ❑ Multivariate Calculus
- ❑ Linear Algebra

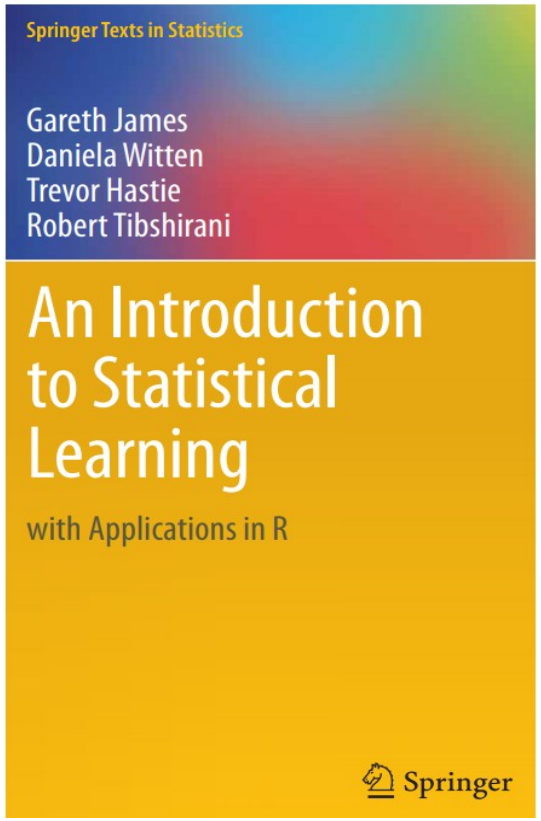
Textbook AI (optional)

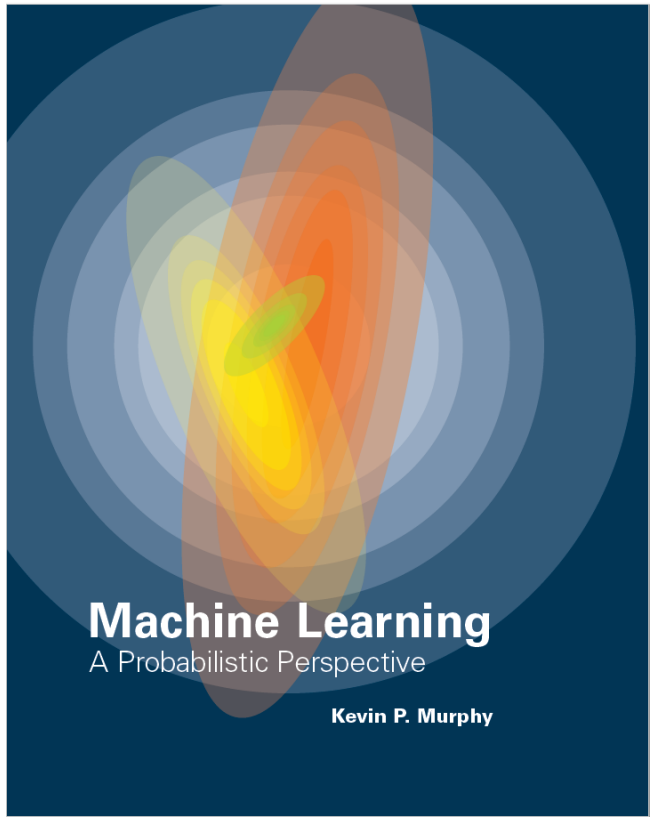
Russell & Norvig, AI: A Modern Approach, 3rd Ed.



Textbook ML (Mandatory)







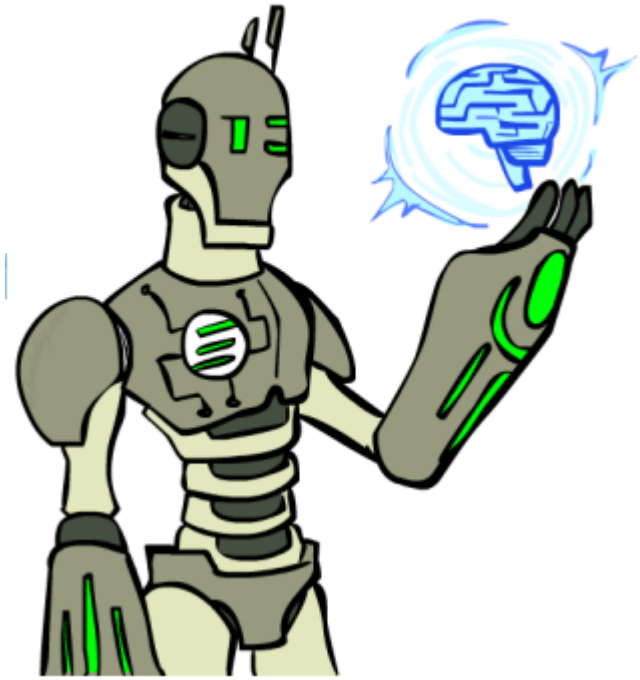
Part 0

Introduction

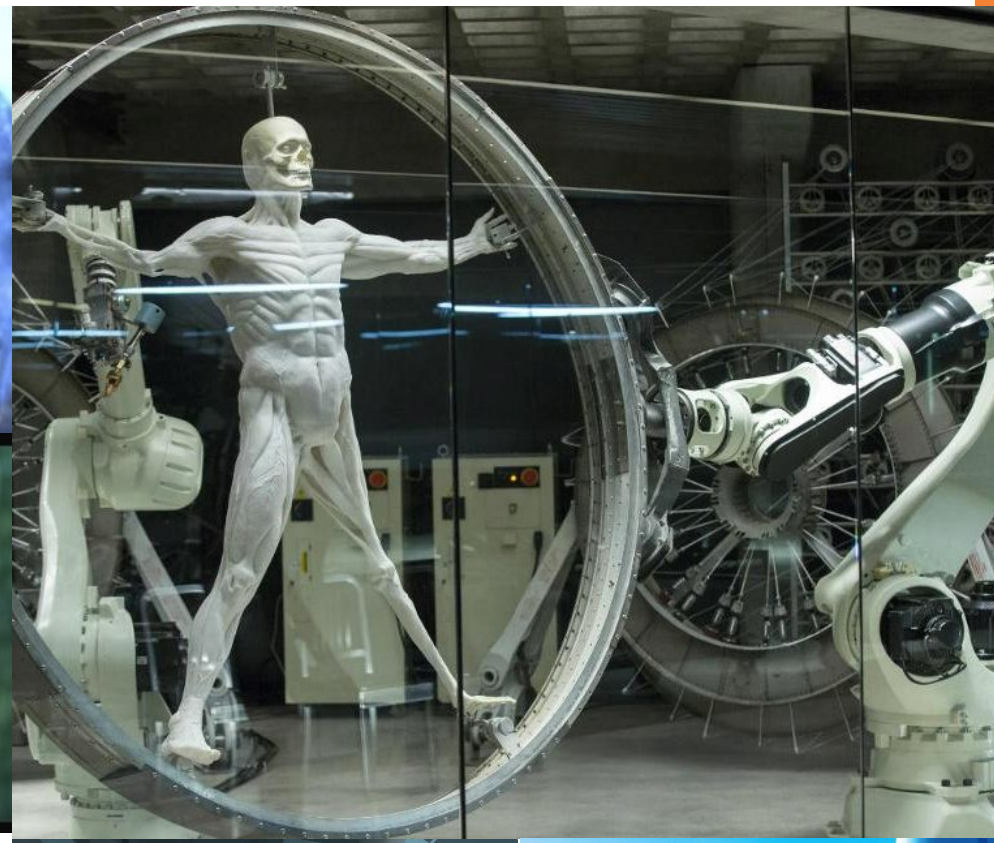
LINEACT cesi

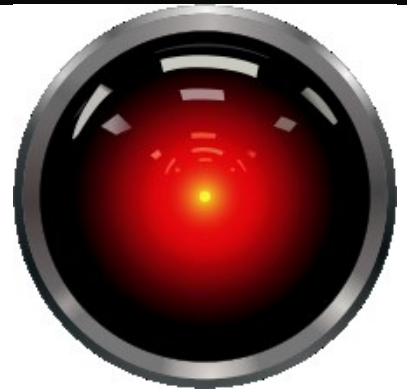
Introduction to Artificial Intelligence

- What is artificial intelligence?
- Past: how did the ideas in AI come about?
- Present: what is the state of the art?
- Future: will robots take over the world?



AI





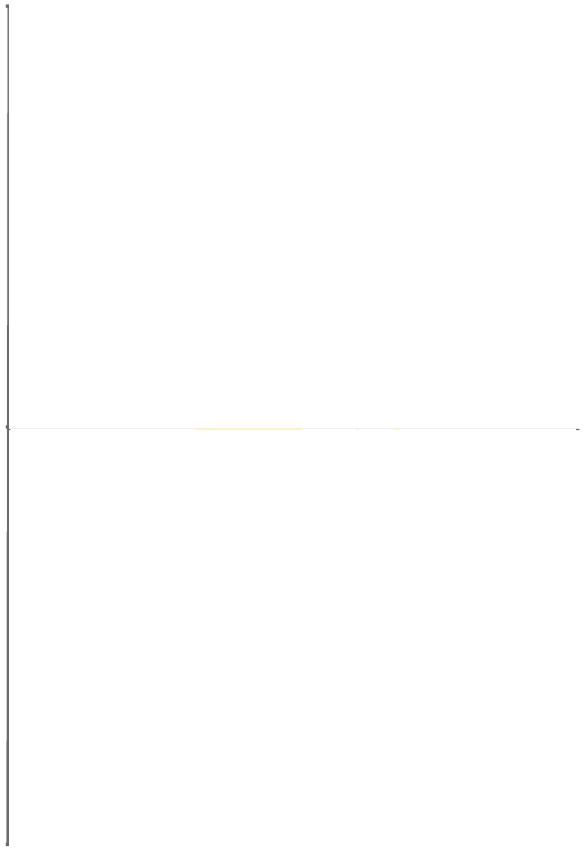

WAYMO


TUG
CAUTION
MAY CONTAIN
CHEMOTHERAPY DRUG
CAUTION
MAY CONTAIN
CHEMOTHERAPY DRUG

LINEAC

What is AI?

