Regression Methods

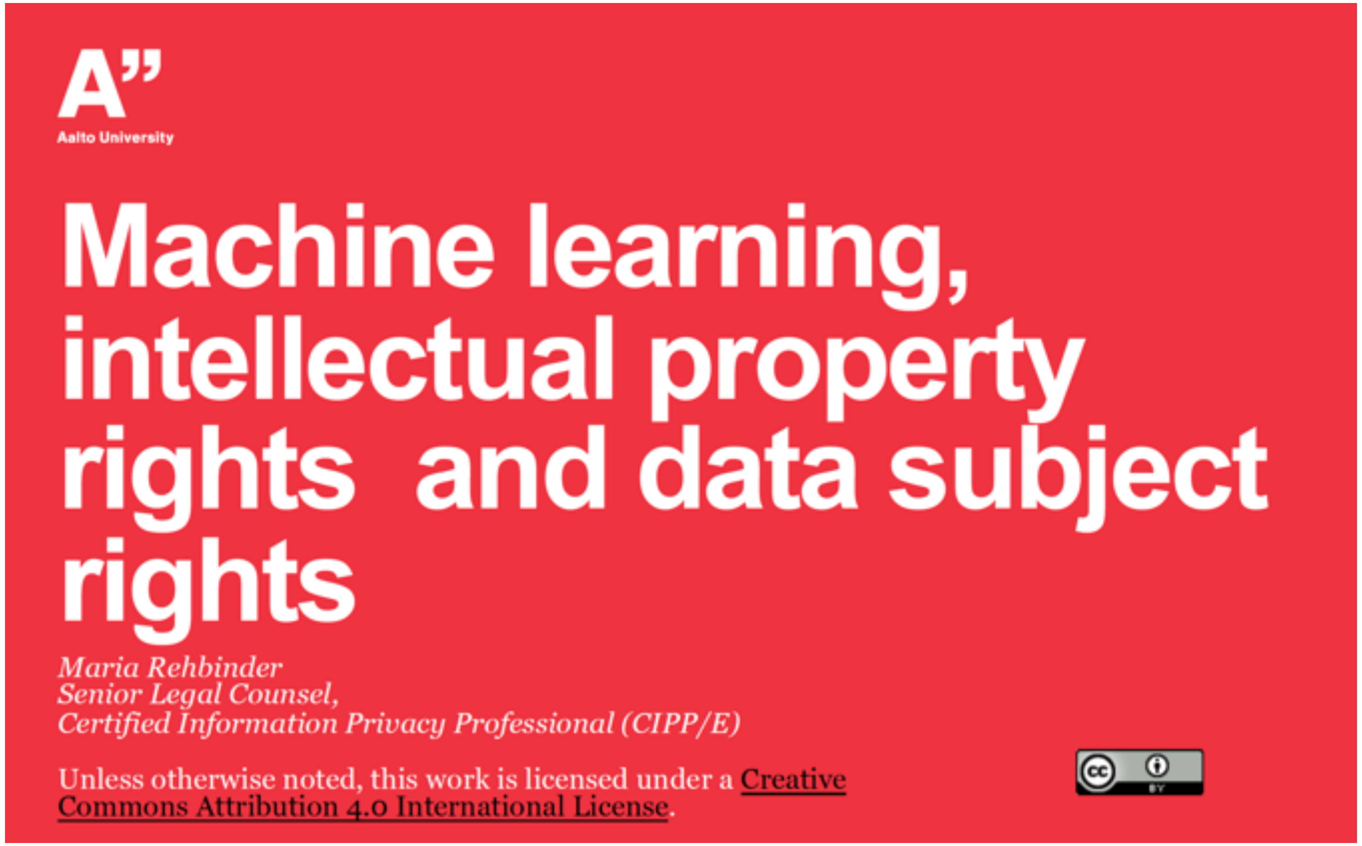
Alexander Jung

Assistant Professor

Department of Computer Science

Aalto University

Guest Zoom Lecture Tmrw at 18:00!

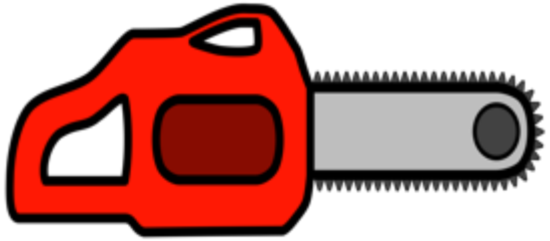


Maria Rehbinder

Senior Legal Counsel of Aalto University

Legal Aspects of Machine Learning

non-sensitive data
(e.g.. from wikidata.org)

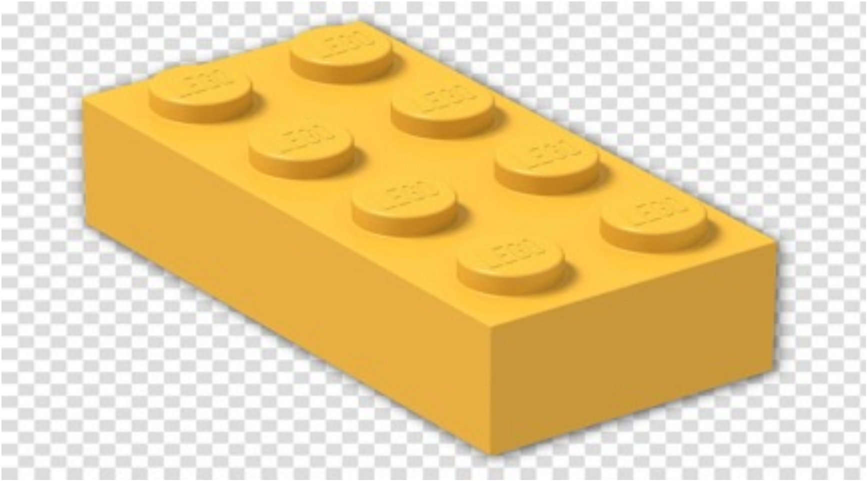


Machine Learning Methods

sensitive data
protected!

Quick Refresher

Components
of
Machine Learning



data: features, labels

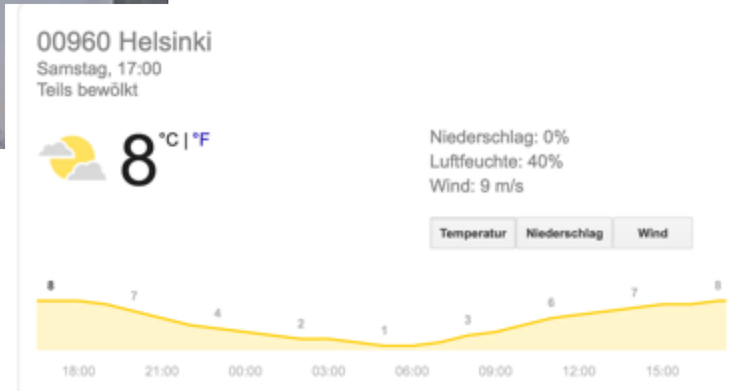
loss function

hypothesis space

Data Point = “Some Ski-Day Ahead”

features:

- snapshot in the morning
- morning temperature
- weather forecast



label:

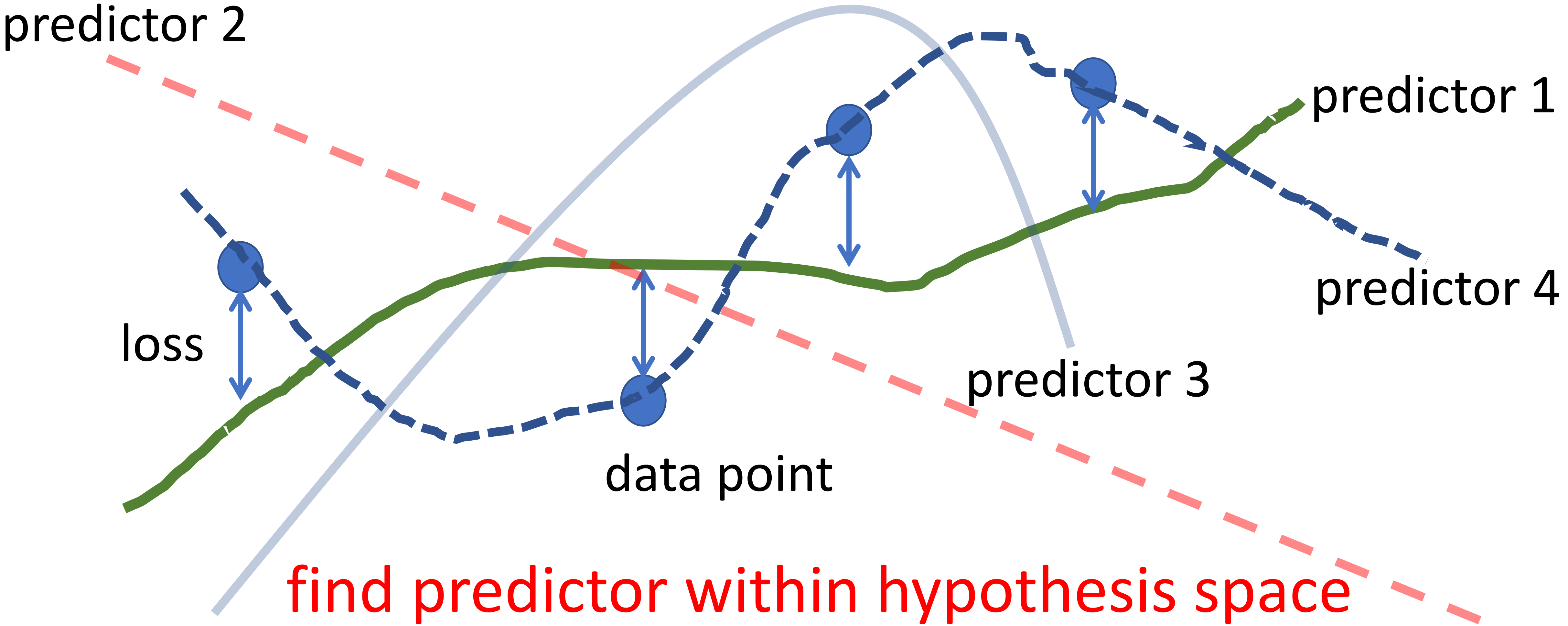
- maximum daytime temperature (important for ski waxing)

