Artificial Intelligence

CSC 665

tyler dae devlin

Machine Learning II

10.31.2023

- **Search:** make decisions by looking ahead
- **Logic:** deduce new facts from existing facts
- **Constraints:** find a way to satisfy a given specification
- **Probability:** reason quantitatively about uncertainty
- **Learning:** make future predictions from past observations

Framework: empirical risk minimization

Goal: reproduce a mapping from inputs to outputs given a training set of example input-output pairs.

- **Representation:** How will we model the relationship between inputs and outputs?
- **Cost:** How will we evaluate whether we're successfully modeling the input-output relationship?
- **Optimizer:** How will we find the best possible (cost-minimizing) model among all possible choices given our representation?

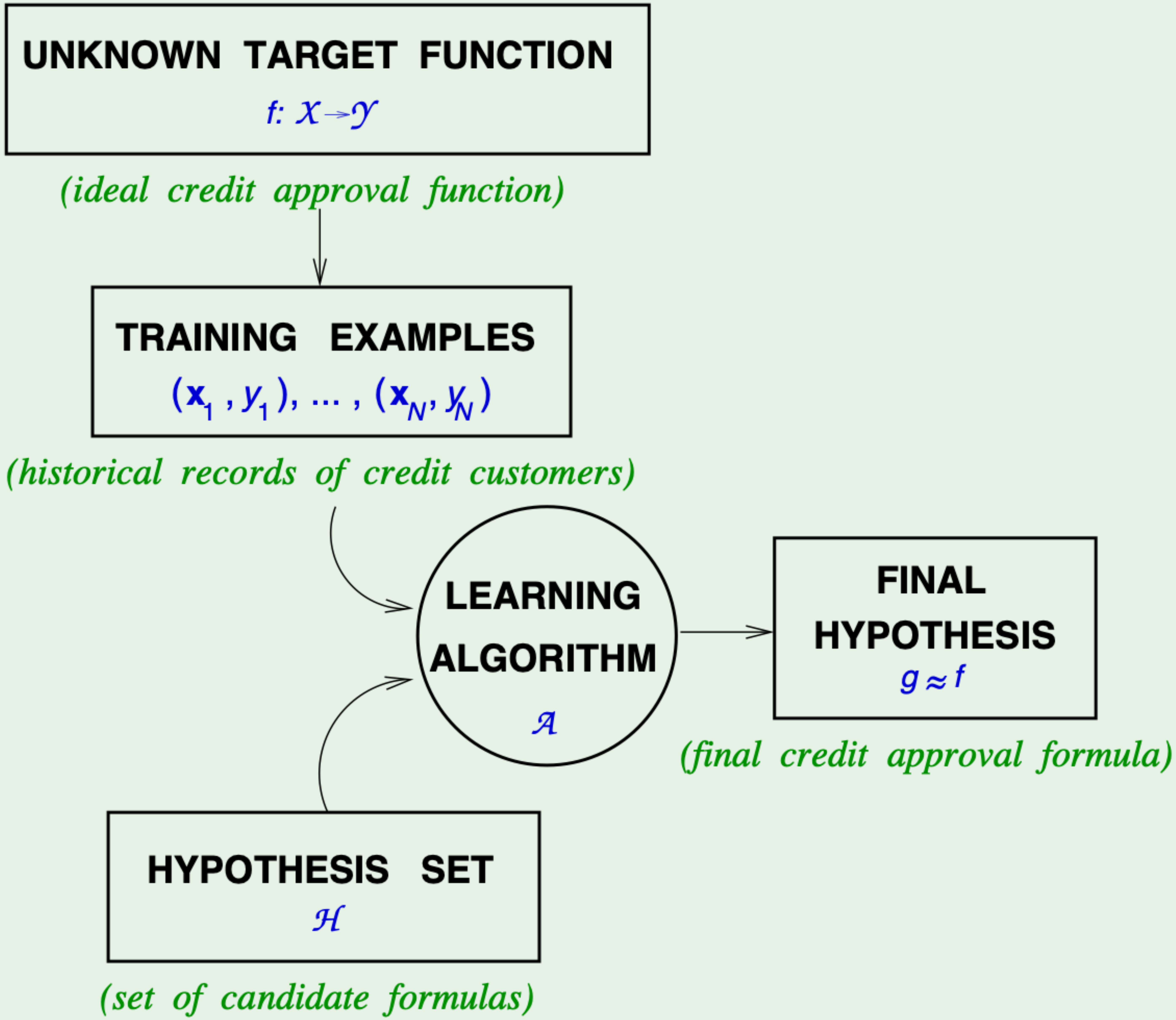
Modeling (Representation, Cost)

Inference (Optimizer)

ERM, mathematically

Goal: approximate a function $f : \mathcal{X} \to \mathcal{Y}$ from features to labels given a training set of examples $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n))$.

