Fundamentals of Machine Learning:

Introduction and Basic Concepts

Prof. Andrew D. Bagdanov (`andrew.bagdanov AT unifi.it`)

UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Outline

Introduction

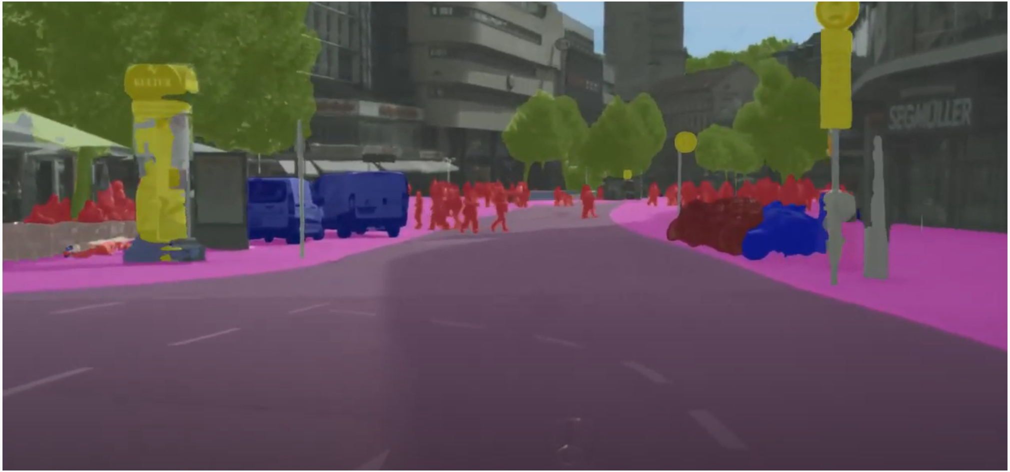
Lecture Objectives

At the end of this lecture you will:

- Have developed basic intuitions about what Machine Learning is.
- Understand how your mastery of course topics will be measured in the final exam.
- Understand the Empirical Risk Minimization formulation of learning.
- Have acquired basic intuitions about the main components of the risk minimization approach and what they mean.

The world of ”tomorrow”

- Link to video



What is Machine Learning?

> *"I've studied all available charts of the planets and stars and none of them match the others. There are just as many measurements and methods as there are astronomers and all of them disagree. What's needed is a long term project with the aim of mapping the heavens conducted from a single location over a period of several years."*
>
> *– Tycho Brahe, 1563 (age 17).*

- The term Machine Learning dates back to Arthur Samuel in the 1950s.
- In the intervening years its scope has expanded and contracted.
- One way of thinking of machine learning is:

$$\begin{aligned}\text{Machine Learning} &= (\text{Computational Statistics} + \text{Optimization}) \\ &\quad + \text{Data (usually lots of it)}\end{aligned}$$

What is Machine Learning?

> *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*
>
> *– Tom Mitchell, 1987*

- This definition is similar to (modern) definitions of Artificial Intelligence.
- That is, it is operational instead of cognitive.
- Alan Turing started this trend by changing the question from "Can machines think?" to "Can machines do what we (as thinking entities) can do?".

Supervised versus unsupervised learning

- Machine Learning is (very loosely) divided into two macro categories of learning approaches:
  - Supervised Learning: sometimes called “learning from a teacher” refers to a class of approaches that aim to learn to predict outputs from*inputs* from a dataset of paired inputs and outputs.
  - Unsupervised Learning: which instead tries to learn “something” from unlabeled input data – that is, without any pre-specified labels.
- In this introductory course we will limit ourselves mostly to supervised learning.
- Note that there is, in fact, a complete spectrum of supervision regimes: supervised, weakly-supervised, semi-supervised, self-supervised, unsuwpervised.

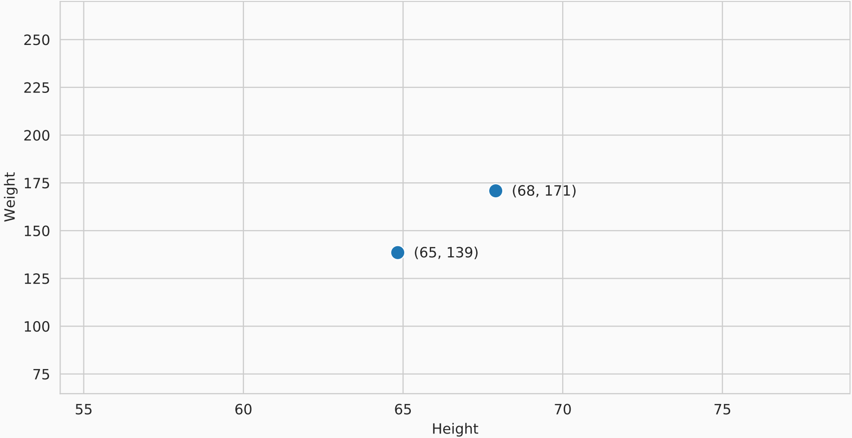
(More on unsupervised learning in the Data Mining course.)

Supervised learning: an example

- Say we are analyzing the correlation between height and weight.
- (Aside: we will often use synthetic examples of this type to illustrate key concepts and techniques.)
- And let's say that we have only two data points:
$$(67.9, 170.85) \text{ and } (61.9, 122.5).$$
- Ideally, we wish to infer a relation between height and weight that explains the data.
- A good first step us usually to visualize.

