Theory of Statistical Learning
Part II

Damien Garreau

Université Côte d'Azur

2022

Outline

1. Linear predictors
   - Linear classification
   - Linear regression
   - Ridge regression
   - Polynomial regression
   - Logistic regression
2. Tree-based classifiers
   - Partition rules
   - Random forests
3. Boosting
   - Adaboost
   - XGBoost
4. Nearest neighbors

1. Linear predictors

1.1. Linear classification

Linear functions

- $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$
- thus $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})^\top$
- we consider no bias term (otherwise *affine*):

$$\left\{h : x \mapsto w^\top x, w \in \mathbb{R}^d\right\}.$$

- **Reminder:** given two vectors $u, v \in \mathbb{R}^d$,

$$\langle u, v \rangle = u^\top v = \sum_{j=1}^{d} u_i v_i \,.$$

- binary classification: 0-1 loss, $\mathcal{Y} = \{-1, +1\}$
- **Important:** compose h with $\phi : \mathbb{R} \to \mathcal{Y}$ (typically the sign)

The sign function

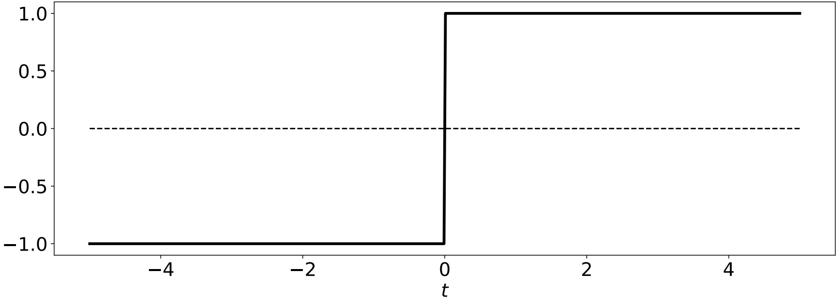


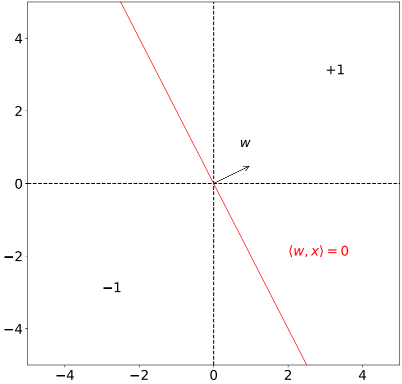
Figure: the sign function

Halfspaces

- thus our function class is

$$\mathcal{H} = \left\{x \mapsto \operatorname{sign}\left(w^\top x\right), w \in \mathbb{R}^d\right\}.$$

- gives label $+1$ to vector pointing in the same direction as w



VC dimension of halfspaces

Proposition: the VC dimension of halfspaces in dimension d is exactly $d+1$.

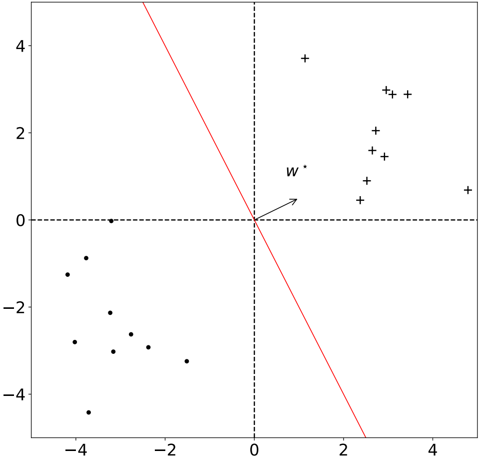
- **Consequence:** $\mathcal{H}$ is PAC learnable with sample complexity

$$\Omega\left(\frac{d+\log(1/\delta)}{\varepsilon}\right) .$$

Linearly separable data

- **Important assumption:** data is linearly separable
- that is, there is a $w^\star \in \mathbb{R}^d$ such that

$$y_i = \operatorname{sign}\left(\langle w^\star, x_i \rangle\right) \quad \forall 1 \leq i \leq n\,.$$



Linear programming

- **Empirical risk minimization:** recall that we are looking for w such that
$$\hat{\mathcal{R}}_S(w) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{y_i \neq \mathrm{sign}(w^\top x_i)}$$
is minimal
- **Question:** how to solve this?
- we want $y_i = \mathrm{sign}\left(w^\top x_i\right)$ for all $1 \leq i \leq n$
- equivalent formulation: $y_i\langle w, x_i\rangle > 0$
- we know that there is a vector that satisfies this condition ($w^\star$)
- let us set $\gamma = \min_i\{y_i\langle w^\star, x_i\rangle\}$ and $\overline{w} = w^\star/\gamma$
- we have shown that there is a vector such that $y_i\langle \overline{w}, x_i\rangle \geq 1$ for any $1 \leq i \leq n$ (and it is an ERM)

Linear programming, ctd.

