Lecture 2: Linear Methods

Machine Learning, Winter Term 2022/2023

Josif Grabocka Michael Tangermann Frank Hutter

University of Freiburg

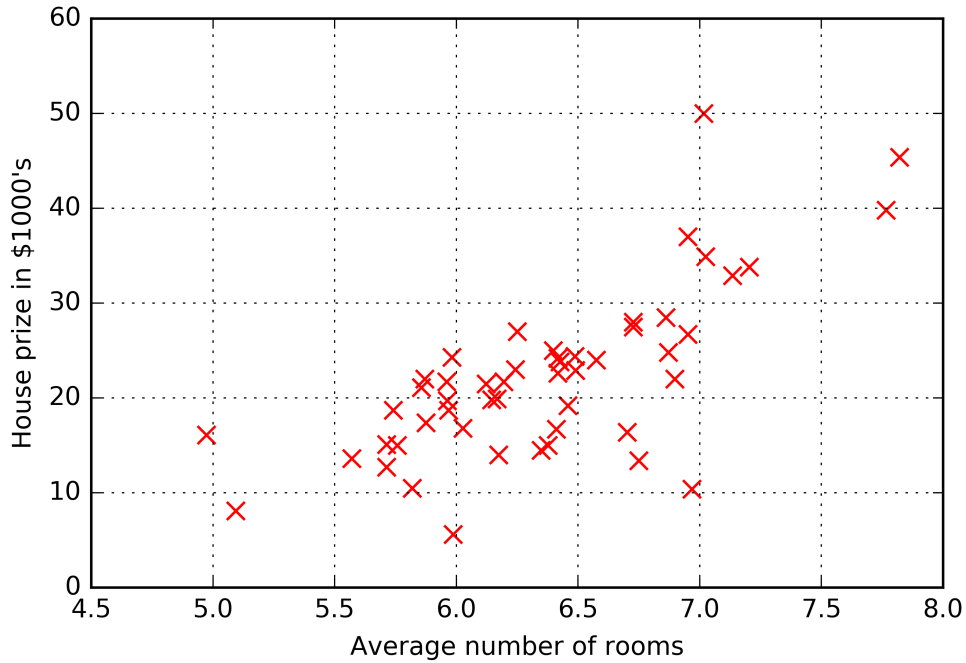
Lecture Overview

Lecture Overview

Motivation: Predicting housing prices

- Let's say we only know the average number of rooms in an area
- Aim is to predict the prize for a house in that area

avg. # rooms	price
6.575	24
6.377	21.6
5.57	34.7
5.713	33.4
7.024	36.2
5.963	28.7
5.741	22.9
6.417	27.1
...	...



The Formal Regression Problem

Given a dataset of N instances, each instance consisting of:

- A feature vector $x_i \in \mathbb{R}^D$ are D-dimensional vectors $\forall i \in \{1, \ldots, N\}$
- Target is $y_i \in \mathbb{R}$ uni-dimensional $\forall i \in \{1, \ldots, N\}$

Objective: Find a prediction model $\hat{y} := f : \mathbb{R}^D \to \mathbb{R}$ that estimates y by minimizing a loss $J : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$:

$$\arg\min_f \sum_{i=1}^{N} J\left(y_i, f(x_i)\right)$$

Refined Objective: Find a parametric model $\hat{y} := f : \mathbb{R}^D \times W \to \mathbb{R}$

$$\arg\min_{w \in W} \sum_{i=1}^{N} J\left(y_i, f(x_i; w)\right)$$

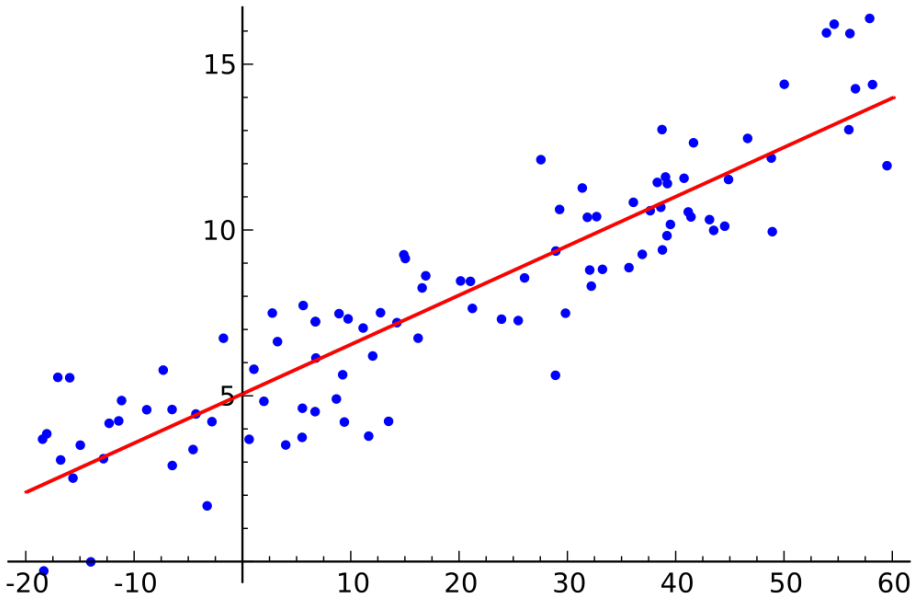
Linear Regression

The simplest regression model is the linear regression:

$$f(\mathbf{x}_i; \mathbf{w}) = w_0 + w_1 x_{i,1} + \ldots + w_D x_{i,D} = w_0 + w^T x_i$$

where

- w_0 is a fixed offset, called the *bias*
- $\mathbf{w}$ is the weight vector or parameter vector



$$f(x; w = [5, 0.125]) = 0.125\, x + 5.0$$

Linear Regression: A limited model?

Characteristics:

- Linear function of the weights / parameters $w \in \mathbb{R}^{D+1}$
- Linear function of the input dimensions / variables $x \in \mathbb{R}^{D+1}$

What **significant limitation** does this model have?

Nonlinear Mapping

- Can the function $f(x) = (x + 1)^2$ be approximated through a linear function?

Nonlinear Mapping

- Can the function $f(x) = (x + 1)^2$ be approximated through a linear function?
- Yes, but only if we **map** the feature x into a new space:

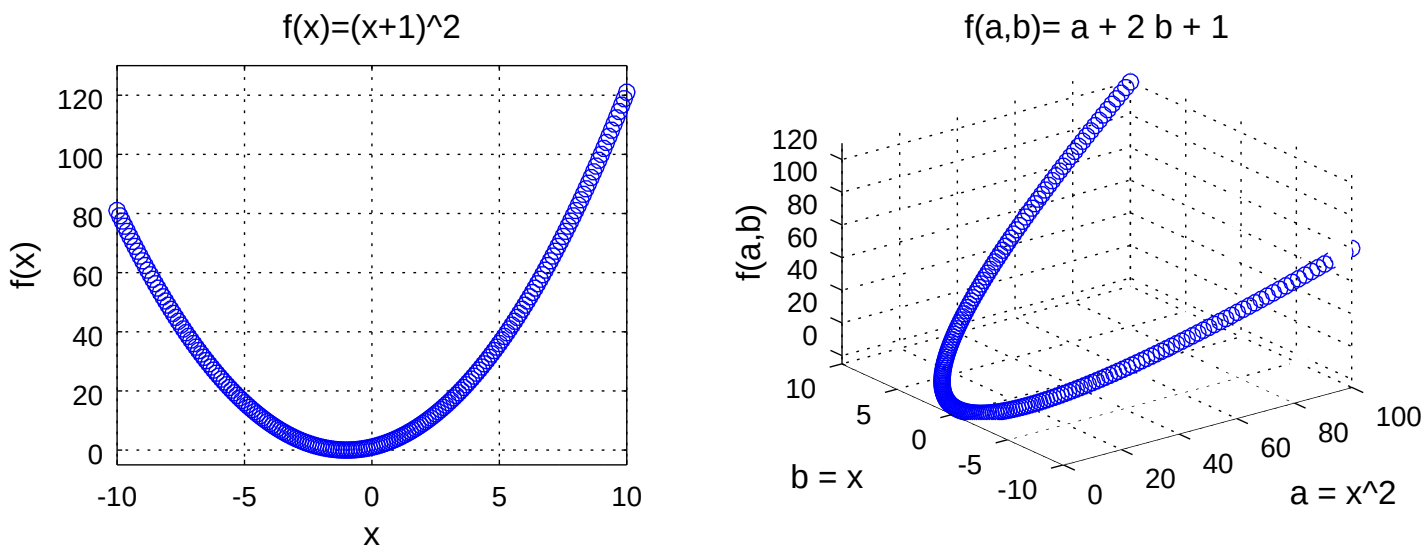


Figure: Mapping feature x into a new dimensionality $x \rightarrow \phi(x) = (a, b)$

Linear Regression with Basis Functions

Nonlinear projection of features $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$

Linear combinations of fixed nonlinear functions of the input variables:

$$f(\mathbf{x_i}, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j \phi(\mathbf{x_i})_j$$

where $\phi(\mathbf{x_i})_j$ are known as *basis functions*.

