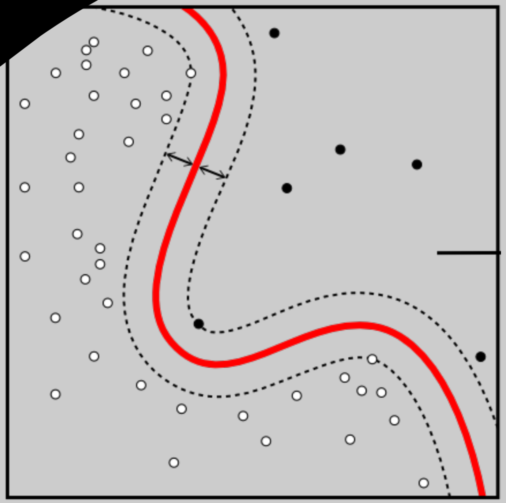
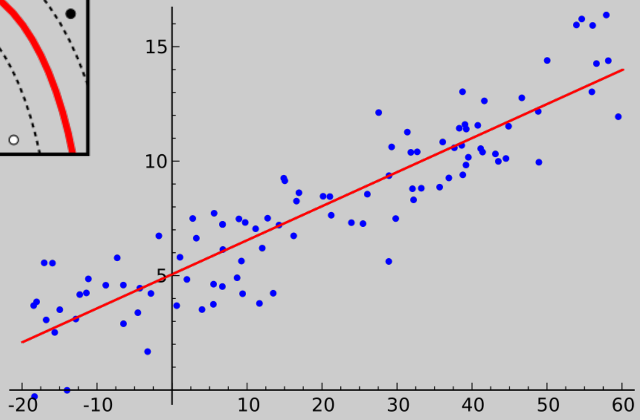
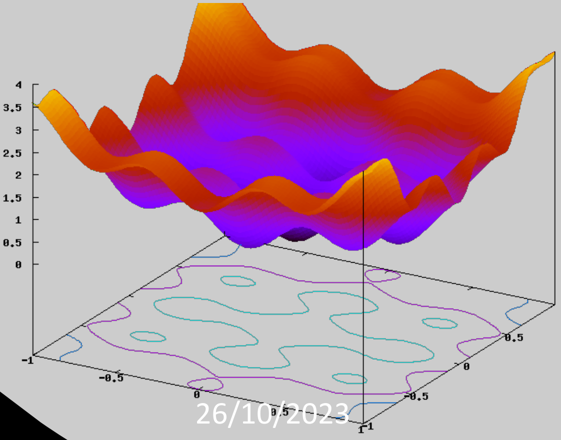






B1:Engineering Computation – Project MT23

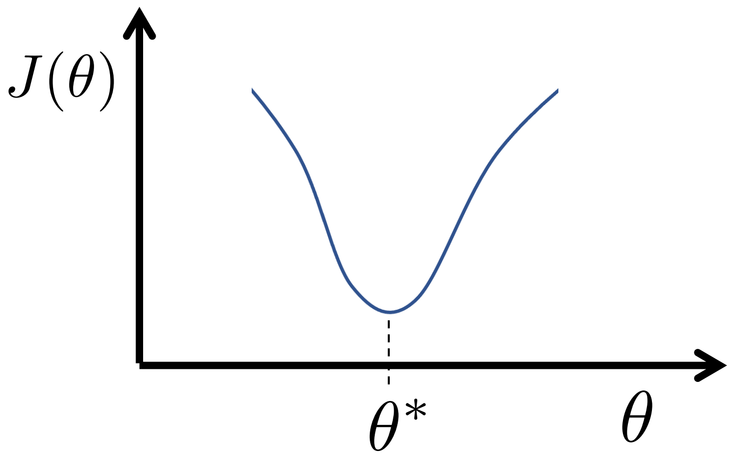
Optimization for Regression and Classification models

Optimization:

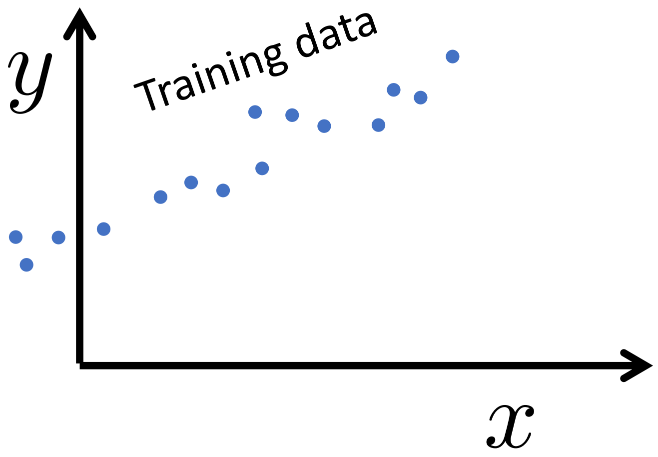
Methods for finding the **minimum** or **maximum** value of a function
(and the parameters θ that min/maximize it)

$$\min_{\theta} \mathcal{J}(\theta)$$

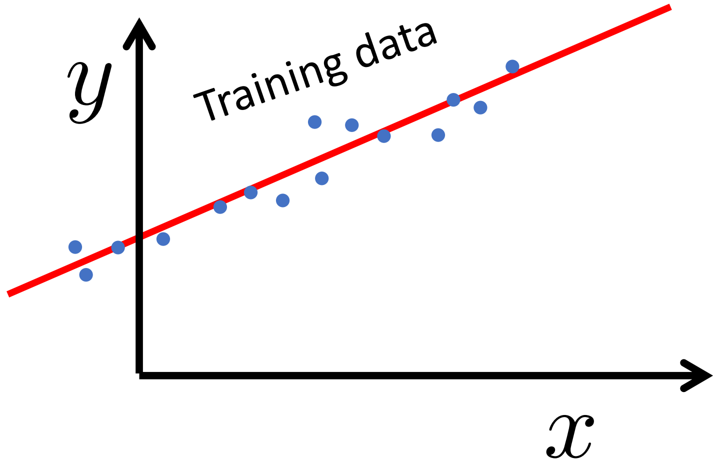
Cost function,
a.k.a. Objective function
(sometimes "Loss" function)



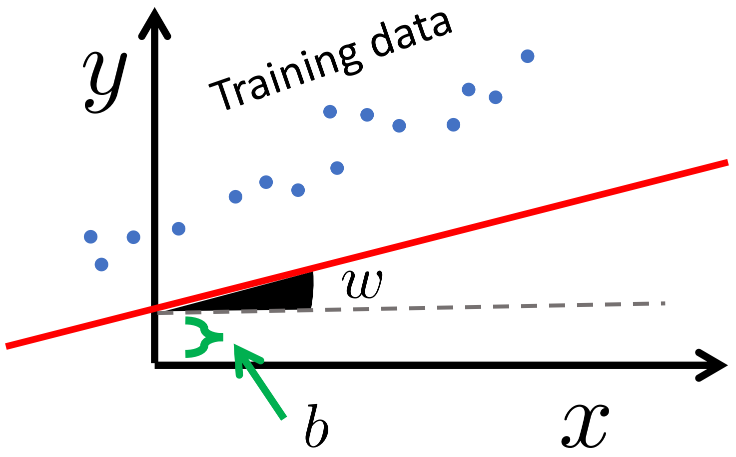
Example: Optimization of a Regression model



Example: Optimization of a Regression model



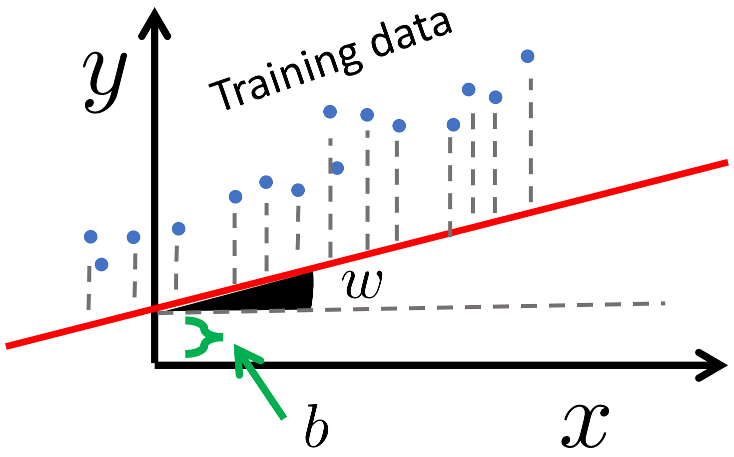
Example: Optimization of a Regression model



$$\hat{y} = wx + b$$
$$\theta = \{w, b\}$$

Non-optimized model
with parameters θ

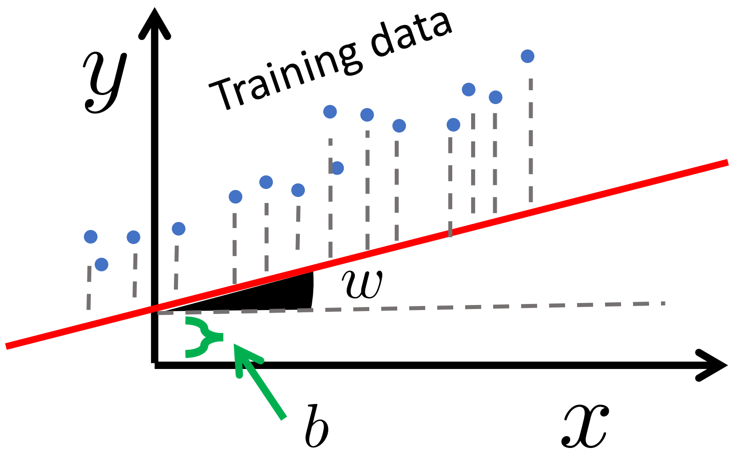
Example: Optimization of a Regression model



$$\hat{y} = wx + b$$
$$\theta = \{w, b\}$$

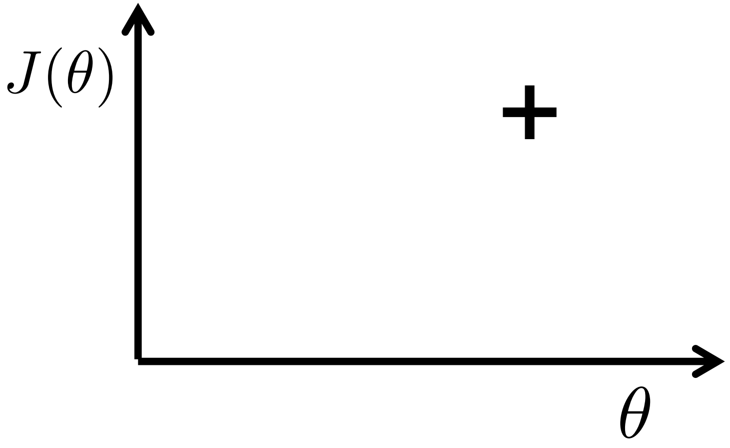
Non-optimized model
with parameters θ

Example: Optimization of a Regression model



$$\hat{y} = wx + b$$
$$\theta = \{w, b\}$$

Non-optimized model with parameters θ

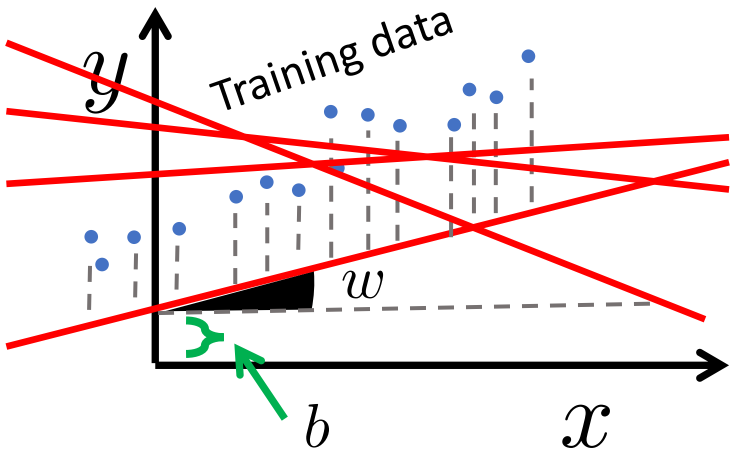


Cost Loss

$$J(\theta) = \frac{1}{N} \sum_{x,y} \mathcal{L}(\hat{y}_\theta(x), y)$$
$$= \frac{1}{N} \sum_{x,y} (\hat{y}_\theta(x) - y)^2$$