The science of making machines that:



Rational Decisions

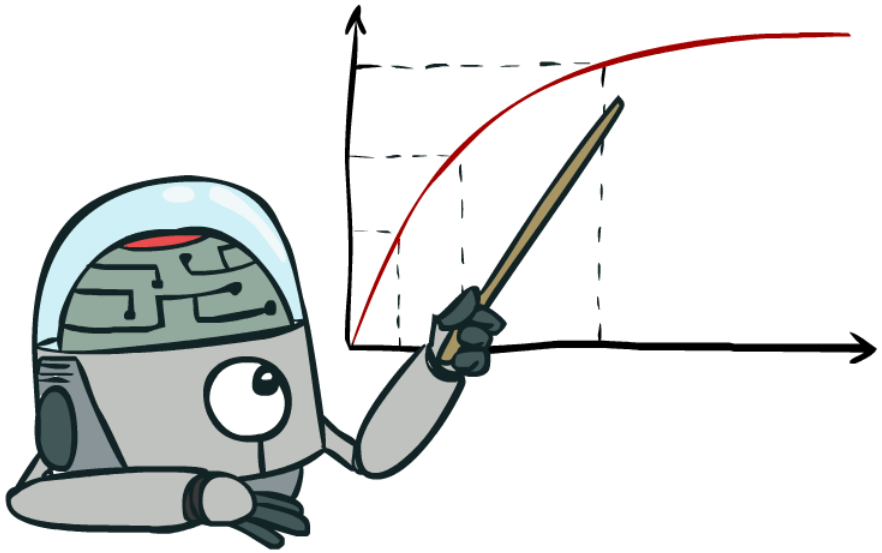
We'll use the term **rational** in a very specific, technical way:

- Rational: maximally achieving pre-defined goals
- Rationality only concerns what decisions are made (not the thought process behind them)
- Goals are expressed in terms of the **utility** of outcomes
- Being rational means **maximizing your expected utility**

A better title for this course would be:

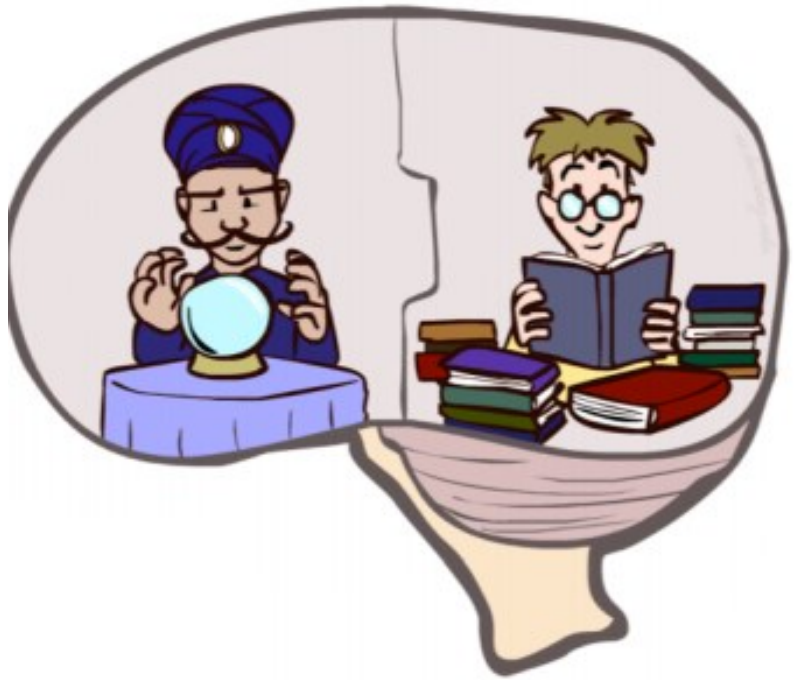
Computational Rationality

Maximize Your Expected Utility



What About the Brain?

- Brains (human minds) are very good at making rational decisions, but far from perfect; they result from accretion over evolutionary timescales
- We don't know how they work
- Lessons learned from human minds: memory, knowledge, feature learning, procedure formation, and simulation are key to decision making

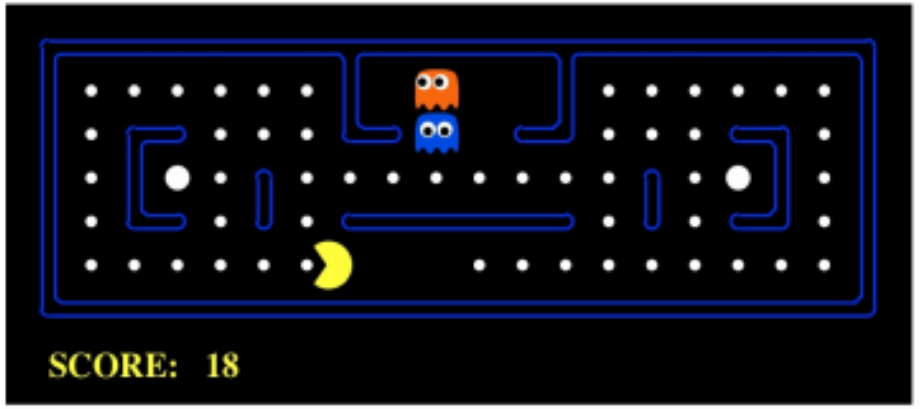


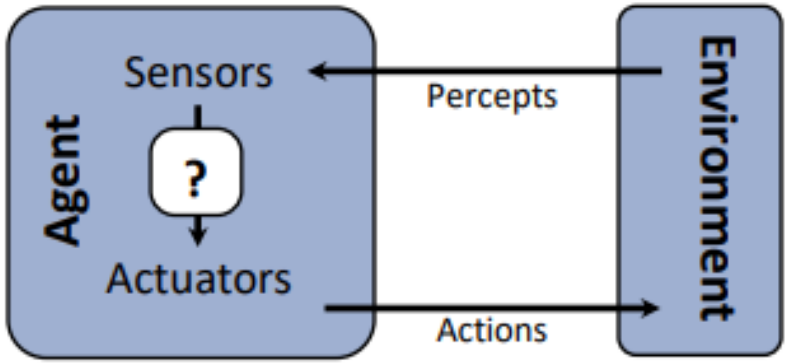
AI as computational rationality

- Humans are intelligent to the extent that our actions can be expected to achieve our objectives
- Machines are intelligent to the extent that their actions can be expected to achieve their objectives
  - Control theory: minimize cost function
  - Economics: maximize expected utility
  - Operations research: maximize sum of rewards
  - Statistics: minimize loss function
  - AI: all of the above, plus logically defined goals
- AI ≈ computational rational agents

Designing Rational Agents

- An agent is an entity that perceives and acts.
- A rational agent selects actions that maximize its (expected) utility.
- Characteristics of the percepts, environment, and action space dictate techniques for selecting rational actions





Pac-Man is a registered trademark of Namco-Bandai Games, used here for educational purposes

A (Short) History of AI

1940-1950: Early days

- 1943: McCulloch & Pitts: Boolean circuit model of brain
- 1950: Turing's “Computing Machinery and Intelligence”

1950—70: Excitement: Look, Ma, no hands!

- 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1956: Dartmouth meeting: “Artificial Intelligence” adopted
- 1965: Robinson's complete algorithm for logical reasoning

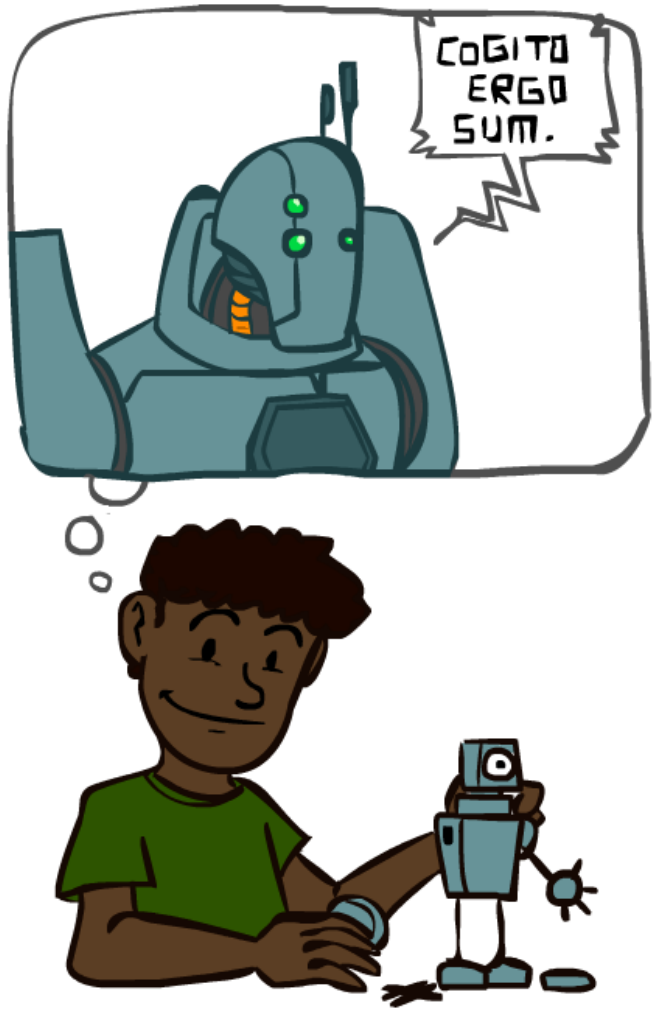
1970—90: Knowledge-based approaches

- 1969—79: Early development of knowledge-based systems
- 1980—88: Expert systems industry booms
- 1988—93: Expert systems industry busts: “AI Winter”

1990—: Statistical approaches

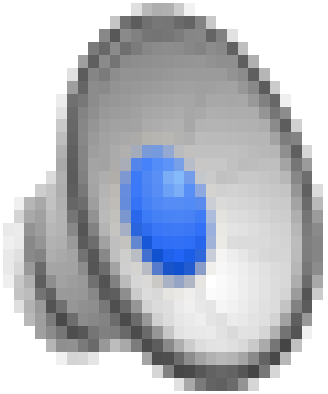
- Resurgence of probability, focus on uncertainty
- General increase in technical depth
- Agents and learning systems… “AI Spring”?

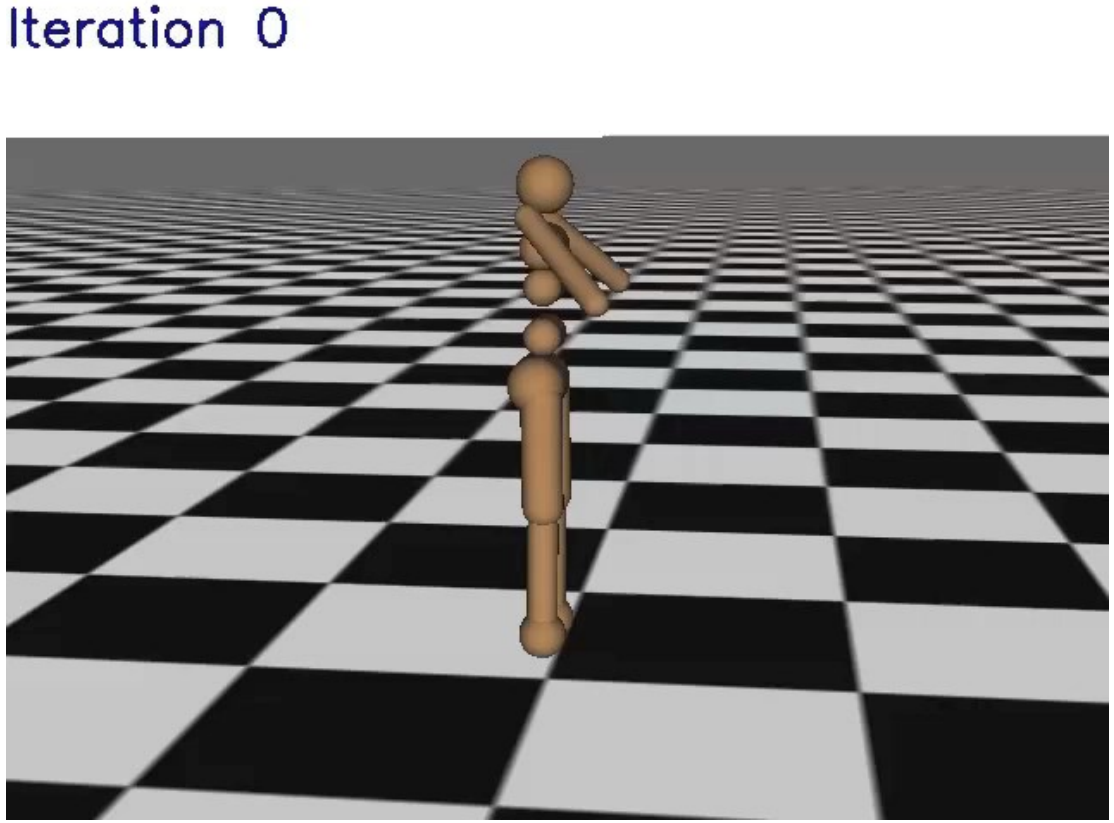
2000—: Where are we now?