- define the matrix $A \in \mathbb{R}^{n \times d}$ such that

$$A_{i,j} = y_i x_{i,j} \, .$$

- **Intuition:** observations $\times$ labels
- remember that we have the ± 1 label convention
- define $v = (1, \dots, 1)^\top \in \mathbb{R}^n$
- then we can rewrite the above problem as

$$\text{maximize} \quad \langle u, w \rangle \text{ subject to } Aw \leq v \, .$$

- we call this sort of problems **linear programs**[1]
- solvers readily available, e.g., `scipy.optimize.linprog` if you use Python

[1] Boyd, Vandenberghe, *Convex optimization*, Cambridge University Press, 2004

The perceptron

- another possibility: the *perceptron*[2]
- **Idea:** iterative algorithm that constructs $w^{(1)}, w^{(2)}, \ldots, w^{(T)}$
- update rule: at each step, find i that is misclassified and set

$$w^{(t+1)} = w^{(t)} + y_i x_i \,.$$

- **Question:** why does it work?
- pushes w in the right direction:

$$y_i \langle w^{(t+1)}, x_i \rangle = y_i \langle w^{(t)} + y_i x_i, x_i \rangle = y_i \langle w^{(t)}, x_i \rangle + \|x_i\|^2$$

- remember, we want $y_i \langle w, x_i \rangle > 0$ for all i

[2] Rosenblatt, *The perceptron, a perceiving and recognizing automaton*, tech report, 1957

1.2. Linear regression

Least squares

- regression $\Rightarrow$ squared-loss function

$$\ell(y, y') = (y - y')^2 \,.$$

- still looking at linear functions:

$$\mathcal{H} = \{h : x \mapsto \langle w, x \rangle \text{ s.t. } w \in \mathbb{R}^d\} \,.$$

- empirical risk in this context:

$$\hat{\mathcal{R}}_S(h) = \frac{1}{n} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 = F(w) \,.$$

- also called **mean squared error**
- empirical risk minimization: we want to minimize $w \mapsto F(w)$ with respect to $w \in \mathbb{R}^d$
- F is a convex, smooth function

Least squares, ctd.

- let us compute the gradient of F:

$$\begin{aligned}
\frac{\partial F}{\partial w_j}(w) &= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial w_j}(w^\top x_i - y_i)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot \frac{\partial}{\partial w_j}\left(w^\top x_i - y_i\right) \cdot (w^\top x_i - y_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot \frac{\partial}{\partial w_j}\left(\cdots + w_j x_{i,j} + \cdots - y_i\right) \cdot (w^\top x_i - y_i) \\
\frac{\partial F}{\partial w_j}(w) &= \frac{2}{n}\sum_{i=1}^{n} x_{i,j} \cdot (w^\top x_i - y_i)\,.
\end{aligned}$$

Least squares, ctd.

- it is more convenient to write $\nabla F(w) = 0$ in matrix notation
- define $X \in \mathbb{R}^{n\times d}$ the matrix such that line i of X is observation x_i
- one can check that, for any $1 \leq j, k \leq d$,

$$\left(X^\top X\right)_{j,k} = \sum_{i=1}^{n} x_{i,j} x_{i,k} \,.$$

- thus

$$\begin{aligned}
\left(X^\top X w\right)_j &= \sum_{k=1}^{d} \left(X^\top X\right)_{j,k} w_k \\
&= \sum_{k=1}^{d} \sum_{i=1}^{n} x_{i,j} x_{i,k} w_k \\
&= \sum_{i=1}^{n} x_{i,j} w^\top x_i \,.
\end{aligned}$$

Least squares, ctd.

- thus, if we define

$$A = X^\top X = \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{d \times d} \text{ and } b = X^\top y = \sum_{i=1}^{n} y_i x_i \in \mathbb{R}^d\,,$$

  solving $\nabla F(w) = 0$ is equivalent to solving

$$Aw = b\,.$$

- if A is invertible, straightforward:

$$\boxed{\hat{w} = A^{-1} b}$$

- computational cost: $\mathcal{O}\left(d^3\right)$ (inversion is actually a bit less)
- what happens when A is not invertible?

Singular value decomposition

- since A is symmetric, it has an eigendecomposition

$$A = VDV^\top \,,$$

  with $D \in \mathbb{R}^d$ diagonal and V orthonormal
- define D^+ such that

$$D^+_{i,i} = 0 \text{ if } D_{i,i} = 0 \text{ and } D^+_{i,i} = \frac{1}{D_{i,i}} \text{ otherwise.}$$

- define $A^+ = VD^+V^\top$
- then we set

$$\boxed{\hat{w} = A^+ b \,.}$$

Singular value decomposition, ctd.

- why did we do that?
- let v_i denote the ith column of V, then

$$
\begin{aligned}
A\hat{w} &= AA^+ b && \text{(definition of } \hat{w}\text{)} \\
&= VDV^\top VD^+ V^\top b && \text{(definition of } A^+\text{)} \\
&= VDD^+ V^\top b && \text{(} V \text{ is orthonormal)} \\
A\hat{w} &= \sum_{i : D_{i,i} \neq 0} v_i v_i^\top b \,.
\end{aligned}
$$

- in definitive, $A\hat{w}$ is the projection of b onto the span of v_i such that $D_{i,i} \neq 0$
- since the span of these v_i is the span of the x_i and b is in the linear span of the x_i, we have $A\hat{w} = b$
- cost of SVD: $\mathcal{O}\left(dn^2\right)$ if $d > n$ (SVD of X)

Exercise

Exercise: Of course, one does not have to use the squared loss. Instead, we may prefer to use

$$\ell(y, y') = |y - y'| \, .$$

1. show that, for any $c \in \mathbb{R}$,

$$|c| = \min_{a \geq 0} a \quad \text{subject to} \quad a \geq c \text{ and } a \geq -c \, .$$

2. use the previous question to show that ERM with the absolute value loss function is equivalent to minimizing the linear function $\sum_{i=1}^{n} s_i$, where the s_i satisfy linear constraints
3. write it in matrix form, that is, find $A \in \mathbb{R}^{2n \times (n+d)}$, $v \in \mathbb{R}^{d+n}$, and $b \in \mathbb{R}^{2n}$ such that the LP can be written

$$\text{minimize } c^\top v \quad \text{subject to} \quad Av \leq b \, .$$