Introduction to Supervised Learning

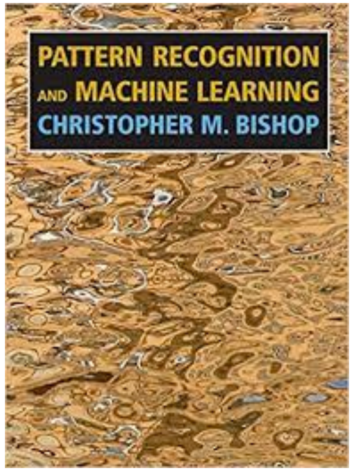
Machine Learning

Daniele Loiacono

POLITECNICO
MILANO 1863

References

- *Pattern Recognition and Machine Learning*, Bishop
  - Chapter 1



What is supervised learning?

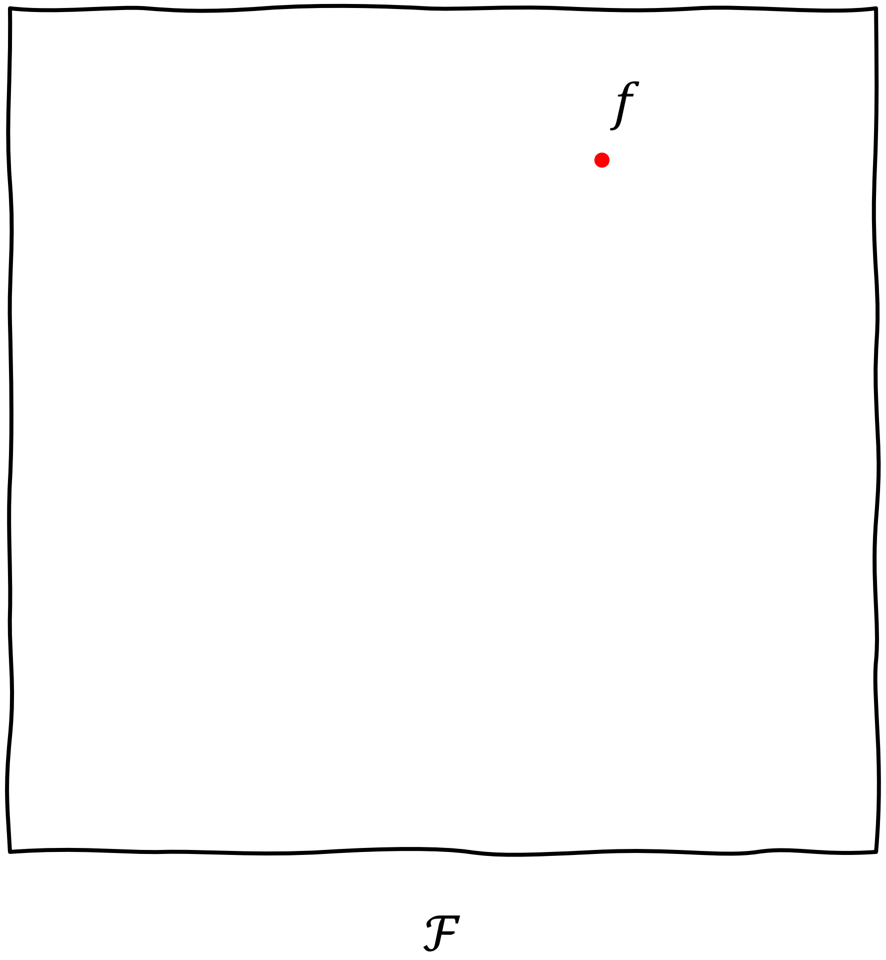
- It is the most popular and well-established learning paradigm
- Data from an unknown function that maps an input x to an output t: $\mathcal{D} = \{\langle x, t \rangle\}$
- Goal: learn a good approximation of f
- Input variables x are usually called **features** or **attributes**
- Output variables t are also called **targets or labels**
- Tasks
  - **Classification** if t is discrete
  - **Regression** if t is continuous
  - **Probability estimation** if t is a probability

When to apply supervised learning?

