Regression for Perceptron

1. **Motivation**
2. **Linear Regression**
3. **Linear Regression for polynomial regression tasks**

Motivation

$$H^L = \{h_{w,b} : \boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}} \ \ \boldsymbol{and}\ \boldsymbol{b} \in \mathbb{R}\} \quad |H^L| = \infty$$

$$\boldsymbol{h_{w,b}(x_i) = \langle w, x_i \rangle + b}$$

$H^L$is one of the most useful families of hypothesis classes.

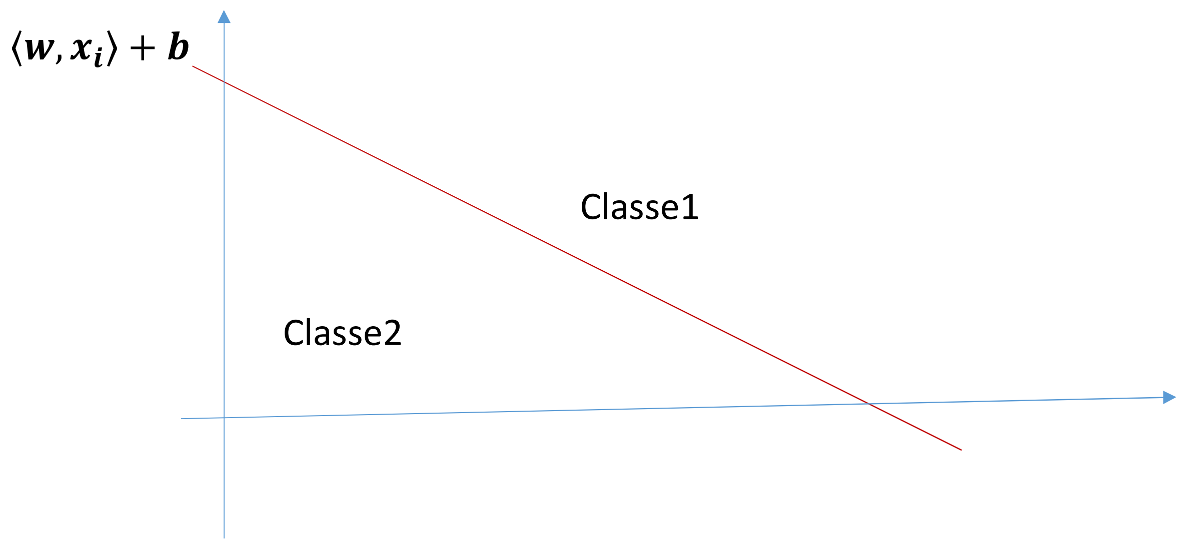
Many models that are used in practice rely on linear predictors.

There exist many types of linear models, including:

- Perceptron(classification). $\boldsymbol{y} \in \{\boldsymbol{0}, \boldsymbol{1}\}$
- Linear regression. $\boldsymbol{y} \in \mathbb{R}$
- Logistic regression. $\text{output}\ \boldsymbol{probability}\ \boldsymbol{p} \in\ (\boldsymbol{0}, \boldsymbol{1})\boldsymbol{and\ input\ y} \in \{-1, +1\}$

Motivation

Perceptron(classification). $\boldsymbol{h_{w,b}(x_i) = sign(\langle w, x_i \rangle + b)}$ $\boldsymbol{y} \in \{-1, +1\}$
Linear regression. $\boldsymbol{y_i} \in \mathbb{R}$ $\boldsymbol{h_{w,b}(x_i) = \langle w, x_i \rangle + b = \ y_i}$



Motivation

Let's define the class of affine functions L=L_d: input = $\boldsymbol{x}$

$$H^{L_d} = \{h_{w,b} : (\boldsymbol{w}, \boldsymbol{b}) \in \mathbb{R}^{\boldsymbol{d+1}}\}, \qquad |H^{L_d}| = \infty$$

Where:

$$\boldsymbol{h_{w,b}(x_i) = \langle w, x_i \rangle + b = \sum_{j=1}^{d} w_j} x_i^j \boldsymbol{+ b = y_i}$$

