Professor Han Liu Scribe: Amy Hua

Contents

1 Introduction

Four years ago, ORF 350 was co-created with Professor Erhan Cinlar, founder of the ORFE Department at Princeton. We don't want to teach technicians or the next generation of software engineers. We want to teach leaders—the next generation of CEOs. COS 424 teaches methods. ORF 350 teaches math and theory.

This course isn't about covering as much as possible; rather, it is about understanding concepts right from the start. Eliminate what you already know, and relearn in this class—you will get a deeper, more general understanding. Enjoy the course!

1.1 Logistics

Course staff:

- Instructor: Professor Han Liu
- TAs: Ziwei Zhu (GS) and Jian Ge (GS)
- Undergraduate Designer: Xiaoyan Han (ORF '16)

Grading:

- Participation (20%)
- Midterm (10%)
- Final (10%)
- Homework assignments (60%)

1.2 Overview of course topics

1. Fundamental theory
2. Regression
3. Classification
4. Clustering
5. Dimensionality reduction

1.3 Big data analytics

Analysis flows as follows: Data $\rightarrow$ Model $\rightarrow$ Decision $\rightarrow$ Value. We ask: how do we use the data, model, and decision to achieve value (what we really care about)? Modern decision making is decision making by explicitly modeling uncertainty and incompleteness.

We take a dual perspective approach to analyzing big data—data that is massive (large n), high-dimensional (large d), and/or unstructured.

Perspective	Statistics	Machine Learning
Foundation	Likelihood principle	Concentration principle (LLN)
Goal	Inference	Prediction
Approach	Likelihood maximization (model-based)	Risk minimization (model-free)

Likelihood maximization and risk minimization are essentially equivalent: we can view likelihood as the negative of risk.

1.4 Fundamental principles for big data analytics

Definition 1.1. (Likelihood principle). Everything is model-based, allowing us to derive theory (such as asymptotic theory).

Definition 1.2. (Concentration principle). In the big data regime, the sample size is big.

$$\text{Data} = \text{Signal} + \text{Noise}.$$

Data: random samples $X_1, ..., X_n$.
Signal: θ. Ex: $N(\theta, 1)$.
Noise: uncertainty from nature. Ex: $\varepsilon \sim N(0, 1)$.

Inference: We hope $\hat{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$.
Concentration phenomenon (LLN): as $n \to \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{P} E(X).$$

Remark 1.3. We need the data to have some stationary pattern to "sum out" noise/uncertainty. Note that stationary does not necessarily have to be i.i.d. (stationary includes i.i.d., autoregressive model, etc.)

Definition 1.4. (Parsimonious principle). In the big data regime, the number of dimensions is big. Intuition: If two explanations are almost equally good, we prefer the simpler one. Thus leads to the **regularization technique**, a very important concept in this course.

Remark 1.5. All the simple models we use are wrong—true models are complex. We always use the "wrong" model to control the possible variability of high dimensions. However, these simple models may still be useful for inference, prediction, etc. Ex: Relation to NLP.

Remark 1.6. Many techniques that are good for big data analytics may not be good for small data.

2 Theoretical foundation

2.1 Basic concepts

Definition 2.1. (Sample space). All possible outcomes of a statistical experiment.

Definition 2.2. (Random sample). Data collected. $X_1, ..., X_n \overset{i.i.d.}{\sim} p(x)$, or $X_{1:n} \sim p(x)$, where $p(x)$ is the density function.

Definition 2.3. (Realizations/observed values/outcomes). $x_1, ..., x_n$, or $x_{1:n}$.

Definition 2.4. (Statistic). Any measurable function of the random samples.

Definition 2.5. (Sample mean).

$$\overline{X}_n := \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Definition 2.6. (Cumulative density function (CDF)). $F(x) := P(X \leq x)$, where X is a random variable.

Definition 2.7. (Probability density function (PDF)). $p(x) := \frac{\partial}{\partial x} F(x)$. $p(x)$ can also be used to represent the probability mass function (pmf) if the random variable X is discrete. We use $\frac{\partial}{\partial x}$ to simplify notation.

Remark 2.8. We use $p_\theta(x)$ to denote that the density function is parameterized by θ.

2.2 Asymptotic theory

Asymptotic theory characterizes what happens as we get more and more data. Key ingredients are the Law of Large Numbers (LLN) and the Central Limit Theorem (CLT).

Definition 2.9. A sequence of random variables $X_1, ..., X_n$, such that

$$X_1, ..., X_n \overset{i.i.d.}{\sim} P,\ E(X_i) = \mu,\ \mathrm{Var}(X_i) = \sigma^2,$$

is said to **converge in probability** to μ if for any $\varepsilon > 0$,

$$\lim_{n \to \infty} P(|\overline{X}_n - \mu| > \varepsilon) = 0.$$

We denote this by

$$\overline{X}_n \xrightarrow{P} \mu.$$

More generally, a sequence of random variables $X_1, ..., X_n$ converges in probability to a random variable X if for any $\varepsilon > 0$,

$$\lim_{n \to \infty} P(|\overline{X}_n - X| > \varepsilon) = 0.$$

We denote this by

$$\overline{X}_n \xrightarrow{P} X.$$

Note that in the definition with μ, X is a random variable that is always μ in value. (The entire sample space consists only of μ.)

Definition 2.10. A sequence of random variables $X_1, ..., X_n$ (with corresponding CDFs F_{X_n}) is said to **converge in distribution** to a random variable X (with CDF F_x) if at all x such that F_X is continuous,

$$\lim_{n\to\infty} F_{X_n}(x) = F_X(x).$$

We denote this by

$$X_n \xrightarrow{D} X.$$

Definition 2.11. (Stochastic convergence). Rather than dealing with scalars (convergence in calculus), we are dealing with random variables $X_1, ..., X_n$.

Remark 2.12. Random variables are random, but the functions themselves are deterministic.

Definition 2.13. (Law of Large Numbers (LLN)). Assume $X_1, ..., X_n \overset{i.i.d.}{\sim} P$ with $E(X_i) = \mu$ and $\text{Var}(X_i) = \sigma^2$. Then, $\overline{X}_n \xrightarrow{P} \mu$.

Definition 2.14. (Central Limit Theorem (CLT)). Assume $X_1, ..., X_n \overset{i.i.d.}{\sim} P$ with $E(X_i) = \mu$ and $\text{Var}(X_i) = \sigma^2$. Then,

$$\sqrt{n}\left(\frac{\overline{X}_n - \mu}{\sigma}\right) \xrightarrow{D} N(0,1).$$

Remark 2.15. $X_n \xrightarrow{P} X \Rightarrow X_n \xrightarrow{D} X$. Convergence in probability is stronger.

Remark 2.16. Note that the CLT is stronger than the LLN. In other words,

$$\sqrt{n}\left(\frac{X_n - X}{\sigma}\right) \xrightarrow{D} N(0,1) \Rightarrow X_n \xrightarrow{P} X.$$

From the CLT, we know that the limit converges, and we also know the rate of convergence ($\sqrt{n}$). The notion of convergence in distribution is very strong here because of the $\sqrt{n}$ term.

Remark 2.17. LLN is useful for estimation. CLT is useful for constructing confidence intervals/p-values.

2.3 Estimation theory

Definition 2.18. (Statistical model). A set of probability distributions indexed by a parameter set Θ.

$$\mathcal{P} := \{p_\theta : \theta \in \Theta\}.$$

Statistical models can be parametric or nonparametric. Note that we use $p_\theta(x)$ to denote that the density function is parameterized by θ.

Definition 2.19. (Parametric model). A statistical model that can be indexed by a finite-dimensional parameter set Θ.

Definition 2.20. (Nonparametric model). A statistical model that cannot be indexed by a finite-dimensional parameter set Θ. Nonparametric models also have parameters, but their parameters are infinite-dimensional.

Example 2.21. (Gaussian model).

$$\mathcal{P} := \left\{ p_{\mu,\sigma^2}(x) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},\ \mu \in \mathbb{R},\ \sigma^2 > 0 \right\}.$$

$\boldsymbol{\theta} := (\mu, \sigma^2)^T \in \Theta := \mathbb{R} \times \mathbb{R}_+$, so this is a parametric model.

Example 2.22. (Sobolev class).

$$\mathcal{P} := \left\{ p(x) \text{ is continuous and } \int p''(x)^2 dx < \infty \right\}.$$

This is a nonparametric model.

Definition 2.23. (Point estimation). Let $X_1, ..., X_n \overset{i.i.d.}{\sim} p_\theta(x)$, where $X_1, ..., X_n$ is our data, and $p_\theta(x)$ is some density function. We want to make a single best guess (estimate) at θ, the unknown parameter. The statistic $\hat{\theta}_n$ (estimator) is defined as $\hat{\theta}_n := g(X_1, ..., X_n)$. We hope $\hat{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$.

Definition 2.24. (Estimate). The value we use to guess the true parameter.

Definition 2.25. (Estimator). A rule for calculating an estimate of a parameter based on the data.

Remark 2.26. The rule is the estimator. The result (or the actual value) is the estimate.

Definition 2.27. (Consistent estimator). An estimator $\hat{\theta}_n$ is consistent if $\hat{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$. As we accumulate more and more data, we converge to the truth.

Definition 2.28. (Unbiased estimator). Define **bias** to be $\text{Bias}(\hat{\theta}_n) := E(\hat{\theta}_n) - \theta$. If $\text{Bias}(\hat{\theta}_n) = 0$ for all n, then we call $\hat{\theta}_n$ unbiased. We don't want $\hat{\theta}_n$ systematically too high or too low.

Remark 2.29. There is no relationship between consistency and unbiasedness: neither consistency nor unbiasedness implies each other.

Example 2.30. (Consistent and unbiased estimator). $X_1, ..., X_n \sim N(\theta, 1)$.

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

By LLN, $\hat{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$, so $\hat{\theta}_n$ is consistent. $E(\hat{\theta}_n) := \frac{1}{n} \sum_{i=1}^{n} E(X_i) = \theta$, so $\hat{\theta}_n$ is unbiased.

