Machine Learning — Statistical Methods for Machine Learning

Linear classification

Instructor: *Nicolò Cesa-Bianchi* version of April 20, 2019

A linear classifier for $\mathcal{X} = \mathbb{R}^d$ is a function $h : \mathbb{R}^d \to \{-1, +1\}$ such that $h(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x} - c)$ for some $\boldsymbol{w} \in \mathbb{R}^d$ and $c \in \mathbb{R}$. Geometrically, a linear classifier defines an hyperplane $\Gamma \subset \mathbb{R}^d$ such that, if Γ^+, Γ^- are the two halfspaces on each side of the hyperplane, we have

$$h(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \boldsymbol{x} \in \Gamma^+ \\ -1 & \text{if } \boldsymbol{x} \in \Gamma^- \end{cases}$$

Recall that an hyperplane with coefficients $\boldsymbol{w}, c$ is defined by $\left\{\boldsymbol{x} \in \mathbb{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x} = c\right\}$, where $\boldsymbol{w}^\top \boldsymbol{x} = \|\boldsymbol{w}\| \, \|\boldsymbol{x}\| \cos\theta$ and θ is the angle between $\boldsymbol{w}$ and $\boldsymbol{x}$. Also, $\|\boldsymbol{x}\| \cos\theta$ is the length of the projection of $\boldsymbol{x}$ onto $\boldsymbol{w}$ and, symmetrically, $\|\boldsymbol{w}\| \cos\theta$ is the length of the projection of $\boldsymbol{w}$ onto $\boldsymbol{x}$. Hence, the hyperplane $\left\{\boldsymbol{x} \in \mathbb{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x} = c\right\}$ is orthogonal to $\boldsymbol{w}$ and intersects it at distance $c/\|\boldsymbol{w}\|$ from the origin.

The halfspaces Γ^+ e Γ^- defined by the hyperplane $\left\{\boldsymbol{x} \in \mathbb{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x} = c\right\}$ are

$$\Gamma^+ \equiv \left\{\boldsymbol{x} \,:\, \boldsymbol{w}^\top \boldsymbol{x} > c\right\} \qquad \text{and} \qquad \Gamma^- \equiv \left\{\boldsymbol{x}' \,:\, \boldsymbol{w}^\top \boldsymbol{x}' \leq c\right\}$$

That is, all points $\boldsymbol{x}$ whose projection onto $\boldsymbol{w}$ has length strictly bigger than $c/\|\boldsymbol{w}\|$, and all points $\boldsymbol{x}'$ whose projection onto $\boldsymbol{w}$ has length not larger than $c/\|\boldsymbol{w}\|$.

Hyperplanes of the form $\left\{\boldsymbol{x} \in \mathbb{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x} = 0\right\}$ pass through the origin and are called *homogeneous*. Any non-homogeneous hyperplane $\left\{\boldsymbol{x} \in \mathbb{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x} = c\right\}$, with $c \neq 0$, is equivalent to the homogeneous hyperplane $\left\{\boldsymbol{x} \in \mathbb{R}^{d+1} \,:\, \boldsymbol{v}^\top \boldsymbol{x} = 0\right\}$ with $\boldsymbol{v} = (w_1, \ldots, w_d, -c)$ and when the points $\boldsymbol{x} \in \mathbb{R}^d$ are mapped to the points $\boldsymbol{x}' = (x_1, \ldots, x_d, 1) \in \mathbb{R}^{d+1}$. Indeed, $\mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x} - c) = \mathrm{sgn}(\boldsymbol{v}^\top \boldsymbol{x}')$. For this reason, without any loss of generality we will only deal with algorithms that learn linear classifiers corresponding to homogeneous hyperplanes. This amounts to saying that we automatically add an extra feature with value 1 to all of our data points.

Training linear classifiers. Let $\mathcal{H}_d$ be the family of linear classifiers $h(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x})$ for $\boldsymbol{w} \in \mathbb{R}^d$. Consider the ERM algorithm for zero-one loss that, given a training set $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_m, y_m) \in \mathbb{R}^d \times \{-1, +1\}$, outputs

$$\widehat{h} = \operatorname*{argmin}_{h \in \mathcal{H}_d} \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}\{h(\boldsymbol{x}_t) \neq y_t\} \,. \tag{1}$$

Unfortunately, it is unlikely to find an efficient implementation of ERM for linear classifiers with zero-one loss. In fact, the decision problem associated with finding $\widehat{h}$ is NP-complete even when $\boldsymbol{x}_t \in \{0, 1\}^d$ for $t = 1, \ldots, m$. More precisely, introduce the following decision problem.