To simplify the notation, we will integrate the bias as an extra coordinate into w:

$$x_i = (x_i^1, \dots, x_i^d) \in \mathbb{R}^d \rightarrow x_i = (1, x_i) \in \mathbb{R}^{d+1}, w \rightarrow (b, w) \in \mathbb{R}^{d+1}$$

$$\boldsymbol{h_w}(x_i) = \sum_{\boldsymbol{j=0}}^{\boldsymbol{d}} \boldsymbol{w_j} x_i^j \ \boldsymbol{avec} \quad \boldsymbol{x_i^0 = 1 \ and \ w_0 = b}$$

Hence, the class of affine functions is called « homogenous affine functions »

$$H^{L_d} = \boldsymbol{L_d = \{h_w : w \in \mathbb{R}^{d+1}\}} \rightarrow |L_d| = \infty$$

$$\boldsymbol{h_w}(x_i) = \boldsymbol{y_i}$$

Motivation

Therefore, we can generate different hypothesis classes H^L, defining different models, by using the composition of φ over L_d such that: ($\boldsymbol{h_w} \in L_d$)

$$\varphi : \mathbb{R} \rightarrow Y$$

- **Perceptron(classification):**

$$\varphi_p(x) = sign(x) \text{ and } Y = \{-1, +1\}$$

$$H_p = sing(\varphi_p \circ L_d = \{\varphi_p \circ \boldsymbol{h_w}(\boldsymbol{x}) : \boldsymbol{h_w} \in L_d\})$$

- **Linear regression:**

$$\varphi_{reg}(x) = Id(x) = x \text{ and } Y = \mathbb{R}$$

$$H_{reg} = \varphi_{reg} \circ L_d = \{\varphi_p \circ \boldsymbol{h_w}(\boldsymbol{x}) : \boldsymbol{h_w} \in L_d\} = \{\boldsymbol{h_w}(\boldsymbol{x}) : \boldsymbol{h_w} \in L_d\}$$

- **Logistic regression:**

$$\boldsymbol{\varphi_{sig}(x) = \frac{1}{1+e^{-x}}} \textbf{ and } Y = \{-1, +1\}, \boldsymbol{\varphi_{sig}\left(h_w\left(x_j^i\right) = \sum_{i=0}^{d} w_i x_j^i\right) = \frac{1}{1+e^{-\sum_{i=0}^{d} w_i x_i}}}$$

$$\boldsymbol{H_{sig} = \varphi_{sig} \circ L_d(x) = \frac{1}{1 + e^{-\sum_{i=0}^{d} w_i x_i}}}$$

Linear Regression

Definition:

Linear regression is a type of model used for regression tasks by studying the relationship between some explanatory variables and some real valued outcome.

Here we have: $x = (1, x) \in X$

$$X \subset \mathbb{R}^{d+1} \text{ for some } d$$

And

$$Y = \mathbb{R}$$

Objective:

Learn a linear predictor $h_w \in L_d$ that best approximate the relationship between our variables:

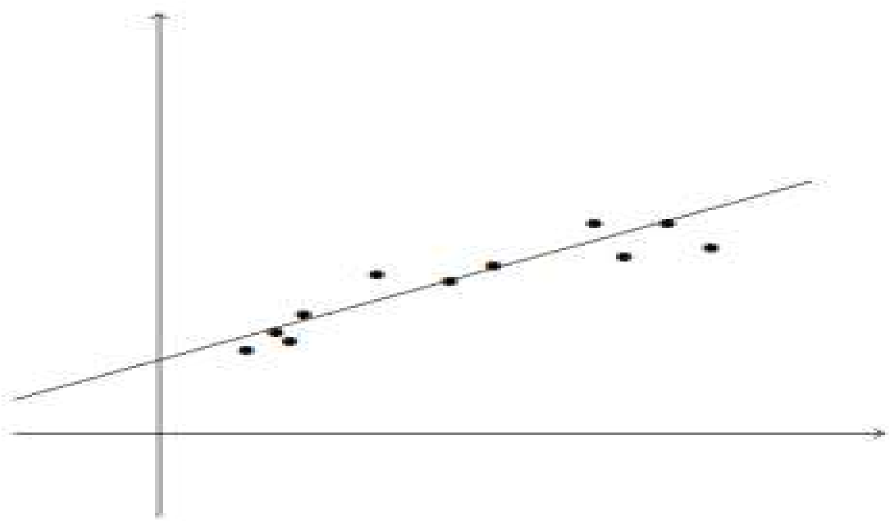
$$\begin{aligned} h_w: \quad \mathbb{R}^{d+1} &\longrightarrow \qquad \mathbb{R} \\ x \quad &\longrightarrow \quad h_w(x) = w^t x = y \end{aligned}$$

