Lecture 1: Linear methods for statistical learning

Minwoo Chae

Department of Industrial and Management Engineering
Pohang University of Science and Technology

KMS-NIMS Summer School on AI, 2025

Outline

1. Introduction
2. Statistical decision theory
3. Linear regression
4. Linear logistic regression

Basic set-up

- Input variables: $\mathbf{X} = (X_1, \ldots, X_p)^T$
  - Covariates, predictors, features, independent variables
  - $\mathcal{X}$-valued ($\mathcal{X} \subset \mathbb{R}^p$) random variable
- Output variables: Y
  - Responses, dependent variables
  - $\mathcal{Y}$-valued random variable
- Data: $(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)$
- Goal: To predict outputs using inputs for unobserved future data
  - Regression: Continuous response
  - Classification: Discrete response

Notations

- Vectors are bold, except (some) Greeks.
  - $\mathbf{x}, \mathbf{y}, \mathbf{z}$
  - θ, β
- Vector-valued functions are bold.
  - $\mathbf{f}(\cdot), \mathbf{g}(\cdot)$
- Scalars and scalar-valued functions are non-bold.
- Vectors are column vectors by default.
  - Row vectors: $\mathbf{x}^\top, \mathbf{y}^\top, \mathbf{z}^\top$
- Probability distributions are upper cases: P, Q
  - The corresponding densities are lower cases: p, q

Goal

- For a future input $\mathbf{X}$, we aim to predict the corresponding response value.
- There exist many possible predictors $\hat{f}_k : \mathcal{X} \to \mathcal{Y}$, which may depend on the observed data.
- A key objective is to select (or develop) an effective predictor.
- To do so, we must first establish appropriate criteria for good prediction.

Outline

1. Introduction

2. Statistical decision theory

3. Linear regression

4. Linear logistic regression

Statistical decision theory

- Let $L : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ be a loss function.
- A smaller loss is better, but we also want to account for the uncertainty in the data.
- To this end, we assume that each observation is i.i.d. from an unknown distribution.
- Given a loss function L and a function $f : \mathcal{X} \to \mathcal{Y}$, define

  $$R(f) = \mathbb{E}[L(Y, f(\mathbf{X}))]$$

  as the risk associated with f.
  - Often referred to as the prediction (generalization, test) error.

Statistical decision theory (cont.)

- For regression, the most commonly used loss is the squared error loss:

$$L(Y, f(\mathbf{X})) = [Y - f(\mathbf{X})]^2$$

- For classification, the most commonly used loss is the 0-1 loss:

$$L(Y, C(\mathbf{X})) = I(Y \neq C(\mathbf{X}))$$

- The primary objective of supervised learning is to minimize the prediction error.
- In many cases, the minimizer of the risk over all measurable functions can be characterized explicitly.

Statistical decision theory: regression

THEOREM For the squared error loss,

$$f_0(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$$

minimizes the risk $R(f)$.

DEFINITION f_0 is called the regression function.

- Note that f_0 is an unknown function because the joint distribution of $(Y, \mathbf{X})$ is not known.
- The primary goal of regression is to estimate f_0.

Statistical decision theory: classification

THEOREM For the 0-1 loss with $\mathcal{Y} = \{1, \ldots, K\}$,

$$C_0(\mathbf{x}) = \operatorname*{argmax}_{k \in \mathcal{Y}} \mathbb{P}(k \mid \mathbf{X} = \mathbf{x})$$

minimizes the prediction error.

DEFINITION C_0 is called the Bayes classifier, and $R(C_0)$ is called the Bayes error rate.

- The primary goal of classification is to estimate the Bayes classifier.

Other loss functions

- Regression
  - $L(Y, f(\mathbf{X})) = |Y - f(\mathbf{X})|$ (absolute error loss)
  - $L(Y, f(\mathbf{X})) = (Y - f(\mathbf{X})) \cdot (\tau - I(Y < f(\mathbf{X})))$ for $\tau \in (0, 1)$ (check loss for quantile regression)
- Classification
  - Asymmetric 0-1 loss for binary classification
  - Surrogate losses for 0-1 loss (e.g., logistic loss, hinge loss)

Surrogate losses for 0-1 loss

- In practice, directly minimizing the 0-1 loss is computationally challenging due to its non-convexity and non-differentiability.
- Instead, one can use convex surrogate losses that are easier to optimize.
- Although we employ surrogate losses, our ultimate goal remains to estimate the Bayes classifier C_0.
- In most cases, we do not work with the classifier $C : \mathcal{X} \to \mathcal{Y}$ directly.