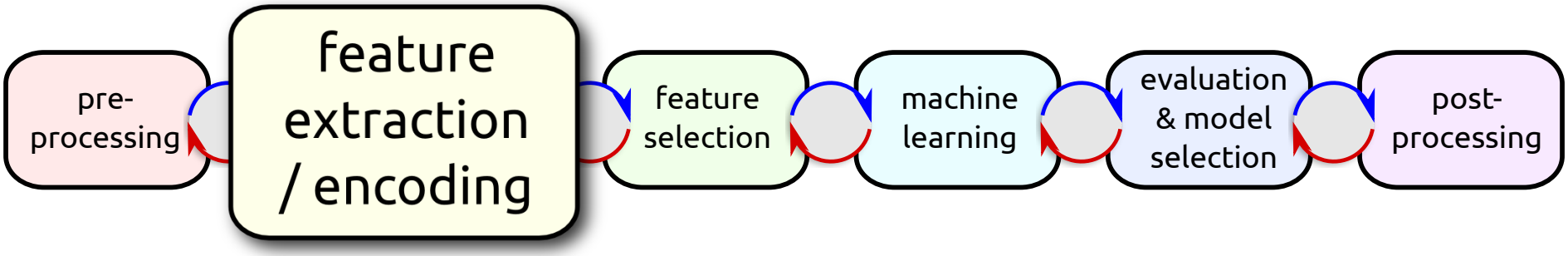
Characteristics of this *enhanced* linear regression model:

- $f(\mathbf{x}, \mathbf{w})$ is still a linear function of the weights / parameters w_i
- $f(\mathbf{x}, \mathbf{w})$ is a **nonlinear** function of the input dimensions / variables x_i

Linear Regression with Basis Functions (II)

Characteristics of this *enhanced* linear regression model:

- Adding basis functions may enlarge the dimensionality
- The use of fixed basis functions corresponds to an earlier step in the ML pipeline

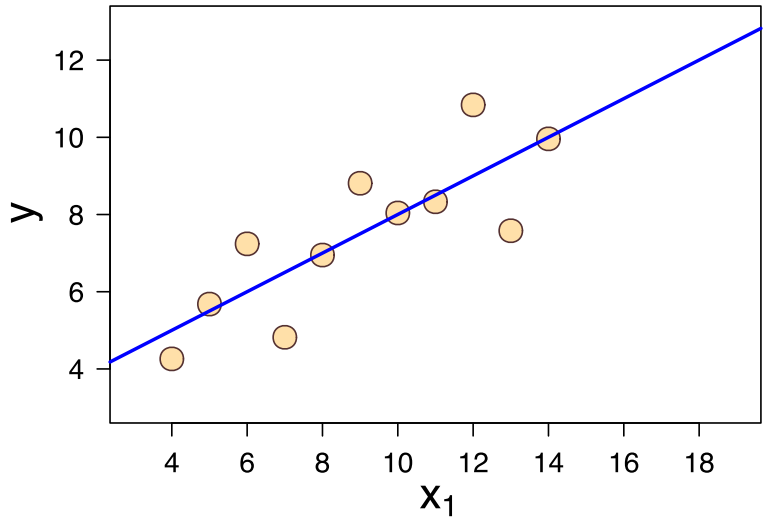


Combine features:

- We can use complex operations to combine features
- Combined features can be more expressive than their components (use expert knowledge!)

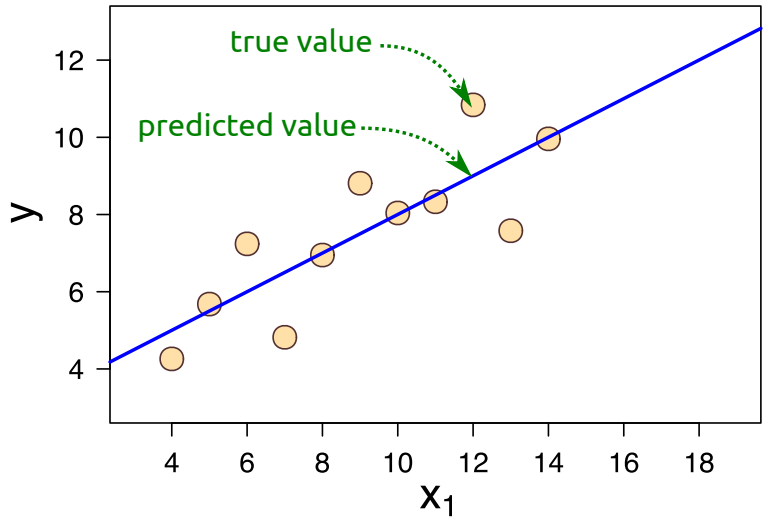
Interpretation of the Quality of the Model

How can we guess the quality of the regression model?



Interpretation of the Quality of the Model

How can we guess the quality of the regression model?



- Residual Error:

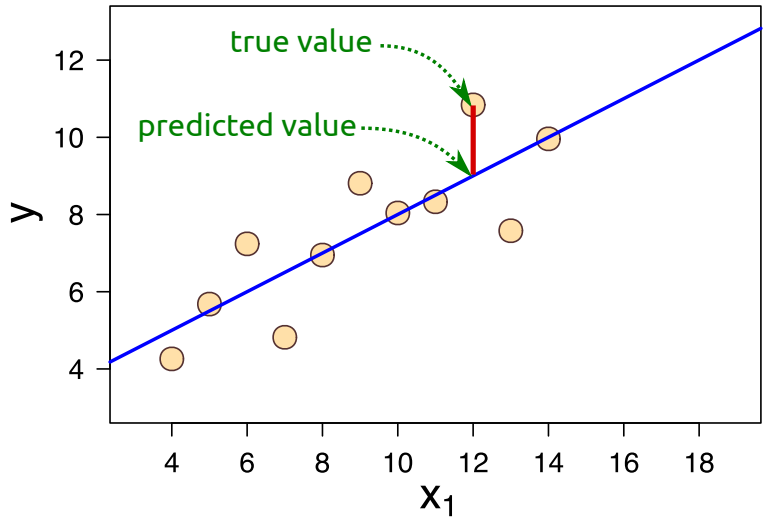
$$\epsilon_i = y_i - f(x_i; w)$$

- Loss/Cost:

$$J(w) = \sum_{i=1}^{N} J(y_i, f(x_i; w)), \text{e.g. } J(y_i, f(x_i; w)) = \epsilon_i^2$$

Interpretation of the Quality of the Model

How can we guess the quality of the regression model?



- Residual Error:

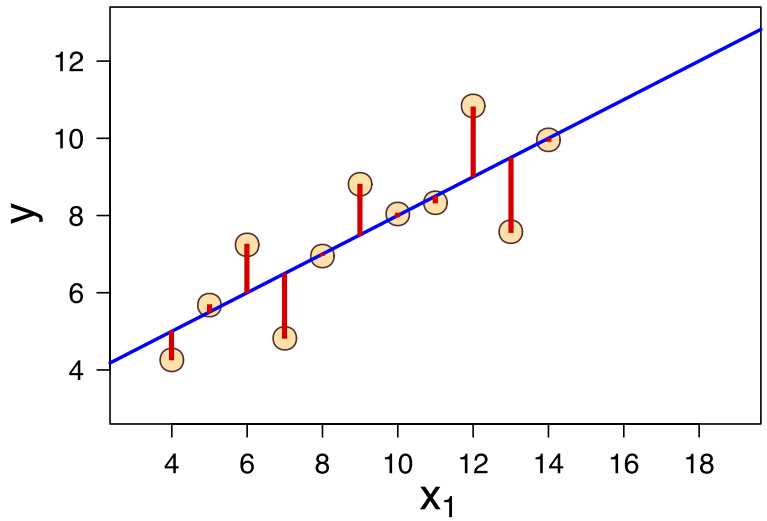
$$\epsilon_i = y_i - f(x_i; w)$$

- Loss/Cost:

$$J(w) = \sum_{i=1}^{N} J(y_i, f(x_i; w)), \text{e.g. } J(y_i, f(x_i; w)) = \epsilon_i^2$$

Interpretation of the Quality of the Model

How can we guess the quality of the regression model?



- Residual Error:

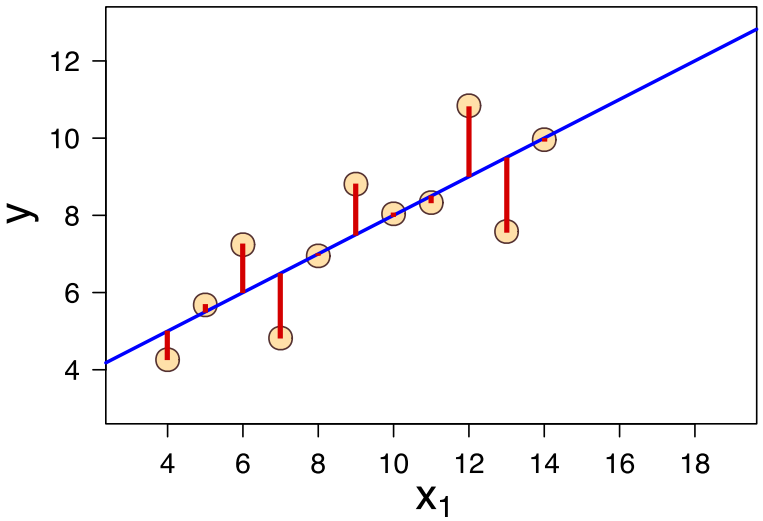
$$\epsilon_i = y_i - f(x_i; w)$$

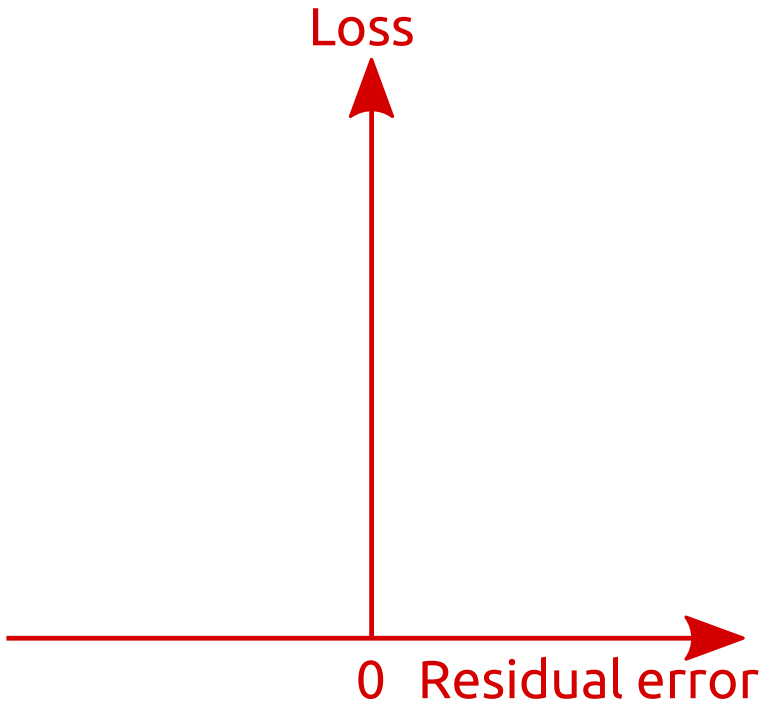
- Loss/Cost:

$$J(w) = \sum_{i=1}^{N} J(y_i, f(x_i; w)), \text{e.g. } J(y_i, f(x_i; w)) = \epsilon_i^2$$