Supervised learning: an example

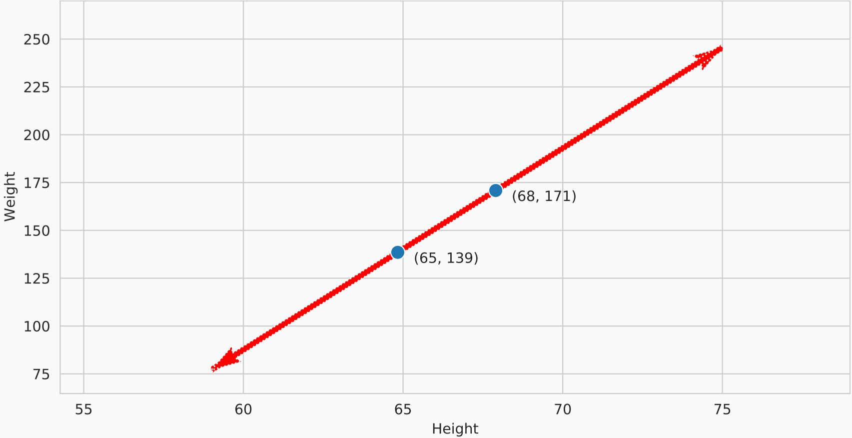
- So, we have a situation like this...
- What can we do?



Supervised learning: an example

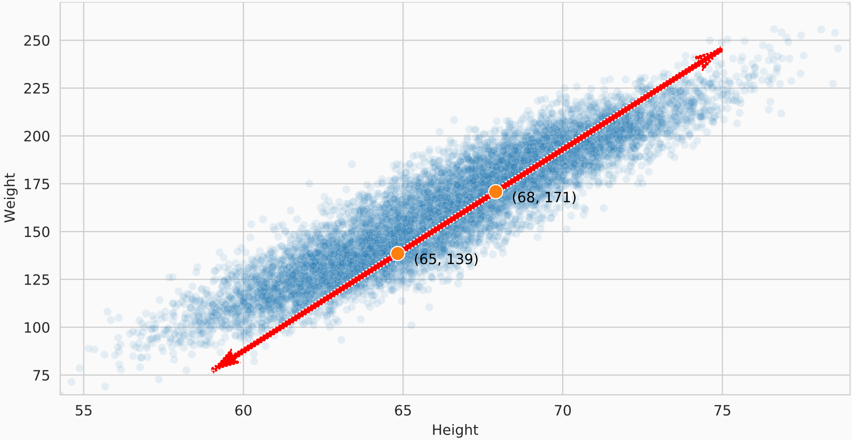
- Well, some grade-school algebra lets us connect the dots:

$$y = 8.013x - 373.247 \text{ (why this model?)}$$



Supervised learning: an example

- Now lets say that we have a lot more data.
- Does our "model" generalize?



Course Organization and Objectives

Prerequisites (Math)

- This course requires a degree of mathematical maturity, as well as a mastery of the basics of programming, algorithms, and data structures.
- The Key Math Concepts you should know include:
  - What is the rank of a matrix. What does it mean to be rank-deficient?
  - What are the eigenvalues and eigenvectors of matrix? What do they mean?
  - What is the gradient of a multivariate function? In what direction does it point?
  - What is a scalar (dot) product between two vectors?
  - What is the norm of a vector? What does it have to do with the scalar product?

Prerequisites (Statistics and Probability Theory)

- The Key Statistics Concepts you should know include:
  - What is the Sum Rule of probability? What about the Product Rule?
  - What do these rules have to do with Joint, Conditional and Marginal probability density functions?
  - What is Bayes Rule? What does it mean and how can we derive it?
  - What is a Random Variable? What is the Expected Value of a (function of a) random variable?

Prerequisites (Computer Science)

- The key skills from Computer Science are harder to list:
    - You must have a working mastery of at least one high-level programming language (preferably a dynamic one like Python, R, Matlab, or Lisp).
    - The laboratory sessions will be done using Python and its excellent ecosystem for numerical programming.
    - Please use This Programming Skills Self-assessment to verify that your knowledge and skillset is at least somewhere between levels A2 and B1.

Organization

This course is on the Fundamentals of Machine Learning, and in it we will cover:

- Foundations of the Foundations: probability theory and statistics for machine learning, probability distributions, basics of information theory, Bayesian versus frequentist interpretations, linear models for regression, linear models for classification, the bias-variance decomposition, overfitting and underfitting, model regularization, probabilistic generative models, probabilistic discriminative models, Maximum Likelihood Estimation (MLE), Maximum a Posteriori (MAP) inference, Bayesian inference.
- Machine Learning: Support Vector Machines (SVMs), kernel machines, graphical models, decision trees, ensemble methods, boosting, bagging, Bayesian model averaging, random forests, Expectation Maximization (EM), mixture density estimation.

Organization

- Deep Learning: connectionist models, Hebbian learning rules, the perceptron, neural networks, Stochastic Gradient Descent (SGD), the Backpropagation algorithm, the Multilayer Perceptron (MLP), vanishing and exploding gradients, model size and regularization, network regularization.
- Special Topics and Applications: Long Short-term Memory Networks (LSTMs), natural language processing and language models (Transformers), Convolutional Neural Networks (CNNs), self-supervised learning, continual learning, domain adaptation, transfer learning.
- Tools, Techniques, and Best Practices: numerical programming, visualization, model diagnostics and monitoring training, scikit-learn, PyTorch.