Linear Regression

Example:

- $h_w(x) = w^t x_i = y_i \Longrightarrow L_S(w) = \frac{1}{m}\sum_{i=1}^{m} d(w^t x_i, y_i) \ = \frac{1}{m}\sum_{i=1}^{m}(w^t x_i - y_i)^2 \ \approx \|w^t x - y\|^{\mathbf{2}}$
- $\min_{w \in \mathbb{R}^{d+1}} L_S(w) \approx 0$

Predicting the weight of a baby as a function of his age and weight at birth.
Here, $d = 1$.



Linear Regression

The hypothesis class for linear regression model:

In linear regression model, we have:

$$\varphi_{regL}(x) = Id(x) = x \Longrightarrow \varphi_{regL} \circ \boldsymbol{h_w}(\mathrm{x}) = \mathrm{Id}(\boldsymbol{h_w}(\mathrm{x})) = \boldsymbol{h_w}(\mathrm{x})$$

The hypothesis class of linear regression predictors is simply the set of linear functions:

$$H_{reg} \;\; = \varphi \circ L_d = L_d = \left\{\varphi_{regL} \circ \boldsymbol{h_w} \colon \boldsymbol{h_w} \in \; L_d\right\}$$

$$\boldsymbol{H_{reg}} = \{\boldsymbol{h_w} \colon\; \boldsymbol{x} \longmapsto \langle \boldsymbol{w}, \boldsymbol{x} \rangle \;:\; \boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d+1}}\} \rightarrow \left|\boldsymbol{H_{reg}}\right| \approx \infty$$

Linear Regression

The loss function for linear regression model:
It measures how much the model should be penalized for the discrepancy between $h_w(x)$ and y. One common way is to use the squared-loss function:

$$0 \approx d(h_w(x), y) = l\big(h_w, (x, y)\big) = (h_w(x) - y)^2$$

For this loss function, the empirical risk is called the Mean Squared Error:

$$L_S(h_w) = E_{empi}\Big(l\big(h_w, (x, y)\big)\Big) = \frac{1}{m}\sum_{i=1}^{m}(h_w(x_i) - y_i)^2$$

Notice:
There are a variety of other loss functions that one can use, for example, the absolute value loss function:

$$l\big(h_w, (x, y)\big) = |h_w(x) - y| \Longrightarrow \boldsymbol{\partial l\big(h_w, (x, y)\big)}\ \boldsymbol{is\ a\ set}\ \textbf{if}\ h_w(x) = y$$

Linear Regression: Best Metric

- $\|w^t x - y\|_2 = \sqrt{\sum_{i=1}^{\infty}(w^t x_i - y_i)^2} \leftrightarrow \sum_{i=1}^{\infty}(w^t x_i - y_i)^2$
- $\|w^t x - y\|_1 = \sum_{i=1}^{\infty}|(w^t x_i - y_i|$
- $\|w^t x - y\|_\infty = \max(|(w^t x_i - y_i|)$
- $L_S(w) = \frac{1}{m}\sum_{i=1}^{m} d(w^t x_i, y_i) \;\rightarrow \min_{w\in\mathbb{R}^{d+1}} L_S(w) \approx 0$
- $d_2(w^t x_i, y_i) = (w^t x_i - y_i)^2$
- $d_1(w^t x_i, y_i) = |w^t x_i - y_i| \rightarrow ?\, diff: 1{,}2$
- $\varepsilon = 0{,}02$
- $Min\ f(x) = |x| \rightarrow |x| = \varepsilon = 0{,}02 \rightarrow x = \pm 0{,}02$ (subgradient algorithm by yourself)
- $Min\ g(x) = x^2 \rightarrow x^2 = \varepsilon = 0{,}02 \rightarrow x = \sqrt{0{,}02} = 0{,}14$
- $|w^t x_i - y_i| = 0{,}0002 \rightarrow w^t x_i - y_i = \pm 0{,}0002$
- $(w^t x_i - y_i)^2 = 0{,}0002 \rightarrow w^t x_i - y_i = \sqrt{0{,}0002} = 0{,}014$