Surrogate losses for 0-1 loss (cont.)

- In classification, estimating the Bayes classifier is closely related to estimating the conditional probability function

$$\mathbf{p}_0(\mathbf{x}) = \big(p_{0,1}(\mathbf{x}), \ldots, p_{0,K}(\mathbf{x})\big)^\top,$$

where

$$p_{0,k}(\mathbf{x}) = \mathbb{P}(Y = k \mid \mathbf{X} = \mathbf{x}).$$

- Once an estimator $\hat{\mathbf{p}}$ is given, one can easily define a classifier as

$$\hat{C}(\mathbf{x}) = \operatorname*{argmax}_k \hat{p}_k(\mathbf{x}).$$

Surrogate losses for 0-1 loss (cont.)

- Directly working with the conditional probability $\mathbf{p}(\cdot)$ is also often challenging due to the sum-to-one constraint.
- Therefore, a transformation of $\mathbf{p}(\cdot)$ is often considered in practice.
- Accordingly, surrogate losses are typically defined in terms of the conditional probability function or its transformation.
- To accommodate this, we allow the second argument of the loss function to take values in $\overline{\mathcal{Y}}$, which is not necessarily equal to $\mathcal{Y}$.

Log odds in binary classification

- In binary classification with $\mathcal{Y} = \{+1, -1\}$, the scalar function

  $$p_0(\mathbf{x}) = \mathbb{P}(Y = +1 \mid \mathbf{X} = \mathbf{x})$$

  fully determines the vector-valued function $\mathbf{p}_0(\cdot)$.
- In most applications, we work with its logit transform:

  $$f_0(\mathbf{x}) = \log \frac{\mathbb{P}(Y = +1 \mid \mathbf{X} = \mathbf{x})}{\mathbb{P}(Y = -1 \mid \mathbf{X} = \mathbf{x})} = \log \frac{p_0(\mathbf{x})}{1 - p_0(\mathbf{x})}.$$

Logistic loss for binary classification

- For binary classification with $\mathcal{Y} = \{+1, -1\}$, the logistic (also known as binomial or cross-entropy) loss is defined as

$$L(y, f(\mathbf{x})) = \log\big(1 + \exp(-yf(\mathbf{x}))\big).$$

- One can show that the population minimizer of $R(f)$ is f_0.
- Therefore, minimizing the logistic loss at the population level leads to the Bayes classifier, since

$$C_0(\mathbf{x}) = \operatorname{sign}(f_0(\mathbf{x})).$$

Remarks

- The logistic loss is essentially the negative binomial log-likelihood, up to an additive constant.
- The target parameter corresponds to the log-odds.
- Depending on the coding or parametrization, the expression may vary slightly.
- Other surrogate loss functions for binary classification that yield the Bayes classifier include the hinge loss (SVM) and the exponential loss (AdaBoost).
- The logistic loss is the most convenient for extension to multi-class classification.

Cross-entropy

- For two probability vectors $\mathbf{p} = (p_1, \ldots, p_K)^\top$ and $\mathbf{q} = (q_1, \ldots, q_K)^\top$, the cross-entropy is defined as

$$H(\mathbf{p}, \mathbf{q}) = -\sum_{k=1}^{K} p_k \log q_k.$$

- One can show that the function $\mathbf{q} \mapsto H(\mathbf{p}, \mathbf{q})$ is minimized at $\mathbf{p}$.

Cross-entropy loss for multi-class classification

- In multi-class classification with $\mathcal{Y} = \{1, \ldots, K\}$, the cross-entropy loss is defined as

$$L(Y, \mathbf{p}(\mathbf{X})) = -\mathbf{Y}^\top \log\big(\mathbf{p}(\mathbf{X})\big),$$

  where $\mathbf{Y}$ is the one-hot encoded response vector, $\mathbf{p} : \mathcal{X} \to \Delta^{K-1}$, and the logarithm is applied componentwise.
- The cross-entropy loss is essentially the negative multinomial log-likelihood, up to an additive constant.
- One can show that the population risk minimizer is $\mathbf{p}_0(\cdot)$.
- Therefore, minimizing the population risk yields the Bayes classifier C_0.