A Rough Timeline of the Material

Part I: Classical Machine Learning

- Preliminaries: The math, fundamental concepts, notation, and useful techniques.
- Linear Models: ”Simple” models for regression and classification, geometric and probabilistic interpretations, generative and discriminative models, regularization and prior knowledge.
- Kernel Methods: The linear Support Vector Machine (SVM), non linearly separable problems and relaxation, the kernel trick and nonlinear SVMs.
- Local Methods: Nearest Neighbor methods, nonparametric density estimation.
- Unsupervised Learning: Principal Component Analysis (PCA), Gaussian Mixture Models (GMMs), the Expectation-Maximization (EM) algorithm.

Part II: Deep Learning

- Connectionist models: History, Hebbian learning, the Perceptron and gradient-based learning.
- Deep Networks: Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs).
- Advanced Topics: Long Short-Term Memory Networks (LSTMs), Transfer learning, Self-supervised learning, Transformers.

Laboratories

- This year there will be four (4) laboratory sessions focusing on key topics:
  - Linear models for regression and classification
  - Unsupervised learning
  - Deep Learning I
  - Deep Learning II
- In the laboratory sessions (always on Tuesdays) we will work together on a set of problems.
- A week before I will publish the labs and a short tutorial on getting started.
- You will submit your *individual* solutions to the labs after 7-10 days – there will be a small bonus for submitting lab solutions on time.
- Only the *top three (3) laboratory grades* will be used (i.e. you can skip one).

Student Evaluation

- The final evaluation is based on several components:

Type	Component	Weight
Mandatory	Laboratory/Homeworks	1/3 (+ 1/30 on-time bonus)
Optional	Written Midterm Exam	1/3
Mandatory	Written Final Exam	1/3 (or 2/3)
Optional	Oral Final Exam	$\pm$1/30 (for *cum laude*)

- Important: The midterm exam grade is valid for the first four final exams following the end of the course (i.e. until and including the Easter exam).
- Important: If you use the midterm exam grade, the final exam will cover only the second half of the course. Otherwise, the final is comprehensive.

High Level Objectives

- Machine Learning is a broad and very active field.
- At a basic level, it is about learning from training data to make inferences about new data.
- This rather vague description already articulates key concepts we will study in detail.
- In order to employ Machine Learning in practice, we need to familiarize ourselves with some theoretical machinery.
- This machinery will help us model learning problems, evaluate performance of learning systems, and quantify belief in our solutions.

Objectives: Theory and Practice

Let's take a step back and think about more abstract objectives:

- In this course we will take a relatively deep look at several fundamental Machine Learning theories.
- Theory is important, but it isn't everything.
- Probably 95% of the time you don't explicitly need sophisticated theory.
- However, for that last 5% it suddenly becomes indispensable – especially when trying to understand why things don't work as expected.
- So, don't worry if you don't grasp every intuition or derivation from the abbreviated versions here.
- Drink it in, build intuition about the theory that will inform your practice.

Good books

- Three great ML books (and an authoritative deep learning book):
  - C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
  - D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
  - Zhang et al., *Dive into Deep Learning*, 2023.
- The first is a classic text on ML, and is the basis of the treatment (and many of the plots) I use here.
- The Bishop Book is also freely available (2006 edition) online at This Link
- The second book is full of excellent insights into the relationships between optimization, information theory and Bayesian inference.
- The third is a great online resource for all types of deep learning.

But... That sounds like a *lot* of work.

- Correct. In this course I ask many different things to meet our learning objectives.
- But, let's break it down:
  - 25 hours/cfu × 6 cfu = 150 total hours
  - 8 lecture hours in aula/cfu × 6 cfu = 48 total hours in aula
  - Leaving: 102 total independent study hours for labs and exam prep
- However,
  - 10-15 laboratory hours in aula.
  - On-time lab submission contributes 33+% to final grade.
  - Plus, 4-6 hours of exam preparation sessione (esercitazioni).
- The point: Course organization designed to help you succeed, and to *succeed on time*.

Good news and bad news (mostly good, really)

- Everyone take a deep breath before reading on:

Good news and bad news (mostly good, really)

- Everyone take a deep breath before reading on:

Your days of learning in structured learning environments are (almost) over.

What is expected of you (and of your professor (me))

- Has demonstrable knowledge and insight, based on the knowledge and insight at the level of Bachelor and which exceed and/or deepen it, as well as providing a basis or an opportunity to make an original contribution to the development and/or application of ideas, often in the context of research.
- Is able to apply knowledge, insight and problem-solving skills in new or unknown circumstances within a broader (or multidisciplinary) context related to the field of expertise; is able to integrate knowledge and deal with complex matter.
- Is able to formulate judgments on the basis of incomplete or limited information, taking into account the social and ethical responsibilities associated with the application of his or her own knowledge and judgments.
- Is able to communicate conclusions, as well as the knowledge, motives and considerations that underlie them, clearly and unambiguously to an audience of specialists or non-specialists.
- Possesses the learning skills that enable him or her to enter into a follow-up study with a largely self-directed or autonomous character.