Alternative Loss Functions

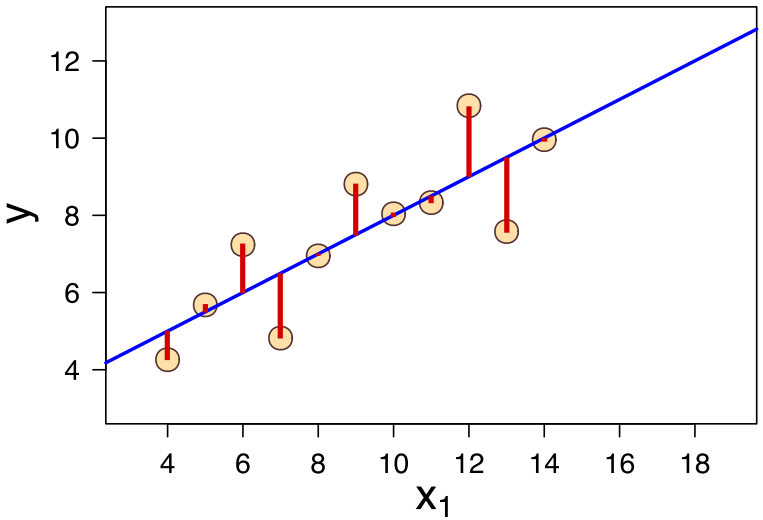
Typical loss choices, either L2: $J(y, \hat{y}) = (y - \hat{y})^2$, or L1: $J(y, \hat{y}) = |y - \hat{y}|$

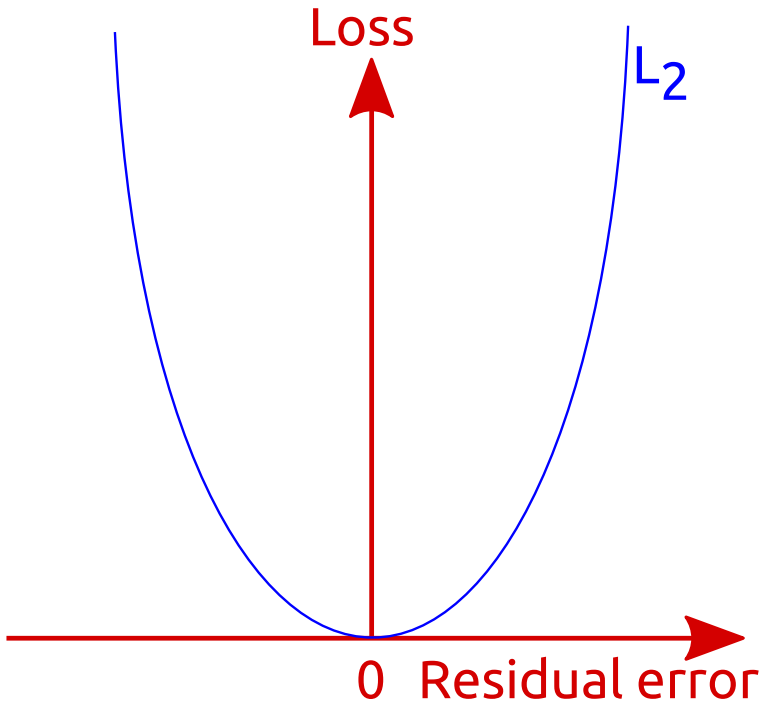




Alternative Loss Functions

Typical loss choices, either L2: $J(y, \hat{y}) = (y - \hat{y})^2$, or L1: $J(y, \hat{y}) = |y - \hat{y}|$

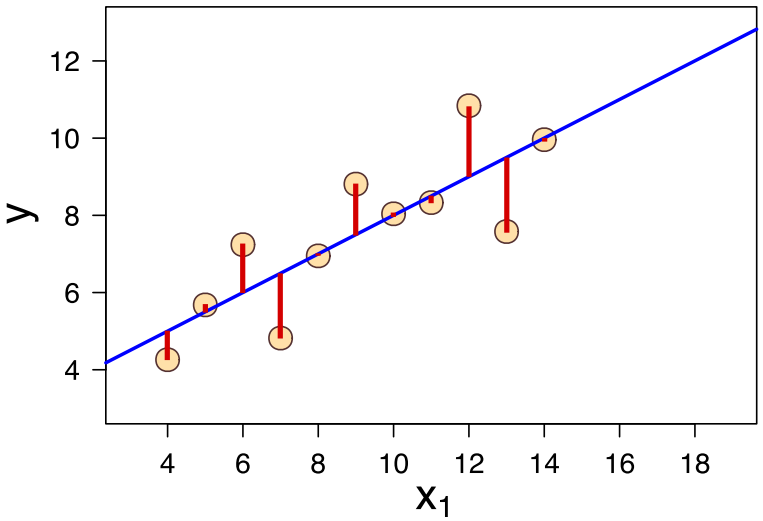


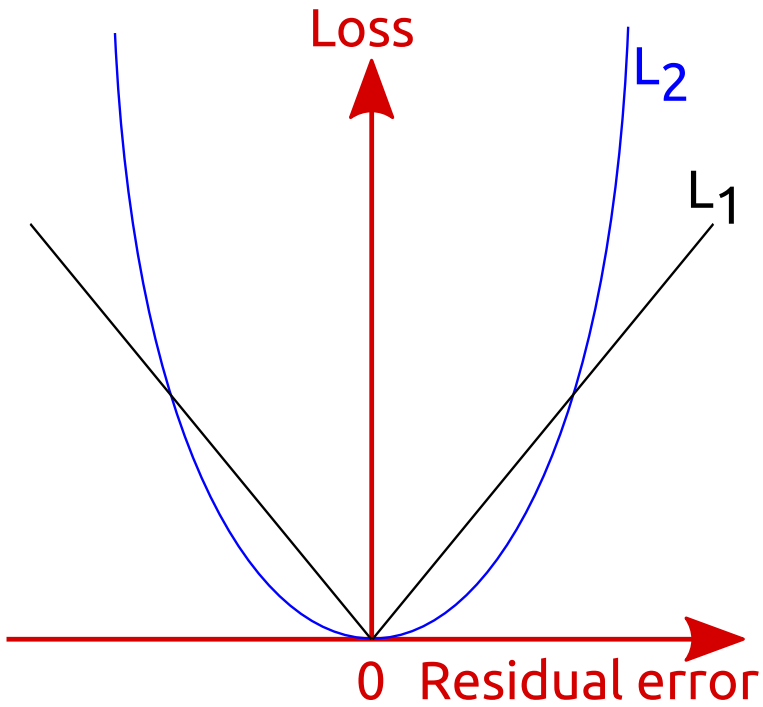


-  In which situations would you expect absolute loss (L_1) to be preferable compared to squared loss (L_2)?

Alternative Loss Functions

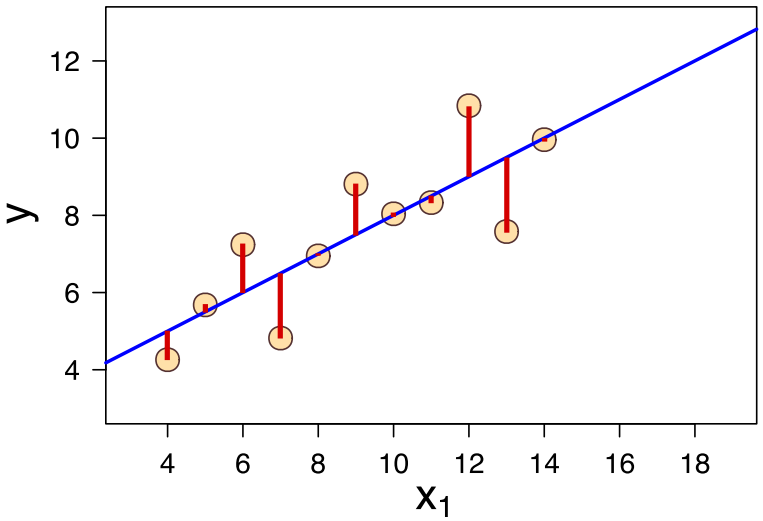
Typical loss choices, either L2: $J(y, \hat{y}) = (y - \hat{y})^2$, or L1: $J(y, \hat{y}) = |y - \hat{y}|$

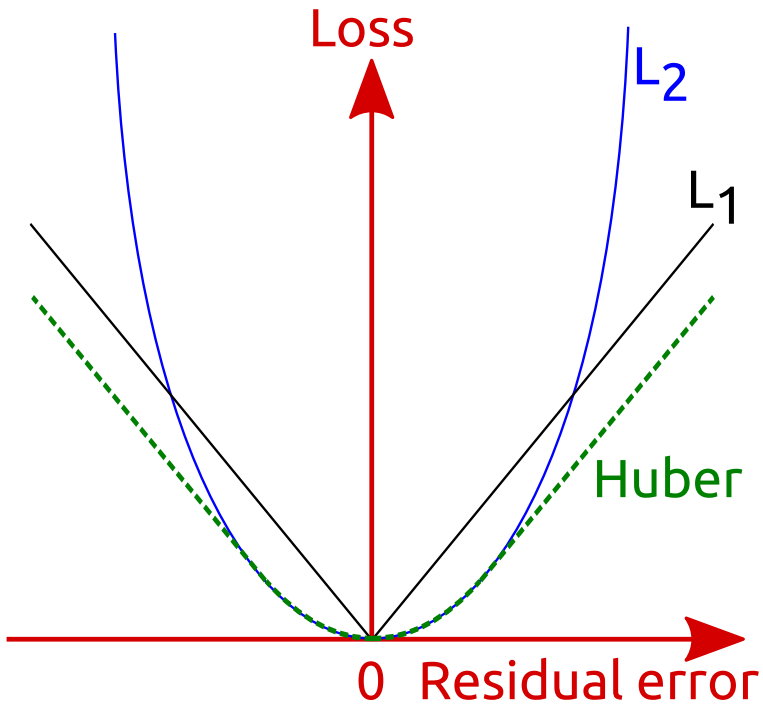




Alternative Loss Functions

Typical loss choices, either L2: $J(y, \hat{y}) = (y - \hat{y})^2$, or L1: $J(y, \hat{y}) = |y - \hat{y}|$





How Can We Derive the Weight Parameters $\mathbf{w}$?

$$\arg\min_{w} \sum_{i=1}^{N} J\left(y_i, f(x_i; w)\right)^2$$

Weight vector $\mathbf{w}$ can be derived in (at least) two manners:

Option 1: Determine the weights analytically.

