NATURAL LANGUAGE PROCESSING (PRACTICE)

NLP 242 - Lab 6: Linear - Logistic Regression

Department of Computer Science and Engineering
Ho Chi Minh University of Technology, VNU-HCM

Linear Regression

Supervise Learning

Supervised Learning

- The training data consists of observations (*examples, observations*), where each observation is *associated with a desired output value.*
- The goal is to learn a function (e.g., a classifier, a regression function, etc.) that fits the given dataset and generalizes well.
- The learned function is then used to make predictions for new observations.
- *Classification*: If the output (y) belongs to a finite and discrete set.
- *Regression*: If the output (y) is a real number.

Basic set-up for supervised learning

- **Data:** $(X_1, Y_1), \ldots, (X_n, Y_n)$, where $X_i \in \mathcal{X},\ Y_i \in \mathcal{Y}$
  - In most slides, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$.
- **Loss:** $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty]$
- **Assumption:** $(X_1, Y_1), \ldots, (X_n, Y_n)$ are i.i.d. copies of (X, Y).
- **Risk:** $R(f) = \mathbb{E}\ell(Y, f(X))$
- **Goal:** To estimate a function minimizing the risk.

Empirical: We study the behavior of a minimizer

$$\hat{\theta} = \arg\min_{\theta \in \Theta} L(\theta),$$

where Θ is a (possibly constrained) parameter space. We hope the prediction error $\mathbb{E}\ell(Y, f_{\hat{\theta}}(\mathbf{X}))$ is small enough.

Regression

Empirical Regression Problem: The goal is to learn a function $y = f(x)$ from a given training set:

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$

such that $y_i \approx f(x_i)$ for all i.

- Each observation is represented as an D-dimensional vector, for example:

$$x_i = (x_{i1}, \ldots, x_{iD})^T$$

- Each dimension represents an attribute (feature).

Regression Problem:

- For squared error loss $\ell(y, y') = (y - y')^2$, The regression function, defined as $f_0(\mathbf{x}) = \mathbb{E}(\, Y \mid \mathbf{X} = \mathbf{x})$, minimizes the risk.
- In this sense, the regression model is often written as:

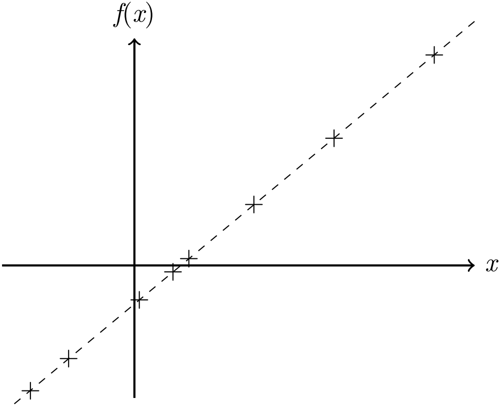
$$Y = f(\mathbf{X}) + \epsilon, \quad \mathbb{E}(\, Y \mid \mathbf{X}) = 0.$$

Linear Regression: Introduction

Linear Model: If the hypothesis function $y = f(x)$ is linear, it has the form:

$$f(x) = b + w_1 x_1 + \cdots + w_D x_D$$

b is called the **bias** term. Learning a linear regression function is equivalent to learning the weight: $w = (b, w_1, \ldots, w_D)^T$



x	**y**
0.13	-0.91
1.02	-0.17
3.17	1.61
-2.76	-3.31
1.44	0.18
5.28	3.36
-1.74	-2.46
7.93	5.56
...	...

Empirical Risk Minimization

- A standard strategy for estimating f_0 is the empirical risk minimization:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2$$

  - For squared error loss, the minimizer is called the least square estimator.
- Note that the original goal was to minimize the population loss:

$$\underbrace{\mathbb{E}(Y - f(\mathbf{X}))^2}_{\text{population loss}} \approx \underbrace{\frac{1}{n} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2}_{\text{empirical loss}}$$

Empirical Loss Function

- We only observe a dataset $(\mathcal{X}, \mathcal{Y})$:

$$(\mathcal{X}, \mathcal{Y}) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

  The goal is to learn a function f from $(\mathcal{X}, \mathcal{Y})$.
- **Empirical Loss** (residual sum of squares, RSS):

$$RSS(f) = \sum_{i=1}^{M} (y_i - f(x_i))^2 = \sum_{i=1}^{M} (y_i - w_0 - w_1 x_{i1} - \cdots - w_n x_{in})^2$$

  - RSS/M is an approximation of $\mathbb{E}_x[r(x)]$ over the training set $(\mathcal{X}, \mathcal{Y})$.
- The term:

$$\left| \frac{1}{M} RSS(f) - \mathbb{E}_x[r(x)] \right|$$

  is often referred to as the **generalization error** of function f.
- Many learning methods are typically associated with RSS.

Ordinary Least Squares (OLS)

Given D, we seek the function f that minimizes the RSS.

$$f^* = \arg\min_{f \in H} RSS(f) \Leftrightarrow w^* = \arg\min_{w} \sum_{i=1}^{M} (y_i - w_0 - w_1 x_{i1} - \cdots - w_n x_{in})^2$$

This is called the least squares method: Find the solution w^* by taking the derivative of RSS and solving the equation RSS' $= 0$. We obtain:

$$w^* = (A^T A)^{-1} A^T y$$

Here, A is a data matrix of size $M \times (n+1)$ where the i-th row is $A_i = (1, x_{i1}, x_{i2}, \ldots, x_{in})$; B^{-1} is the inverse matrix; $y = (y_1, y_2, \ldots, y_M)^T$.