LINEACT cesi

Nowadays AI





Part 1

Data Mining and Machine Learning

Definition of Data Mining

The nontrivial process of identifying **valid, novel, potentially useful, and ultimately understandable** patterns in data stored in structured databases.
- Fayyad et al (1996)

Keywords: Process, nontrivial, valid, novel, potentially useful, understandable

Other names: knowledge extraction, **pattern analysis**, knowledge discovery, information harvesting, **pattern searching**, …

Definition of Data Mining

Process implies that data mining comprises many **iterative** steps.

Nontrivial means that some **experimentation-type search or inference** is involved; it is not as straightforward as a computation of pre-defined quantities.

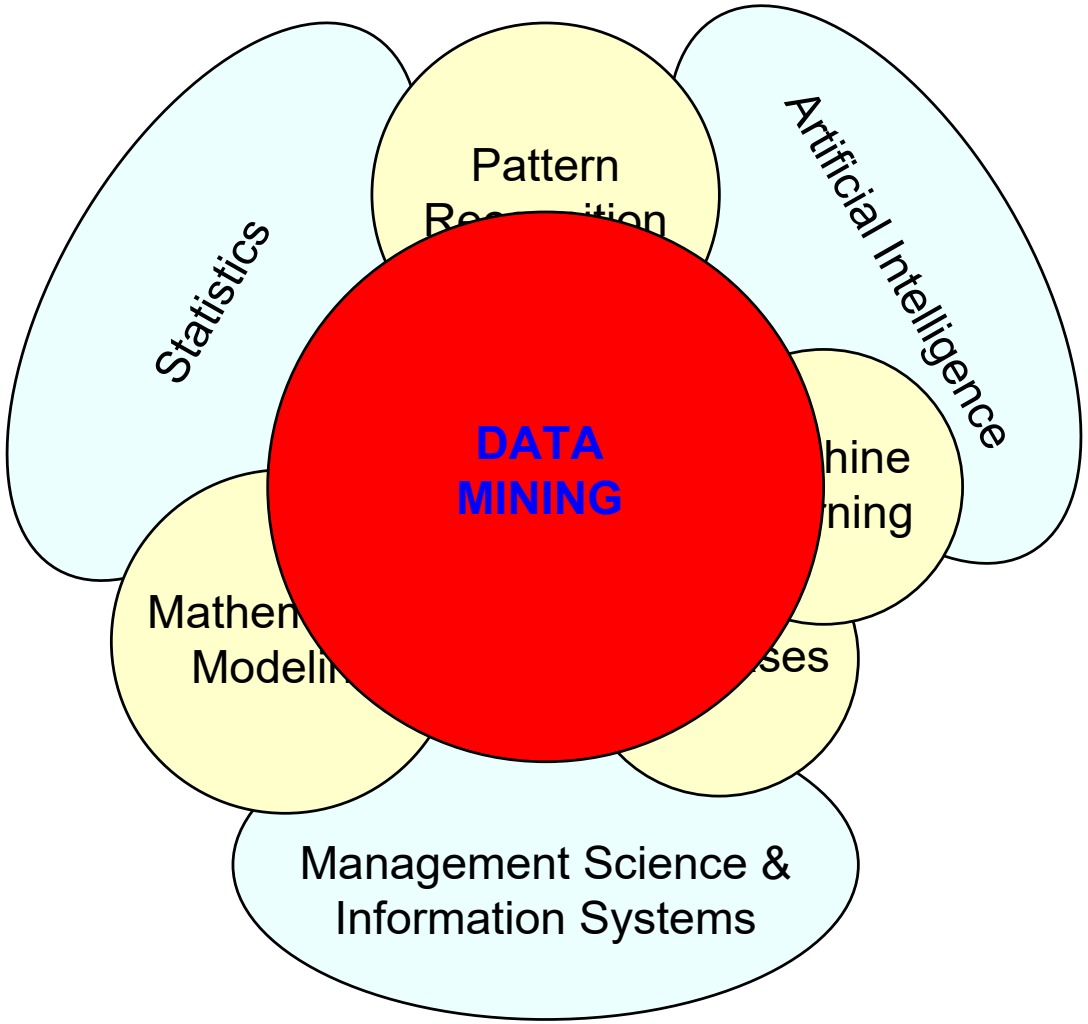
Valid means that the discovered patterns should hold true on new data with - sufficient degree of certainty.

Novel means that the patterns are not previously known to the user within the - context of the system being analyzed.

Potentially useful means that the discovered patterns should lead to some benefit to the user.

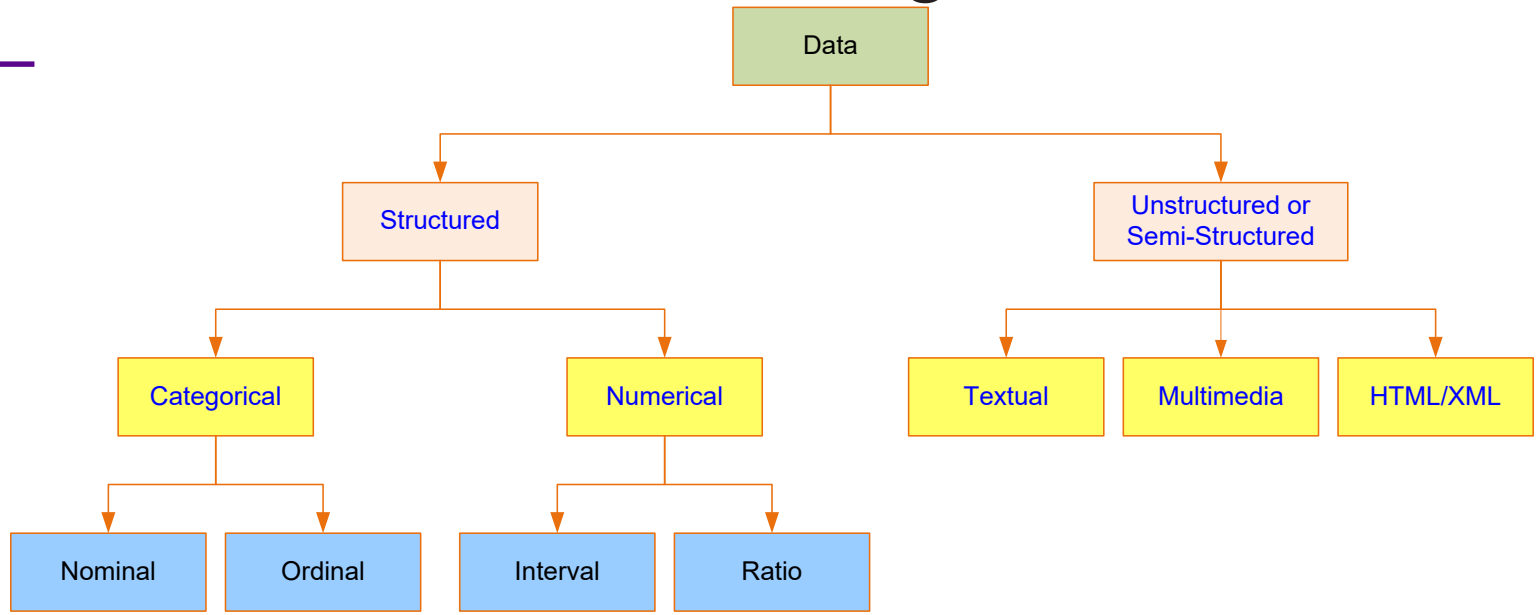
Ultimately **understandable** means that the pattern should make business sense.

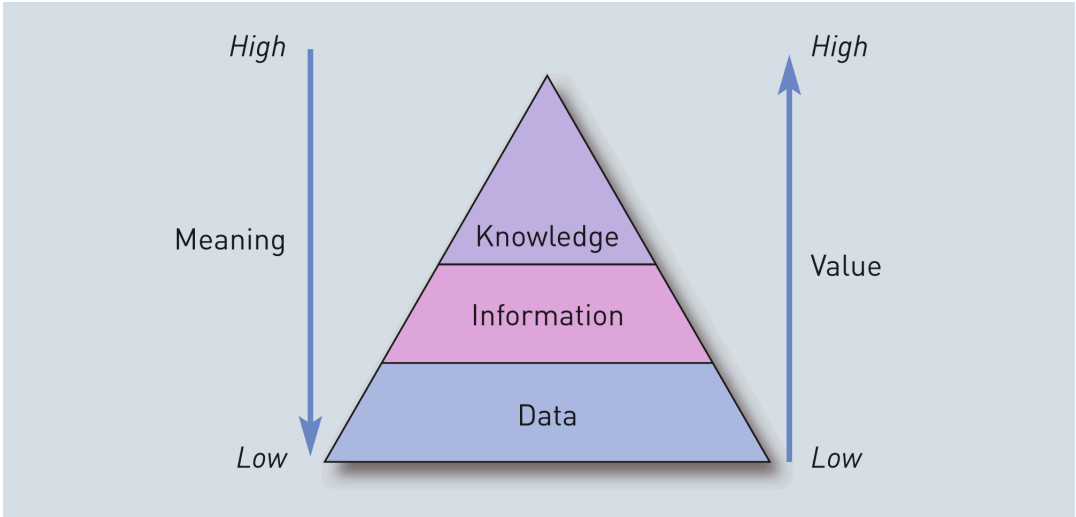
Data Mining is at the Intersection of Many Disciplines



LINEACT cesi

Data, Information and Knowledge





LINEACT cesi

What Does DM Do? How Does it Work?

DM extract patterns from data

- A mathematical (numeric and/or symbolic) relationship among data items

Types of patterns

- Association
- Prediction
- Cluster (segmentation)
- Sequential (or time series) relationships

Types of Patterns

Associations find the commonly co-occurring groupings of things, such as beer and diapers going together in market- basket analysis.

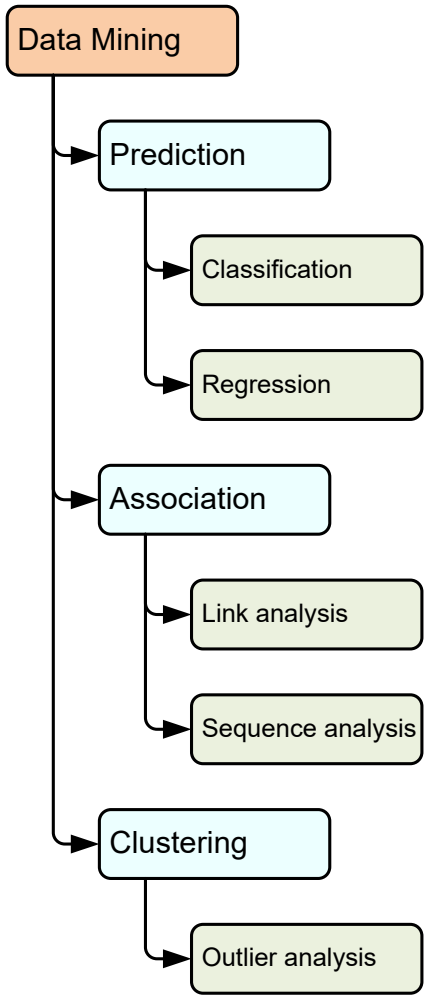
Predictions tell the nature of future occurrences of certain events based on what has happened in the past (predicting the absolute temperature of a particular day).