Regression = ML with Numeric Labels

label values (predictions) can be compared by distance measures

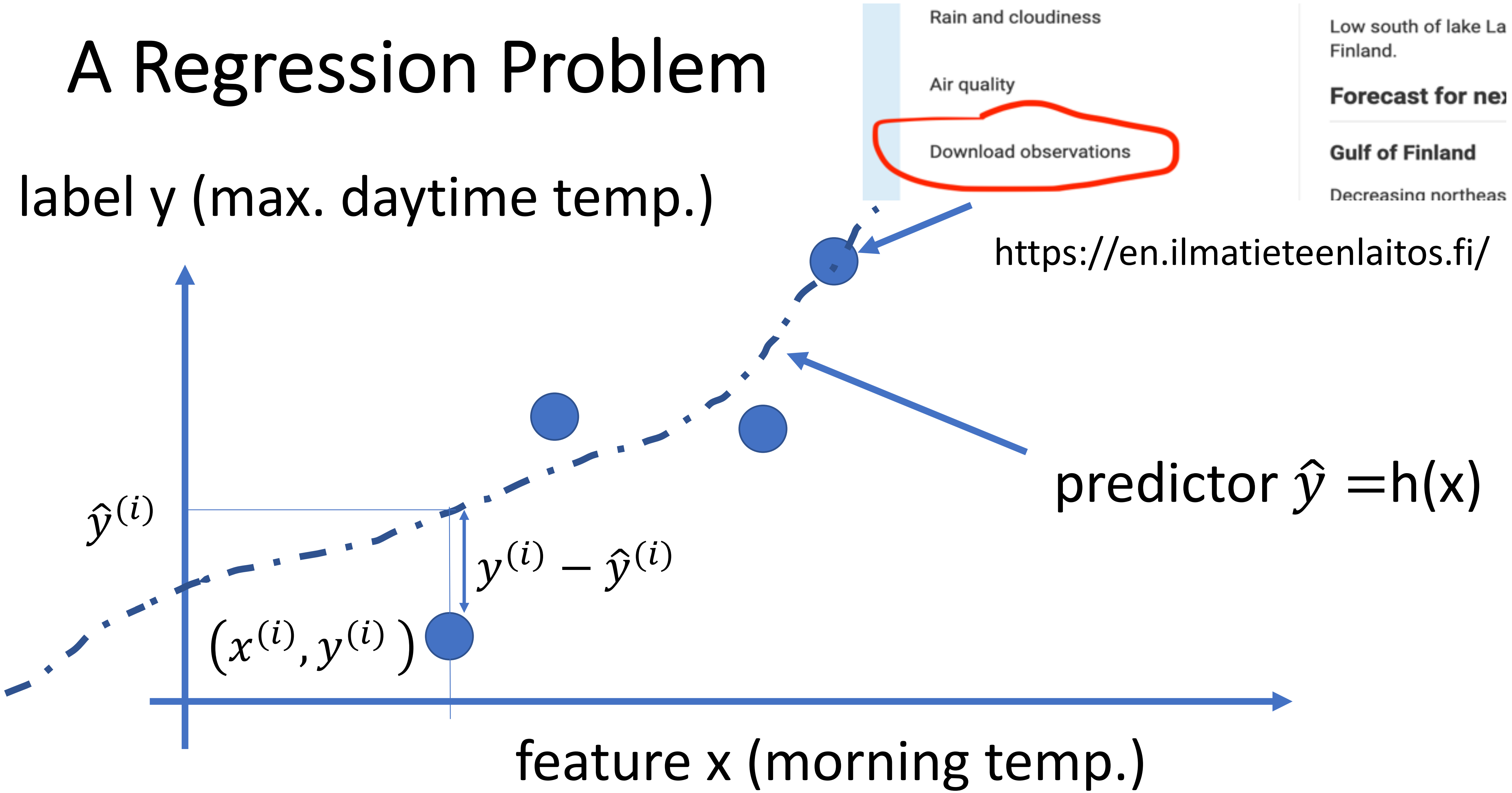
prediction 10 is closer to label value $y=11$ than the prediction 100

Machine Learning ≈ Fitting Models to Data



find predictor within hypothesis space
such that average loss is smallest

A Regression Problem



Get Some Weather Data (by Roope Tervo)

tervo Added simple zarr example for SILAM AWS data share 343426e on 2 Jul 2019

1 contributor

1005 lines (1005 sloc) | 205 KB

Raw Blame History

This short example show how to get data from FMI Open Data multipointcoverage format. The format is used in INSPIRE specifications and is somewhat complex. Anyway, it's the most efficient way to get large amounts of data.

Here we fetch all observations from Finland during two days.

This example is for "old" format WFS2. You may try to use new WFS3 beta service as well. It's available in: http://beta.fmi.fi/data/3/wfs/sofp/

In [7]:

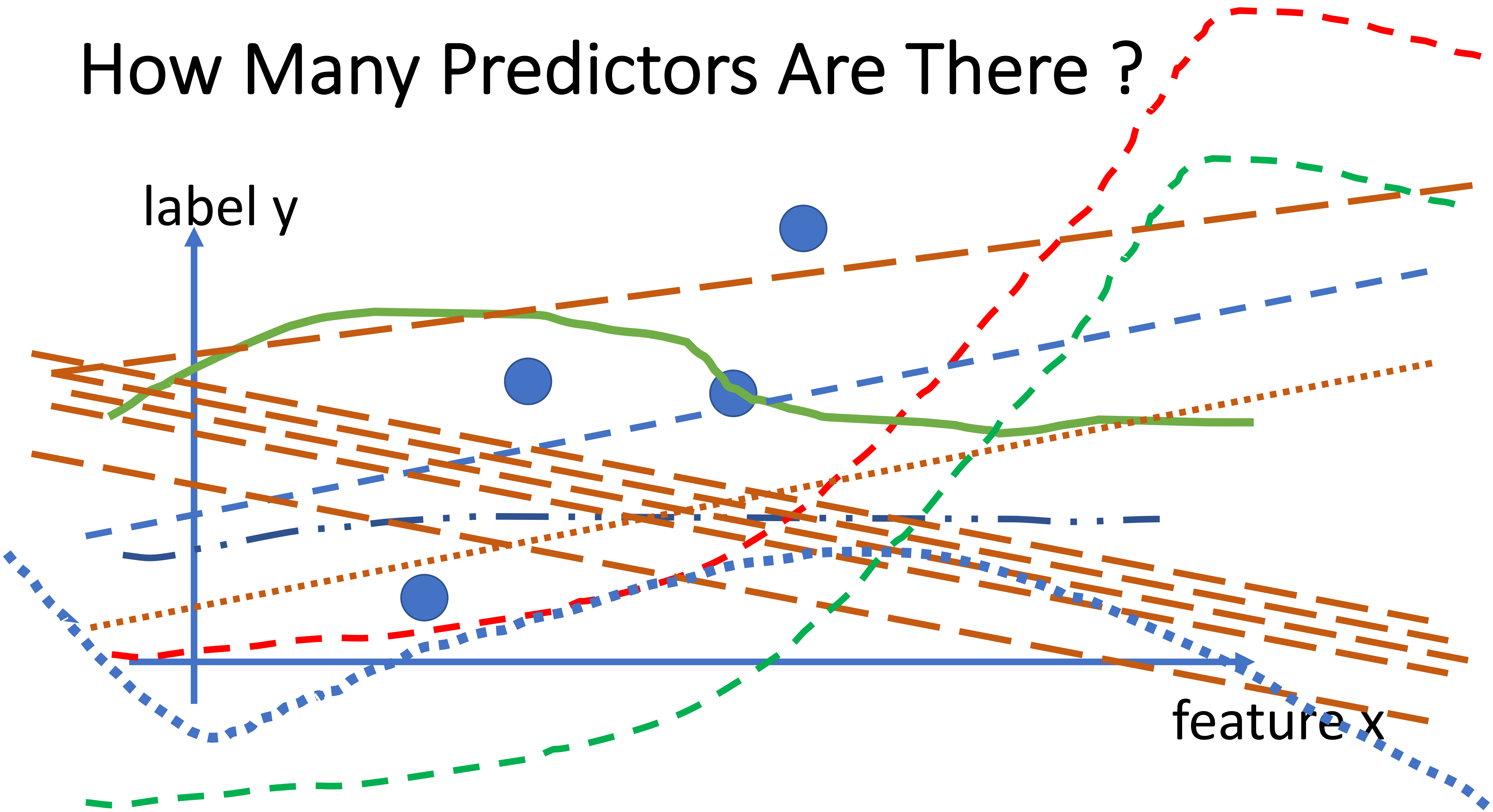
```
import requests
import datetime as dt
import xml.etree.ElementTree as ET
import numpy as np
import re
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
from matplotlib import colorbar, colors
```

Required functions to get param names. Param keys are in the response document but longer names along with other metadata need to be fetched separately.

