Principles of Machine Learning

CSCI-B455

Supervised Learning — II

M. Oğuzhan Kulekci

Probably Approximately Correct (PAC) Learning

- **Accuracy:** We want the hypothesis h to be close to the true target class C.
- A **probability** that a given point is misclassified is **at most ϵ.**
- The **level of desired accuracy** is denoted by ϵ.

h : If $(weight > 115)$ then Orange, else Apple

C : If $(weight > 125)$ then Orange else Apple

Training Data	Learned	True Class
1. Apple 120g	**1. Orange**	**1. Apple**
2. Orange: 150g	2. Orange	2. Orange
3. Apple: 130g	3. Orange	3. Orange
4. Orange: 140g	4. Orange	4. Orange
5. Apple:110g	5. Apple	5. Apple

h deviates from C by 20%, accuracy is 80%.

Error rate is then $\epsilon = 0.2$.

Does h provide a confidence ?

- **Confidence:** We want to be confident that the h provides the desired level of accuracy.
- The level of desired confidence is maintained with at least $(1 - \delta)$ probability.

Probably Approximately Correct (PAC) Learning

LEARNABILITY of a concept class with probabilistic guarantees

- **APPROXIMATELY CORRECT :** We allow an error rate of ϵ in the classification (Accuracy).
- **PROBABLY :** We want to maintain a confidence level $(1-\delta)$ on our accuracy (Confidence).

The error rate is ϵ, and the probability of an error exceeding ϵ is less than δ.

PAC — Learnable Problem: With **enough** examples, possible to learn with desired ϵ, δ guarantees. In other words, there is a hypothesis in the hypothesis space that provides the guarantees.

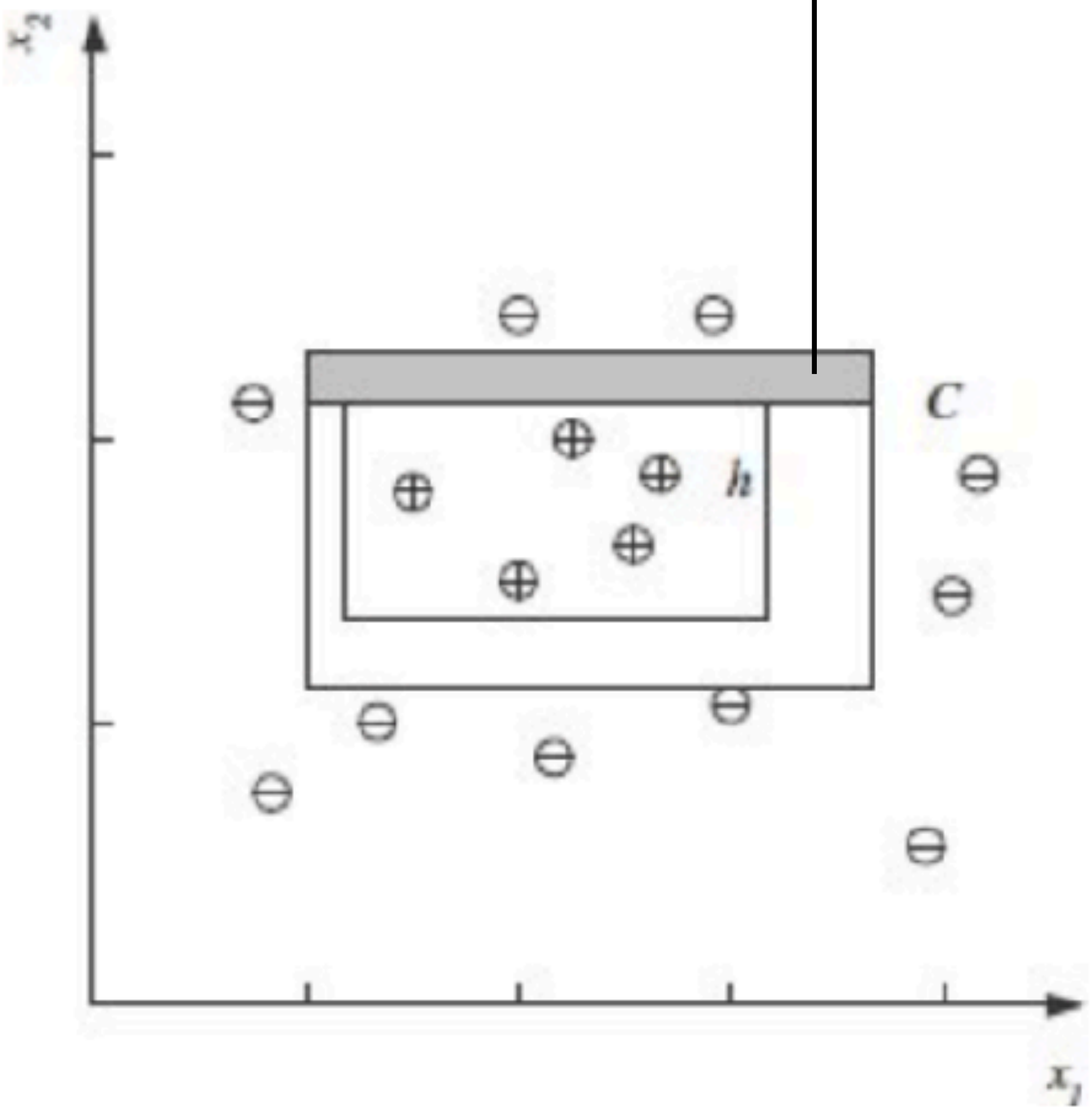
Sample complexity: How many examples we need to learn a hypothesis with the desired **accuracy** and **confidence** ?

