From Data to Models classification, prediction, and synthesis

Carlo A. Furia

Software Institute, USI

bugcounting.net

Roadmap

- Introduction: who I am, who you are
- AI, machine learning, and other buzzwords
- Classification: Bayesian probability
 - Spam or ham?
- Prediction: linear regression
 - How much will this car cost?
- Synthesis: propositional satisfiability
 - A program to solve Sudoku

In the afternoon: hands on!

Set-up

How to follow this tutorial: software

Download and install the R platform:

https://www.r-project.org/

Optionally, download and install RStudio: https://www.rstudio.com/

Install the R libraries we'll use, by launching R and executing the command:

```
> install.packages(c("tm", "rpicosat"))
```

How to follow this tutorial: data

Download and unzip this archive:

https://github.com/bugcounting/data2models/archive/master.zip

Unzipping it will create a folder data2models

Launch R in subfolder data2models/spam, and execute the script pull_data.R as follows:

> source("pull_data.R")

This will download and unpack additional data that we will use.

A (very) brief history of Al

A brief history of Al

- Birth: 1950-1956
- Golden years: 1956-1974
- First Al winter: 1974-1980
- Expert systems: 1980-1987
- Second Al winter: 1987-1993
- Modern times: 1993-today

https://en.wikipedia.org/wiki/History of artificial intelligence

Birth: 1950-1956

Strong Al

- Turing's "Artificial computing an intelligence" introduces Turing's test
- Pitts & McCulloch: neural networks
- Newell & Simon: logic reasoning
- McCarthy: Lisp

Golden years: 1956-1974

Strong Al

- Logic planning/constrain-based programming
- Natural language processing
 - Weizenbaum's ELIZA
- Minsky & Papert's micro worlds

First Al winter: 1974-1980

- Limited computational power
- Intractable problems
- Domain knowledge

Expert systems: 1980-1987

Weak(er) Al

- Feigenbaum's expert systems
- Knowledge-based systems
- Focus on domain-specific knowledge and rules

Second Al winter: 1987-1993

- Limited learning power
- Brittle behavior
- Slow progress

Modern time: 1993-today

- Machine learning
 - Bayesian statistics
- Deep learning
 - More computational power
 - Featurless learning

From data to models: the idea

This tutorial in one picture



The model can then be used for various things:

- Classification
- Prediction
- Synthesis (i.e., as an implementation)

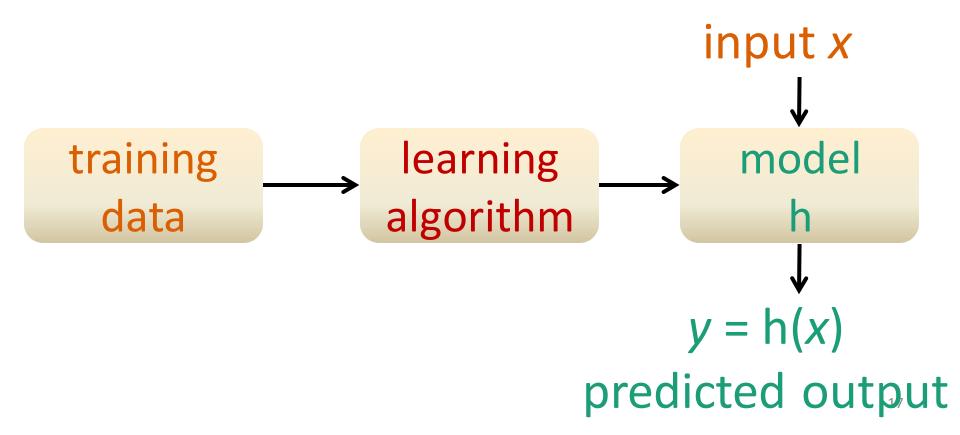
Learning from data

Learning denotes a broad category of algorithms to build computational models of data.