Example 2.31. (Consistent and biased estimator). $X_1, ..., X_n \sim N(\theta, 1)$.

$$\tilde{\theta}_n := \hat{\theta}_n + \frac{1}{n}.$$

$\tilde{\theta}_n \xrightarrow{P} \theta$ as $n \to \infty$, so $\tilde{\theta}_n$ is consistent. $E(\tilde{\theta}_n) = E(\hat{\theta}_n) + \frac{1}{n} = \theta + \frac{1}{n}$. $\text{Bias}(\tilde{\theta}_n) = \frac{1}{n} \neq 0$, so $\tilde{\theta}_n$ is biased.

Example 2.32. (Inconsistent and unbiased estimator). $X_1, ..., X_n \sim N(\theta, 1)$.

$$\hat{\theta}_n = X_1.$$

$E(\hat{\theta}_n) = E(X_1) = \theta$, so $\hat{\theta}_n$ is unbiased. $\hat{\theta}_n \nrightarrow \theta$ (does not converge in probability) as $n \to \infty$, so $\hat{\theta}_n$ is not consistent.

Remark 2.33. If an estimator $\hat{\theta}_n$ is consistent, then $\lim_{n\to\infty} E(\hat{\theta}_n) = \theta$. For this reason, we sometimes say that consistent estimators are **asymptotically unbiased**.

Remark 2.34. If we had to choose either consistency or unbiasedness, which would we prefer?

Argument for unbiasedness: Classical statistics. Practically, there is only finite data, so we cannot observe when $n \to \infty$. The classical regime emphasizes unbiasedness.

Argument for consistency: Modern statistics. In the big data regime, we can observe $n \to \infty$, so we care more about consistency.

Often, we need to sacrifice unbiasedness to achieve convergence/consistency—a tradeoff when it comes to classical vs. modern statistics.

2.4 Maximum likelihood estimation

Definition 2.35. (Likelihood). The likelihood function of θ related to a random sample X_i is

$$\mathcal{L}(X_i, \theta) := p_\theta(X_i).$$

The likelihood function is just a density function evaluated at stochastic values. Although $\mathcal{L}$ is a function of X_i and θ, we typically keep X_i fixed and think of $\mathcal{L}$ as a function of θ. Nonetheless, it is still a random quantity because X_i is a random variable.

Remark 2.36. All models are random, but parameters are deterministic. This concept of random vs. deterministic is important to keep in mind.

Definition 2.37. (Joint likelihood). The joint likelihood of θ with respect to the entire dataset of random samples $X_1, ..., X_n$ is defined as

$$\mathcal{L}_n(\theta) := p_\theta(X_1, ..., X_n).$$

Note that this definition involves a general joint distribution of the random samples. In the special case when the samples are i.i.d. following distribution p_θ, we have

$$\mathcal{L}_n(\theta) = \prod_{i=1}^{n} p_\theta(X_i).$$

Definition 2.38. (Joint log-likelihood). The joint log-likelihood of θ with respect to $X_1, ..., X_n$ is

$$\ell_n(\theta) := \log[L_n(\theta)].$$

Again, if the samples are i.i.d., then

$$\ell_n(\theta) = \sum_{i=1}^{n} \log(p_\theta(X_i)).$$

Because the logarithm is an increasing function,

$$\arg\max_{\theta\in\Theta} \mathcal{L}_n(\theta) = \arg\max_{\theta\in\Theta} \ell_n(\theta).$$

Remark 2.39. Note that in this class, by log, we mean the natural log, or ln. We can safely assume that every reference to log is a reference to ln.

Definition 2.40. (Maximum likelihood estimator (MLE)). $\hat{\theta}_n$ is MLE if $\mathcal{L}_n(\hat{\theta}_n) \geq \mathcal{L}_n(\theta)$ for all $\theta \in \Theta$. MLE does not have to be unique (though in most cases will be) and works for parametric and nonparametric models. If $\mathcal{L}_n(\theta)$ has a unique maximizer $\hat{\theta}_n$ over $\theta \in \Theta$, then we write

$$\hat{\theta}_n := \arg\max_{\theta\in\Theta} \mathcal{L}_n(\theta).$$

Example 2.41. (MLE of Gaussian distribution). $X_1, ..., X_n \overset{i.i.d.}{\sim} N(\mu, 1)$.

$$\begin{aligned}
\mathcal{L}_n(\mu) &= \prod_{i=1}^{n} p_\mu(X_i) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-(X_i-\mu)^2/2} \\
&= \frac{1}{(2\pi)^{n/2}} e^{-\sum_{i=1}^{n}(X_i-\mu)^2/2}.
\end{aligned}$$

The MLE is

$$\hat{\mu}_n = \arg\max_{\mu} \mathcal{L}_n(\mu) = \frac{1}{n}\sum_{i=1}^{n} X_i =: \overline{X}_n,$$

which is the sample mean.

Question 2.42. Why MLE? MLE provides a "unified" and "systematic" framework to obtain a "good" statistical estimator.

- "Unified": Once we have a statistical model, we can define the likelihood and maximize it.
- "Systematic": We can write a joint density function and throw it into an optimization solver—this is useful in the big data regime.
- "Good": MLE is justified by an important theorem to be discussed in the next theorem.

Definition 2.43. (Fisher information). Given a statistical model $\{p_\theta : \theta \in \Theta\}$ indexed by θ such that $\log p_\theta(x)$ is twice differentiable with respect to θ, the Fisher information is defined as

$$I(\theta) := -E\left(\frac{\partial^2}{\partial\theta^2}\log p_\theta(X)\right) = -\int \left[\frac{\partial^2}{\partial\theta^2}\log p_\theta(x)\right] p_\theta(x)dx.$$

If θ is a scalar, then $I^{-1}(\theta) = \frac{1}{I(\theta)}$. If θ is a vector, then $I^{-1}(\theta)$ is a matrix.

Theorem 2.44. *MLE is **asymptotically normal** and "**efficient**."*
Let θ be the true parameter. Under certain conditions, the MLE is asymptotically normal, i.e.,

$$\sqrt{n}(\hat{\theta}_n^{MLE} - \theta) \xrightarrow{D} N(0, I^{-1}(\theta)).$$

For any unbiased estimator $\tilde{\theta}_n$, $Var(\tilde{\theta}_n) \geq I^{-1}(\theta)$ (MLE is "good"/"efficient"). In other words, the variance of any unbiased estimator is at least as large as the variance of the MLE.

Remark 2.45. This theorem is a better theorem than the Gauss-Markov Theorem, which is outdated and less general/powerful.

Application 2.46. ($(1-\alpha)\%$ asymptotic confidence interval of θ).

$$C_n := \left[\hat{\theta}_n - z_{\alpha/2} \cdot \frac{I^{-1/2}(\hat{\theta}_n)}{\sqrt{n}},\ \hat{\theta}_n + z_{\alpha/2} \cdot \frac{I^{-1/2}(\hat{\theta}_n)}{\sqrt{n}}\right],$$

where $\alpha/2$ represents the $\alpha/2$th quantile of a standard Gaussian distribution. Then we have

$$\lim_{n\to\infty} P(\theta \in C_n) \geq 1 - \alpha$$

for any $\theta \in \Theta$. (When involving confidence intervals, we only consider the univariate case.)

Remark 2.47. In the big data regime, t-value and z-value are interchangeable.

3 Predictive analytics (supervised learning)

3.1 Introduction

Now we move from classical statistics to modern analytics. We revisit our original flow of analysis—Data → Model → Decision → Value—to flesh it out. Our revised flow to represent modern decision making is:
Business/scientific plan $\xrightarrow{Acquisition}$ Raw data (text/image/numeric) $\xrightarrow{Feature\ engineering}$ Analytic data → Statistics (inference) and Machine learning (prediction) → Knowledge and information $\xrightarrow{Intervention}$ (revisit) Business/scientific plan. This is the iterative process of modern decision making.

Definition 3.1. (Predictive analytics). A set of techniques that analyze current and historical data to make predictions about the future. (Ex: Regression analysis is a type of predictive analytics.)

Remark 3.2. A typical predictive analysis process is as follows: Given a generic paradigm or training data $(Y_1, X_1), ..., (Y_n, X_n)$, build a prediction function $\hat{f}$, and then given a new observation X, predict $\hat{Y} := \hat{f}(X)$.

Example 3.3. Image classification.

Remark 3.4. Two learning tasks are:

1. **Prediction**: Given a new X, predict Y.
2. **Variable selection**: Find a small subset of predictors $X_1, ..., X_d$ that have the most predictive power or are "most related" to Y.

3.1.1 Notation

Remark 3.5. Some notes on notation:

- n to index samples: $X_1, ..., X_n$.
- d to index dimensions: $\mathbf{X} = (X_1, ..., X_d)^T,\ \mathbf{X} \in \mathbb{R}^d$.
- Population version: $Y, \mathbf{X} \sim P_{Y,\mathbf{X}}$. Here, $Y \in \mathbb{R},\ \mathbf{X} \in \mathbb{R}^d$. Y and $\mathbf{X}$ are population variables.
- Sample version: We observe $(Y_1, \mathbf{X_1}), ..., (Y_n, \mathbf{X_n}) \sim P_{Y,\mathbf{X}}$. Note: $\mathbf{X_1}, ..., \mathbf{X_n}$ are i.i.d. with $\mathbf{X}$.
- We denote the **response vector** as $\mathbf{Y} = (Y_1, ..., Y_n)^T$.
- We denote the **design matrix** as

$$\mathbb{X} = \begin{pmatrix} X_{11} & \cdots & X_{1d} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nd} \end{pmatrix} \in \mathbb{R}^{n \times d}.$$

Example 3.6.

- $Z \sim N(\theta, 1)$ is the **population version**. We never observe this.
- $Z_1, .., Z_n \sim N(\theta, 1)$ is the **(random) sample version**. We actually observe this.
- $z_1, ..., z_n$ are the **realizations**—these are deterministic values. We only use this notation in the last step when plugging in.

Remark 3.7. In this class, we almost always use capital letters/stochastic random variables, not lowercase letters/deterministic realizations.

3.1.2 Statistics / inference

With statistical inference, there is an *underlying population* that has some population variable X with parameter(s) we wish to estimate. (For instance, we could analyze the population of individuals with HIV and look at blood pressure as a population variable.) We take random samples from the population, moving from the "population world" to the "sample world". (In our example, random samples might be the blood pressure of 100 individuals from the population we are examining.) Note that these samples need not be i.i.d. (for example, time series data involves dependent X_i). The values that these random samples take on are referred to as realizations. (As soon as we collect the numerical data from the 100 individuals, we transition from random samples to realizations.)