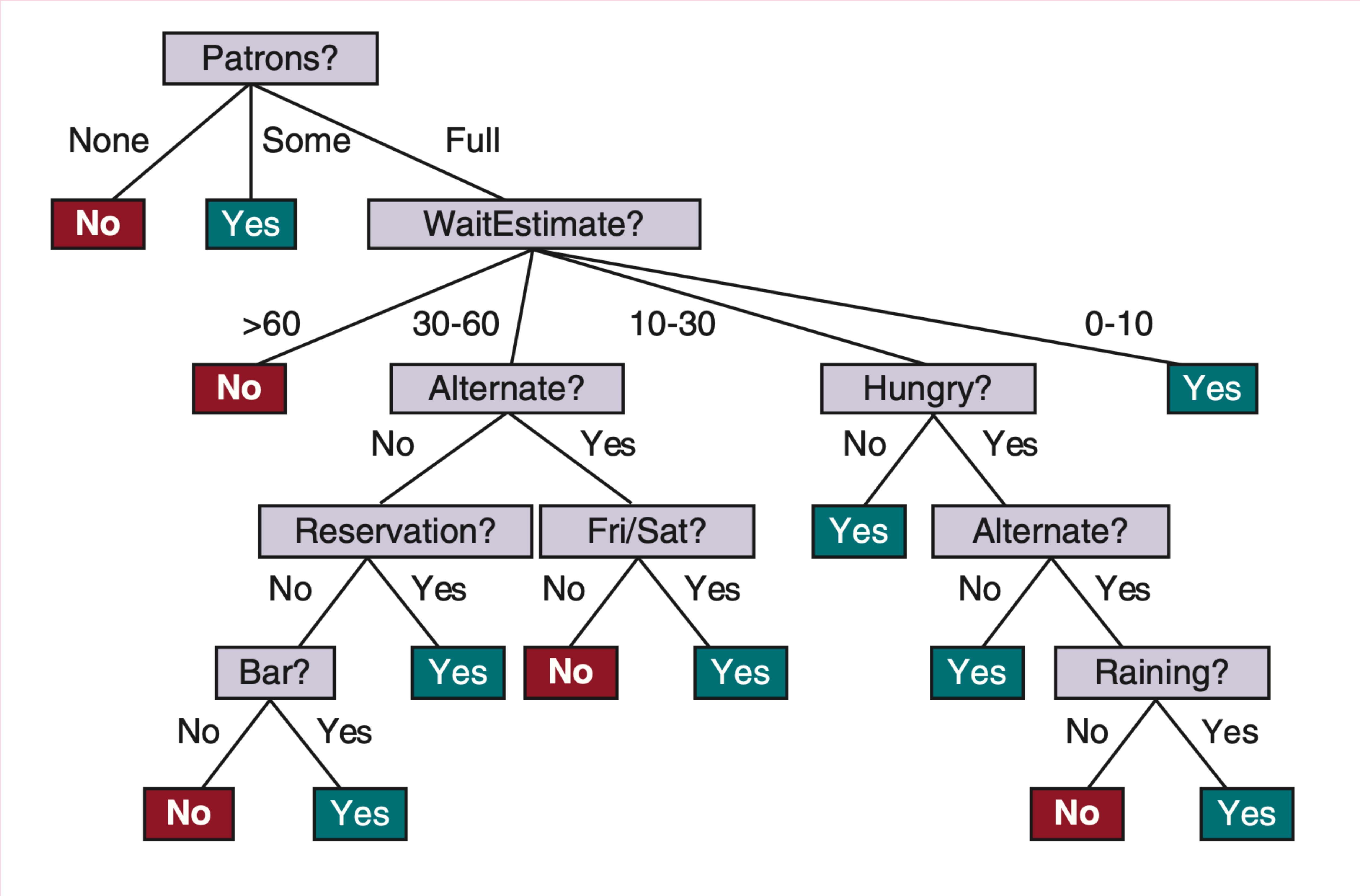
- **Representation:** The hypothesis class $\mathcal{H}$ consisting of all possible candidates to approximate f. E.g. the set of all linear functions $\mathcal{H} = \{x \mapsto wx + b \mid w, b \in \mathbb{R}\}$.
- **Cost:** A function $C : \mathcal{H} \to \mathbb{R}$ that scores elements of the hypothesis class (higher cost is worse). Often the sum of pointwise costs on the data set, i.e. $\sum_{i=1}^{n} c(h(x_i), f(x_i))$ for some function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. E.g. squared error $c(\hat{y}, y) = (\hat{y} - y)^2$.
- **Optimizer:** An algorithm for finding a hypothesis $h \in \mathcal{H}$ that minimizes $C(h)$. $\mathcal{H}$ is usually infinite, so you can't check all hypotheses. E.g. gradient descent.