- When **human cannot perform the task**
  - e.g., DNA analysis
- When human can perform the task but **cannot explain how**
  - e.g., medical image analysis
- When the task **changes over time**
  - e.g., stocks price prediction
- When the the task is **user-specific**
  - e.g., movie recommendation

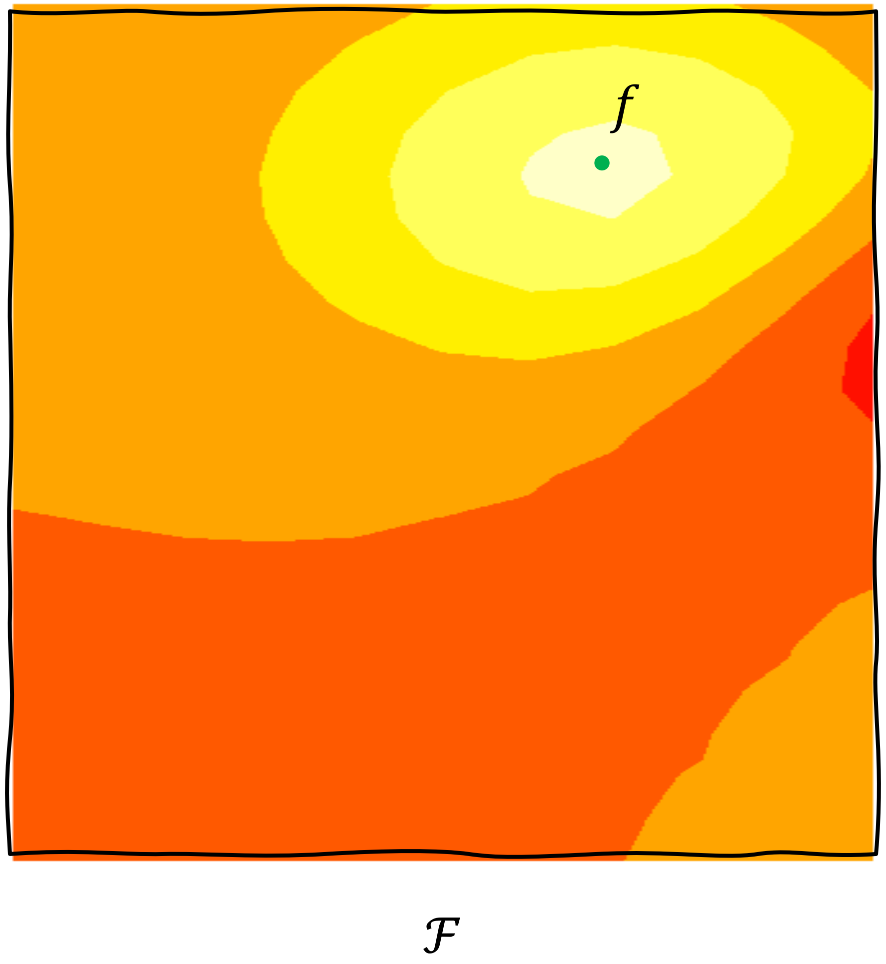
Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are