Probably Approximately Correct (PAC) Learning

How many examples we need to learn a hypothesis with the desired accuracy and confidence ?

- We assume h is the tightest hypothesis $\mathcal{S}$.
- N examples are drawn from C with a fixed but unknown probability distribution
- We aim to estimate N such that $P(C\Delta h \leq \epsilon) \geq (1-\delta)$, which also means $P(C\Delta h > \epsilon) < \delta$.
- $C\Delta h$ denotes the error, which is the region between C and h.

Anything falling in this strip is an **error**. We have 4 strips around h



Probability that an example point is out of that strip is $1-\epsilon/4$.
All N points are out of it is $(1-\epsilon/4)^N$.
N draws missing any of those 4 strips is $4(1-\epsilon/4)^N$
Solve for $4(1-\epsilon/4)^N < \delta$ (see book chapter 2.3 for details)

$$N \geq \left(\frac{4}{\epsilon}\right) \log\left(\frac{4}{\delta}\right)$$

Probably Approximately Correct (PAC) Learning

- Confidence: 95% , $1 - \delta = 0.95 \Rightarrow \delta = 0.05$
- Accuracy: 99%, error rate $\epsilon = 0.01$
- Number of samples we will need is N.

$$N \geq \left(\frac{4}{\epsilon}\right) \log\left(\frac{4}{\delta}\right) \Rightarrow N \geq \frac{4}{0.01} \log \frac{4}{0.05} \Rightarrow N \geq 400 \log 80 \Rightarrow N \geq 762$$