Cross-entropy loss for multi-class classification (cont.)

- As in binary classification, the cross-entropy loss can also be defined in terms of an arbitrary function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^K$ as

$$L(Y, \mathbf{f}(\mathbf{X})) = -\mathbf{Y}^\top \log\big(\sigma(\mathbf{f}(\mathbf{X}))\big),$$

where $\sigma : \mathbb{R}^K \to \Delta^{K-1}$ is the softmax function defined by

$$\sigma(\mathbf{x}) = \left(\frac{e^{x_1}}{\sum_{k=1}^K e^{x_k}}, \ldots, \frac{e^{x_K}}{\sum_{k=1}^K e^{x_k}} \right)^\top .$$

Remarks

- Given a loss function, a central goal in supervised learning is to minimize

$$R(f) = \mathbb{E}[L(Y, f(\mathbf{X}))]$$

  over all measurable functions $f : \mathcal{X} \to \overline{\mathcal{Y}}$.
- For certain loss functions, the learning problem is equivalent to estimating a specific target function:
  - Squared error loss: regression function
  - 0-1 loss: Bayes classifier
- Although the population distribution of $(Y, \mathbf{X})$ is unknown, we are given samples $(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)$.

Empirical risk minimization

- There are several approaches to achieving the goal of supervised learning.
- Some of them are tailored to specific loss functions.
- We introduce a general approach called empirical risk minimization, which can be applied to a wide range of loss functions.
- For a given function f, the population risk $R(f)$ can be estimated by the empirical risk:

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, f(\mathbf{X}_i)).$$

Empirical risk minimization (cont.)

- For a given class $\mathcal{F}$ of functions, we expect that the empirical risk minimizer

  $$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} R_n(f)$$

  is close to the population risk minimizer

  $$f_* = \operatorname*{argmin}_{f \in \mathcal{F}} R(f)$$

  over $\mathcal{F}$.
- Note that f_* may differ from f_0, the population risk minimizer over all measurable functions.

Empirical risk minimization (cont.)

- In practice, there are two key components of empirical risk minimization:
  - The choice of a model class $\mathcal{F}$ for the function f.
  - The choice of an optimization algorithm.
- In this lecture, we focus on the choice of the model class $\mathcal{F}$.
- Optimization itself is a vast field of study.

Remarks

- While not exactly the same, minimizing the population risk $R(f)$ (over all measurable functions) is closely related to estimating f_0.
- In the regression setting, we have

$$\begin{aligned} R(f) &= \mathbb{E}\big(Y - f(\mathbf{X})\big)^2 \\ &= \underbrace{\mathbb{E}\big(f(\mathbf{X}) - f_0(\mathbf{X})\big)^2}_{\text{deviation of } f \text{ from } f_0} + \underbrace{\mathbb{E}\big(f_0(\mathbf{X}) - Y\big)^2}_{\text{intrinsic noise}} \end{aligned}$$

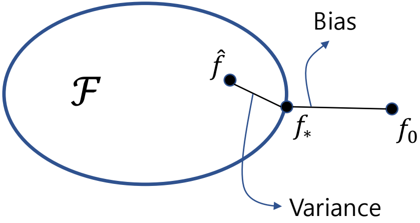
- Consequently, the performance of supervised learning methods is often evaluated using the norm $\|\hat{f} - f_0\|$.

Bias-variance tradeoff

- To analyze the risk of the empirical risk minimizer $\hat{f}$ over $\mathcal{F}$, we can decompose the error $\|\hat{f} - f_0\|$ as

$$\|\hat{f} - f_0\| \leq \underbrace{\|f_* - f_0\|}_{\substack{\text{Bias} \\ \text{(approximation error)}}} + \underbrace{\|\hat{f} - f_*\|}_{\substack{\text{Variance} \\ \text{(estimation error)}}} .$$

- Typically, the amount of bias and variance depends on the complexity of $\mathcal{F}$.
  - This is known as the bias-variance tradeoff.



