Introduction to Machine Learning

Linear Regression

Mingchen Gao

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
Slides adapted from Varun Chandola
mgao8@buffalo.edu

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

Outline

Taking the next step

Input Space, $\mathbf{x}$

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

Output Space, y

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

Linear Regression

- There is one scalar **target** variable y
- 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*

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

- The matrix inversion is expensive or numerically unstable.

Recap - Linear Regression

Geometric

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

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

1. Least Squares

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

2. Gradient Descent

Probabilistic

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

1. Maximum Likelihood Estimation

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

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

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 $\mathbf{w}$

- 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
- Regularized least squares estimate of $\mathbf{w}$

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

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 $\mathbf{w}$

$$\bar{\mathbf{w}}_{\mathbf{MAP}} = (\mathbf{X}^\top \mathbf{X} + \frac{\sigma^{\mathbf{2}}}{\tau^{\mathbf{2}}} \mathbf{I}_{\mathbf{N}})^{-\mathbf{1}} \mathbf{X}^\top \mathbf{y}$$

$$\bar{\sigma}_{\mathbf{MAP}} = \sigma^{\mathbf{2}} (\mathbf{X}^\top \mathbf{X} + \frac{\sigma^{\mathbf{2}}}{\tau^{\mathbf{2}}} \mathbf{I}_{\mathbf{N}})^{-\mathbf{1}}$$

- "Penalize" large values of $\mathbf{w}$, equivalent to *Ridge Regression*

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

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

- Increasing p 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

$$\widehat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} J(\mathbf{\Theta}) + \lambda R(\mathbf{\Theta})$$

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

l_2 Regularization

Ridge Regression

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

- Helps in reducing impact of correlated inputs

Parameter Estimation for Ridge Regression

Exact Loss Function

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

Ridge Estimate of **w**

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

- Equivalent to MAP estimate for Bayesian Regression with Gaussian prior on **w**

l_1 Regularization

Least Absolute Shrinkage and Selection Operator - LASSO

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

- Helps in feature selection – favors sparse solutions
- Optimization is not as straightforward as in Ridge regression
  - Gradient not defined for $w_i = 0, \forall i$
- Equivalent to MAP estimate for Bayesian Regression with *Laplace* prior on **w**

Laplace Distribution

$$p(\mathbf{w}) = \frac{1}{2b} \exp\left(-\frac{|\mathbf{w} - \boldsymbol{\mu}|}{b}\right)$$

- Has two parameters, $\boldsymbol{\mu}$ and b
- Has a less "fatter" tail than Gaussian

LASSO vs. Ridge

- Both control overfitting
- Ridge helps reduce impact of correlated inputs, LASSO helps in feature selection

Elastic Net Regularization

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) + \lambda|\mathbf{w}| + (1-\lambda)\|\mathbf{w}\|_2^2$$

- The best of both worlds
- Again, optimizing for $\mathbf{w}$ is not straightforward

Impact of outliers on regression

- Linear regression training gets impacted by the presence of outliers
- The square term in the exponent of the Gaussian pdf is the culprit
  - Equivalent to the square term in the loss
- How to handle this (*Robust Regression*)?
- Probabilistic:
  - Use a different distribution instead of Gaussian for $p(y|\mathbf{x})$
  - Robust regression uses Laplace distribution

$$p(y|\mathbf{x}) \sim Laplace(\mathbf{w}^\top \mathbf{x}, b)$$

- Geometric:
  - *Least absolute deviations* instead of least squares

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

References

Murphy Book Chapter 11