Probably Approximately Correct (PAC) Learning

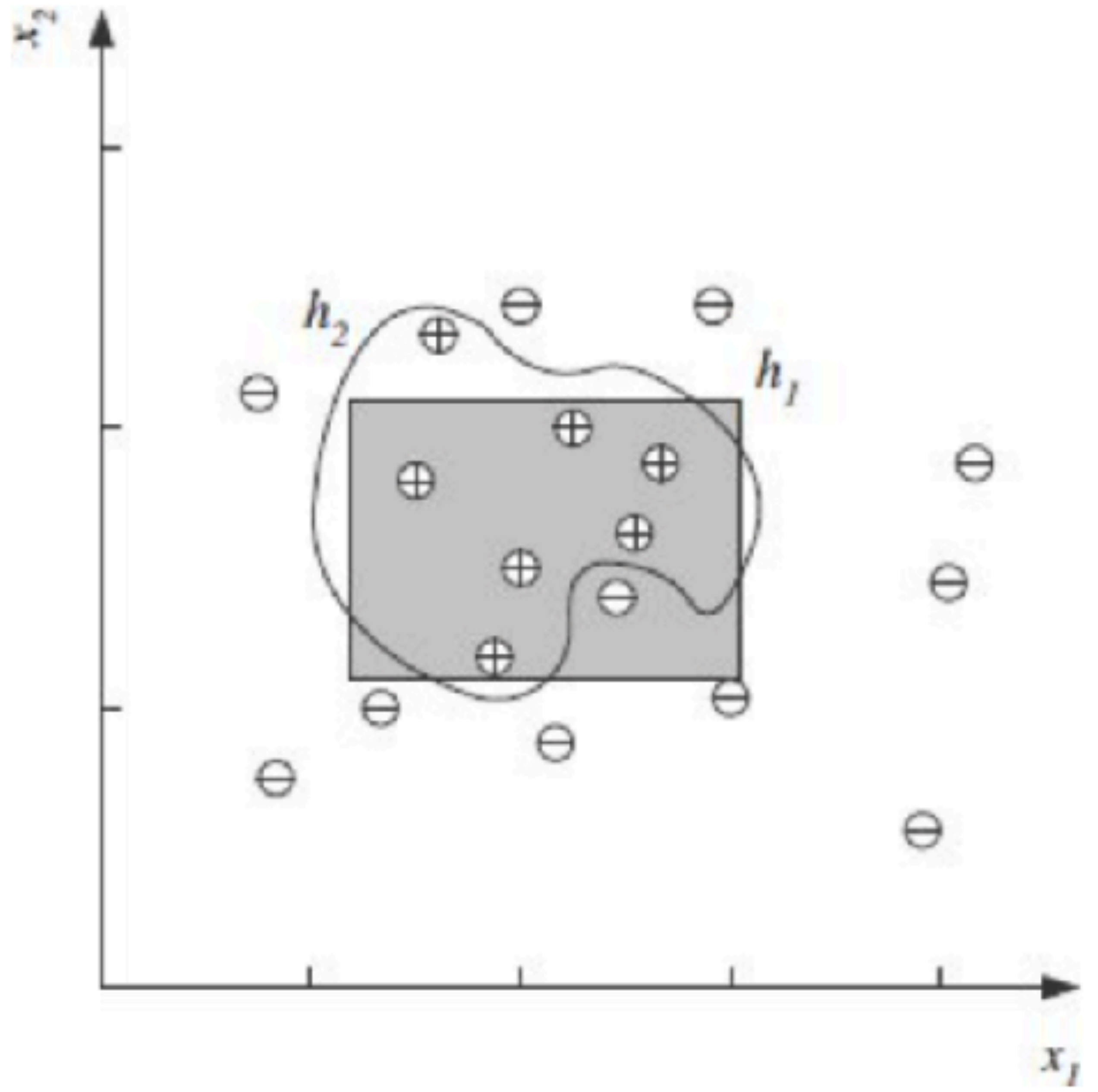
How many examples we need to learn a hypothesis with the desired accuracy and confidence ?

Another approach by proving "The probability that there exists a hypothesis h that is consistent with m examples and satisfies $Error(h) > \epsilon$ is less then $|H|(1-\epsilon)^m$. [1]

- The hypothesis $h \in H$ is a **bad** one when $Error(h) > \epsilon$,. Then, correct decision on a single point is **less than** $(1-\epsilon)$, and the probability that such an h is consistent with all m points is **less** than $(1-\epsilon)^m$.
- The probability that any one of the $|H|$ hypothesis satisfies $Error(h) > \epsilon$ is $|H|(1-\epsilon)^m$.
- Now, we have a learning problem and the chosen hypothesis comes with **worse** than ϵ error probability.
- We want this situation to be **upper bounded by** δ, hence $|H|(1-\epsilon)^m < \delta$.
- The number of points, m, to satisfy this is $m > \dfrac{1}{\epsilon}\left(\ln|H| + \ln\dfrac{1}{\delta}\right)$

[1] *https://www.cis.upenn.edu/~danroth/Teaching/CS446-17/LectureNotesNew/colt/main.pdf*

Noise in Learning



- Errors in measurements, recordings, etc…
- Errors in labeling the training data (teacher error)
- Effect of neglected attributes
- **Actually, this is the real-life scenario :(**

- No simple boundaries
- No zero-error on learning

- **Occam's Razor, principle or law of parsimony:** Among possible hypothesis, simpler is better unless there is strong evidence to choose the more complex one
- Avoid **overfitting** that results more than necessarily complex models.
- More complex, less generalizable!