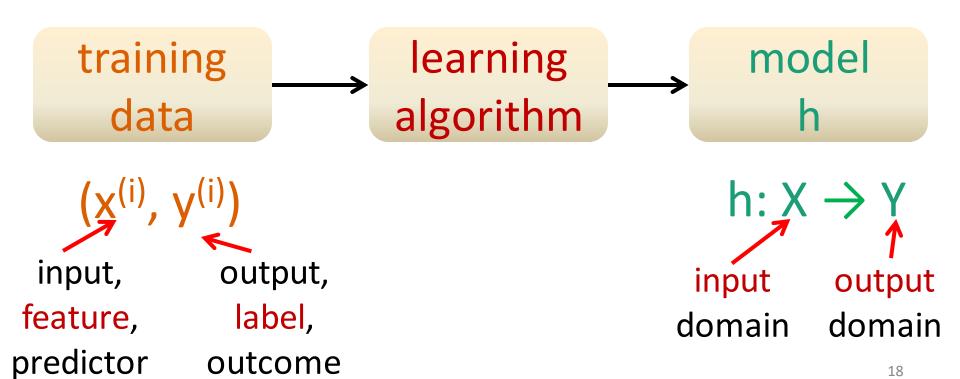
Learning from data

Learning denotes a broad category of algorithms to build computational models of data.



Inputs and outputs

The kinds of data models depend on the kinds of inputs and outputs.



training learning model algorithm h

(email message, spam/not spam)

training learning model algorithm h

(email message, spam/not spam)

spam filter



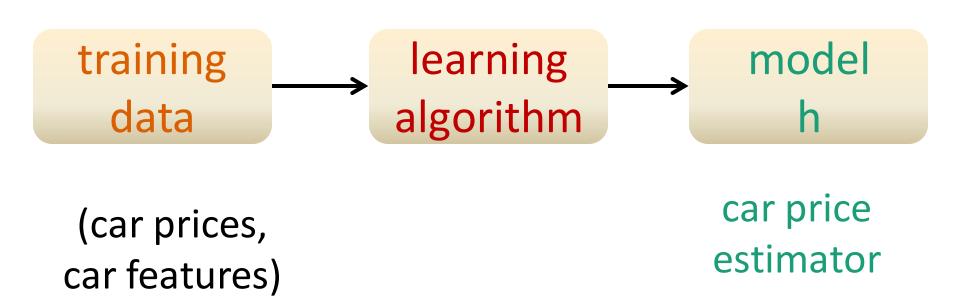
(email message, spam/not spam)