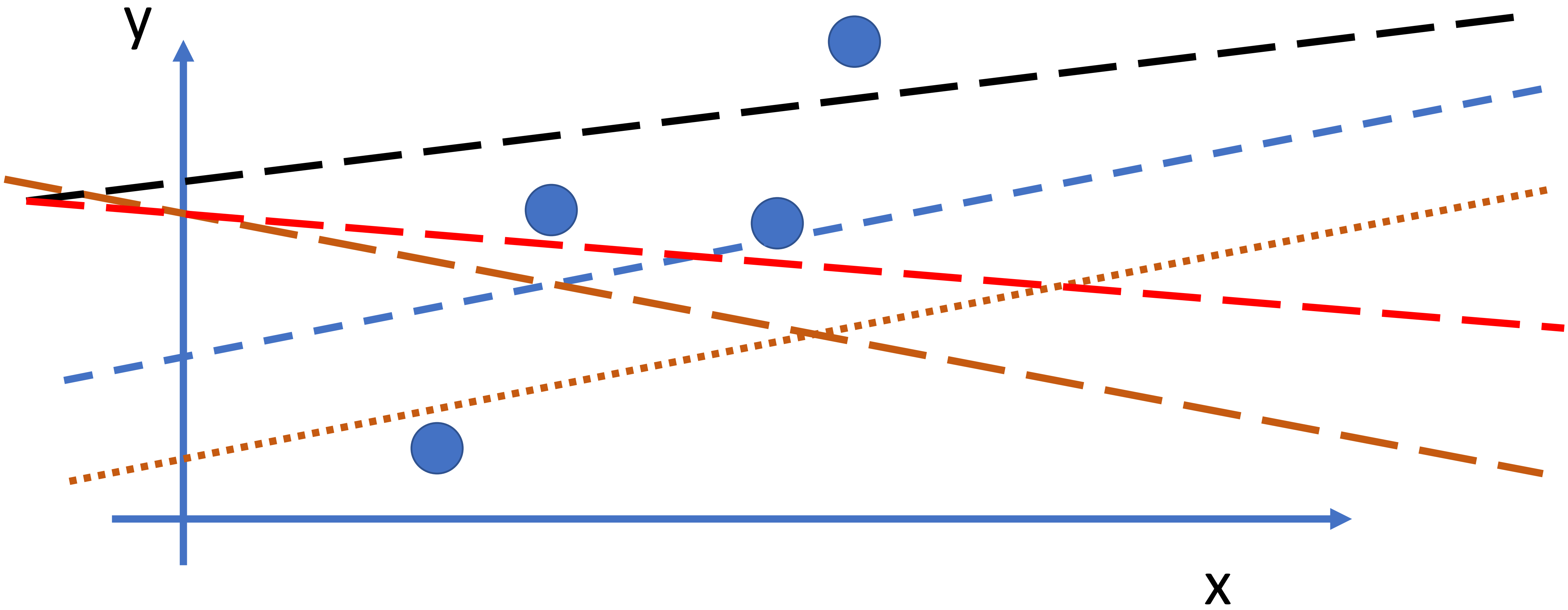
https://github.com/fmidev/opendata-resources/blob/master/examples/python/FMI_WFS2_getobs_multipointcoverage_example.ipynb

Hypothesis Spaces for Regression

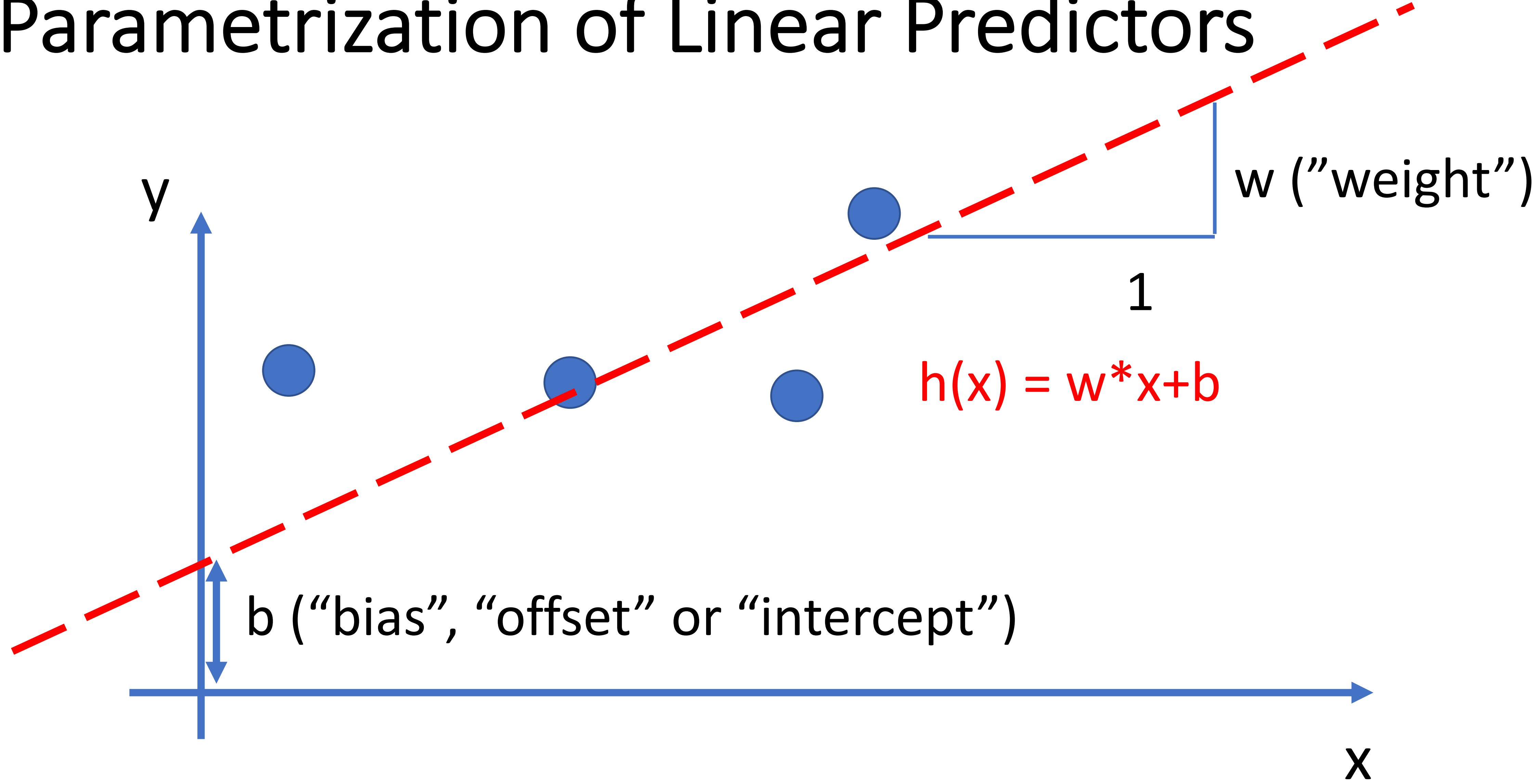
How Many Predictors Are There ?



Restrict to Linear Predictors



Parametrization of Linear Predictors



each linear predictor determined by two numbers, w and b !

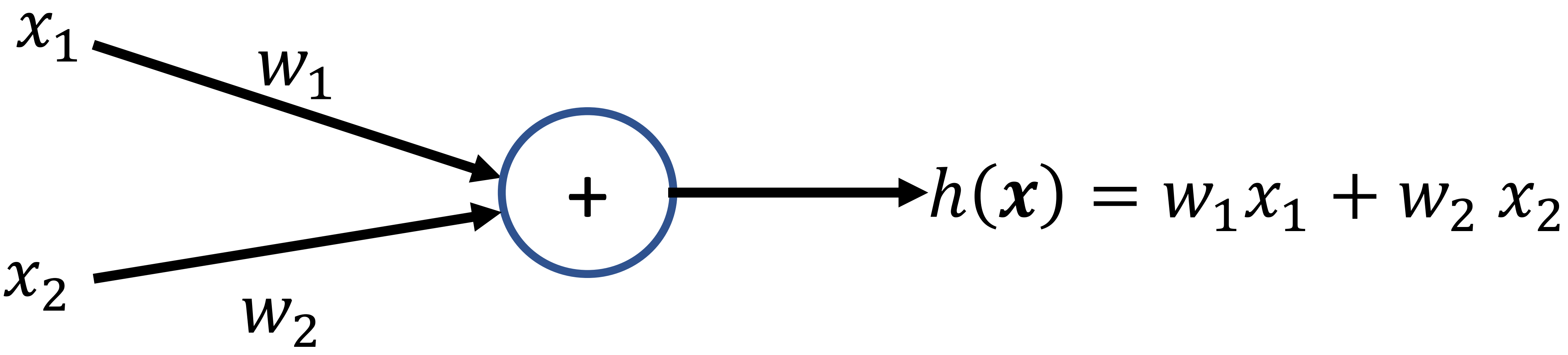
Linear Predictors

- subset of predictors having the form h(x) = w*x+b
- each linear predictor determined by numbers w and b
- still infinitely many predictor maps!
- but we can handle them via numbers (nice)
- linear predictors structured like Euclidean space (super nice !)