Definition 3.8. (Inference—statistics). We aim to infer the population quantities based on random samples. $\hat{\theta}_n(X_1, ..., X_n) \xrightarrow{P} \theta$.

3.1.3 Machine learning / prediction

Definition 3.9. (Prediction—machine learning). Based on $X_1, ..., X_n$, predict X_{n+1}.

Question 3.10. Is prediction ($X_1, ..., X_n \to X_{n+1}$) statistical inference?
No. We are not assuming that there is some underlying population. But, note that we can use statistical inference to address the prediction problem by assuming an underlying population: $X_1, ..., X_n$ (sample world) $\to N(\theta, 1)$ (population world) $\to X_{n+1}$. (Ex: Online machine learning.)

Remark 3.11. Understand the difference between inference and prediction, between statistics and machine learning.

3.2 Regression analysis

Definition 3.12. (Regression). The act of summarizing the relationship between two variables Y and $\mathbf{X}$. Y is the response/outcome. $\mathbf{X}$ is the vector of predictors/features/covariates. (Note that $\mathbf{X}$ can have $d \geq 1$ elements to represent d features.)
Population world: $Y, \mathbf{X} \sim P_{Y,\mathbf{X}}$. Sample world: $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n) \sim P_{Y,\mathbf{X}}$,
where $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$ are the observed random samples (assume i.i.d.) and n is the sample size. We aim to find a mapping/function f such that $f(\mathbf{X})$ is "close" to Y.

Question 3.13. $f(\mathbf{X})$ and Y are stochastic. How should we quantify their "closeness?" Some examples of **loss functions** are:

- (L_1-loss). $l(f(\mathbf{X}), Y) := |f(\mathbf{X}) - Y|$.
- (L_2-loss). $l(f(\mathbf{X}), Y) := |f(\mathbf{X}) - Y|^2$.

We take the expectation of the loss functions to find the **risk functions**:

- (L_1-risk). $E_{\mathbf{X},Y}|f(\mathbf{X}) - Y|$.
- (L_2-risk). $E_{\mathbf{X},Y}|f(\mathbf{X}) - Y|^2$. (Least squares criterion—used in this class.)
- (Empirical L_2-risk). $\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2$.
- (Risk measure). $P[|f(\mathbf{X}) - Y| > \epsilon]$, $E_{\mathbf{X},Y}|f(\mathbf{X}) - Y|^2 + P(f)$.

Remark 3.14. In real-world applications, people mainly use L_2-risk. Why do we prefer to use L_2-/least squares risk?

1. Solves the problems well.
2. Mathematically simple.
3. Computationally simple.
4. Statistically justifiable as the MLE under a Gaussian noise model.
5. By Taylor expansion, all smooth loss functions are locally quadratic.

Still, L_2-risk might not always be the best—there are many different options. Ex: When dealing with a lot of noise, we may prefer L_1-risk.

Definition 3.15. (L_2-risk). $R(f) := E|Y - f(\mathbf{X})|^2$. $R(f)$ is the L_2-risk of f. $E|Y - f(\mathbf{X})|^2$ is the **least squares criterion**.

Definition 3.16. (Regression/mean function). $f^* = \arg\min_f R(f)$.

f^* is in the population world, so we use samples to infer f^*.

Theorem 3.17. *Let $f^* := \arg\min_f E|Y - f(\mathbf{X})|^2$. Then, $f^*(x) = E(Y|\mathbf{X} = \mathbf{x})$ is the regression function, or mean function, that minimizes L_2-risk. Note that this regression function is only for L_2-risk.*

Proof. Let $\overline{f}(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$. Then,

$$\begin{aligned}
E|Y - f(\mathbf{X})|^2 &= E|Y - \overline{f}(\mathbf{X}) + \overline{f}(\mathbf{X}) - f(\mathbf{X})|^2 \\
&= E|Y - \overline{f}(\mathbf{X})|^2 + E|\overline{f}(\mathbf{X}) - f(\mathbf{X})|^2 + 2E\left[(Y - \overline{f}(\mathbf{X}))(\overline{f}(\mathbf{X}) - f(\mathbf{X}))\right] \\
&= E|Y - \overline{f}(\mathbf{X})|^2 + E|\overline{f}(\mathbf{X}) - f(\mathbf{X})|^2 + 2E_X\left[E[(Y - \overline{f}(\mathbf{X}))(\overline{f}(\mathbf{X}) - f(\mathbf{X}))]|\mathbf{X}\right] \\
&= E|Y - \overline{f}(\mathbf{X})|^2 + E|\overline{f}(\mathbf{X}) - f(\mathbf{X})|^2 + 2E_X\left[(\overline{f}(\mathbf{X}) - f(\mathbf{X}))E[(Y - \overline{f}(\mathbf{X}))|\mathbf{X}]\right] \\
&= E|Y - \overline{f}(\mathbf{X})|^2 + E|\overline{f}(\mathbf{X}) - f(\mathbf{X})|^2.
\end{aligned}$$

Thus, since $E|Y - \overline{f}(\mathbf{X})|^2$ does not depend on f, we minimize $E|Y - f(\mathbf{X})|^2$ by minimizing $E|\overline{f}(\mathbf{X}) - f(\mathbf{X})|^2$.

$$E|\overline{f}(\mathbf{X}) - f^*(\mathbf{X})|^2 = 0 \Longrightarrow f^*(\mathbf{X}) = \overline{f}(\mathbf{X}).$$

□

Remark 3.18. The goal of L_2-regression (regression utilizing L_2-risk) is to estimate $f(x) = E(Y|\mathbf{X} = \mathbf{x})$. Recall that $Y, \mathbf{X} \sim P_{Y,\mathbf{X}}$ (population world).

Remark 3.19. In the population world, L_2-regression aims to find f that minimizes

$$R(f) := E((Y - f(\mathbf{X}))^2),\ Y, \mathbf{X} \sim P,$$

where $R(f)$ is the **population (true) risk**.

Definition 3.20. (Empirical risk).

$$\mathcal{D} = \{(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)\},$$

where $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$ are random samples. The empirical risk is given by

$$\hat{R}(f) := \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2.$$

By the concentration principle,

$$\hat{R}(f) \xrightarrow{P} R(f).$$

Remark 3.21.

$$\hat{f} := \arg\min_f \hat{R}(f).$$

Minimizing $\hat{R}(f)$ without further assumptions restricting our solution space is problematic, as any function f that satisfies $f(\mathbf{x}) = Y_i$ for every $\mathbf{x} = \mathbf{X_i}$ will minimize $\hat{R}$, regardless of how the function acts outside of $\mathbf{X}_1, ..., \mathbf{X}_n$ (on unobserved data).

Definition 3.22. (Overfitting). A phenomenon when a statistical model has too much flexibility (or too many degrees of freedom or parameters) that the model starts to fit the noise, instead of just the signal. If we entirely fit noise to training data, for example, we cannot generalize our model to work well on testing data.

Ex: Fishing with a net that is too dense and gets everything. We only want the fish and need to filter out the "noise."

Our solution to overfitting: regularization.

Definition 3.23. (Regularization). Introduction of additional information or constraints to reduce the flexibility (or capacity) of the model.

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}(f),$$

where $\mathcal{F}$ is the solution space.

1. Linear model:
$$\mathcal{F} = \{f(x) : f(x) = \beta^T \mathbf{x}\}.$$

2. Polynomial regression (nonlinear model):
$$\mathcal{F} = \{f(x) : f(x) = \text{Poly}(\mathbf{x}, r)\}.$$

3. Nonparametric model:
$$\mathcal{F} = \{f(x) : \int f''(x)^2 dx < \infty\}.$$
(Sobolev space.)

3.2.1 Ordinary least squares (OLS) regression

Definition 3.24. (Ordinary least squares (OLS)).
Population space: $Y \in \mathbb{R}$, $\mathbf{X} \in \mathbb{R}^d$, $\mathbb{X} = (X_1, ..., X_d)$.
Sample space: $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$. $\mathbf{Y} := (Y_1, ..., Y_n)^T$. $\mathbb{X} := [X_{i,j}] \in \mathbb{R}^{n \times d}$, where the first column of $\mathbb{X}$ is $(1, ..., 1)^T$.

If we define $||\beta||_2 = \sqrt{\beta^T \beta}$, we have that the OLS estimator is

$$\begin{aligned}\hat{\beta}^{\text{OLS}} &:= \arg\min_\beta ||\mathbf{Y} - \mathbb{X}\beta||_2^2 \\ &= \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n (Y_i - \beta^T \mathbf{X}_i)^2,\end{aligned}$$

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

where

$$\mathcal{F} = \left\{ f(x) : f(x) = \beta^T \mathbf{x} \right\}.$$

Remark 3.25. Let $F(\beta) := ||Y - \mathbb{X}\beta||_2^2 = \mathbf{Y}^T\mathbf{Y} + \beta^T\mathbb{X}^T\mathbb{X}\beta - 2\beta^T\mathbb{X}^T\mathbf{Y}$. Then the gradient of F is

$$\frac{\partial F(\beta)}{\partial \beta} = 2\mathbb{X}^T\mathbb{X}\beta - 2\mathbb{X}^T\mathbf{Y} = 0,$$

and

$$\hat{\beta}^{\text{OLS}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{Y},$$

assuming that $d < n$ and that $\mathbb{X}^T\mathbb{X} \in \mathbb{R}^{d \times d}$ is invertible.

Theorem 3.26. *(Model-based interpretation of OLS).* $\hat{\beta}^{OLS}$ *is the MLE under the* ***Gaussian noise model***

$$Y = \beta^T\mathbf{X} + \varepsilon,\ \varepsilon \sim N(0, \sigma^2),\ E(\varepsilon|\mathbf{X}) = 0.$$

Proof. We assume that

$$P(Y, \mathbf{X}) = P(Y|\mathbf{X})P(\mathbf{X}),$$

where $Y|\mathbf{X} \sim N(\beta^T\mathbf{X}, \sigma^2)$ and $\mathbf{X} \sim P_{\mathbf{X}}$, where $P_{\mathbf{X}}$ is an arbitrary distribution.

