Statistical Learning

Linear Models for Regression

Spring 2024

Outline

- Linear Regression Review
- Training vs. Testing Errors
- Model Selection Criteria and Algorithms

Linear Models for Regression

Regression Models

- Observe a collection of n i.i.d. training samples

$$\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$$

where x_i is a p dimensional vector (predictors, covariates, features, inputs)

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

and $y_i \in \mathbb{R}$ is a continuous response (outcome, output).

- We assume the underlying model $Y = f(X) + \epsilon$
- Estimate f using $\widehat{f}$

Notation

- $\mathbf{x}_j$ is a n dimensional vector of the jth feature, i.e.,

$$\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^{\mathsf{T}}$$

- The design matrix $\mathbf{X}$ is $n \times p$:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$$

- $\mathbf{x}_j$ is one column in $\mathbf{X}$

Loss and Risk

- What is a good model fitting?
- Loss function, risk, and empirical risk
- A loss function L measures the discrepancy between Y and any function $f(X)$
- Risk is the expected loss over the entire population

$$\mathsf{R}(f) = \mathsf{E}\left[L\big(Y, f(X)\big)\right]$$

- The true function $f(x)$ would minimize this risk.

Loss and Risk

- In regression, the squared error loss is commonly used:

$$L\big(Y, f(X)\big) = \big(Y - f(X)\big)^2$$
$$\mathsf{R}(f) = \mathsf{E}\left[\big(Y - f(X)\big)^2\right]$$

- Other examples for regression: Huber loss
- For classification: 0/1, logistic, hinge, etc.

The Empirical Risk

- With the training data $\mathcal{D}_n$, estimate $f(x)$ by minimizing the empirical risk

$$\mathsf{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

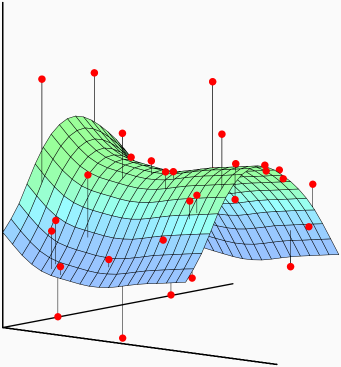
$$\widehat{f}(x) = \underset{f \in \mathcal{F}}{\arg\min} \quad \mathsf{R}_n(f)$$

  where $\mathcal{F}$ is some space of models

- Using the squared error loss for regression, we have

$$\widehat{f} = \underset{f \in \mathcal{F}}{\arg\min} \quad \frac{1}{n}\sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2.$$

Regression



Estimating $f(X)$, figure from ESL

Linear Regression

- A linear regression model assumes a functional form of f

$$f(X) = X^{\mathsf{T}}\boldsymbol{\beta}$$

- Note: set $X_1 = 1$ as the intercept term.
- We express the regression problem in the matrix form

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1} + \mathbf{e}_{n\times 1}$$

Linear Regression

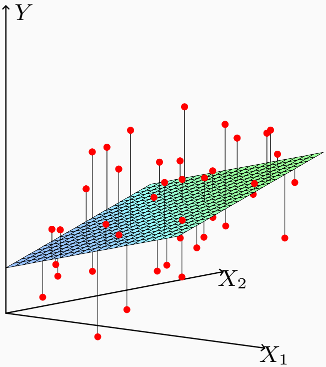
- Solve $\boldsymbol{\beta}$ by minimizing the residual sum of squares (RSS)

$$\begin{aligned}\text{RSS} =& \sum_{i=1}^{n} \Big(y_i - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p \Big)^2 \\ =& \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ =& \big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)^{\mathsf{T}} \big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)\end{aligned}$$

- The ordinary least squares estimator (OLS) is

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \quad \big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)^{\mathsf{T}} \big(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)$$

Linear Regression



Solving linear regression, figure from ESL

Estimate β

- To estimate $\boldsymbol{\beta}$, we set the derivative equal to 0

$$\begin{aligned}\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} =& -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0 \\ \implies \quad \mathbf{X}^\top\mathbf{y} =& \mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}\end{aligned}$$

  which is the normal equation.

