Introduction to Machine Learning

Linear Regression

Varun Chandola

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
chandola@buffalo.edu

University at Buffalo
Department of Computer Science and Engineering
School of Engeering and Applied Sciences

Outline

Taking the next step

Hypothesis Space, $\mathcal{H}$

- Conjunctive
- Disjunctive
  - Disjunctions of k attributes
- Linear hyperplanes
- $\mathbf{c}_* \notin \mathcal{H}$
- Non-linear network

Input Space, $\mathbf{x}$

- $\mathbf{x} \in \{0, 1\}^d$
- $\mathbf{x} \in \mathbb{R}^d$

Input Space, y

- $y \in \{0, 1\}$
- $y \in \{-1, +1\}$
- $y \in \mathbb{R}$

Linear Regression

- There is one scalar **target** variable y (instead of hidden)
- There is one vector **input** variable x
- Inductive bias:

$$y = \mathbf{w}^\top \mathbf{x}$$

Linear Regression Learning Task

Learn $\mathbf{w}$ given training examples, $\langle \mathbf{X}, \mathbf{y} \rangle$.

Two Interpretations

1. Probabilistic Interpretation

- y is assumed to be normally distributed

$$y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

- or, equivalently:

$$y = \mathbf{w}^\top \mathbf{x} + \epsilon$$

  where $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- y is a *linear combination* of the input variables
- *Given* $\mathbf{w}$ *and* σ^2*, one can find the probability distribution of* y *for a given* $\mathbf{x}$

Two Interpretations

2. Geometric Interpretation

- Fitting a straight line to d dimensional data

$$y = \mathbf{w}^\top \mathbf{x}$$

$$y = \mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

- Will pass through origin
- Add intercept

$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

- Equivalent to adding another column in $\mathbf{X}$ of 1s.

Learning Parameters - MLE Approach

- Find $\mathbf{w}$ and σ^2 that maximize the likelihood of training data

$$
\begin{aligned}
\widehat{\mathbf{w}}_{MLE} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\
\widehat{\sigma}^2_{MLE} &= \frac{1}{N}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w})
\end{aligned}
$$

Learning Parameters - Least Squares Approach

- Minimize *squared loss*

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

- Make prediction ($\mathbf{w}^\top \mathbf{x}_i$) as close to the target (y_i) as possible
- Least squares estimate

$$\widehat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$

Gradient Descent Based Method

- Minimize the squared loss using *Gradient Descent*

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

- Why?

Recap - Linear Regression

Geometric

$$y = \mathbf{w}^\top \mathbf{x}$$

1. Least Squares

$$\widehat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

2. Gradient Descent

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

Bayesian

$$p(y) = \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

1. Maximum Likelihood Estimation

$$\begin{aligned} \widehat{\mathbf{w}} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \\ \widehat{\sigma}^2_{MLE} &= \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X} \end{aligned}$$

Issues with Linear Regression

1. Not truly Bayesian
2. Susceptible to outliers
3. *Too simplistic* - Underfitting
4. No way to control overfitting
5. Unstable in presence of correlated input attributes
6. Gets "confused" by unnecessary attributes

Putting a Prior on $\mathbf{w}$

- "Penalize" large values of $\mathbf{w}$
- A zero-mean Gaussian prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \tau^2 I)$$

- What is posterior of $\mathbf{w}$

$$p(\mathbf{w}|\mathcal{D}) \propto \prod_i \mathcal{N}(y_i|\mathbf{w}^\top \mathbf{x}_i, \sigma^2) p(\mathbf{w})$$

- Posterior is also Gaussian

Posterior Estimates of the Weight Vector

- MAP estimate of **w**

$$\arg\max_{\mathbf{w}} \sum_{i=1}^{N} log\mathcal{N}(y_i|\mathbf{w}^\top\mathbf{x}_i,\sigma^2) + \log\mathcal{N}(\mathbf{w}|0,\tau^2 I)$$

Parameter Estimation for Bayesian Regression

- Prior for $\mathbf{w}$

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \tau^2\mathbf{I}_D)$$

- Posterior for $\mathbf{w}$

$$\begin{aligned} p(\mathbf{w}|\mathbf{y},\mathbf{X}) &= \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\ &= \mathcal{N}(\bar{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}_N)^{-1}\mathbf{X}\mathbf{y}, \sigma^2(\mathbf{X}^\top\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}_N)^{-1}) \end{aligned}$$

- Posterior distribution for $\mathbf{w}$ is also Gaussian
- What will be MAP estimate for $\mathbf{w}$?

Prediction with Bayesian Regression

- For a new $\mathbf{x}^*$, predict y^*
- Point estimate of y^*

$$y^* = \widehat{\mathbf{w}}_{MLE}^\top \mathbf{x}^*$$

- Treating y as a Gaussian random variable

$$p(y^*|\mathbf{x}^*) = \mathcal{N}(\widehat{\mathbf{w}}_{MLE}^\top \mathbf{x}^*, \widehat{\sigma}_{MLE}^2)$$

$$p(y^*|\mathbf{x}^*) = \mathcal{N}(\widehat{\mathbf{w}}_{MAP}^\top \mathbf{x}^*, \widehat{\sigma}_{MAP}^2)$$

Full Bayesian Treatment

- Treating y and $\mathbf{w}$ as random variables

$$p(y^*|\mathbf{x}^*) = \int p(y^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{y})d\mathbf{w}$$

- This is also *Gaussian*!

Handling Non-linear Relationships

- Replace $\mathbf{x}$ with non-linear functions $\phi(\mathbf{x})$

$$p(y|\mathbf{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{w}^\top \phi(\mathbf{x}))$$

- Model is still linear in $\mathbf{w}$
- Also known as **basis function expansion**

Example

$$\phi(x) = [1, x, x^2, \ldots, x^d]$$

- Increasing d results in more complex fits

How to Control Overfitting?

- Use simpler models (linear instead of polynomial)
  - Might have poor results (underfitting)
- Use regularized complex models

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) + \alpha R(\boldsymbol{\Theta})$$

- $R()$ corresponds to the penalty paid for complexity of the model

Examples of Regularization

Ridge Regression

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \alpha\|\mathbf{w}\|^2$$

- Also known as l_2 or *Tikhonov* regularization
- Helps in reducing impact of correlated inputs

Least Absolute Shrinkage and Selection Operator - LASSO

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \alpha|\mathbf{w}|$$

- Also known as l_1 regularization
- Helps in feature selection – favors sparse solutions

Parameter Estimation for Ridge Regression

Exact Loss Function

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \frac{1}{2}\lambda||\mathbf{w}||_2^2$$

MAP Estimate of $\mathbf{w}$

$$\widehat{\mathbf{w}}_{MAP} = (\lambda \mathbf{I}_D + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$

References