7. Machine Learning

7.1 Supervised Learning

Given (x_i, y_i), $i = 1, \dots, n$ training examples/samples

$x_i \in \mathcal{X}$: input variables, feature variables

$y_i \in \mathcal{Y}$: output " , target variables

$\{(x_i, y_i) \mid i = 1, \dots, n\}$ training set.

Supervised learning: determine a fct.

$$h : \mathcal{X} \to \mathcal{Y} \qquad \text{(a "hypothesis")}$$

so that $h(x)$ is a "good" predictor of y.

y continuous $\Rightarrow$ regression problem

y discrete $\Rightarrow$ classification problem

7.1.1 Linear regression

Training examples $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, $i = 1, \dots, n$

Assumption / hypothesis

$$y_i = \vartheta_0 + x_{i1}\vartheta_1 + x_{i2}\vartheta_2 + \dots + x_{ip}\vartheta_p + \varepsilon_i$$

ε_i : random error

$$= (1, x_i^T)\,\vartheta + \varepsilon_i$$

Hence, learn $h_\vartheta(x) = (1, x^T)\,\vartheta$, $\vartheta = (\vartheta_0, \vartheta_1, \dots, \vartheta_p)^T$ parameter

RWTH AACHEN UNIVERSITY

Set $\overset{n\times(p+1)}{X} = \begin{pmatrix} 1 & x_1^T \\ \vdots & \\ 1 & x_n^T \end{pmatrix}$, $y = (y_1, \dots, y_n)^T$

$\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^T$

Then $y = X\vartheta + \varepsilon$

Problem: Find the best ϑ by solving

$$\min_{\vartheta \in \mathbb{R}^{p+1}} \| y - X\vartheta \|$$

Solution: (i) project y onto $\mathrm{Im}(X)$: $\hat{y}$

(ii) find ϑ such that $\hat{y} = X\vartheta$

(i) $X(X^TX)^{-1}X^T$ is an orthogonal proj. onto $\mathrm{Im}(X)$.
provided $(X^TX)^{-1}$ exists.

$$\hat{y} = X(X^TX)^{-1}X^Ty = \arg\min_{z \in \mathrm{Im}(X)} \| y - z \|$$

(ii) $\hat{y} = X\vartheta \Rightarrow \cancel{X^TX}(\cancel{X^TX})^{-1}X^Ty = X^TX\vartheta$

$\Rightarrow X^Ty = X^TX\vartheta$ (normal equations)

$\Rightarrow \vartheta^* = (X^TX)^{-1}X^Ty$

and $X\vartheta^* = X(X^TX)^{-1}X^Ty = \hat{y}$

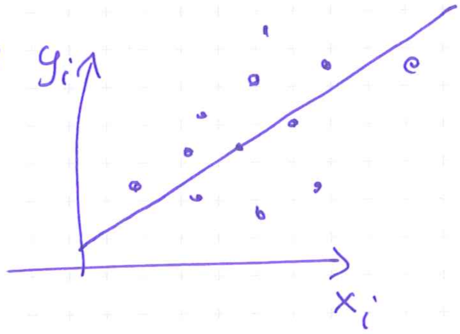
In summary: $\vartheta^* = (X^TX)^{-1}X^Ty$ is a solution.

Note: the inverse $(X^TX)^{-1}$ must exist. If not, replace $(X^TX)^{-1}$ by the so called Moore-Penrose inverse $(X^TX)^+$.

Thinking the Future
Zukunft denken

Example 7.1. (1-dim. regression)

$$y_i = \vartheta_0 + \vartheta_1 x_i + \varepsilon_i\ , \quad i = 1,\dots,14$$



Solution:

$$\vartheta_1^* = \frac{\sigma_{xy}}{\sigma_x^2} \qquad \sigma_{xy} = \frac{1}{4}\sum x_i y_i - \bar{x}\bar{y}$$

$$\sigma_x^2 = \frac{1}{4}\sum x_i^2 - \bar{x}^2$$

$$\vartheta_0^* = \bar{y} - \vartheta_1^* \bar{x} \qquad \bar{x} = \frac{1}{4}\sum x_i\ , \quad \bar{y} = \frac{1}{4}\sum y_i$$

7.1.2. Logistic Regression

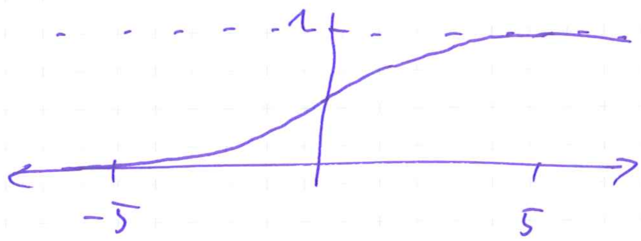
Classification approach. In the following regression with output 0, 1.

Hypothesis:

$$h_\vartheta(x) = g(\vartheta^T x) = \frac{1}{1+e^{-\vartheta^T x}} \in (0,1)$$

$g(z) = \dfrac{1}{1+e^{-z}}$ is called logistic or sigmoid fct.