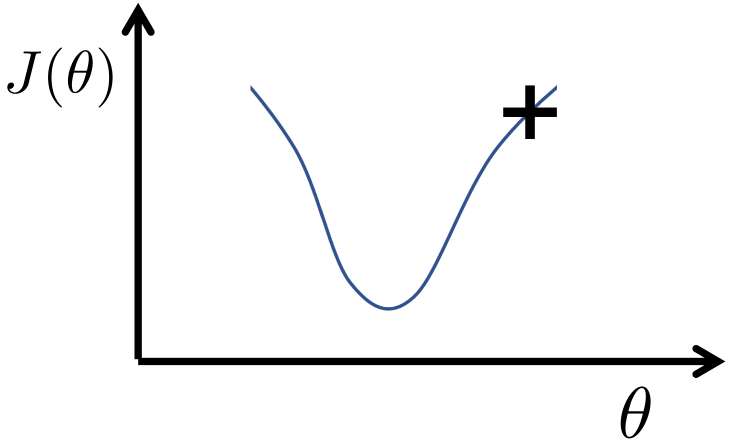
Mean Squared Error (MSE) loss

Example: Optimization of a Regression model



$$\hat{y} = wx + b$$
$$\theta = \{w, b\}$$

Non-optimized model
with parameters θ



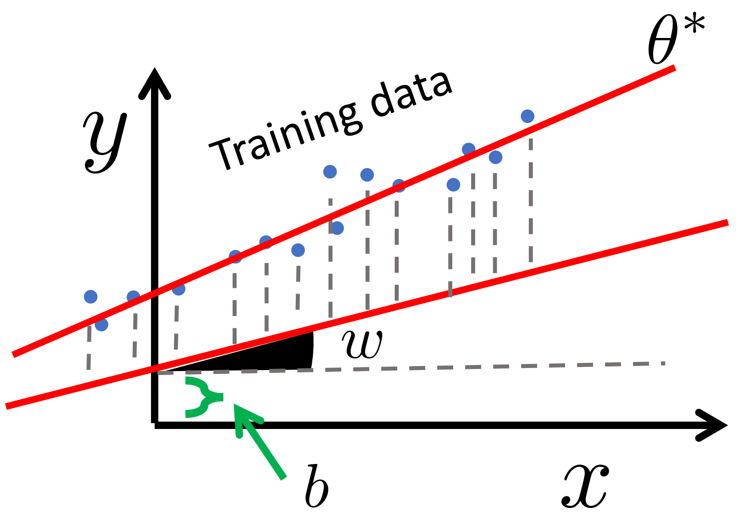
Cost

Loss

$$J(\theta) = \frac{1}{N}\sum_{x,y} \mathcal{L}(\hat{y}_\theta(x), y)$$
$$= \frac{1}{N}\sum_{x,y} (\hat{y}_\theta(x) - y)^2$$

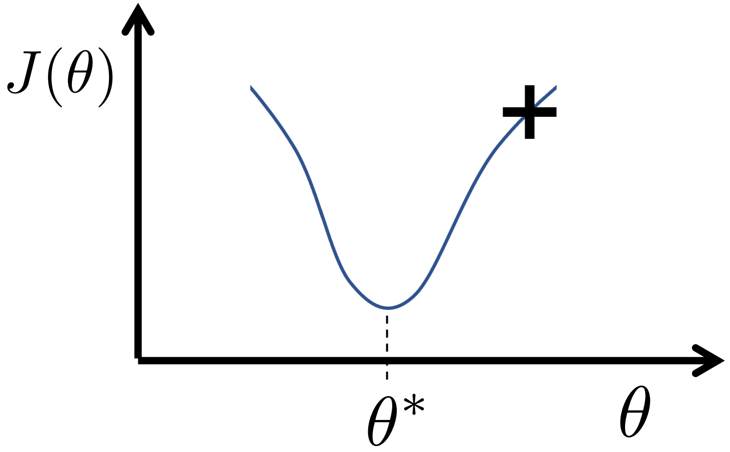
Mean Squared Error (MSE)
loss

Example: Optimization of a Regression model



$$\hat{y} = wx + b$$
$$\theta = \{w, b\}$$

Non-optimized model
with parameters θ



Cost Loss

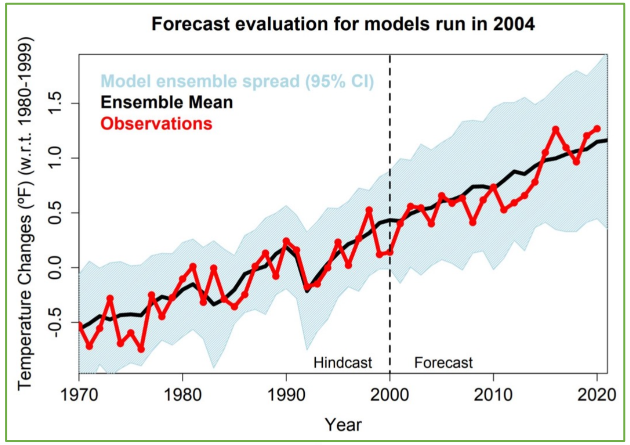
$$J(\theta) = \frac{1}{N} \sum_{x,y} \mathcal{L}(\hat{y}_\theta(x), y)$$

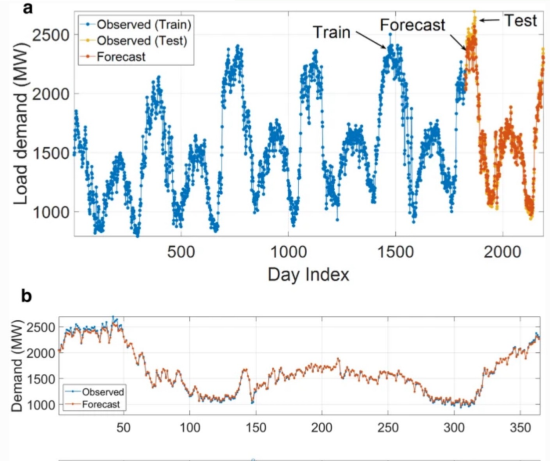
$$= \frac{1}{N} \sum_{x,y} (\hat{y}_\theta(x) - y)^2$$

Mean Squared Error (MSE)
loss

Optimization:
Statistics, Machine Learning, Control, Robotics, AI…

[1]



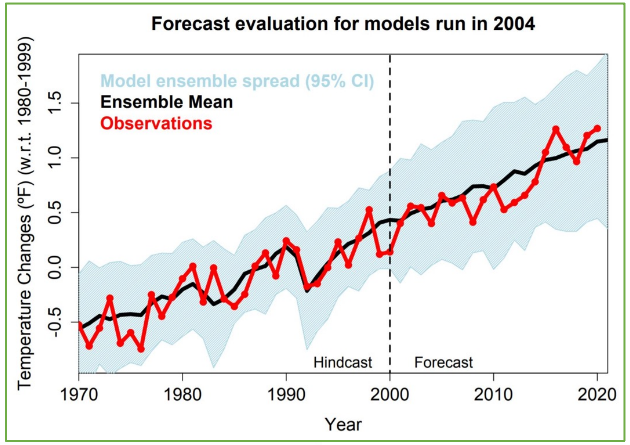


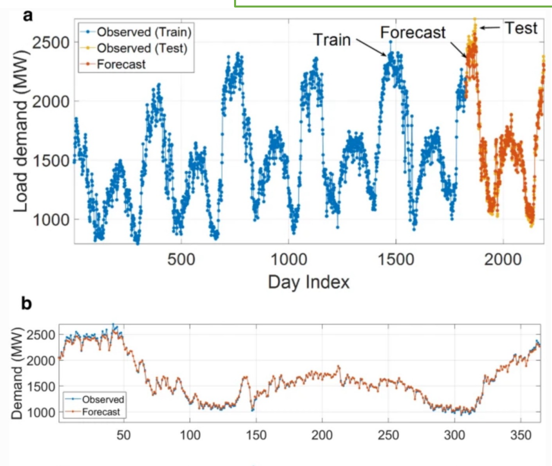
[2]

[1] https://climate.nasa.gov/news/2943/study-confirms-climate-models-are-getting-future-warming-projections-right/
https://climate.nasa.gov/solutions/adaptation-mitigation/
[2] Hamad et al, Deep learning-based load forecasting considering data reshaping using MATLAB\Simulink

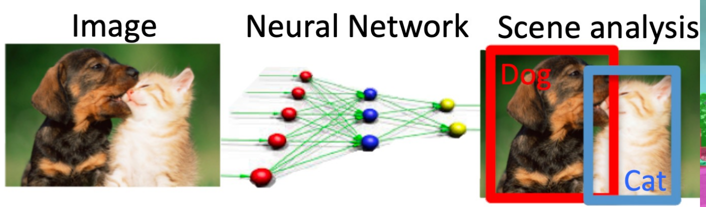
Optimization:
Statistics, Machine Learning, Control, Robotics, AI…

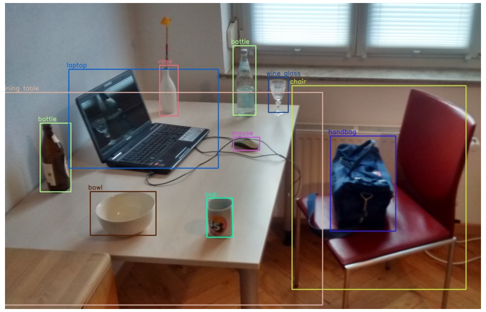
[1]

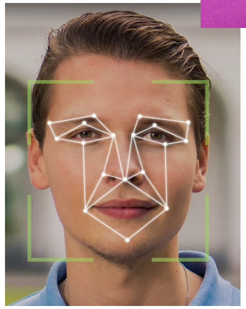


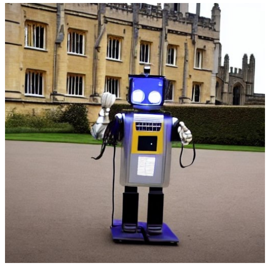


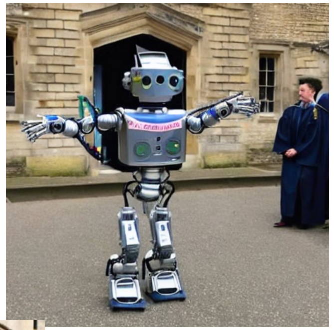
[2]



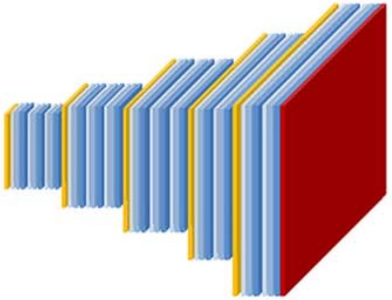


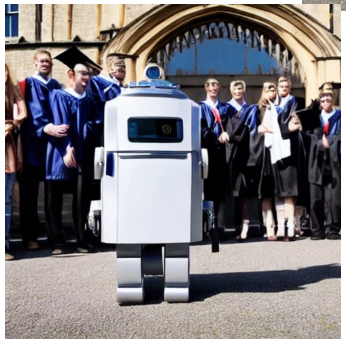


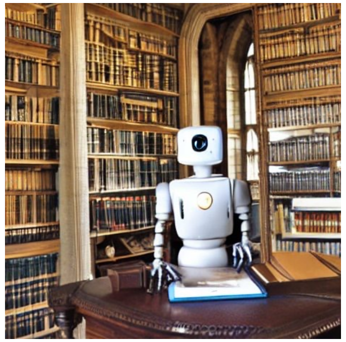




"A robot studying at University of Oxford"