The log-likelihood is

$$\begin{aligned} l_n(\beta, \sigma^2) &= \sum_{i=1}^{n} \log p_{\beta,\sigma^2}(Y_i|\mathbf{X}_i) \\ &= \sum_{i=1}^{n} \log p_{\beta,\sigma^2}(Y_i|\mathbf{X}_i) + \sum_{i=1}^{n} \log p(\mathbf{X}_i). \end{aligned}$$

Note that $p(\mathbf{X}_i)$ does not depend on β or σ^2 since the model conditions upon $\mathbf{X}_i$. Thus, we do not need to worry about this term when optimizing $l_n(\beta, \sigma^2)$.

Thus, the MLE is

$$\begin{aligned} \hat{\beta}^{\text{MLE}} &= \arg\max_{\beta,\sigma^2} l_n(\beta, \sigma^2) \\ &= \arg\max_{\beta,\sigma^2} \sum_{i=1}^{n} \log p_{\beta,\sigma^2}(Y_i|\mathbf{X}_i) \\ &= \arg\max_{\beta,\sigma^2} \left[-\frac{1}{2\sigma^2} \left(\sum_{i=1}^{n} (Y_i - \beta^T\mathbf{X}_i)^2 \right) - n \log \sqrt{2\pi\sigma^2} \right] \\ &= \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - \beta^T\mathbf{X}_i)^2. \end{aligned}$$

Note that since $n \log \sqrt{2\pi\sigma^2}$ does not involve β, we do not worry about this term when analyzing $\hat{\beta}^{\text{OLS}}$. □

Question 3.27. Why do we use the model-based interpretation?

- We can construct confidence intervals, obtain p-values, analyze statistical significance.
- A statistical model tells us a generative story that makes it easier to incorporate prior information.
- Bayesian inference. We cannot do Bayesian statistics without a model.

  $$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta),$$

  where $p(\mathcal{D}|\theta)$ is model-based and $p(\theta)$ is prior information.

3.2.2 Linear regression with basis expansion

Remark 3.28. Almost all regression analysis can be viewed as a form of linear regression. **Feature engineering** allows us to incorporate nonlinearity.

1. Inputs can be transformations of original features (hand-crafted features). Examples:
   - $X_1' \leftarrow \log X_1$.
   - $X_1' \leftarrow \sqrt{X_1}$.
   - $X_1' \leftarrow X_1^2$.
2. Inputs can have interaction terms. To $X_1, ..., X_d$, we can add $X_1X_2, X_1X_3, ..., X_{d-1}X_d$.
3. Inputs can have basis expansions. Instead of using

   $$f(\mathbf{X}) = \sum_{j=1}^{d} \beta_j X_j,$$

   we consider

   $$f(\mathbf{X}) = \sum_{j=1}^{p} \beta_j h_j(\mathbf{X}),$$

   for some arbitrary p, where $h_j(\mathbf{X})$ is the generated basis function. We can move from parametric to nonparametric for a suitable choice of $h_1, ..., h_p$. Ex: Neural networks.
4. Inputs can be indicator functions of qualitative inputs. Ex: I(X_j lies in City A). This leads to categorical data analysis (very useful in real applications).

3.2.3 Categorical data analysis

Definition 3.29. (Categorical variable). A variable that can take on one value of a limited set of values. Ex: $X \in \{M, F\}$.

Remark 3.30. How do we convert categorical variables to numerical variables? Note that there is no order relationship between these variables. Solution: Dummy coding. Ex: 1 if F, 0 if M. More generally, if a categorical variable has k categories $X \in \{C_1, ..., C_k\}$, then we use $k-1$ **dummy variables**.
Ex: k cities.

$$\text{City 1: } (0,0,0,...,0)$$
$$\text{City 2: } (1,0,0,...,0)$$
$$\vdots$$
$$\text{City } k\text{: } (0,0,0,...,1)$$

3.3 High-dimensional data analysis

Definition 3.31. (High-dimensional data). Data with "a lot of" features. In the regression setting, $d > n$. (If $d \leq n$, we call this "low-dimensional" data.)

Question 3.32. When $d > n$, what will happen to OLS?

$$\hat{\beta}^{\text{OLS}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{Y}.$$

Since $d > n$ and $\mathbb{X} \in \mathbb{R}^{n\times d}$, $\text{rank}(\mathbb{X}^T\mathbb{X}) \leq n$, so $\mathbb{X}^T\mathbb{X}$ is rank-deficient and not invertible. This leads to overfitting: the model is too big/has too many features, so the system $\mathbf{Y} = \mathbb{X}\beta$ becomes underdetermined, and there are infinitely many β that perfectly determine $\mathbf{Y} = \mathbb{X}\beta$. Thus, we will need to regularize the model. We will address this in the next subsections by analyzing various estimators.

3.3.1 Ridge estimator

Definition 3.33. (Ridge estimator). **Closed-form representation of the ridge estimator**:

$$\hat{\beta}^{\text{Ridge},\lambda} = (\mathbb{X}^T\mathbb{X} + \lambda I_d)^{-1}\mathbb{X}^T\mathbf{Y},$$

where λ is a **tuning parameter** that satisfies $\lambda > 0$. We add λ to the diagonal entries of I_d to make the matrix invertible, or so that the smallest eigenvalue of $\mathbb{X}^T\mathbb{X} + \lambda I_d$ is at least λ (regularization).

Optimization representation of the ridge estimator (a more intuitive form, from which we can derive the closed-form representation):

$$\hat{\beta}^{\text{Ridge},\lambda} = \arg\min_{\beta\in\mathbb{R}^d} \left\{||\mathbf{Y} - \mathbb{X}\beta||_2^2 + \lambda||\beta||_2^2\right\}.$$

Proof.

$$(\mathbf{Y} - \mathbb{X}\beta)^T(\mathbf{Y} - \mathbb{X}\beta) + \lambda\beta^T\beta = \mathbf{Y}^T\mathbf{Y} - 2Y^T\mathbb{X}\beta + \beta^T\mathbb{X}^T\mathbb{X}\beta + \lambda\beta^T\beta = f(\beta).$$

$$\frac{\partial f(\beta)}{\partial\beta} = -2\mathbb{X}^T\mathbf{Y} + 2\mathbb{X}^T\mathbb{X}\beta + 2\lambda\beta = 0 \Longrightarrow (\mathbb{X}^T\mathbb{X} + \lambda I_d)\beta = \mathbb{X}^T\mathbf{Y} \Longrightarrow \beta = (\mathbb{X}^T\mathbb{X} + \lambda I_d)^{-1}\mathbb{X}^T\mathbf{Y}.$$

□

Lemma 3.34. *For each $\lambda > 0$, there exists a one-to-one mapping for t (i.e. there exists a unique t) such that*

$$\hat{\beta}^{Ridge,\lambda} = \arg \min_{||\beta||_2^2 \leq t} ||\mathbf{Y} - \mathbb{X}\beta||_2^2,$$

which is the ***constraint form or regularization form of the ridge estimator****. This extra constraint $||\beta||_2^2 \leq t$ shrinks the model space (regularization via the ridge estimator).*

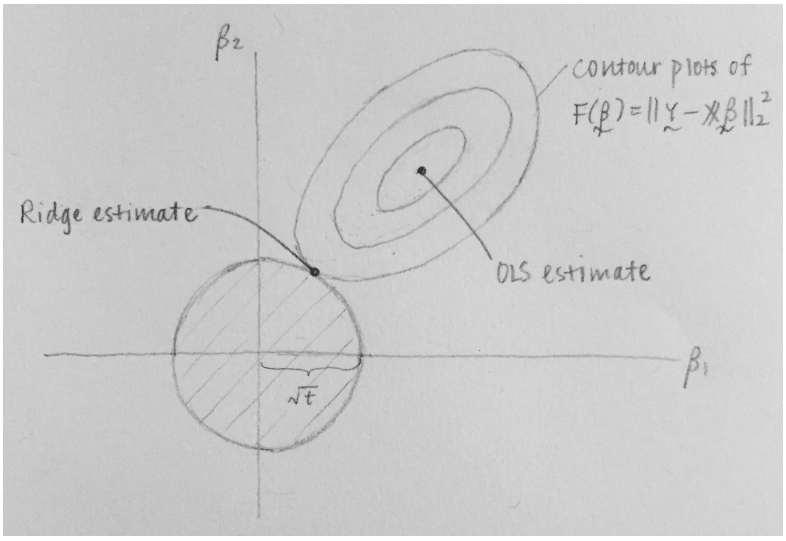
Proof. Sketch of proof: By Lagrangian duality, for each λ in

$$\min_{\beta} ||\mathbf{Y} - \mathbb{X}\beta||_2^2 + \lambda(||\beta||_2^2 - t),$$

where λ is the Lagrange multiplier, there must exist a corresponding unique t. □

Remark 3.35. When t is big enough, such that $t \geq ||\hat{\beta}^{\text{OLS}}||_2^2$, this corresponds to $\lambda = 0$. If $t = 0$, then $\lambda = \infty$. Smaller values of t correspond to larger values of λ.

Remark 3.36. (Geometry of the ridge estimator for $d = 2$). See the following figure for the ridge estimator when $d = 2$. The contour lines (along which the function has a constant value) for $F(\beta) = ||\mathbf{Y} - \mathbb{X}\beta||_2^2$ are ellipses since the objective function is quadratic in β.



Remark 3.37. Computation of the ridge regression estimator is a convex optimization. However, never naively solve it using a general purpose solver.

3.3.2 Bridge estimator

Definition 3.38. (L_p-norm). $\mathbf{x} \in \mathbb{R}^n$. $||\mathbf{x}||_p := (\sum_i |\mathbf{x}_i|^p)^{1/p}$.

