Supervised Learning in Data Science

Data Analysis and Policy

• Google wants to hire new software engineers that increase productivity as much as possible. It collects the following data on 2017 hires:
  - Productivity increase contribution
  - Cumulative college GPA
  - Number of undergraduate internships held
  - Extracurricular activity participation
  - Number of CS courses taken

• What can Google do with this data to inform this year’s hiring decisions?

Supervised Learning

• Have a training data set, know what output given inputs should look like
• Want to teach computer to give correct output given inputs

The Regression Problem

• Theory posits that some quantitative output is a function of input and random error:
  \[ Y = f(X) + \epsilon \]

• Given a set of observations \((X^{(1)}, Y^{(1)}), (X^{(2)}, Y^{(2)}), ..., (X^{(n)}, Y^{(n)})\)
  try to create predictor function \(f(X)\) which produces an output as close to \(Y\) as possible on a new input value of \(X\)

Linear Predictor Function

• With inputs \(X_1, ..., X_n\), actual output \(Y\), linear predictor function takes the form
  \[ f(X) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_n X_n \]

  where \(\hat{\beta}_0, ..., \hat{\beta}_n\) are user determined parameters

• Single variable case:
  \[ f(X) = \hat{\beta}_0 + \hat{\beta}_1 X \]

• Note: predicted output \(\hat{Y} = f(X)\)
Linear Predictor Function

- E.g. income based on years of education
  \[ \hat{\text{Inc}} = f(\text{YOE}) + \epsilon \]
  \[ \hat{\text{Inc}} = \hat{\beta}_0 + \hat{\beta}_1 \text{YOE} \]

Cost Function

- How to evaluate performance of \( f \) i.e. measure goodness-of-fit?
  Mean squared error
  \[ C(\hat{\beta}_0, \hat{\beta}_1) = \frac{1}{2n} \sum_{i=1}^{n} (f(X^{(i)}) - Y^{(i)})^2 \]
- Goal: choose \( \hat{\beta}_0, \hat{\beta}_1 \) to minimize \( C(\hat{\beta}_0, \hat{\beta}_1) \), maximize goodness of fit

Gradient Descent

- How to "learn" from cost function and reduce it? Correction factors
  \[ \delta_0 = \frac{1}{n} \sum_{i=1}^{n} (f(X^{(i)}) - Y^{(i)}) \]
  \[ \delta_1 = \frac{1}{n} \sum_{i=1}^{n} (f(X^{(i)}) - Y^{(i)})X^{(i)} \]
- Reassign \( \hat{\beta}_0, \hat{\beta}_1 \) such that:
  \[ \hat{\beta}_0 = \hat{\beta}_0 - \alpha \delta_0 \]
  \[ \hat{\beta}_1 = \hat{\beta}_1 - \alpha \delta_1 \]
- \( \alpha \) is the learning rate of the gradient descent algorithm, user determined
  - Too slow if too small; could overshoot if too large

Gradient Descent Example

- \( \mathcal{T}_S = \{(X^{(1)} = 1, Y^{(1)} = 1), (X^{(2)} = 2, Y^{(2)} = 2), (X^{(3)} = 3, Y^{(3)} = 3)\} \)
- Initialize \( \hat{\beta}_0 = 0, \hat{\beta}_1 = 0 \), such that \( f(X) = 0 + 0X \)
  - \( \hat{\beta}_0 \) already optimal, just optimize by \( \hat{\beta}_1 \)
- \( C(\hat{\beta}_0, \hat{\beta}_1) = 2.3333 \)
- Let \( \alpha = 0.1 \)
- \( \delta_0 = -4.6667 \)
- \( \hat{\beta}_1 = 0 - 0.1 \times (-4.6667) = 0.4667 \)

- \( \mathcal{T}_S = \{(X^{(1)} = 1, Y^{(1)} = 1), (X^{(2)} = 2, Y^{(2)} = 2), (X^{(3)} = 3, Y^{(3)} = 3)\} \)
- \( C(\hat{\beta}_0, \hat{\beta}_1) = 0.6636 \)
- Let \( \alpha = 0.1 \)
- \( \delta_0 = -2.4887 \)
- \( \hat{\beta}_1 = 0.4667 - 0.1 \times (-2.4887) = 0.7156 \)
Gradient Descent Example

- \( T = \{(x^{(1)} = 1, y^{(1)} = 1), (x^{(2)} = 2, y^{(2)} = 2), (x^{(3)} = 3, y^{(3)} = 3)\}\)
- \( C(\beta_0, \beta_1) = 0.1887\)
- Let \( \alpha = 0.1\)
- \( \delta = -1.3272\)
- \( \beta_1 = 0.7156 - 0.1 \times (-1.3272) = 0.8483\)

Solving for a Multivariate Predictor

- Linear predictor function: \( f(x_1, x_2, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k\)
- Goal: minimize \( C(\beta_0, \beta_1, \beta_2, \ldots, \beta_k) = \frac{1}{2m} \sum_{i=1}^{m} (y^{(i)} - f(x^{(i)}))^2\)
- \( \delta_0 \) unchanged, \( \delta_j = \frac{1}{n} \sum_{i=1}^{n} (f^{(0)} - y^{(i)}) x_j^{(i)}\)
- For \( j = 1, 2, \ldots, k \), \( \beta_j = \beta_j - \alpha \delta_j\)

Solving for a Multivariate Predictor

- Algorithm:
  1. Initialize \( \beta_0, \beta_1, \beta_2, \ldots, \beta_k\)
  2. Calculate \( \delta_0, \delta_1, \ldots, \delta_k\)
  3. Repeat step 2 until stopping condition reached

Overfitting

- Occurs when estimation method adheres too closely to training data, picks up random noise and poorly expresses actual structure of data
- Can result from too many variables being used in linear regression model
- E.g. if one is trying to predict an individual's income, birthday probably isn't relevant, but doesn't hurt to consider it

Omitted Variable Bias

- Occurs when important factors aren't considered in estimation method
  - i.e. when relevant variables aren't included in linear regression model
- Results in certain variables being considered more or less important than they really are
  - E.g. if innate ability explains both years of education and income, education can act as a proxy for ability in absence of data on the latter, but will be weighted too highly

Bias in Data

- Occurs when training data set has structural differences from overall population
- E.g. if predicting height while only looking at schoolchildren, age is a highly descriptive variable; however, less relevant for overall population