What is Machine Learning?

1) A long list of methods/algorithms for different data analysis problems
   - in sciences
   - in commerce

2) Frameworks to develop your own learning algorithm/method

3) Machine Learning = model formulation + optimization
What is Machine Learning?

- Pedro Domingos: *A Few Useful Things to Know about Machine Learning*

  learning = representation + evaluation + optimization

- “Representation”: Choice of model, choice of hypothesis space
- “Evaluation”: Choice of objective function, optimality principle

  Notes: The *prior* is both, a choice of representation and, usually, a part of the objective function.
  In Bayesian settings, the choice of model often directly implies also the “objective function”

- “Optimization”: The algorithm to compute/approximate the best model
Pedro Domingos: *A Few Useful Things to Know about Machine Learning*

- It’s generalization that counts
  - Data alone is not enough
  - Overfitting has many faces
  - Intuition fails in high dimensions
  - Theoretical guarantees are not what they seem

- Feature engineering is the key
- More data beats a cleverer algorithm
- Learn many models, not just one
- Simplicity does not imply accuracy
- Representable does not imply learnable
- Correlation does not imply causation
What is Machine Learning?

- In large parts, ML is: (let’s call this ML$^0$)
  
  *Fitting a function* $f : x \mapsto y$ *to given data* $D = \{(x_i, y_i)\}_{i=1}^n$
What is Machine Learning?

• In large parts, ML is: (let’s call this ML\(^0\))

\[
\text{Fitting a function } f : x \mapsto y \text{ to given data } D = \{(x_i, y_i)\}_{i=1}^n
\]

• Why is function fitting so omnipresent?
What is Machine Learning?

- In large parts, ML is: (let’s call this ML$^0$)
  
  \[ \text{Fitting a function } f : x \mapsto y \text{ to given data } D = \{(x_i, y_i)\}_{i=1}^{n} \]

- Why is function fitting so omnipresent?
  - Dynamics, behavior, decisions, control, predictions – are all about functions
  - Thinking? (i.e., planning, optimization, (logical) inference, CSP solving, etc?)
  - In the latter case, algorithms often provide the scaffolding, ML the heuristics to accelerate/scale them. (E.g., an evaluation function within a MCTS planning algorithm.)
ML^0 objective: Empirical Risk Minimization

- We have a hypothesis space \( \mathcal{H} \) of functions \( f : x \mapsto y \)
  In a standard parameteric case \( \mathcal{H} = \{ f_\theta \mid \theta \in \mathbb{R}^n \} \) are functions \( f_\theta : x \mapsto y \) that are described by \( n \) parameters \( \theta \in \mathbb{R}^n \)

- Given data \( D = \{(x_i, y_i)\}_{i=1}^n \), the standard objective is to minimize the “error” on the data

\[
 f^* \arg\min_{f \in \mathcal{H}} \sum_{i=1}^n \ell(f(x_i), y_i),
\]

where \( \ell(\hat{y}, y) > 0 \) penalizes a discrepancy between a model output \( \hat{y} \) and the data \( y \).

- Squared error \( \ell(\hat{y}, y) = (\hat{y} - y)^2 \)
- Classification error \( \ell(\hat{y}, y) = [\hat{y} \neq y] \)
- neg-log likelihood \( \ell(\hat{y}, y) = -\log p(y \mid \hat{y}) \)
- etc
What is Machine Learning beyond ML$^0$?

● Fitting more structured models to data, which includes
  – Time series, recurrent processes
  – Graphical Models
  – Unsupervised learning (semi-supervised learning)

...but in all these cases, the scenario is still not interactive, the data $D$ is static, the decision is about picking a single model $f$ from a hypothesis space, and the objective is a loss based on $f$ and $D$ only.