Clusters identify natural groupings of things based on their known characteristics (assigning customers in different segments based on their demographics and past purchase behavior).

Sequential relationships discover time-ordered events, such as predicting an existing banking customer who already has a checking account will open a savings account followed by an investment account within a year.

Data Mining Tasks

Based on the way in which the "patterns" are extracted from the historical data, the **learning algorithms of data mining methods** can be classified as either **supervised** or **unsupervised**.



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision trees, ANN/MLP, SVM, Rough sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression trees, ANN/MLP, SVM
Unsupervised	Apriory, OneR, ZeroR, Eclat
Unsupervised	Expectation Maximization, Apriory Algorithm, Graph-based Matching
Unsupervised	Apriory Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)

LINEACT cesi

Data Mining Applications

Computer hardware and software

Science and engineering

Government and defense

Homeland security and law enforcement

Travel industry

Healthcare

Medicine

Increasingly more popular application areas for data mining

Entertainment industry

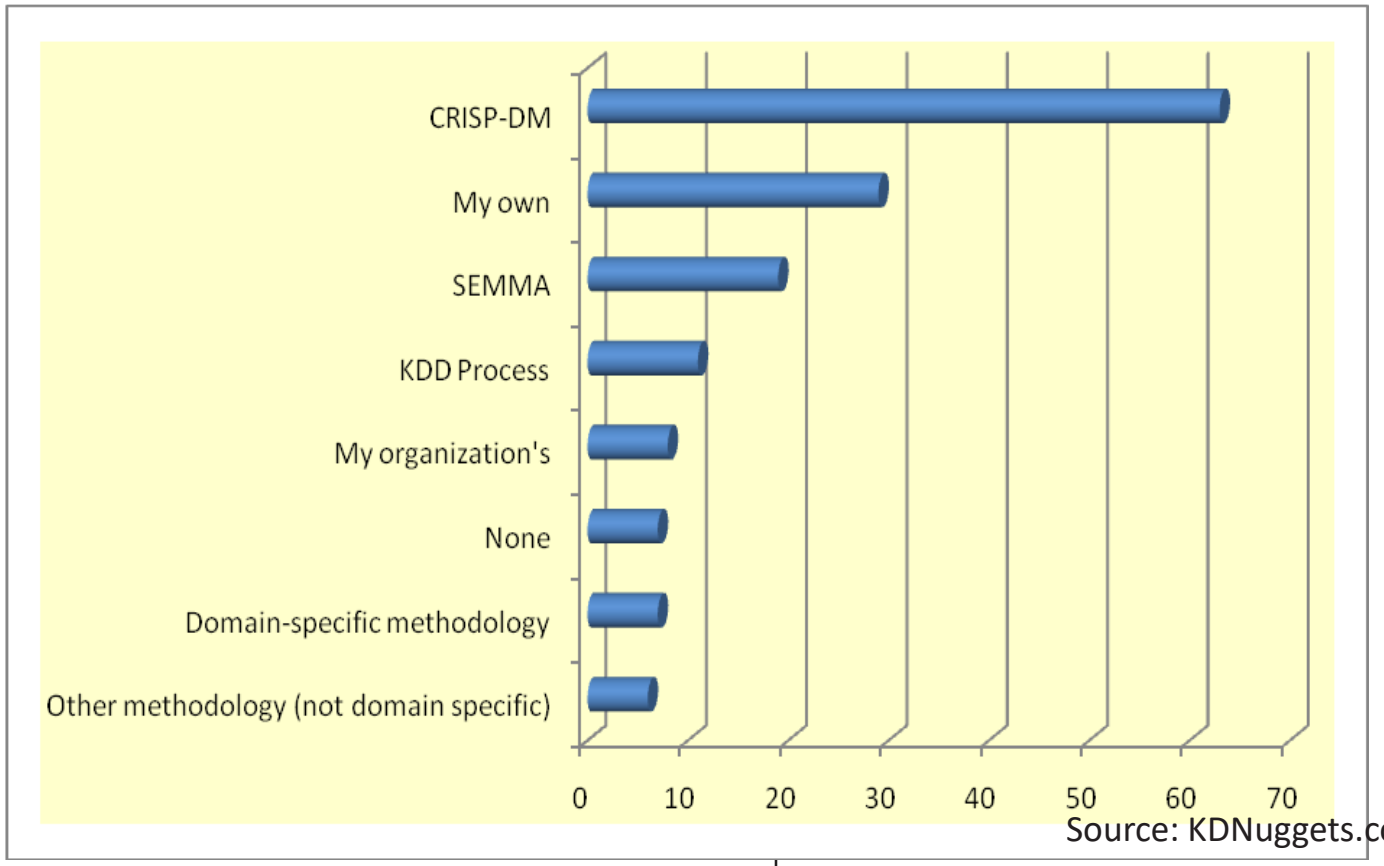
Sports

Etc.

LINEACT cesi

Data Mining Process

CRISP-DM (Cross-Industry Standard Process for Data Mining)
SEMMA (Sample, Explore, Modify, Model, and Assess)
KDD (Knowledge Discovery in Databases)



Source: KDNuggets.com

LINEACT cesi

Data Mining Process: CRISP-DM

Step 1: Business Understanding

Step 2: Data Understanding

Step 3: Data Preparation

Accounts for ~85% of total project time!

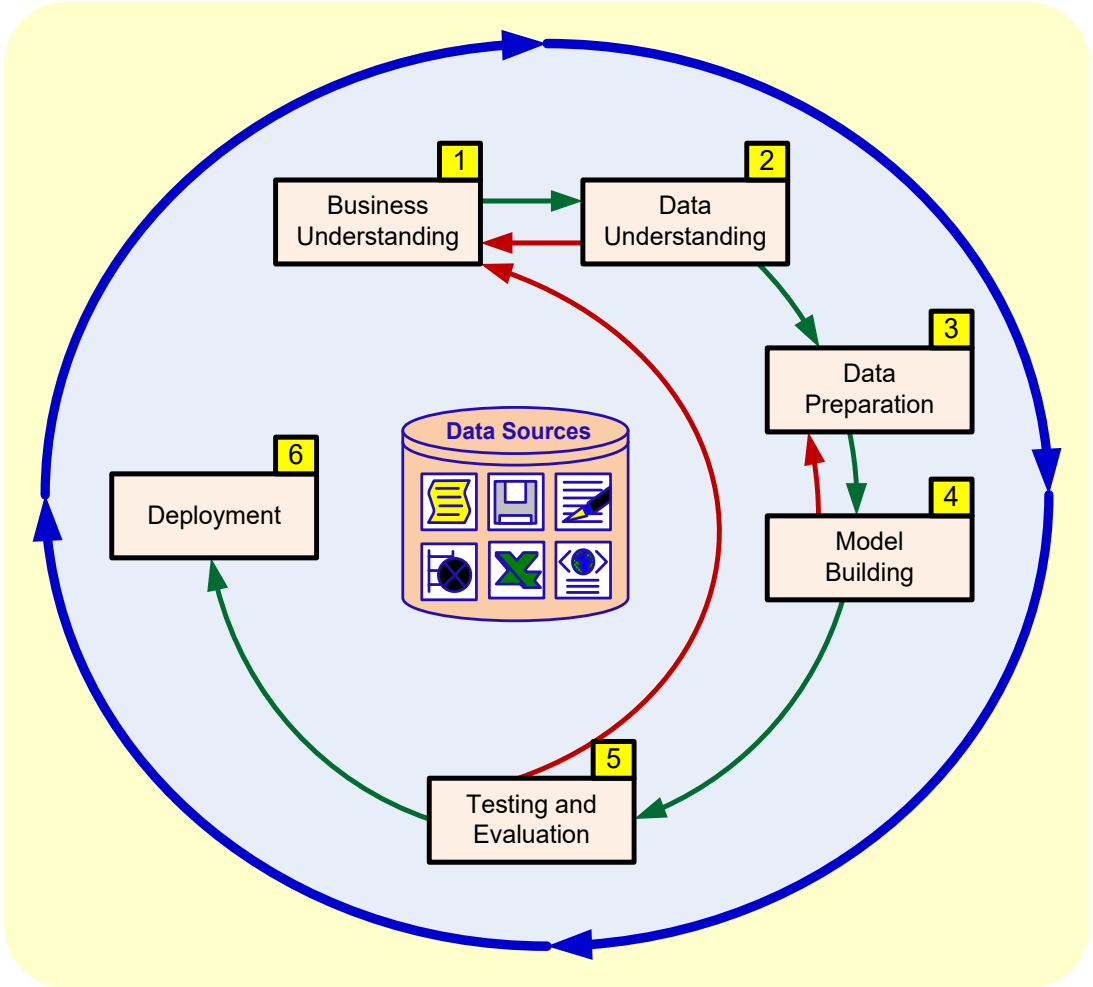
Step 4: Model Building

Step 5: Testing and Evaluation

Step 6: Deployment

The process is highly repetitive and experimental

Data Mining Process: CRISP-DM



Data Mining Process

Step 1: Business Understanding

The key element is to know what the study is for.

Specific goals such as:

What are the common characteristics of the - customers we have lost to our competitors recently?

What are typical profiles of our customers, and how much value does each of them provide to us?

Project Plan

Data Mining Process

Step 2: Data Understanding

A data mining study is specific to addressing a well-defined business task, and **different business tasks** require **different sets of data**.

Identify the relevant data from many available databases

The analyst should be clear and concise about the description of the data mining task so that **the most relevant** data can be identified.

e.g. A retail data mining project may seek to identify spending behaviors of female shoppers who purchase seasonal clothes based on their demographics, credit card transactions, and socioeconomic attributes.

The analyst should build an intimate understanding of the data sources (where the relevant data are stored and in what form; what the process of collecting the data is (automated versus manual); who the collectors of the data are and how often the data are updated) and the variables (What are the most relevant variables? Are there any synonymous variables?

Data Mining Process

Step 4: Model Building

Various modeling techniques are selected and applied to an already prepared data set in order to address the specific business need.

Step 5: Testing and Evaluation

The developed models are evaluated for their accuracy and generality. This step assesses the degree to which the selected model(s) meets the business objectives and, if so, to what extent.

Another option is to test the developed model(s) in a real-world scenario if time and budget constraints permit.

This step is a critical and challenging task. No value is added by the data mining task until the business value obtained from discovered knowledge patterns is identified and recognized.

Data Mining Process

Step 6: Deployment

Development and assessment of the models is not the end of the data mining project.

The knowledge gained from exploration of data need to be organized and presented in a way that the end user can understand and benefit from.

The deployment step may also include maintenance activities for the deployed models. Because everything about the business is constantly changing, the data that reflect the business activities also are changing.

Learning Machines

- Alan Turing proposed the concept of a learning machine in 1950 (in the same paper that proposed the Turing test).
- Idea: Divide the problem into two parts:
  - A machine that simulates a child's brain (analogous to a blank notebook: should function by simple mechanisms and have lots of blank sheets)
  - A way of teaching the child machine (should be simple since we know how to teach a human child)
- • Teacher rewards good behaviour and penalizes bad behaviour.