Linear Predictors with More Features

- data point with n different features x_1,…, x_n
- stack into vector (1D numpy array) **x**=(x_1,…, x_n)
- linear predictor h(x) = w_1 * x_1 +…. w_n * x_n $+$ b
- number of features (and weights) can be billions !

The Weights in a Linear Predictor



weight w_1 determines influence of x_1 on prediction h(x)

weight w_2 determines influence of x_2 on prediction h(x)

Linear Predictors in Python

predictor h(x) = $\sum_{j=1}^{n} w_j \, x_j + b$ represented by Python object

sklearn.linear_model.LinearRegression

class `sklearn.linear_model.`**`LinearRegression`**(*fit_intercept=True, normalize=False, copy_X=True, n_jobs=None*) [source]

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

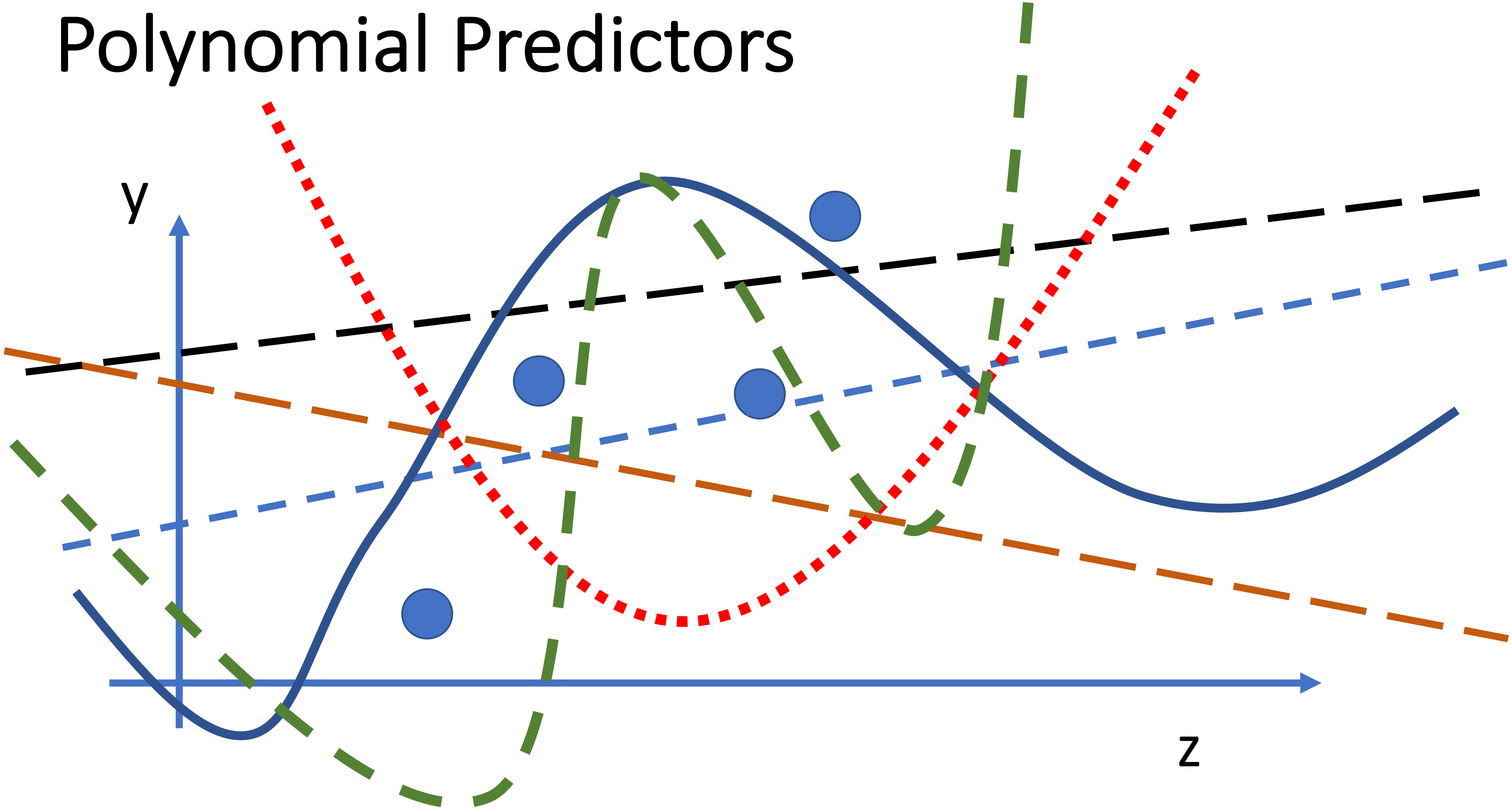
weights **w** stored in the attribute “LinearRegression.coeff_”

bias b stored in “LinearRegression.**intercept_**”

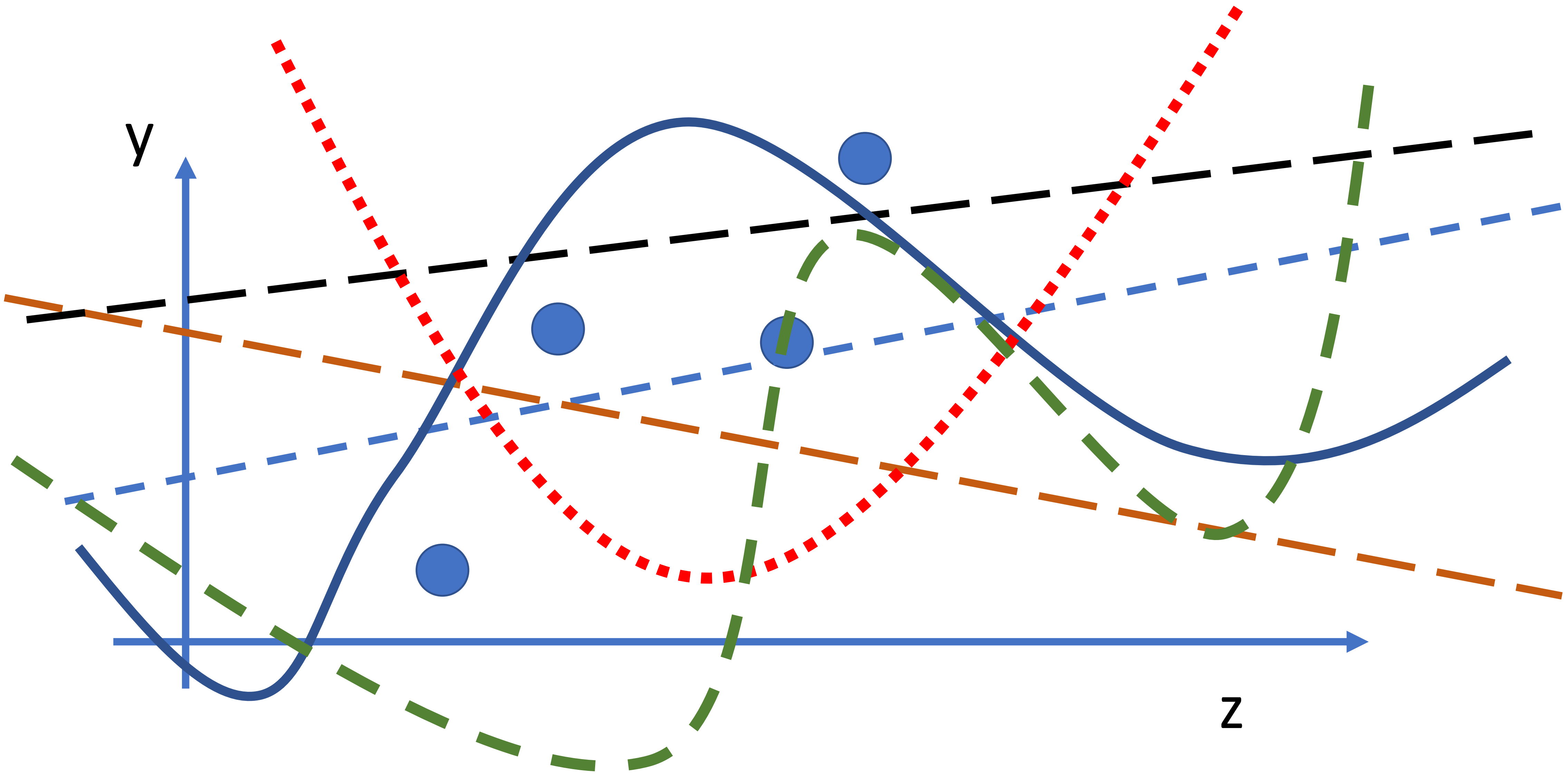
Polynomial Regression

- data point with single numeric feature z
- construct features $x_1 = z^0, \ldots, x_n = z^{n-1}$
- linear predictor h(x) = $w_1 * x_1$ +.... $w_n * x_n$
- predictor function is polynomial in z !