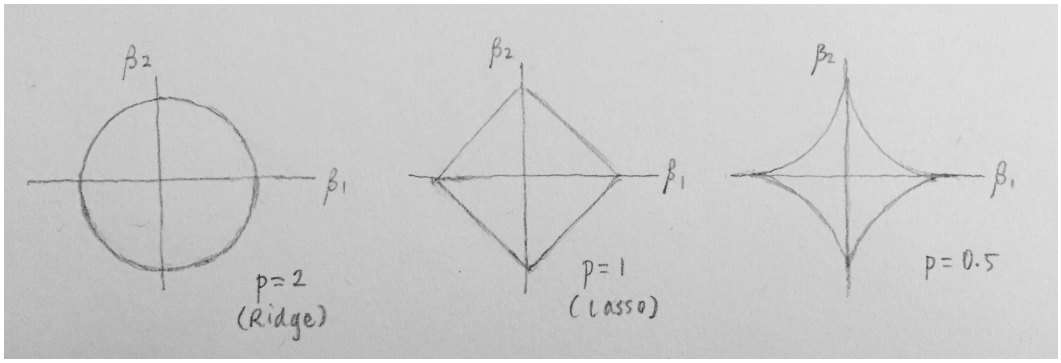
Remark 3.39. If $1 \leq p < \infty$, then $||\mathbf{x}||_p$ is a norm (the triangle inequality is satisfied), but if $0 < p < 1$, then $||\mathbf{x}||_p$ is not a norm but a pseudo-norm.

Definition 3.40. (Bridge estimator).

$$\hat{\beta}^{\text{Bridge},\lambda} = \arg \min_{\beta \in \mathbb{R}^d} ||Y - \mathbf{X}\beta||_2^2 + \lambda||\beta||_p^p, \ 0 < p < \infty, \ \lambda > 0.$$

This is a family of estimators—an extension of the ridge estimator. Note that the ridge estimator is the bridge estimator with $p = 2$.

We plot the constraint region $||\beta||_p^p \leq t$ when $d = 2$ in the following figure.



3.3.3 Lasso estimator

Definition 3.41. (Lasso (Least Absolute Shrinkage and Selection Operator) estimator). The lasso estimator is the bridge estimator with $p = 1$. The **constrained form of the lasso estimator** is

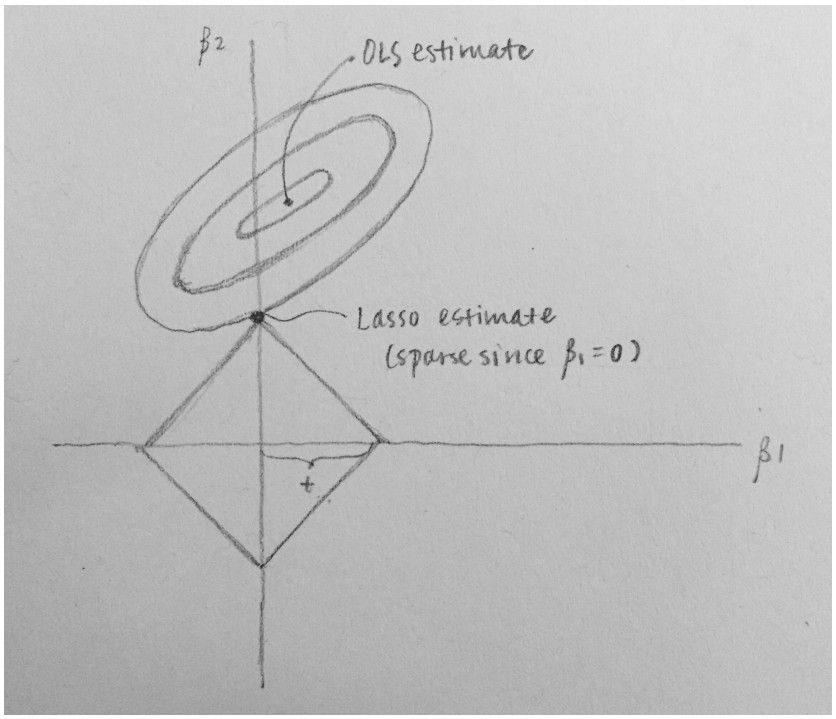
$$\hat{\beta}^{\text{Lasso},\lambda} = \arg \min_{||\beta||_1 \leq t} ||\mathbf{Y} - \mathbb{X}\beta||_2^2,$$

or equivalently,

$$\hat{\beta}^{\text{Lasso},\lambda} = \arg \min_{\beta} ||\mathbf{Y} - \mathbb{X}\beta||_2^2 + \lambda||\beta||_1.$$

The lasso estimator is associated with a property called **sparsity**, which means that many elements of β are 0. (If $\hat{\beta}_j^{\lambda} = 0$, then the jth feature is not selected.) Sparsity helps with **variable selection** and is aligned with the parsimonious principle.

Remark 3.42. (Geometry of the lasso estimator for $d = 2$). See the following figure for the lasso estimator when $d = 2$. Note that sparsity is achieved since $\beta_1^{\text{Lasso}} = 0$.



Question 3.43. Why is the lasso estimator ($p = 1$) so unique within the bridge estimator family?

1. $0 < p \leq 1 \rightarrow$ "corner" sparsity.
2. $1 \leq p < \infty \rightarrow$ convexity.

The lasso estimator is both sparse and convex (which occurs only when $p = 1$).

Remark 3.44. (Geometry of the lasso estimator for $d = 2$). Think of a polytope touching a balloon, or of a football hitting a guy in the nose.

Definition 3.45. (Collinearity). A phenomenon in which two or more predictor variables are highly correlated.

Remark 3.46. (Comparison of ridge and lasso estimators).

Ridge		Lasso
Not sparse	$<$	Sparse (good for variable selection)
Strongly convex	$>$	Convex
Solution is unique $\rightarrow$ stable	$>$	Solution may not be unique $\rightarrow$ less stable
Handles multicollinearity well	$>$	Does not handle multicollinearity well

Remark 3.47. Unless $d < 20$, we do not prefer OLS and need to regularize.

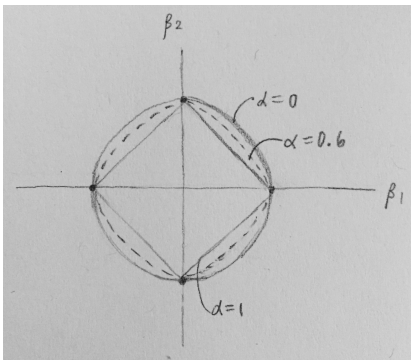
3.3.4 Elastic-net estimator

Definition 3.48. (Elastic-net estimator).

$$\hat{\beta}^{\text{Elastic},\lambda,\alpha} = \arg\min_{\beta} ||Y - \mathbb{X}\beta||_2^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2),$$

where $\lambda > 0,\ \alpha \in [0, 1]$. When $\alpha = 1 \rightarrow$ Lasso. When $\alpha = 0 \rightarrow$ Ridge.

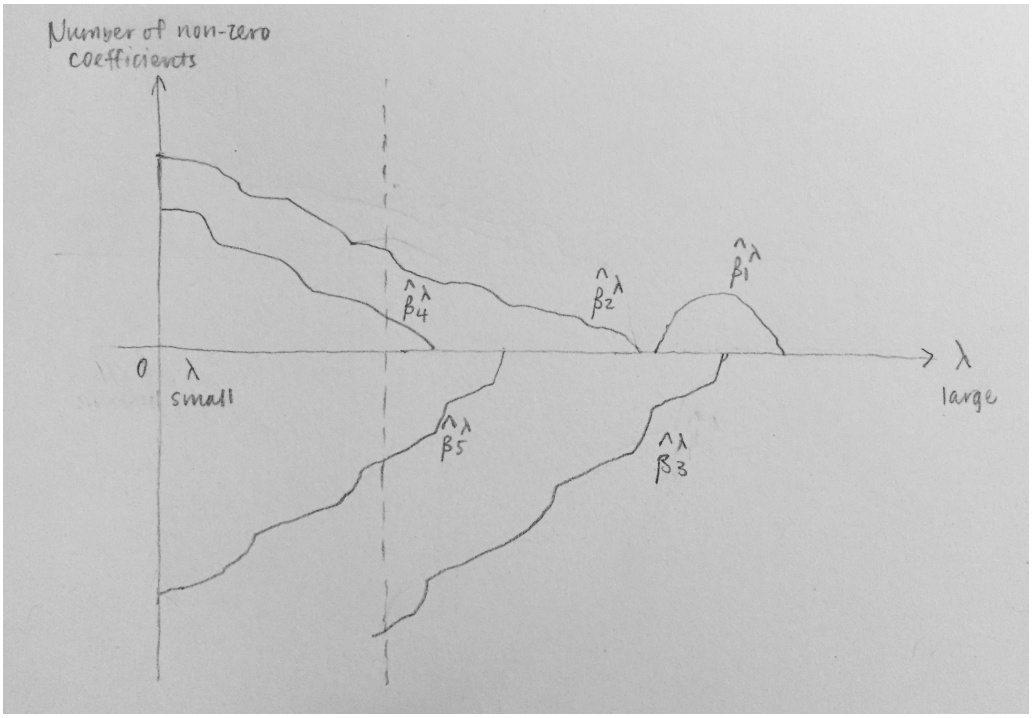
Remark 3.49. (Geometry of the elastic-net penalty estimator). See the following figure. Recall that with lasso estimation, coefficients are sent to 0 one by one, resulting in a sparse solution with large λ. With ridge estimation, coefficients approach 0 but are never 0. When $\lambda = 0$, the solution is the OLS estimator.



Question 3.50. How do we choose the two tuning parameters λ and α? Solution: regularization path.

Definition 3.51. (Regularization path). Regularization paths are also used to detect multicollinearity.

Consider the lasso estimator. Plot a visualization of $\hat{\beta}^{\text{Lasso},\lambda}$ vs. λ. As λ becomes closer to 0, more non-zero coefficients come out. The following figure illustrates a regularization path.



Definition 3.52. (Two-stage model diagnosis approach for elastic-net tuning parameter selection).

1. Use $\alpha = 1$. Fit a full lasso path, and visualize the regularization path.
2. Use $\alpha = 1 - \frac{1}{e} \approx 0.63$. Fit the regularization path, then examine whether there is a significant change in the fitted path. If there is no significant change, then there is probably no multicollinearity, so use $\alpha = 1$ to take full advantage of lasso. If there is a significant change, then this is probably due to the instability of lasso due to multicollinearity, so use $\alpha = 0.6$ to compensate.
3. For both cases, use cross-validation to choose λ_k. (We define cross-validation in the following discussion of model selection.)