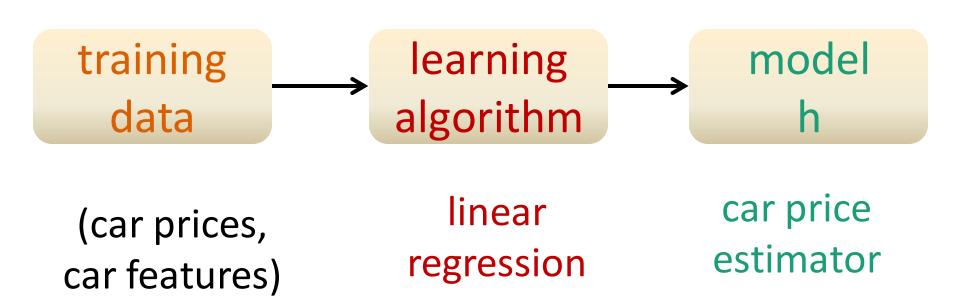
naïve Bayes classifier

spam filter

training _____ learning _____ model h

car price estimator

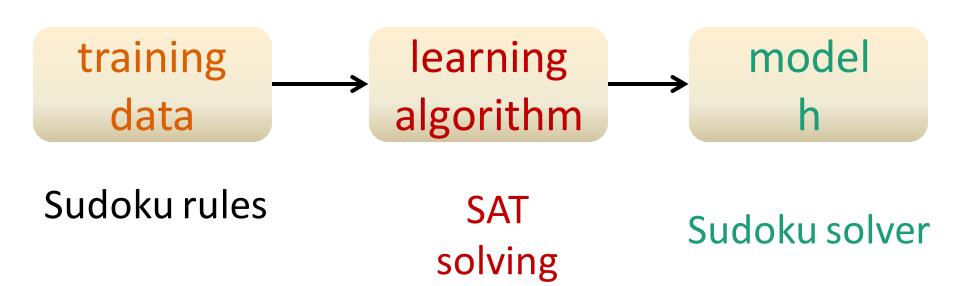




training learning model algorithm h

Sudoku rules

Sudoku solver



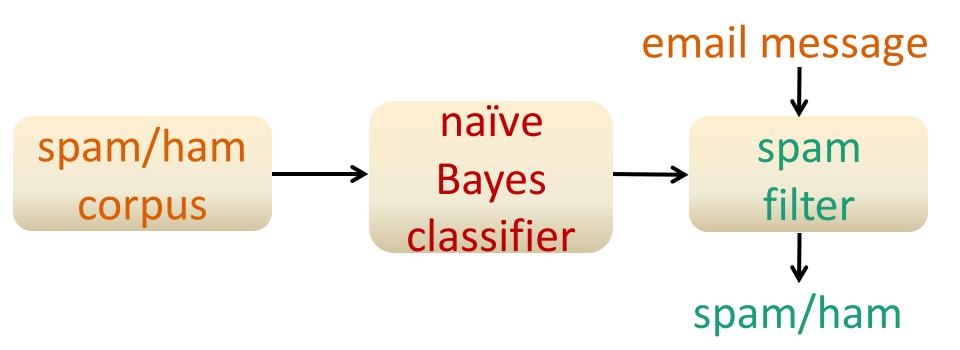
Classification

Spam or ham?



https://www.youtube.com/watch?v=zLih-WQwBSc

Spam filter



Probabilities

For a message m, the model estimates the probabilities that:

- P_s = P[spam | m]: m is a spam message
- $P_H = P[ham \mid m]$: m is a ham message

- $P_s > P_H$: classify m as spam
- $P_s \le P_H$: classify m as ham

Bayes' theorem

We compute the probabilities P_s and P_H by means of Bayes theorem:

$$P[A \mid B] = P[B \mid A] \times P[A]/P[B]$$

$$P[spam|m] = P[m | spam] \times P[spam]/P[m]$$

 $P[ham|m] = P[m | ham] \times P[ham]/P[m]$

Prior and likelihood

```
P[spam|m] = P[m | spam] \times P[spam]/P[m]
P[ham|m] = P[m | ham] \times P[ham]/P[m]
```

- Since we want to compare the two probabilities, we don't need to compute P[m]
- P[spam] and P[ham] are the priors, which reflect our initial assumptions
- P[m | spam] and P[m | ham] are the likelihoods, which we compute based on the distribution of data in the training set (the spam/ham corpus)

Bag of words model

Each message m is represented as a set of the words it contains (including repetitions).

$$m = \{ w_1, w_2, ... \}$$

Bag of words model

Each message m is represented as a set of the words it contains (including repetitions).

```
naïve = independent  m = \{ w_1, w_2, ... \}  probabilities  P[m \mid spam] = P[w_1 \mid spam] \times P[w_2 \mid spam] \times ...
```

Bag of words model

Each message m is represented as a set of the words it contains (including repetitions).

