Linear Regression

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

Hypothesis:

$$y_i = \vartheta_0 + x_{i1}\vartheta_1 + \dots + x_{ip}\vartheta_p + \varepsilon_i, \quad i = 1, \dots, n$$

$$y = X\vartheta + \varepsilon, \quad x_{ij}, y_i \in \mathbb{R}$$

$$\longrightarrow \hat{\vartheta} = (X^T X)^{-1} X^T y$$

New observation $x \in \mathbb{R}^p$

$$\hat{y} = (1, x^T)\hat{\vartheta} \quad \text{best predictor}$$

7.1.2 Logistic Regression

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

Hypothesis

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

Set $x_{i0} = 1$, $\vartheta^T x_i = \vartheta_0 + \sum_{j=1}^{p} \vartheta_j x_{ij}$

Find the best ϑ!

Probabilistic interpretation:

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

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

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

Assume n independent training samples,

$$X = \begin{pmatrix} 1 & x_1^T \\ \vdots & \vdots \\ 1 & x_n^T \end{pmatrix} = (x_{ij})_{\substack{1 \le i \le n \\ 0 \le j \le p}}$$

Likelihood:

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

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

Log-Likelihood:

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

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

An appr. algorithm: gradient descent

$\theta^{(k+1)} = \theta^{(k)} + \alpha \nabla_\theta \ell(\theta^k)$, α learning parameter

Compute ∇ for each element in $(*)$, set $(x_i, y_i) = (x, y)$

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

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

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

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

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

Hence:

$$\frac{\partial}{\partial \vartheta_j} \ell(\vartheta) = \sum_{i=1}^{n}(y_i - h_\vartheta(x_i))x_{ij}, \quad j = 0, 1, \dots, p$$

Update rule:

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

$$j = 0, \dots, p.$$

Alternate: Newton's method

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

7.1.3 The perceptron learning algorithm

In logistic regression values in $(0,1)$. (Soft decision)

Force 0 or 1! by taking

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

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

Use the update rule

$$\theta_j^{(k+1)} = \theta_j^{(k)} + \alpha \sum_{i=1}^{n} \left(y_i - h_{\theta^{(k)}}(x_i)\right) x_{i,j}$$

the so called perceptron learning algorithm.

Lacking: meaningful prob. interpretation.

7.2. Reinforcement Learning

Basis are Markov decision processes. (MDP)

7.2.1 MDP

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

- S set of states
- A set of actions
- $P_{s,a}$ transition probabilities

  $P_{s,a}$ prob. distribution over the state space S

  Distr. of states we will transition to

  From state s taking action a.
- $\gamma \in [0,1)$ discount factor
- $R : S \times A \to \mathbb{R}$ reward function,

  often $R : S \to \mathbb{R}$

Dynamics of the process

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \to \dots$$

Total payoff:

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \gamma^3 R(s_3, a_3) + \dots$$

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

Goal:

$$\max_{\text{over actions in } A} \quad E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots\right]$$

Policy: $\pi : S \to A : s \mapsto a = \pi(s)$

Value fct:

$$V^{\pi}(s) = E\left[R(s_0) + \gamma R(s_1) + \cdots \mid s_0 = s, \pi\right]$$

$$V^{*}(s) = \max_{\pi} V^{\pi}(s)$$

It holds:

$$V^{*}(s) = R(s) + \max_{a \in A} \sum_{s' \in S} P_{sa}(s')\, V^{*}(s')$$

Optimaal policy:

$$\pi^{*}(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s')\, V^{*}(s')$$

It holds: Any opt. policy π^* is independent of the initial state $s \in S$ and achieves the same value $V^{\pi^*}(s) = V^{*}(s)$.