Learning Machines

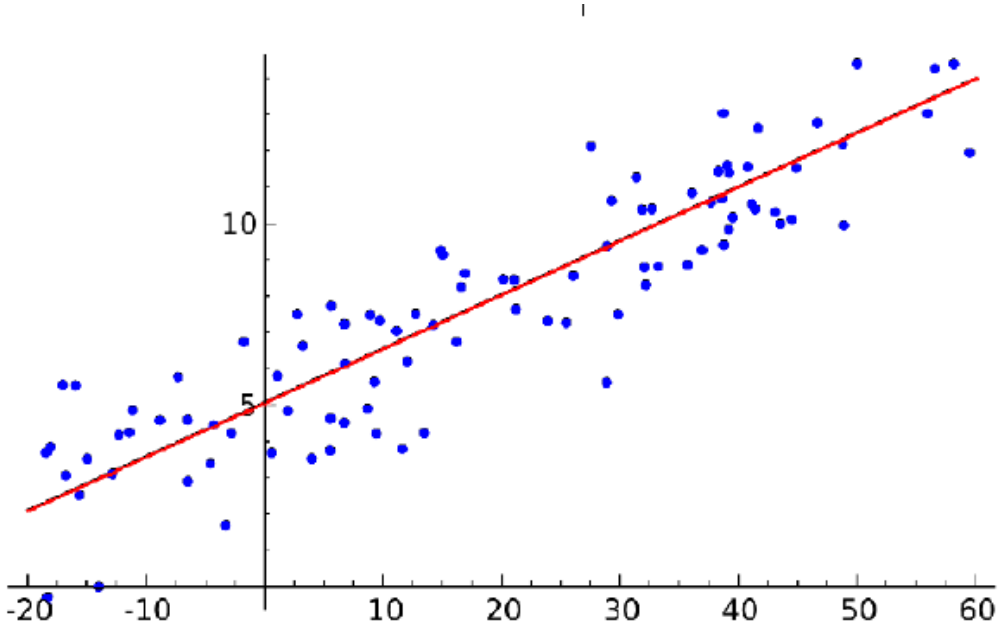
“An important feature of a learning machine is that its teacher will often be very largely ignorant of quite what is going on inside”

Alan Turing

- While we don’t know *how* our brain converts input to output, we know what the output should be for every input.
- We can use this knowledge to teach the machine.

Machine Learning

- In modern terms:
  - Child machine: *Model*
  - Blank sheets: *Model parameters*
  - Teacher: *Loss function*



Input

Predicted Output

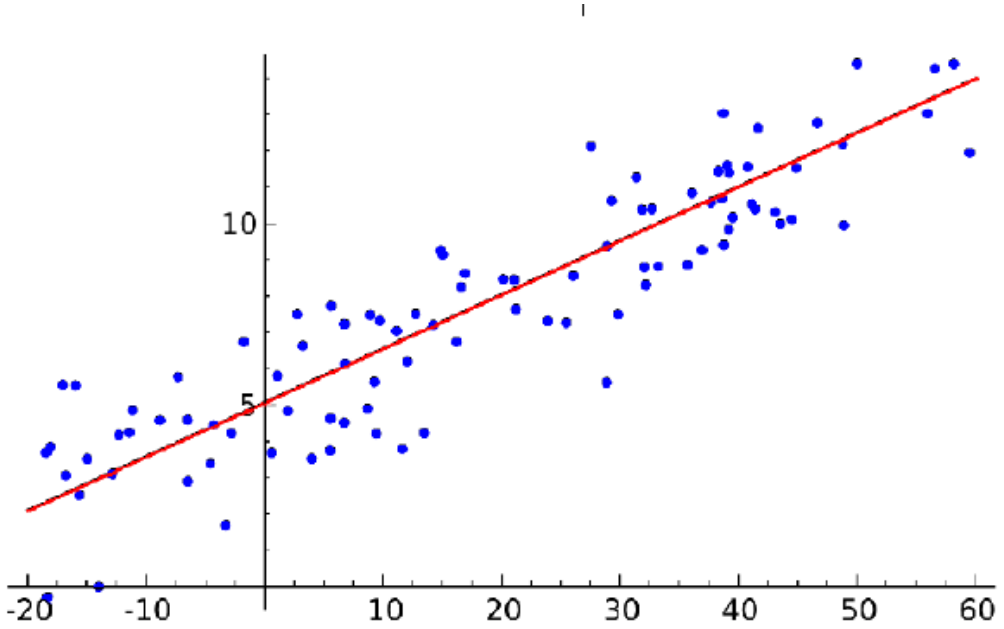
$$\hat{y} = wx + b$$

$$L = (y - \hat{y})^2$$

Desired Output

Machine Learning

- In modern terms:
  - Child machine: *Model*
  - Blank sheets: *Model parameters*
  - Teacher: *Loss function*



Parameters

Model → $\hat{y} = wx + b$

Loss Function → $L = (y - \hat{y})^2$

Part 2

Machine Learning

(Regression aka Fitting Curves to Data)

Part 2.1

Regression aka Fitting Curves to Data

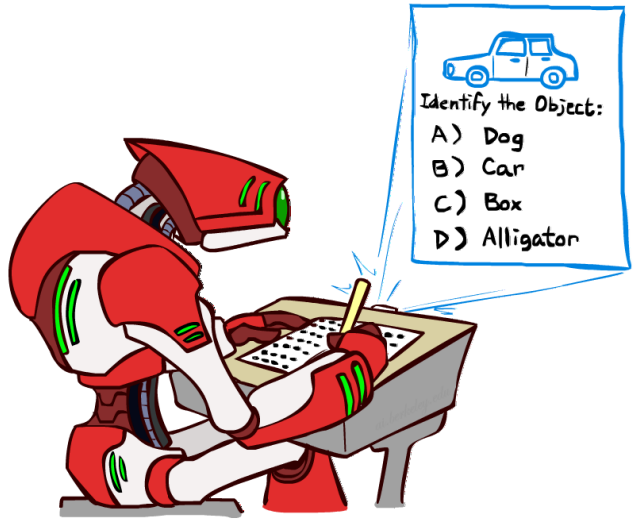
Linear Regression (Ordinary Least Squares)

Regression tasks

Regression = Fitting Curves to Data

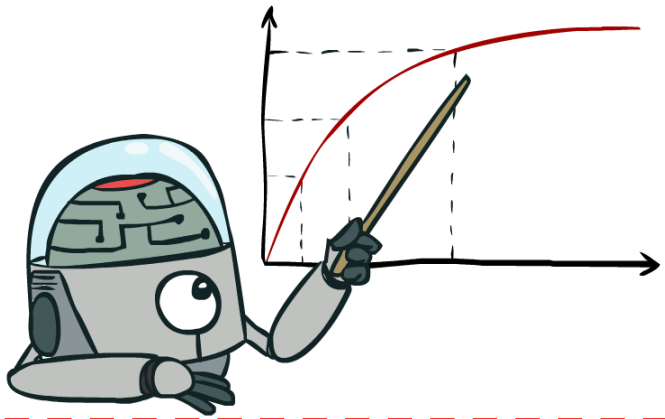
Classification

- Given point *x*, predict class
- Often binary
- Gives a descrite prediction



Regression

- Given point *x*, predict a numerical value
- Gives a quantitative prediction
- Usually on a continuous scale



Ordinary least squares

$$y \approx X\omega$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \approx \begin{bmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_n^T \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \end{bmatrix}$$

Goal: find the best w

Four levels for ML problems

1. Data & application
2. Model
3. Optimization problem
4. Optimization algorithm

1. Data & Application

$$y \approx X\omega$$

$$X = \begin{bmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_n^T \end{bmatrix} \qquad \vec{x}_i = \begin{bmatrix} \vec{x}_i^1 \\ \vdots \\ \vec{x}_i^d \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

2. Model

$$y \approx X\omega$$

$$\hat{y} = X\omega$$

3. Optimization problem

$$y \approx X\omega$$

$$\min_{w} \|X\omega - y\|_2^2$$

Goal: find the best w

3. Optimization problem

$$\|X\omega - y\|_2^2 = \sum_{\{i=1\}}^{n} (\vec{x}_i^T \omega - y_i)^2$$

$$= \sum_{\{i=1\}}^{n} (\hat{y} - y_i)^2$$

4. Optimization algorithm

Find ω that minimize

$$\sum_{i=1}^{n} (\vec{x}_i^T \omega - y_i)^2$$

- Convention
  - X is $n \times d$ **design matrix** of sample points
  - Y is *n*-vector of scalar labels
- Usually $n > d$ [but not always]

Point X_i^T

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1d} \\ \vdots & \vdots & & \vdots & & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nj} & \cdots & X_{nd} \end{bmatrix}$$

Feature column X_{*j}

4. Optimization algorithm

Note: we usually add a feature column $x_{i0} = 1$, generally called the *slope*

So $x_i \in \mathbb{R}^{\{d+1\}}$ and

$$X = \begin{bmatrix} X_{10} & X_{11} & X_{12} & & X_{1j} & X_{1d} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ X_{n0} & X_{n1} & X_{n2} & & X_{nj} & X_{nd} \end{bmatrix}$$

4. Optimization algorithm

- There's no solution to the system, so we try to fit the data as good as possible
  - Let ω be the best fit (parameter) solution to $y \approx X\omega$
  - We aim to minimize $J(w) = \|y - X\omega\|_2^2$
- Thus the optimal solution will be :

Residual sum of squares

$$\hat{\omega} = \min_{\omega} J(w) = \min_{\omega} \|y - Xw\|_2^2 = \min_{\omega} RSS(\omega)$$

4. Optimization algorithm

- Let's expand

$$J(\omega) = \left|\left|y - X\omega\right|\right|^2 = (y - X\omega)^T(y - X\omega) = [y^T - (X\omega)^T][y - X\omega]$$
$$= y^T y - y^T X\omega - (X\omega)^T y + (X\omega)^T(X\omega) = \cdots = y^T y - 2\omega^T X^T + \omega^T X^T X\omega$$

- We want to minimize $J(\omega)$, with respect to ω
  - Let's compute :

$$\frac{dJ(\omega)}{d\omega} = -2X^T y + 2X^T X\omega \overset{\text{def}}{=} 0 \Rightarrow \boldsymbol{X^T X\omega = X^T y} \Rightarrow \omega = (X^T X)^{-1} X^T y = X^+ y$$

$X^+, pseudoinverse\ of\ X, (d + 1 \times n)$

Looks familiar!
Normal Equations

4. Optimization algorithm

- Obeservation: The values of *y* are:

$$\hat{y}_i = \omega \cdot X_i \Rightarrow \hat{y} = X\omega = XX^{+}y = Hy$$

- The *H* matrix is called the « Hat matrix », because it puts the *Hat* on the *y*

OLS: example $\mathbb{R}^2$, case

Suppose we have the following dataset:

$$D = \{(1,1), (2,2), (3,2)\}$$

so we have this system:

$$\begin{cases} \omega_0 + \omega_1 = 1 \\ \omega_0 + 2\omega_1 = 2 \\ \omega_0 + 3\omega_1 = 2 \end{cases}$$