Option 2: Guess the entries of $\mathbf{w}$ and try to improve them iteratively

Assumptions Required for Analytical Solution

$$y_i = w_0 + w_1 x_1 + \ldots + w_D x_D + \varepsilon_i$$

The analytical solution is based on **three assumptions**. If violated, the model may fail or underperform.

- **A1:** The expected value of the residual errors is zero:
  $\forall i\colon \mathrm{E}(\varepsilon_i) = 0$
- **A2:** The residual errors are uncorrelated and share the same variance:
  $\forall i\colon \mathrm{Var}(\varepsilon_i) = \sigma^2$
- **A3:** The residual errors follow a normal distribution:
  $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$

Optimization of $\mathbf{w}$ via Least Squares Loss Function

Assuming the slightly more elegant formulation for linear regression (in augmented vector notation):

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}$$

we would like to minimize the squared error of the model:

$$\underset{\mathbf{w}}{\operatorname{argmin}} ||\boldsymbol{\varepsilon}||^2 = \underset{\mathbf{w}}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2$$

Optimization of $\mathbf{w}$ via Least Squares Loss Function

$$\underset{\mathbf{w}}{\operatorname{argmin}} \; ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 \quad = \quad \underset{\mathbf{w}}{\operatorname{argmin}} \; (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\begin{aligned}
(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) &= \mathbf{y}^T\mathbf{y} - (\mathbf{X}\mathbf{w})^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + (\mathbf{X}\mathbf{w})^T\mathbf{X}\mathbf{w} \\
&\rightarrow \; -(\mathbf{X}\mathbf{w})^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + (\mathbf{X}\mathbf{w})^T\mathbf{X}\mathbf{w} \\
&\rightarrow \; -\mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \\
&\rightarrow \; -\mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \\
&\rightarrow \; -2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}
\end{aligned}$$

Optimization of $\mathbf{w}$ via Least Squares Loss Function (II)

$$\underset{\mathbf{w}}{\operatorname{argmin}} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

Find optimum by setting the partial derivatives w.r.t. w to zero:

$$0 \quad = \quad \frac{\partial}{\partial w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

Use $\frac{\partial}{\partial \mathbf{w}} \ \mathbf{w}^T A\mathbf{w} = \mathbf{w}^T(A + A^T)$ to derive:

$$0 = -2\mathbf{y}^T\mathbf{X} + \mathbf{w}^T(\mathbf{X}^T\mathbf{X} + (\mathbf{X}^T\mathbf{X})^T)$$

Optimization of $\mathbf{w}$ via Least Squares Loss Function

$$
\begin{aligned}
0 &= -2\mathbf{y}^T\mathbf{X} + \mathbf{w}^T(\mathbf{X}^T\mathbf{X} + (\mathbf{X}^T\mathbf{X})^T) \\
&= -2\mathbf{y}^T\mathbf{X} + 2\mathbf{w}^T\mathbf{X}^T\mathbf{X}
\end{aligned}
$$

$$
\begin{aligned}
\Leftrightarrow \quad \mathbf{y}^T\mathbf{X} &= (\mathbf{w}^T\mathbf{X}^T)\mathbf{X} \\
\Leftrightarrow \quad \mathbf{X}^T\mathbf{y} &= \mathbf{X}^T\mathbf{X}\mathbf{w} \\
\Leftrightarrow \quad (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} &= \mathbf{w}
\end{aligned}
$$

Optimization of $\mathbf{w}$

This delivers the analytical solution:

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{w}$$

- What is the costly part of the model training? 

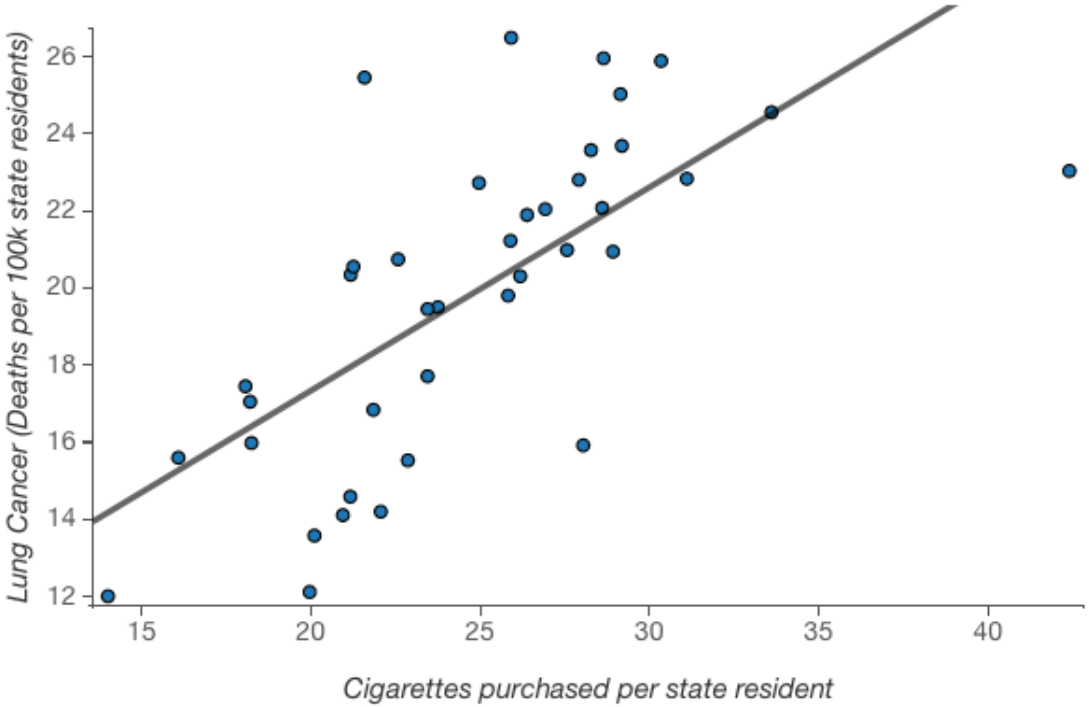
Optimization of $\mathbf{w}$

This delivers the analytical solution:

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{w}$$

- What is the costly part of the model training? 
  $\rightarrow$ Calculation of inverse $\in \mathbb{R}^{D\times D}$
  - Gauss-Jordan elimination procedure: O(D^3)
  - Coppersmith-Winograd: O($D^{2.37\ldots}$)
  - For large dimensions: stochastic gradient descend (nice: the error function is convex!)

Typical Use of Linear Regression



- To predict unknown values y_i for novel input variables $\mathbf{x}_i$
- Estimate the influence of a single input variable or several variables (e.g. in medicine), i.e. estimating the strength of the **correlation** between x_i and y. **BUT: does not allow for causal interpretation!**
- Visualization to understand a novel dataset: linear or non-linear relationships? Outliers? Distribution of residual errors?

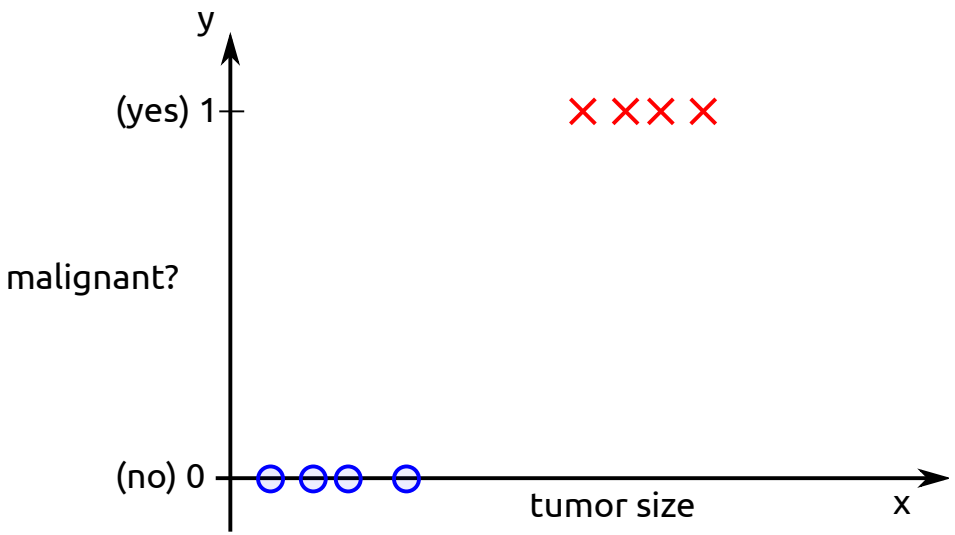
Lecture Overview

Simple Example: Tumor Classification

Is the tumor benign or malignant?

