INTRODUCTION TO MACHINE LEARNING

LINEAR REGRESSION

* Some contents are adapted from Dr. Hung Huang and Dr. Chengkai Li at UT Arlington

Mignon Kang, Ph.D.
Department of Computer Science @ UNLV
Correlation (r)

- Linear association between two variables
- Show how to determine both the nature and strength of relationship between two variables
- Correlation lies between +1 to -1
- Zero correlation indicates that there is no relationship between the variables
- Pearson correlation coefficient
  - most familiar measure of dependence between two quantities
Correlation (r)
Correlation ($r$)

$$
\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},
$$

where $E$ is the expected value operator, $\text{cov}(.,.)$ means covariance, and $\text{corr}(.,.)$ is a widely used alternative notation for the correlation coefficient.

Reference: https://en.wikipedia.org/wiki/Correlation_and_dependence
Linear Regression

Samples with ONE independent variable

Samples with TWO independent variables

Given examples \((x_i, y_i)_{i=1...n}\)

Predict \(y_{n+1}\) given a new point \(x_{n+1}\)
Linear Regression

Samples with ONE independent variable

\[ \hat{y}_{n+1} \]

\[ x_{n+1} \]

Samples with TWO independent variables
How to represent the data as a vector/matrix

We assume a model:

$$y = b_0 + bX + \epsilon,$$

where $b_0$ and $b$ are intercept and slope, known as coefficients or parameters. $\epsilon$ is the error term (typically assumes that $\epsilon \sim N(\mu, \sigma^2)$)
Linear Regression

- Simple linear regression
  - A single independent variable is used to predict
- Multiple linear regression
  - Two or more independent variables are used to predict
Linear Regression

- How to represent the data as a vector/matrix
  - Include bias constant (intercept) in the input vector
    - \( X \in \mathbb{R}^{n \times (p+1)} \), \( y \in \mathbb{R}^n \), \( b \in \mathbb{R}^{p+1} \), and \( e \in \mathbb{R}^n \)

\[
y = X \cdot b + e
\]

\[
X = \{1, x_1, x_2, \ldots, x_p\}, \quad b = \{b_0, b_1, b_2, \ldots, b_p\}^T
\]
\[
y = \{y_1, y_2, \ldots, y_n\}^T, \quad e = \{e_1, e_2, \ldots, e_n\}^T
\]

\( \cdot \) is a dot product

equivalent to
\[
y_i = 1 \cdot b_0 + x_{i1} b_1 + x_{i2} b_2 + \cdots + x_{ip} b_p \quad (1 \leq i \leq n)
\]
Find the optimal coefficient vector $\mathbf{b}$ that makes the most similar observation

$$
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_n
\end{bmatrix} = 
\begin{bmatrix}
  1 & x_{11} & \cdots & x_{1p} \\
  1 & x_{21} & \cdots & \vdots \\
  1 & x_{n1} & \cdots & x_{np}
\end{bmatrix}
\begin{bmatrix}
  b_0 \\
  \vdots \\
  b_p
\end{bmatrix} + 
\begin{bmatrix}
  e_1 \\
  \vdots \\
  e_n
\end{bmatrix}
$$
Ordinary Least Squares (OLS)

\[ y = Xb + e \]

- Estimate the unknown parameters \((b)\) in linear regression model
- Minimizing the sum of the squares of the differences between the observed responses and the predicted by a linear function

\[
\text{Sum squared error} = \sum_{i=1}^{n} (y_i - x_i \cdot b)^2
\]
Ordinary Least Squares (OLS)

\[ y = X \beta + \epsilon \]

\[ X_i = \begin{pmatrix} 1 \\ x_{i,1} \\ x_{i,2} \end{pmatrix} \]
Need to minimize the error

$$\min J(b) = \sum_{i=1}^{n} (y_i - x_{i,*}b)^2$$

To obtain the optimal set of parameters ($b$), derivatives of the error w.r.t. each parameter must be zero.
Optimization

\[ J = e^T e = (y - Xb)'(y - Xb) \]
\[ = (y' - b'X')(y - Xb) \]
\[ = y'y - y'Xb - b'X'y + b'X'Xb \]
\[ = y'y - 2b'X'y + b'X'Xb \]

\[
\frac{\partial e'e}{\partial b} = -2X'y + 2X'Xb = 0 \\
(X'X)b = X'y \\
b = (X'X)^{-1}X'y
\]
The Happiness Formula

- \[ H = \frac{(G + DH + C + 3R)}{6} \]

- Happiness ("H") is equal to
  - your level of Gratitude ("G") +
  - the degree to which you are living consistent with your own personal Definition of Happiness ("DH") +
  - how much you Contribute to others ("C") +
  - your success in what I call the 3 R’s of happiness ("3R")

Ref: http://www.behappy101.com/happiness-formula.html
We assumed that all variables are continuous variables.

Categorical variables:
- Ordinal variables - Encode data with continuous values
  - Evaluation: Excellent (5), Very good (4), Good (3), Poor (2), Very poor (1)
- Nominal variables – Use dummy variables
  - Department: Computer, Biology, Physics

<table>
<thead>
<tr>
<th></th>
<th>Computer</th>
<th>Biology</th>
<th>Physics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Biology</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Physics</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Linear regression for classification

- For binary classification
  - Encode class labels as $y = \{0, 1\}$ or $\{-1, 1\}$
  - Apply OLS
  - Check which class the prediction is closer to
    - If class 1 is encoded to 1 and class 2 is -1.

$$
\text{class 1 if } f(x) \geq 0 \\
\text{class 2 if } f(x) < 0
$$

- Linear models are NOT optimized for classification
- Logistic regression
Linear regression for classification

- ROC for classification

$$f(x) \geq \lambda$$

If $f(x)$ is less than $\lambda$, class 1. Otherwise class 2.

How can we know the optimal $\lambda$?

- Let’s revisit EVALUATION.
Linear regression for classification

- Multi-label classification
  - Encode classes label as:

<table>
<thead>
<tr>
<th></th>
<th>Computer</th>
<th>Biology</th>
<th>Physics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Biology</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Physics</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Perform linear regression multiple times for each class
- Consider $y$ and $b$ as matrix
Assumptions in Linear regression

- Linearity of independent variable in the predictor
  - normally good approximation, especially for high-dimensional data

- Error has normal distribution, with mean zero and constant variance
  - important for tests

- Independent variables are independent from each other
  - Otherwise, it causes a **multicollinearity** problem; two or more predictor variables are highly correlated.
  - Should remove them
Think more!

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1</td>
<td>5.2</td>
</tr>
<tr>
<td>Feature 2</td>
<td>0.1</td>
</tr>
<tr>
<td>Feature 3</td>
<td>-6.6</td>
</tr>
<tr>
<td>Feature 4</td>
<td>0</td>
</tr>
</tbody>
</table>

- How can we interpret this model?
- What is the most useless feature?
  - Is it always useless to explain the dependent variable?
- What do negative coefficients represent?
- What is the most informative feature?
Different views between Statistics and CS

- In Statistics, description of the model is often more important.
  - Which variables are more informative and reliable to describe the responses? → p-values
  - How much information do the variables have?

- In Computer Science, the accuracy of prediction and classification is more important.
  - How well can we predict/classify?
Discussion

- What if data is imbalanced data?
- Why does OLS take squares instead of absolute values?