3.3.5 Model selection

Question 3.53. How do we choose t or λ? (Each λ indexes a model.) We refer to this process as **model selection**. We typically use cross-validation to choose the tuning parameter λ: if we select a set of candidates for λ, with each defining a model, then we can then perform cross-validation to select a model with its corresponding λ.

Question 3.54. Problem step: Given a set of tuning parameters $\Lambda = \{\lambda_1, ..., \lambda_K\}$, $\lambda_1 =$ big, $\lambda_2 = \rho\lambda_1$, $\lambda_3 = \rho^2\lambda_1, ...$, $\rho = 0.95$, how do we choose the "best" one? We want λ s.t. the noise is filtered out but the signal is captured (Data = Signal + Noise). We define a "good" model to be one that behaves well on the testing data. (Note, however, that we might not have testing data.)

Definition 3.55. (Data splitting). We split (usually randomly) the data $\mathcal{D} = \{X_1, ..., X_n\}$ into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}$ with sizes $n_1 + n_2 = n$. (Ex: $n_1 = n_2 = \frac{1}{2}$.) Let $\hat{\beta}^{\lambda_1}, ..., \hat{\beta}^{\lambda_K}$ be ridge estimators on subset $\mathcal{D}_1$. We define the **data splitting (DS) score** corresponding to λ_k as

$$DS(k) = \frac{1}{n_2} \sum_{i \in \mathcal{D}_2} \left(Y_i - \mathbf{X}_i^T \hat{\beta}^{\lambda_k} \right)^2.$$

We then pick the model with the smallest DS score.

Remark 3.56. Intuition: Conditioning on $\mathcal{D}_1$, it is easy to see that $DS(k)$ is an unbiased estimator of

$$R(\hat{\beta}^{\lambda_k}) = E((Y - \hat{\beta}^{\lambda_k}\mathbf{X})^2 | \mathcal{D}_1),$$

by the concentration principle.

Remark 3.57. There are pros and cons to data splitting. Pros: Theoretically and computationally simple. Cons: "Waste" of training data. (Traditionally, data were expensive to obtain, or there were not much data to work with.) Solution: Cross-validation (CV).

Definition 3.58. (J-fold cross-validation). We split $\mathcal{D}$ into J equally-sized parts $\mathcal{D}_1, ..., \mathcal{D}_J$, forming J **binary splits**. Generally, we choose $J = 10$. We can now use more data to train and fewer data to test:

$$DS_1 : \mathcal{D}_1 \text{ (testing) vs. } \mathcal{D} \backslash \mathcal{D}_1 \text{ (training).}$$

$$DS_2 : \mathcal{D}_2 \text{ (testing) vs. } \mathcal{D} \backslash \mathcal{D}_2 \text{ (training).}$$

$$\vdots$$

$$DS_J : \mathcal{D}_J \text{ (testing) vs. } \mathcal{D} \backslash \mathcal{D}_J \text{ (training).}$$

For each $\lambda_k \in \Lambda$, we calculate the data-splitting scores $DS_1, ..., DS_J$ and denote the results $DS_1(k), ..., DS_J(k)$. The cross-validation (CV) score is

$$CV(k) := \frac{1}{J} \sum_{j=1}^{J} DS_j(k).$$

We then pick the model with the smallest CV score. After we pick λ, we use this λ to fit on the entire dataset.

Remark 3.59. We quickly summarize our coverage of regression thus far. The goal of regression is to minimize the risk

$$R(f) := E[|Y - f(\mathbf{X})|^2].$$

This perspective is completely model-free since we do not assume anything about the joint distribution $(\mathbf{X}, Y) \sim P_{\mathbf{X},Y}$. We proved that we minimize $R(f)$ by using

$$f^*(\mathbf{x}) := E(Y|\mathbf{X} = \mathbf{x}).$$

However, we cannot observe $R(f)$, so we analyze the empirical risk

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2.$$

But minimizing empirical risk leads to overfitting, so we need to consider regularization: only considering linear models $f(\mathbf{X}) = \beta^T\mathbf{X}$ gives us the OLS estimator

$$\hat{\beta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T Y.$$

(This solution was also derived using a model-based perspective—using the Gaussian noise model.) This approach fails with high-dimensional data, so we considered the ridge, bridge, lasso, and elastic-net estimators.

3.4 Classification analysis

Definition 3.60. (Classification). Classification is a special case of regression: regression with categorical response variables.

$$Y \in \{C_1, ..., C_k\}.$$

- $k = 2$: Binary classification (our focus).
- $k > 2$: Multiclass classification, which is reducible to binary cases.

Remark 3.61. Goal: Similar to regression, we aim to find a mapping h such that Y and $h(\mathbf{X})$ are "close" to each other. $Y \in \{+1, -1\}$, $\mathbf{X} \in \mathbb{R}^d$, $h(\mathbf{X}) \in \{+1, -1\}$. $h : \mathcal{X} \to \{+1, -1\}$, where $\mathcal{X}$ is the input space.

Definition 3.62. (Population risk).

$$R(h) = P\{Y \neq h(\mathbf{X})\}.$$

Remark 3.63. How shall we measure the closeness? In the regression setting, we gave examples of L_1-risk, L_2-risk, etc. (They were all valid, but we chose L_1-risk.) Now, in the classification setting, can we still use L_2-risk? Answer: yes.

$$L(h) := ||Y - h(\mathbf{X})||_2^2 = 4I(Y \neq h(\mathbf{X})).$$

We call $I(Y \neq h(\mathbf{X}))$ the **0/1-loss**. We will only use 0/1-loss for classification. The risk function is defined by

$$R(h) := E(l(h)) = E(I(Y \neq h(\mathbf{X})) = P\{Y \neq h(\mathbf{X})\}.$$

Unfortunately, the indicator function is not convex, so empirical risk minimization is NP-hard.

Definition 3.64. (Bayes classification rule/Bayes rule).

$$h^* = \arg\min_h R(h).$$

Definition 3.65. (Bayes risk).

$$R^* = R(h^*).$$

Theorem 3.66.

$$h^*(\mathbf{x}) = \begin{cases} +1 \ \textit{if } P(Y=+1|\mathbf{X}=\mathbf{x}) > \frac{1}{2} \\ -1 \ \textit{otherwise.} \end{cases}$$

Proof.

$$\begin{aligned} R(h) &= P(Y \neq h(\mathbf{X})) \\ &= 1 - P(Y = h(\mathbf{X})) \\ &= 1 - \sum_{y \in \{+1,-1\}} P(Y=y, h(\mathbf{X})=y) \\ &= 1 - \sum_{y \in \{+1,-1\}} E_{\mathbf{X},Y}[I(Y=y, h(\mathbf{X})=y)] \\ &= 1 - \sum_{y \in \{+1,-1\}} E_X\left[E_{Y|\mathbf{X}}[I(Y=y) \cdot I(h(\mathbf{X})=y)|\mathbf{X}]\right] \\ &= 1 - \sum_{y \in \{+1,-1\}} E_X[I(h(\mathbf{X})=y) \cdot P(Y=y|\mathbf{X})] \\ &= 1 - \int_{\mathcal{X}} \left[I(h(\mathbf{x})=+1)P(Y=+1|\mathbf{X}=\mathbf{x}) + I(h(\mathbf{x})=-1)P(Y=-1|\mathbf{X}=\mathbf{x})\right] p(\mathbf{x})dx. \end{aligned}$$

We aim to maximize the integrand, so

$$h^*(\mathbf{x}) = \begin{cases} +1 \text{ if } P(Y=+1|\mathbf{X}=\mathbf{x}) > P(Y=-1|\mathbf{X}=\mathbf{x}) \\ -1 \text{ otherwise.} \end{cases}$$

We finish the proof using $P(Y=+1|\mathbf{X}=\mathbf{x}) + P(Y=-1|\mathbf{X}=\mathbf{x}) = 1$. □

Remark 3.67. The key of classification is to model

$$P(Y=+1|\mathbf{X}=\mathbf{x}) =: r(\mathbf{x}),$$

where $r(\mathbf{x})$ is the discriminant function.

Remark 3.68. What is the connection to regression?

$$E(Y|\mathbf{X}=\mathbf{x}) = P(Y=+1|\mathbf{X}=\mathbf{x}) - P(Y=-1|\mathbf{X}=\mathbf{x}) = 2P(Y=+1|\mathbf{X}=\mathbf{x}) - 1.$$

Thus, there is a 1-to-1 mapping of the regression function and discriminant function:

$$f^*(\mathbf{x}) = 2r(\mathbf{x}) - 1.$$

(Recall that regression analysis includes classification analysis.)

Definition 3.69. (Decision boundary).

$$D(r) := \left\{ \mathbf{x} : r(\mathbf{x}) = \frac{1}{2} \right\}.$$

If $r(\mathbf{x})$ is a linear function of $\mathbf{x}$, then $r(\mathbf{x})$ is called a **linear classifier**.

Question 3.70. How shall we model $P(Y = +1|\mathbf{X} = \mathbf{x})$? We want to parameterize by some function $f \to P_f(Y = +1|\mathbf{X} = \mathbf{x})$ such that it is nonnegative and lies in $[0, 1]$.

3.4.1 Discriminative modeling (logistic regression, ...)

Definition 3.71. (Logistic model).

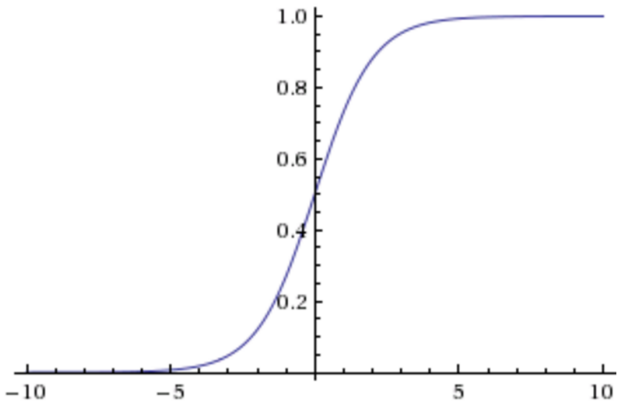
$$P_f(Y = +1|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}}.$$

$$\implies P_f(Y = -1|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{f(\mathbf{x})}}.$$

A more compact representation of the logistic model is

$$P_f(Y = y|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{-yf(\mathbf{x})}}.$$

The following plots the logistic function $f(u) = \frac{1}{1+e^{-u}}$.