Polynomial Predictors

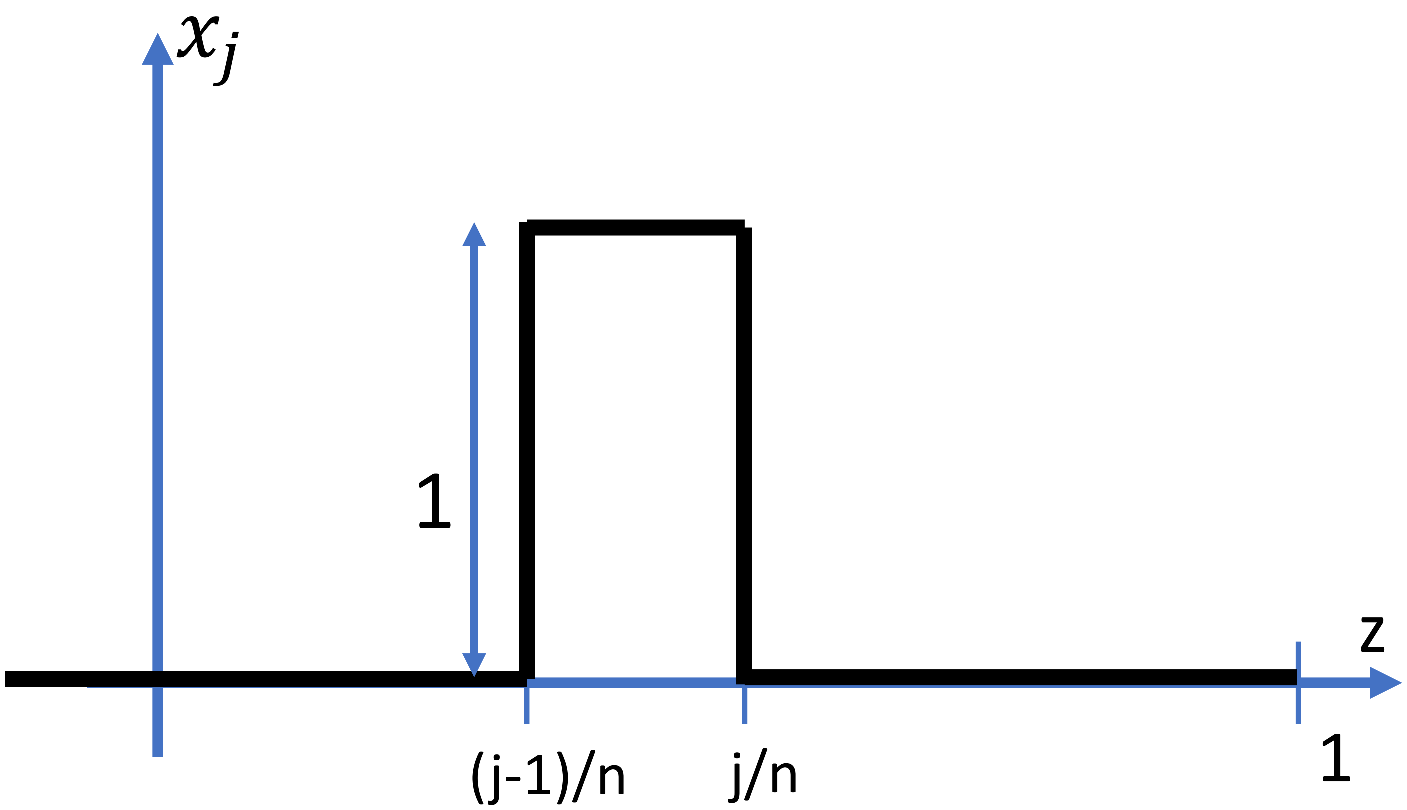


How Many Data Points Can We Fit Perfectly?



You Can Do Anything with Linear Predictors!

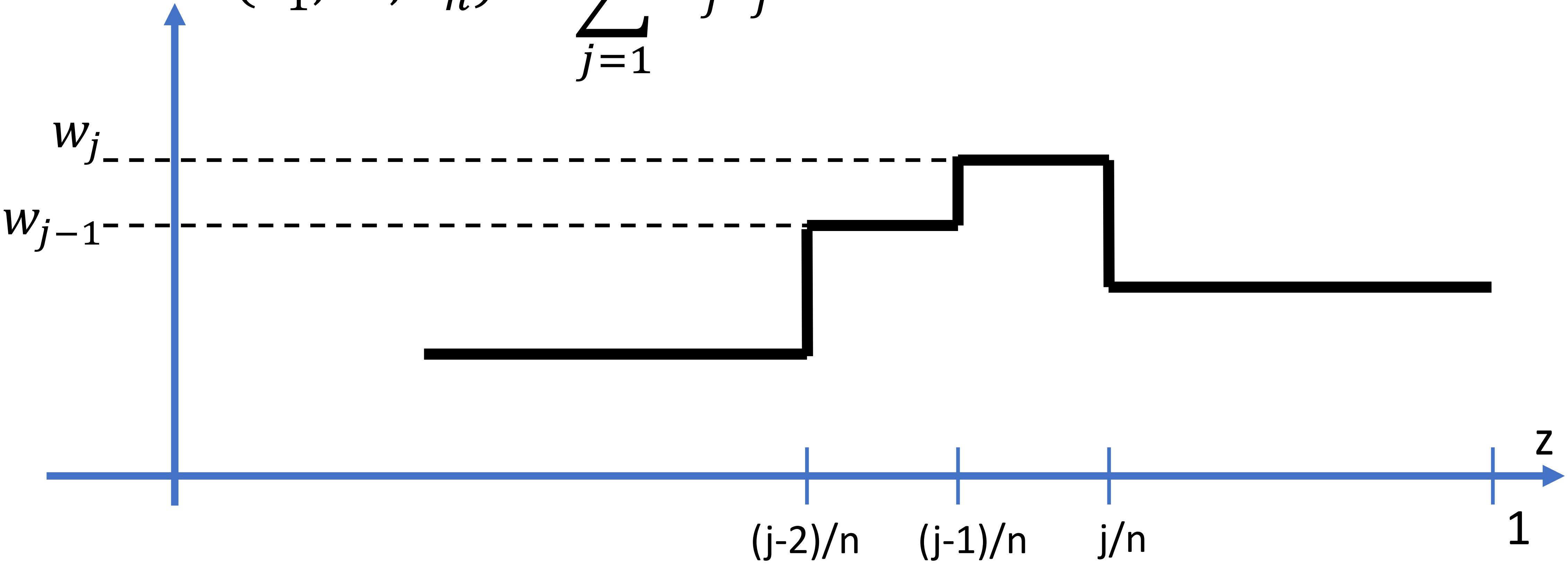
- consider data points with single numeric feature z
- construct new features $x_1, \ldots, x_n$
- $x_j = \begin{cases} 1\ for\ \frac{j-1}{n} \leq z \leq \frac{j}{n} \\ 0\ for\ all\ other\ z. \end{cases}$



You Can Do Anything with Linear Predictors!

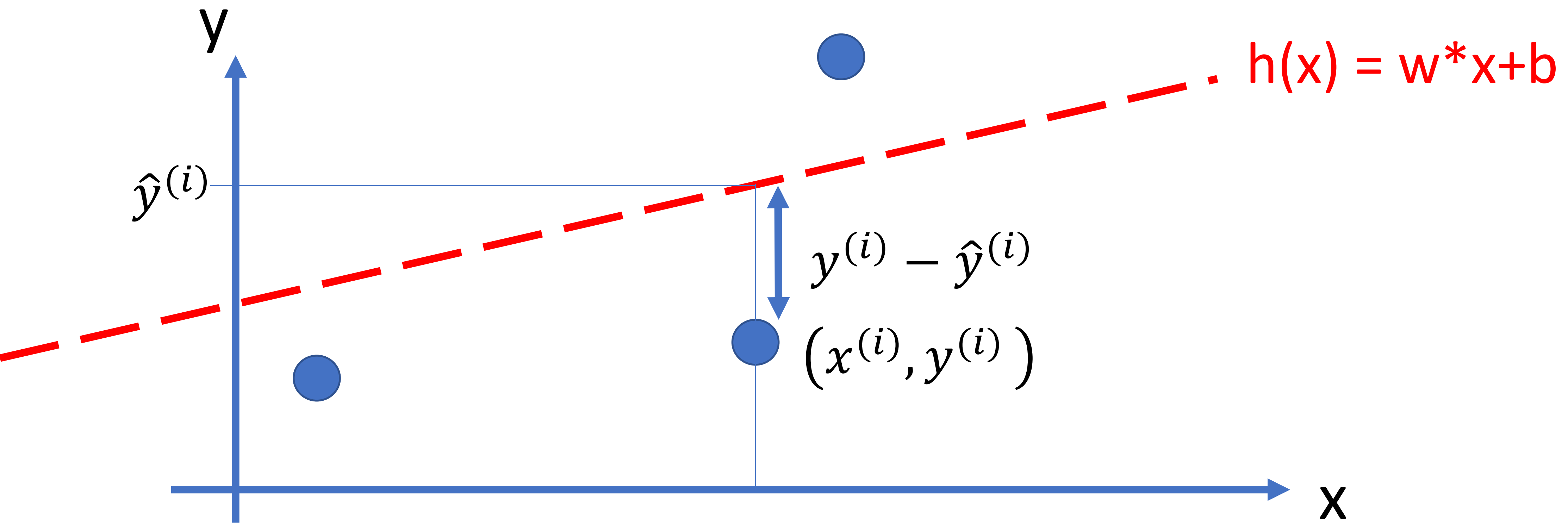
- linear predictor in new features is non-linear in z!

$$h(x_1, \ldots, x_n) = \sum_{j=1}^{n} w_j x_j$$



How to Learn a Good Linear Predictor?

