Machine Learning - Lecture 2

- Review
- ERH principle
- Examples to ERH principle
- The Perceptron algorithm and its convergence
- Learning with ERH in general.

**Review**

- \( X \) - set of instances or objects. Eg. \( X = \mathbb{R}^d \) vectors in \( \mathbb{R}^d \)
- \( Y \) - set of possible labels. Eg. \( Y = \{ -1, +1 \} \) or \( Y = \{ 0, 1, 3 \} \) binary classification
- \( Y = \{ 0, 1, 2, \ldots, k-1 \} \) multiclassification
- \( Y = \mathbb{R} \) regression

- \( H \) - set of possible functions / hypothesis that participate learning
- \( l(y, \hat{y}) \) - task loss function, Measures how \( y \) diverges from \( \hat{y} \)

  - \( l(y, \hat{y}) = \| y - \hat{y} \|_2 \)

- Training:
  \[ \alpha^* = \arg \min_{\theta} \mathbb{E}_{(x, y) \sim D} [l(y, f_\theta(x))]. \]

  Since we don't know \( D \) we use a sample from it:

  \[ S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \text{ where } (x_i, y_i) \sim D \]

  and

  \[ \alpha^* = \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^{m} l(y_i, f_\theta(x_i)) \]

- Inference:
  \[ \hat{y} = f_{\alpha^*}(x) \]
**ERM Principle**

**ERM - Empirical Risk Minimization**

Training risk or training error is defined as

\[
\frac{1}{m} \sum_{i=1}^{m} l(y_i, f_\theta(x_i))
\]

The learning paradigm that resulted with a predictor for that minimizes the training error is called ERM. For example

\[
\begin{array}{c}
  \times \\
  \times \\
  \times \\
  \times \\
  \times \\
  \times \\
\end{array}
\]

If the set of \( x \) are all squares, the hypothesis found

Another example \( x \in \mathbb{R}^d \), \( Y = \{0, 1\} \)

\[
f_\theta(x) = \begin{cases} 
  y_i & \text{if } \exists i \in [1,m] \text{ such that } x = x_i \\
  0 & \text{otherwise}
\end{cases}
\]

In words, the hypothesis (function) is equals the label \( y_i \) if the object \( x \) is one of the objects \( x_i \) in the training set.
Linear Precisors

The set of linear (affine) functions:

\[ f_{w,b}(x) = w \cdot x + b = \sum_{i=1}^{d} w_i x_i + b \quad x \in \mathbb{R}^d, \quad w \in \mathbb{R}^d, \quad b \in \mathbb{R} \]

\[ H = \{ f : \mathbb{R}^d \to \mathbb{R} | \mathbb{R} \} \]

The prediction function is:

\[ \hat{y} = \text{sign}(w \cdot x + b) \quad y \in \{ -1, 1 \} \]

For simplicity we will be focused on the prediction function of the form:

\[ y = \text{sign}(w \cdot x) \]

What if we need the bias \( b \)?

Let \( w' = (w_1, w_2, \ldots, w_d, b) \) and \( x' = (x_1, x_2, \ldots, x_d, 1) \) and we have:

\[ w' \cdot x' = \sum_{i=1}^{d} w_i x_i + b \cdot 1 \]
The Perception algorithm

Input: training set \((x_i, y_i), \ldots, (x_m, y_m)\), \(x_i \in \mathbb{R}^d, y_i \in \{-1, +1\} \quad \forall i\)

Initialize: \(w^{(0)} = \theta\)

for \(t = 0, 1, 2, \ldots\)

if \(\exists i\) such that \(y_i w_t^T x_i > 0\) then

\[ w^{(t+1)} = w^{(t)} + y_i x_i \]

else

output \(w^{(t)}\)

---

**Theorem (Novikoff, 1962)**

Assume that \((x,y_1), \ldots, (x_m, y_m)\) is separable, that is there exist a vector \(w^*\) such that

\[ y_i w^* x_i > 0 \quad \forall i \]

where \(y > 0\).