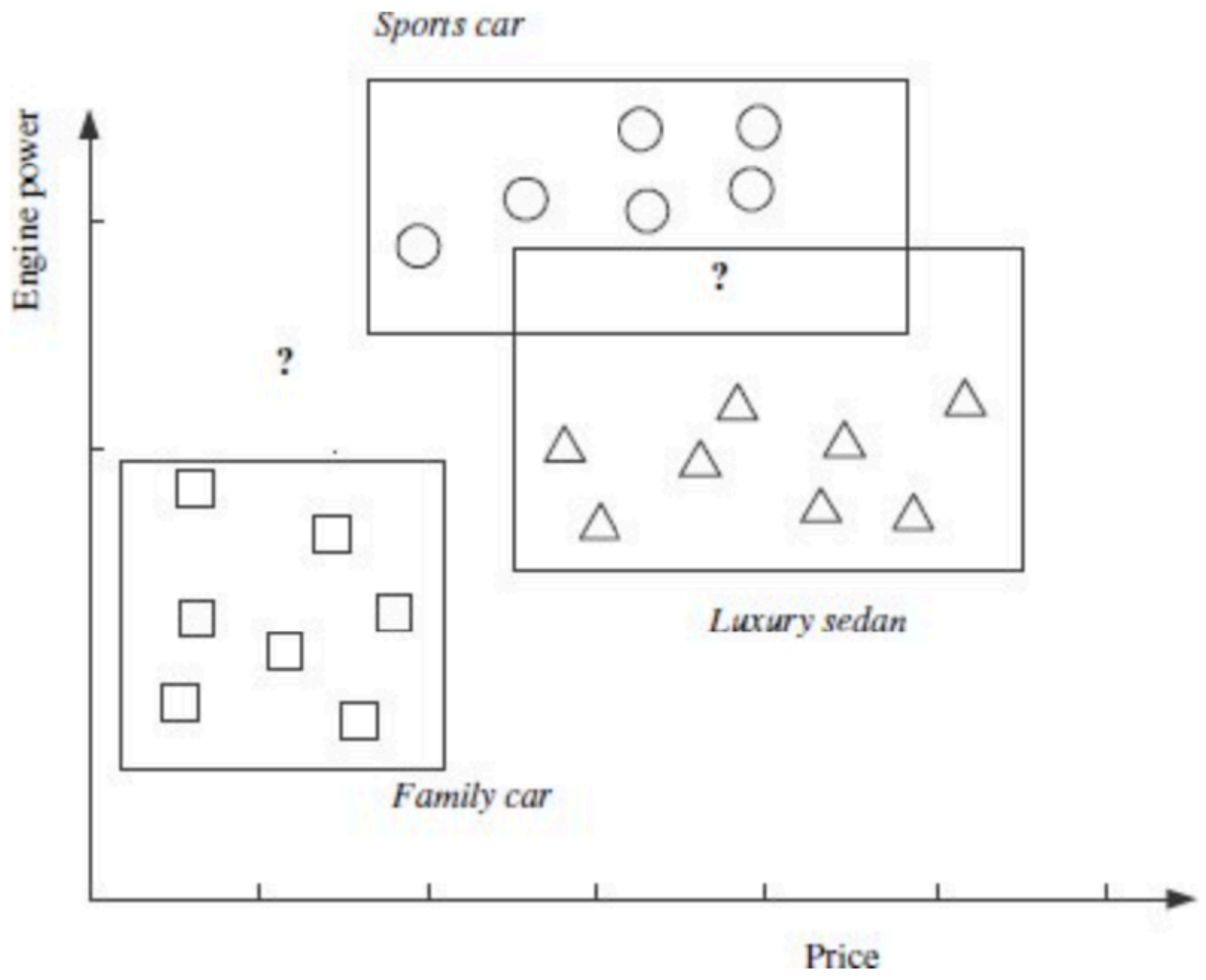
Learning Multiple Classes

What if we have $K > 2$ classes ?

Classes: $C_i,\ i = 1,2,\ldots,K$

Training data: $\mathscr{X} = \{x^t, r^t\}_{t=1}^N$

Class attribution r_i^t of $x_i^t \in \mathscr{X}$ is a $K-$ dimensional binary vector as $r^t = \langle r_1^t, r_2^t, \ldots, r_K^t, \rangle$, where $r_i^t = 1$ if $x^t \in C_i$, else $r_i^t = 0$



$K-$class classification defines K two$-$class classification problems.

Learning Multiple Classes

K—class classification defines K two—class classification problems.