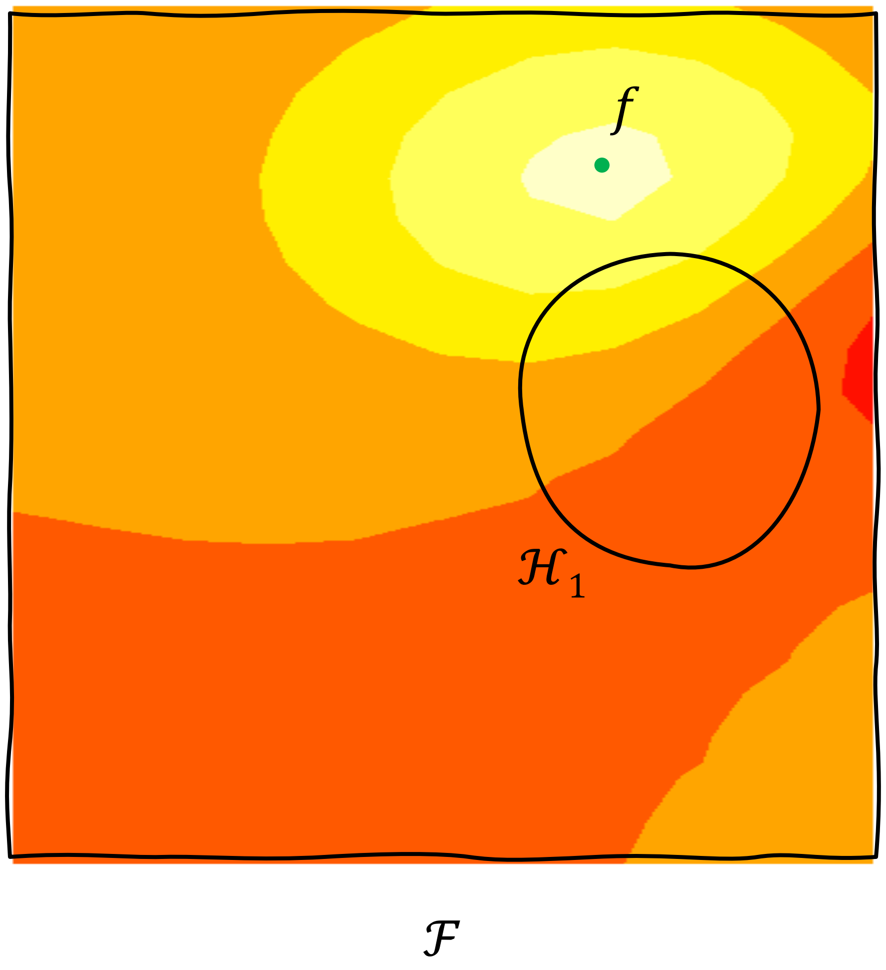
Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - Define a **loss function** $\mathcal{L}$



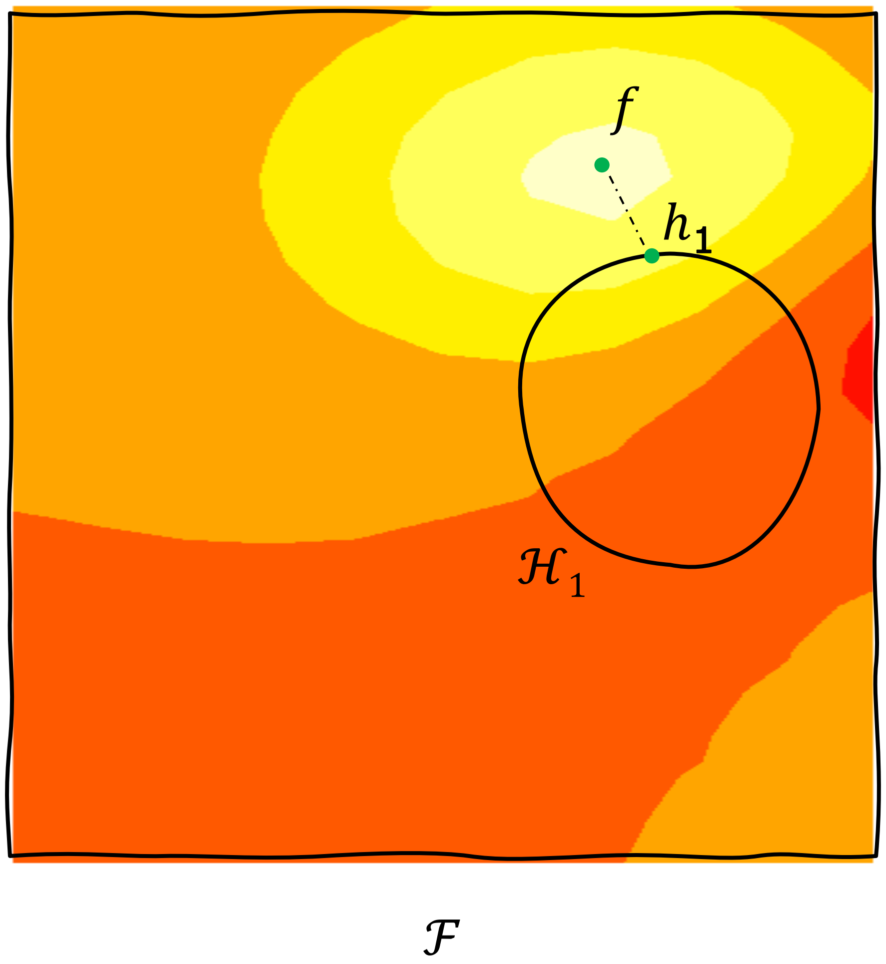
Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - Define a **loss function** $\mathcal{L}$
  - Choose the **hypothesis space** $\mathcal{H}$