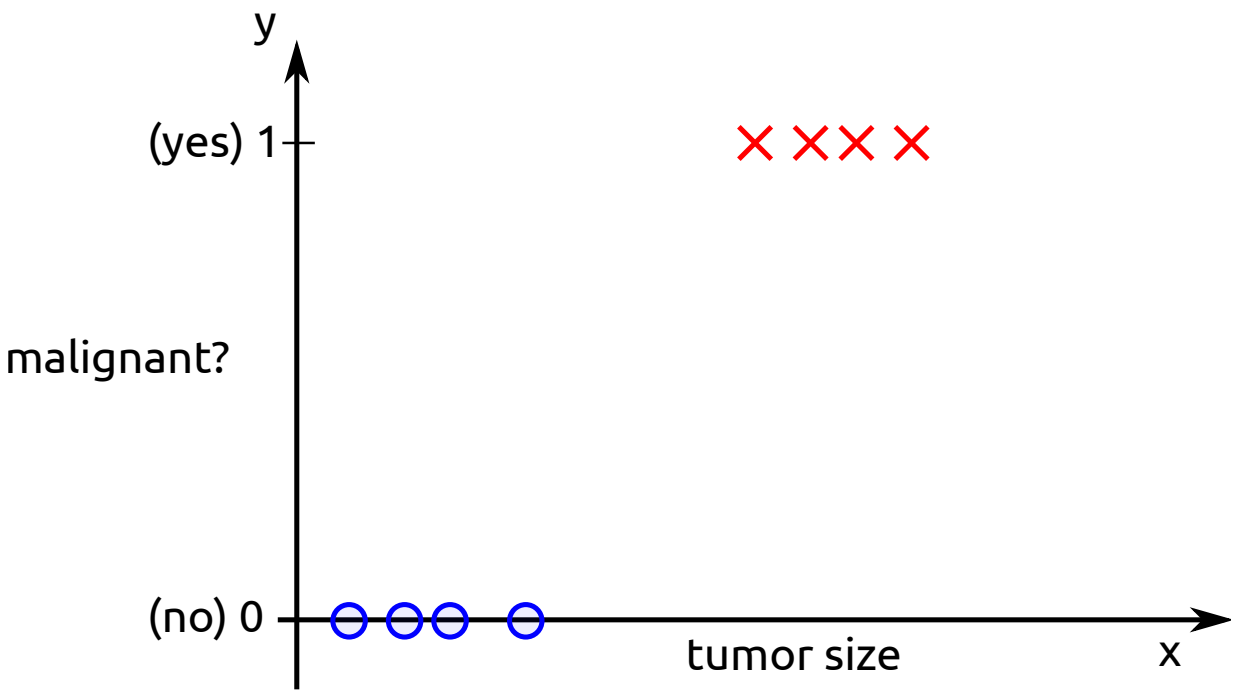
Labels $y \in \{0, 1\}$, with

- $y = 0$: "Negative class" (e.g. benign, not malignant)
- $y = 1$: "Positive class" (e.g. malignant)

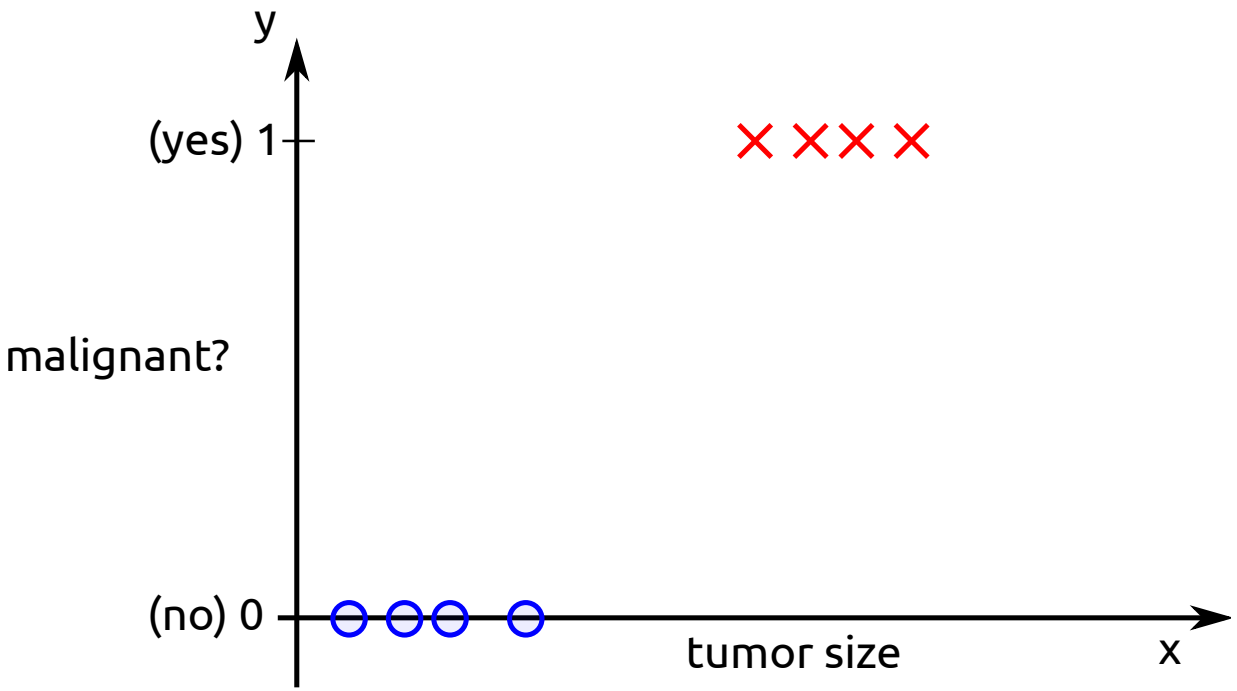


Try fitting a (augmented) model $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ as in linear regression...

Oracle Solution for Tumor Classification



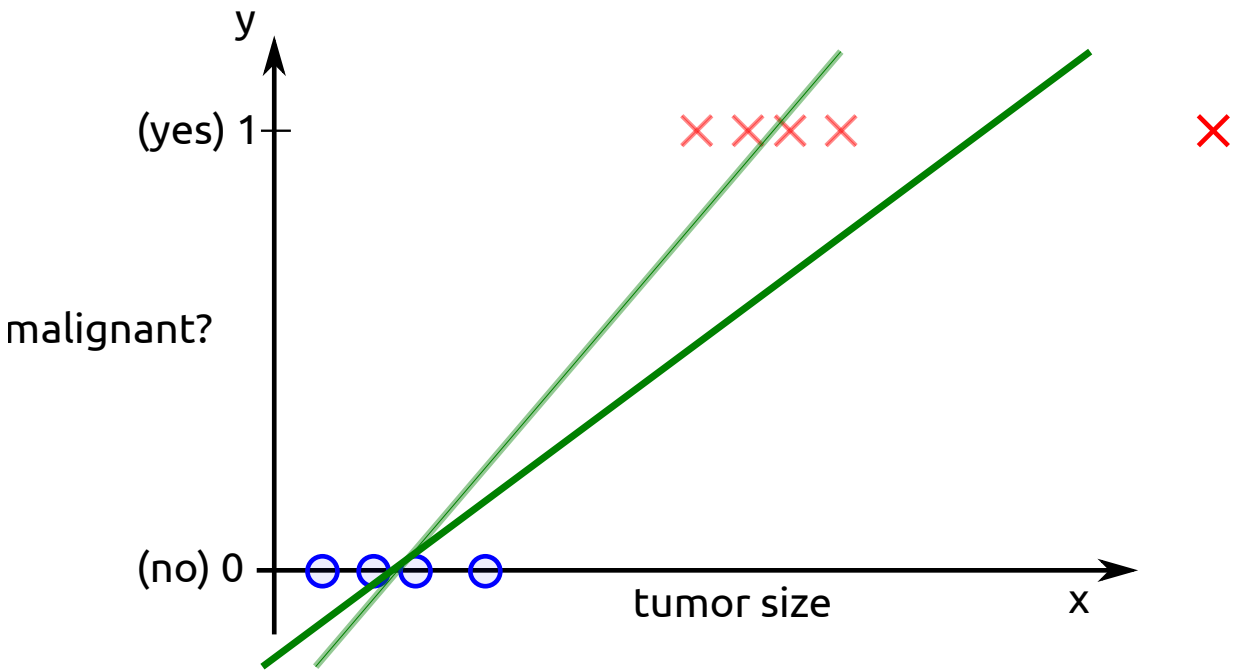
Oracle Solution for Tumor Classification



Oracle solution is:

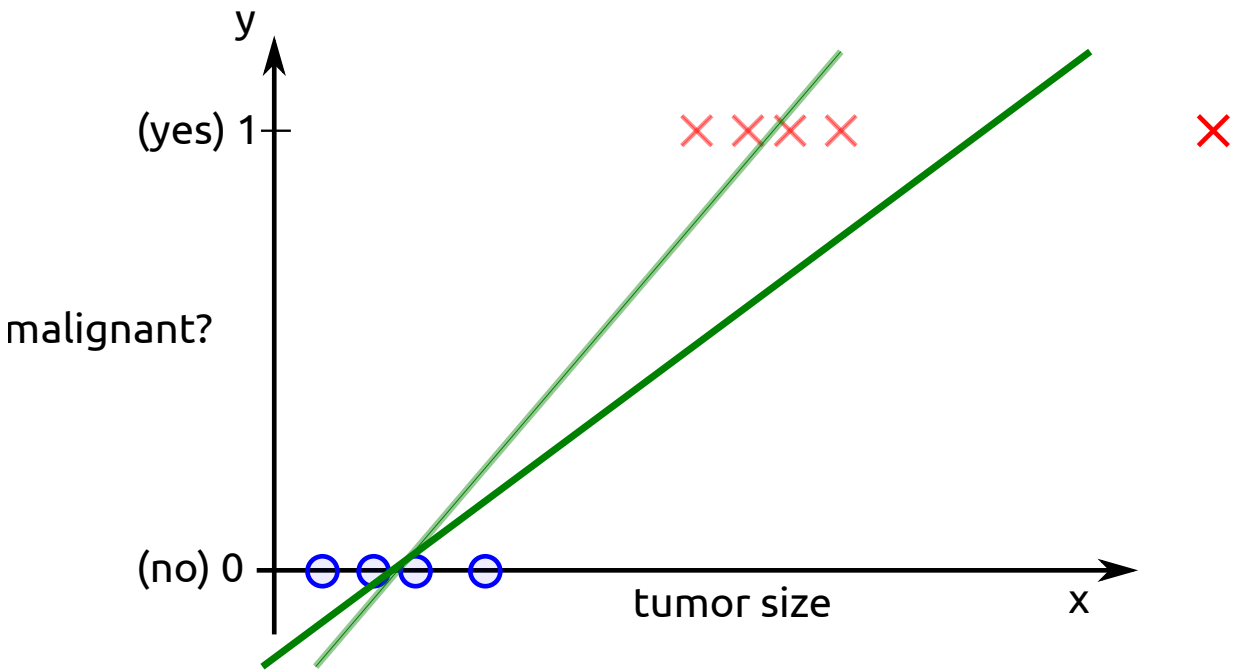
- If $h_{\mathbf{w}}(\mathbf{x}) > 0.5$, predict $y = 1$: (malignant)
- If $h_{\mathbf{w}}(\mathbf{x}) \leq 0.5$, predict $y = 0$: (not malignant)

Tumor Classification with Linear Regression



- For classification case we should only accept $y = 0$ or $y = 1$.
- However, linear regression model $h_{\mathbf{w}}(\mathbf{x})$ can generate $y > 1$ or $y < 0$.