Complexity of statistical models

- There are various ways to quantify model complexity.
  - Number of (nonzero) parameters
  - Metric entropy
  - Vapnik–Chervonenkis (VC) dimension
  - Rademacher complexity
- Quantifying model complexity and estimation error lies at the heart of empirical process theory.
- Technically, the key is to bound the empirical process

$$\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(\mathbf{X}_i)) - R(f) \right| .$$

van der Vaart, A. W. & Wellner, J. A. *Weak Convergence and Empirical Processes.* (Springer, 1996)

Giné, E. & Nickl, R. *Mathematical Foundations of Infinite-Dimensional Statistical Models.* (Cambridge University Press, 2016)

Remarks

- In general, the approximation error decreases and the estimation error increases as the complexity of $\mathcal{F}$ increases.
- Overly complex models often lead to overfitting.
- Balancing approximation and estimation errors is key to successful supervised learning.
  - In practice, optimization is another key component.

k-NN for regression

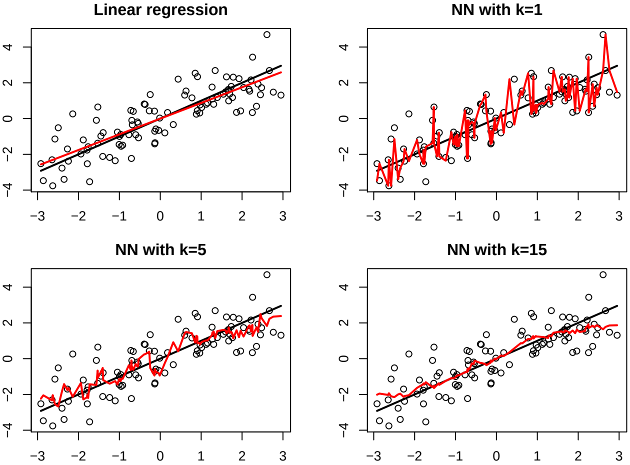
- Consider the problem of estimating the regression function $f_0(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$.
- A simple estimator is

$$\hat{f}(\mathbf{x}) = \frac{1}{k} \sum_{i:\mathbf{X}_i \in N_k(\mathbf{x})} y_i,$$

  where $N_k(\mathbf{x})$ denotes the neighborhood of $\mathbf{x}$ consisting of the k closest points $\mathbf{X}_i$ in the training sample.
- This is known as the k-nearest neighbor (k-NN) estimator.

k-NN for regression (cont.)

$$f_0(x) = x$$



Linear regression

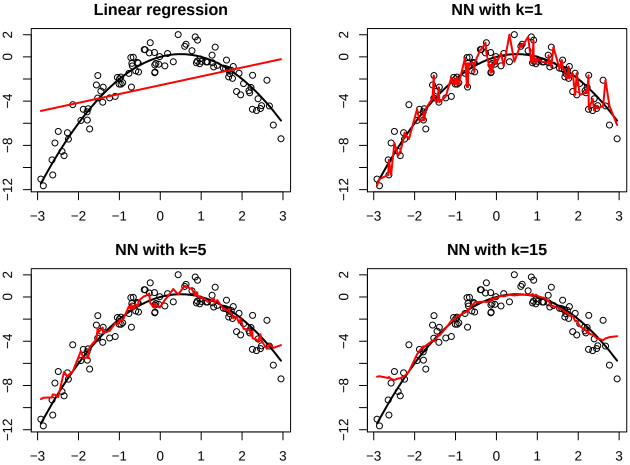
NN with k=1

NN with k=5

NN with k=15

k-NN for regression (cont.)

$$f_0(x) = x(1 - x)$$



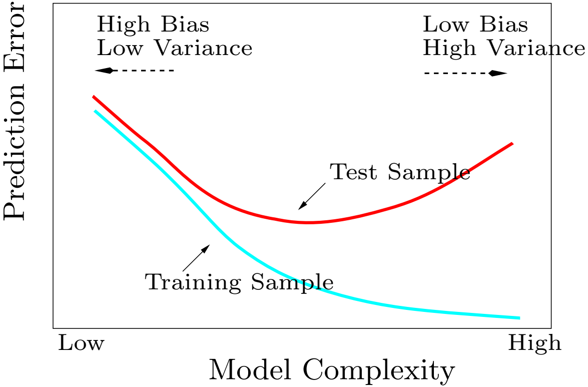
Remarks

- Linear models perform well when the true relationship is linear, but poorly when it is nonlinear.
- The k-NN method tends to perform reasonably well regardless of the true underlying model.
- The choice of neighborhood size k plays a role analogous to the complexity of $\mathcal{F}$ in empirical risk minimization.
- More complex models typically achieve smaller training error,

$$R(\hat{f}) = \sum_{i=1}^{n} L(Y_i, \hat{f}(\mathbf{X}_i)),$$

but may not generalize well.