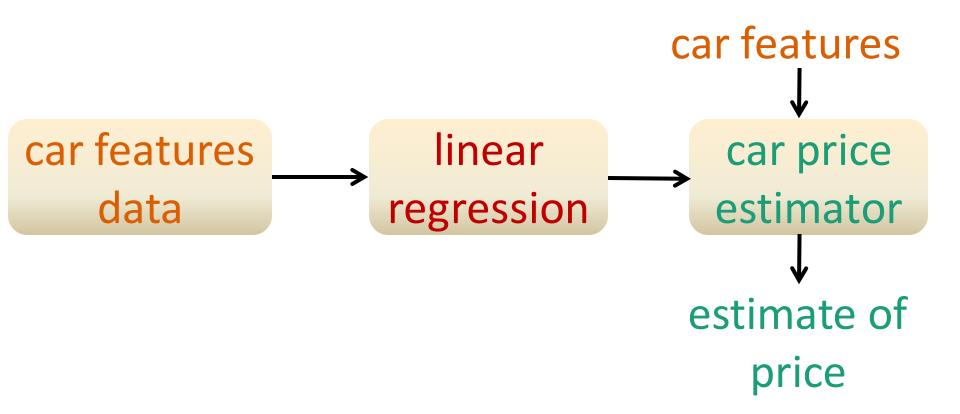
$$m = \{ w_1, w_2, ... \}$$

 $P[m \mid spam] = P[w_1 \mid spam] \times P[w_2 \mid spam] \times ...$

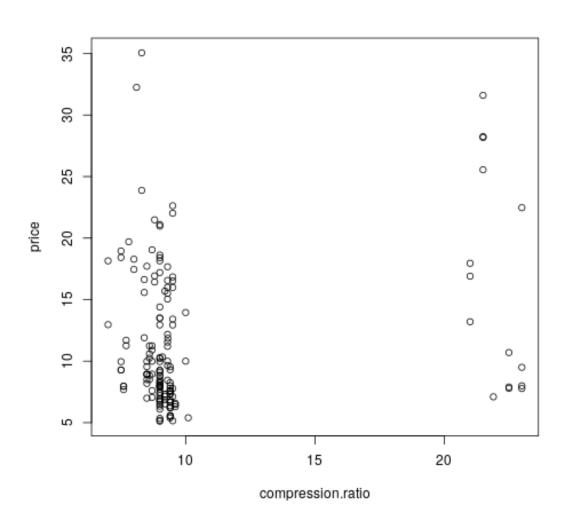
 $P[w_k \mid spam] = frequency of word w_k in spam message corpus$

Prediction

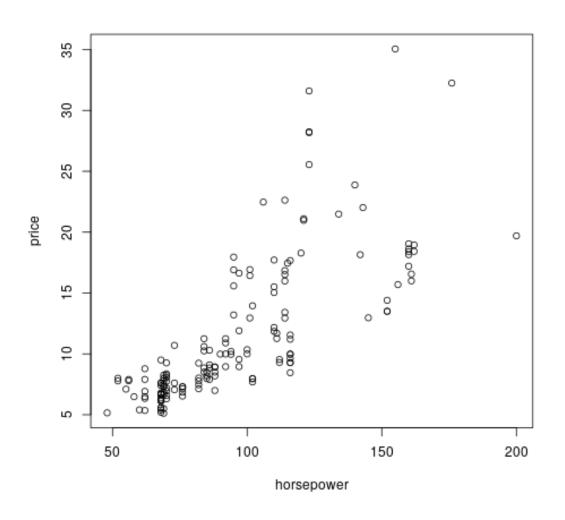
Car price estimator



Feature: compression ratio



Feature: horsepower



Predicting price from horsepower

Find a and b that fit the data as well as possible.

Coefficients and interpretation

coefficient	estimate
intercept a	-2.5
slope b	0.15

- Slope: 100 hp of additional power cost around \$ 15'000
- Intercept: a car with 0 hp costs \$ -2'500?!
- Intercept after centering: price of an average-power car

Probabilistic interpretation of linear regression

$$y = a + bx + e$$

where $e \sim \text{Normal}(0, v^2)$
mean variance

measurement errors, imprecisions in the model, ...

Confidence in the fitted model

coefficient	estimate	standard error
intercept a	-2.5	1
slope b	0.15	0.001

Under the probabilistic interpretation, the value of each coefficient follows a

Normal(estimate, standard error²)

mean variance

How good is a fit?

The residuals measure how close a regression line is to the actual data:

residual
$$r_i$$
 = actual output – predicted
= $y^{(i)}$ – $(a + b x^{(i)})$

Predicting price from two features

two predictors

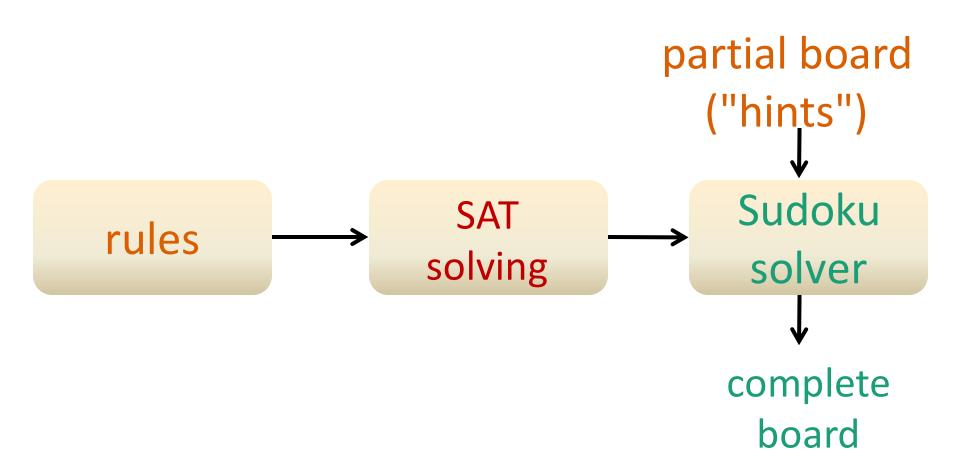
price = $a + b_1 \times hp + b_2 \times engine_size + e$

categorical (binary) variable

price =
$$a + b_1 \times hp + b_2 \times aspiration + e$$

Synthesis

Sudoku solver



Sudoku: rules

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
8 4 7			8		3			1
7				2				1 6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	თ	4	8
1	9	8	ന	4	2	15	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	ഠ	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Brute force solver