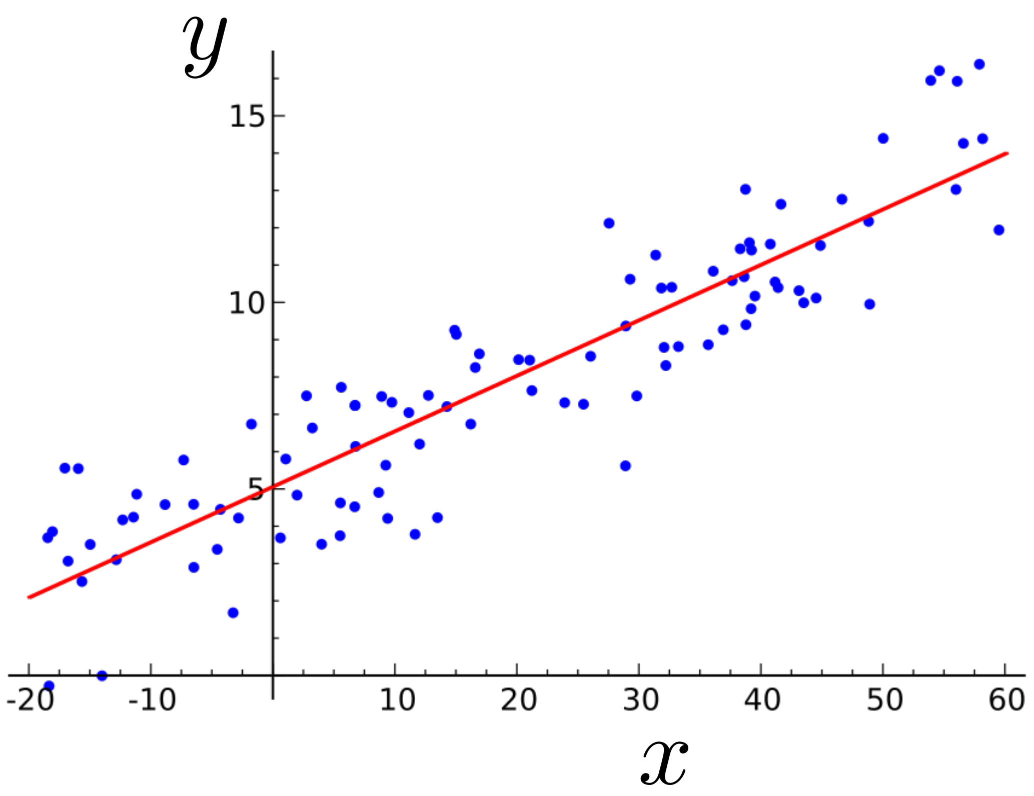




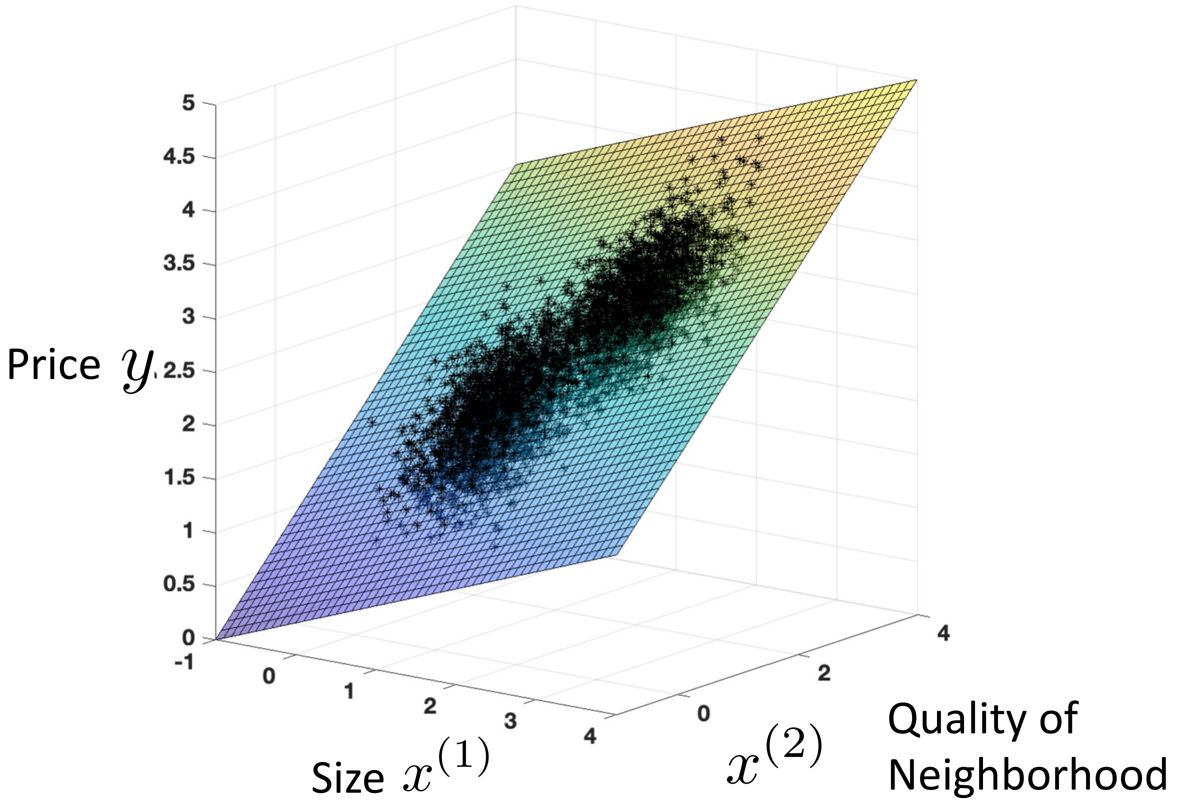
[1] https://climate.nasa.gov/news/2943/study-confirms-climate-models-are-getting-future-warming-projections-right/
https://climate.nasa.gov/solutions/adaptation-mitigation/
[2] Hamad et al, Deep learning-based load forecasting considering data reshaping using MATLAB\Simulink

Tasks

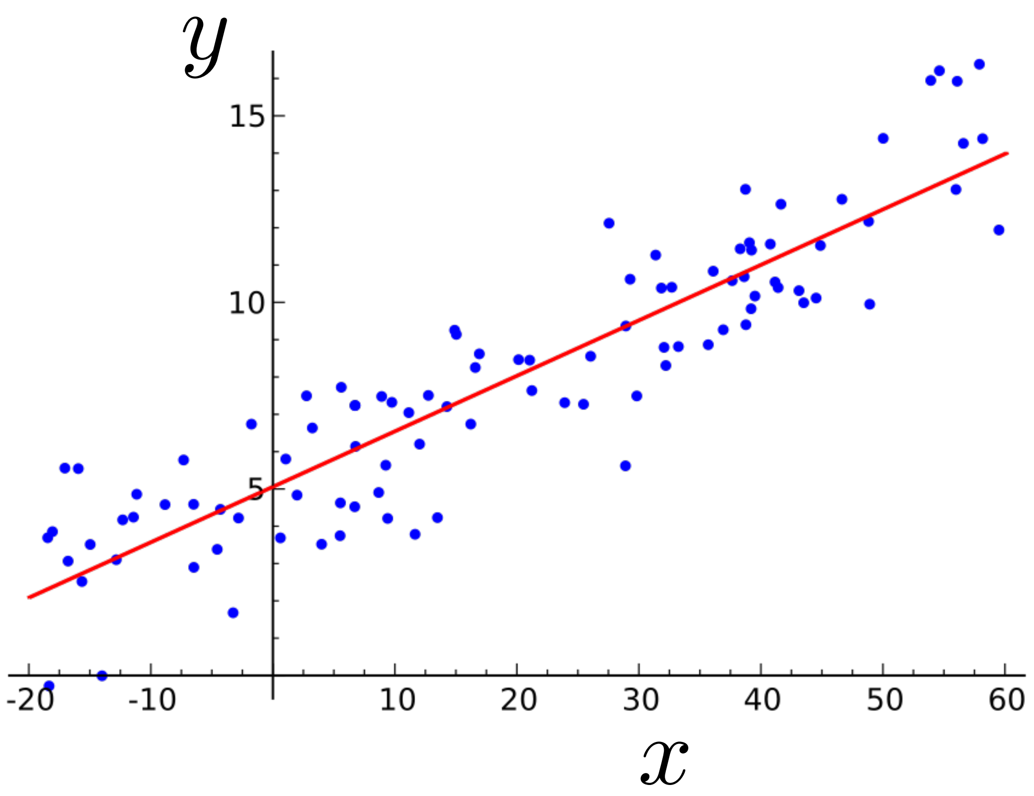
Linear Regression



Example: Price of a house



Closed form solution for MSE



$$\min_{\theta} J(\theta) = \min_{\theta} \frac{1}{N} \sum_{x,y} (\hat{y}_\theta(x) - y)^2$$

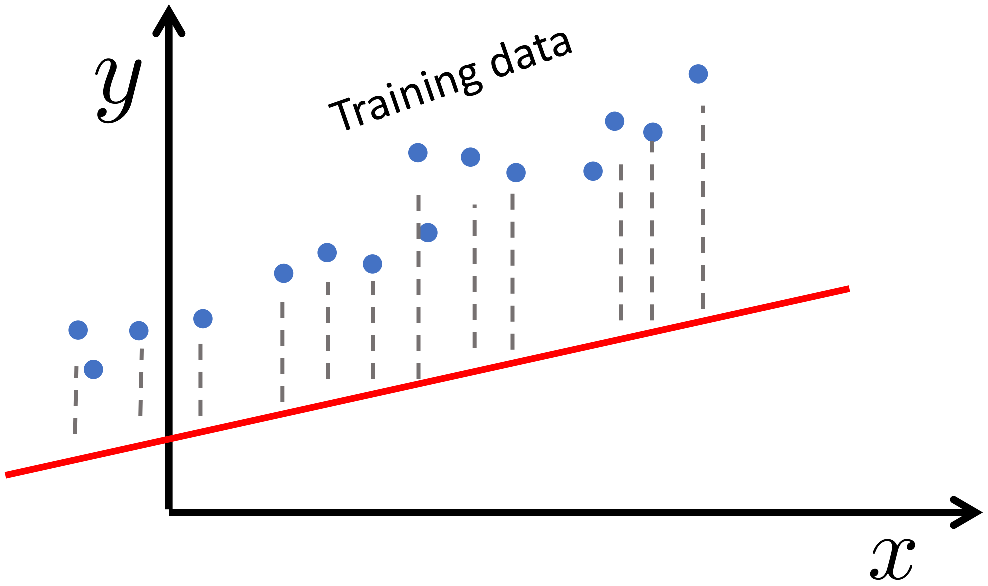
<u>Solution</u> given by:

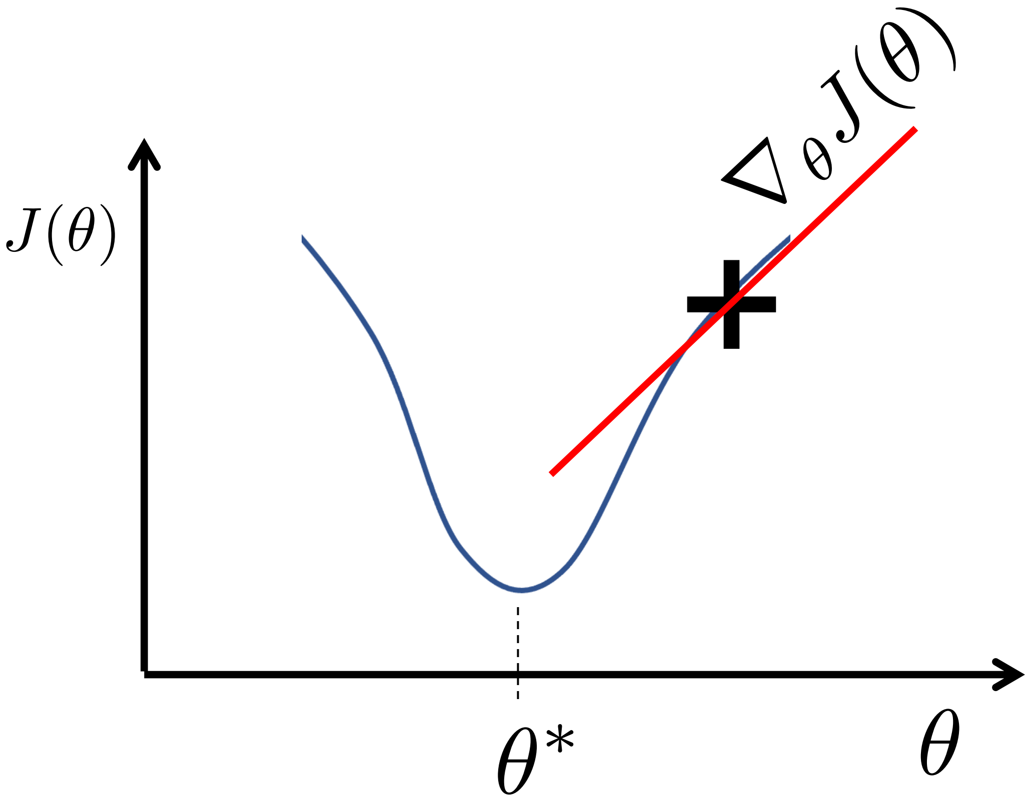
$$\theta^* = (X^T X)^{-1} X^T Y$$

Where:

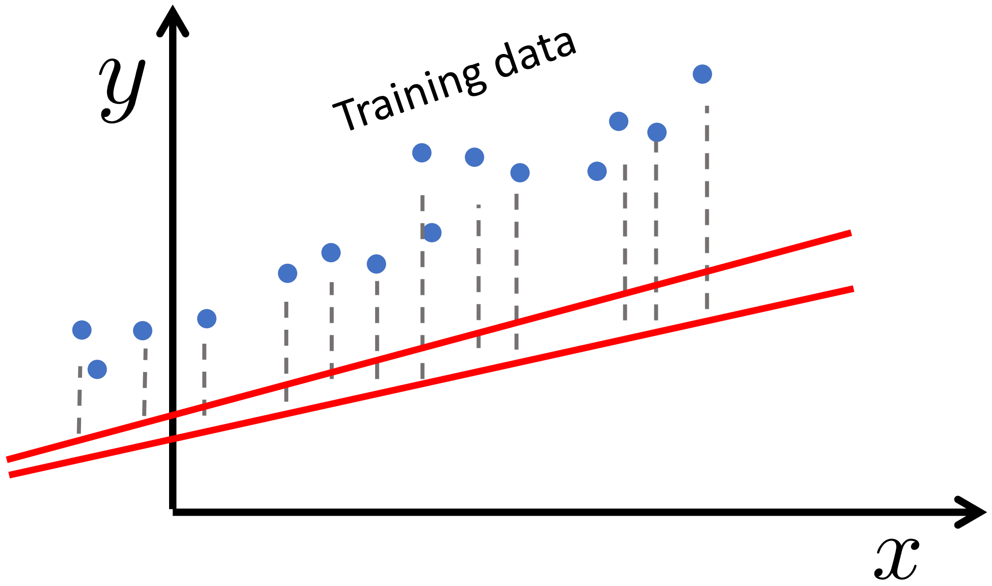
$$X = [x^{(1)}, x^{(2)}, \ldots, x^{(D)}, 1]$$
$$\theta = [w^{(1)}, w^{(2)}, \ldots, w^{(D)}, b]^T$$

