Machine Learning

Linear Models

Fabio Vandin

October 10, 2017
Consider $\mathcal{X} = \mathbb{R}^d$

**Affine functions:**

$$L_d = \{ h_{w,b} : w \in \mathbb{R}^d, b \in \mathbb{R} \}$$

where

$$h_{w,b}(x) = \langle w, x \rangle + b = \left( \sum_{i=1}^{d} w_i x_i \right) + b$$

**Note:**

- each member of $L_d$ is a function $x \mapsto \langle w, x \rangle + b$
- $b$: bias
Hypothesis class $\mathcal{H}: \phi \circ L_d$, where $\phi: \mathbb{R} \to \mathcal{Y}$
- $h \in \mathcal{H}$ is $h: \mathbb{R}^d \to \mathcal{Y}$

$\phi$ depends on the learning problem

Example
- binary classification, $\mathcal{Y} = \{-1, 1\} \Rightarrow \phi(z) = \text{sign}(z)$
- regression, $\mathcal{Y} = \mathbb{R} \Rightarrow \phi(z) = z$
Equivalent Notation

Given $\mathbf{x} \in \mathcal{X}$, $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, define:

- $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

Then:

$$h_{\mathbf{w}, b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle \quad (1)$$

$\Rightarrow$ we will consider bias term as part of $\mathbf{w}$ and assume $\mathbf{x} = (1, x_1, x_2, \ldots, x_d)$ when needed, with $h_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$
Linear Classification

$\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$, 0-1 loss

Hypothesis class = \textit{halfspaces}

$$HS_d = \text{sign} \circ L_d = \{ x \mapsto \text{sign}(h_{x,b}(x)) : h_{w,b} \in L_d \} \quad (2)$$

Example: $\mathcal{X} = \mathbb{R}^2$
Finding a Good Hypothesis

Linear classification with hypothesis set $\mathcal{H} = \text{halfspaces}$.

How do we find a good hypothesis?

Good $= \text{ERM rule}$

$\Rightarrow$ Perceptron Algorithm (Rosenblatt, 1958)

**Note:**
if $y_i \langle w, x_i \rangle > 0$ for all $i = 1, \ldots, m \Rightarrow$ all points are classified correctly by model $w$
Perceptron

**Input:** training set \((x_1, y_1), \ldots, (x_m, y_m)\)

initialize \(w^{(1)} = (0, \ldots, 0)\);

for \(t = 1, 2, \ldots\) do

\[
\text{if } \exists i \text{ s.t. } y_i \langle w^{(t)}, x_i \rangle \leq 0 \text{ then } w^{(t+1)} \leftarrow w^{(t)} + y_i x_i;
\]

else return \(w^{(t)}\);

Interpretation of update:

Note that:

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

\(\Rightarrow\) update guides \(w\) to be “more correct” on \((x_i, y_i)\).

Termination? Depends on the realizability assumption!
Perceptron with Linearly Separable Data

Realizability assumption for halfspace = “linearly separable data”

Proposition

Assume that \((x_1, y_1), \ldots, (x_m, y_m)\) is linearly separable, let:

- \(B = \min \{ ||w|| : y_i \langle w, x_i \rangle \geq 1 \ \forall i, i = 1, \ldots, m \}\), and
- \(R = \max_i ||x_i||\).

Then the Perceptron algorithm stops after at most \((RB)^2\) iterations and when it stops it holds that \(\forall i, i \in \{1, \ldots, m\} : y_i \langle w^{(t)}, x_i \rangle > 0\).
Perceptron: Notes

• simple to implement

• for separable data
  • convergence is guaranteed
  • converge depends on $B$, which can be exponential in $d$... \( \Rightarrow \) an ILP approach may be better to find ERM solution in such cases
  • potentially multiple solutions, which one is picked depends on starting values

• non separable data?
  • run for some time and keep best solution found up to that point (pocket algorithm)
Linear Regression

\( \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \mathbb{R} \)

Hypothesis class:

\[ \mathcal{H}_{\text{reg}} = L_d = \{ \mathbf{x} \rightarrow \langle \mathbf{w}, \mathbf{x} \rangle + b : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R} \} \]

**Note:** \( h \in \mathcal{H}_{\text{reg}} : \mathbb{R}^d \rightarrow \mathbb{R} \)

Commonly used loss function: *squared-loss*

\[ \ell(h, (\mathbf{x}, y)) \overset{\text{def}}{=} (h(\mathbf{x}) - y)^2 \]