Generalizing our Intuition

Machine Learning in a (mathematically dense) nutshell

The ingredients:

- An input space $\mathcal{X}$ (often $\mathbb{R}^m$) and an output space $\mathcal{Y}$ (often $\mathbb{R}^n$).
- A generative assumption of a function $h : \mathcal{X} \to \mathcal{Y}$ (often $\mathbf{y} = h(\mathbf{x}) + \varepsilon$).
- As unknown joint probability density $p(\mathbf{x}, \mathbf{y})$ over $\mathcal{X}$ and $\mathcal{Y}$.
- A hypothesis space $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$.
- A loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

A learning objective:

- Assuming the true $h \in \mathcal{H}$, we can just:

$$
\begin{aligned}
h^* &= \arg\min_{h \in \mathcal{H}} \mathbb{E}_p[\mathcal{L}(h(\mathbf{x}), \mathbf{y})] \\
&= \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(x), y) p(\mathbf{x}, \mathbf{y}) dxdy
\end{aligned}
$$

Are we done?

Can we go home?

$$h^* = \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

Are we done?

Can we go home?

$$h^* = \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- We can not go home yet. Let's start with the big unknown p...

Are we done?

Can we go home?

$$h^* \quad = \quad \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- We can not go home yet. Let's start with the big unknown p...
- With no information about p we must resort to sampling:

$$(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}, \text{ for } i \in \{1, \ldots, N\}$$

- Important: $(\mathbf{x}_i, \mathbf{y}_i) \sim p(\mathbf{x}, \mathbf{y})$

Are we done?

Can we go home?

$$h^* \quad = \quad \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) dx dy$$

- We can not go home yet. Let's start with the big unknown p...
- With no information about p we must resort to sampling:

$$(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}, \text{ for } i \in \{1, \ldots, N\}$$

- Important: $(\mathbf{x}_i, \mathbf{y}_i) \sim p(\mathbf{x}, \mathbf{y})$
- We can then approximate the objective with the empirical expected loss:

$$h^* \quad = \quad \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i)$$

Are we done now?

Cool. Now can we go home?

$$
\begin{aligned}
h^* &= \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\
&\approx \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i)
\end{aligned}
$$

Are we done now?

Cool. Now can we go home?

$$
\begin{aligned}
h^* &= \arg\min_{h\in\mathcal{H}} \int \mathcal{L}(h(\mathbf{x}),\mathbf{y})p(\mathbf{x},\mathbf{y})dxdy \\
&\approx \arg\min_{h\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i),\mathbf{y}_i)
\end{aligned}
$$

- We can not.

Are we done now?

Cool. Now can we go home?

$$
\begin{aligned}
h^* &= \operatorname*{arg\,min}_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\
&\approx \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i)
\end{aligned}
$$

- We can not.
- What about $\mathcal{L}$?

Are we done now?

Cool. Now can we go home?

$$
\begin{aligned}
h^* &= \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\
&\approx \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i)
\end{aligned}
$$

- We can not.
- What about $\mathcal{L}$?
- What about $\mathcal{H}$?

Are we done now?

Cool. Now can we go home?

$$\begin{aligned} h^* &= \arg\min_{h \in \mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &\approx \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i) \end{aligned}$$

- We can not.
- What about $\mathcal{L}$?
- What about $\mathcal{H}$?
- What about that scary minimization?

Are we done now?

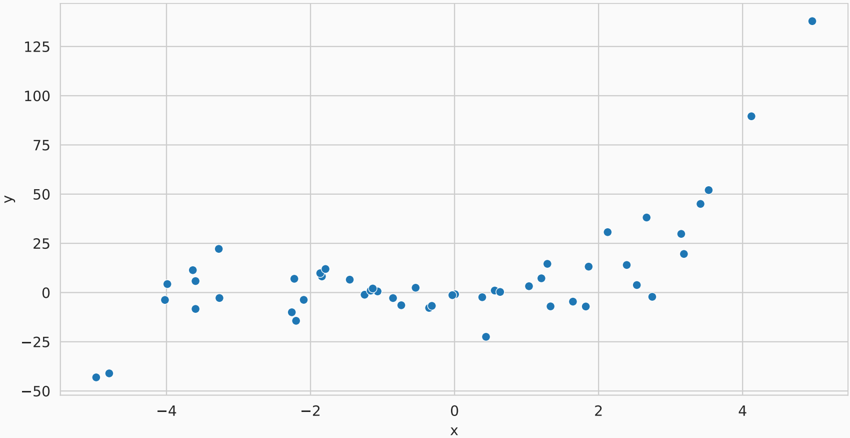
Cool. Now can we go home?

$$\begin{aligned} h^* &= \arg\min_{h\in\mathcal{H}} \int \mathcal{L}(h(\mathbf{x}), \mathbf{y}) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &\approx \arg\min_{h\in\mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i) \end{aligned}$$

- We can not.
- What about $\mathcal{L}$?
- What about $\mathcal{H}$?
- What about that scary minimization?
- Finally, what about $\mathcal{X}$ and $\mathcal{Y}$?