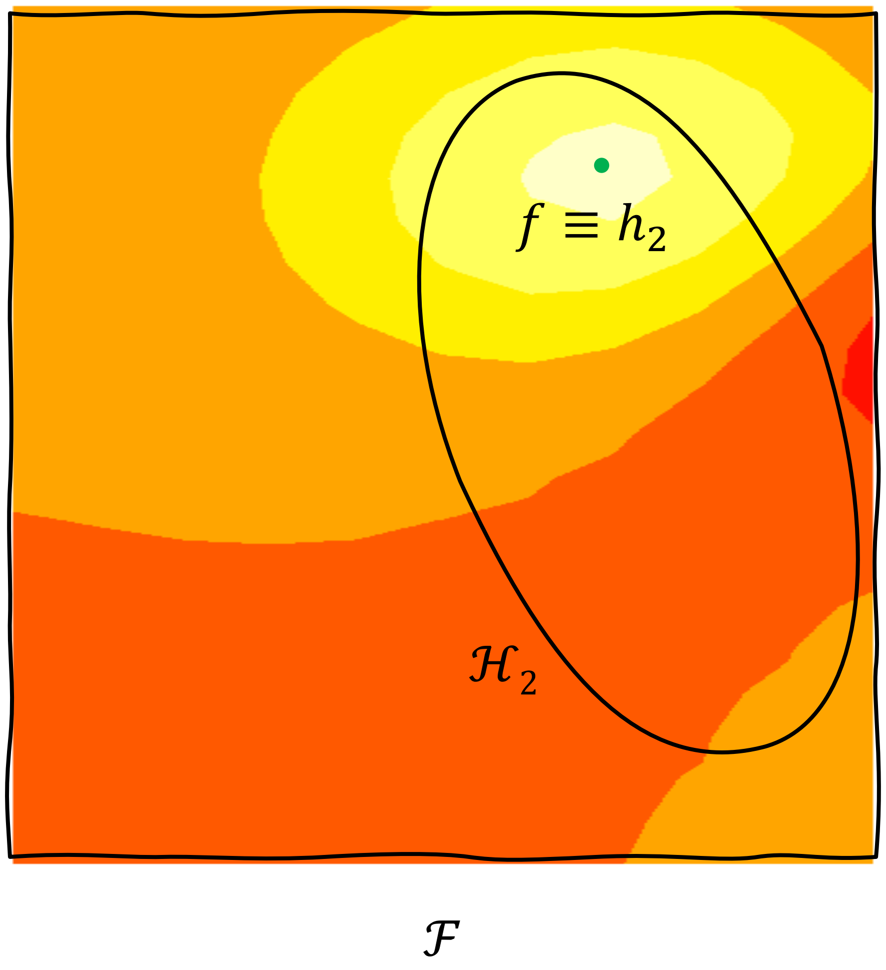
Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - Define a **loss function** $\mathcal{L}$
  - Choose the **hypothesis space** $\mathcal{H}$
  - Find in $\mathcal{H}$ an approximation h of f that **minimizes** $\mathcal{L}$