The matrix form is

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

No line goes through these points at once

OLS: example $\mathbb{R}^2$, case

We solve the *Normal Equations*

$$X^T X\omega = X^T y$$

$$X^T X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix},$$

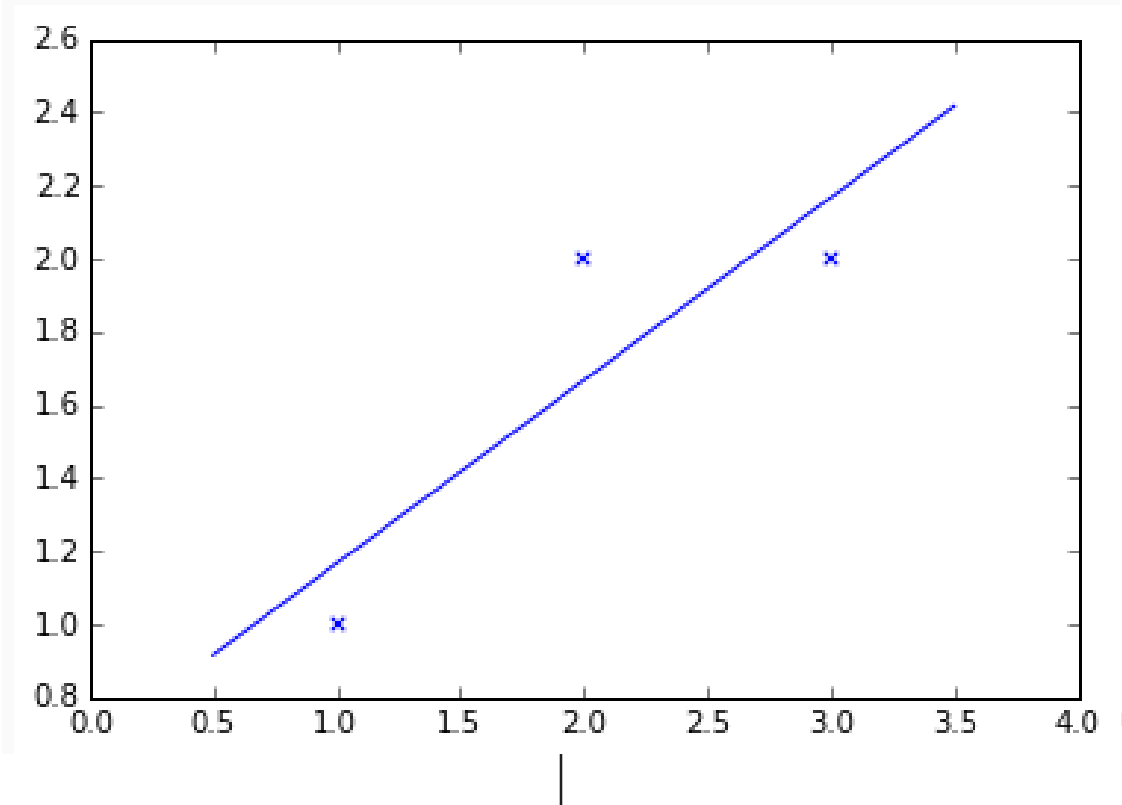
$$(X^T X)^{-1} = \frac{Com\,(A)}{\det(X^T X)} = \frac{1}{6}\begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix}$$

$$(X^T X)^{-1} y = \frac{1}{6}\begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix}\begin{bmatrix} 5 \\ 11 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{2} \end{bmatrix}$$

OLS: example $\mathbb{R}^2$, case

thus the best line is

$$y = \frac{2}{3} + \frac{1}{2}x$$



OLS: example $\mathbb{R}^2$, case

Let's compute the error vector

$$\hat{\omega} = \begin{bmatrix} 2 \\ \frac{1}{3} \\ \frac{1}{2} \end{bmatrix}, X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, y = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

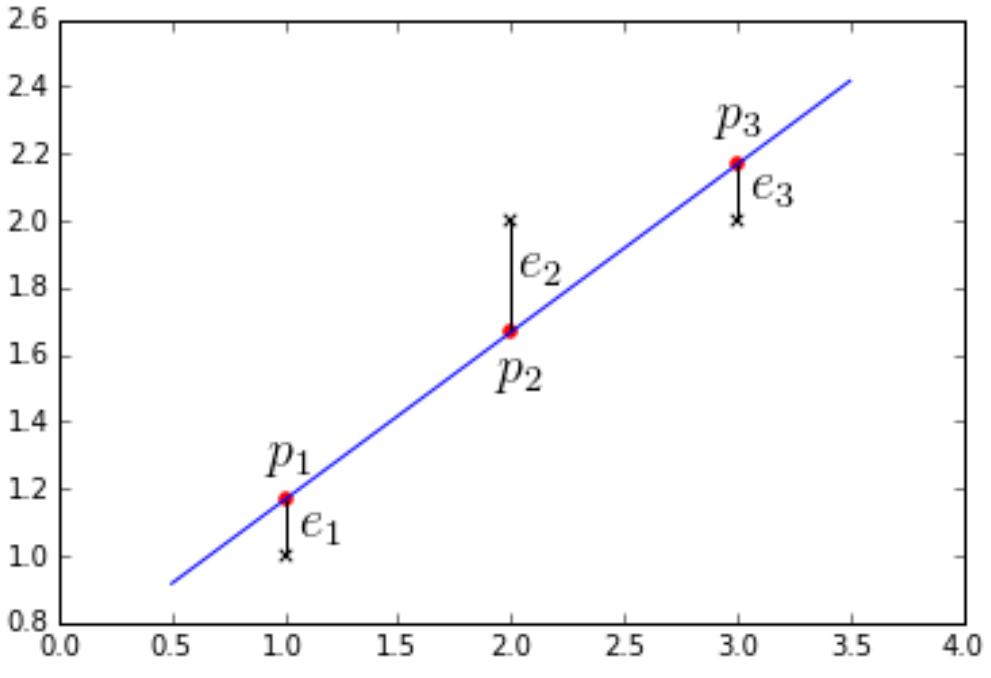
Projections

$$P = X\hat{\omega} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ \frac{1}{3} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{7}{6} \\ \frac{5}{3} \\ \frac{13}{6} \end{bmatrix}$$

OLS: example $\mathbb{R}^2$, case

Let's compute the error vector

$$e = y - \hat{y} = y - X\hat{\omega} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} \frac{7}{6} \\ \frac{5}{3} \\ \frac{13}{6} \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} \\ \frac{2}{3} \\ -\frac{1}{6} \end{bmatrix}$$



https://www.youtube.com/watch?v=ZUU57Q3CFOU&ab_channel=MITOpenCourseWare
https://www.youtube.com/watch?v=osh80YCg_GM&ab_channel=MITOpenCourseWare
http://mlwiki.org/index.php/Normal_Equation

OLS Summary

1.	Data & Application	Matrix *X* and vector *y*
2.	Model	Linearly *parametrized* function
3.	Optimization problem	Minimize sum of squared diffs
4.	Optimization algorithm	Normal equations

Part 2.2

Regression aka Fitting Curves to Data

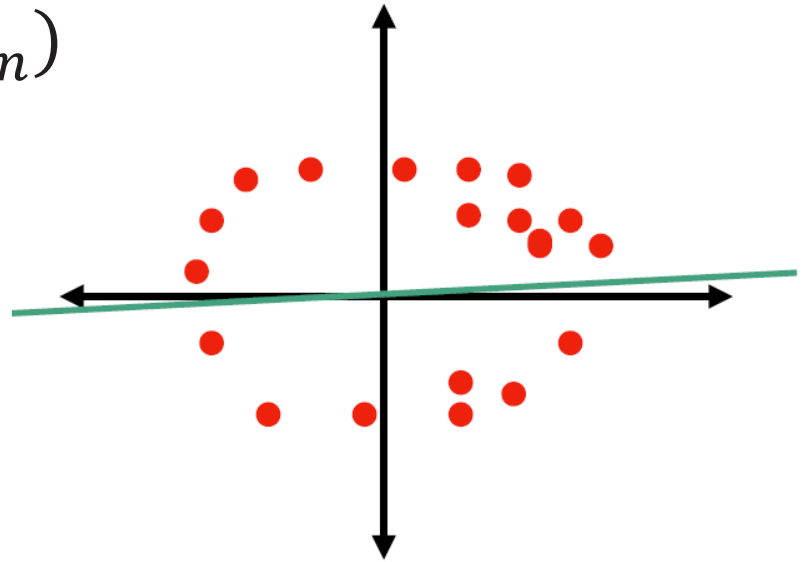
Polynomial Regression

LINEACT cesi

Linearity limits

Can we model this linearly?

(x_1, y_1)
(x_2, y_2)
$\vdots$
(x_n, y_n)

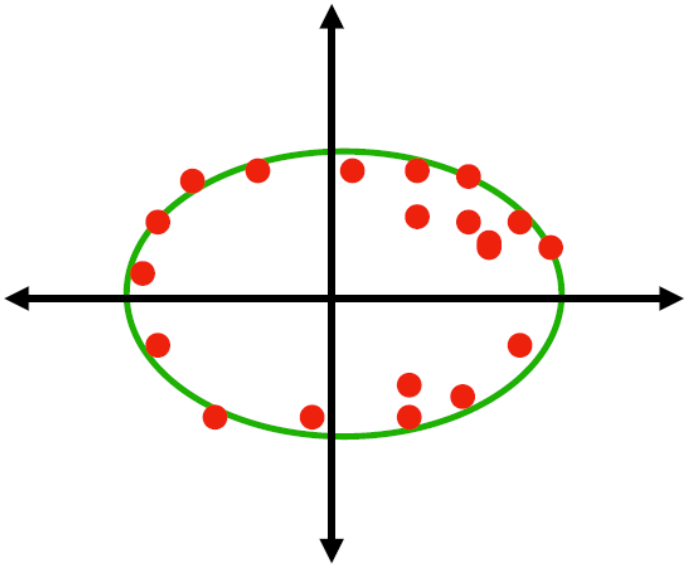


Can we model this linearly?