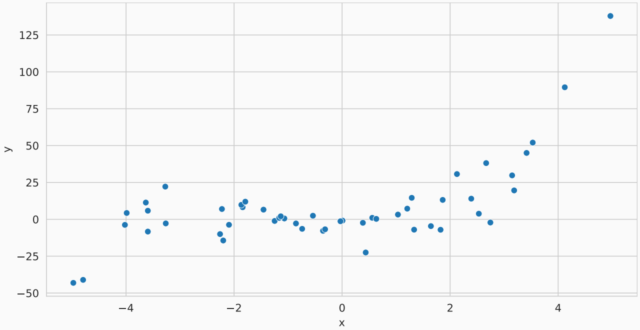
Generalizing

- Back to the simple example, but what if we have data distributed like below?
- Process: $y = f(x) + \varepsilon$ (where ε is Gaussian noise).



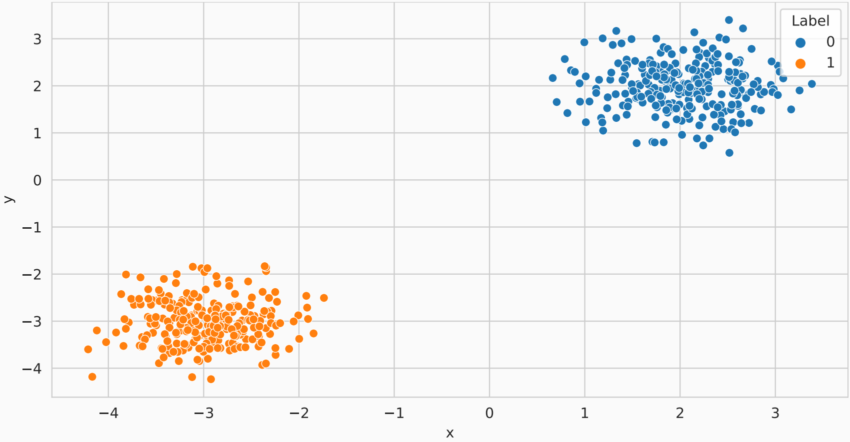
Generalizing

- Our goal is to exploit this training set in order to make predictions.
- That is, to predict the target $\hat{y} = f(\hat{x})$ for new $\hat{x}$.
- In doing this we are implicitly trying to learn what the underlying f is.
- Learning should be independent of ε (which we do not want to capture).



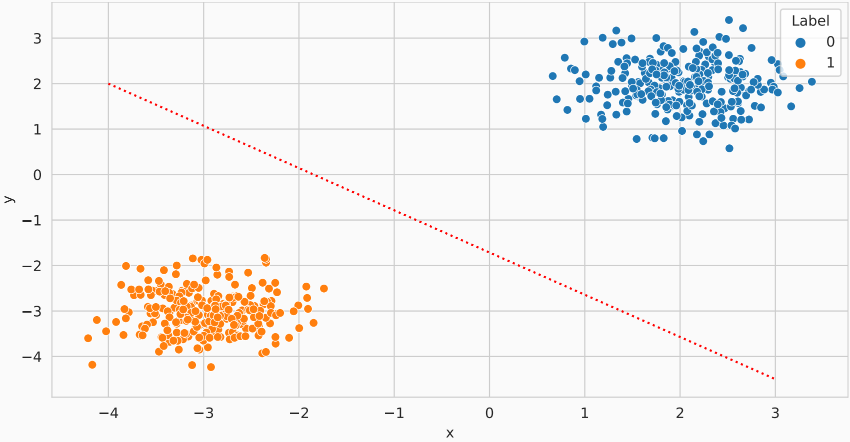
A different sort of problem

- Sometimes we want to understand how data is generated from N sources.
- With the goal of discriminating sources from each other.



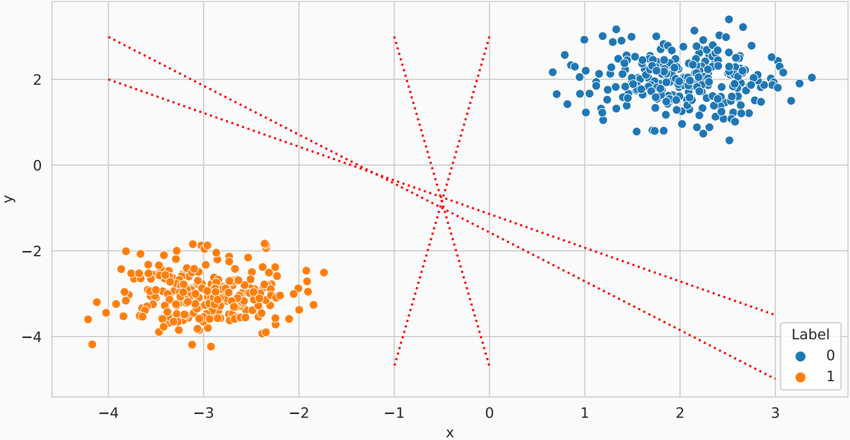
Discriminating

- General idea: find a separating hyperplane.
- That is, one that separates one class from the other.