- We then have, if $\mathbf{X}^\top\mathbf{X}$ is invertible,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}.$$

- $\mathbf{X}$ full rank $\iff$ $\mathbf{X}^\top\mathbf{X}$ invertible

Hat Matrix

- The fitted values (i.e., prediction at observed x_i's) are

$$\widehat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \doteq \mathbf{H}_{n\times n}\mathbf{y}$$

- $\mathbf{H}$ ("hat matrix") is a project matrix
  - symmetric: $\mathbf{H}^\intercal = \mathbf{H}$
  - idempotent: $\mathbf{H}\mathbf{H} = \mathbf{H}$
- The residual $\mathbf{r}_{n\times 1} = \widehat{\mathbf{e}} = \mathbf{y} - \widehat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$
- $\mathbf{r}$ can be used to estimate the error variance

$$\widehat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} r_i^2 = \frac{\text{RSS}}{n-p}$$

Hat Matrix

- The essential machinery of linear regression is projection
- Decompose the outcome vector $\mathbf{y}$ into two orthogonal vectors

$$\mathbf{y} = \widehat{\mathbf{y}} + \mathbf{r}$$

- $\widehat{\mathbf{y}}$ lives in the column space of $\mathbf{X}$, since $\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}$
- $\mathbf{r}$ is orthogonal to $\mathbf{X}$, i.e., $\mathbf{X}^{\mathsf{T}}\mathbf{r} = \mathbf{0}$

Vector Space Interpretation

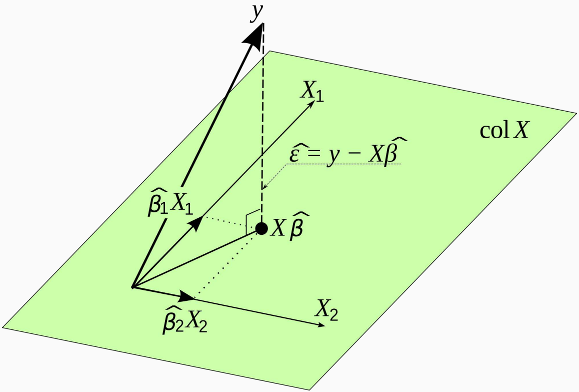


Figure from Wiki

Properties of $\widehat{\beta}$

- We assume that the samples are generated from the model

$$Y = X^{\mathsf{T}}\boldsymbol{\beta} + \epsilon,$$

  where the errors ϵ_i are i.i.d. with $\mathsf{E}(\epsilon_i) = 0$ and $\mathsf{Var}(\epsilon_i) = \sigma^2$
- Then $\widehat{\boldsymbol{\beta}}$ is unbiased: $\mathsf{E}(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$
- Variance-covariance

$$\mathsf{Var}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\sigma^2$$

- σ^2 can be estimated using $\widehat{\sigma}^2$

Properties of $\widehat{\beta}$

- Among all unbiased linear estimators, $\widehat{\boldsymbol{\beta}}$ has the smallest variance. (Gauss-Markov Theorem)

- An unbiased linear estimator is defined as

$$\widehat{\boldsymbol{\beta}} = \mathbf{A}\mathbf{y}, \text{ and } \mathrm{E}(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$$

- Further assuming ϵ is normal, $\widehat{\boldsymbol{\beta}}$ is also UMVUE

- Question: What if we have a biased estimator? Can we trade a little bias for a large reduction in variance?

Training vs. Testing Errors

Dealing with large p

- In many applications nowadays, we have many explanatory variables, i.e., p is large or even $p \gg n$.
  - There are more than 20,000 human protein-coding genes
  - About 10 million single nucleotide polymorphisms (SNPs)
  - Number of subjects, n, is usually in hundreds or thousands
- In some applications, the key question is to identify a subset of X variables that are most relevant to Y
- Let's examine the training and testing errors from a linear model

Training vs. Testing error

- Training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$
- Suppose $\{x_i, y_i^*\}_{i=1}^n$ is an independent (imaginary) testing dataset collected at the same location x_i's (aka, in-sample prediction)
- Assume that the data are indeed from a linear model

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$
$$\mathbf{y}^* = \boldsymbol{\mu} + \mathbf{e}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}^*$$

  where both $\mathbf{y}$ and $\mathbf{y}^*$ are $n \times 1$ response vectors, $\mathbf{e}$ and $\mathbf{e}^*$ are i.i.d. error terms with mean 0 and variance σ^2.

- The true model is indeed linear: $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$

Training vs. Testing error

$$
\begin{aligned}
\mathsf{E}[\textsf{Test Err}] &= \mathsf{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 \\
&= \mathsf{E}\|(\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|^2 \\
&= \mathsf{E}\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\
&= \mathsf{E}\|\mathbf{e}^*\|^2 + \mathsf{Trace}\big(\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathsf{Cov}(\widehat{\boldsymbol{\beta}})\big) \\
&= n\sigma^2 + p\sigma^2
\end{aligned}
$$

Training vs. Testing error

$$\begin{aligned}
\mathsf{E}[\text{Train Err}] &= \mathsf{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{y}\|^2 \\
&= \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \text{Trace}\big((\mathbf{I} - \mathbf{H})^{\mathsf{T}}(\mathbf{I} - \mathbf{H})\text{Cov}(\mathbf{e})\big) \\
&=(n - p)\sigma^2
\end{aligned}$$

- Hence, the testing error increases with p and training error decreases with p. As p gets larger, this could be a big trouble...