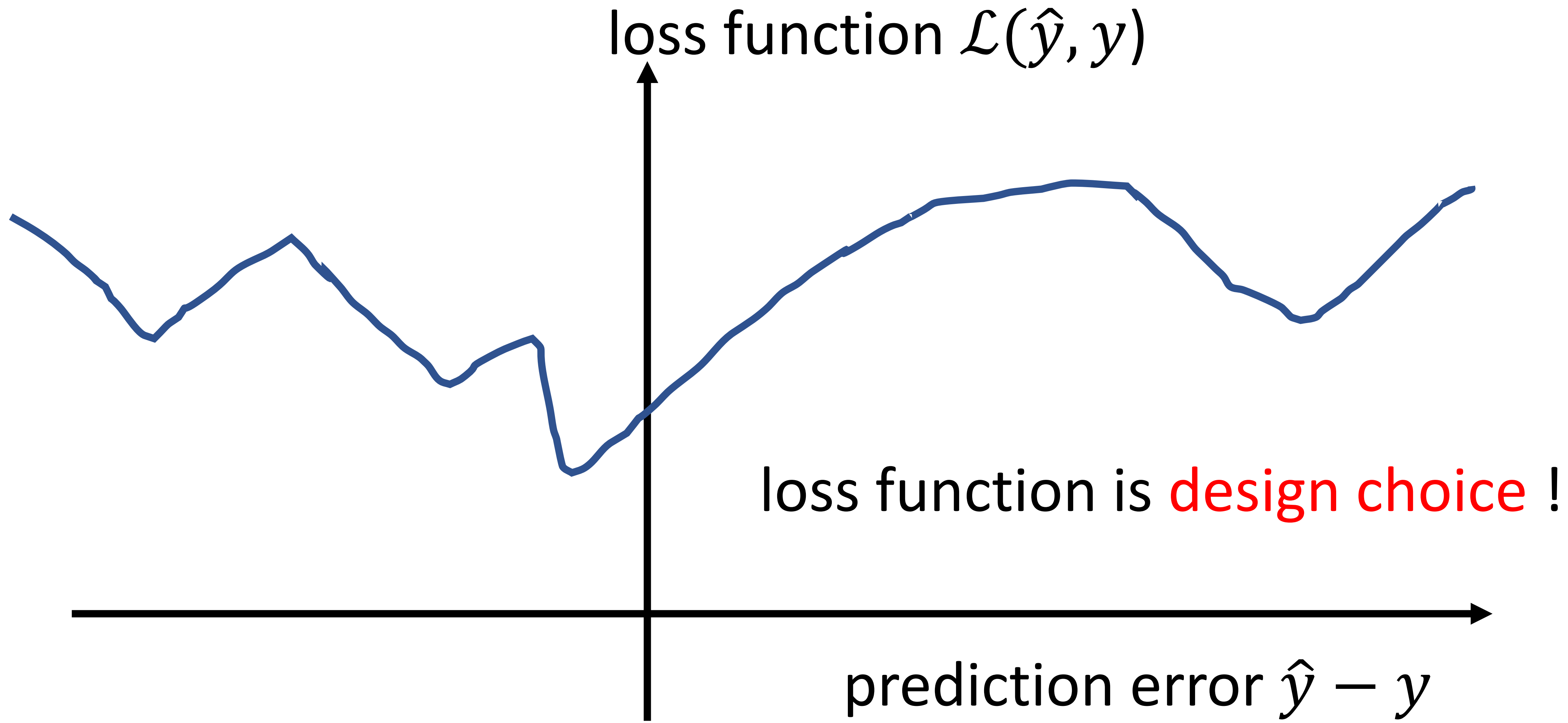
Learning a Linear Predictor



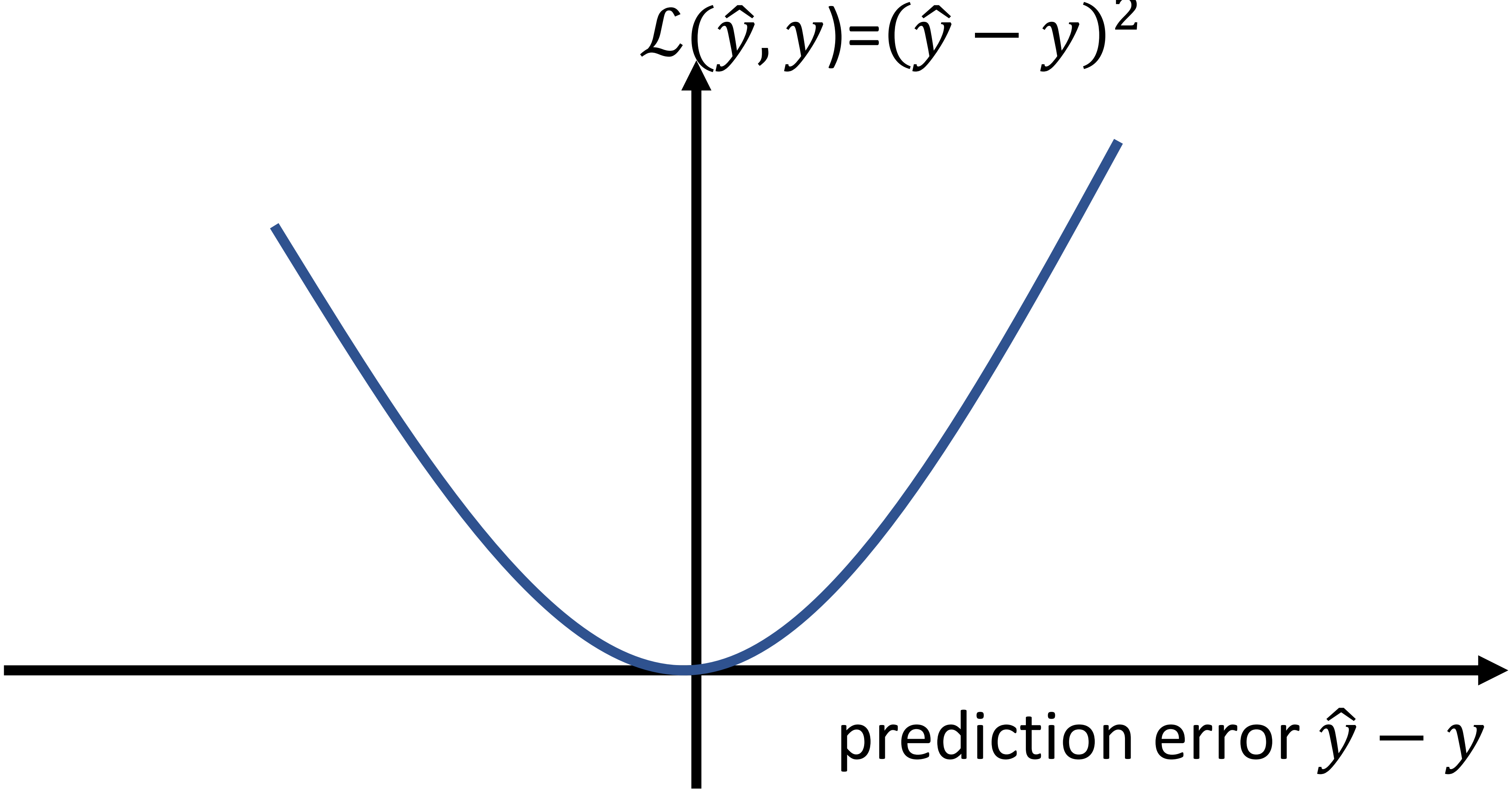
choose w,b to minimize average "size" of prediction errors $y^{(i)} - \hat{y}^{(i)}$

can be done with method "LinearRegression.**fit()**"

Measuring Error Size via Loss Functions



The Squared Error Loss



ID-Card of Linear Least Squares Regression

- features: real numbers
- labels: numeric (typically modelled as real number)
- hypothesis space: linear predictor maps
- loss: squared loss
- instance of a linear regression method

ID-Card of Polynomial Regression

- features: real numbers
- labels: numeric (typically modelled as real number)
- hypothesis space: polynomial predictor maps
- loss: squared loss
- instance of a linear regression method