Illustration of bias-variance decomposition



Curse of dimensionality

- The k-NN approach is quite reasonable regardless of the true underlying model.
- However, this intuition breaks down in high dimensions.
- This phenomenon is known as the curse of dimensionality.

Curse of dimensionality (cont.)

EXAMPLE Let **X** be uniformly distributed on $[0, 1]^p$.

- Consider a hypercubic neighborhood around a target point.
- Suppose we want to capture a fraction r of the sample.
- Then, the expected edge length of the cube is $e_p(r) = r^{1/p}$.
- For example, $e_{10}(0.01) = 0.63$ and $e_{10}(0.1) = 0.83$.
- To capture 1% or 10% of the data for local averaging, we must cover 63% or 83% of the range of each input variable.
- Such neighborhoods are no longer truly "local."

Curse of dimensionality (cont.)

EXAMPLE Let $\mathbf{X}$ be uniformly distributed on the unit ball in $\mathbb{R}^p$.

- For i.i.d. copies $\mathbf{X}_1, \ldots, \mathbf{X}_n$ of $\mathbf{X}$, let $R_i = \|\mathbf{X}_i\|$.
- Then, the median of $\min\{R_1, \ldots, R_n\}$ is

$$\left(1 - \frac{1}{2^{1/n}}\right)^{1/p}.$$

- For $n = 5000$ and $p = 10$, the median is approximately 0.52.
- That is, most data points lie closer to the boundary of the sample space than to the origin.
- Prediction near the boundary is difficult, as one must extrapolate rather than interpolate.

Remarks

- The k-NN method can also be applied to classification problems.
- In ERM, the choice of the function class $\mathcal{F}$ and the optimization algorithm are key components.
- Frequently used models:
  - $\mathcal{F} = \{\text{Linear functions}\}$
  - $\mathcal{F} = \{\text{Bases expansions}\}$
  - $\mathcal{F} = \{\text{Ensemble of trees}\}$
  - $\mathcal{F} = \{\text{Neural networks}\}$
- In both k-NN and ERM, careful model selection is crucial.
- For high-dimensional inputs, mitigating the curse of dimensionality is another important challenge.

Outline

1. Introduction
2. Statistical decision theory
3. Linear regression
4. Linear logistic regression

Linear regression via ERM

- Linear regression refers to ERM with squared error loss and the linear model

$$\mathcal{F} = \left\{ f_\beta : f_\beta(\mathbf{x}) = \beta^\top \mathbf{x}, \;\; \beta \in \mathbb{R}^p \right\}.$$

- That is, the function f modeling the regression function is parameterized by a Euclidean vector β.
- Here, we implicitly assume that the first component of $\mathbf{x}$ is 1, so that the intercept term is included in the regression coefficient β.

Least square estimator

- The empirical risk in linear regression can be written as

$$R_n(f_\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^\top \mathbf{X}_i)^2.$$

- The empirical risk minimizer

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} R_n(\beta)$$

is called the least squares estimator (LSE).

Least square estimator (cont.)

- Since the empirical risk is quadratic in β, $\hat{\beta}$ can be obtained in closed form.
- For example, if the design matrix $\mathsf{X} \in \mathbb{R}^{n \times p}$ has full rank, then

  $$\hat{\beta} = (\mathsf{X}^\top \mathsf{X})^{-1} \mathsf{X}^\top \mathbf{Y},$$

  where $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$ is the response vector.
- For a given new observation $\mathbf{X}$, one may predict the response as $\beta^\top \mathbf{X}$.

Pros and cons of linear models

- Pros
  - Simple and interpretable
  - Convex objective function
  - Enables various forms of statistical inference
  - Building blocks for more complicated models
- Cons
  - Risk of model misspecification
  - Sensitivity to outliers
  - High variance in high-dimensional settings

Remarks

- Linear models tend to be unstable in high-dimensional settings.
- This instability can arise even when the model is well-specified.
- A major reason is the high variance of the LSE.
- Note that the variance of $\hat{\beta}$ is proportional to $(\mathsf{X}^\top \mathsf{X})^{-1}$.
- Several remedies have been proposed to address this issue:
  - Subset (variable) selection
  - Shrinkage methods
  - Dimension reduction techniques

Best subset selection

- For a given $k \leq p$, choose k input variables such that the residual mean squared error is minimized among all models with k predictors.
- Denote this model by $\mathcal{M}_k$.
- Then, select the optimal model from among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$.
- If p is large (e.g., $p \geq 40$), this approach becomes computationally infeasible.
- Common alternatives include forward and backward stepwise selection.