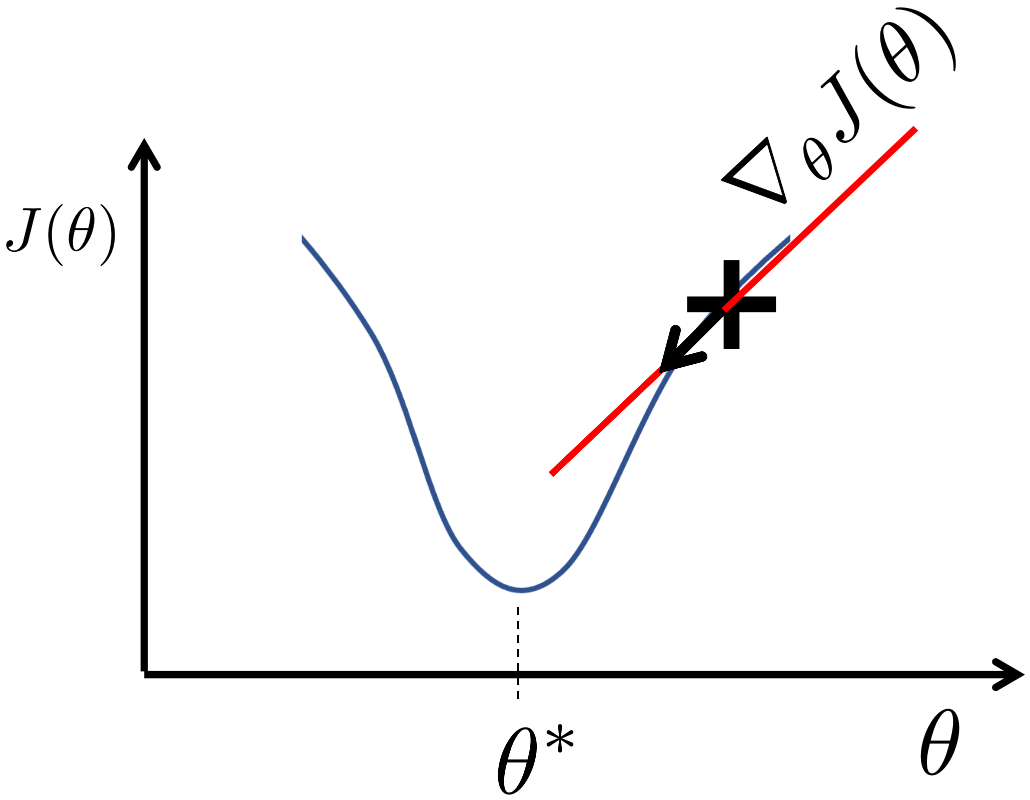
Gradient Descent for Optimization



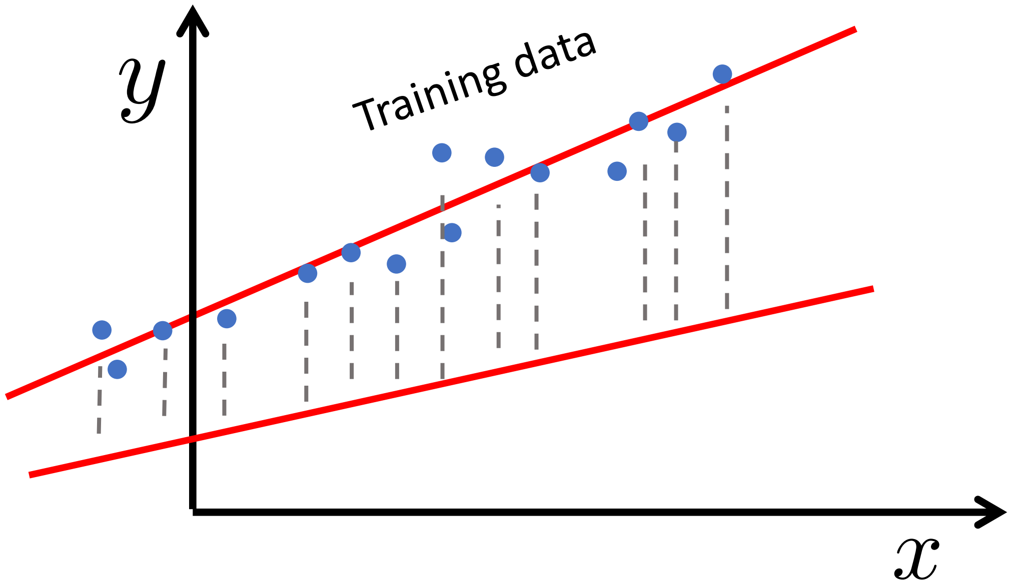


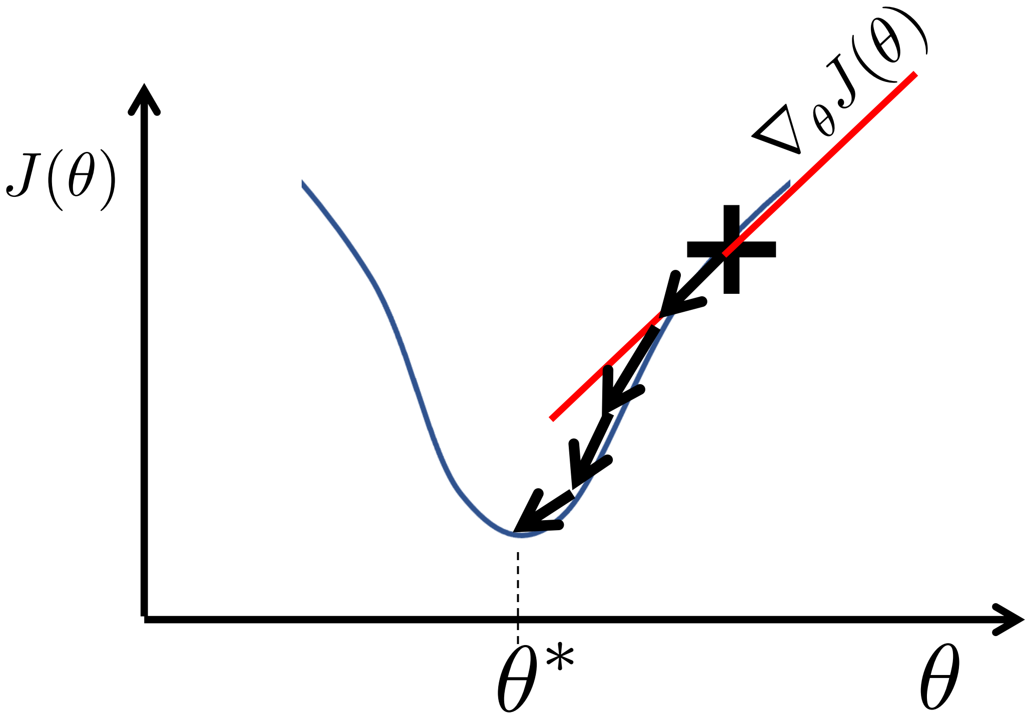
Gradient Descent for Regression



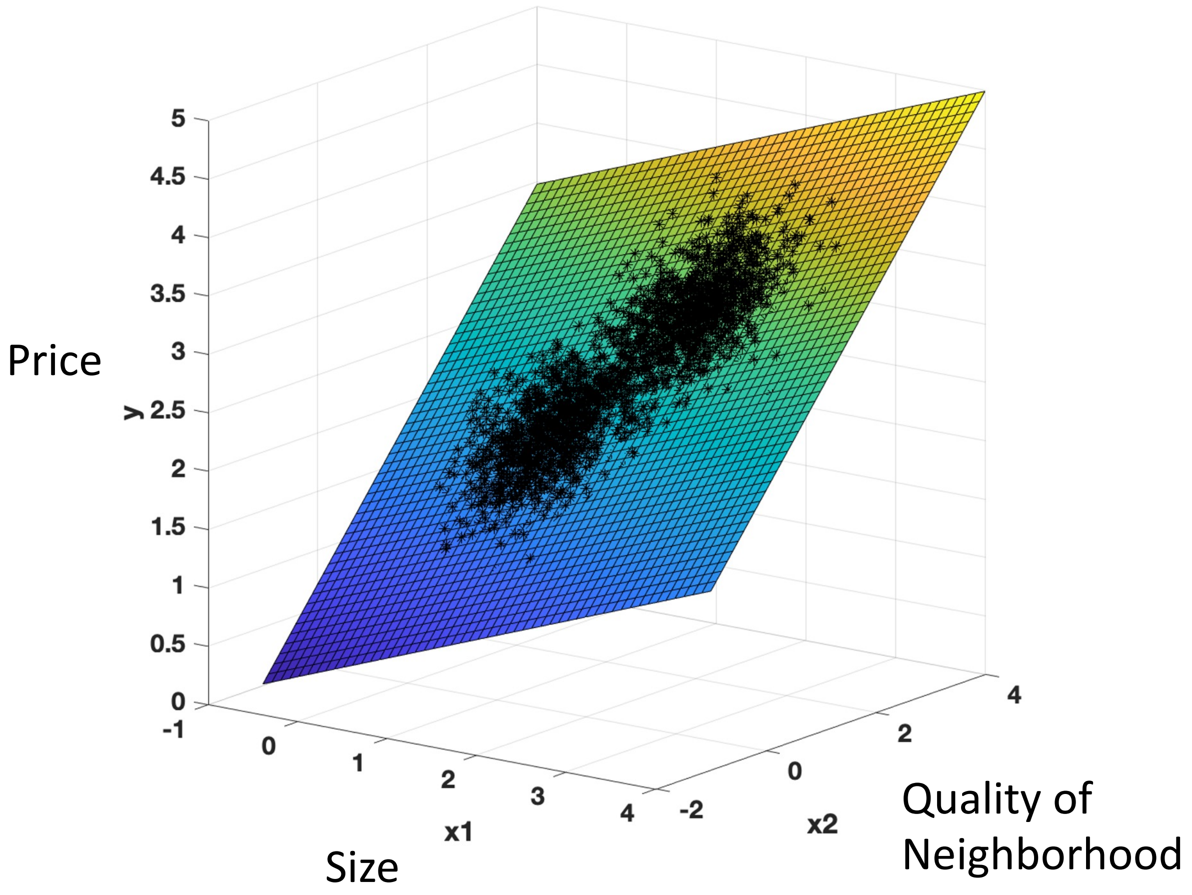


Gradient Descent for Regression

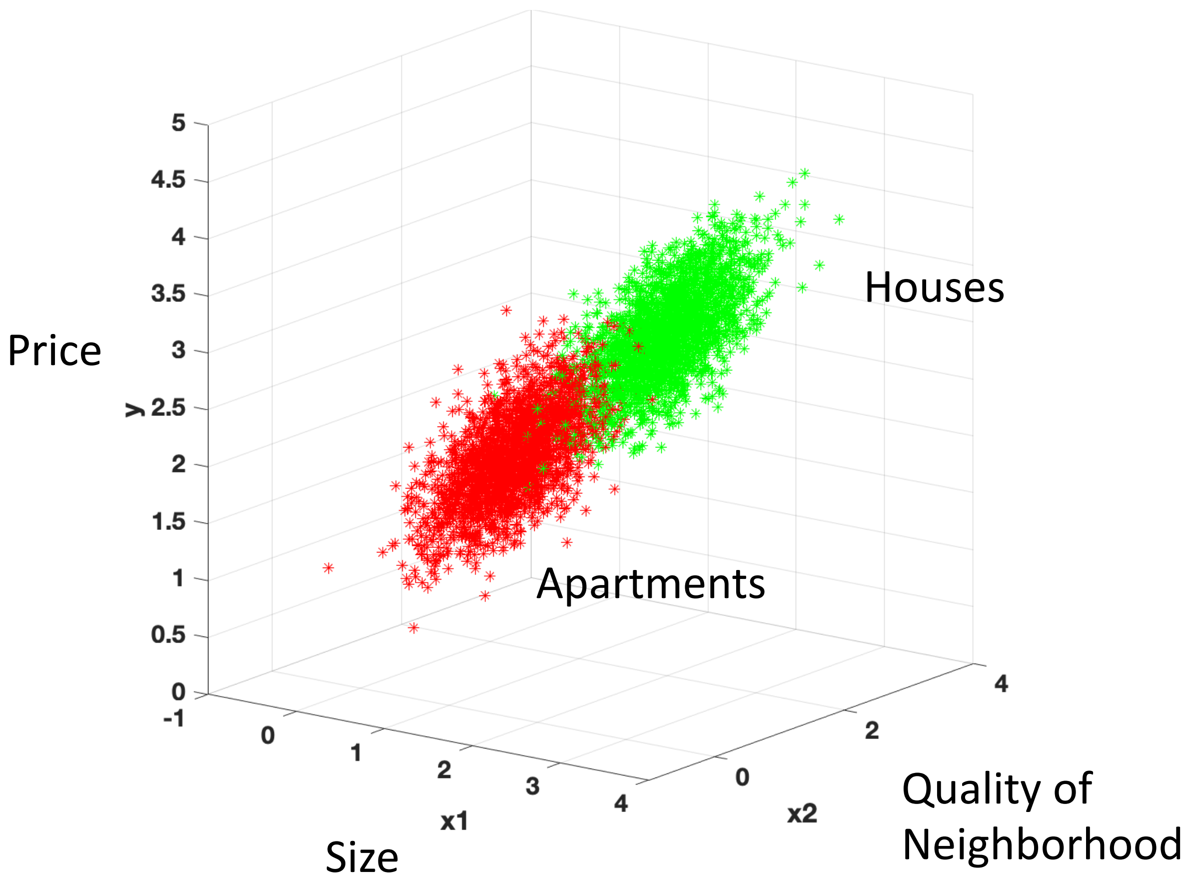




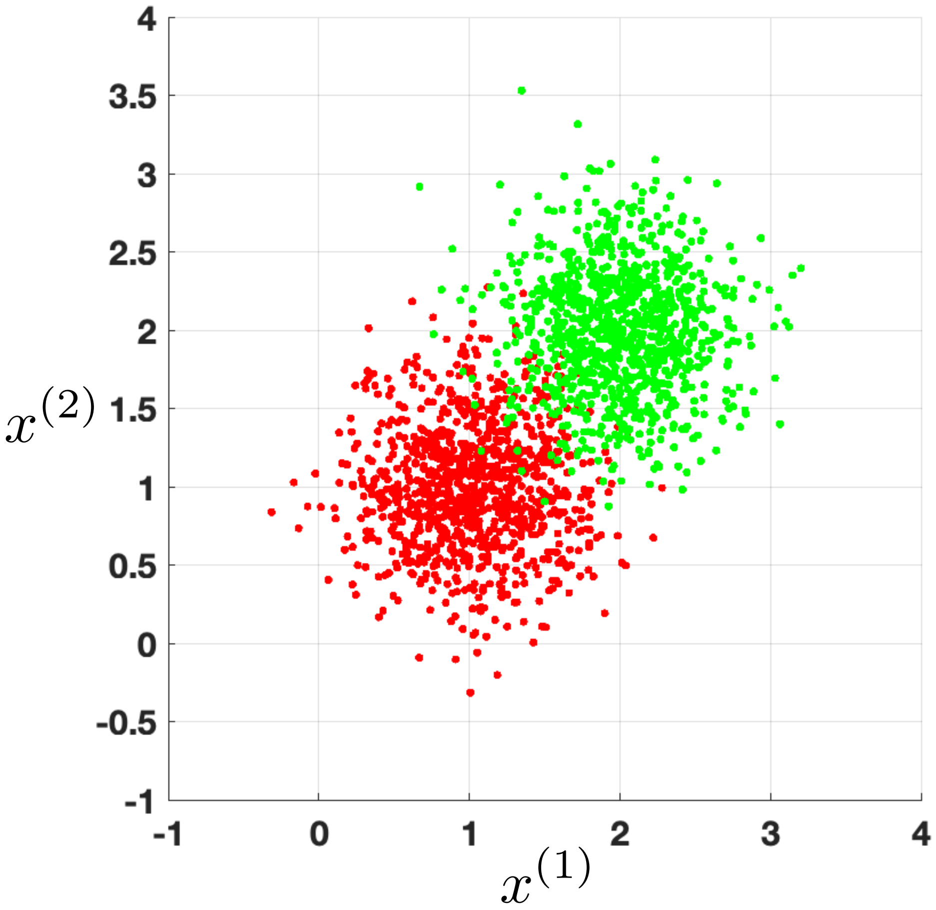
Regression


Price
y
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-1
0
1
2
3
4
x1
Size
-2
0
2
4
x2
Quality of
Neighborhood