Tumor Classification with Linear Regression



- For classification case we should only accept $y = 0$ or $y = 1$.
- However, linear regression model $h_{\mathbf{w}}(\mathbf{x})$ can generate $y > 1$ or $y < 0$.

Solution: logistic regression, which bounds the output to $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$

Logistic Regression Model

We want: $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$

Standard linear regression
(using the inner product) delivers:
$h_{\mathbf{w}}(\mathbf{x}) = \quad \mathbf{w}^T \mathbf{x}$

Logistic Regression Model

We want: $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$

Standard linear regression
(using the inner product) delivers:
$h_{\mathbf{w}}(\mathbf{x}) = \quad \mathbf{w}^T\mathbf{x}$

A subtle change introduces non-linearity:
$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$ with $g(z) = \frac{1}{1+e^{-z}}$

Logistic Regression Model

We want: $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$

Standard linear regression
(using the inner product) delivers:
$h_{\mathbf{w}}(\mathbf{x}) = \quad \mathbf{w}^T\mathbf{x}$

A subtle change introduces non-linearity:
$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$ with $g(z) = \frac{1}{1+e^{-z}}$

$$\Rightarrow h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

Logistic Regression Model

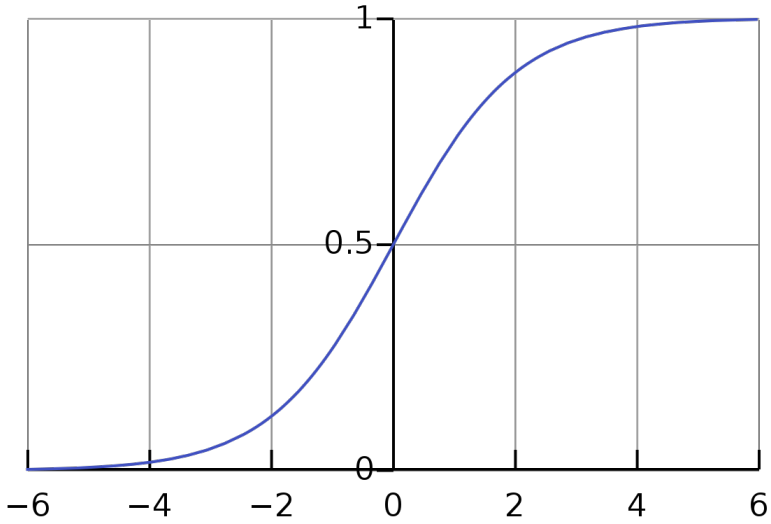
We want: $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$

Standard linear regression (using the inner product) delivers:
$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

$g(z)$ is called sigmoid function or logistic function

A subtle change introduces non-linearity:
$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$ with $g(z) = \frac{1}{1+e^{-z}}$

$$\Rightarrow h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$



Interpretation of Model Output

$h_{\mathbf{w}}(\mathbf{x}) \approx$ estimated probability, that $y = 1$

More formally, we work with a *hypothesis*:

Interpretation of Model Output

$h_{\mathbf{w}}(\mathbf{x}) \approx$ estimated probability, that $y = 1$

More formally, we work with a *hypothesis*:

$h_{\mathbf{w}}(\mathbf{x}) = P(y = 1|\mathbf{x}; \mathbf{w})$

"probability that $y = 1$,
given that the input is $\mathbf{x}$ and
the model is parameterized by $\mathbf{w}$"

Interpretation of Model Output

$h_{\mathbf{w}}(\mathbf{x}) \approx$ estimated probability, that $y = 1$

More formally, we work with a *hypothesis*:

$h_{\mathbf{w}}(\mathbf{x}) = P(y = 1|\mathbf{x}; \mathbf{w})$

"probability that $y = 1$,
given that the input is $\mathbf{x}$ and
the model is parameterized by $\mathbf{w}$"

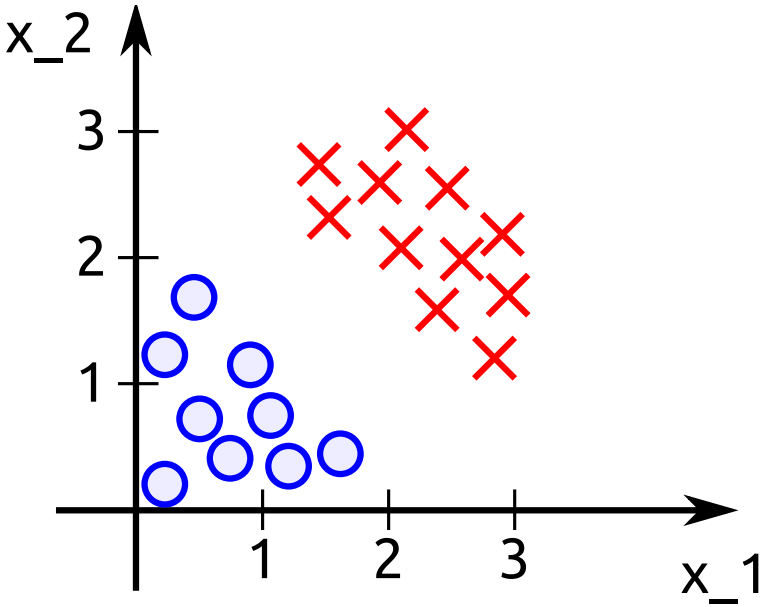
The actual labels are still discrete ($y = 0$ or $y = 1$), but the probabilities need to add to one:

$$P(y = 0|\mathbf{x}; \mathbf{w}) + P(y = 1|\mathbf{x}; \mathbf{w}) = 1$$

$$P(y = 0|\mathbf{x}; \mathbf{w}) = 1 - P(y = 1|\mathbf{x}; \mathbf{w})$$

Example in $\mathbb{R}^2$

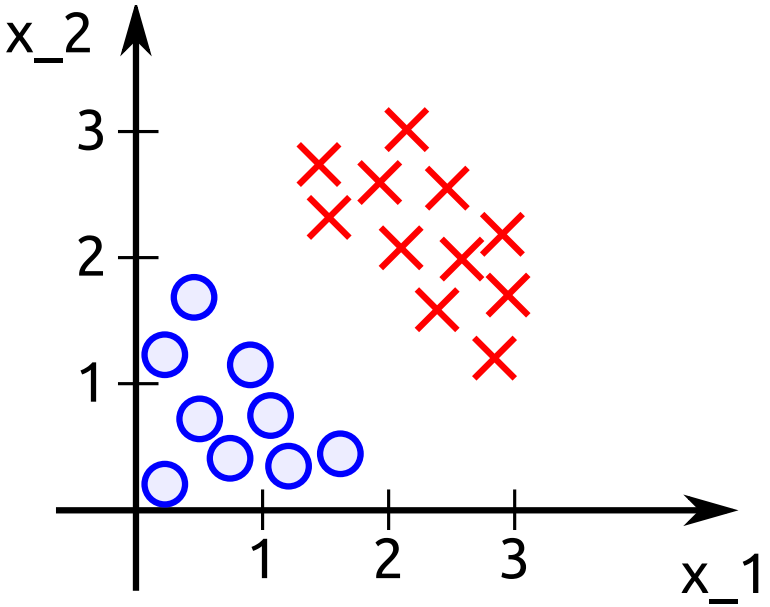
Where is the decision boundary?



$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2)$$

Example in $\mathbb{R}^2$

Where is the decision boundary?



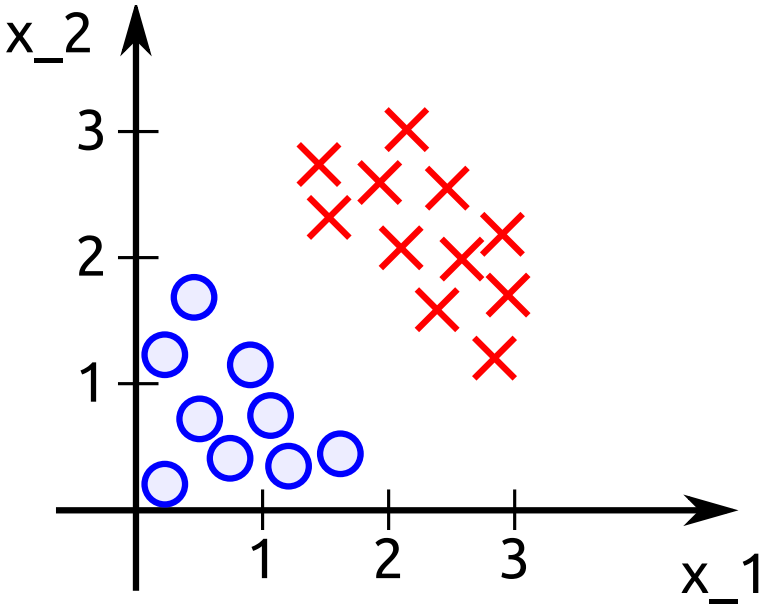
$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2)$$

$$\mathbf{w} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

Predict "$y = 1$" if $\quad -3 + \mathbf{x}_1 + \mathbf{x}_2 \geq 0$

Example in $\mathbb{R}^2$

Where is the decision boundary?



