Supervised learning

Hadrien Montanelli

Context

Setup Consider data $\mathcal{X} \subset \mathbb{R}^d$, labels $\mathcal{Y} \subset \mathbb{R}$ and a classifier $f : \mathcal{X} \mapsto \mathcal{Y}$. We approximate f by $\hat{f}$ using n labelled data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathcal{X}_0 \times \mathcal{Y}$, $\mathcal{X}_0 \subset \mathcal{X}$.

Bias-variance tradeoff For $y = f(\boldsymbol{x}) + \epsilon$, $\epsilon \sim \mathrm{N}(0, \sigma^2)$,

$$\mathrm{E}([y - \hat{f}(\boldsymbol{x})]^2) = [\mathrm{Bias}(\hat{f}(\boldsymbol{x}))]^2 + \mathrm{var}(\hat{f}(\boldsymbol{x})) + \sigma^2,$$
$$\mathrm{Bias}(\hat{f}(\boldsymbol{x})) = \mathrm{E}(\hat{f}(\boldsymbol{x})) - f(\boldsymbol{x}),$$
$$\mathrm{var}(\hat{f}(\boldsymbol{x})) = \mathrm{E}([\hat{f}(\boldsymbol{x}) - \mathrm{E}(\hat{f}(\boldsymbol{x}))]^2).$$

1 Naive Bayes classifier

$$\hat{f}(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \left(\mathrm{P}(Y = y) \prod_{j=1}^{d} \mathrm{P}(X_j = x_j | Y = y) \right)$$

Training $\mathcal{O}(nd)$

- Compute the class prior $\mathrm{P}(Y = y)$ from $\mathcal{X}_0$.
- Choose a model for each $\mathrm{P}(X_j = x_j | Y = y)$.
- Use the closed-form for the MLE to find the parameters for each $\mathrm{P}(X_j = x_j | Y = y)$.

Testing $\mathcal{O}(d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

2 k-NNs

$$\hat{f}(\boldsymbol{x}) = \mathrm{label}\left(\arg\min_{1 \leq i \leq n} d_{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{x}_i) \right) \quad (k = 1)$$

Training $\mathcal{O}(nd^2 s + n\log(n)d)$

- Find distance $d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j)$, with s steps of BFGS, that best separates the data.
- Choose k. Find medians to produce k-d trees.

Testing $\mathcal{O}(\log(n)d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

3 Perceptron $\mathcal{Y} = \{-1, 1\}$

$$\hat{f}(\boldsymbol{x}) = \mathrm{sign}\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right)$$

Training $\mathcal{O}(ndP)$, with $P = \max_{1 \leq i \leq n} \|x_i\|^2/\gamma^2$ (margin γ)

- Initialize $\boldsymbol{w} = \boldsymbol{0}$ and $w_0 = 0$.
- For $k = 1, 2, \ldots,$
  - if there is $\boldsymbol{x}_i \in \mathcal{X}_0$ such that
  $$\mathrm{sign}\left(\boldsymbol{w}^T \boldsymbol{x}_i + w_0\right) \neq y_i,$$
  set $\boldsymbol{w} = \boldsymbol{w} + y_i \boldsymbol{x}_i$ and $w_0 = w_0 + y_i$.

Testing $\mathcal{O}(d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

4 Kernel perceptron $\mathcal{Y} = \{-1, 1\}$

$$\hat{f}(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) \right)$$

Training $\mathcal{O}(ndP)$, with $P = \max_{1 \leq i \leq n} \|x_i\|^2/\gamma^2$ (margin γ)

- Choose K. (Perceptron is $K(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x}^T \boldsymbol{y})$.)
- Initialize $\boldsymbol{\alpha} = \boldsymbol{0}$.
- For $k = 1, 2, \ldots,$
  - set $\alpha_i = \alpha_i + 1$ if there is $\boldsymbol{x}_i \in \mathcal{X}_0$ such that
  $$\mathrm{sign}\left(\sum_{j=1}^{n} \alpha_j y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) \right) \neq y_i.$$

Testing $\mathcal{O}(nd)$ Evaluate $\hat{f}(\boldsymbol{x})$.

5 SVMs $\mathcal{Y} = \{-1, 1\}$

$$\hat{f}(\boldsymbol{x}) = \mathrm{sign}\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right)$$

Training $\mathcal{O}(nd^2 s)$ Minimize, with s steps of PGD,

$$J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2, \text{ such that } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1,\ 1 \leq i \leq n.$$

Testing $\mathcal{O}(d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

6 Logistic regression $\mathcal{Y} = \{0, 1\}$

$$\hat{f}(\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x} + w_0)}}$$

Training $\mathcal{O}(nd^3 s)$ Minimize the likelihood with s steps of Newton's method.

Testing $\mathcal{O}(d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

7 Neural networks

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \sigma(\boldsymbol{w}_i^T \boldsymbol{x} + w_{i,0}) \quad \text{(one layer)}$$

Training $\mathcal{O}(nd^2 N^2 s)$ Minimize, with s steps of SGD,

$$J(\boldsymbol{\alpha}, \boldsymbol{w}, \boldsymbol{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\hat{f}(\boldsymbol{x}_i) \neq y_i].$$

Testing $\mathcal{O}(dN)$ Evaluate $\hat{f}(\boldsymbol{x})$.

8 Decision trees

$$\hat{f}(\boldsymbol{x}) = \mathrm{label}\left(C(\boldsymbol{x})\right)$$

Training Find feature and threshold that maximize

$$u(C) - (p_L u(C_L) + p_R u(C_R)),\ u(C) = -\sum_{y\in\mathcal{Y}} p_y \log p_y.$$

Testing $\mathcal{O}(\log(n)d)$ Evaluate $\hat{f}(\boldsymbol{x})$.

9 Random forests

$$\hat{f}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{label}\left(C_t(\boldsymbol{x})\right)$$

Training Sample T times, with replacement, $m < n$ training examples from $\mathcal{X}_0$ to construct T decision trees.

Testing $\mathcal{O}(\log(n)dT)$ Evaluate $\hat{f}(\boldsymbol{x})$.