Lecture 9: Linear Regression

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

Department of Computer Science & Information Engineering
National Taiwan University
(國立台灣大學資訊工程系)
Roadmap

1. When Can Machines Learn?
2. Why Can Machines Learn?

Lecture 8: Noise and Error
learning can happen with target distribution $P(y|x)$ and low $E_{in}$ w.r.t. $err$

3. How Can Machines Learn?

Lecture 9: Linear Regression
- Linear Regression Problem
- Linear Regression Algorithm
- Generalization Issue
- Linear Regression for Binary Classification

4. How Can Machines Learn Better?
Credit Limit Problem

unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$

(training examples $D : (x_1, y_1), \ldots, (x_N, y_N)$)

(historical records in bank)

hypothesis set $\mathcal{H}$

(set of candidate formula)

$\mathcal{Y} = \mathbb{R}$: regression

<table>
<thead>
<tr>
<th>age</th>
<th>23 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>female</td>
</tr>
<tr>
<td>annual salary</td>
<td>NTD 1,000,000</td>
</tr>
<tr>
<td>year in residence</td>
<td>1 year</td>
</tr>
<tr>
<td>year in job</td>
<td>0.5 year</td>
</tr>
<tr>
<td>current debt</td>
<td>200,000</td>
</tr>
</tbody>
</table>

credit limit? 100,000

final hypothesis $g \approx f$

('learned' formula to be used)
Linear Regression Hypothesis

<table>
<thead>
<tr>
<th>age</th>
<th>23 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>annual salary</td>
<td>NTD 1,000,000</td>
</tr>
<tr>
<td>year in job</td>
<td>0.5 year</td>
</tr>
<tr>
<td>current debt</td>
<td>200,000</td>
</tr>
</tbody>
</table>

- For \( \mathbf{x} = (x_0, x_1, x_2, \cdots, x_d) \) ‘features of customer’, approximate the desired credit limit with a weighted sum:

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

- linear regression hypothesis: \( h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \)

\( h(\mathbf{x}) \): like perceptron, but without the sign
Linear Regression

**Linear Regression Problem**

Illustration of Linear Regression

\[ \mathbf{x} = (x) \in \mathbb{R} \]

\[ \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 \]

**linear regression:**
find lines/hyperplanes with small residuals
The Error Measure

popular/historical error measure:
squared error \( \text{err}(\hat{y}, y) = (\hat{y} - y)^2 \)

in-sample

\[
E_{\text{in}}(hw) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2
\]

out-of-sample

\[
E_{\text{out}}(w) = \mathbb{E}_{(x,y) \sim P} (w^T x - y)^2
\]

next: how to minimize \( E_{\text{in}}(w) \)?
Consider using linear regression hypothesis $h(x) = w^T x$ to predict the credit limit of customers $x$. Which feature below shall have a positive weight in a **good hypothesis** for the task?

1. birth month
2. monthly income
3. current debt
4. number of credit cards owned

**Reference Answer:** 2

Customers with higher monthly income should naturally be given a higher credit limit, which is captured by the positive weight on the ‘monthly income’ feature.
Matrix Form of $E_{in}(\mathbf{w})$

$$E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{x}_n - y_n)^2 = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n^T \mathbf{w} - y_n)^2$$

$$= \frac{1}{N} \left\| \begin{bmatrix} \mathbf{x}_1^T \mathbf{w} - y_1 \\ \mathbf{x}_2^T \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_N^T \mathbf{w} - y_N \end{bmatrix} \right\|^2$$

$$= \frac{1}{N} \left\| \begin{bmatrix} - & \mathbf{x}_1^T & - \\ - & \mathbf{x}_2^T & - \\ - & \mathbf{x}_N^T & - \end{bmatrix} \right\| \mathbf{w} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \right\|^2$$

$$= \frac{1}{N} \left\| \mathbf{X} \mathbf{w} - \mathbf{y} \right\|^2$$
\[ \min_w E_{in}(w) = \frac{1}{N} \| Xw - y \|^2 \]

- \( E_{in}(w) \): continuous, differentiable, **convex**
- necessary condition of ‘best’ \( w \)

\[ \nabla E_{in}(w) \equiv \begin{bmatrix} \frac{\partial E_{in}(w)}{\partial w_0} \\ \frac{\partial E_{in}(w)}{\partial w_1} \\ \vdots \\ \frac{\partial E_{in}(w)}{\partial w_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \]

—not possible to ‘roll down’

**task:** find \( w_{\text{LIN}} \) such that \( \nabla E_{in}(w_{\text{LIN}}) = 0 \)
The Gradient $\nabla E_\text{in}(\mathbf{w})$

$$E_\text{in}(\mathbf{w}) = \frac{1}{N} \| \mathbf{Xw} - \mathbf{y} \|^2 = \frac{1}{N} \left( \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} - 2 \mathbf{w}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right)$$