Linear Regression

The learning algorithm for linear regression model:

The learning algorithm follows ERM_H learning rule.

Least squares:

Least squares is the algorithm that solves the ERM_H problem for the hypothesis class of linear regression predictors with respect to squared loss.

$$w^* = \underset{w \in \mathbb{R}^{d+1}}{\operatorname{argmin}} L_S(h_w) = \underset{w \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 \right)$$

To solve this problem, we calculate the gradient of the objective function and compare it to zero. That is, we need to solve:

$$\nabla L_S(h_w) = \frac{2}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i) x_i = 0 \longrightarrow \boldsymbol{\nabla^2 L_S(h_w)} = \frac{2}{m} \sum_{i=1}^{m} x_i (x_i)^T = \boldsymbol{cte}$$

Linear Regression

We can rewrite the problem as the problem :

$$\nabla L_S(h_w) = \frac{2}{m}\sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)x_i = 0 \Rightarrow \sum_{i=1}^{m}(\langle w, x_i\rangle x_i) - \sum_{i=1}^{m} y_i x_i = 0$$

- $\Longrightarrow Aw - b = 0 \Longrightarrow Aw = b$
- $\Longrightarrow \boldsymbol{\nabla^2 L_S(h_w) = A}$

Where:

$$A = \left(\sum_{i=1}^{m} x_i . x_i^T\right), \qquad b = \sum_{i=1}^{m} y_i x_i$$

Linear Regression

$$Aw = b$$

Or in matrix $A(d \times d)$ (symetric) form: $A = \left(\sum_{i=1}^{m} x_i . x_i^T\right)$, $b = \sum_{i=1}^{m} y_i x_i$

- $A^T = A = XX^T = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix}^T : X = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix}$

And

- $b = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$

If A is invertible, then the solution to the ERM problem is:

$$w = A^{-1} b$$

Linear Regression: Example 1

- $x_1^T = (1{,}0{,}0), x_2^T = (1{,}1{,}0), x_3^T = (0{,}1{,}0)$
- $A = \left(\sum_{i=1}^{3} x_i . x_i^T\right) = x_1 . x_1^T + x_2 . x_2^T + x_3 . x_3^T$
- $= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (1{,}0{,}0) + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} (1{,}1{,}0) + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (0{,}1{,}0)$
- $= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
- A isn’t inve

Linear Regression: Example2

- $x_1^T = (1,0,0), x_2^T = (1,1,0), x_3^T = (0,0,1)$
- $A = \left(\sum_{i=1}^{3} x_i . x_i^T\right) = x_1 . x_1^T + x_2 . x_2^T + x_3 . x_3^T$
- $= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (1,0,0) + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} (1,1,0) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} (0,0,1)$
- $= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- A is inv

Linear Regression

If A is not invertible, we require a few standard tools from linear algebra.

A is not invertible when the training data do not cover the entire space of $\mathbb{R}^d$.
Even if A is not invertible, we can always find a solution to the system:

$$Aw = b$$

because b is in the range of A.

Indeed, since **A is symmetric**, then we can write it using its eigenvalue decomposition as:

$$\boldsymbol{A = VDV^T}$$

Where:
D is a diagonal matrix.
V is an orthonormal matrix (because $V^T V = I$ which is a $d \times d$ matrix).