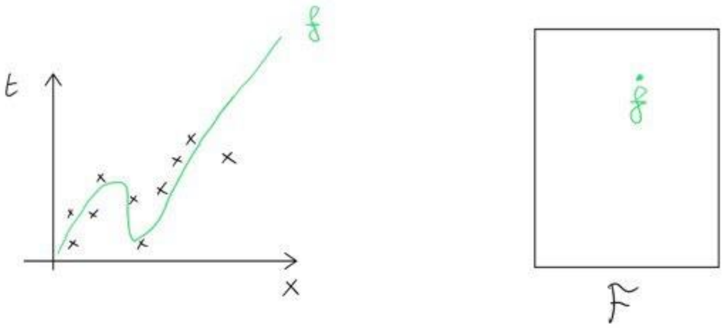


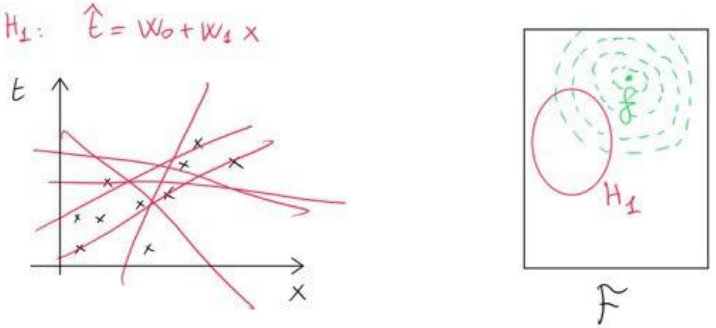
Overview of Supervised Learning

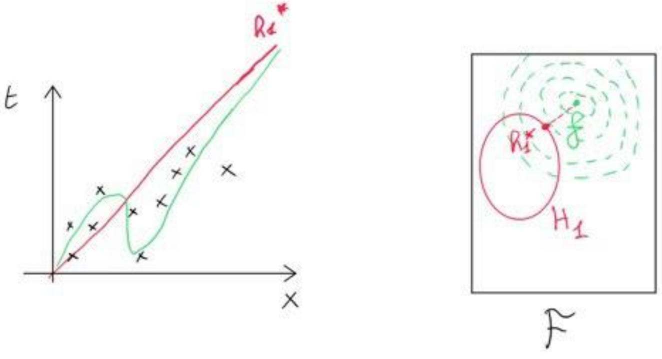
- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - 1 Define a **loss function** $\mathcal{L}$
  - 2 Choose the **hypothesis space** $\mathcal{H}$
  - 3 Find in $\mathcal{H}$ an approximation h of f that **minimizes** $\mathcal{L}$
- What if we enlarge the hypothesis space?
  - We can approximate f without error!

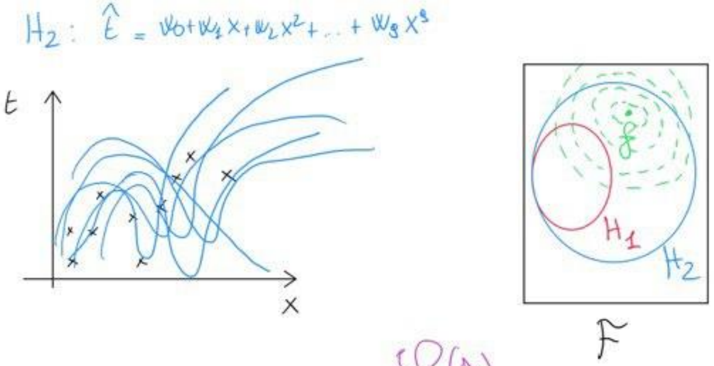


Let see a regression example.





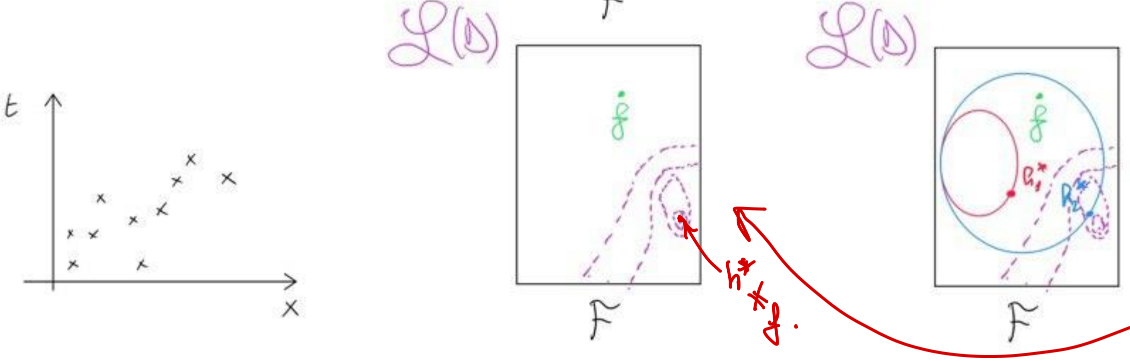


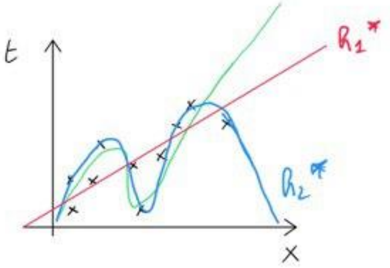


So, $h_2^* \equiv f$?