**one $\mathbf{w}$ only**

$$E_\text{in}(\mathbf{w}) = \frac{1}{N} \left( a \mathbf{w}^2 - 2b \mathbf{w} + c \right)$$

$$\nabla E_\text{in}(\mathbf{w}) = \frac{1}{N} \left( 2a \mathbf{w} - 2b \right)$$

simple! :-)

**vector $\mathbf{w}$**

$$E_\text{in}(\mathbf{w}) = \frac{1}{N} \left( \mathbf{w}^T \mathbf{A} \mathbf{w} - 2 \mathbf{w}^T \mathbf{b} + c \right)$$

$$\nabla E_\text{in}(\mathbf{w}) = \frac{1}{N} \left( 2 \mathbf{A} \mathbf{w} - 2 \mathbf{b} \right)$$

similar (derived by definition)

$$\nabla E_\text{in}(\mathbf{w}) = \frac{2}{N} \left( \mathbf{X}^T \mathbf{X} \mathbf{w} - \mathbf{X}^T \mathbf{y} \right)$$
**Optimal Linear Regression Weights**

**Task:** Find \( w_{\text{LIN}} \) such that 

\[
\frac{2}{N} \left( X^T X w - X^T y \right) = \nabla E_{\text{in}}(w) = 0
\]

**Invertible \( X^T X \)**
- **Easy and unique solution**
  \[
  w_{\text{LIN}} = \left( X^T X \right)^{-1} X^T y
  \]
- **Often the case because** \( N \gg d + 1 \)

**Singular \( X^T X \)**
- **Many optimal solutions**
- **One of the solutions**
  \[
  w_{\text{LIN}} = X^\dagger y
  \]
  by defining \( X^\dagger \) in other ways

**Practical Suggestion:**
- Use **well-implemented \( \dagger \) routine**
  instead of \( (X^T X)^{-1} X^T \)
- For numerical stability when **almost-singular**
Linear Regression Algorithm

1. from $\mathcal{D}$, construct input matrix $X$ and output vector $y$ by

$$X = \begin{bmatrix} \vdots & \vdots & \vdots \\ - & - & x_1^T \\ - & - & x_2^T \\ & & \ddots \\ - & - & x_N^T \end{bmatrix}_{N \times (d+1)}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1}$$

2. calculate pseudo-inverse

$$X^\dagger \in (d+1) \times N$$

3. return

$$w_{\text{LIN}} = X^\dagger y \in (d+1) \times 1$$

simple and efficient with good $\dagger$ routine
After getting $\mathbf{w}_{\text{LIN}}$, we can calculate the predictions $\hat{y}_n = \mathbf{w}_{\text{LIN}}^T \mathbf{x}_n$. If all $\hat{y}_n$ are collected in a vector $\hat{\mathbf{y}}$ similar to how we form $\mathbf{y}$, what is the matrix formula of $\hat{\mathbf{y}}$?

1. $\mathbf{y}$
2. $\mathbf{X}^T \mathbf{y}$
3. $\mathbf{X} \mathbf{X}^+ \mathbf{y}$
4. $\mathbf{X} \mathbf{X}^+ \mathbf{X}^T \mathbf{y}$

Reference Answer: 3

Note that $\hat{\mathbf{y}} = \mathbf{X} \mathbf{w}_{\text{LIN}}$. Then, a simple substitution of $\mathbf{w}_{\text{LIN}}$ reveals the answer.
Is Linear Regression a ‘Learning Algorithm’?

No!
- analytic (closed-form) solution, ‘instantaneous’
- not improving $E_{in}$ nor $E_{out}$ iteratively

Yes!
- good $E_{in}$? yes, optimal!
- good $E_{out}$?
  - yes, finite $d_{VC}$ like perceptrons
- improving iteratively?
  - somewhat, within an iterative pseudo-inverse routine

If $E_{out}(\mathbf{w}_{\text{LIN}})$ is good, learning ‘happened’!
**Benefit of Analytic Solution:**

‘Simpler-than-VC’ Guarantee

\[
\overline{E_{\text{in}}} = \mathcal{E}_{D \sim P^N} \left\{ E_{\text{in}}(\mathbf{w}_{\text{LIN}} \text{ w.r.t. } D) \right\} \text{ to be shown} = \text{noise level} \cdot \left(1 - \frac{d+1}{N}\right)
\]

\[
E_{\text{in}}(\mathbf{w}_{\text{LIN}}) = \frac{1}{N} \| \mathbf{y} - \hat{\mathbf{y}} \|^2 = \frac{1}{N} \| \mathbf{y} - \mathbf{X} \mathbf{X}^\dagger \mathbf{y} \|_2^2
\]

\[
= \frac{1}{N} \| (\mathbf{I} - \mathbf{X} \mathbf{X}^\dagger) \mathbf{y} \|_2^2
\]

call \( \mathbf{X} \mathbf{X}^\dagger \) the hat matrix \( \mathbf{H} \)
because it puts \( \wedge \) on \( \mathbf{y} \)
Geometric View of Hat Matrix