Linear Regression

Let's define D^+ to be the diagonal matrix such that:

$$\begin{cases} \boldsymbol{D^+_{i,i} = 0 \quad if \quad D_{i,i} = 0} \\ \boldsymbol{D^+_{i,i} = \dfrac{1}{D_{i,i}} \quad if \quad D_{i,i} \neq 0} \end{cases} \Rightarrow \boldsymbol{DD^+ = I \setminus D_{i,i} = 0?}$$

Now, define:

$$A^+ = VD^+V^T \text{ and } \boldsymbol{\hat{w} = A^+ b}$$

Let v_i denote the ith column of V. Then we have:

$$\boldsymbol{A\hat{w} = AA^+b = VDV^TVD^+V^Tb = VDD^+V^Tb = VV^Tb = \sum_{i\,:\,D_{i,i}\neq 0} v_i v_i^T b}$$

This means that $A\hat{w}$ is the projection of b on the space of vectors v_i for which $D_{i,i} \neq 0$.

Linear Regression

Since the linear space of $(x_1, \dots, x_m) \in \mathbb{R}^m$ is the same as the linear space of those $\{v_i\}$.

And, since b is in the linear space of x_i.

We obtain that:

$$A\widehat{w} = b$$

Then, $\widehat{w}$ is a solution of $Aw = b$.

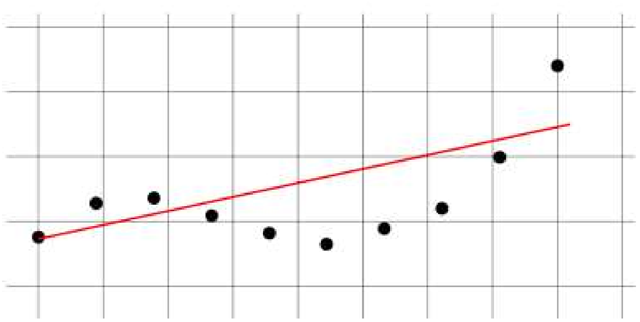
Linear Regression for polynomial regression tasks $x \in \mathbb{R}$

Some learning tasks call for nonlinear predictors, such as polynomial predictors.
Let's consider a one dimensional(one feature) polynomial function of degree n: $A_\alpha \;\; n = \alpha$

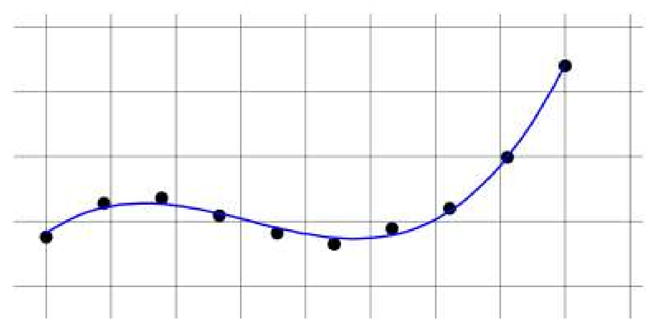
- $P_1(x) = w_0 + w_1 x$
- $P_n(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n \Longrightarrow P_w\ (Z) = w_0 * 1 + w_1 z_1 + w_2 z_2 + \cdots + w_n z_n$

With $\mathbf{z} = (1, z_1, \dots, z_n) \leftarrow \mathbf{z} = (1, x, \dots, x^n) \in \{1\} \times \mathbb{R}^n \;\Leftrightarrow\; \mathbf{space}(X) \curvearrowright \boldsymbol{space}(Z)$
Where $\mathrm{w} = (w_0, \dots, w_n)$ is a vector of coefficients of size $n + 1$.