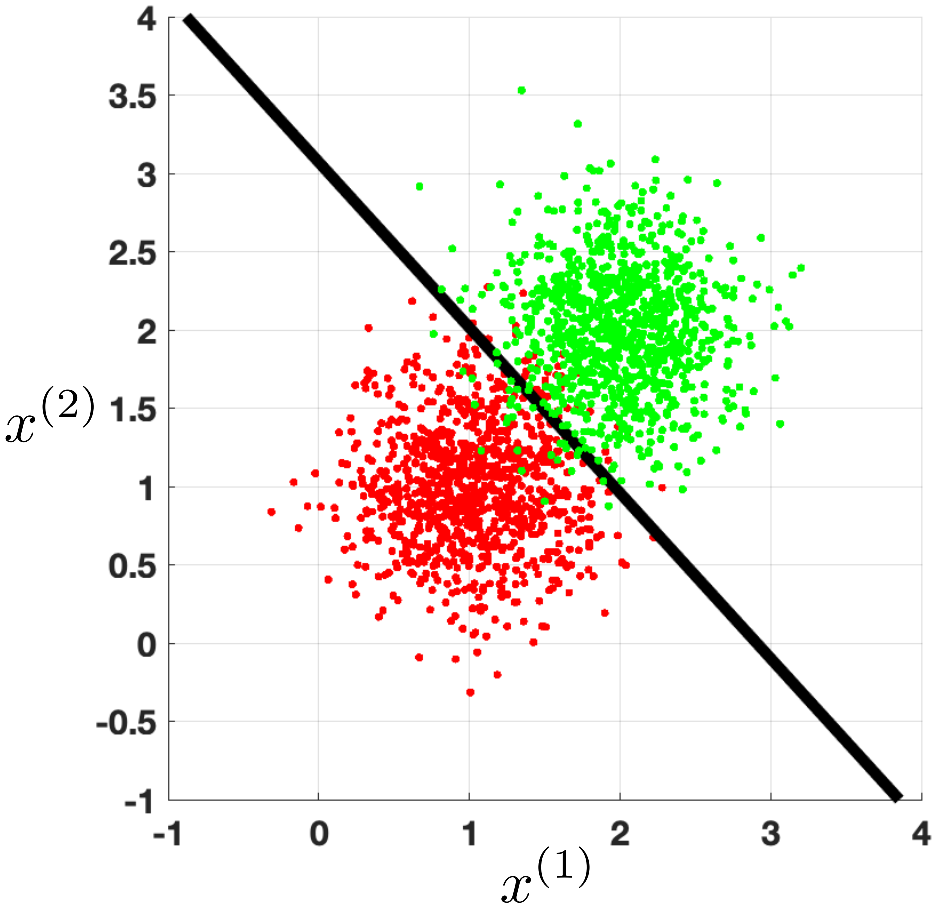
Classification



Classification



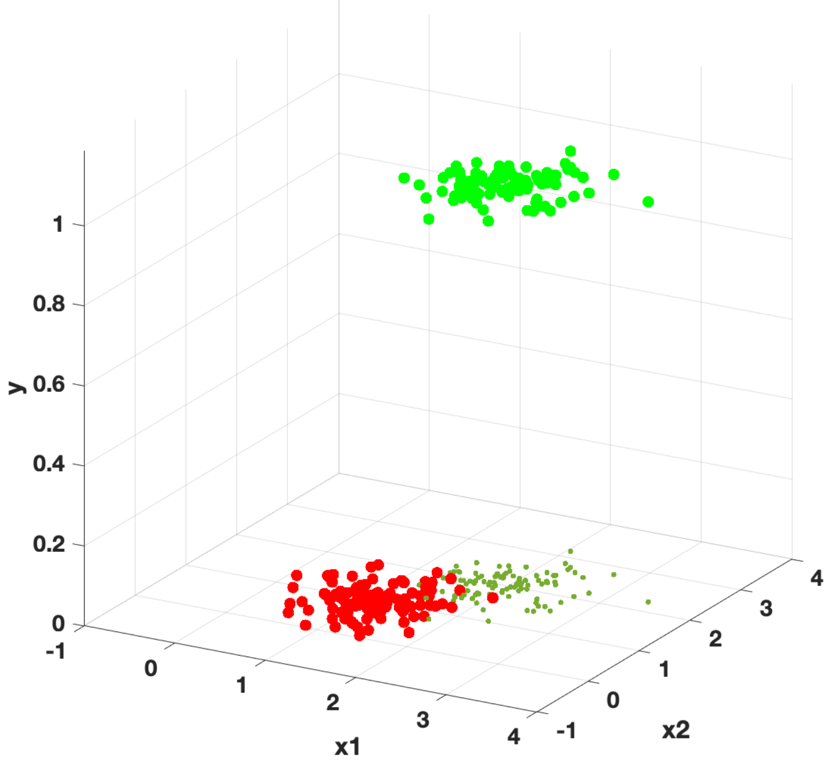
Classification

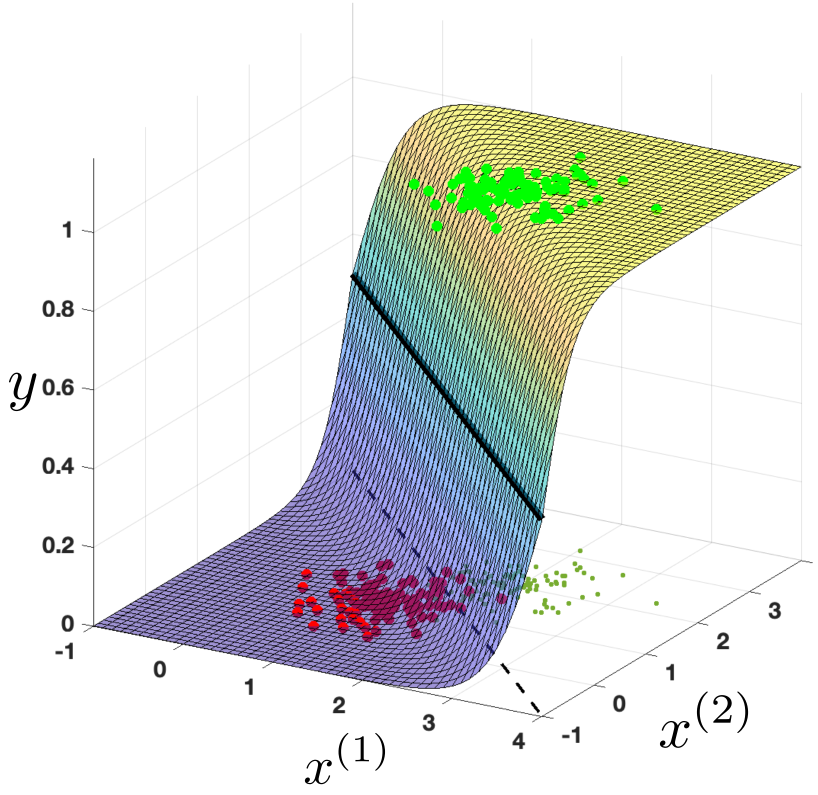


Classification with Logistic Regression

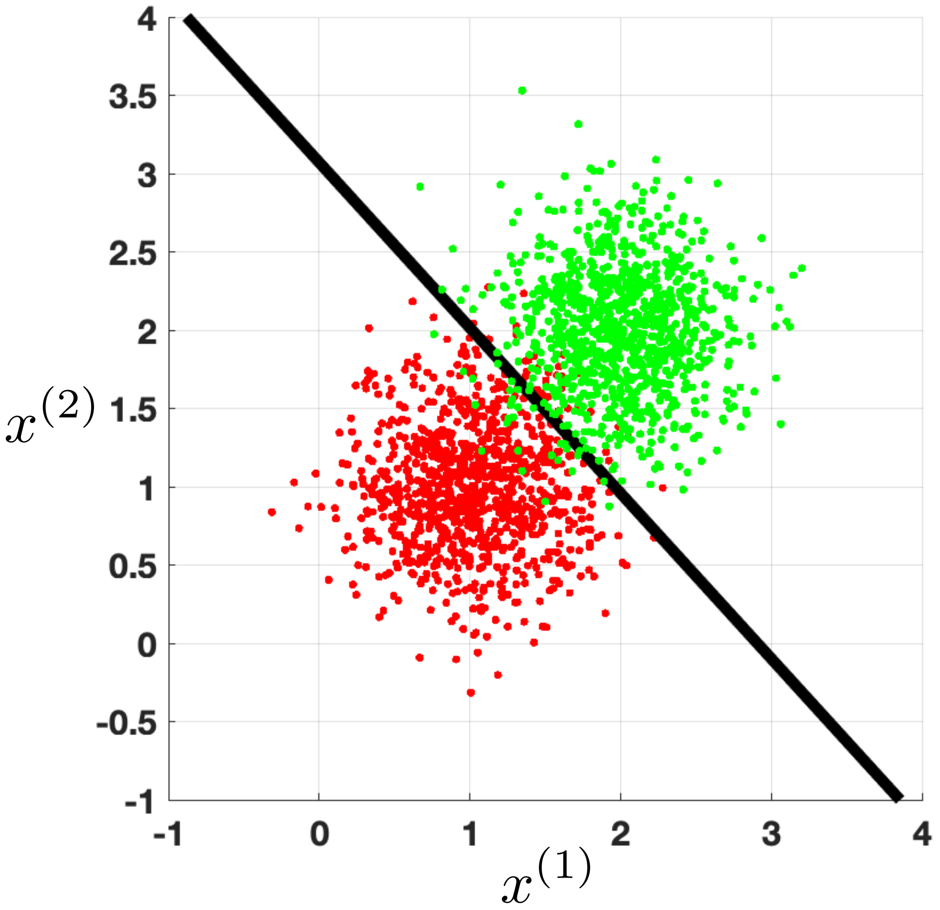
We fit a **non-linear function** to the data.
No closed form solution.
Fit with Grad Descent similar to Linear Regression.

y is Probability that a sample is of the “green” class

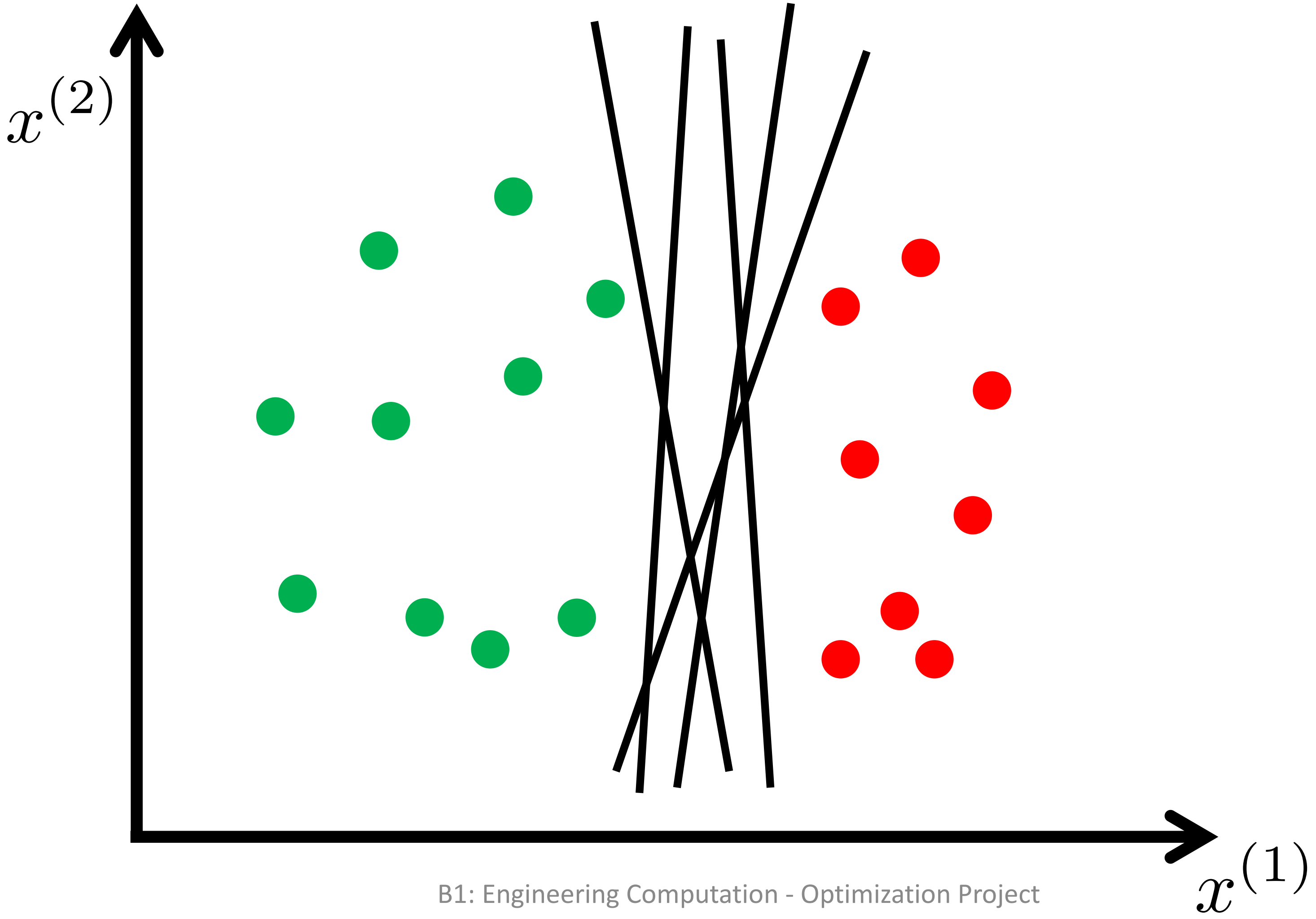




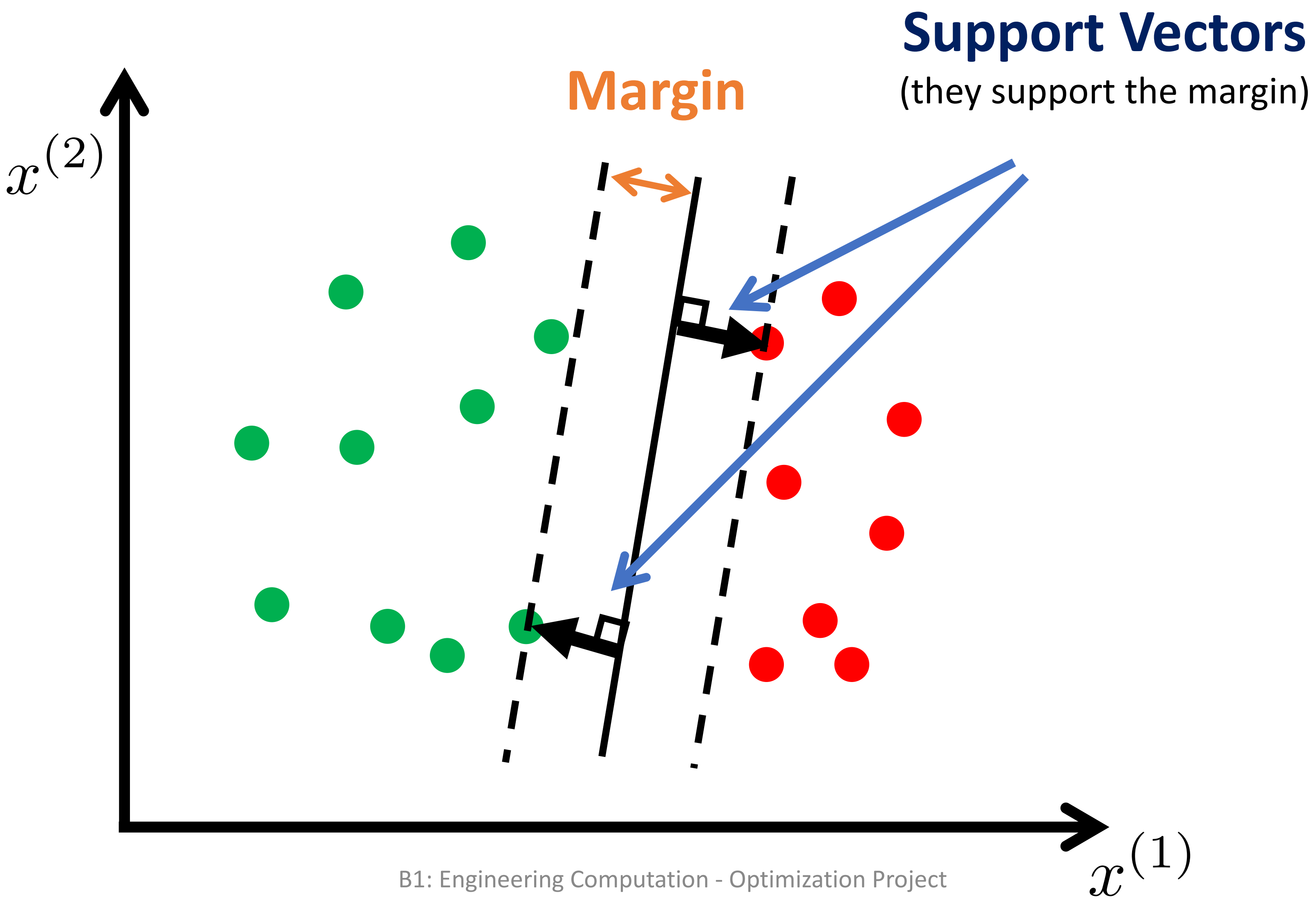
Decision boundary of LogRegr ("from up top")