⇒ empirical risk function (training error): *Mean Squared Error*

\[ L_S(h) = \frac{1}{m} \sum_{i=1}^{m} (h(\mathbf{x}_i) - y_i)^2 \]
Linear Regression - Example

\[ d = 1 \]
How to find a ERM hypothesis? \textit{Least Squares} algorithm

Best hypothesis:

$$\arg\min_w L_S(h_w) = \arg\min_w \frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$

Equivalent formulation: \textbf{w} minimizing \textit{Residual Sum of Squares} (RSS), i.e.

$$\arg\min_w \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2$$
Let

\[ X = \begin{bmatrix}
\cdots & x_1 & \cdots \\
\cdots & x_2 & \cdots \\
\cdots & \cdots & \cdots \\
\cdots & x_m & \cdots
\end{bmatrix} \]

\[ X: \text{design matrix} \]

\[ y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \]

⇒ we have that RSS is

\[ \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 = (y - Xw)^T (y - Xw) \]
Want to find \( \mathbf{w} \) that minimizes RSS (=objective function):

\[
RSS(\mathbf{w}) = \arg \min_{\mathbf{w}} (\mathbf{y} - \mathbf{Xw})^T (\mathbf{y} - \mathbf{Xw})
\]

How?

Compute gradient \( \frac{\partial RSS(\mathbf{w})}{\partial \mathbf{w}} \) of objective function w.r.t \( \mathbf{w} \) and compare it to 0.

\[
\frac{\partial RSS(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{Xw})
\]

Then we need to find \( \mathbf{w} \) such that

\[
-2\mathbf{X}^T(\mathbf{y} - \mathbf{Xw}) = 0
\]
\[-2X^T(y - Xw) = 0\]

is equivalent to

\[X^TXw = X^Ty\]

If $X^TX$ is invertible $\Rightarrow$ solution to ERM problem is:

\[w = (X^TX)^{-1}X^Ty\]
Complexity Considerations

We need to compute

$$(X^T X)^{-1} X^T y$$

Algorithm:

1. compute $X^T X$: product of $(d + 1) \times m$ matrix and $m \times (d + 1)$ matrix
2. compute $(X^T X)^{-1}$ inversion of $(d + 1) \times (d + 1)$ matrix
3. compute $(X^T X)^{-1} X^T$: product of $(d + 1) \times (d + 1)$ matrix and $(d + 1) \times m$ matrix
4. compute $(X^T X)^{-1} X^T$: product of $(d + 1) \times m$ matrix and $m \times 1$ matrix

Most expensive operation? Inversion!

$\Rightarrow$ done for $(d + 1) \times (d + 1)$ matrix
\( X^T X \) not invertible?

How do we get \( w \) such that

\[
X^T X w = X^T y
\]

if \( X^T X \) is not invertible?

Let

\[
A = X^T X
\]

Let \( A^+ \) be the \textit{generalized inverse} of \( A \), i.e.:

\[
AA^+ A = A
\]

**Proposition**

If \( A = X^T X \) is not invertible, then \( \hat{w} = A^+ X^T y \) is a solution to \( X^T X w = X^T y \).

**Proof:** [UML] 9.2.1
Generalized Inverse of $A$

Note $A = X^T X$ is symmetric $\Rightarrow$ eigenvalue decomposition of $A$:

$$A = VDV^T$$

with
- $D$: diagonal matrix (entries = eigenvalues of $A$)
- $V$: orthonormal matrix ($V^T V = I_{d \times d}$)

Define $D^+$ diagonal matrix such that:

$$D^+_{i,i} = \begin{cases} 
0 & \text{if } D_{i,i} = 0 \\
\frac{1}{D_{i,i}} & \text{otherwise}
\end{cases}$$
Let $A^+ = VD^+V^T$

Then

$$AA^+A = VDV^TVD^+V^TVDV^T$$
$$= VDD^+DV^T$$
$$= VDV^T$$
$$= A$$

$\Rightarrow$ $A^+$ is the generalized inverse of $A$. 
Logistic Regression

Learn a function $h$ from $\mathbb{R}^d$ to $[0, 1]$.

What can this be used for?

Classification!

**Example:** binary classification $(\mathcal{Y} = \{-1, 1\})$ - $h(x) =$ probability that label of $x$ is 1.

For simplicity of presentation, we consider binary classification with $\mathcal{Y} = \{-1, 1\}$, but similar considerations apply for multiclass classification.