$P_1(x) = w_0 + w_1 x$



$P_3(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$



Linear Regression for polynomial regression tasks

We will focus on the class of one dimensional, n-degree, polynomial regression hypotheses. Therefore, the class of polynomial hypotheses is:

$$\boldsymbol{H_{poly}^{n}} = \left\{\boldsymbol{P_{w,n=\alpha}}: \boldsymbol{X} \subseteq \mathbb{R} \longmapsto \mathbb{R}: \boldsymbol{n} \in \mathbb{N}^{*}, \boldsymbol{w} \in \mathbb{R}^{\boldsymbol{n+1}} \right\} \Longrightarrow \left|\boldsymbol{H_{poly}^{n}}\right| = \infty, \boldsymbol{A_{\alpha}} = \boldsymbol{P_{w,n=\alpha}}$$

Where $\boldsymbol{P_{w,n=\alpha}}$ is a one dimensional polynomial of degree n, parameterized by a vector of coefficients $(w_0, \ldots, w_n)$.
In that case, we have:

$$X \subseteq \mathbb{R} \quad \text{and} \quad Y \subseteq \mathbb{R}$$

One way to learn the class H_{poly}^{n} is by reduction to the problem of linear regression.

Linear Regression for polynomial regression tasks

To translate a polynomial regression problem to a linear regression problem, we define the mapping:

$$\psi_n : \mathbb{R} \longrightarrow \mathbb{R}^{n+1}$$

Such that:

$$\boldsymbol{\psi_n(x) = (1, x, x^2, \dots, x^n)^{\ T}}$$

Then, we have that: $\boldsymbol{P_{w,n} o \psi_n(x) = P_{w,n}\left(\psi_n(x)\right)}$

$$\boldsymbol{P_{w,n}\left(\psi_n(x)\right) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n = \langle w, \psi_n(x) \rangle \Longrightarrow \nabla_w P_{w,n} = \psi_n(x)}$$

Finally, we can find the optimal vector of coefficients w by using the Least Squares Algorithm.

Logistic Regression

Definition:

Logistic regression is a type of model used for classification tasks by studying the relationship between some explanatory variables and some binary outcome.

Here we have:

$$X \subset \mathbb{R}^d \text{ for some } d \text{ and } Y = \{-1, +1\}$$

Objective:

Learn a linear predictor that best approximate the relationship between our variables:

$$h_w : \mathbb{R}^d \longrightarrow [0,1]$$

We can interpret $h_w(x)$ as the probability that the label of x is 1 or -1 :

$$\mathbf{P(y|x) = P(y = 1 \vee y = -1|x) = P(y = 1|x) + P(y = -1|x) = 1}$$

$$\mathbf{h_w(x) = P(y = 1|x) = 1 - P(y = -1|x)}$$

Logistic Regression: Reminder

we have:

1. $X \subset \mathbb{R}^d$ for some d and $Y = \{-1, +1\}$
2. $X \subset \mathbb{R}^d$ for some d and $Y \in \mathbb{R}$
3. $X \subset \mathbb{R}^d$ for some d and $Y \in [0,1]$

- **If $A \cap B = \emptyset \Leftrightarrow A$ and B are disjoint $\Longrightarrow P(A \cup B) = P(A) + P(B)$**
- **A and B are independent $\Leftrightarrow P(A \cap B) = P(A).P(B) \Leftrightarrow P(A|B) = P(A)$**

Logistic Regression

$X \subset \mathbb{R}^{d}$ **for some** d **and** $y \in Y = \{-1,+1\} \to y = -1\ or\ y = 1$

- **Output:**

$$\begin{array}{cccc} \varphi_{sig} o h_w & \mathbb{R}^d & \to & [\mathbf{0},\mathbf{1}] \\ & \boldsymbol{x} & \to & \varphi_{sig}(h_w(\boldsymbol{x})) = \mathbf{P}(\mathbf{y}|\mathbf{x}) \end{array}$$

- $\varphi_{sig}(h_w(\boldsymbol{x})) = \mathbf{P}(\mathbf{y} = \mathbf{1}|\mathbf{x}) \Longrightarrow \mathbf{P}(\mathbf{y} = -\mathbf{1}|\mathbf{x})$
- $\varphi_{sig}(h_w(\boldsymbol{x})) = \mathbf{P}(\mathbf{y} = -\mathbf{1}|\mathbf{x}) \Longrightarrow \mathbf{P}(\mathbf{y} = -\mathbf{1}|\mathbf{x})$
- $\mathbf{h_w}(\mathbf{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} = \boldsymbol{y}\left(\boldsymbol{y}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} > \mathbf{0}\right) \to \boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}+\mathbf{1}}$
- $\varphi_{sig}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{x}}} \in [\mathbf{0},\mathbf{1}]$