Variable Selection

- It might be necessary to select a set of most relevant variables, especially when p is large.
- Variable selection may improve
  - Prediction accuracy
  - Interpretability
- This is a difficult task
  - No natural ordering of importance for the variables
  - The role of a variable needs be measured conditioning on others, high correlation causes trouble
  - It is essential to check all possible combinations, however, this may be computationally expensive

Model Selection Criteria and Algorithms

Model Selection Criteria

- Model selection is usually done in the following way
  1) Give each model a score
  2) Design an algorithm to find the model with the best (smallest) score
- The score of a model fitting takes the form

Goodness-of-fit + Complexity-Penalty

1) The first term decreases as the model gets more complicated (recall 1NN)
2) The second term increases with the number of predictor variables (recall degrees of freedom), which prefers "smaller" model

Model Selection Criteria

- Popular choices of scores:
  - Mallows' C_p (Mallows, 1973): RSS + $2\widehat{\sigma}^2_{\text{full}} \cdot p$
  - AIC (Akaike, 1970): -2 Log-likelihood $+ 2 \cdot p$
  - BIC (Schwarz, 1978): -2 Log-likelihood $+ \log n \cdot p$
- When n is large, adding an additional predictor costs a lot more in BIC than AIC (or C_p). So AIC tends to pick a larger model than BIC.
- C_p performs similarly to AIC.

Justification of Mallows' C_p

- Recall our previous analysis of the training and testing errors with $\mathbf{y}$ and $\mathbf{y}^*$
- Now, lets assume that the model is not necessarily a linear model, i.e.,

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e}$$
$$\mathbf{y}^* = \boldsymbol{\mu} + \mathbf{e}^*$$

- We assume mean 0 and variance σ^2 for the two error vectors, but we don't have $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$. However, we still perform linear regression regardless. This will introduce bias of the estimations.

Justification of Mallows' C_p

$$\begin{aligned}
\mathsf{E}[\text{Test Err}] &= \mathsf{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 = \|\mathbf{y}^* - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathsf{E}\|(\mathbf{y}^* - \boldsymbol{\mu}) + (\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}) + (\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y})\|^2 \\
&= \mathsf{E}\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + \mathsf{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathsf{E}\|\mathbf{e}^*\|^2 + \mathsf{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{H}\mathbf{e}\|^2 \\
&= n\sigma^2 + \text{Bias}^2 + p\sigma^2
\end{aligned}$$

$$\begin{aligned}
\mathsf{E}[\text{Train Err}] &= \mathsf{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathsf{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu} + (\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \mathsf{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu}\|^2 + \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \text{Bias}^2 + (n - p)\sigma^2
\end{aligned}$$

Hence, Test Err is approximately Train Err $+ 2\sigma^2 p$, which justifies Mallows' C_p.

Model Selection

- To perform linear model selection, we need to decide on a selection criterion and use an computational algorithm to find the solution
    - Criteria: Mallows' C_p; AIC; BIC
    - Algorithm: Best subset (Brute force); stepwise (forward/backward/...) selection
- Different algorithms may have different advantage

Best Subset Selection

- Best subset selection is a level-wise search algorithm, which returns the global optimal solution for a given model size.
- Only feasible for p not very large (< 50)
- Algorithm:
    1. For each $k = 1, \ldots, p$, check 2^k possible combinations, and find the model with smallest RSS
        - The penalty term is the same for models with the same size
    2. To choose the best k, use model selection criteria

Best Subset Selection

- Note: if RSS$(X_1, X_2) <$ RSS(X_3, X_4, X_5, X_6) then we do not need to visit any size 2 or 3 sub-models of (X_3, X_4, X_5, X_6), which can be leaped over.
- Implemented in R package leaps, using the leaps and bounds algorithm by Furnival and Wilson (1974)

Stepwise Regression

- Greedy algorithms: fast, but only return a local optimal solution (which might be good enough in practice).
  - Backward: start with the full model and sequentially delete predictors until the score does not improve.
  - Forward: start with the null model and sequentially add predictors until the score does not improve.
  - Stepwise: consider both deleting and adding one predictor at each stage.