$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2)$$

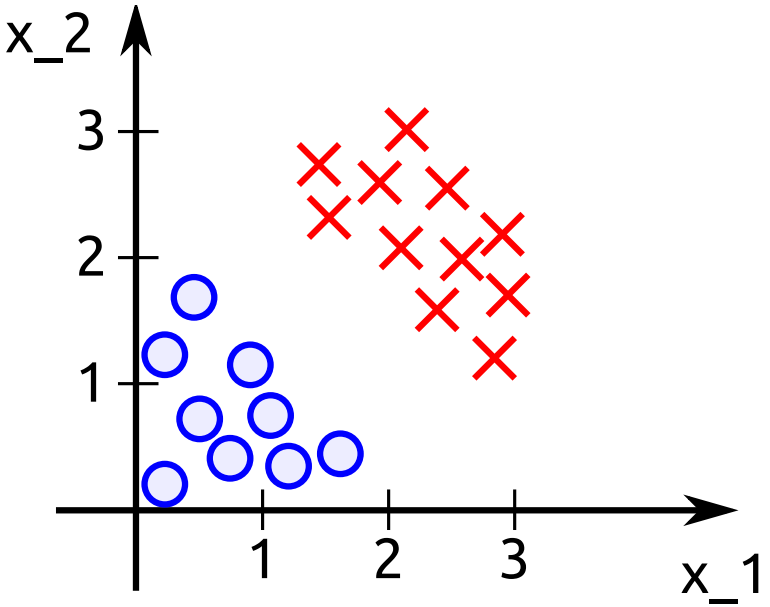
$$\mathbf{w} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

Predict "$y = 1$" if $-3 + \mathbf{x}_1 + \mathbf{x}_2 \geq 0$

$$\mathbf{x}_1 + \mathbf{x}_2 = 3 \Leftrightarrow h_{\mathbf{w}}(\mathbf{x}) = 0.5$$

Example in $\mathbb{R}^2$

Where is the decision boundary?



$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2)$$

$$\mathbf{w} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

Predict "$y = 1$" if $-3 + \mathbf{x}_1 + \mathbf{x}_2 \geq 0$

$$\mathbf{x}_1 + \mathbf{x}_2 = 3 \Leftrightarrow h_{\mathbf{w}}(\mathbf{x}) = 0.5$$

Remark: as in linear models, the use of additional dimensions and basis functions allows for non-linear decision boundaries!

Loss Function of Logistic Regression (I)

Proposed loss function (specifically adapted to logistic regression):

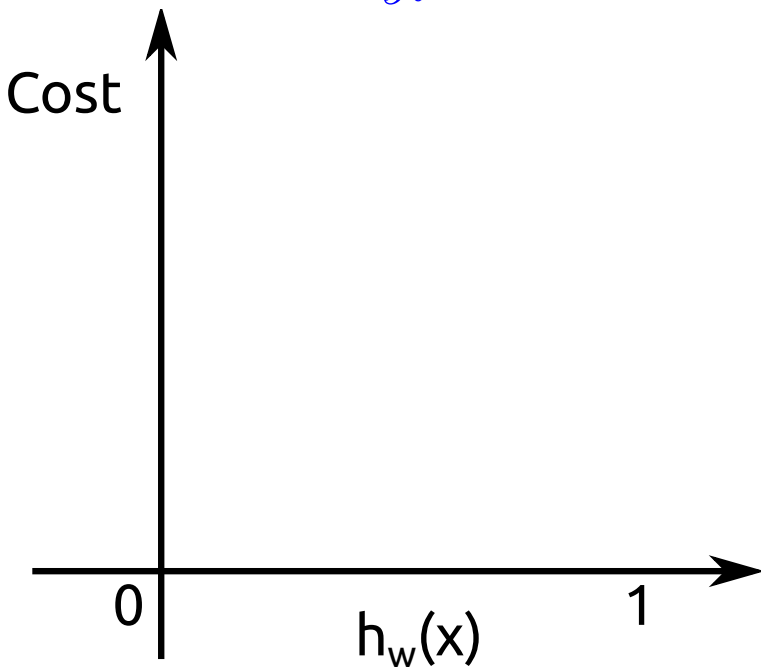
$$J(h_{\mathbf{w}}(y_i, \mathbf{x}_i)) = \begin{cases} -\log(h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 1 \\ -\log(1 - h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 0 \end{cases}$$

Loss Function of Logistic Regression (I)

Proposed loss function (specifically adapted to logistic regression):

$$J(h_{\mathbf{w}}(y_i, \mathbf{x}_i)) = \begin{cases} -\log(h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 1 \\ -\log(1 - h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 0 \end{cases}$$

Case: $y_i = 1$

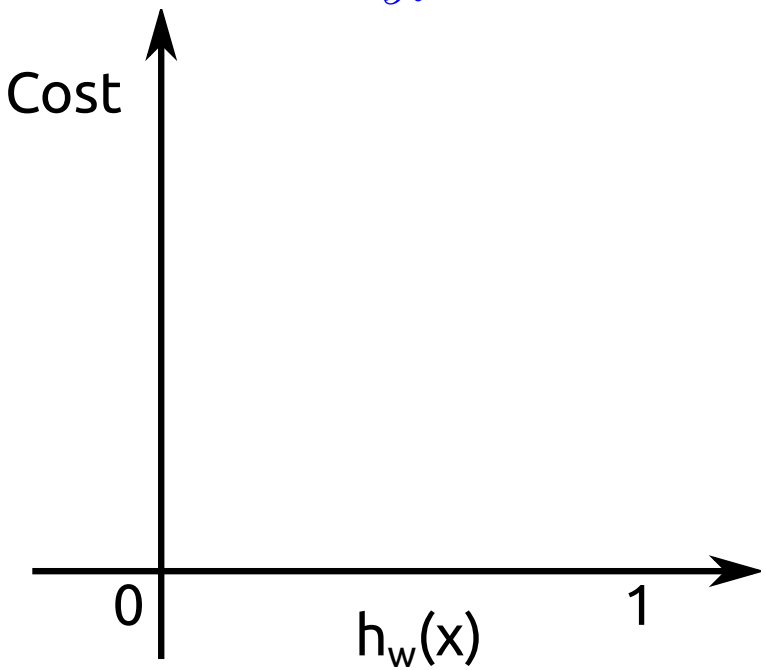


Loss Function of Logistic Regression (I)

Proposed cost function for the logistic regression model:

$$J(h_{\mathbf{w}}(y_i, \mathbf{x}_i)) = \begin{cases} -\log(h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 1 \\ -\log(1 - h_{\mathbf{w}}(\mathbf{x}_i)) & \text{for} \quad y_i = 0 \end{cases}$$

Case: $y_i = 0$



Loss Function for Logistic Regression (II)

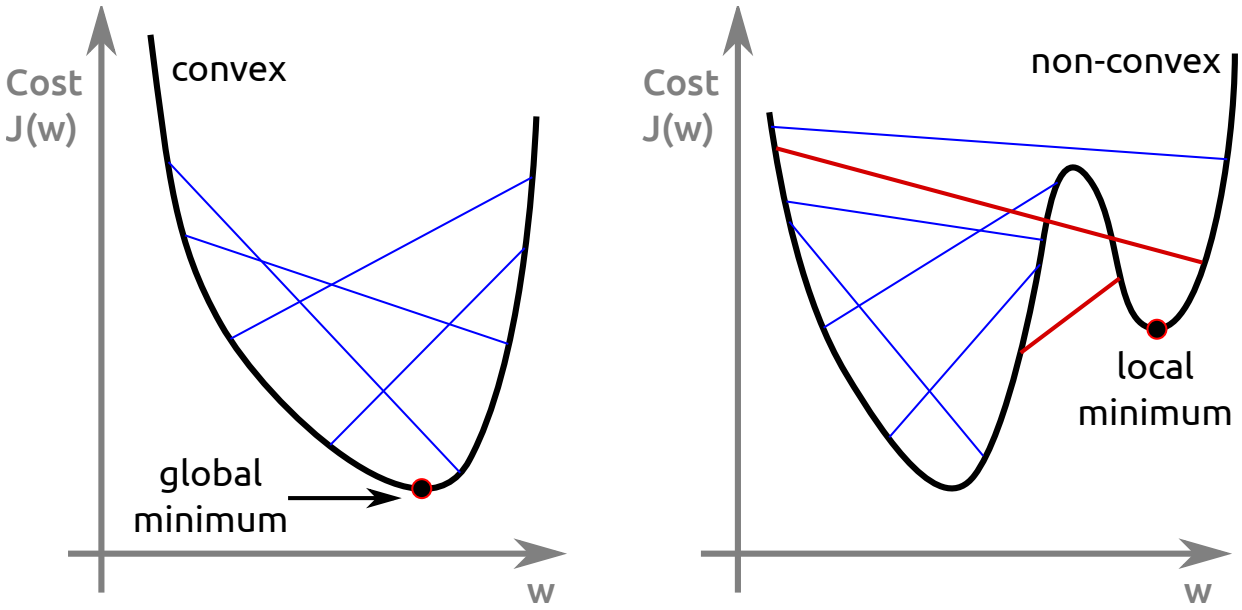
The case distinction can be avoided using this formulation (check by setting $y_i = 0$ and $y_i = 1$):

$$J(h_{\mathbf{w}}(\mathbf{x}_i), y) = -y_i \log(h_{\mathbf{w}}(\mathbf{x}_i)) - ((1 - y_i) \log(1 - h_{\mathbf{w}}(\mathbf{x}_i)))$$

- Nice property: J is convex.
- Not so nice property: There is no analytic solution (but gradient descent works well).