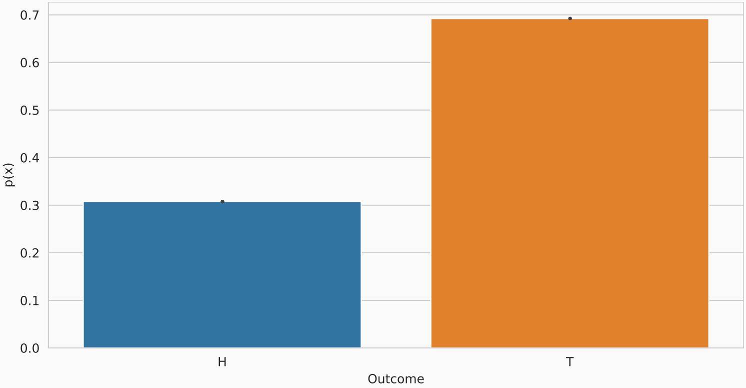
But which one?

- Which one is the "best" discriminant?
- What does best even mean here?



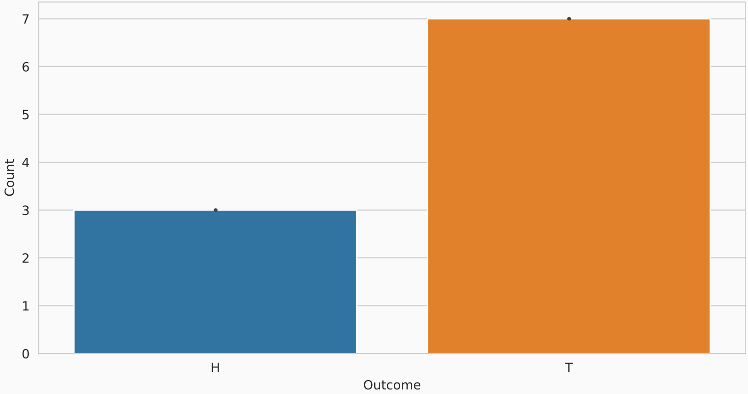
Two views (and the obligatory coin flip example)

- Let's say we have a coin and we want to decide if it is fair or not.
- Someone performed an experiment from which they derived this estimate:



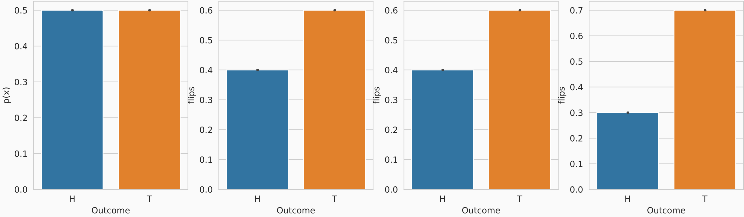
Two views

- What if the data are summarized instead in this way.
- Does this cause you to rethink your inference about the coin?



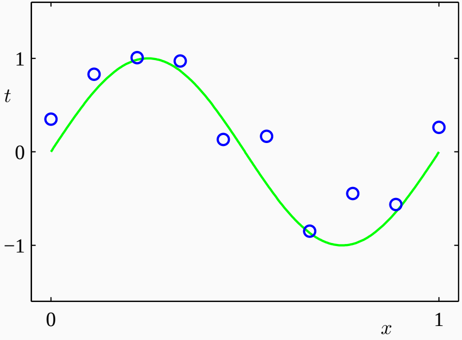
Two views

- When we estimate parameters of a model (sometimes referred to as inference), we need to apply everything we know.
- In particular, we need to be careful to quantify whenever possibile our belief in the accuracy of our inferences.



A motivating example

- Going back to our simple regression problem: we observe a real-valued input variable x and want to predict a real-valued target variable t.
- For the purposes of demonstration, we consider an artificial example of synthetically generated data: $y = f(x|\mathbf{w}) + \varepsilon$.



- We are given a training set of (x, y) pairs sampled from $p(x, y)$.
- Goal: learn the underlying function f that generated this data.
- This way, for unseen $\hat{x}$ we can use $f(\hat{x}|\mathbf{w}^*)$ to predict the target $\hat{y}$.

A motivating example

- Let's model this problem as one of curve fitting, for example using a polynomial model:

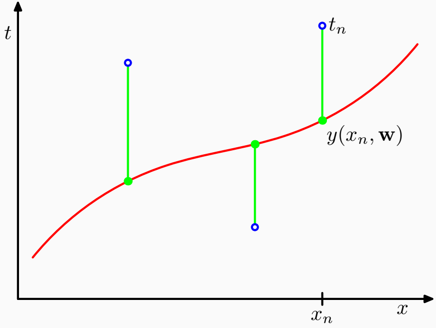
$$
\begin{aligned}
y(x|\mathbf{w}) &= w_0 + w_1 x + w_2 x^2 + \cdots w_M x^M \\
&= \sum_{j=0}^{M} w_j x^j
\end{aligned}
$$

- Note how, even though $y(x|\mathbf{w})$ is a non-linear function of x, it is a linear function in the coefficients $\mathbf{w}$ (i.e. the model parameters).
- By learning we mean estimating the "best" parameters $\mathbf{w}$ from dataset $\mathcal{D} = \{ (x_i, y_i) \mid i = 1, \ldots, N \}$.