Logistic modeling is the most "principled" approach (relates to justifications in GLM and leads to convex optimization).

Note that in ML, $Y \in [+1, -1]$ is the convention, whereas in statistics, $Y \in [+1, 0]$ is typically used.

Definition 3.72. (Probit modeling). Another idea is modeling by

$$P(Y = +1|\mathbf{X} = x) = \Phi\left(f(\mathbf{x})\right),$$

where $\Phi(\cdot)$ is the Gaussian CDF. Note that probit modeling leads to non-convex computations.

Definition 3.73. (Statistical model of logistic regression).

$$\{p(y, \mathbf{x}) = P_f(Y = y|\mathbf{X} = \mathbf{x})p_{\mathbf{X}}(\mathbf{x})\}_{f, p(\mathbf{x})}.$$

Note that logistic regression is a nonparametric model—strictly speaking, f is infinite-dimensional. For classification, we only care about $P_f(Y = y|\mathbf{X} = \mathbf{x})$. f is the parameter of interest.

Remark 3.74. (Parallel in regression setting). $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n) \sim P$. The parallel in the regression setting is: $R(h) = P(Y \neq h(\mathbf{X})) = E[I(Y \neq h(\mathbf{X}))]$. $\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} I(Y \neq h(\mathbf{X}_i)$. Note that the indicator function is not convex. Thus, empirical risk minimization[1] is NP-hard. We will cover the risk minimization approach after the model-based perspective.

Remark 3.75. (MLE of logistic regression—a model-based perspective). Note that there is no regularization involved yet—we give the model full freedom at this point.

$(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n) \sim P$.

$$L_n(f) = \prod_{i=1}^{n} p_f(Y_i|\mathbf{X}_i)p_{\mathbf{X}}(\mathbf{X}_i).$$

$$l_n(f) = \sum_{i=1}^{n} \log p_f(Y_i|\mathbf{X}_i) + \sum_{i=1}^{n} \log p_{\mathbf{X}}(\mathbf{X}_i).$$

Since the second term of $l_n(f)$ does not depend on f,

$$\hat{f} = \arg\max_f l_n(f) = \arg\min_f \sum_{i=1}^{n} \log\left(1 + e^{-Y_i f(\mathbf{X}_i)}\right).$$

We aim to drive $e^{-Y_i f(\mathbf{X}_i)}$ to 0. Thus,

$$\hat{f}(\mathbf{x}) = \begin{cases} +\infty \text{ if } \mathbf{x} = \mathbf{X}_i \text{ and } Y_i = +1 \\ -\infty \text{ if } \mathbf{x} = \mathbf{X}_i \text{ and } Y_i = -1 \\ \text{arbitrary otherwise.} \end{cases}$$

This results in overfitting (observed data are perfectly fit), so regularization is needed. Different ways to regularize f follow.

Example 3.76. (Linear logistic regression). $f \in \mathcal{F}$, where

$$\mathcal{F} := \{f(x) : f(\mathbf{x}) = \beta_0 + \beta^T \mathbf{x}\}.$$

Note that the intercept β_0 is important—we need to explicitly write out a more complicated classification.

[1] Recall that empirical risk minimization with 0-1 loss is

$$\min_{h \in \{+1, -1\}} \frac{1}{n} \sum_{i=1}^{n} I(Y_i \neq h(\mathbf{X}_i)).$$

Example 3.77. (Nonparametric logistic regression). $f \in \mathcal{F}$, where

$$\mathcal{F} := \left\{ f(\mathbf{x}) : \int f''(\mathbf{x})^2 dx < \infty \right\}.$$

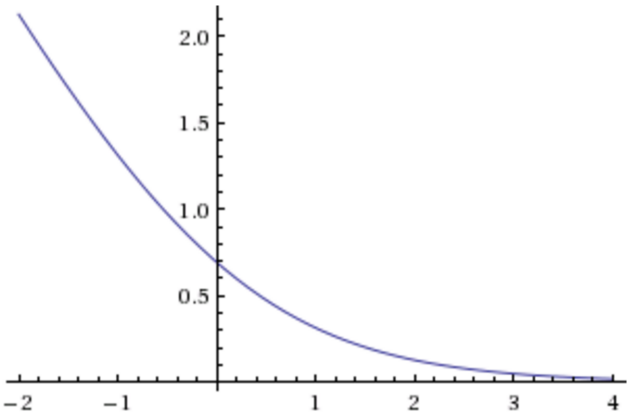
Remark 3.78. (Risk-minimization interpretation of logistic regression). We already have a model-based interpretation, but this second interpretation is more extendable and model-free and allows us to use advanced ML techniques (ex: SVM boosting).

Remark 3.79. Key insight: the MLE of the logistic model induces a new objective function/loss.

Definition 3.80. (Logistic loss).

$$\ell^{\text{Logistic}}(y, f(\mathbf{x})) := \log(1 + e^{-yf(\mathbf{x})}).$$

The loss function is monotonically decreasing and, thus, encourages the **functional margin** $yf(\mathbf{x})$ to be big, in order to minimize loss. The following plots the loss function $\ell^{\text{Logistic}}(y, f(\mathbf{x}))$ against the functional margin $yf(\mathbf{x})$. The negative x-axis is the zone of wrong classification. The positive x-axis is the zone in which y and $f(\mathbf{x})$ have the same sign.



Definition 3.81. (Logistic risk).

$$R(f) := E_{Y,\mathbf{X}} \log(1 + e^{-Yf(\mathbf{X})}).$$

$Yf(\mathbf{X})$ is the functional margin. By LLN/concentration principle, logistic loss will converge to logistic risk. Note that the R of logistic risk is not necessarily related to model-based R, though they do have a relationship—this is beyond this course.

Remark 3.82. Key intuition: To minimize $R(f)$, we encourage Y and $f(\mathbf{X})$ to have the same sign. In other words, if $Y = +1$ AND $f(\mathbf{X}) > 0$, or $Y = -1$ AND $f(\mathbf{X}) < 0$, then classification is correct. Thus, the bigger $yf(\mathbf{X})$ is, the better.

$$P(Y = +1|\mathbf{X}) = \frac{1}{1 + e^{-f(\mathbf{X})}} = \frac{e^{f(\mathbf{X})}}{1 + e^{f(\mathbf{X})}} \Longrightarrow P(Y = -1|\mathbf{X}) = \frac{1}{1 + e^{f(\mathbf{X})}}$$

$$\implies \frac{P(Y=+1|\mathbf{X})}{P(Y=-1|\mathbf{X})} = e^{f(\mathbf{X})} \implies f(\mathbf{X}) = \log\left(\frac{P(Y=+1|\mathbf{X})}{P(Y=-1|\mathbf{X})}\right).$$

We call

$$\log\left(\frac{P(Y=+1|\mathbf{X})}{P(Y=-1|\mathbf{X})}\right)$$

the **log-odds ratio**.

Assuming $Y=+1$, we need the log-likelihood ratio to be big, or for

$$P(Y=+1|\mathbf{X}) > P(Y=-1|\mathbf{X}) \implies f(\mathbf{X}) > 0.$$

Note that Y and $f(\mathbf{X})$ have the same sign.

Assuming $Y=-1$, we need the log-likelihood ratio to be small, or for

$$P(Y=+1|\mathbf{X}) < P(Y=-1|\mathbf{X}) \implies f(\mathbf{X}) < 0.$$

Note that Y and $f(\mathbf{X})$, again, have the same sign.

Remark 3.83. $f(\mathbf{X})$ can be whatever function you want (that allows regularization) and can have an arbitrary range. Ex: $f(\mathbf{x}) = \beta^T\mathbf{x}$.

Remark 3.84. Note that this risk minimization approach is not NP-hard, unlike a different risk minimization approach mentioned previously. These two approaches are under different loss functions.

We now analyze high-dimensional logistic regression.

Definition 3.85. (Ridge logistic regression).

$$\hat{\beta}^{\text{Ridge},\lambda} = \arg\min_{\beta} \sum_{i=1}^{n} \log(1+e^{-Y_i(\beta^T\mathbf{X}_i)}) + \lambda||\beta||_2^2.$$

Similarly for other loss functions, we add the relevant corresponding penalty term.

Definition 3.86. (L_1-logistic/Lasso-logistic regression).

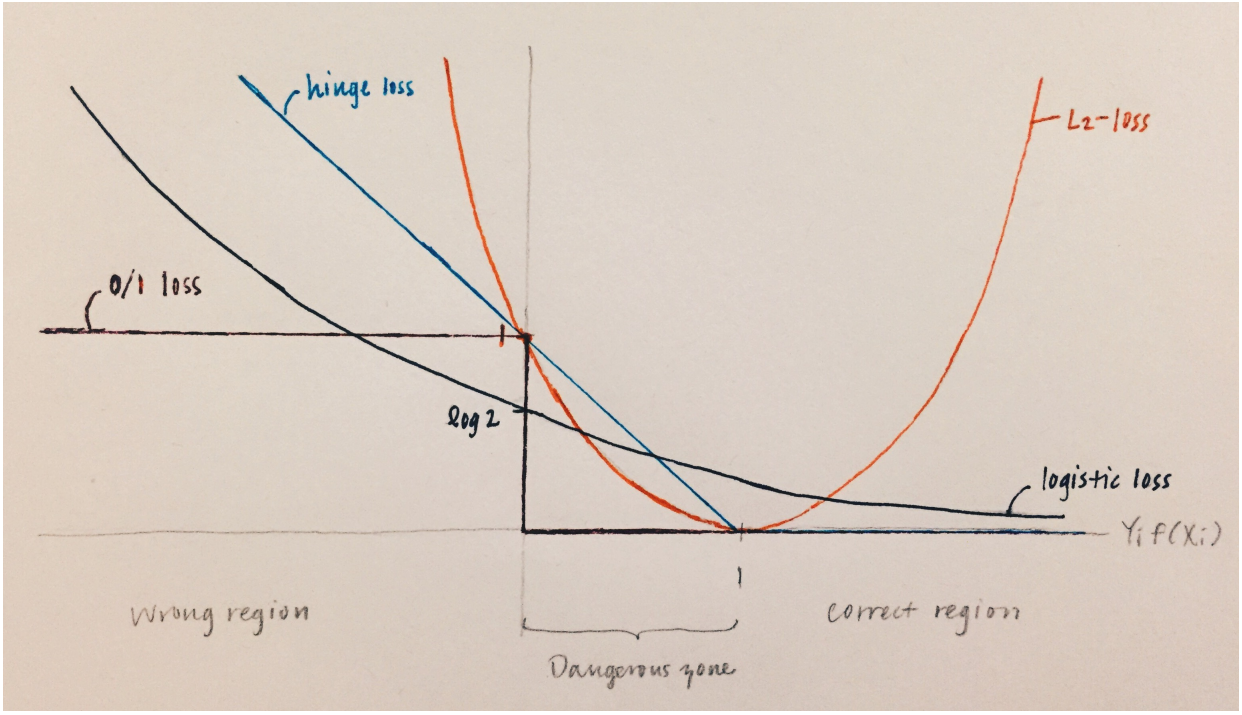
$$\hat{\beta}^{\text{Lasso},\lambda} = \arg\min_{\beta} \sum_{i=1}^{n} \log(1+e^{-Y_i(\beta^T\mathbf{X}_i)}) + \lambda||\beta||_1.$$

Note that the geometric interpretation of the lasso estimator carries over (we still get sparsity, etc.).