The Squared Error Loss – Pros and Cons

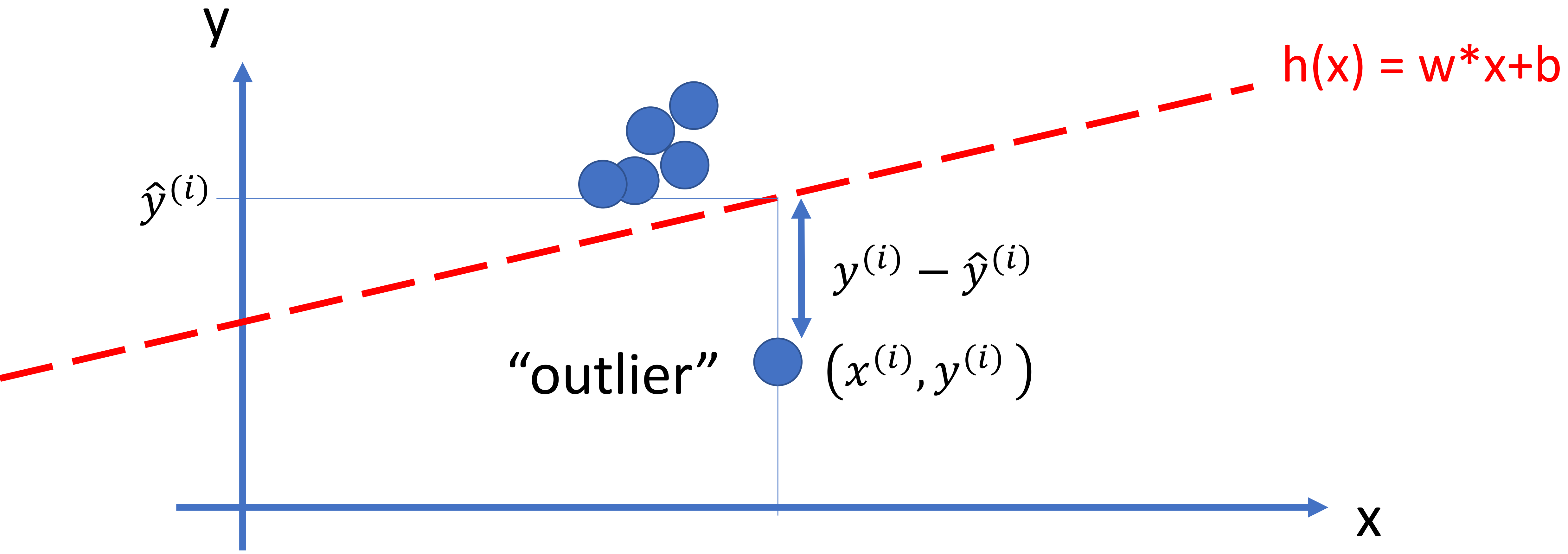
☺ smooth convex optimization problem for linear predictors