Logistic Regression

The hypothesis class for logistic regression model:

In logistic regression model, we have:

$$\boldsymbol{\varphi_{sig}(x) = \frac{1}{1 + e^{-x}}}$$

The hypothesis class of logistic regression predictors is the composition of a sigmoid function over the set of linear functions:

$$H_{sig} = \varphi \circ L_d$$

$$\boldsymbol{H_{sig}} = \{\boldsymbol{\varphi_{sig}}(\mathbf{h}_{\mathrm{w}}):\ \boldsymbol{x} \longmapsto \boldsymbol{\varphi_{sig}}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\langle \boldsymbol{w}, \boldsymbol{x} \rangle}} : \boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}+\mathbf{1}}\}$$

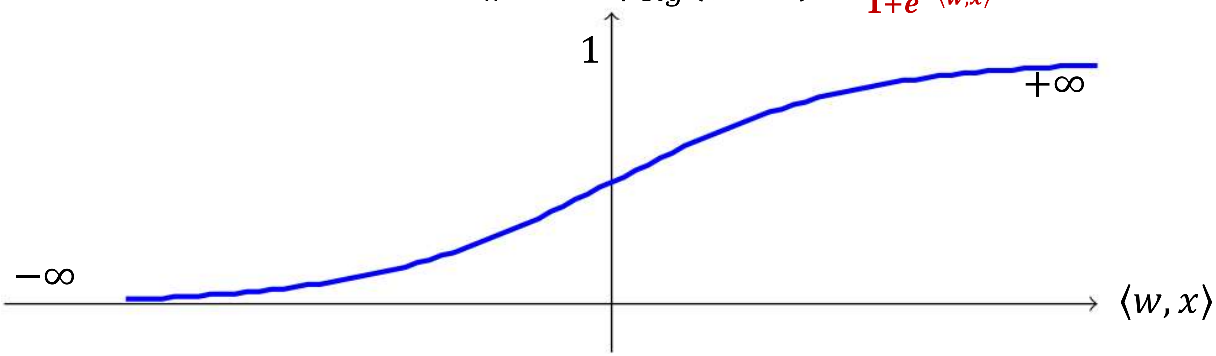
$$\left|\boldsymbol{H_{sig}}\right| = \infty$$

Logistic Regression

The name « sigmoid » means «S-shaped », referring to the plot of this function shown in the figure:

- $e^{+\infty} = +\infty, e^{-\infty} = 0 \Longrightarrow$
  - $if\, \langle w, x \rangle = +\infty \rightarrow -\langle w, x \rangle = -\infty \rightarrow \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\infty}} = \mathbf{1}$
  - $if\, \langle w, x \rangle = -\infty \rightarrow -\langle w, x \rangle = +\infty \rightarrow \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{+\infty}} = \mathbf{0}$

$$h_w(x) = \varphi_{sig}(\langle w, x \rangle) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\langle w,x \rangle}}$$



Logistic Regression

Logistic regression Vs Perceptron:

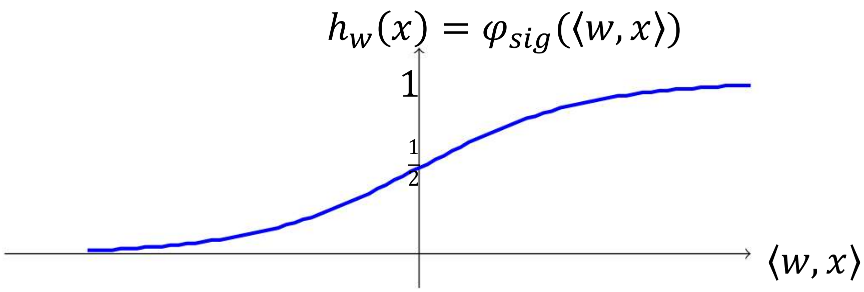
Whenever, $|\langle w, x\rangle|$ is large, the predictions of logistic regression hypothesis and perceptron hypothesis are similar.