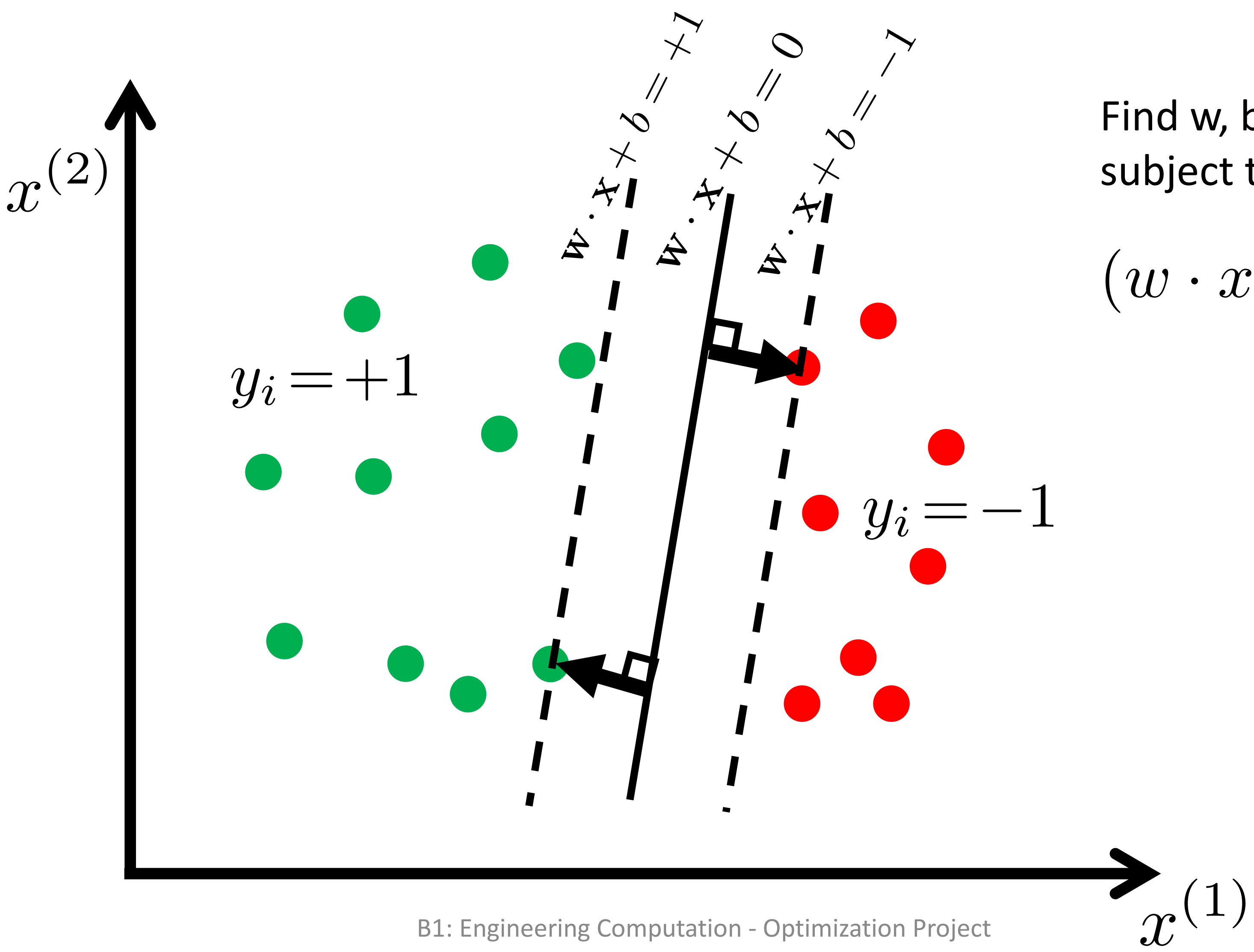
Which Decision Boundary is Best?



Support Vector Machine



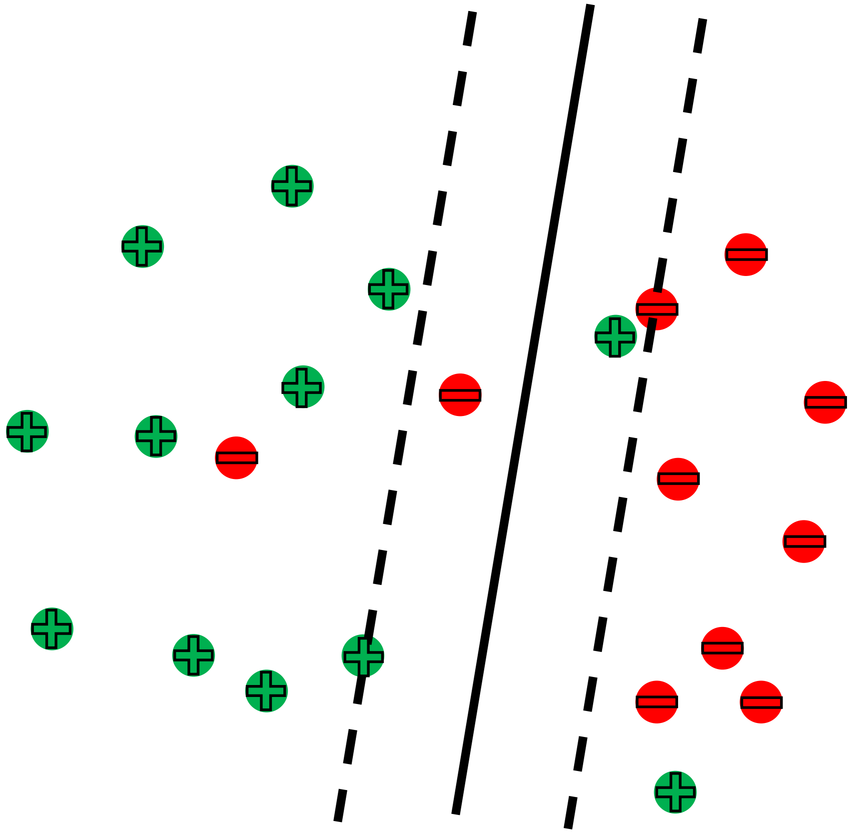
Support Vector Machine



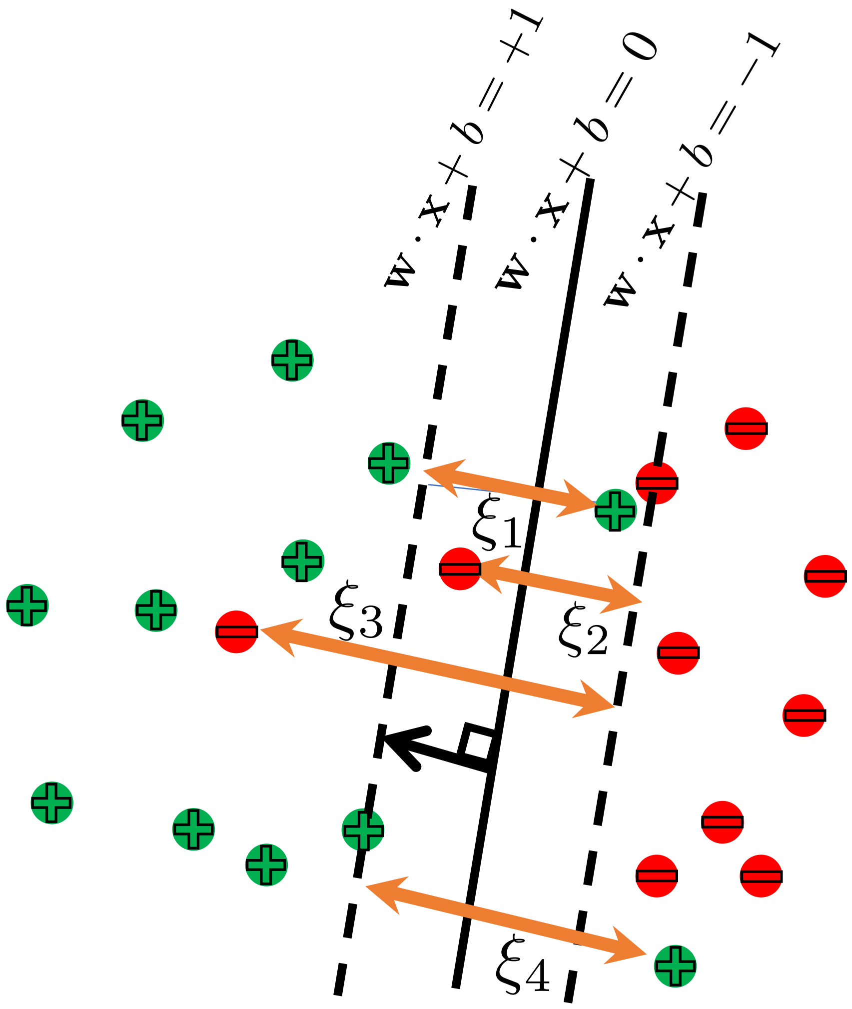
Find w, b (line)
subject to (constraints):

$$(w \cdot x_i + b)y_i \geq 1, \forall i$$

What if perfect linear boundary doesn't exist?



Support Vector Machine

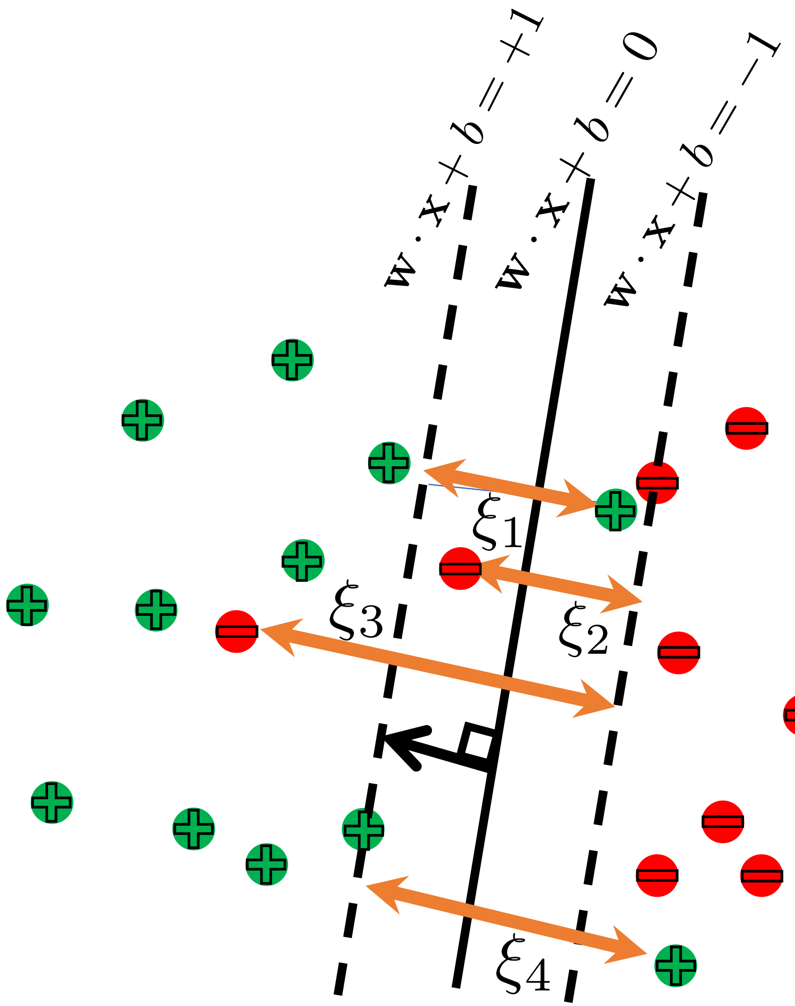


s.t. (subject to)

$$(w \cdot x_i + b)y_i \geq 1 - \xi_i, \ \forall i$$

$$\xi_i \geq 0, \ \forall i$$

Support Vector Machine - Optimization



$$\min_{w,b,\xi} \sum_i \xi_i$$

s.t. (subject to)

$$(w \cdot x_i + b) y_i \geq 1 - \xi_i, \ \forall i$$

$$\xi_i \geq 0, \ \forall i$$

SVM optimized with *Linear Programming*

Example of
Linear Programming
in 2-dimensions
(2 parameters)