MinDisagreement
Instance: Pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m) \in \{0, 1\}^d \times \{-1, +1\}$. Integer k.
Question: Is there $\boldsymbol{w} \in \mathbb{Q}^d$ such that $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$ for at most k indices $t = 1, \ldots, m$?

The following result can be shown.

Theorem 1. *MinDisagreement is NP-complete.*

In addition to that, the following stronger hardness-of-approximation result can be also shown.

MinDisOpt
Instance: Pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m) \in \{0,1\}^d \times \{-1,+1\}$.
Solution: A point $\boldsymbol{w} \in \mathbb{Q}^d$ minimizing the number of indices $t = 1, \ldots, m$ such that $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$.

Given an instance S (i.e., a training set) of MinDisOpt, let $\mathrm{Opt}(S)$ the number of examples in S that are misclassified by the best possible hyperplane (i.e., the one minimizing misclassifications).

Theorem 2. *If $P \neq NP$, then for all $C > 0$ there are no polynomial time algorithms that approximately solve every instance S of MinDisOpt with a number of misclassified examples bounded by $C \times \mathrm{Opt}(S)$.*

This implies that, unless $P = NP$ (which is believed unlikely), there are no efficient algorithms that approximate the solution of (1) to within a constant factor. Here efficient means in time polynomial in the input size md.

The ERM problem (1) becomes easier when the training set is **linearly separable**. A training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ is linearly separable where there exists a linear classifier with zero training error. In other words, there exists a separating hyperplane $\boldsymbol{u} \in \mathbb{R}^d$ such that

$$\gamma(\boldsymbol{u}) \stackrel{\text{def}}{=} \min_{t=1,\ldots,m} y_t \, \boldsymbol{u}^\top \boldsymbol{x}_t > 0$$

The quantity $\gamma(\boldsymbol{u})$ is known as the **margin** of $\boldsymbol{u}$ on the training set. The scaled margin $\gamma(\boldsymbol{u}) / \|\boldsymbol{u}\|$ measures the distance between the separating hyperplane and the closest training example. Since $\gamma(\boldsymbol{u}) > 0$ can be multiplied by any positive constant by rescaling $\boldsymbol{u}$, we assume any separating hyperplane $\boldsymbol{u}$ always satisfies $\gamma(\boldsymbol{u}) \geq 1$.

Now observe that the ERM problem (1) can be expressed as a system of linear inequalities,

$$y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t > 0 \qquad t = 1, \ldots, m \ .$$

When the training set is linearly separable, the system has at least a solution. This solution can be found in polynomial time using a linear solver.

We now introduce a very simple algorithm for learning linear classifiers that can be used to solve the ERM problem in the linearly separable case. The Perceptron algorithm finds a homogeneous separating hyperplane by runnning through the training examples one after the other. The current linear classifier is tested on each training example and, in case of misclassification, the associated hyperplane is adjusted.

Algorithm: Perceptron
Input: Training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$.
Initialization $\boldsymbol{w} = (0, \ldots, 0)$.
Repeat
Read next training example $(\boldsymbol{x}_t, y_t)$
If $y_t\, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$, **then** $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_t\, \boldsymbol{x}_t$
Until $\gamma(\boldsymbol{w}) > 0$
Output $\boldsymbol{w}$

Note that if the algorithm terminates, then $\boldsymbol{w}$ is a separating hyperplane. The update $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_t\, \boldsymbol{x}_t$ when $y_t\, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$ makes $y_t\, \boldsymbol{w}^\top \boldsymbol{x}_t$ bigger. Indeed,

$$y_t\big(\boldsymbol{w} + y_t\, \boldsymbol{x}_t\big)^\top \boldsymbol{x}_t = y_t\, \boldsymbol{w}^\top \boldsymbol{x}_t + \|\boldsymbol{x}_t\|^2 > y_t\, \boldsymbol{w}^\top \boldsymbol{x}_t$$

Geometrically, each update moves $\boldsymbol{w}$ towards $\boldsymbol{x}_t$ if $y_t = 1$ and moves $\boldsymbol{w}$ away from $\boldsymbol{x}_t$ if $y_t = -1$.

We now prove that Perceptron always terminates on linearly separable training sets.