Forward selection

- Start with the model $\mathcal{M}_0$ containing only the intercept.
- Construct a sequence of models $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ by sequentially adding the predictor that most improves the fit at each step.
- Select the best model from among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$.

Backward selection

- Start with the full model $\mathcal{M}_p$ including all predictors.
- Construct a sequence of models $\mathcal{M}_p, \mathcal{M}_{p-1}, \ldots, \mathcal{M}_0$ by sequentially removing the predictor that has the least impact on the fit at each step.
- Select the best model from among $\mathcal{M}_p, \mathcal{M}_{p-1}, \ldots, \mathcal{M}_0$.

Remarks

- Variable selection methods are often unstable and may lead to suboptimal predictive performance.
- Penalized/constrained least squares methods have become more popular for the last few decades.
- The key idea behind penalized least squares is to shrink the LSE toward the origin, thereby substantially reducing its variance.

Penalized/constrained least squares method

- The least squares method can be viewed as ERM with squared error loss and a linear model:

$$\mathcal{F} = \Big\{ f_\beta : f_\beta(\mathbf{x}) = \beta^\top \mathbf{x},\ \ \beta \in \mathbb{R}^p \Big\}.$$

- Penalized/constrained least squares methods consider a different model:

$$\mathcal{F} = \Big\{ f_\beta : f_\beta(\mathbf{x}) = \beta^\top \mathbf{x},\ \ J(\beta) \le t \Big\},$$

where $J(\cdot)$ is a suitable penalty function.

Penalized/constrained least squares method (cont.)

- In other words, the penalized (or constrained) least squares method solves

$$\underset{\beta:\,J(\beta)\le t}{\operatorname{minimize}}\, R_n(f_\beta).$$

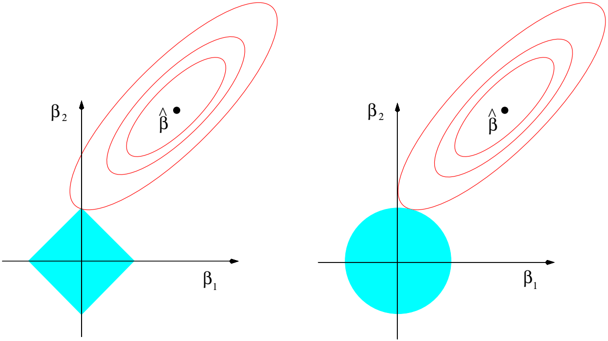
- An equivalent formulation is

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{minimize}}\left\{R_n(f_\beta) + \lambda\, J(\beta)\right\}.$$

Penalized/constrained least squares method (cont.)

- Two popular choices:
  - $J(\beta) = \|\beta\|_2^2$ (ridge regression)
  - $J(\beta) = \|\beta\|_1$ (lasso regression)
- Both penalties lead to shrinkage toward the origin.
- Remarkably, lasso tends to produce sparse solutions.
- Since the lasso penalty is non-differentiable, standard numerical optimization techniques do not apply.

Penalized/constrained least squares method (cont.)



Remarks

- Compared to classical parametric models, several new phenomena arise in high-dimensional settings.
- There is a vast literature on penalized least squares approaches.
- Developing feasible methods for statistical inference in high-dimensional linear models remains an active area of research.

Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity.* (CRC Press, 2015)

Bühlmann, P. & van de Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications.* (Springer, 2011)

Wainwright, M. J. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint.* (Cambridge University Press, 2019)

Outline

1. Introduction

2. Statistical decision theory

3. Linear regression

4. Linear logistic regression

Binary logistic regression

- Linear regression serves as a foundational model for regression tasks.
- In classification, the (linear) logistic model plays a similar foundational role.
- We begin with binary classification.
- Two common codings for Y:
  - $\mathcal{Y} = \{0, 1\}$
  - $\mathcal{Y} = \{-1, +1\}$
- There is no fundamental difference, but certain methods may prefer a specific coding for mathematical convenience.

Binary logistic regression (cont.)