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final credit approval formula)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

Modeling

Hypothesis class: decision trees

- **Internal nodes** test a particular feature and branch based on the feature value
- **Leaf nodes** correspond to label predictions
- Decision trees are highly expressive
- Small trees can be easy to interpret, and mimic algorithms people might use to make decisions



Inference

Evaluating decision trees

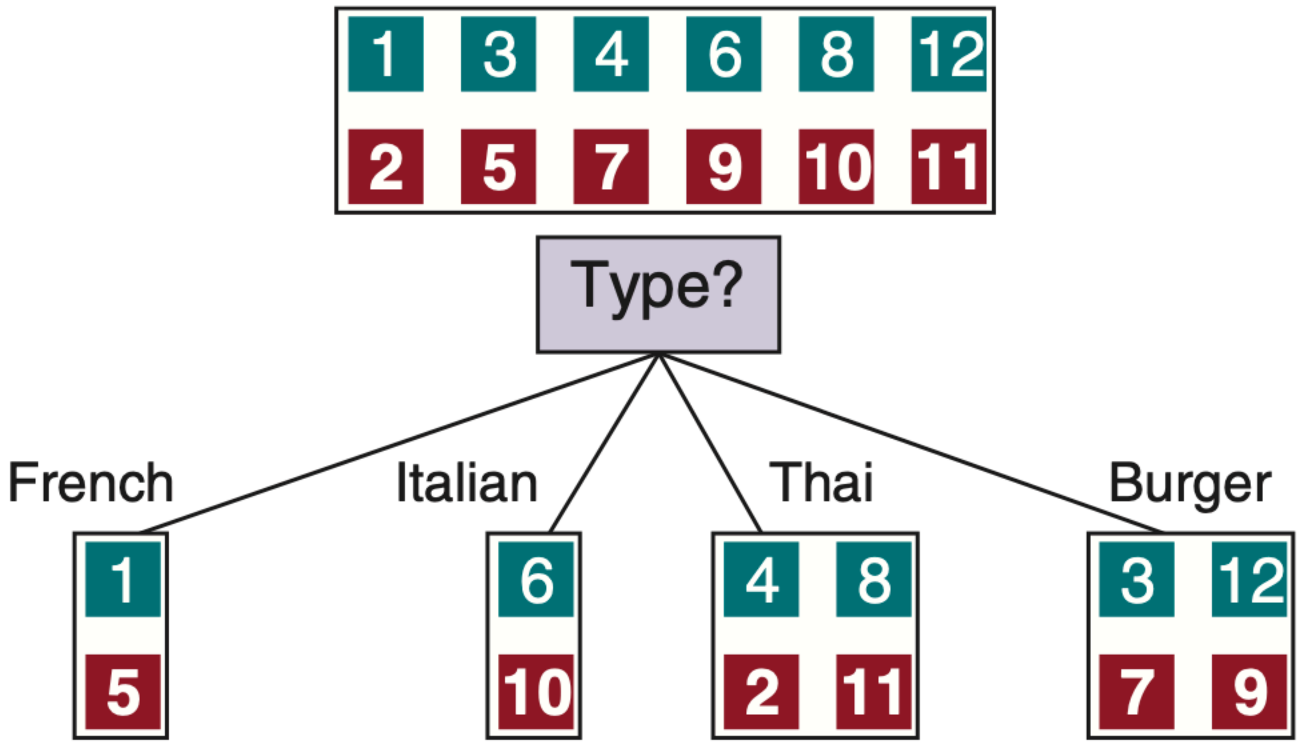
- How do we know if a decision tree is fitting the data well?
- One simple cost function: the proportion of misclassified examples $C(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{h(x_i) \neq f(x_i)\}$. Also known as the *classification error rate.*
- The term inside the sum is often called 0-1 loss

