When Models Meet Data

Data, Models, and Learning
Empirical Risk Minimization

Section 8.1 and 8.2 from *Mathematics for Machine Learning*

Sara Martín, Mariona Pla, Martina Carretta i Ares Sellart

INDEX

DATA, MODELS AND LEARNING

- DATA AS VECTORS
- MODELS AS FUNCTIONS
- MODELS AS PROBABILITY DISTRIBUTIONS
- FINDING PARAMETERS

Introduction: what is a good model?

3 components of a machine learning system:

- **DATA**: unprocessed value, text, picture, sound…
  - Training data: initial data we use to train our model.
- **MODELS**: file that has been trained to recognize certain types of patterns
- **LEARNING**

A good model should perform well on unseen data.

→ **Performance metrics**: measure the performance of a model.

Data as vectors

Our computer should be able to read our data → **TABULAR DATA**

Each column = particular feature ; Each row = particular example

Even thought machine learning can be applied in other types of data (ex. Genomic sequences), we will focus on tabular data.

Owner	Country	File_Date	IPC_Class
Company A	US	6/18/2008	H05H13
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	EP	1/30/1998	A61N5
Company A	JP	8/28/1997	A61N5
Company A	JP	10/4/2002	A61N5
Company A	JP	1/27/2003	A61N5
Company A	JP	4/14/2003	A61N5
Company A	JP	5/13/2011	A61N5
Company B	JP	4/2/1998	G12B13
Company B	JP	4/2/1998	G12B13
Company B	JP	5/28/1997	A61N5
Company B	JP	11/12/1997	A61N5
Company B	JP	2/29/2000	A61N5
Company B	JP	4/30/2002	A61N5

Example of data represented as rows and columns

... Our tabular data is not yet prepared.

Data as vectors: Transforming tabular data

Name	Gender	Degree	Postcode	Age	Annual salary
Aditya	M	MSc	W21BG	36	89563
Bob	M	PhD	EC1A1BA	47	123543
Chloé	F	BEcon	SW1A1BH	26	23989
Daisuke	M	BSc	SE207AT	68	138769
Elisabeth	F	MBA	SE10AA	33	113888

Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

Data as vectors: Transforming tabular data into vector representation

Each input (x_n) is a D-dimensional vector of real numbers, called:

- **Features**: individual measurable properties (ex. height)
- **Attributes**: features of each data point (ex. 170cm)
- **Covariants**: independent variables

Notation:

- N= number of examples in dataset
- n= index of the examples (n=1,...,n=N)
- Each row = one example x_n
- Each column = one feature (d=1,...,d=D)

Features

Examples

Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

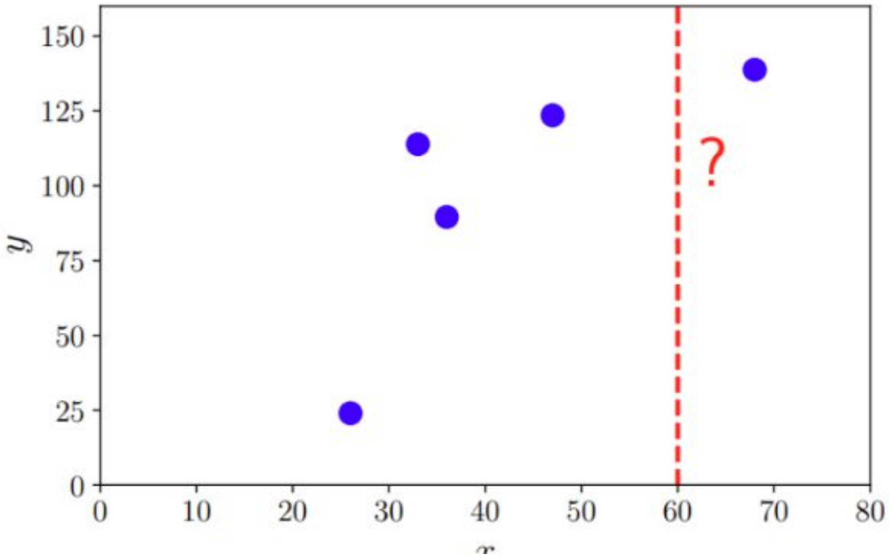
Example: predicting annual salary from age