- Binary classification (under either coding) is closely related to estimating the conditional probability

$$p_0(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}).$$

- A simple approach to modeling the unknown function p_0 is to use linear models.
- However, since p_0 is a probability taking values in $[0, 1]$, a linear model is not appropriate.
- A natural remedy is to transform $[0, 1]$-valued probabilities to real-valued scores in $\mathbb{R}$.

Binary logistic regression (cont.)

- The logit function is a bijection from $(0, 1)$ to $\mathbb{R}$:

$$\mathrm{Logit}(p) = \log \frac{p}{1 - p}$$

- The logistic function is its inverse:

$$\mathrm{Logistic}(x) = \frac{e^x}{1 + e^x}$$

Binary logistic regression (cont.)

- In the linear logistic model, the logit of the conditional class probability is modeled as a linear function:

$$\operatorname{logit}\big(p(\mathbf{x})\big) = \log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} = \beta^\top \mathbf{x},$$

  where $p(\mathbf{x}) = \Pr(Y = 1 \mid \mathbf{X} = \mathbf{x})$.

- Equivalently,

$$p(\mathbf{x}) = \operatorname{logistic}(\beta^\top \mathbf{x}) = \frac{\exp(\beta^\top \mathbf{x})}{1 + \exp(\beta^\top \mathbf{x})}.$$

Binary logistic regression (cont.)

- Given data, the regression coefficient can be estimated via maximum likelihood:

$$\hat{\beta} = \operatorname*{argmax}_{\beta \in \mathbb{R}^p} \prod_{i=1}^{n} \Pr(Y_i \mid \mathbf{X}_i).$$

- The maximum likelihood approach can be interpreted as empirical risk minimization with the negative log-likelihood as the loss function.

Binary logistic regression (cont.)

- Ignoring the additive constant, the loss function under $\{0, 1\}$ coding can be written as

$$L(y, f_\beta(\mathbf{x})) = -y f_\beta(\mathbf{x}) + \log\big(1 + \exp(f_\beta(\mathbf{x}))\big),$$

  where $f_\beta(\mathbf{x}) = \beta^\top \mathbf{x}$.

- Under $\{-1, +1\}$ coding, the loss function becomes

$$L(y, f_\beta(\mathbf{x})) = \log\big(1 + \exp(-y f_\beta(\mathbf{x}))\big).$$

Extension to high dimensions

- When p is large, similar issues arise in logistic regression as in linear regression.
- As with linear regression, shrinkage approaches are effective in addressing these issues.
- Formally, one may consider the penalized negative log-likelihood estimator:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} L(Y_i, f_\beta(\mathbf{X}_i)) + \lambda J(\beta),$$

where $J(\cdot)$ is a suitable penalty function.

Multi-class logistic regression

- Consider multi-class classification with $\mathcal{Y} = \{1, \ldots, K\}$.
- As in binary logistic regression, we model the conditional class probabilities as

$$\Pr(Y = k \mid \mathbf{X} = \mathbf{x}) = \frac{\exp(\beta_k^\top \mathbf{x})}{\sum_{j=1}^{K} \exp(\beta_j^\top \mathbf{x})}.$$

- This can be succinctly written as

$$\Pr(Y = k \mid \mathbf{X} = \mathbf{x}) \propto \exp(\beta_k^\top \mathbf{x}).$$

- For identifiability, it is common to set $\beta_K = \mathbf{0}$.

Multi-class logistic regression (cont.)

- Parameters can be estimated via maximum likelihood, which corresponds to ERM with the cross-entropy loss:

  $$L(Y, \mathbf{f}_{\boldsymbol{\beta}}(\mathbf{X})) = -\mathbf{Y}^\top \log\big(\sigma(\mathbf{f}_{\boldsymbol{\beta}}(\mathbf{X}))\big),$$

  where $\mathbf{Y}$ is the one-hot encoded response, σ is the softmax function, $\boldsymbol{\beta} = (\beta_k)_{k=1}^K$, and

  $$\mathbf{f}_{\boldsymbol{\beta}}(\mathbf{x}) = (\beta_1^\top \mathbf{x}, \ldots, \beta_K^\top \mathbf{x})^\top.$$

Remarks

- Linear regression and logistic regression serve as fundamental building blocks for more complex models in regression and classification, respectively.
- Therefore, a deep understanding of these basic models is essential for understanding more advanced methods.

Thank you for attention!