Example: waiting for a restaurant

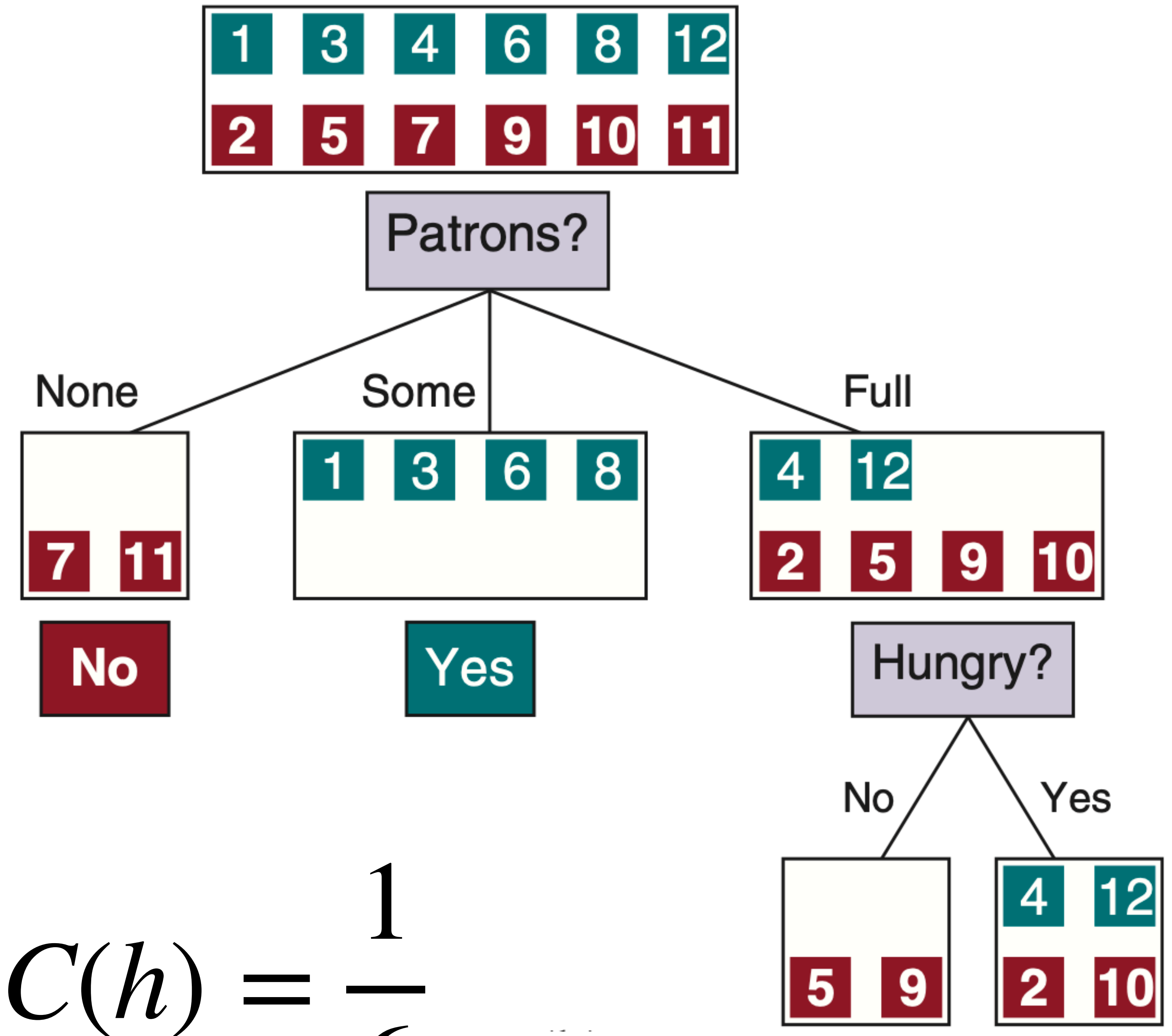
Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} =$ *No*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} =$ *No*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} =$ *Yes*