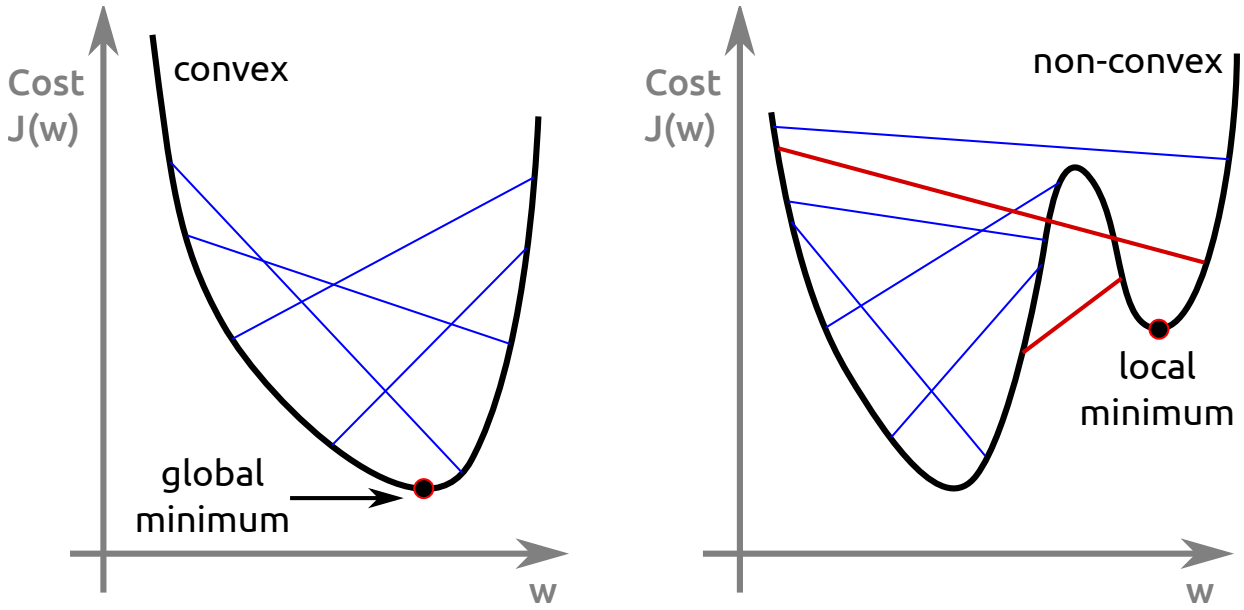
Convex Loss functions

- Loss for data point $(\mathbf{x}, y)$ depends on the choice of $\mathbf{w}$: $J(h_{\mathbf{w}}, y)$.



Convex Loss functions

- Loss for data point $(\mathbf{x}, y)$ depends on the choice of $\mathbf{w}$: $J(h_{\mathbf{w}}, y)$.
- Parameters w_i can be optimized easier, if the loss function is convex and (continuously) differentiable.



Iterative Optimization

Loss to minimize:

$$\begin{aligned}
&\arg\min_w J(w) \\
&= \arg\min_w \sum_{i=1}^{N} -y_i \log\left(h_w(x_i)\right) - (1-y_i)\log\left(1-h_w(x_i)\right)
\end{aligned}$$

- Start with a guess for w
- Move w towards the minimum of $J(w)$

How to train w for minimizing the loss?

Minimize $J(w)$ by updating w in the negative direction of $\frac{\partial J(w)}{\partial w}$:

$$w^{(\text{next})} \leftarrow w^{(\text{prev})} - \eta \frac{\partial J(w)}{\partial w^{(\text{prev})}}$$

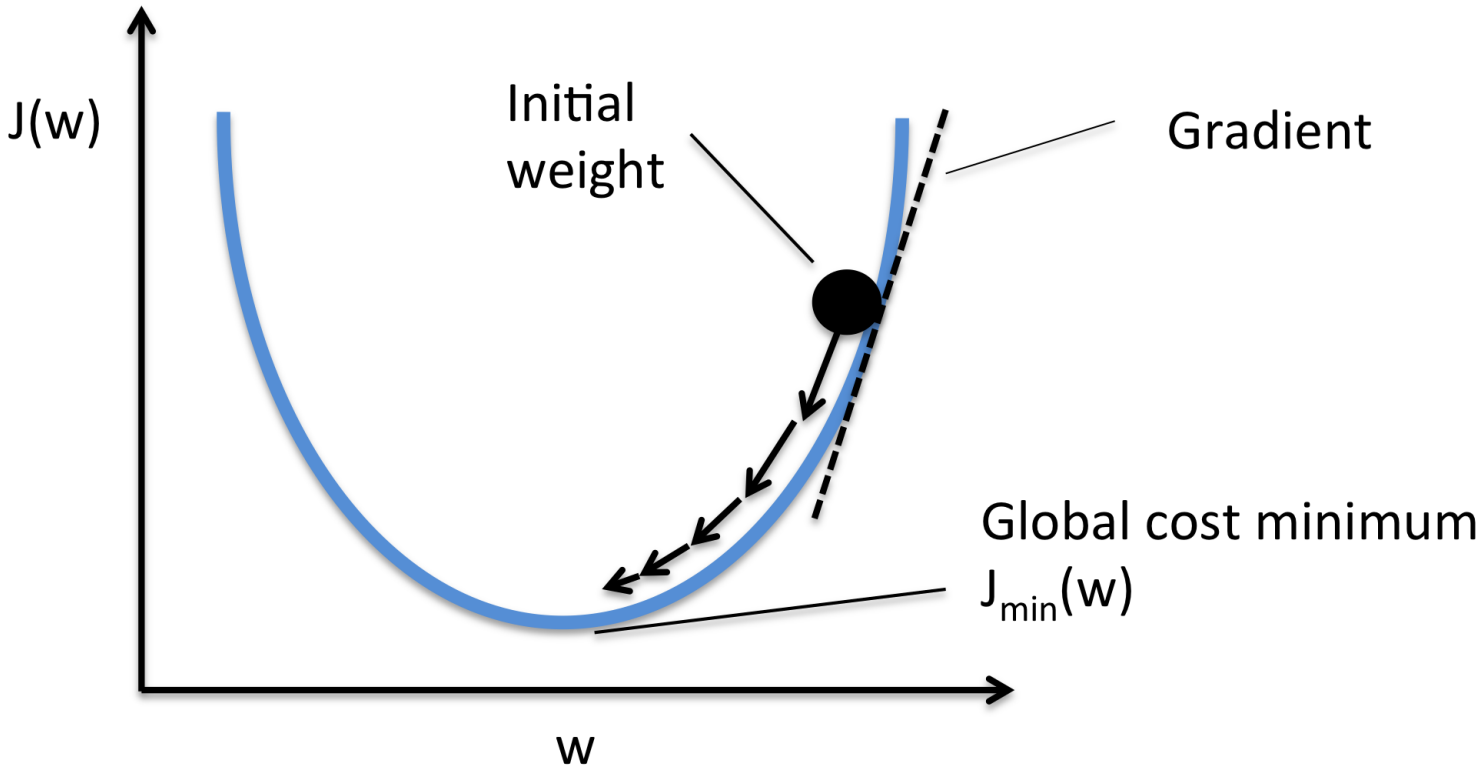


Figure: Source: http://raghd.github.io/

Gradient Descent

Find the optimal parameters $w^* \in \mathbb{R}^K$ that minimize an objective function $J(w)$, given data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, i.e.:

$$w^* := \arg\min_w J(\mathcal{D}, w)$$

Algorithm 1: Gradient Descent Optimization

Require: Data $\mathcal{D}$, Learning rate $\eta \in \mathbb{R}^+$, Iterations $\mathcal{I} \in \mathbf{N}^+$

Ensure: $w \in \mathbb{R}^K$

1: $w \sim \mathcal{N}(0, \sigma^2)$
2: **for** $1, \ldots, \mathcal{I}$ **do**
3: $\quad w \leftarrow w - \eta \frac{\partial J(\mathcal{D}, w)}{\partial w}$
4: **end for**
5: **return** w

Stochastic Gradient Descent

Divide the dataset into R partitions (mini-batches) as $\mathcal{D} = \bigcup_{r=1}^{R} \mathcal{D}_r$, yielding a decomposition of the loss:

$$J(\mathcal{D}, w) \quad := \quad \sum_{r=1}^{R} J(\mathcal{D}_r, w) \quad := \quad \sum_{r=1}^{R} J_r$$

Algorithm 2: Stochastic Gradient Descent Optimization

Require: Data $\mathcal{D} = \bigcup_{r=1}^{R} \mathcal{D}_r$, Learning rate $\eta \in \mathbb{R}^+$, Iters $\mathcal{I} \in \mathbb{N}^+$

Ensure: $w \in \mathbb{R}^K$

1: $w \sim \mathcal{N}(0, \sigma^2)$
2: **for** $1, \ldots, \mathcal{I}$ **do**
3: **for each** $r \in \{1, \ldots, R\}$ *in random order* **do**
4: $w \leftarrow w - \eta \frac{\partial J_r}{\partial w}$
5: **end for**
6: **end for**

Gradients

- For Linear Regression

$$\frac{\partial J(w)}{\partial w} = \frac{\partial \left[\sum_{i=1}^{N} \left(y_i - f(x_i; w)\right)^2\right]}{\partial w} = \sum_{i=1}^{N} -2\left(y_i - f(x_i; w)\right) x_i$$

- For Logistic Regression

$$\frac{\partial J(w)}{\partial w} = \frac{\partial \left[\sum_{i=1}^{N} -y_i \log\left(h_w(x_i)\right) - (1 - y_i) \log\left(1 - h_w(x_i)\right)\right]}{\partial w}$$

$$= \sum_{i=1}^{N} -\left(y_i - h_w(x_i)\right) x_i$$