Notation

- x=age
- y= salary

Age	Annual Salary (in thousands)
36	89.563
47	123.543
26	23.989
68	138.769
33	113.888

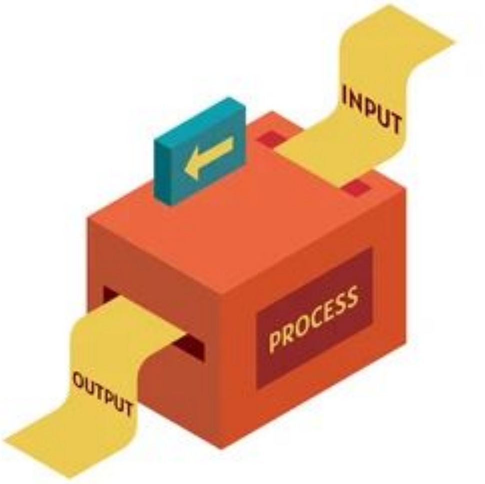
We want to know the salary of a person of age 60:



Models as function

INPUT: vector of D dimensions → Features

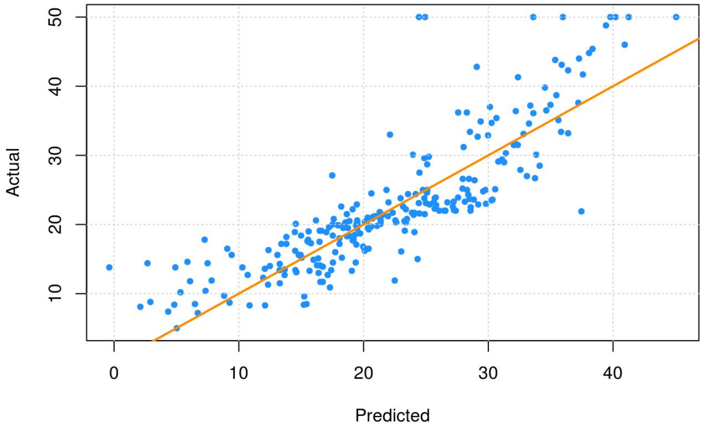
$f : R^D \rightarrow R$



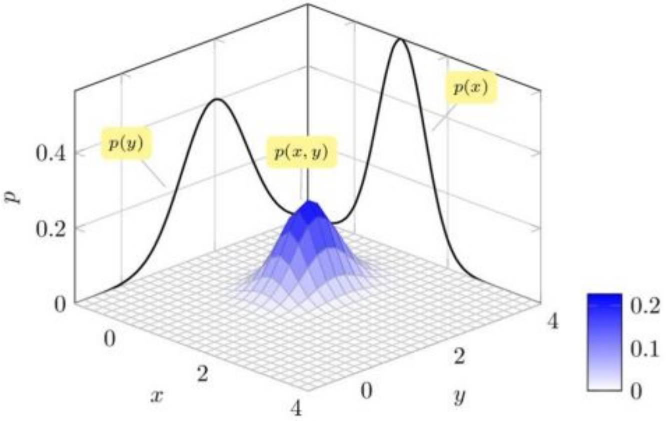
Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

Models as probability distributions

- NOISY OBSERVATIONS!!

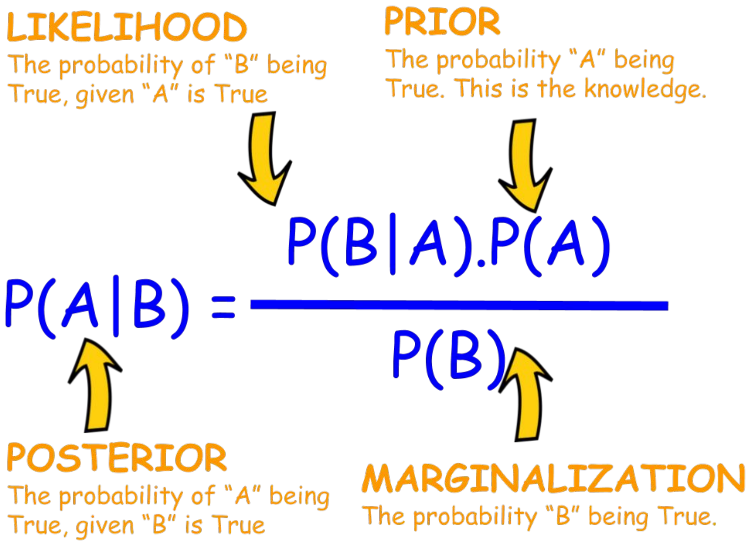


- Distribution of possible functions and multivariate probability distributions!!