Assume \(\|w^*\| = 1\) and \(R = \max_i \|x_i\|\)

Then the number of mistakes made by the Perceptron algorithm is at most

\[ M \leq \left(\frac{4R}{y}\right)^2 \]

**Proof**

1. If \(y_i w^{(t)} x_i < 0\) we have

\[ w^{(t+1)} = w^{(t)} + y_i x_i \]

\[ w^{(t+1)} = w^* + (w^{(t)} + y_i x_i) w^* = w^{(t)} w^* + y_i w^* x_i \]

\[ \geq w^{(t)} w^* + y \]

Therefore

\[ w^{(t)} w^* \geq M y \]

2. \(\|w^{(t+1)}\|^2 = \|w^{(t)} + y_i x_i\|^2 = \|w^{(t)}\|^2 + 2 y_i \cdot w^{(t)} x_i + \|x_i\|^2 \leq \|w^{(t)}\|^2 + R^2 \]

Therefore

\[ \|w^{(t+1)}\|^2 \leq M R^2 \]

(1) + (2) \implies \(\sqrt{M} R \geq \|w^{(t+1)}\| \geq w^{(t+1)} w^* \geq M y \implies M \leq \frac{R^2}{y^2} \)
Remarks: 1) the number of mistakes depends on R but not on d, the number of features

2) Perceptron algorithm solves ERM with the following surrogate loss function

\[ \ell(y, \hat{y}) = \mathbb{I} \{ y \neq \hat{y} \} \]

\[ \downarrow \]

surrogate loss \[ \ell(x, y; w) = \max \{ 0, -y \cdot w \cdot x \} \]

\[ \hat{w}^* = \arg \min_w \frac{1}{m} \sum_{i=1}^{m} \max \{ 0, -y_i \cdot w \cdot x_i \} \]

we will see next class that the algorithm to find \( \hat{w} \) is the Perceptron algo.

What happens if the data is not separable?

Let \( (x_i, y_i), \ldots, (x_m, y_m) \) a sequence of labeled examples with \( R = \max_i \| x_i \| \).

Let \( u \) be a vector such that \( \| u \| = 1 \). Define the deviation of each example as

\[ d_i = \max \{ 0, y_i - y \cdot u \cdot x_i \} \quad \forall i \]

\[ y > 0 \]

and let \( D = \sqrt{\sum_{i=1}^{m} d_i^2} \). The number of mistakes by the Perceptron is

\[ M \leq \left( \frac{R + D}{y} \right)^2 \]
\[ x_i' = \begin{bmatrix} x_i \\ \vdots \\ 0 \\ \Delta o \vdots \\ 0 \end{bmatrix} \{ n \} \quad m \]

\[ u' = \begin{bmatrix} \uparrow \\ u/2 \\ \vdots \\ y_i d_i/z \Delta \\ \vdots \\ y_m d_m/z \Delta \end{bmatrix} \{ n \} \quad m \]

\[ Z = \sqrt{1 + D^2 / \Delta^2} \]

\[ \| M \|^2 = \frac{1}{2^2} \| M \|^2 + \sum_{i=1}^{m} \frac{(y_i d_i)^2}{2 \Delta} \]

\[ = \| M \|^2 \left( \frac{1}{2^2} + \frac{1}{2^2 \Delta^2} \sum_{i=1}^{m} d_i^2 \right) \]

\[ = \frac{1}{\Delta^2} \left( 1 + \frac{D^2}{\Delta^2} \right) = 1 \]

\[ Z = \sqrt{1 + D^2 / \Delta^2} \]

\[ y_i u' x_i' = y_i \frac{m x_i}{2} + y_i \Delta \frac{y_i d_i}{2 \Delta} = \frac{(y_i u x_i + d_i)}{2} \]

\[ \geq \frac{y_i m x_i + y - y_i u x_i'}{Z} = \frac{y}{Z} \]

\[ \frac{y}{\sqrt{1 + D^2 / \Delta^2}} \]

\[ \| x_i' \|^2 \leq R^2 + \Delta^2 \]

\[ \Rightarrow M \leq \frac{(R^2 + \Delta^2)}{\frac{y^2}{1 + D^2 / \Delta^2}} \]

Setting \( \Delta = \sqrt{R D} \) minimizes the bound \( M \leq \left( \frac{R + \Delta}{y} \right)^2 \)