A motivating example

- What does good mean in this context?
- Well, we can begin by thinking of measuring the error in the estimated function in terms of the observed data:

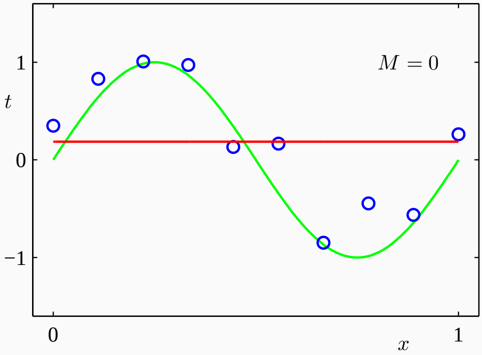
$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{2} \sum_{(x,t)\in\mathcal{D}} \{y(x, \mathbf{w}) - t\}^2$$

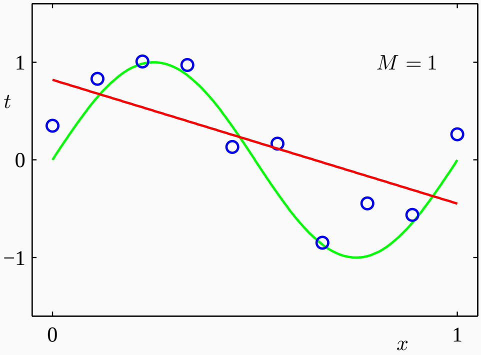


- Which is a quadratic function in $\mathbf{w}$, so its derivatives are linear.
- And $\mathcal{L}(\mathbf{w}|\mathcal{D})$ has a unique minimizer $\mathbf{w}^*$.
- Are we done?

A motivating example

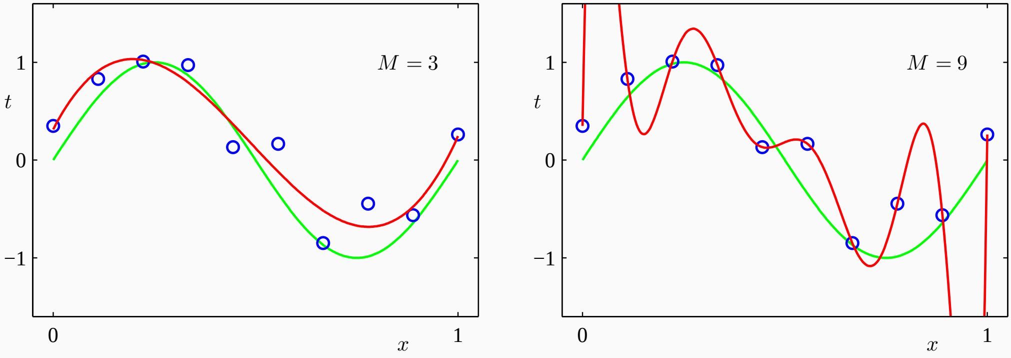
- We are not done. There is one hyperparameter of our model that we have been conveniently forgetting: the order of polynomial M.





A motivating example

- We are not done. There is one hyperparameter of our model that we have been conveniently forgetting: the order of polynomial M.

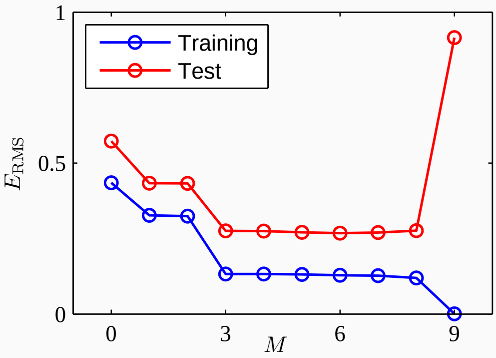


A motivating example

- The remaining problem is model selection, and it is a fundamental element of machine learning. *How might we approach this*?

A motivating example

- The remaining problem is model selection, and it is a fundamental element of machine learning. How might we approach this?
- We gain insight into the underfitting and overfitting by drawing an independent test set and plotting $E_{\mathrm{RMS}} = \sqrt{2\mathcal{L}(\mathbf{w}^*|\mathcal{D})/N}$



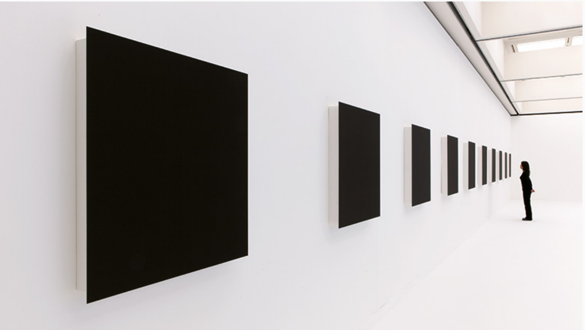
Gen-X Teaches Gen-Y and Gen-Z (about Xs, Ys, and Zs)

A little bit about me

Early years: 80s and 90s (big math)

Math: large cardinals and determinacy

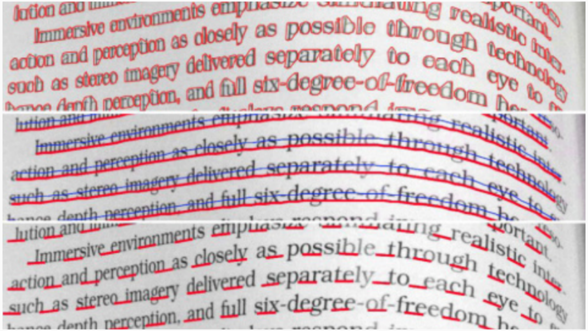
- My first love was mathematics, which I studied at the University of Nevada, Las Vegas (yes, that Las Vegas).
- Specifically, descriptive set theory and the relationship between Large Cardinal Hypotheses, determinacy of simple games on sets of real numbers, and *the consistency of all mathematics*.