Evaluating decision trees

$$C(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{h(x_i) \neq f(x_i)\}$$



$$C(h) = \frac{1}{2}$$



$$C(h) = \frac{1}{6}$$

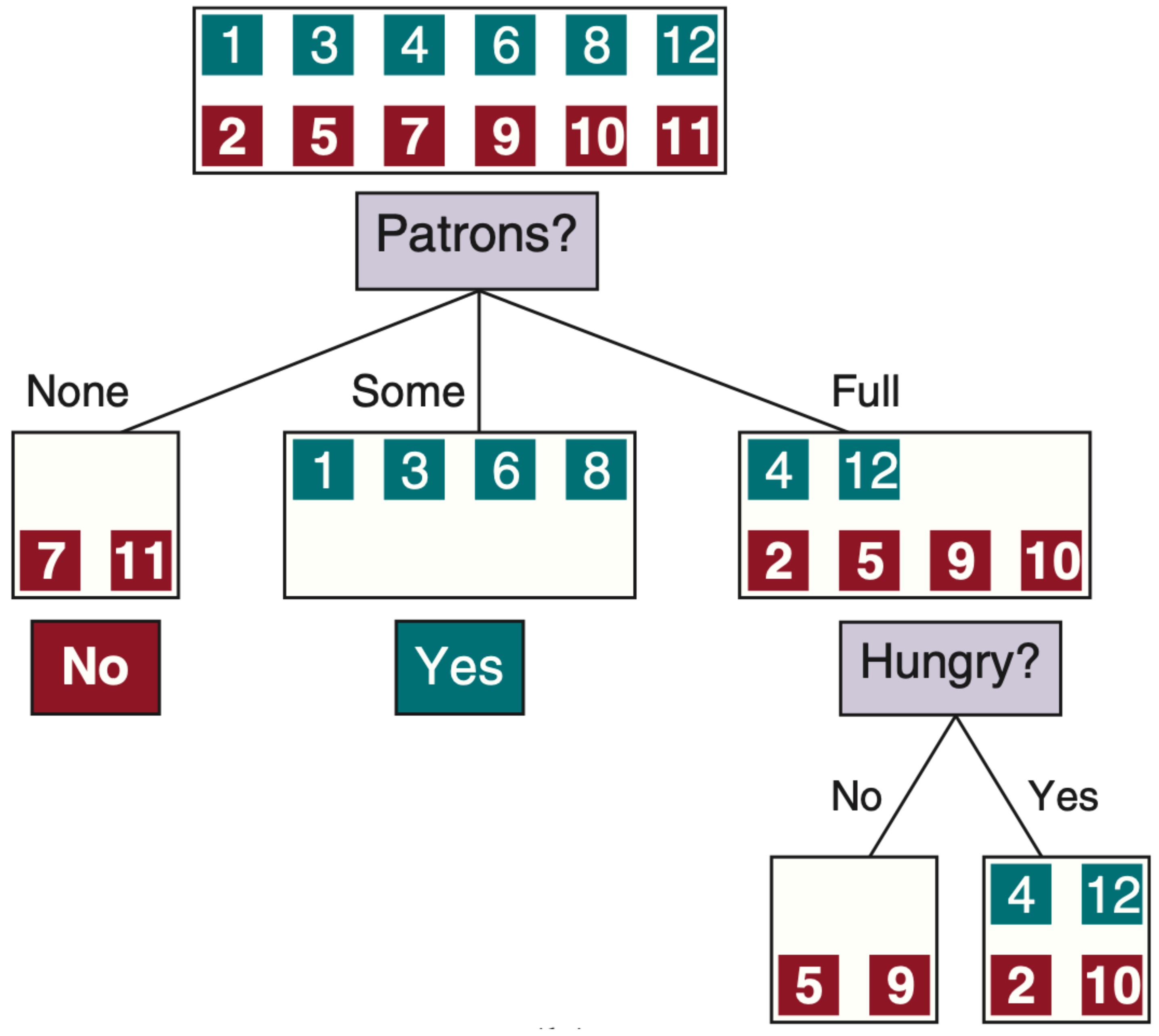
Fitting decision trees

- Want a decision tree with small misclassification rate on the training set
- Want a small tree (why?)
- How many possible decision trees are there?
- For categorical features, number of trees is finite but combinatorially large; impossible to try them all
- Finding the smallest possible tree that is perfectly consistent with the training data is an NP-hard problem
- Need to rely on heuristics to build the tree

Heuristic fitting algorithm

- **Pick the feature** with the most discriminative power, i.e. the one that separates the training examples into groups that are as pure as possible
- **For each child node** of this internal node:
  - If all the examples have the same class, **done**
  - Otherwise,* **recurse**

*Ignoring some edge cases: no examples in a child node, no more features left to use



Restaurant problem via ERM

Goal: approximate $f : \mathcal{X} \to \mathcal{Y}$ given $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n))$.

$\mathcal{X}$ is the set of all possible combinations of feature values (hungry, raining, restaurant type, etc.). $\mathcal{Y} = \{0,1\}$.

- **Representation:** Hypothesis class $\mathcal{H}$

$\mathcal{H}$ is the set of all possible decision trees.

- **Cost:** A function $C : \mathcal{H} \to \mathbb{R}$ that scores hypotheses, often the sum of pointwise costs, i.e. $\sum_{i=1}^{n} c(h(x_i), f(x_i))$ for some function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

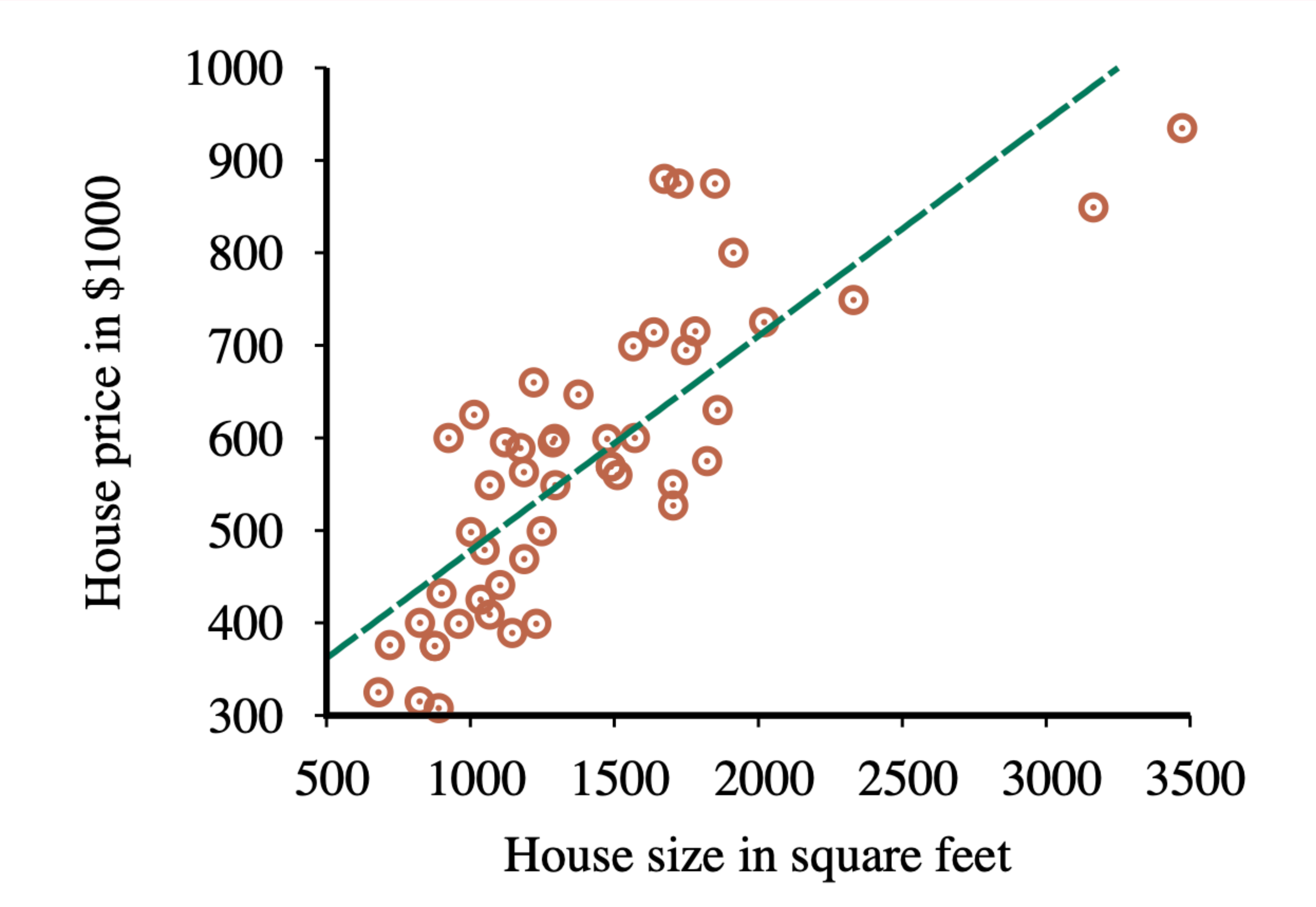
$$C(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{h(x_i) \neq f(x_i)\}$$