Plot



It holds $g'(z) = g(z)(1-g(z))$

Given training set (x_i, y_i), $i = 1, \dots, n$, $x_i \in \mathbb{R}^p$, $y_i \in \{0,1\}$

As before set $x_{i0} = 1$, s.t.

$$\vartheta^T x_i = \vartheta_0 + \sum_{j=1}^{p} \vartheta_j x_{ij}, \quad x_i = (1, x_{i1}, \dots, x_{ip})$$

Probabilistic interpretation:

Assume: $P(y = 1 \mid x, \vartheta) = h_\vartheta(x)$

$P(y = 0 \mid x, \vartheta) = 1 - h_\vartheta(x)$

Write: $p(y \mid x, \vartheta) = (h_\vartheta(x))^y \, (1 - h_\vartheta(x))^{1-y}, \quad y \in \{0,1\}$

Assume n independent training samples

$$X = \begin{pmatrix} 1 & x_1^T \\ \vdots & \\ 1 & x_n^T \end{pmatrix} \in \mathbb{R}^{n \times (p+1)}, \quad X = (x_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le p+1}}$$

Likelihood

$$L(\vartheta) = p(y \mid X, \vartheta) = \prod_{i=1}^{n} p(y_i \mid x_i, \vartheta)$$

$$= \prod_{i=1}^{n} (h_\vartheta(x_i))^{y_i} (1 - h_\vartheta(x_i))^{1-y_i}$$

Log likelihood

$$\ell(\vartheta) = \log L(\vartheta) = \sum_{i=1}^{n} \left[y_i \log h_\vartheta(x_i) + (1-y_i) \log (1 - h_\vartheta(x_i)) \right] \quad (*)$$

Objective: $\max_{\vartheta \in \mathbb{R}^{p+1}} \ell(\vartheta)$

An algorithm herefor: gradient ascent with updates

$$\vartheta^{(k+1)} = \vartheta^{(k)} + \alpha \nabla_\vartheta \ell(\vartheta^{(k)}), \quad \alpha \text{ learning parameter}$$

Compute ∇ for each addend in (*). Set $(x_i, y_i) = (x, y)$

$$\frac{\partial}{\partial \vartheta_j} \left[y \log h_\vartheta(x) + (1-y) \log(1 - h_\vartheta(x)) \right]$$

$$= \left(y \frac{1}{g(\vartheta^T x)} - (1-y) \frac{1}{1 - g(\vartheta^T x)} \right) \frac{\partial}{\partial \vartheta_j} g(\vartheta^T x)$$

$$= (\ldots) \, g(\vartheta^T x)(1 - g(\vartheta^T x)) \frac{\partial}{\partial \vartheta_j} \vartheta^T x$$

$$= \left(y(1 - g(\vartheta^T x)) - (1-y) g(\vartheta^T x) \right) x_j$$

$$= (y - h_\vartheta(x)) x_j$$

Hence

$$\frac{\partial}{\partial \vartheta_j} \ell(\vartheta) = \sum_{i=1}^{n} (y_i - h_\vartheta(x_i)) x_{ij}$$

Update rule:

$$\vartheta_j := \vartheta_j + \alpha \sum_{i=1}^{n} (y_i - h_\vartheta(x_i)) x_{ij}$$

Alternative: Newton's method

$$\vartheta := \vartheta - H^{-1} \nabla_\vartheta \ell(\vartheta)$$

H: Hessian matrix

7.1.3 The perceptron learning algorithm

Logistic regression has values in $(0,1)$ as output. (soft decision). Now, force values to be 0 or 1 by

$$g(z) = \begin{cases} 1 & , z \geq 0 \\ 0 & , z < 0 \end{cases}$$

$$h_\vartheta(x) = g(\vartheta^T x)$$

Use the update rule, analogously to the above

$$\vartheta_j := \vartheta_j + \alpha(y_i - h_\vartheta(x_i))\, x_{ij}$$

This is the <u>perceptron learning alg.</u>

Was taken as a rough model for how neurons in the brain work.

Lacking: meaningful interpretation, no MLE.

7.2. Reinforcement Learning

By now: training sets

Now: no training set, instead: reward function which indicates if the learning is doing well or not.

Applications: robot leg coordination, autonomous flying, cell phone routing, factory control, ...

Basis for an analytical description are

7.2.1. Markov Decision Processes (MDP)

MDP is a tuple $(S, A, \{P_{sa}\}, \gamma, R)$

- S set of states
- A set of actions
- P_{sa} transition probabilities

  P_{sa} is a prob. distr. over ~~the~~ S, transition to take if the present state is s and action a is taken.
- $\gamma \in [0,1]$ discount factor
- $R: S \times A \to \mathbb{R}$ reward function, often $R: S \to \mathbb{R}$ only.

Dynamics of the process:

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \rightarrow \dots\dots$$

Total payoff $R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots$

or $R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \gamma^3 R(s_3) + \dots$

Goal: $\underset{\text{over actions in } A}{\max} E[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots]$