We need K hypotheses as $h_1, h_2, \ldots, h_K$, each for a two-class separation of a specific class among K

$$h(x^t) = \langle\ h_1(x^t), h_2(x^t), \ldots, h_K(x^t)\ \rangle$$

$$h_i(x^t) = \begin{cases} 1, & if\ \ x^t \in C_i \\ 0, & if\ \ x^t \notin C_i \end{cases}$$

Learn $h_1, h_2, ..h_K$ that minimizes the error E.

$$E(\{h_1, h_2, .., h_K\} \mid \mathscr{X}) = \sum_{t=1}^{N} \sum_{i=1}^{K} 1(h_i(x^t) \neq r_i^t)$$

- Compare $h(x^t)$ with r^t to decide on correct class attribution.
- **Exactly one dimension of K should be 1, all others are 0 for valid assignments.**
- If there are more than one dimension set or all are zero, then this is a ***doubt*** point and a ***reject*** case.
- We try to learn the hypothesis that minimize the total error.

Regression

- Given the training set $\mathcal{X} = \{x^t, r^t\}_{t=1}^N$, regression aims to learn the function $f(x^t) = r^t$.
- **Both regression and classification are supervised learning problems.**
- The output of classification is boolean, where regression output is a real number.

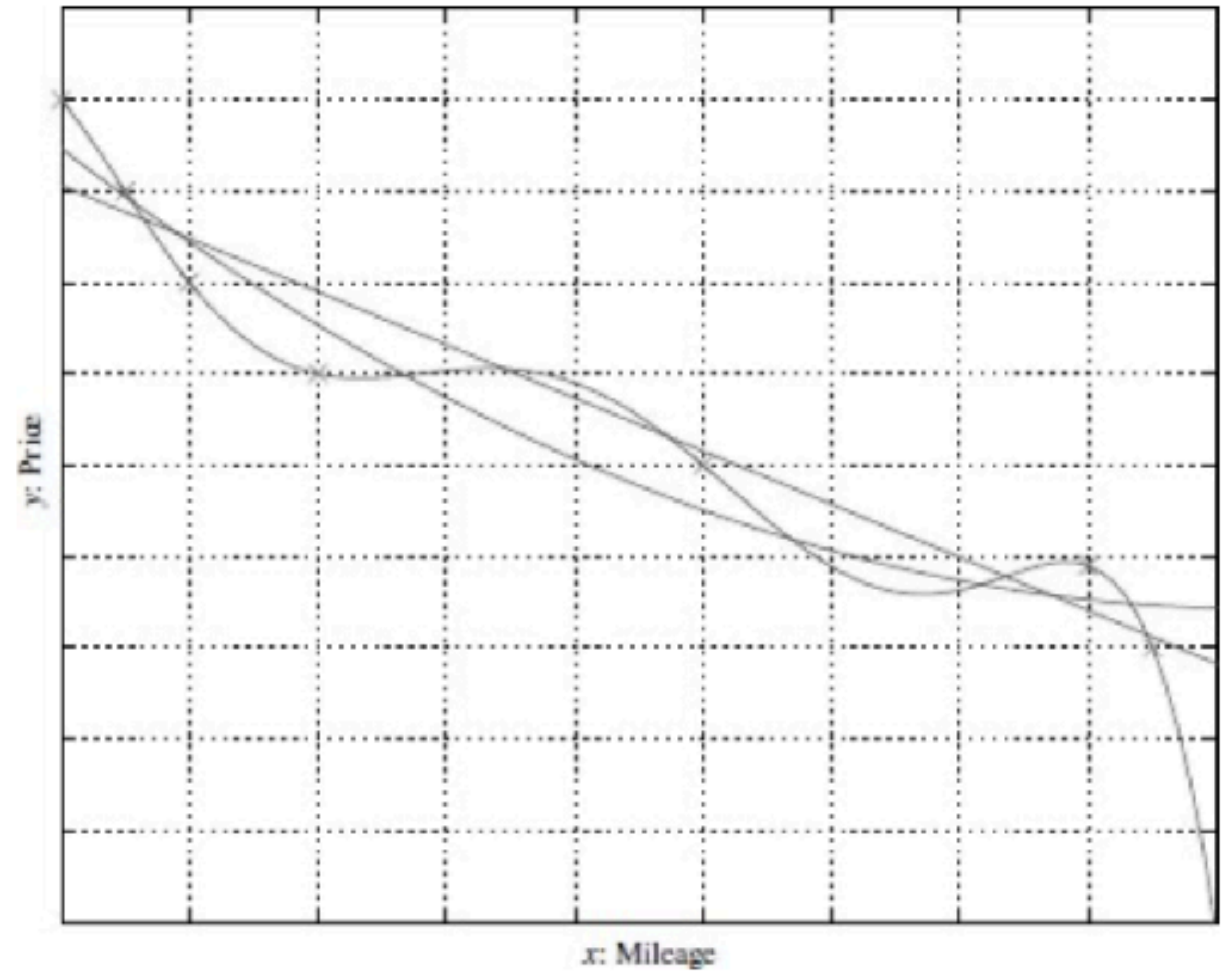
- **Interpolation:** Find function $f()$ such that $r^t = f(x^t), \forall t \in 1,2,..,N$.
- **Polynomial interpolation:** At most degree N polynomial for given N points.
- Finding the output of an input $x \notin \mathcal{X}$ not in the training, is called **extrapolation, e.g., prediction**
- There is **no noise.**