Finding parameters

- Prediction / inference
- Training or parameter estimation
  - Quality
  - Bayesian inference
- Hyperparameter turning or models selection.



EMPIRICAL RISK MINIMIZATION

- ➔ HYPOTHESIS CLASS OF FUNCTIONS
- ➔ EMPIRICAL RISK MINIMIZATION
- ➔ LOSS FUNCTION FOR TRAINING
- ➔ REGULARIZATION TO REDUCE OVERFITTING
- ➔ CROSS-VALIDATION TO ASSES THE GENERALIZATION PERFORMANCE

Hypothesis Class of Functions

PROBLEM: *Separating spam from legitimate emails*

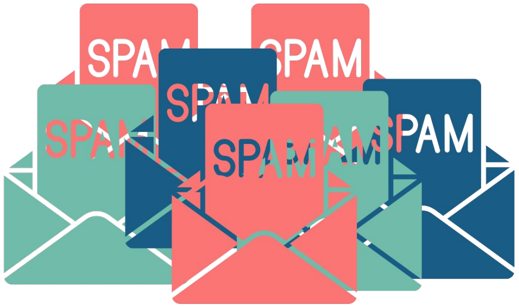
- N examples where emails are labelled 0 or 1
  - 0→ spam
  - 1→ not spam

(x1, y1), . . . , (xn , yn)

PREDICTOR

(θ)

$$f(\boldsymbol{x}_n, \boldsymbol{\theta}^*) \approx y_n \quad \text{for all} \quad n = 1, \ldots, N$$



Empirical Risk Minimization

ERM: provides bounds to the performance of a family of learning algorithms

finding a function that minimizes the empirical risk between predicted output and the actual output

2 0 0

100 WHITE CATS

100 BROWN DOGS

Cats and dogs are different because of its colour

GOAL: find a reliable predictor

How well does the predictor fit the data?

Loss Function for Training

LOSS FUNCTIONS define what a valid prediction is and isn't

INPUT: Truth label and the prediction → positive number (LOSS)

→ Represents how much error we have made on one particular prediction

GOAL: find a parameter vector to minimize the average loss on the set of N examples

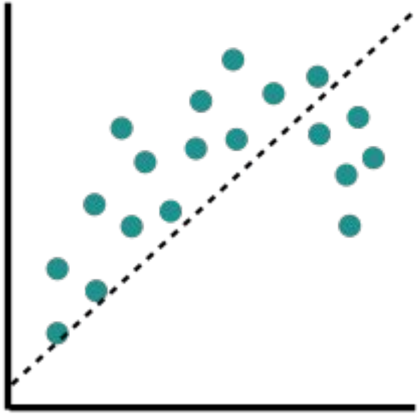
-How should we change our training procedure to generalize well?

-How do we estimate expected risk from (finite) data?

REAL OBJECTIVE: gain a predictor that also performs well (has low risk) on unseen test data

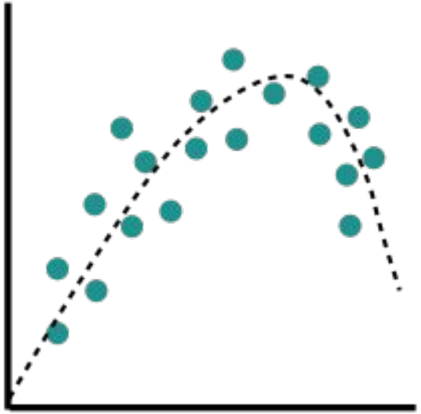
Regularization to Reduce Overfitting: What is overfitting?