However, whenever $|\langle w, x\rangle|$ is close to zero, we have that:

$$\varphi_{sig}(\langle w, x\rangle) \approx \frac{1}{2} \quad \text{and} \quad \varphi_p(\langle w, x\rangle) = \mathrm{sign}(\langle w, x\rangle)$$

The logistic regression hypothesis is not sure about the value of the label.

The perceptron hypothesis always outputs a deterministic prediction $\{-1, +1\}$, even if $|\langle w, x\rangle|$ is very close to zero.



Logistic Regression

The loss function for logistic regression model:

It measures how bad it is to predict some $h_w(x) \in [0,1]$ given that the true label is $y = \{\pm 1\}$.

Clearly, we want that:

$$\boldsymbol{P(y|x) = \begin{cases} h_w(x) \;\; if \;\; y = +1 \\ 1 - h_w(x) \;\; if \; y = -1 \end{cases} \Longrightarrow P(y|x) = P(y = 1|x) + P(y = -1|x) = 1}$$

to be large.

We have:

$$P(y = 1|x) = h_w(x) = \frac{1}{1+e^{-\langle w,x\rangle}} \text{ and } P(y = -1|x) = 1 - h_w(x) = \frac{1}{1+e^{\langle w,x\rangle}}$$

Generally:

$$\boldsymbol{P(y|x) = \frac{1}{1 + e^{-y\langle w,x\rangle}}}$$

Logistic Regression

It is clear that the loss function will increase monotonically if the probability $P(y|x)$ decreases.

This implies that, it will increse monotonically if $1 + e^{-y\langle w,x\rangle}$ increases.

Therefore, the loss function used in logistic regression penalizes h_w based on the log of $1 + e^{-y\langle w,x\rangle}$, that is:

$$\boldsymbol{l\big(h_w, (x, y)\big) = log\big(1 + e^{-y\langle w,x\rangle}\big)}$$

(recall that the log is a monotonic function).

Therefore, given a training set $S = (x_1, y_1), \dots, (x_m, y_m)$, the **ERM** problem associated with logistic regression is:

$$\underset{w}{\operatorname{argmin}}\, L_S(h_w) = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=1}^{m} log\big(1 + e^{-y_i\langle w,x_i\rangle}\big) \right)$$

Logistic Regression

- $\boldsymbol{w} \in \mathbb{R}^{d+1}, \ \boldsymbol{x} = (\mathbf{1}, \boldsymbol{x_1}, \dots, \boldsymbol{x_d})$
- $Ln(x)' = \frac{1}{x} \ if \ x \neq 0$

- $\boldsymbol{l(h_w, (x, y)) = log\left(1 + e^{-y\langle w,x\rangle}\right)}$
- $\boldsymbol{\nabla_w l(h_w, (x, y)) = \begin{pmatrix} \frac{\partial log\left(1+e^{-y\langle w,x\rangle}\right)}{\partial w_0} \\ \vdots \\ \frac{\partial log\left(1+e^{-y\langle w,x\rangle}\right)}{\partial w_d} \end{pmatrix} = \frac{1}{1+e^{-y\langle w,x\rangle}} \begin{pmatrix} -ye^{-y\langle w,x\rangle} \\ \vdots \\ -yx_d e^{-y\langle w,x\rangle} \end{pmatrix}}$

- $\frac{1}{m}\sum_{i=1}^{m} \nabla_w log\left(1 + e^{-y_i\langle w,x_i\rangle}\right) = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\frac{1}{1+e^{-y_i\langle w,x_i\rangle}} \begin{pmatrix} -y_i e^{-y\langle w,x_i\rangle} \\ \vdots \\ -y_i x_d e^{-y_i\langle w,x_i\rangle} \end{pmatrix}}$

- $\boldsymbol{\nabla_w^2 l(h_w, (x, y))}$

Logistic Regression

Notice:

It is clear that the loss function of the logistic regression is a convex function with respect to w.

So, the ERM_H problem for logisitic regression model can be solved using a gradient descent algorithm.