NO! → we don't know f. So our loss is based only on the data points we have!

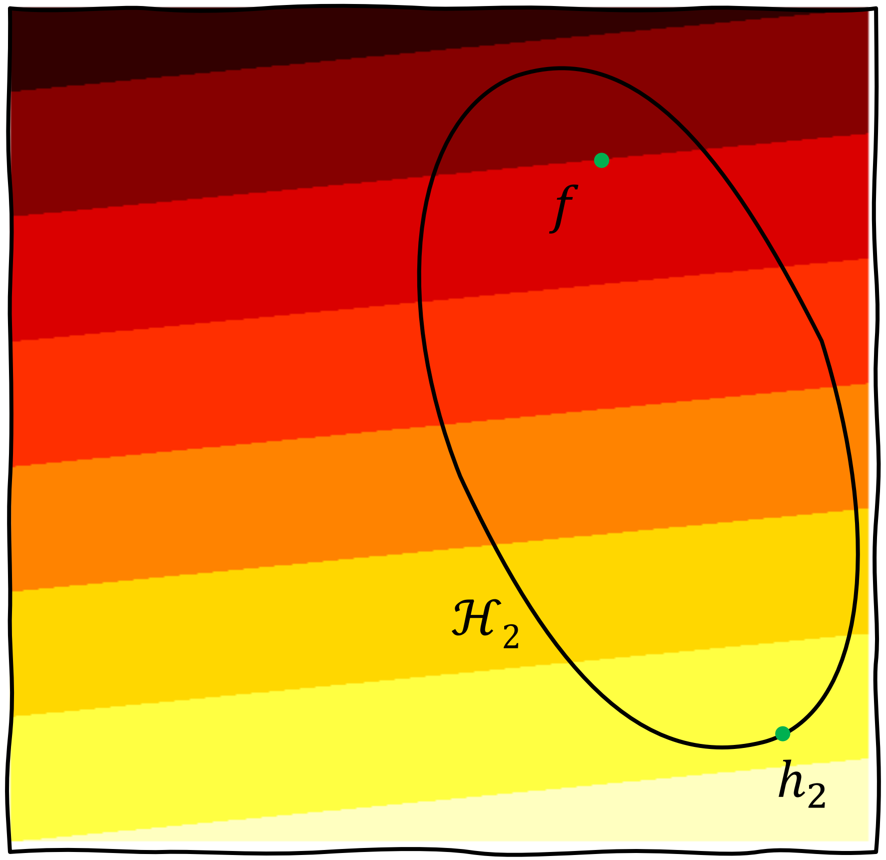
We can minimize our loss but with a h^* which is not f !





Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - Define a **loss function** $\mathcal{L}$
  - Choose the **hypothesis space** $\mathcal{H}$
  - Find in $\mathcal{H}$ an approximation h of f that **minimizes** $\mathcal{L}$
- What if we enlarge the hypothesis space?
  - We can approximate f without error!
  - But we don't know f!