Space for feasible solutions
from the constrains

$$\min_{a,b} f(a,b)$$

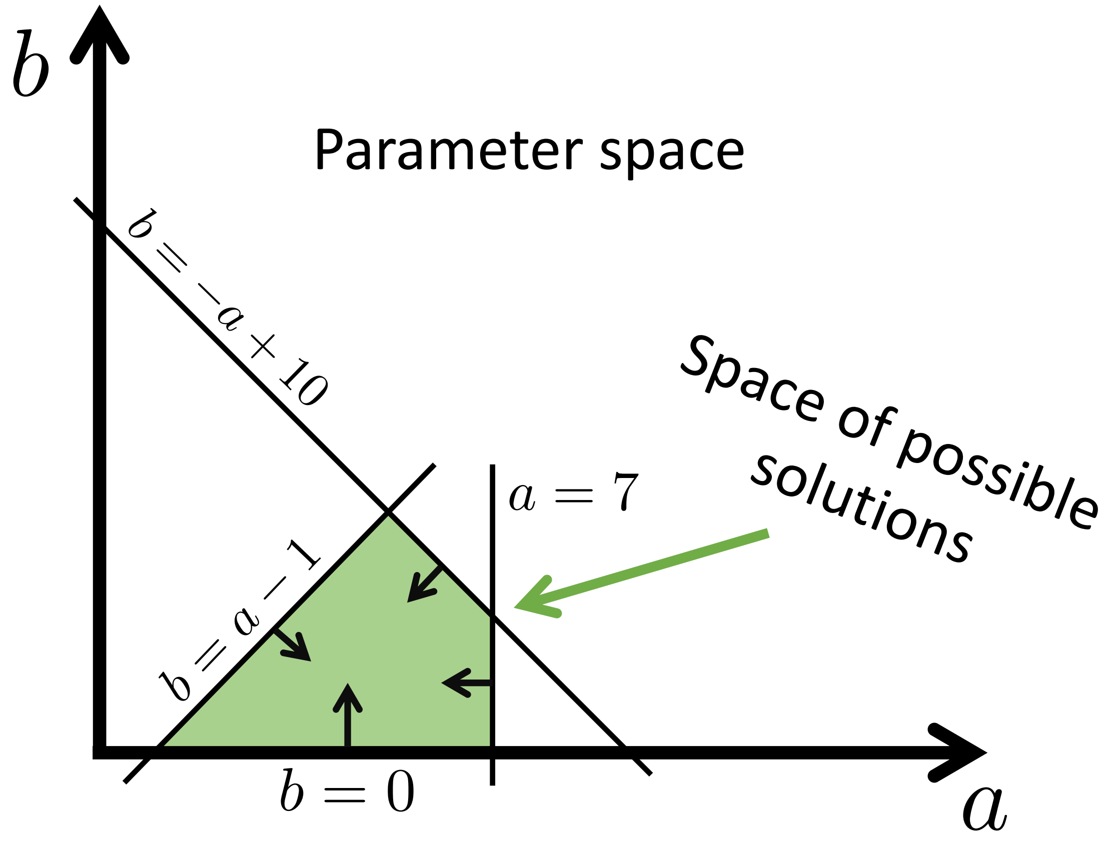
s.t. (subject to)

$$b < -x + 10$$

$$b < x - 1$$

$$b > 0$$

$$a < 7$$

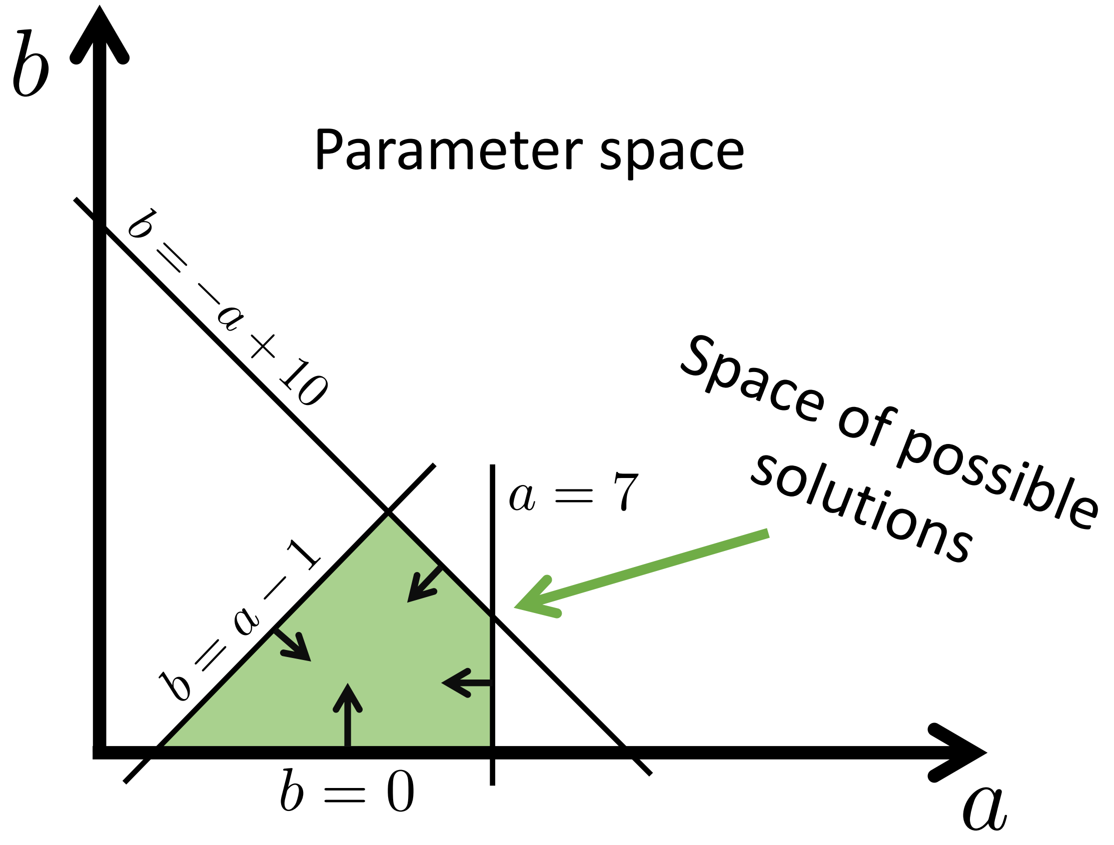


SVM optimized with *Linear Programming*

Example of
Linear Programming
in 2-dimensions
(2 parameters)

Space for feasible solutions
from the constrains

$$\min_{a,b} f(a,b)$$
$$\text{s.t. (subject to)}$$
$$b < -x + 10$$
$$b < x - 1$$
$$b > 0$$
$$a < 7$$



For SVMs, each data point
gives one inequality:

$$\min_{w,b,\xi} \sum_i \xi_i$$
$$\text{s.t. (subject to)}$$
$$(w \cdot x_i + b) y_i \geq 1 - \xi_i, \ \forall i$$
$$\xi_i \geq 0, \ \forall i$$

SVM via Gradient Descent

Equivalent formulation of SVM (will be explained):

$$\min_{w,b} \overbrace{\frac{1}{N}\sum_{i}^{N} \underbrace{\max(0, 1-(wx_i+b)y_i)}_{\textit{Hinge Loss}}}^{\text{Cost function}}$$

*Solve with **Gradient Descent**!*

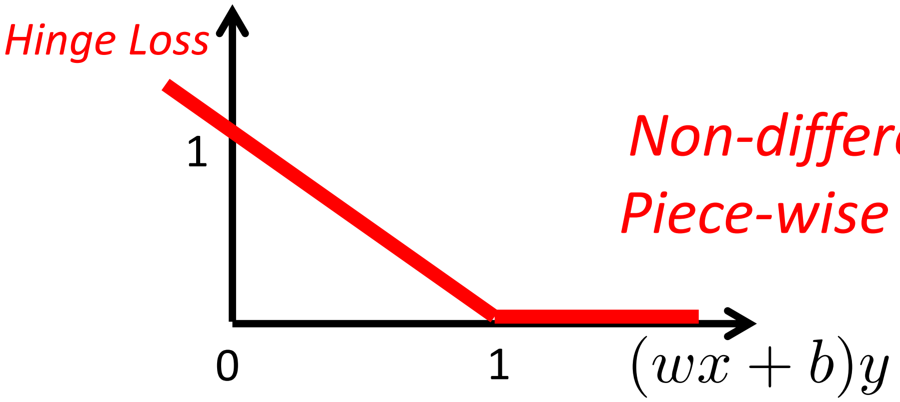
SVM via Gradient Descent

Equivalent formulation of SVM (will be explained):

$$\min_{w,b} \overbrace{\frac{1}{N}\sum_{i}^{N} \underbrace{\max(0, 1 - (wx_i + b)y_i)}_{\text{Hinge Loss}}}^{\text{Cost function}}$$

Solve with ***Gradient Descent****!*

Gradient of max(0,…) ?!?!



Non-differentiable, but…
Piece-wise differentiable!

Overview

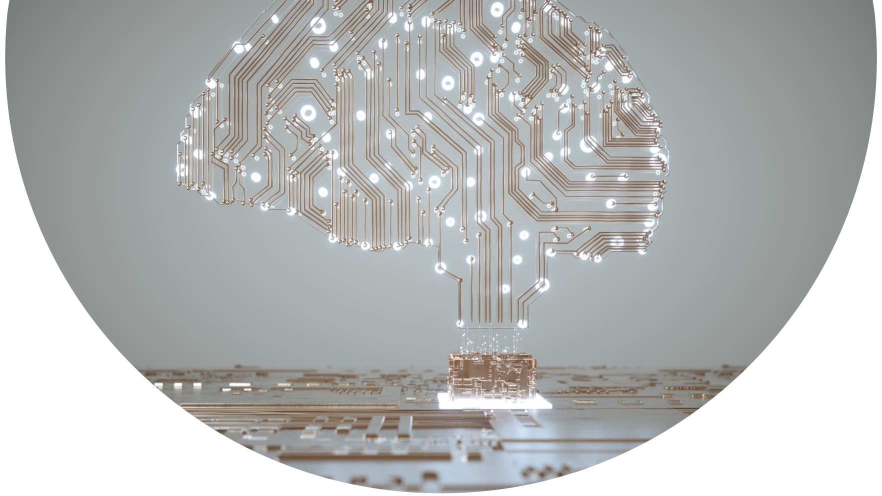
Regression

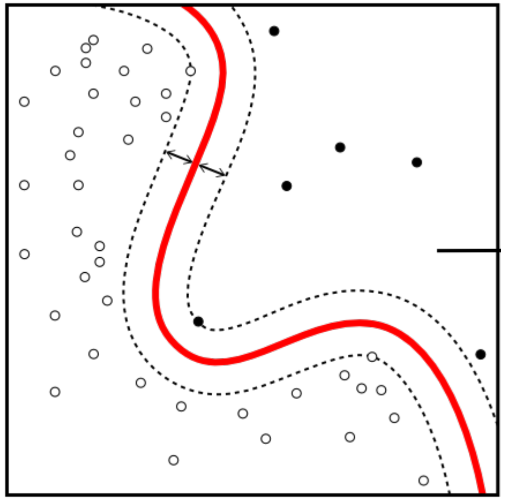
- Linear Regression with MSE (B1 Numerical Algorithms)

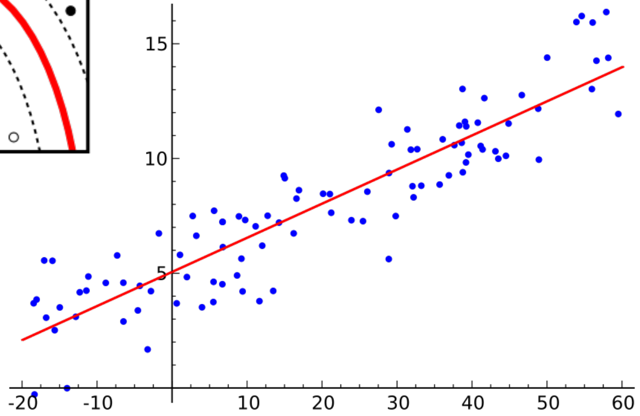
Classification

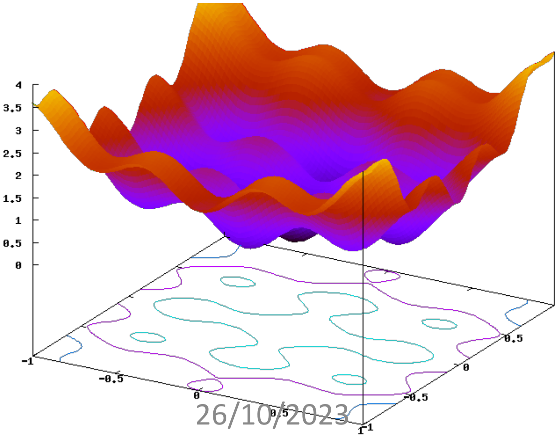
- Logistic Regression (explained in project)
- Support Vector Machines (explained in project)

- Optimizing with Closed-Form solution (B1 Numerical Algorithms)
- Gradient Descent (and stochastic variant) (B1 Numerical Algorithms)
- GD using Sub-gradients for piecewise linear cost functions (B1 Optimization)
- Linear Programming (B1 Optimization)
- Overfitting and generalization (B1 Numerical Algorithms)
- Analysis of properties of algorithms









B1:Engineering Computation – Project MT22

Optimization for Regression and Classification models