Contents

0.1 Definition . . . 1
0.2 Examples . . . 1
0.3 Simplest Linear Regression . . . 1
0.4 Linear Regression Function Model . . . 2
0.5 Error . . . 2
0.6 Optimization . . . 2
0.7 Linear Regression as a System of Linear Equations . . . 3
0.8 Solving Linear Regression . . . 3
0.8.1 Using Matrices . . . 3
0.8.2 Using Gradient Descent . . . 3
0.9 Linear Regression . . . 3
0.10 Input Normalization . . . 4
0.11 L1/L2 Regularization . . . 4
0.12 Other Ways to Control Overfitting . . . 4
0.13 Bias-Variance Tradeoff . . . 4

0.1 Definition

- In a supervised learning problem, given the input variables X and outputs Y, the goal of linear regression is to learn a function that can predict an output given an input.
- We find the best line (linear function $y = f(X)$) to explain the data.

0.2 Examples

Predicting a continuous outcome variable

- Predicting a company's future stock price using its profit and other financial information.
- Predicting annual rainfall based on local flora and fauna.
- Predicting distance from a traffic light using LIDAR measurements.

0.3 Simplest Linear Regression

- x is an input feature.
- y is the value we're trying to predict.
- The regression model is:

$$y = w_1 x + w_0$$

- Two parameters to estimate –
  - the slope of the line w_1,
  - the y-intercept w_0.
- We basically want to find $\{w_0, w_1\}$ that minimize deviations from the predictor line.

$$\begin{aligned} &\min \sum_{i=1,2,\ldots,n} (y_i - \hat{y}_i)^2 \\ =& \min_{w_0,w_1} \sum_{i=1,2,\ldots,n} (y_i - w_1 x_i - w_0)^2 \end{aligned}$$

0.4 Linear Regression Function Model

Function $f : X \to Y$ is a linear combination of input components

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j$$

$w_0, w_1, \ldots, w_d$ are the parameters (weights)

0.5 Error

- Error function measures how much our predictions deviate from the desired answers.
- Mean-squared error (MSE):

$$\begin{aligned} J_n &= \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 \\ &= \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \end{aligned} \tag{1}$$

- Learning: We want to find the weights minimizing the error.
- In (1), $y_i - \mathbf{w}^T \mathbf{x}_i$ is the residual and . . . .

0.6 Optimization

- For the optimal set of parameters, derivatives of the error with respect to each parameter must be 0.

$$\begin{aligned}\frac{\partial}{\partial w_j} J_n(\mathbf{w}) &= -\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 x_{i0} - w_1 x_{i1} - \cdots - w_d x_{id}) x_{ij} \\ &= 0\end{aligned}$$

- Vector of derivatives:

$$\begin{aligned}\nabla_w &= -\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{w}^T \mathbf{x}_i\right)\mathbf{x}_i \\ &= \mathbf{0}\end{aligned}$$

- By rearranging the terms, we get a system of linear equations with $d+1$ unknowns.

$$w_0 \sum_{i=1}^{n} x_{i0} \cdot x_{ij} + w_1 \sum_{i=1}^{n} x_{i1} \cdot x_{ij} + \cdots + w_d \sum_{i=1}^{n} x_{id} \cdot x_{ij} + \sum_{i=1}^{n} y_i \cdot x_{ij}$$

- Can also be solved through matrix inversion if the matrix is not singular.

$$\mathbf{Aw} = \mathbf{b} \Rightarrow \mathbf{w} = \mathbf{A}^{-1}\mathbf{b}$$

0.7 Linear Regression as a System of Linear Equations

The linear regression model is akin to a system of linear equations. Assuming n training examples with $d+1$ features each –

$$\begin{aligned}1^{\text{st}} \text{ training example:} \quad & y_1 = w_0 + x_{11}w_1 + x_{12}w_2 + \cdots + x_{1d}2_d \\ 2^{\text{nd}} \text{ training example:} \quad & y_2 = w_0 + x_{21}w_1 + x_{22}w_2 + \cdots + x_{2d}2_d \\ \vdots \quad & \\ n^{\text{th}} \text{ training example:} \quad & y_n = w_0 + x_{n1}w_1 + x_{n2}w_2 + \cdots + x_{nd}2_n\end{aligned}$$

0.8 Solving Linear Regression

0.8.1 Using Matrices

- $J_n(\mathbf{w})$ can be rewritten in terms of data matrices X and vectors:

$$J_n(\mathbf{w}) = \frac{1}{2}(\dots)$$

0.8.2 Using Gradient Descent

- Linear regression problem comes down to the problem of solving a set of linear equations:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \nabla_{\mathbf{w}} J_n(\mathbf{w})$$
$$\nabla J_n(\mathbf{w}) = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$
$$\mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

0.9 Linear Regression

- The error function defined for the whole dataset for the linear regression is:

$$J_n = \frac{1}{2n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$$

- Online Gradient Descent: use the most recent sample at each iteration. Instead of MSE for all data points, it uses MSE for an individual sample.

$$\begin{aligned} J_{online} &= Error_i(\mathbf{w}) \\ &= \frac{1}{2}\left(y_i - f(\mathbf{x}_i)\right)^2 \\ \mathbf{w} \leftarrow \mathbf{w} - \eta \ldots \end{aligned}$$

0.10 Input Normalization

- Makes the data very roughly on the same scale.
- Can make a huge difference in online learning.
- For inputs with a large magnitude, the change in the weight is huge.
- Solution: Make all inputs vary in the same range.
- New output:

$$\hat{x_{ij}} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$$

0.11 L1/L2 Regularization

Using L1/L2 Regularization, we can rewrite our loss function as:

$$L_{lasso} = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - f(\mathbf{w}^T\mathbf{x}_i)\right)^2 + \lambda||\mathbf{w}||_1$$
$$L_{ridge} = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - f(\mathbf{w}^T\mathbf{x}_i)\right)^2 + \lambda||\mathbf{w}||_2^2$$

0.12 Other Ways to Control Overfitting

Early-stopping : stopping training when a monitored matrix has stopped improving.

Bagging : learning multiple models in parallel and applying majority voting to choose final predictor.

Dropout : in each iteration, don't update some of the weights.

Injecting noise in the inputs.

0.13 Bias-Variance Tradeoff

- Bias captures the inherent error present in the model. The bias error originates from erroneous assumption(s) in the learning algorithms.
- Bias is the contrast between the mean prediction of our model and the correct prediction.
- Variance captures how much the model changes if it is trained on a different training set.
- Variance is the variation or spread of model prediction values across different data samples.
- Underfitting happens when a model unable to capture the underlying pattern of the data. Such models usually have high bias and low variance.
- It usually happens when there is much fewer amount of data to build an accurate model or when a linear model is used to learn non-linear data.
- Overfitting happens when our model captures the noise along with the underlying pattern in data.
- ...

Bias:

$$(y - \hat{y}) \tag{2}$$

Variance:

$$\frac{1}{k-1}\sum_{j=1}^{k-1}(\hat{y}_j - \hat{y})^2 \tag{3}$$

Total Error:

$$TE = Bias^2 + Variance = (y - \hat{y})^2 + \frac{1}{k-1}\sum_{j=1}^{k-1}(\hat{y}_j - \hat{y})^2 \tag{4}$$

$$\text{Expected Loss} = \text{Total Error} = \text{Bias}^2 + \text{Variance}$$