$R = c$ $\omega_5 y + \omega_6$

If $R =$

$\min_{\omega} ||$



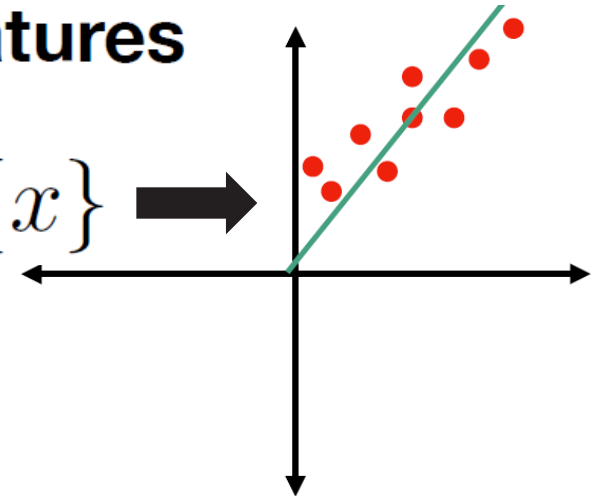
Plynomial regression introduction

Model

$$\hat{y}_i = w_1 x_i$$

Features

$\{x\}$

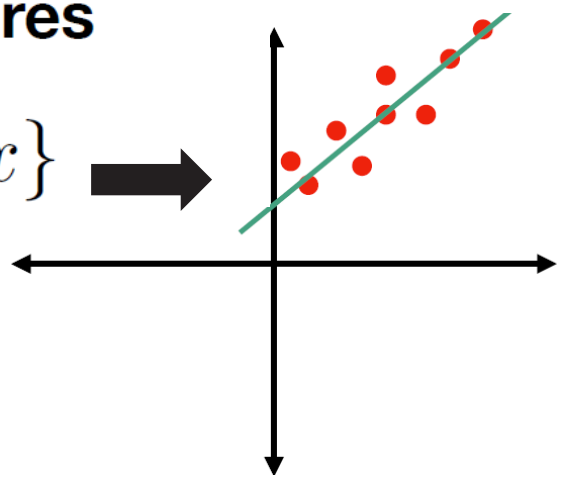


Model

$$\hat{y}_i = w_1 x_i + w_0$$

Features

$\{1, x\}$

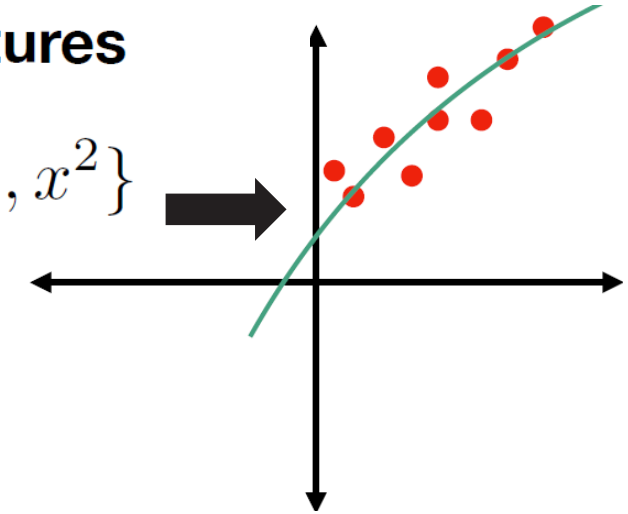


Model

$$\hat{y}_i = w_2 x_i^2 + w_1 x_i + w_0$$

Features

$\{1, x, x^2\}$



Polynomial features

Original data

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

Model

$$\widehat{y_i} = \sum_{j=0}^{p} w_i x_j^j$$

Features

$$\{0, x, x^2, x^3, \dots, x^p\}$$

Challenges with polynomial features

1. How to pick polynomial degree?

2. Sensitivity / numerical instability

3. Computational challenges for high degree (not covered today)

Challenges with polynomial features

1. ~~How to pick polynomial degree?/~~
   How to choose *hyperparameters*

2. Sensitivity / numerical instability

3. Computational challenges for
   high degree (not covered today)

How to choose hyperparameters?

Unrealistic Assumption : what if we have acess to the underlying model?

Real world : We can use validation sets!

How to choose hyperparameters?

Unrealistic Assumption : what if we have acess to the underlying model?

Real world : We can use validation sets!

How to choose hyperparameters?

Unrealistic Assumption : what if we have acess to the underlying model?

Real world : We can use validation sets!

Training set

How to choose hyperparameters?

Unrealistic Assumption : what if we have acess to the underlying model?

Real world : We can use validation sets!

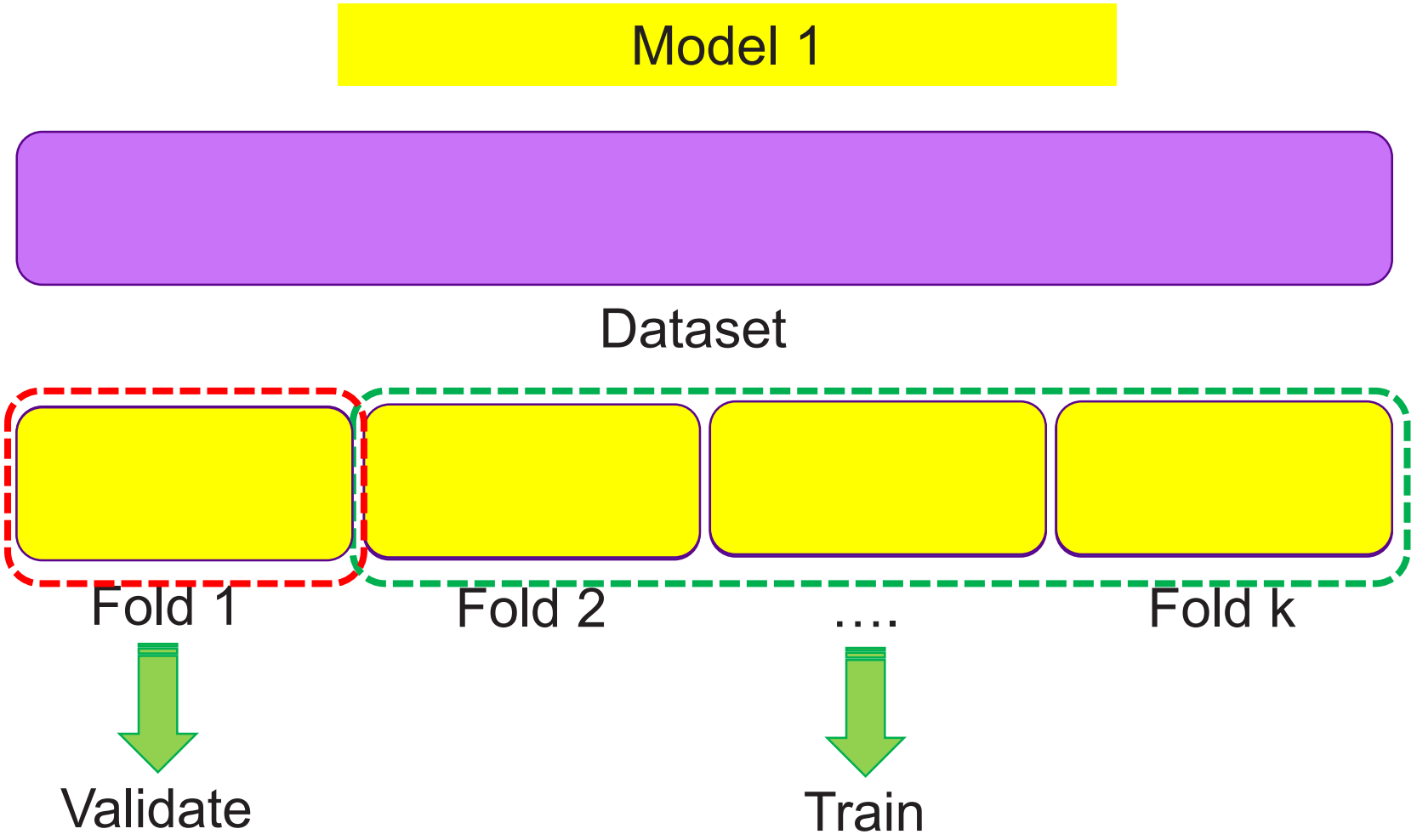
Training set

~70%

Validation set

~30%

Making more with less: k-fold cross validation



K-Fold Cross-Validation: Pseudo-Algorithm

K-fold cross validation

- for all hyperparameters:
  for i=1:k:
    train model on all folds except i
    evaluate performance on fold i
- Choose the hyperparameter with the best *average* val performance
- k is itself a hyperparameter, but usually set to 10 or 4

Challenges with polynomial features

1. ~~How to pick polynomial degree?/~~
   How to choose *hyperparameters*

2. Sensitivity / numerical instability

3. Computational challenges for
   high degree (not covered today)

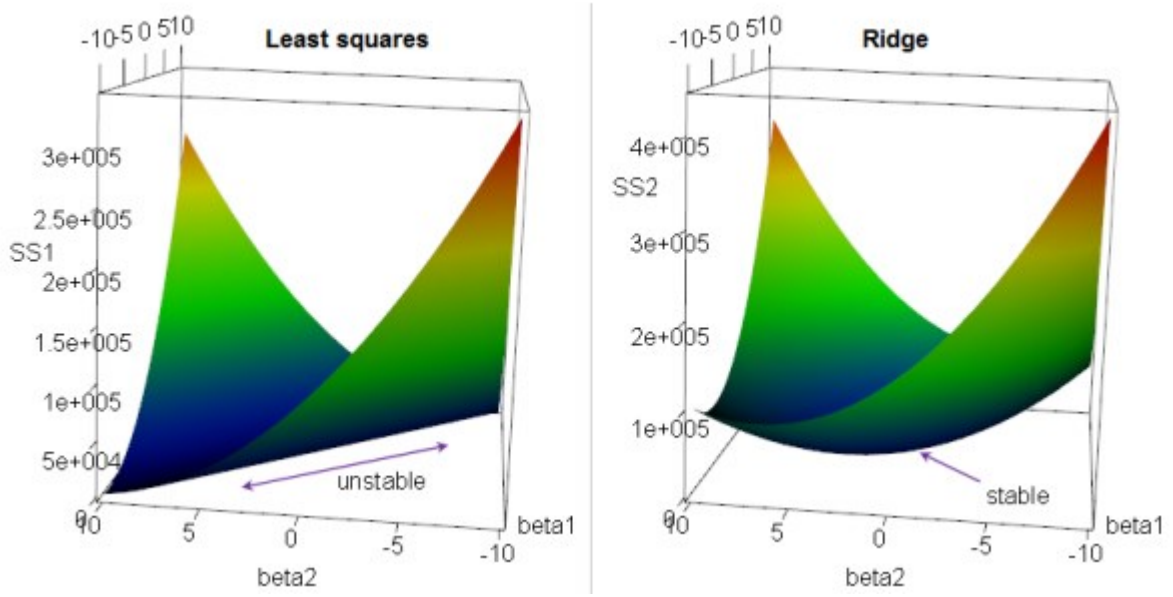
Regularization: Ridge regression *aka* Tikhonov Regularization

Motivation

- « Correct » for small eigenvalues in X^T X, by adding a multiple of the identity
- Prevent from overfitting and offer a good generalization

Solution

- Add a regularization term (aka penality term), for shrinkage
- Encourage small weights, to guarantee positive definite normal equations ➔ Always a unique solution



The cost function J(w) with and without regularization

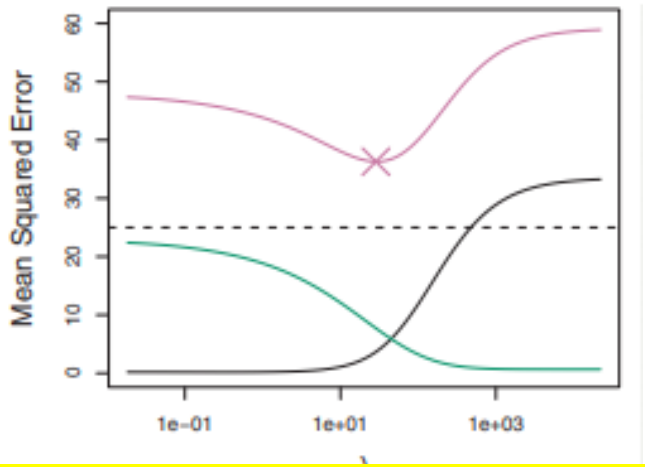
Regularization: Ridge regression *aka* Tikhonov Regularization

Prenvents/Reduces overfiting

- Imagine that: $500x_1 + 500x_2$ is the best fit for well-separated points with $y_i \in [0,1]$
- In this scenario, a small change in the X values ➔ a big change in the *y* values
- Given that all the y values are in *[0,1]* (i.e., data are small) and the x values are not, it's a sure sign of overfitting if tiny changes in x cause huge changes in y

So we penalize large weights

- By adding the regularization term in the *J(w)* equation
- Large variance, and a lot of overfiting ➔ Problem *close to* being ill-posed
- Setting $\nabla J(w) = 0$ gives the normal equations
$$(X^T X + \lambda I')\omega = X^T y$$
- λ is a hyperparameter; tune by (cross-)validation



Plot of bias^2 & variance as λ increases.

Ridge Regularization: summary

-

Hyperparameter

$$\min_w ||Xw - y||_2^2 + \lambda ||w||_2^2$$

Training error **Keep weights small**

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T y$$

Tricks for Ridge Regularization

- λ is a hyperparameter; tune by (cross-)validation.
- Ideally, features should be *"normalized"* to have same variance.
- Alternative: use asymmetric penalty by replacing *I'* with other diagonal matrix.