Theorem 3 (Convergence of Perceptron)**.** *Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ be a linearly separable training set. Then the Perceptron algorithm terminates after a number of updates not bigger than*

$$\left(\min_{\boldsymbol{u}\,:\,\gamma(\boldsymbol{u})=1} \|\boldsymbol{u}\|^2\right)\left(\max_{t=1,\ldots,m} \|\boldsymbol{x}_t\|^2\right) \tag{2}$$

Proof. Let $\boldsymbol{w}_0 = (0, \ldots, 0)$ be the initial hyperplane. Let $\boldsymbol{w}_M$ be the hyperplane after M updates and let $t_M \in \{1, \ldots, m\}$ be the index of the training example $(\boldsymbol{x}_{t_M}, y_{t_M})$ that caused the M-th update $\boldsymbol{w}_M = \boldsymbol{w}_{M-1} + y_{t_M} \boldsymbol{x}_{t_M}$. We prove an upper bound on M by deriving upper and lower bounds on $\|\boldsymbol{w}_M\|\, \|\boldsymbol{u}\|$. We start by observing that

$$\|\boldsymbol{w}_M\|^2 = \|\boldsymbol{w}_{M-1} + y_{t_M} \boldsymbol{x}_{t_M}\|^2 = \|\boldsymbol{w}_{M-1}\|^2 + \|\boldsymbol{x}_{t_M}\|^2 + 2\, y_{t_M} \boldsymbol{w}_{M-1}^\top \boldsymbol{x}_{t_M} \leq \|\boldsymbol{w}_{M-1}\|^2 + \|\boldsymbol{x}_{t_M}\|^2$$

because $y_{t_M} \boldsymbol{w}_{M-1}^\top \boldsymbol{x}_{t_M} \leq 0$ due to the update $\boldsymbol{w}_M = \boldsymbol{w}_{M-1} + y_{t_M} \boldsymbol{x}_{t_M}$. Iterating this argument M times, and recalling that $\boldsymbol{w}_0 = (0, \ldots, 0)$, we obtain

$$\|\boldsymbol{w}_M\|^2 \leq \|\boldsymbol{w}_0\|^2 + \sum_{i=1}^{M} \|\boldsymbol{x}_{t_i}\|^2 \leq M \left(\max_{t=1,\ldots,m} \|\boldsymbol{x}_t\|^2\right) .$$

Hence

$$\|\boldsymbol{w}_M\|\, \|\boldsymbol{u}\| \leq \|\boldsymbol{u}\| \left(\max_{t=1,\ldots,m} \|\boldsymbol{x}_t\|\right) \sqrt{M} .$$

To prove the lower bound, fix any separating hyperplane $\boldsymbol{u}$ with $\gamma(\boldsymbol{u}) \geq 1$ and let θ be the angle between $\boldsymbol{u}$ and $\boldsymbol{w}_M$. We have

$$\begin{aligned}
\|\boldsymbol{w}_M\|\, \|\boldsymbol{u}\| &\geq \|\boldsymbol{w}_M\|\, \|\boldsymbol{u}\|\, \cos(\theta) && \text{(since } -1 \leq \cos(\theta) \leq 1\text{)} \\
&= \boldsymbol{w}_M^\top \boldsymbol{u} && \text{(by definition of inner product } \boldsymbol{w}_T^\top \boldsymbol{u}\text{)} \\
&= \left(\boldsymbol{w}_{M-1} + y_{t_M} \boldsymbol{x}_{t_M}\right)^\top \boldsymbol{u} \\
&= \boldsymbol{w}_{M-1}^\top \boldsymbol{u} + y_{t_M} \boldsymbol{u}^\top \boldsymbol{x}_{t_M} \\
&\geq \boldsymbol{w}_{M-1}^\top \boldsymbol{u} + 1
\end{aligned}$$

where the last inequality holds because $1 \leq \gamma(\boldsymbol{u}) \leq y_t \boldsymbol{u}^\top \boldsymbol{x}_t$ for all $t = 1, \ldots, m$. Iterating M times we get

$$\|\boldsymbol{w}_M\| \, \|\boldsymbol{u}\| \geq \boldsymbol{w}_0^\top \boldsymbol{u} + M = M$$

Where we used $\boldsymbol{w}_0^\top \boldsymbol{u} = 0$ since $\boldsymbol{w}_0 = (0, \ldots, 0)$. Combining upper and lower bound we obtain

$$M \leq \|\boldsymbol{u}\| \left(\max_{t=1,\ldots,M} \|\boldsymbol{x}_t\| \right) \sqrt{M} \ .$$

Solving for M, and recalling the choice of $\boldsymbol{u}$, we obtain (2). Hence, the update count M cannot grow larger than (2). Since the algorithm stops when no more updates are possible, we conclude that the Perceptron terminates after a bounded number of updates. □

Note that the Perceptron convergence theorem does not imply that the Perceptron algorithm terminates in polynomial time on any linearly separable training set. Although each update takes constant time, the number of updates can still be exponential in m or d.