- **Optimizer:** An algorithm for finding an $h \in \mathcal{H}$ that minimizes $C(h)$.

“Optimize” via the heuristic fitting algorithm

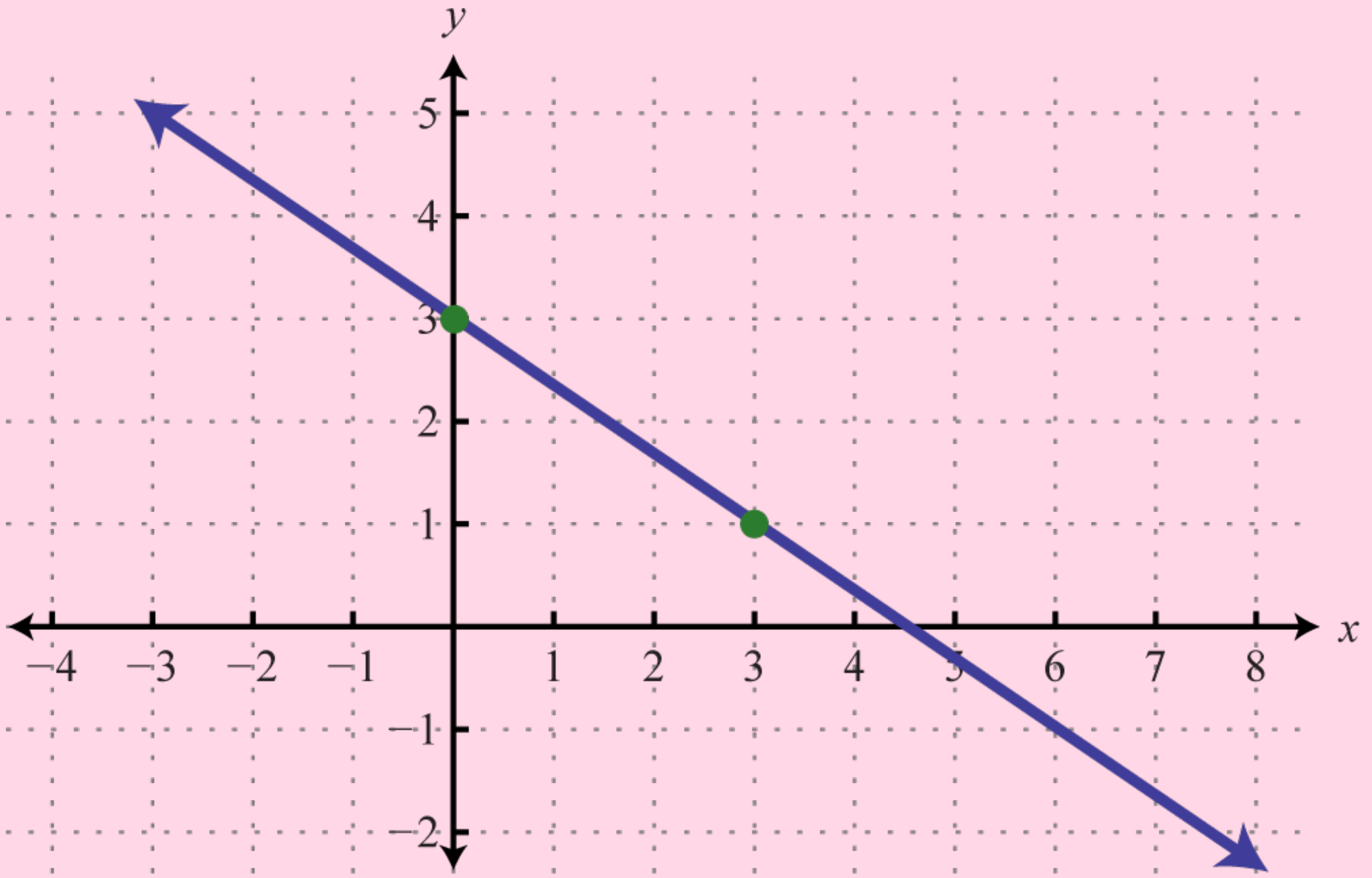
Modeling

Example: predicting housing prices



Representation: linear models

- Hypothesis class: the set of all linear functions $\mathbb{R} \to \mathbb{R}$
- Every one of these functions has the form $h(x) = w_1 x + w_0$
- E.g. $w_0 = 3$ and $w_1 = -2/3$ on the right
- Straight lines are simple and easy to interpret: as x increases by 1 unit, y increases by w_1 units.
- Every hypothesis $h \in \mathscr{H}$ can be identified by a pair of numbers (w_0, w_1).

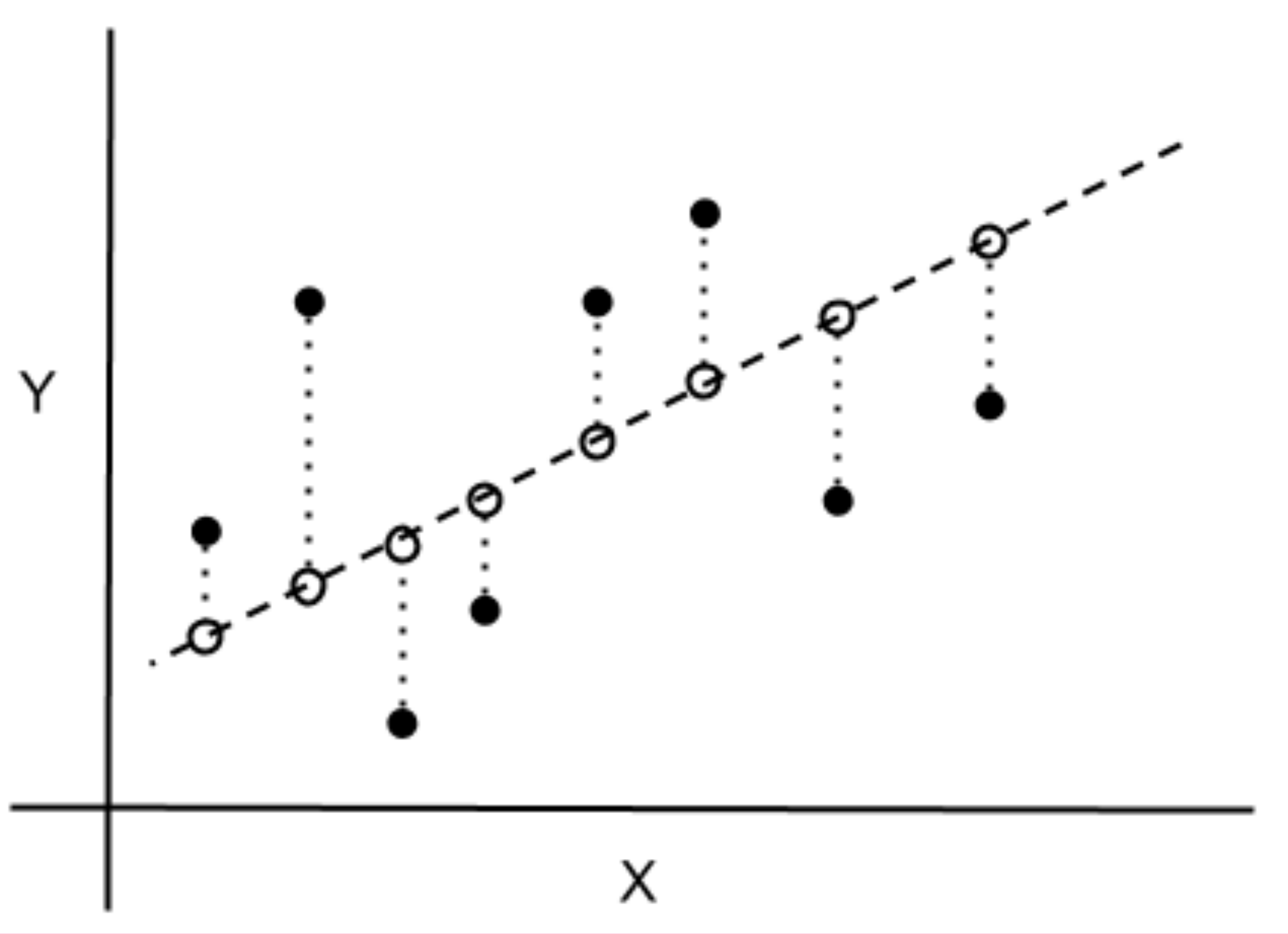


Cost: squared error

- How do we know if a line is a good fit to the data?
- Add up squared errors between predictions and labels.

$$C(h) = \sum_{i=1}^{n} (y_i - h(x_i))^2$$
$$= \sum_{i=1}^{n} (y_i - (w_1 x_i + w_0))^2$$

- Sometimes called ℓ_2 loss, because this is the squared ℓ_2 norm of the residual vector $y - \hat{y}$.