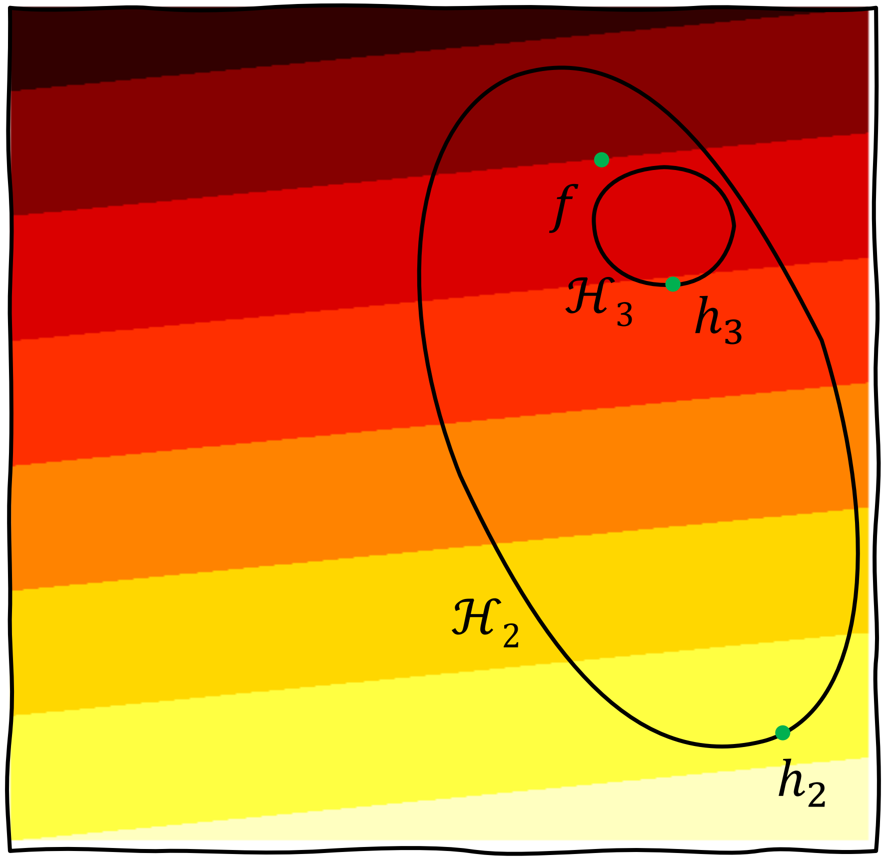


$\mathcal{F}$

Overview of Supervised Learning

- We want to **approximate** a function f given a data set $\mathcal{D}$
- The steps are
  - Define a **loss function** $\mathcal{L}$
  - Choose the **hypothesis space** $\mathcal{H}$
  - Find in $\mathcal{H}$ an approximation h of f that **minimizes** $\mathcal{L}$
- What if we enlarge the hypothesis space?
  - We can approximate f without error!
  - But we don't know f! → we only know data points

indeed: the notion of loss is only based on data points. } we can build a loss $\mathcal{F}$ that goes to 0 but with an f^* which is not f.



Elements of Supervised Learning Algorithms

Representation

Evaluation

Optimization

Examples of representation

- **Linear models**
- **Instance-based**
- Decision trees
- Set of rules
- Graphical models
- Neural networks
- **Gaussian Processes**
- **Support vector machines**
- **Model ensembles**
- etc.

Examples of evaluation

- Accuracy
- Precision and recall
- Squared Error
- Likelihood
- Posterior probability
- Cost/Utility
- Margin
- Entropy
- KL divergence
- etc.

Examples of optimization

- Combinatorial optimization
  - e.g.: Greedy search
- Convex optimization
  - e.g.: Gradient descent
- Constrained optimization
  - e.g.: Linear programming

A Supervised Learning Taxonomy

- Parametric vs Nonparametric
  - Parametric: **fixed and finite** number of parameters
  - Nonparametric: the number of parameters **depends on the training set**
- Empirical Risk Minimization vs Structural Risk Minimization
  - Empirical Risk: Error over the **training set**
  - Structural Risk: Balance training error with **model complexity**
- Direct vs Generative vs Discriminative
  - Generative: Learns the **joint** probability distribution $p(x,t)$
  - Discriminative: Learns the **conditional** probability distribution $p(t|x)$
- Frequentist vs Bayesian
  - Frequentist: use probabilities to model the **sampling** process
  - Bayesian: use probability to **model uncertainty** about the estimate

Direct, Discriminative, or Generative

- Our goal, is learn from **data** a **function** that maps **inputs** to **outputs**

$$\mathcal{D} = \{\langle x, t\rangle\} \Rightarrow t = f(x)$$

- **Direct approach**
  - Learn directly an approximation of f from $\mathcal{D}$
- **Discriminative approach**
  - Model **conditional density** $p(t|x)$
  - Marginalize to find **conditional mean** $\mathbb{E}[t|x] = \int t \cdot p(t|x)\, dt$
- **Generative approach**
  - Model **joint density** $p(x, t)$
  - Infer **conditional density** $p(t|x)$
  - Marginalize to find **conditional mean** $\mathbb{E}[t|x] = \int t \cdot p(t|x)\, dt$