- \( \hat{y} = Xw_{\text{LIN}} \) within the span of \( X \) columns
- \( y - \hat{y} \) smallest: \( y - \hat{y} \perp \text{span} \)
- \( H \): project \( y \) to \( \hat{y} \in \text{span} \)
- \( I - H \): transform \( y \) to \( y - \hat{y} \perp \text{span} \)

claim: \( \text{trace}(I - H) = N - (d + 1) \). Why? :-)

\( y \)
\( \hat{y} \)
\( \text{span of } X \)
An Illustrative ‘Proof’

- if \( y \) comes from some ideal \( f(X) \in \text{span} \) plus noise
- noise transformed by \( I - H \) to be \( y - \hat{y} \)

\[
E_{in}(w_{LIN}) = \frac{1}{N} \| y - \hat{y} \|^2 = \frac{1}{N} \| (I - H)\text{noise} \|^2 = \frac{1}{N} (N - (d + 1)) \| \text{noise} \|^2
\]

\[
\overline{E}_{in} = \text{noise level} \cdot \left( 1 - \frac{d+1}{N} \right)
\]

\[
\overline{E}_{out} = \text{noise level} \cdot \left( 1 + \frac{d+1}{N} \right) \text{(complicated!)}
\]
The Learning Curve

\[
\overline{E_{\text{out}}} = \text{noise level} \cdot (1 + \frac{d+1}{N})
\]

\[
\overline{E_{\text{in}}} = \text{noise level} \cdot (1 - \frac{d+1}{N})
\]

- both converge to \( \sigma^2 \) (noise level) for \( N \to \infty \)
- expected generalization error: \( \frac{2(d+1)}{N} \)
  — similar to worst-case guarantee from VC

linear regression (LinReg): learning ‘happened’!
Which of the following property about $H$ is not true?

1. $H$ is symmetric
2. $H^2 = H$ (double projection = single one)
3. $(I - H)^2 = I - H$ (double residual transform = single one)
4. none of the above

Reference Answer: 4

You can conclude that 2 and 3 are true by their physical meanings! :-)

Hsuan-Tien Lin (NTU CSIE)
### Linear Classification vs. Linear Regression

#### Linear Classification

- **Function Space**: \( \mathcal{Y} = \{-1, +1\} \)
- **Hypothesis**: \( h(x) = \text{sign}(w^T x) \)
- **Error**: \( \text{err}(\hat{y}, y) = \left\| \hat{y} \neq y \right\| \)

**NP-hard to solve in general**

#### Linear Regression

- **Function Space**: \( \mathcal{Y} = \mathbb{R} \)
- **Hypothesis**: \( h(x) = w^T x \)
- **Error**: \( \text{err}(\hat{y}, y) = (\hat{y} - y)^2 \)

**Efficient analytic solution**

\( \{-1, +1\} \subset \mathbb{R} \): Linear regression for classification?

1. Run LinReg on binary classification data \( D \) (efficient)
2. Return \( g(x) = \text{sign}(w_{\text{LIN}}^T x) \)

But explanation of this heuristic?
Relation of Two Errors

\[
\text{err}_{0/1} = \left[ \text{sign}(w^T x) \neq y \right] \quad \text{err}_{\text{sqr}} = (w^T x - y)^2
\]

desired \( y = 1 \)

\[
\text{err}_{0/1} \leq \text{err}_{\text{sqr}}
\]
Linear Regression for Binary Classification

\[ \text{err}_{0/1} \leq \text{err}_{sqr} \]

\[
\text{classification } E_{\text{out}}(w) \leq \text{classification } E_{\text{in}}(w) + \sqrt{\ldots}
\]

\[
\leq \text{regression } E_{\text{in}}(w) + \sqrt{\ldots}
\]

- (loose) upper bound \( \text{err}_{sqr} \) as \( \hat{\text{err}} \) to approximate \( \text{err}_{0/1} \)
- trade bound tightness for efficiency

\( w_{\text{LIN}} \): useful baseline classifier, or as initial PLA/pocket vector
Which of the following functions are upper bounds of the pointwise 0/1 error \[ \| \text{sign}(w^T x) \neq y \| \] for \( y \in \{-1, +1\} \)?

1. \[ \exp(-y w^T x) \]
2. \[ \max(0, 1 - y w^T x) \]
3. \[ \log_2(1 + \exp(-y w^T x)) \]
4. all of the above

Reference Answer: 4

Plot the curves and you’ll see. Thus, all three can be used for binary classification. In fact, all three functions connect to very important algorithms in machine learning and we will discuss one of them soon in the next lecture. 

Stay tuned. :-(
Linear Regression

Summary

1. When Can Machines Learn?
2. Why Can Machines Learn?

Lecture 8: Noise and Error

3. How Can Machines Learn?

Lecture 9: Linear Regression

- Linear Regression Problem
  use hyperplanes to approximate real values
- Linear Regression Algorithm
  analytic solution with pseudo-inverse
- Generalization Issue
  \[ E_{out} - E_{in} \approx \frac{2(d+1)}{N} \] on average
- Linear Regression for Binary Classification
  0/1 error \( \leq \) squared error

4. How Can Machines Learn Better?