- **Regression:** $r^t = f(x^t) + \epsilon$**,** where **there is noise** which causes the error ϵ.
- In other words, there are other **hidden unknown** attribute(s) z^t that effect r^t, $r^t = f(x^t, z^t)$

Regression

- **Regression:** $r^t = f(x^t) + \epsilon$, where there is noise which causes the error ϵ.
- In other words, there are other **hidden unknown** attribute(s) z^t that effect r^t, $r^t = f(x^t, z^t)$

The model $g(x^t)$ approximates the function $f(x^t)$. Learn $g(x)$ by minimizing the error $E(g \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}[r^t - g(x^t)]^2$



The hypothesis class for regression:

Assume $g()$ is a linear function and x^t is a $d-$dimensional vector.

$$g(x^t) = w_1x_1^t + w_2x_2^t + \ldots + w_dx_d^t + w_0 = w_0 + \sum_{i=1}^{d} w_i \cdot x_i^t$$

The parameters $w_0, w_1, ..w_d$ define the hypothesis class.

Depending on the assumed function, the hypothesis set is specified by its parameters, and regression aims to learn those parameters from the training set.

Regression

- **Regression:** $r^t = f(x^t) + \epsilon$**,** where there is noise which causes the error ϵ.
- In other words, there are other **hidden unknown** attribute(s) z^t that effect r^t, $r^t = f(x^t, z^t)$

Assume $g(x) = w_0 + w_1 x$, and we aim to minimize $E(w_0, w_1 \mid \mathcal{X}) = \dfrac{1}{N}\sum_{t=1}^{N}[r^t - (w_0 + w_1 x^t)]^2$, which

can be done solving the partial derivatives with respect to to w_0 and w_1 equal to zero, which returns

$$w_1 = \frac{\left(\sum_{t=1}^{N} x^t r^t\right) - \bar{x}\bar{r}N}{\left(\sum_{t=1}^{N} (x^t)^2 N\bar{x}^2\right)} \quad and \quad w_0 = \bar{r} - w_1\bar{x} \quad \text{,where } \bar{x} = \frac{\sum_{t=1}^{N} x^t}{N}, \ and \ \bar{r} = \frac{\sum_{t=1}^{N} r^t}{N}$$

- If the error with the assumed model $g(x) = w_0 + w_1 x$ is still high, then we try the second-order $g(x) = w_0 + w_1 x + w_2 x^2$, find the parameters and check the error.
- Higher-order polynomials will reduce the error. Then, isn't it better to use the highest possible?
- **No! Remember the Occam's razor.**

ILL-POSED Problem

x_1	x_2	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	~~0~~	~~0~~	1	1	~~0~~	~~0~~	1	1	~~0~~	~~0~~	1	1	~~0~~	~~0~~	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

If we know $f(x_1 = 1, x_2 = 0) = 1$, then $h_1, h_2, h_5, h_6, h_9, h_{10}, h_{13}, h_{14}$ are eliminated

- Assume we aim to learn the $f(x_1, x_2)$ BOOLEAN function
- Possible (x_1, x_2) values are $4 = 2^2$
- Possible output values for those 4 cases can be assigned in $16 = 2^4$ ways.
- Each is a hypothesis, and thus, $|\mathscr{H}| = 16$

- Given the training set, inconsistent hypothesis can be eliminated until we are left with a unique one.
- It needs all 2^{2^d} non-contradicting training samples to reach the unique solution.
- However, training data is usually not enough to specify the unique solution.
- Regression and classification problems are in general **ill-posed.**

Inductive Bias

- **Data by itself is not enough for solution in ill-posed problems.**
- We need **assumptions**. What are these?
  - The attributes we use, e.g., in family-car decision we used price and engine power !?
  - The models we assume in regression or classification, e.g., second-degree polynomial, axis-aligned rectangle, etc..
  - The error function that we minimize also creates a bias
  - All introduce a **bias** in the final solution
- We actually need this bias to be able to induce a solution, hence, it is **inductive** bias**.**

- Each hypothesis class has a capacity (expressive power)
- Increased capacity brings increased complexity, e.g., instead of one rectangle, how about two ?
- Thus, to what extend we need to increase the capacity of $\mathscr{H}$?