Early years: 80s and 90s (image processing)

Math: large cardinals and determinacy

- In parallel, I worked on low-level image estimation problems.
- Specifically, on estimating local and global orientations in scanned document images.
- The novelty at the time, was investigating how to estimate in compressed domains.



The Amsterdam years: early 00s (vision)

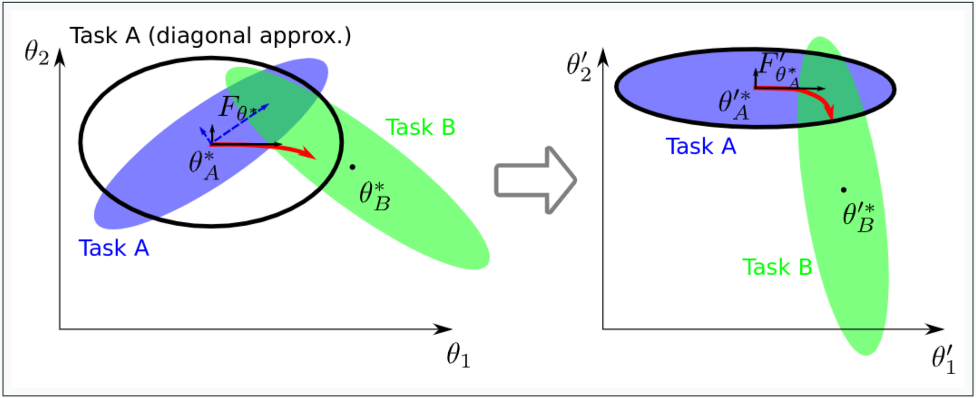
- In 1999 I moved to Europe for a PhD in Multimedia Information Analysis at the *Universiteit van Amsterdam*:
  - Low-level vision: gradient boosting and halftone inversion.
  - Mathematical morphology: granulometric analysis of deep image structure.
  - Graph models: image layout analysis with First-order Gaussian Graphs.
  - Functional programming: formal models of vision programs.

There’s no time, let me sum up...

- 1960s (California): Born, Los Angeles.
- 1970s (Washington): Farm hand, rural Washington.
- 1980s (Las Vegas): High school student; Deadhead; game designer and programmer for Westwood Studios; Emacs user.
- 1990s (Las Vegas): Semi-professional musician; bartender; sports pub bouncer; car counter; math tutor; Math/CS dual Bachelors/Masters student; Senior Network Analysis, US Department of Energy.
- Early 2000s (Amsterdam): PhD, University of Amsterdam; Emacs user; Deadhead.
- Late 2000s (New York/Florence/Rome): Postdoc Renselaer Polytechnic Institute; Deadhead; postdoc University of Florence; Senior Development Chief, Food and Agriculture Organization of United Nations.
- Early 2010s (Florence/Barcelona): Project Leader, Computer Vision Center, Barcelona; Adjunct Professor, *Universitat Autonoma de Barcelona*; Head of Research Unit, MICC, University of Florence, *Ramon y Cajal Fellow*, Computer Vision Center, Barcelona; Emacs user; Deadhead.
- Today: Professor DINFO, University of Florence; Emacs user; Deadhead.

Yes, but what do you *do*, like *now*?

- Right now my research focuses on continual learning problems in computer vision and language:



Also, games!

- With a PhD student (Alessandro Sestini) I also research Deep Reinforcement Learning techniques for building intelligent Non-player Agents (NPAs).

Teaching philosophy and style

- Learning is most effective when it is an interactive give-and-take rather than a passive sit-there-and-listen.
- My job as professor is to put my knowledge at your disposal.
- You job is to suck every last bit of knowledge out of my in these lectures.
- If you don't understand something, interrupt me and ask my to clarify.
- [I know this much parable]

Concluding Remarks

Community Building: The UniFI AI Discord

- We have created a Discord Server to host discussions on Artificial Intelligence.
- There is a dedicated channel for the Fundamentals of Machine Learning (FML) course.
- Please join, and feel free to use this server to share, exchange information, ask for help, and for general chitchat related to ML, AI, datasets, whatever.
- Important: this is a public forum, so be nice, be respectful, and have fun.

https://discord.gg/tUkgrgXdXE

The way forward

- In the next lecture we will cover some (mostly mathematical) preliminaries that will be useful throughout the course.
- More specifically, I will cover some fundamental concepts from linear algebra, statistics and probability, and the important properties of the Gaussian distribution.
- We will also build an intuition about why Machine Learning is hard via an analysis of the Curse of Dimensionality.

> *"Tycho owns the most accurate observations in the world, but he's missing an architect capable of constructing a building starting from his data."*
>
> – *Johannes Keppler*

Reading and Homework Assignments

Reading Assignment:

- Bishop: Chapter 1 (1.1, 1.3, 1.4)

Homework:

1. Familiarize yourself with the UCI Machine Learning Repository of freely available datasets for ML research.
2. Meditate on the coin flip example and think of how we might mitigate the problems discussed during the lecture (i.e. how to cleanly take into account our confidence about our estimate). Hint: Think of it as a sequential estimation problem and use Bayes Rule.