Under-fitting



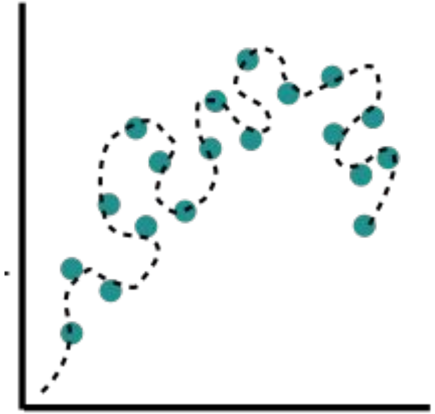
The model does not fit enough points, losing important information

Appropriate-fitting



The model fits the correct amount of data and generalizes without losing important information

Overfitting



The model doesn't generalize well. A lot of irrelevant data points. More possibilities of overfitting when the model has lots of features to learn

Regularization to Reduce Overfitting

Ways to reduce overfitting:

- Reduce the number of variables in a model ➜ increase in the degrees of freedom ➜ loss of relevant information
- **Regularization** ➜ keeps all the features and reduces the magnitude of the features available ➜ trade-off between accuracy and generalizability

Cross-Validation to Assess the Generalization Performance

DATA ➜ train machine learning methods and test them

Same data to train and test ➜ how does the model work with unseen data? ➜ **PROBLEM**: we don’t have unseen data

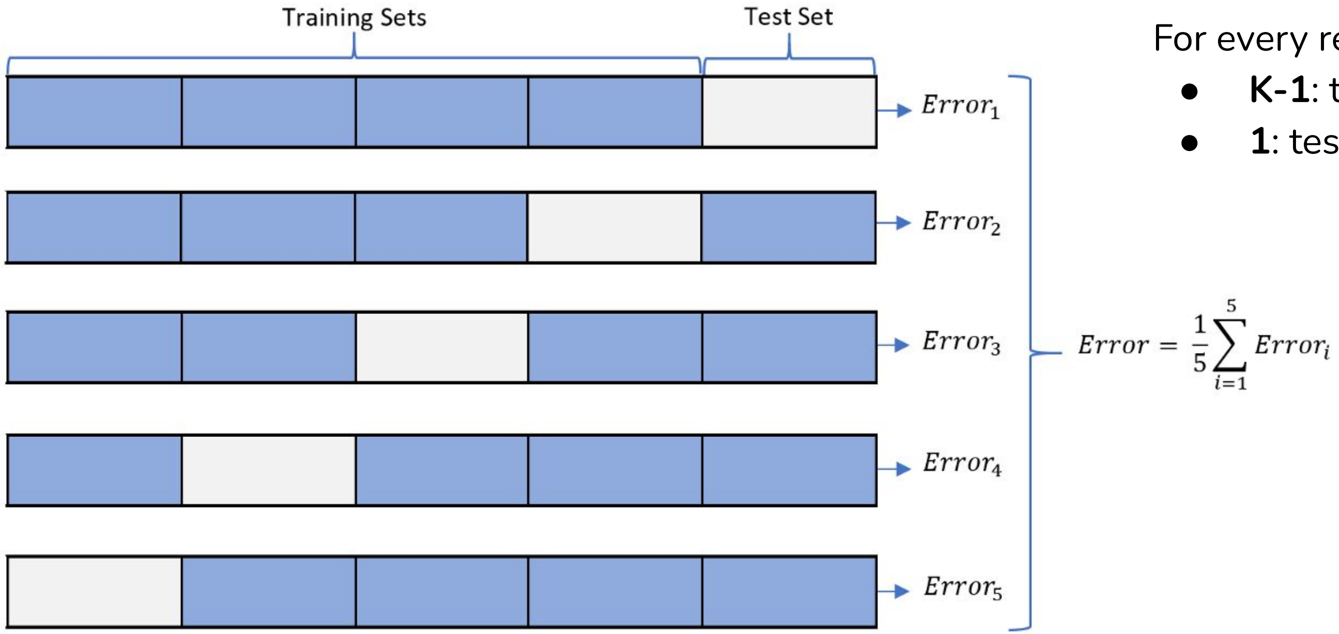
SOLUTION ➜ divide the data (different for training and testing) ➜ **HOW**?

Cross-Validation ➜ compare different machine learning methods and get a sense of how they will work in practice

K-Fold Cross-Validation

➔ **K**: divisions of the data and repetition times

Example of 5-Fold Cross-Validation



For every repetition:

- **K-1**: training set
- **1**: testing set

THE END

Any questions?