Review of Lecture 2

Is Learning feasible?

Yes, in a **probabilistic** sense.

Since $g$ has to be one of $h_1, h_2, \cdots, h_M$, we conclude that

If:

$$|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon$$

Then:

$$|E_{\text{in}}(h_1) - E_{\text{out}}(h_1)| > \epsilon \quad \text{or}$$

$$|E_{\text{in}}(h_2) - E_{\text{out}}(h_2)| > \epsilon \quad \text{or}$$

$$\cdots$$

$$|E_{\text{in}}(h_M) - E_{\text{out}}(h_M)| > \epsilon$$

This gives us an added $M$ factor.

\[
\mathbb{P}\left[ |E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon \right] \leq 2e^{-2\epsilon^2 N}
\]
Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology

Lecture 3: Linear Models I
Outline

• Input representation

• Linear Classification

• Linear Regression

• Nonlinear Transformation
A real data set

7 4 7 3 0 3 1 0 1
8 1 1 7 7 4 8 0 1
2 7 4 8 7 3 7 4 1
0 7 4 1 3 7 7 4 5
9 7 4 1 3 7 7 4 8
0 2 0 8 6 6 2 0 8
'raw' input $\mathbf{x} = (x_0, x_1, x_2, \cdots, x_{256})$

linear model: $(w_0, w_1, w_2, \cdots, w_{256})$

Features: Extract useful information, e.g.,

intensity and symmetry $\mathbf{x} = (x_0, x_1, x_2)$

linear model: $(w_0, w_1, w_2)$
Illustration of features

$x = (x_0, x_1, x_2)$  \hspace{1cm} x_1: intensity  \hspace{1cm} x_2: symmetry
What PLA does

Evolution of $E_{in}$ and $E_{out}$

Final perceptron boundary
The ‘pocket’ algorithm

PLA:

Pocket:
Classification boundary - PLA versus Pocket

PLA:

Pocket:
Outline

- Input representation
- Linear Classification
- Linear Regression \( \text{regression} \equiv \text{real-valued output} \)
- Nonlinear Transformation
Credit again

Classification: Credit approval \((\text{yes/no})\)

Regression: Credit line \((\text{dollar amount})\)

Input: \(\mathbf{x} = \)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>23 years</td>
</tr>
<tr>
<td>annual salary</td>
<td>$30,000</td>
</tr>
<tr>
<td>years in residence</td>
<td>1 year</td>
</tr>
<tr>
<td>years in job</td>
<td>1 year</td>
</tr>
<tr>
<td>current debt</td>
<td>$15,000</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
</tbody>
</table>

Linear regression output: \(h(\mathbf{x}) = \sum_{i=0}^{d} w_i \ x_i = \mathbf{w}^T \mathbf{x}\)
The data set

Credit officers decide on credit lines:

\[(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\]

\(y_n \in \mathbb{R}\) is the credit line for customer \(x_n\).

Linear regression tries to replicate that.
How to measure the error

How well does $h(x) = w^T x$ approximate $f(x)$?

In linear regression, we use squared error $(h(x) - f(x))^2$

\[
\text{in-sample error: } E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2
\]
Illustration of linear regression
The expression for $E_{in}$

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} (w^\top x_n - y_n)^2$$

$$= \frac{1}{N} \|Xw - y\|^2$$

where

$$X = \begin{bmatrix}
-x_1^\top \\
-x_2^\top \\
\vdots \\
-x_N^\top
\end{bmatrix}, \quad y = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{bmatrix}$$
Minimizing $E_{\text{in}}$

$$E_{\text{in}}(w) = \frac{1}{N} \|Xw - y\|^2$$

$$\nabla E_{\text{in}}(w) = \frac{2}{N} X^T (Xw - y) = 0$$

$$X^T Xw = X^T y$$

$$w = X^\dagger y \text{ where } X^\dagger = (X^T X)^{-1} X^T$$

$X^\dagger$ is the ‘pseudo-inverse’ of $X$
The pseudo-inverse

\[ X^\dagger = (X^\top X)^{-1}X^\top \]
The linear regression algorithm

1. Construct the matrix $X$ and the vector $y$ from the data set $(x_1, y_1), \ldots, (x_N, y_N)$ as follows

   $$X = \begin{bmatrix} -x_1^T \\ -x_2^T \\ \vdots \\ -x_N^T \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}.$$ 

   input data matrix target vector

2. Compute the pseudo-inverse $X^\dagger = (X^T X)^{-1} X^T$.

3. Return $w = X^\dagger y$. 

Learning From Data - Lecture 3 17/23
Linear regression for classification

Linear regression learns a real-valued function $y = f(x) \in \mathbb{R}$

Binary-valued functions are also real-valued! $\pm 1 \in \mathbb{R}$

Use linear regression to get $\mathbf{w}$ where $\mathbf{w}^T \mathbf{x}_n \approx y_n = \pm 1$

In this case, $\text{sign}(\mathbf{w}^T \mathbf{x}_n)$ is likely to agree with $y_n = \pm 1$

Good initial weights for classification
Linear regression boundary
Outline

• Input representation

• Linear Classification

• Linear Regression

• Nonlinear Transformation
Linear is limited

Data:  

Hypothesis:
Another example

Credit line is affected by ‘years in residence’

but **not** in a linear way!

Nonlinear $[[x_i < 1]]$ and $[[x_i > 5]]$ are better.

Can we do that with linear models?
Linear in what?

Linear regression implements

\[ \sum_{i=0}^{d} w_i x_i \]

Linear classification implements

\[ \text{sign} \left( \sum_{i=0}^{d} w_i x_i \right) \]

Algorithms work because of **linearity in the weights**
Transform the data nonlinearly

\[(x_1, x_2) \xrightarrow{\Phi} (x_1^2, x_2^2)\]