● Active Learning, where the “ML agent” makes decisions about what data label to query next

● Bandits, Reinforcement Learning, manipulating the domain (and thereby data source)
Machine Learning is everywhere

NSA, Amazon, Google, Zalando, Trading, ...
Chemistry, Biology, Physics, ...
Control, Operations Research, Scheduling, ...
Face recognition

keypoints

(e.g., Viola & Jones)
Hand-written digit recognition (US postal data)

(e.g., Yann LeCun)
Gene annotation

(e.g., Gunnar Rätsch, mGene Project)
Speech recognition

HMM phone models

Lexicon

Sentence model: 'he is new'
# Spam filters

<table>
<thead>
<tr>
<th></th>
<th>george</th>
<th>you</th>
<th>your</th>
<th>hp</th>
<th>free</th>
<th>hpl</th>
<th>!</th>
<th>our</th>
<th>re</th>
<th>edu</th>
<th>remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>spam</td>
<td>0.00</td>
<td>2.26</td>
<td>1.38</td>
<td>0.02</td>
<td>0.52</td>
<td>0.01</td>
<td>0.51</td>
<td>0.51</td>
<td>0.13</td>
<td>0.01</td>
<td>0.28</td>
</tr>
<tr>
<td>email</td>
<td>1.27</td>
<td>1.27</td>
<td>0.44</td>
<td>0.90</td>
<td>0.07</td>
<td>0.43</td>
<td>0.11</td>
<td>0.18</td>
<td>0.42</td>
<td>0.29</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Machine Learning became an important technology in science as well

(Stuttgart Cluster of Excellence “Data-integrated Simulation Science (SimTech)”)

Organization of this lecture

See TOC of last year’s slide collection

- Part 1: The Core: Regression & Classification
- Part 2: The Breadth of ML methods
- Part 3: Bayesian Methods
Is this a theoretical or practical course?

Neither alone.
Is this a theoretical or practical course?

Neither alone.

- The goal is to teach how to design good learning algorithms
  
  data
  ↓
  modelling [requires theory & practise]
  ↓
  algorithms [requires practise & theory]
  ↓
  testing, problem identification, restart
How much math do you need?

• Let $L(x) = \| y - Ax \|^2$. What is

$$\arg\min_x L(x)$$

• Find

$$\min_x \| y - Ax \|^2 \quad \text{s.t.} \quad x_i \leq 1$$

• Given a discriminative function $f(x, y)$ we define

$$p(y \mid x) = \frac{e^{f(y, x)}}{\sum_{y'} e^{f(y', x)}}$$

• Let $A$ be the covariance matrix of a Gaussian. What does the Singular Value Decomposition $A = V D V^\top$ tell us?
How much coding do you need?

- A core subject of this lecture: learning to go from principles (math) to code

- Many exercises will implement algorithms we derived in the lecture and collect experience on small data sets

- Choice of language is fully free. I support C++; tutors might prefer Python; Octave/Matlab or R is also a good choice.

(this course will not go to the full depth in math of Hastie et al.)
Bishop, C. M.: *Pattern Recognition and Machine Learning.*
Springer, 2006
http://research.microsoft.com/en-us/um/people/cmbishop/prml/ (some chapters are fully online)
Books & Readings

• more recently:
  – David Barber: Bayesian Reasoning and Machine Learning
  – Kevin Murphy: Machine learning: a Probabilistic Perspective

• See the readings at the bottom of:
  http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/index.html#readings
Organization

- Course Webpage:
  - Slides, Exercises & Software (C++)
  - Links to books and other resources

- Admin things, please first ask:
  Carola Stahl, [Carola.Stahl@ipvs.uni-stuttgart.de](mailto:Carola.Stahl@ipvs.uni-stuttgart.de), Raum 2.217

- Rules for the tutorials:
  - Doing the exercises is crucial!
  - **Nur Votieraufgaben.** At the beginning of each tutorial:
    - sign into a list
    - vote on exercises you have (successfully) worked on
  - Students are randomly selected to present their solutions
  - You need 50% of completed exercises to be allowed to the exam
  - Please check 2 weeks before the end of the term, if you can take the exam