

Supervised learning

Hadrien Montanelli

Context

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

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

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

$$\text{Bias}(\hat{f}(\mathbf{x})) = E(\hat{f}(\mathbf{x})) - f(\mathbf{x}),$$

$$\text{var}(\hat{f}(\mathbf{x})) = E([\hat{f}(\mathbf{x}) - E(\hat{f}(\mathbf{x}))]^2).$$

1 Naive Bayes $\mathcal{Y} = \{0, 1\}$

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

Training $\mathcal{O}(nd)$

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

Classifying $\mathcal{O}(d)$ Evaluate $\hat{f}(\mathbf{x})$.

2 k -NNs $\mathcal{Y} = \{0, 1\}$

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

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

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

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

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

$$\hat{f}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + w_0)$$

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

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

Classifying $\mathcal{O}(d)$ Evaluate $\hat{f}(\mathbf{x})$.

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

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

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

- Choose K . (Perceptron is $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T \mathbf{y})$.)
- Initialize $\boldsymbol{\alpha} = \mathbf{0}$.
- For $k = 1, 2, \dots$,
 - set $\alpha_i = \alpha_i + 1$ if there is $\mathbf{x}_i \in \mathcal{X}_0$ such that

$$\text{sign} \left(\sum_{j=1}^n \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i) \right) \neq y_i.$$

Classifying $\mathcal{O}(nd)$ Evaluate $\hat{f}(\mathbf{x})$.

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

$$\hat{f}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + w_0)$$

Training Minimize, with s steps of PGD,

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2,$$

such that $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1$, $1 \leq i \leq n$.

Classifying $\mathcal{O}(d)$ Evaluate $\hat{f}(\mathbf{x})$.

6 Kernel SVMs $\mathcal{Y} = \{-1, 1\}$

$$\hat{f}(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0 \right)$$

Training

- Choose K . (SVMs is $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$.)
- Maximize, with s steps of SQP,

$$J(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_j, \mathbf{x}_i),$$

such that $\sum_{i=1}^n \alpha_i y_i = 0$ and $\alpha_i \geq 0$, $1 \leq i \leq n$.

- For i with $\alpha_i \neq 0$, $w_0 = y_i - \sum_{j=1}^n \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_i)$.

Classifying $\mathcal{O}(nd)$ Evaluate $\hat{f}(\mathbf{x})$.

7 Logistic regression $\mathcal{Y} = [0, 1]$

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

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

Classifying $\mathcal{O}(d)$ Evaluate $\hat{f}(\mathbf{x})$.

8 Neural networks $\mathcal{Y} = [0, 1]$

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

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

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

Classifying $\mathcal{O}(dN)$ Evaluate $\hat{f}(\mathbf{x})$.

9 Decision trees

$$\hat{f}(\mathbf{x}) = \text{label}(C(\mathbf{x}))$$

Training Find feature and threshold that maximize

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

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

10 Random forests

$$\hat{f}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T \text{label}(C_t(\mathbf{x}))$$

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

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