Optimizer: calculus

- How do we find the values of w_0 and w_1 that minimize $C(w_0, w_1)$?
- Because $\mathscr{H}$ is simple, we can optimize directly with calculus!

[calculus on board]

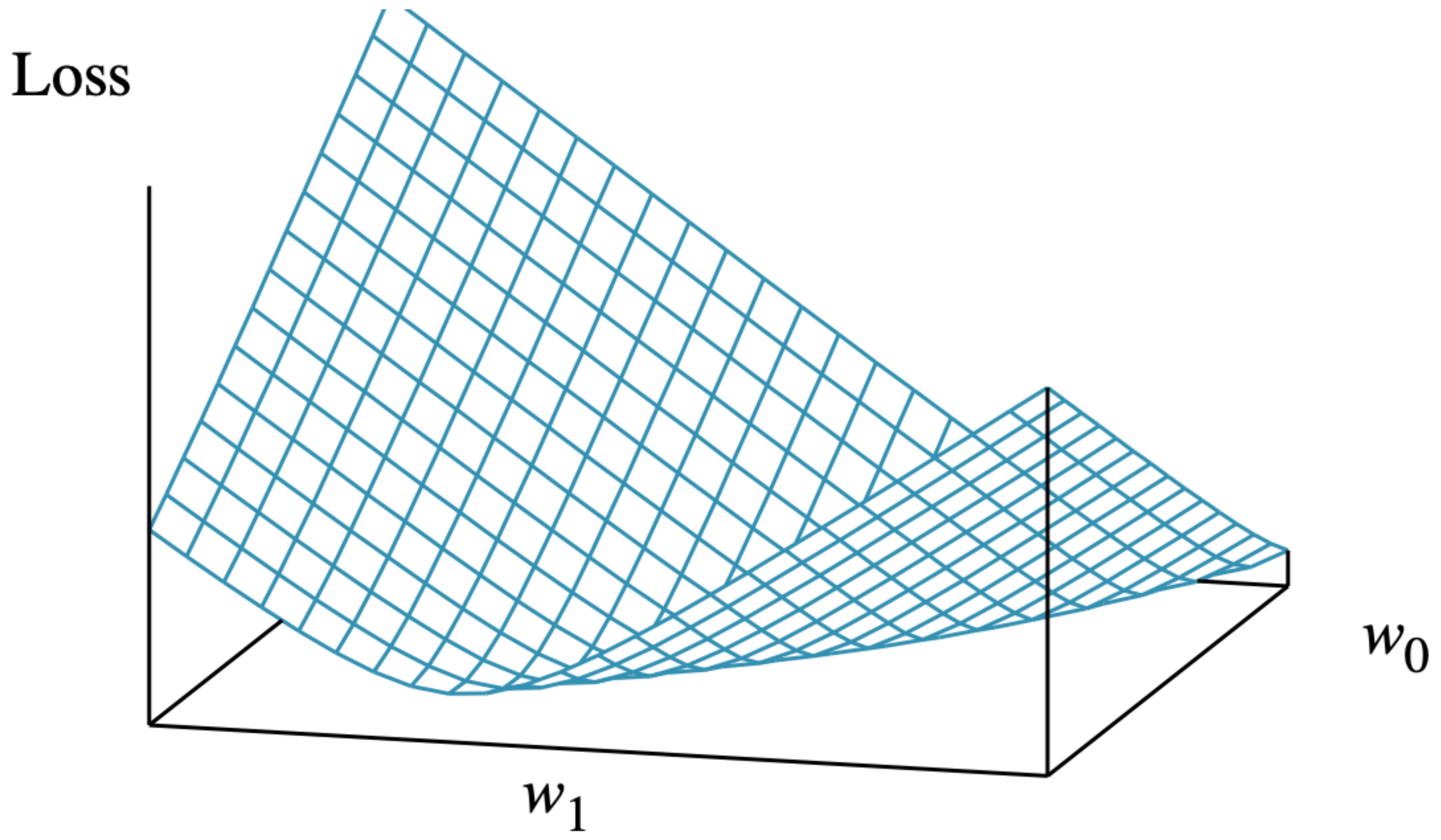
Optimizer: calculus

- How do we find the values of w_0 and w_1 that minimize $C(w_0, w_1)$?
- Because $\mathscr{H}$ is simple, we can optimize directly with calculus!
- Solutions:

$$w_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - w_1 x_i$$

$$w_1 = \frac{n\sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{n\sum_i x_i^2 - \left(\sum x_i\right)^2}$$

- Solution is unique because the cost function is convex in the parameters


Loss
w_1
w_0