Note: The hypothesis that $A^T A$ has an inverse.

Regulization

Ridge Regression (L2 Regularization): This adds an L2 penalty to shrink the coefficients

$$\hat{W} = \arg\min_{W} \sum_{i=1}^{n} (y_i - X_i W_i)^2 + \lambda \sum_{j=1}^{p} W_j^2$$

Lasso Regression (L1 Regularization): This adds an L1 penalty, which can shrink some coefficients to zero.

$$\hat{W} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - X_i W_i)^2 + \lambda \sum_{j=1}^{p} |W_j|$$

Would you like an explanation of when to use each one?

Logistic Regression

Objective function

For a data point: $\langle x_n, y_n \rangle$. The predicted probability is:

$$p(y_n|x_n, w) = \begin{cases} \hat{y}_n & \text{if } y_n = 1 \\ 1 - \hat{y}_n & \text{if } y_n = 0 \end{cases}$$

In compact form:

$$p(y_n|x_n, w) = \hat{y}_n^{y_n}(1 - \hat{y}_n)^{1-y_n} \tag{2.1}$$

The probability of observing N labels in the training set:

$$p(t|X, w) = \prod_{n=1}^{N} \hat{y}_n^{y_n}(1 - \hat{y}_n)^{1-y_n} \tag{2.2}$$

$\Rightarrow$ **Find** w **such that** $p(t|X, w)$ **is maximized.**

Maximum Likelihood Estimation

Using **negative log-likelihood**:

$$\mathcal{L}(w) \triangleq -p(t|X, w) = \sum_{n=1}^{N} y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)$$

Objective: Find w such that $L(w)$[1] is minimized. We have

$$\nabla_w \log(\mathcal{L}(w)) = -\left(\frac{y_i - \hat{y}_i}{\hat{y}_i(1 - \hat{y}_i)}\right)(\nabla_w \hat{y}_i) = \frac{y_i - \hat{y}_i}{\hat{y}_i(1 - \hat{y}_i)}(\nabla_w \hat{y}_i)$$

Let $\hat{y}_i = f(wx)$ and $s = wx$, we have

$$\nabla_w \hat{y}_i = (\nabla_w s)\frac{\partial \hat{y}_i}{\partial s} = \frac{\partial \hat{y}_i}{\partial s}$$

Choose f such that $\frac{\partial \hat{y}_i}{\partial s} = \hat{y}_i(1 - \hat{y}_i)$ so $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$.

Chain rule method

Principle: It is difficult to use mathematical analysis to find a solution for the optimization problem with the objective function.

The problem of minimizing $L(w)$ is an unconstrained optimization problem, and iterative methods can be used.

- Based on the first derivative: Gradient Descent
- Based on the second derivative: Newton-Raphson

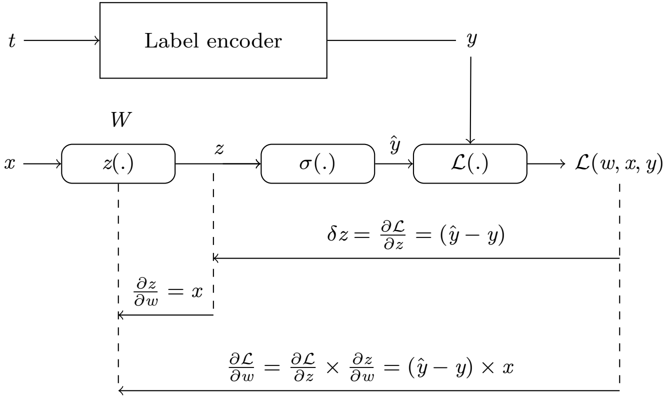
Necessary: Find the derivative of the loss function L with respect to the parameters w.

$$\Delta w = \frac{\partial L(w; x, y)}{\partial w}$$

Use chain-rule method:

$$\frac{\partial \mathcal{L}(w; x, y)}{\partial w} = \frac{\partial \mathcal{L}}{\partial z} \cdot \frac{\partial z}{\partial w}$$

Estimate Model Parameters



Parameter Computation and Update Process

Estimate Model Parameters

Derivatives of some functions in the computational diagram

$$\frac{d\mathcal{L}(w;x,y)}{dy} = \frac{\hat{y}-y}{\hat{y}(1-\hat{y})} \qquad \frac{d\hat{y}}{dz} = \hat{y}(1-\hat{y}) \quad \frac{\partial z}{\partial w} = x$$

The derivative of the function $\mathcal{L}(w;x,y)$ computed on a data point $< x, y >$ is:

$$\Delta w = \frac{\partial L(w;x,y)}{\partial w} = \frac{dL(w;x,y)}{d\hat{y}} \times \frac{d\hat{y}}{dz} \times \frac{\partial z}{\partial w} = (\hat{y}-y)x$$

THANKS FOR LISTENING!