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
8 4 7			8		3			1
7				2				1 6
	6					2	8	
			4	1	9			5
				8			7	9

for every row x
for every column y
assign a value in {1, 2, ..., 9} to cell (x, y)
check that the hints are matched
check that there are no repeated values in rows
check that there are no repeated values in columns
check that there are no repeated values in blocks

Brute force solver

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
8 4 7			8		3			1
7				2				1 6
	6					2	8	
			4	1	9			5
				8			7	9

for every row x
for every column y
assign a value in {1, 2, ..., 9} to cell (x, y)
check that the hints are matched
check that there are no repeated values in rows
check that there are no repeated values in columns
check that there are no repeated values in blocks

enumerates (9!)⁹ > 10⁵⁰ boards!

From constraints to solutions

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
8 4 7			8		3			1
7				2				6
	6					2	8	
			4	1	9			5 9
				8			7	9

Find a solution such that: ←

SAT solvers can do that!

- the hints are matched
- there are no repeated values in rows
- there are no repeated values in columns
- there are no repeated values in blocks

Propositional logic

Propositional logic (also Boolean logic) is a simple yet powerful language to describe constraints.

- Propositions are Boolean variables whose value denotes whether a fact is True or False
- Connectives (operators) compose propositions to form more complex constraints

Connectives

Formula	Meaning
A	A is true
¬A ~A !A	A is not true
A ^ B A & B	Both A and B are true
AVB A B	A or B (or both) are true
$A \Rightarrow B$	A implies B: ¬A V B

Truth tables

A truth table enumerates the Boolean values of a formula for all possible combination of values of its propositions.

А	В	ΑΛΒ
Т	Т	Т
Т	F	F
F	Т	F
F	F	F

Truth tables

A truth table enumerates the Boolean values of a formula for all possible combination of values of its propositions.

А	В	A V B
Т	Т	
Т	F	
F	Т	
F	F	

Truth tables

A truth table enumerates the Boolean values of a formula for all possible combination of values of its propositions.

А	В	AVB
Т	Т	Т
Т	F	Т
F	Т	Т
F	F	F

Conjunctive Normal Form (CNF)

A formula is in CNF when it has the structure:

$$D_1 \wedge D_2 \wedge ... \wedge D_n$$

where each D_k has the structure:

$$C_1 \vee C_2 \vee ... \vee C_m$$

and each C_h is

either a proposition or a negated proposition.

Every propositional formula can be turned into an equivalent one that is in CNF.

CNF: examples

- A
- ¬A ∧ B
- A V B
- A A B A C
- (A ∨ B) ∧ C ∧ (D ∨ ¬F)
- ¬A ∨ (B ∧ C) not CNF
 - Equivalent to $(\neg A \lor B) \land (\neg A \lor C)$

Sudoku grid with propositions

Proposition P_{x,y,v} is true
means
cell at row x, column y has value v

- We define the constraints as collections of disjunctions.
- The overall SAT input formula is the conjunction of these many disjunctions, and hence is in CNF.

Each cell has a value

Each cell has at least one value

for all x, y:
$$P_{x,y,1} \vee P_{x,y,2} \vee ... \vee P_{x,y,9}$$

each cell has at most one value

for all x, y: for all pairs v1, v2:
$$\neg P_{x,y,v1} \lor \neg P_{x,y,v2}$$

Each value appears in a row once

for all rows x: for all values v: for all pairs of columns y1, y2: $\neg P_{x,y1,v} \lor \neg P_{x,y2,v}$

Each value appears in a column once

for all columns y: for all values v: for all pairs of rows x1, x2: $\neg P_{x1,y,v} \lor \neg P_{x2,y,v}$

Each value appears in a block once

for all blocks b: for all values v: for all pairs of cells (x1, y1) and (x2, y2) in b: $\neg P_{x1,v1,v} \lor \neg P_{x2,v2,v}$

These slides' license

© 2019 Carlo A. Furia



Except where otherwise noted, this work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-nd/4.0/