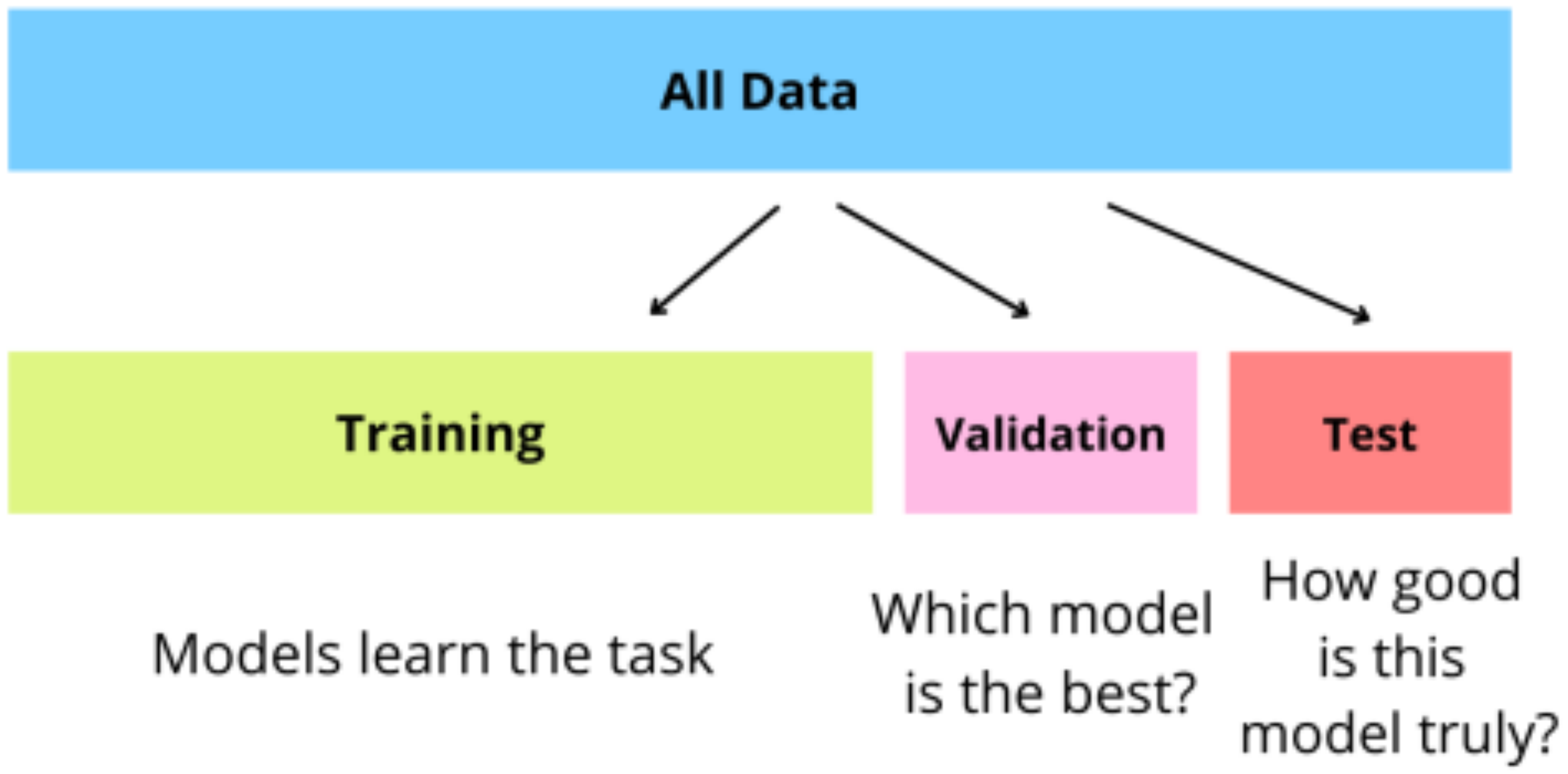
Model Selection and Generalization

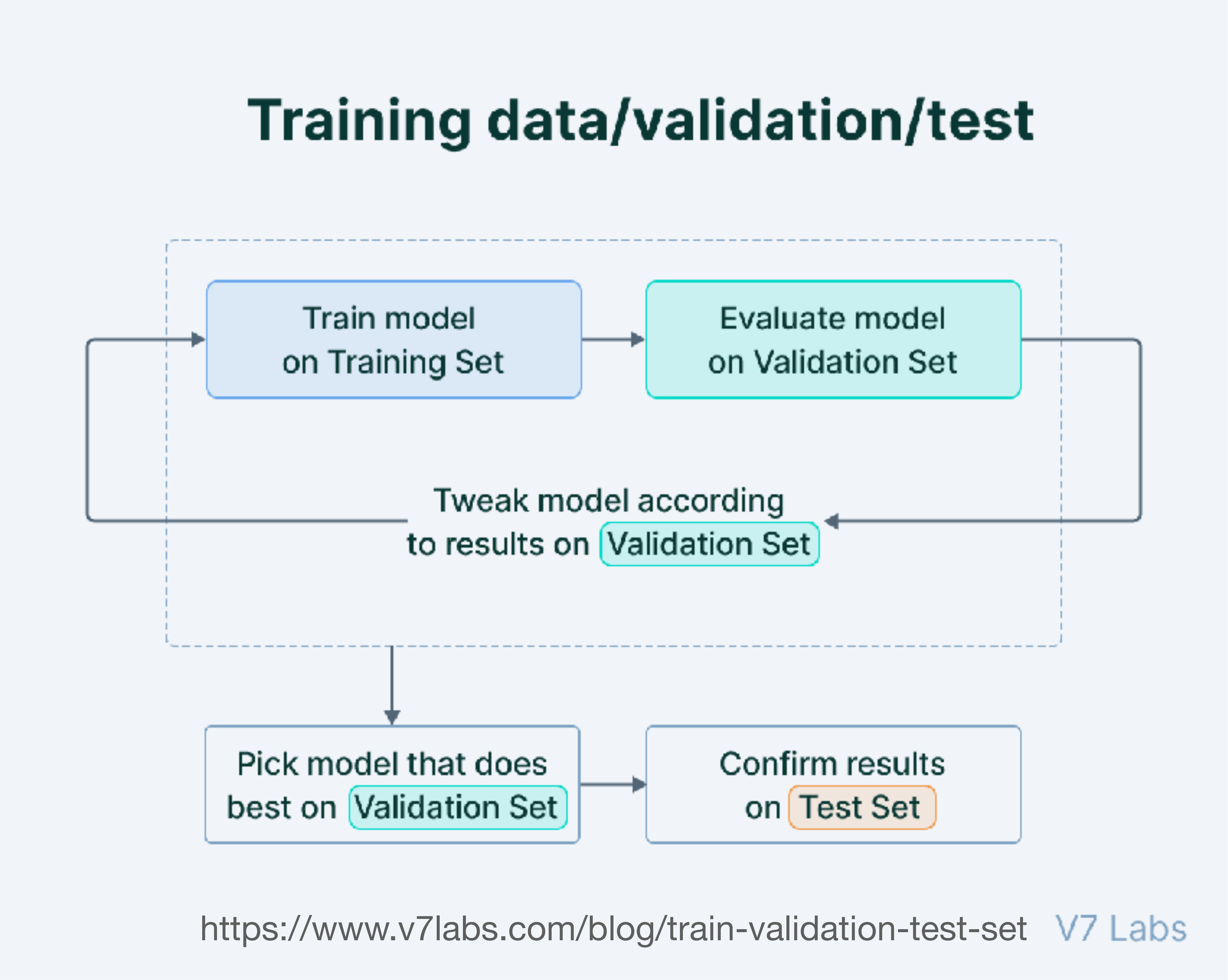
- Inductive bias is unavoidable, but how to choose a good one ?
- What would be a good model selection?
    - What would be the degree of the polynomial?
    - Axis-aligned rectangle, free rectangle, lines, triangles, etc ? How to choose?
- Model selection is deciding on the hypothesis class $\mathscr{H}$.

- Aim of the learning is NOT to replicate or memorize the training data
- **The aim is to do well on future unseen data !**
- Therefore, increasing the performance on training data is good only up to a point !
- The **generalization** performance of the learned model is its success on future instances

Training / Validation / Test Sets & Cross-Validation

- Split the available labeled data into three sets as training, validation, and test sets.





https://medium.com/@rahulchavan4894/understanding-train-test-and-validation-dataset-split-in-simple-quick-terms-5a8630fe58c8