Part 2.3

Regression aka Fitting Curves to Data

Logistic Regression (the Sigmoid function)

Logistic Regression (David Cox, 1958)

Fits « *Probabilities* » in range (0,1)

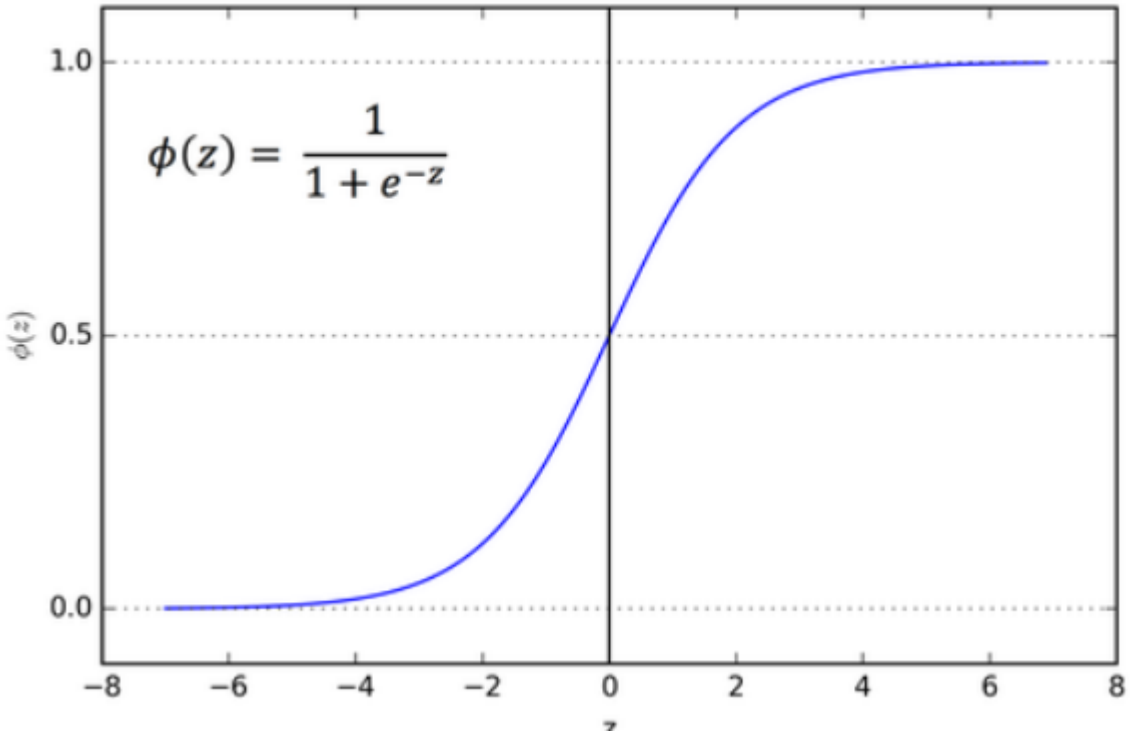
Usually used for classification tasks

- The label y_i can be probabilities
- But in most cases they are all 0 or 1

Logistic regression remind to the familly of disciminative models

Key notation ➔ Sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression : the binary class case (Model)

- Generally used for classification where labels are represented by 0 and 1
- Idea: we would like our model to output the probability that a data point remains to a specific class
  - Start with the raw linear score $\omega^T X$
  - The result is then transformed to a probability using the *Sigmoid* fnc
- For classification, the sigmoid is used to output a probability distribution *P(Y)*

$$P\left(\hat{Y} = 1 \middle| x, \omega\right) = s(\omega^T x), P\left(\hat{Y} = 0 | x, \omega\right) = 1 - s(\omega^T x)$$

Logistic Regression : the binary class case(continued)

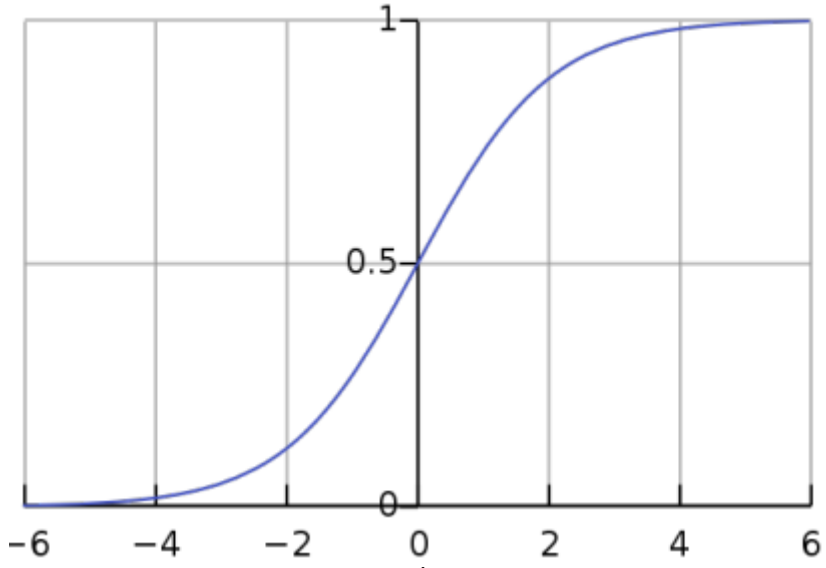
- We classify *X* as the class with the maximum probability

$$\hat{y} = \max_k P(\hat{Y} = k | x, \omega)$$

$$= \begin{cases} 1 & if\ s(\omega^T x) \geq 0.5 \\ 0 & otherwise \end{cases}$$

- Equivalently, we classify x as

$$\hat{y} = \begin{cases} 1 & if\ \omega^T x > 0 \\ 0 & otherwise \end{cases}$$



Logistic Regression : Loss Function (Optimization problem)

- Hypothesis
  - We are given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, of i.i.d (independent, and identically distributed) variables
- We already know OLS

$$\arg\min_{\omega} \sum_{i=1}^{n} \|y_i - s(\omega^T x_i)\|^2 + \lambda \|\omega\|^2$$

- However, this may not the the best choice!
  - Eg.: Consider linear regression on a categorical {0,1} outcomes. If the predicted value is 67 rather than 1, this will concentrate the efforts of OLS on reducing it, while it is not the case in logistic regression (i.e., outcome between 0 and 1).

Logistic Regression : Loss Function(continued)

- Insteadthe loss function we use for logistic regression is called the *log-loss,* or ***categorical cross-entropy***

$$L(\omega) = \sum_{i=1}^{n} y_i \ln\left(\frac{1}{s(\omega^T x_i)}\right) + (1 - y_i) \ln\left(\frac{1}{1 - s(\omega^T x_i)}\right)$$

- If we define $\mathrm{p_i} = s(\omega^T x_i)$, then using the properties of *logs* we can express the previous formula

$$L(\omega) = -\sum_{i=1}^{n} y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)$$

- For each x_i, p_i represents our predicted probaility that its corresponding class is 1
  - Because $y_i \in \{0,1\}$, the loss corresponding to the *i*'th data point is

$$L_i(\omega) = \begin{cases} -\ln(p_i) & y_i = 1 \\ -\ln(1 - p_i) & y_i = 0 \end{cases}$$

Logistic Regression : Maximum Likelihood Estimation (Optimization Problem)

- Each observation y_i is viewed as an independent sample from a *Bernoulli Distribution* $\widehat{Y}_i|x_i, \omega \sim Bern(p_i)$
- Thus, the observation y_i, has probability

$$P(\widehat{Y}_i = y_i) = \begin{cases} p_i & if\ y_i = 1 \\ 1 - p_i & if\ y_i = 0 \end{cases}$$

- One convenient way to write the likelihood of a single data point is

$$P(\widehat{Y}_i = y_i) = p_i^{\{y_i\}}(1 - p_i)^{1-y_i}$$

- Note: p_i is just a transformation of x_i, that results in a number from 0 to 1 (i.e., Sigmoid function)

Logistic Regression : Maximum Likelihood Estimation(continued)

- Now we can estimate the parameters w via maximum likelihood. We have the problem

$$
\begin{aligned}
\hat{\mathbf{w}}_{\text{LR}} &= \arg\max_{\mathbf{w}} P(\hat{Y}_1 = y_1, \ldots, \hat{Y}_n = y_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(\hat{Y}_i = y_i \mid \mathbf{x}_i, \mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{(1-y_i)} \\
&= \arg\max_{\mathbf{w}} \ln \left[\prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{(1-y_i)} \right] \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \\
&= \arg\min_{\mathbf{w}} - \sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) \ln(1 - p_i)
\end{aligned}
$$

Logistic Regression : Training (Optimization algorithm)

- Recall the loss function

$$L(\omega) = -\sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) \ln(1 - p_i)$$

- Where

$$p_i = s(\omega^T x_i) = \frac{1}{1 + e^{-\omega^T x_i}}$$

- First derivate gives

$$\begin{aligned}
\nabla_{\mathbf{w}} L(\mathbf{w}) &= \nabla_{\mathbf{w}} \left(-\sum_{i=1}^{n} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \right) \\
&= -\sum_{i=1}^{n} y_i \nabla_{\mathbf{w}} \ln p_i + (1 - y_i) \nabla_{\mathbf{w}} \ln(1 - p_i) \\
&= -\sum_{i=1}^{n} \frac{y_i}{p_i} \nabla_{\mathbf{w}} p_i - \frac{1 - y_i}{1 - p_i} \nabla_{\mathbf{w}} p_i \\
&= -\sum_{i=1}^{n} \left(\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i} \right) \nabla_{\mathbf{w}} p_i
\end{aligned}$$

Logistic Regression : Training (Optimization algorithm)(continued)

- Note that $\nabla_\omega s(z) = s(z)\big(1 - s(z)\big)$, and from the chain rule we have,

$$\nabla_\omega p_i = \nabla_\omega s(\omega^T x_i) = s(\omega^T x_i)\big(1 - s(\omega^T x_i)\big) = p_i(1 - p_i)x_i$$

- Plugging this gradient value, we have

$$\begin{aligned}
\nabla_{\mathbf{w}} L(\mathbf{w}) &= -\sum_{i=1}^{n} \left(\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i} \right) \nabla_{\mathbf{w}} p_i \\
&= -\sum_{i=1}^{n} \left(\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i} \right) p_i(1 - p_i)\mathbf{x}_i \\
&= -\sum_{i=1}^{n} \big(y_i(1 - p_i) - (1 - y_i)(p_i) \big) \mathbf{x}_i \\
&= -\sum_{i=1}^{n} (y_i - p_i)\, \mathbf{x}_i
\end{aligned}$$

- The *gradient descent* update is thus

$$\omega^{t+1} = \omega^t + \epsilon \sum_{i=1}^{n} (y_i - p_i) x_i$$

Regression and Loss functions

Regression functions

- Linear

$h(x;\omega,\alpha) = \mathrm{w}\cdot x + \alpha$

- Plynomial
  - Equivalent to linear regression with added polynomial features
- Logistic

$h(x;\omega,\alpha) = \mathrm{s}(\mathrm{w}\cdot x + \alpha),$

$s(\gamma) = \frac{1}{1 + \backslash\exp^{-\gamma}}$

Loss functions

- Squared Error

$L(z,y) = (\mathrm{z}-\mathrm{y})^2$

- Abolute Error

$L(z,y) = |\mathrm{z}-\mathrm{y}|$

- Logistic aka cross-entropy

$L(z,y) = -\mathrm{y}\ln(\mathrm{z}) - (1-\mathrm{y})\ln(1-\mathrm{z})$