☺ scalable optimization algorithms can handle big data

☺ statistically optimal for Gaussian features and label

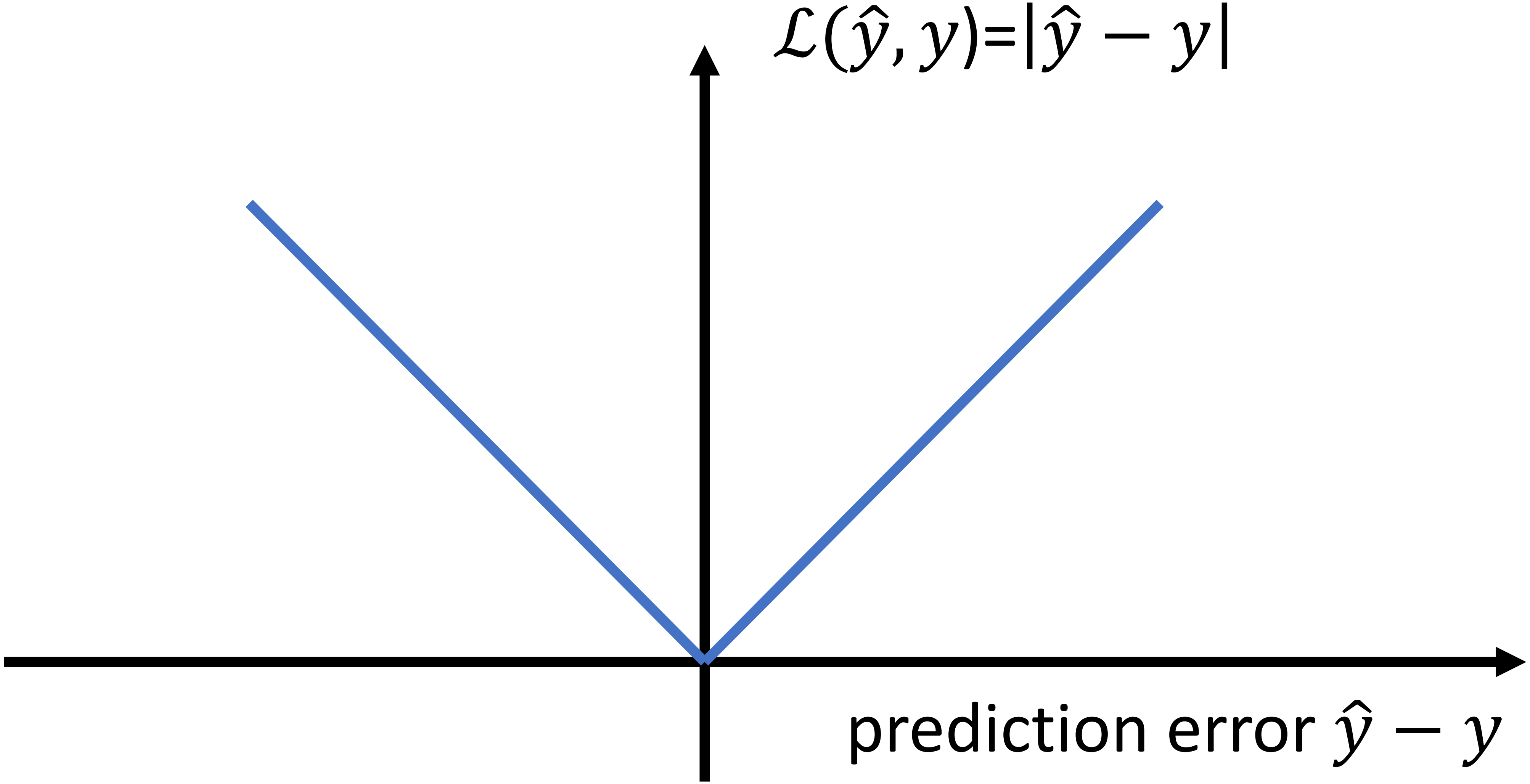
☹ sensitive to outliers

Squared Error Loss Sensitive to Outliers

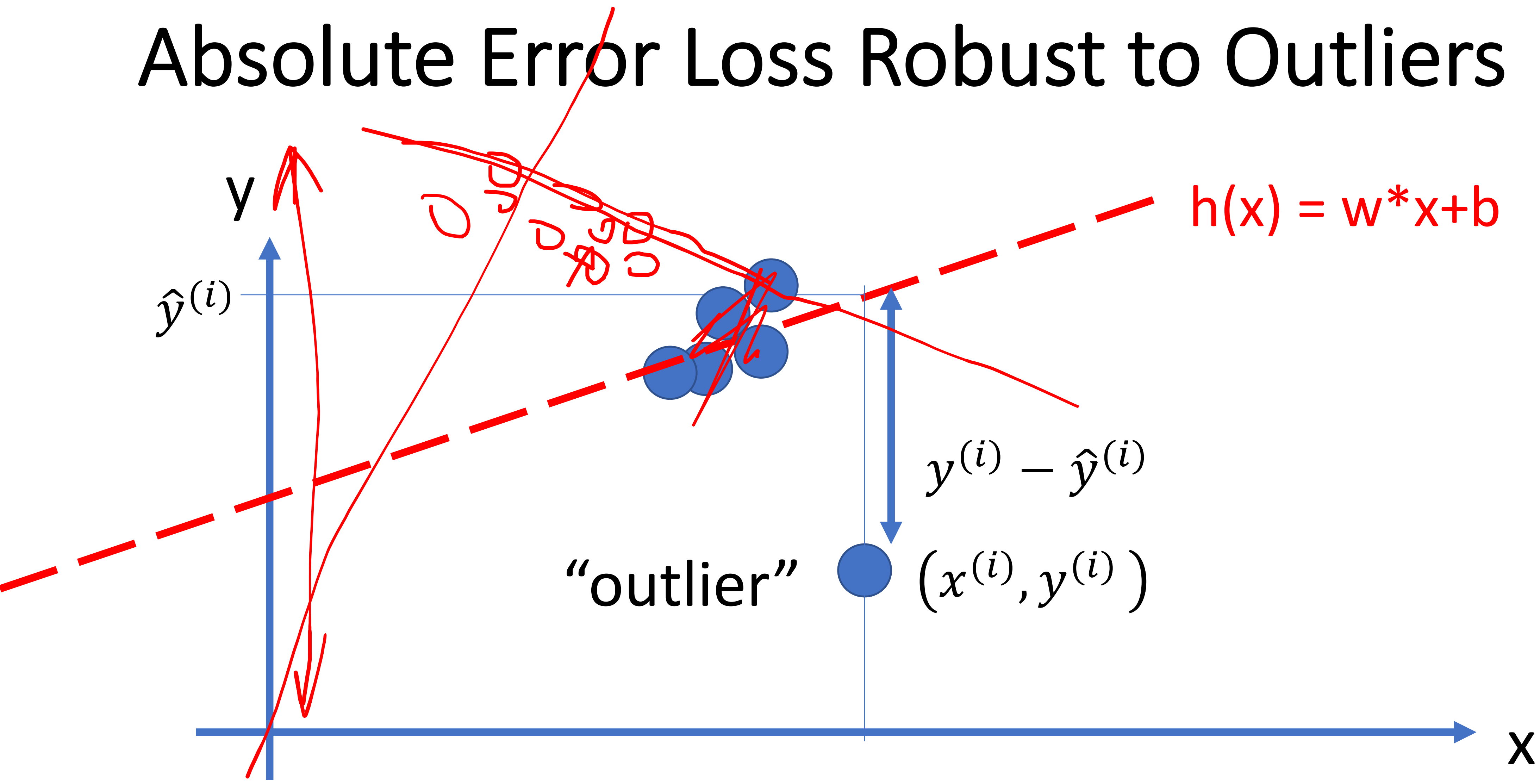


min. squared error loss forces predictor towards outlier

The Absolute Error Loss



Absolute Error Loss Robust to Outliers



absolute error loss "tolerates" larger error for outlier

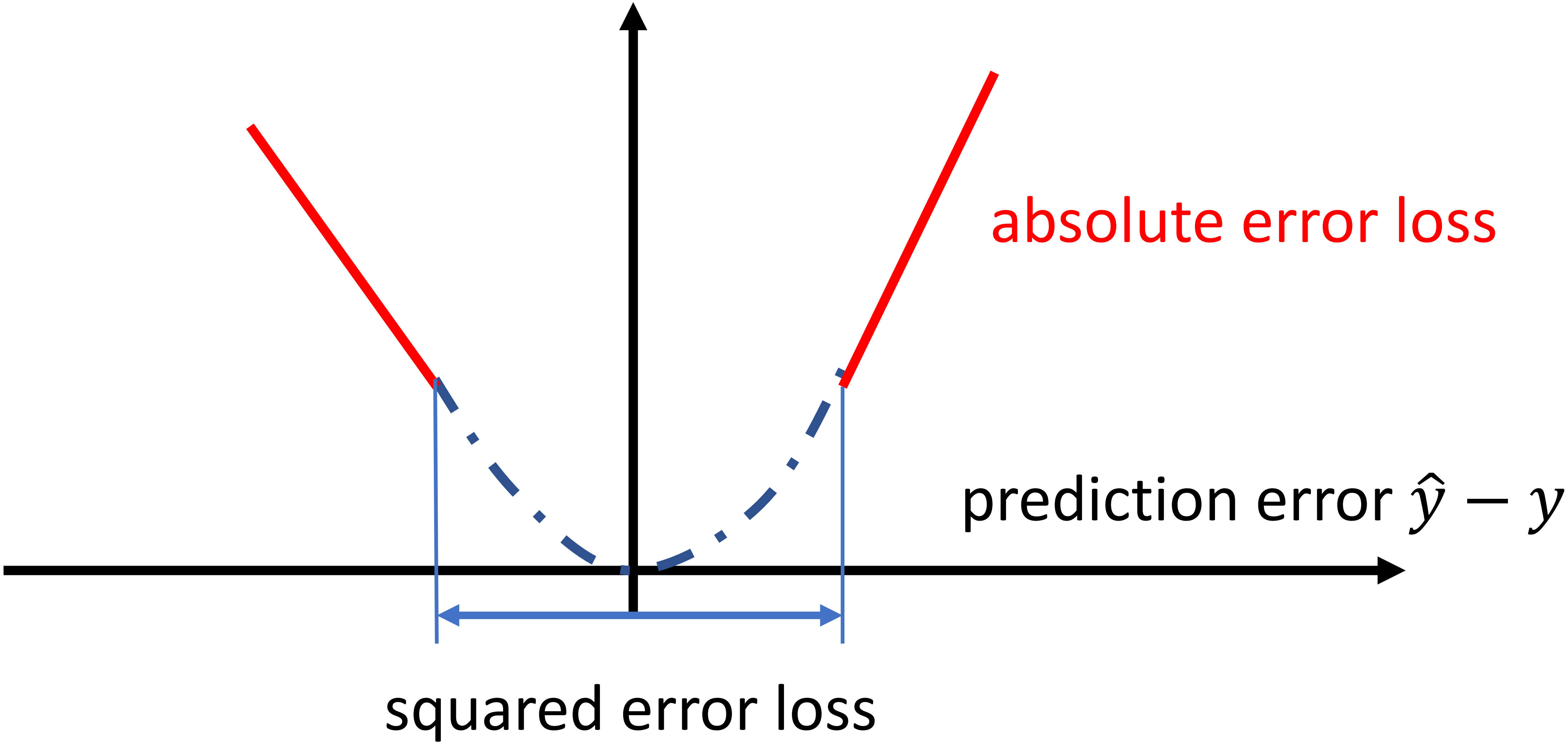
ID-Card of Mean Absolute Error Regression

- features: real numbers
- labels: numeric (typically modelled as real number)
- hypothesis space: linear predictor maps
- loss: absolute error loss
- instance of a linear regression method

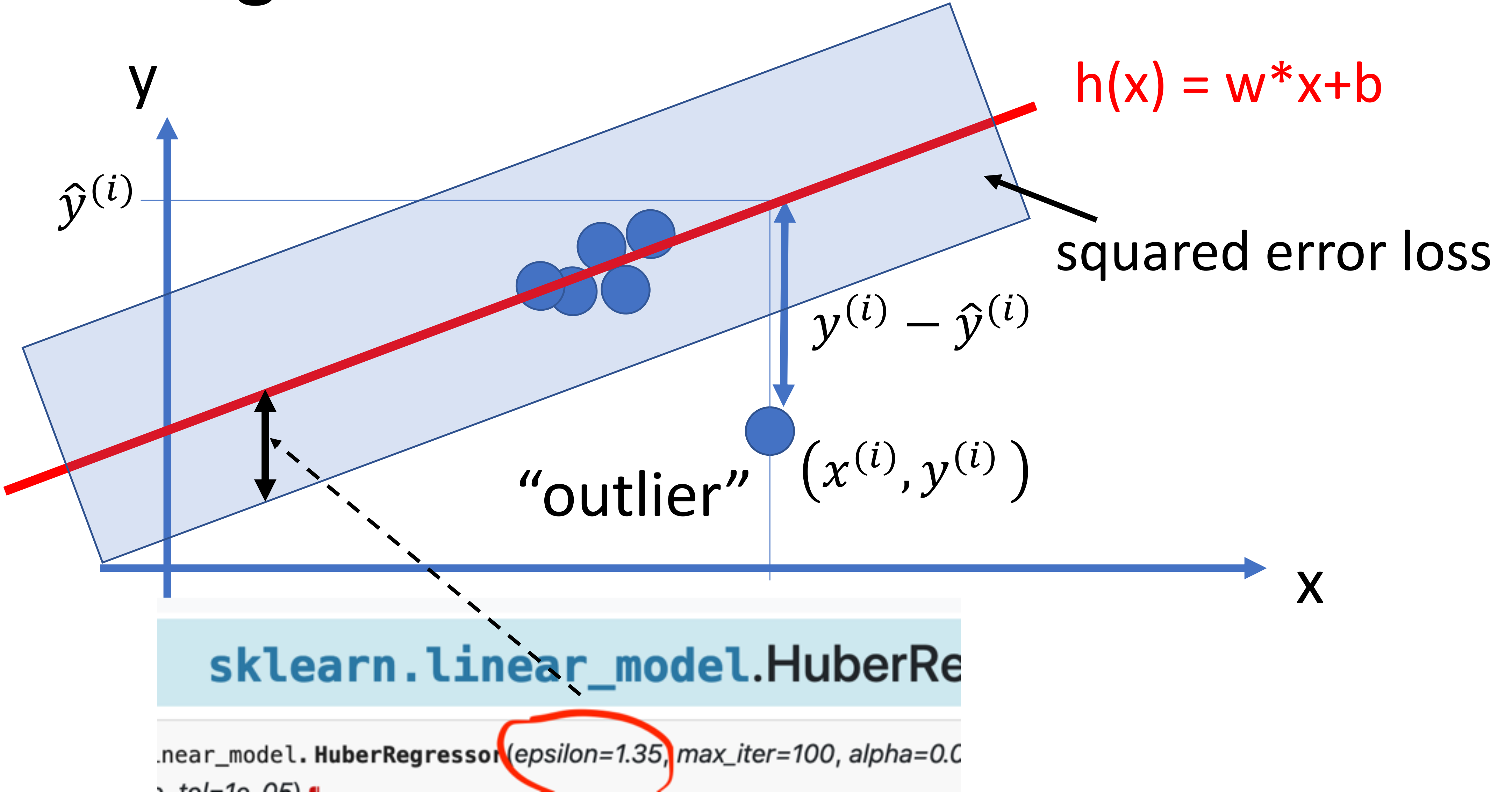
The Absolute Error Loss – Pros and Cons

- ☺ robust to outliers
- ☹ non-smooth optimization problem
- ☹ computationally more challenging to find best linear predictor

Best of Both Worlds - Huber Loss



Fitting Linear Predictor with Huber Loss



So What?

- regression methods use numeric labels
- numeric labels allows to measure distances
- loss functions measure size of error (distance)
- squared error loss computationally attractive
- absolute error loss robust to outliers

Machine Learning
With Python

FITech
NETWORK UNIVERSITY

Thank You !