Definition 3.87. (Elastic-net logistic regression).

$$\hat{\beta}^{\text{Elastic},\lambda} = \arg\min_{\beta} \sum_{i=1}^{n} \log(1+e^{-Y_i(\beta^T\mathbf{X}_i)}) + \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2).$$

Question 3.88. Can we use other loss functions? The following figure corresponds to the following discussion of loss functions—geometric intuition is key to our arguments.



Question 3.89. In particularly, can we naively use L_2-loss? How does L_2-loss compare to logistic loss? Recall that ridge L_2-regression uses

$$\hat{\beta}^{\lambda} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - \beta^T \mathbf{X}_i)^2 + \lambda ||\beta||_2^2.$$

We can rewrite this as

$$\hat{\beta}^{\lambda} \arg\min_{\beta} \sum_{i=1}^{n} (1 - Y_i \beta^T \mathbf{X}_i)^2 + \lambda ||\beta||_2^2.$$

The two formulations involve the same effective loss functions since $Y \in \{+1, -1\}$.

Note that $Y_i \beta^T \mathbf{X}_i$ is the functional margin, and the summation corresponds to a quadratic loss function $\ell(u) = (1-u)^2$. The loss function is small when the functional margin is near 1, but when the functional margin is large and in the correct zone, it incurs a large loss—despite being correct! This is a significant drawback of L_2-loss.

However, using L_2-loss encourages the functional margin to be away from the negative horizontal axis (misclassification) and also far from the ambiguous/"dangerous" zone between 0 and 1 on the horizontal axis. L_2-loss constrains the functional margin to be in a small region—but it is the correct region. On the other hand, logistic loss encourages the functional margin to be large but does not penalize as severely in the dangerous zone and, thus, does a worse job of keeping it away from the dangerous zone. Analogy: logistic loss is like a babysitter—it wants to push kids away from the wrong region—but in this, logistic loss subtly changes the utility function we optimize over. It seeks to maximize global/average margin, perhaps at the expense of individual data points. Furthermore, using logistic loss may require more data than L_2-loss. L_2-loss leads to linear discriminant analysis (LDA), which is powerful. So L_2-loss is not that bad—it is hard to say if L_2-loss or logistic loss is better (it depends on the regime).

Definition 3.90. (0/1-loss). I(Y has a different sign than $\beta^T \mathbf{X}$) = I($Y f(\mathbf{X}) < 0$).

$$R(h) = E[I(Y \neq h(\mathbf{X}))].$$

0/1-loss is great—it penalizes severely in the wrong zone, but if classification is correct, then there is no penalty. However, it is non-convex, so a surrogate for 0/1-loss is useful.

Remark 3.91. Note that logistic loss is a surrogate loss—it tries to minimize 0/1-loss by finding a convex surrogate for 0/1-loss. (In fact, with ridge, it is strongly convex.) Logistic loss maximizes global 0/1-loss.

Definition 3.92. (Support vector machine (SVM)). Ridge logistic regression but replace logistic loss with "hinge" loss. Geometrically, hinge loss is very similar to logistic loss.

Define $(x)_+ := \max\{x, 0\}$.

$$\hat{\beta}^\lambda = \arg\min_\beta \sum_{i=1}^n (1 - Y_i \beta^T \mathbf{X_i})_+ + \lambda ||\beta||_2^2.$$

Remark 3.93. SVM vs. logistic regression? SVM essentially lies in between 0/1 and logistic. If classification is wrong, SVM penalizes severely (even more than 0/1 and logistic do). If classification is right, SVM stops penalizing. SVM is not like quadratic, which penalizes even if classification is correct. SVM pushes points as far away from the dangerous zone as possible.

Reasons to prefer logistic over SVM follow: logistic is model-based, while SVM is model-free. This inherently comes with certain benefits: we can only use the model-free perspective for prediction, but the model-based perspective allows uncertainty assessment that is quite interpretable (old-fashioned statistics, such as the log-odds ratio).

In the real world, SVM is seemingly better than logistic in terms of accuracy. Roughly, SVM is much better than logistic, which is slightly better than LDA/L_2, in line with our geometric intuition.

Definition 3.94. (Boosting). Same intuition as SVM but using exponential loss.

$$\min_{f \in \mathcal{F}} \sum_{i=1}^n e^{-Y_i f(\mathbf{X})}.$$

Boosting penalizes exponentially fast in the wrong zone (see geometric intuition).

3.4.2 Generative modeling

An overview of discriminative vs. generative modeling follows. With **discriminative modeling**, we directly model the distribution of $Y|X$ (the discriminant function) but do not model the distribution of X. Thus, we are unable to generate new data (Y, X) since we do not have information on the distribution of X. With discriminative modeling, we only want to do prediction and do not care about understanding the underlying joint distribution and, thus, disregard the marginal distribution X.

On the other hand, with **generative modeling**, we model both the distribution of $X|Y$ and the distribution of Y and then use Bayes formula to model $Y|X$ (the discriminant

function). We are therefore able to *generate* new data (Y, X) using the joint distribution we modeled. Thus, generative modeling involves some belief of how the data were generated. (In this sense, discriminative modeling relies on fewer assumptions.)

Remark 3.95. Remark on notation: we use P to denote a discrete distribution (pmf) and p to denote a continuous distribution (pdf). (However, these are actually interchangeable—studied in a more advanced class.)

Definition 3.96. (Bayes formula).

$$\begin{aligned} P(Y=+1|\mathbf{X}=\mathbf{x}) &= \frac{p(\mathbf{x}|Y=+1)P(Y=+1)}{p(\mathbf{x}|Y=+1)P(Y=+1)+p(\mathbf{x}|Y=-1)P(Y=-1)} \\ &=: \frac{p_+(\mathbf{x})\eta}{p_+(\mathbf{x})\eta + p_-(\mathbf{x})(1-\eta)}, \end{aligned}$$

where we let $p_+\mathbf{x} := p(\mathbf{x}|Y=+1)$, $p_-\mathbf{x} := p(\mathbf{x}|Y=-1)$, and $\eta := P(Y=+1)$.

Similarly,

$$P(Y=-1|\mathbf{X}=x) =: \frac{p_-(\mathbf{x})(1-\eta)}{p_+(\mathbf{x})\eta + p_-(\mathbf{x})(1-\eta)}.$$

Note that $Y \sim Ber(\eta)$. Note that we let Bernoulli take on values $+1$ and -1 instead of 1 and 0.

Remark 3.97. Once we have modeled $p_+(\mathbf{x})$, $p_-(\mathbf{x})$, and η, we can generate new data $(X_1', Y_1'), ..., (X_n', Y_n')$. Why is this useful?

- We can run simulations.
- We can explicitly describe underlying mechanisms of stochastic phenomena.
- This process is more "scientific": 1) Impose hypothesis. 2) Hope hypothesis gives some insight into system.

Remark 3.98. Given i.i.d. data $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$, we estimate $p_+(\mathbf{x}), p_-(\mathbf{x}), \eta$ with MLE. (We can also use maximum penalized likelihood estimator (MPLE) or Bayesian, but in general in the big data regime, MLE fits better than Bayesian. In this class, we use MLE.)

$$\begin{aligned} \sum_{i=1}^n \log p(Y_i, \mathbf{X}_i) &= \sum_{i:Y_i=+1} \log p(Y_i, \mathbf{X}_i) + \sum_{i:Y_i=-1} \log p(Y_i, \mathbf{X}_i) \\ &= \sum_{i:Y_i=+1} \log \eta + \sum_{i:Y_i=+1} \log p(\mathbf{X}_i|Y_i=+1) + \sum_{i:Y_i=-1} \log \eta + \sum_{i:Y_i=-1} \log p(\mathbf{X}_i|Y_i=-1) \\ &= n_+ \log \eta + n_- \log(1-\eta) + \sum_{i:Y_i=+1} \log p(\mathbf{X}_i|Y_i=+1) + \sum_{i:Y_i=-1} \log p(\mathbf{X}_i|Y_i=-1), \end{aligned}$$

where we define $n_+ := \sum_{i=1}^n I(Y_i=+1)$ and $n_- = n - n_+$.

To optimize over η, we maximize $n_+ \log \eta + n_- \log(1-\eta)$:

$$\begin{aligned}\max_\eta n_+ \log \eta + n_- \log(1-\eta) &\implies \frac{n_+}{\eta} - \frac{n_-}{1-\eta} = 0 \\ &\implies \frac{n_+}{\eta} = \frac{n_-}{1-\eta} \\ &\implies n_+ - n_+\eta = n\eta - n_+\eta \\ &\implies \hat{\eta} = \frac{n_+}{n}.\end{aligned}$$

This makes sense since $Y \sim Ber(\eta)$.

To fit $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$, we need to first model $\mathbf{X}|Y=+1$ and $\mathbf{X}|Y=-1$.

Remark 3.99. We parametrize $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ with **Gaussian discriminant analysis (GDA)**, also known as **quadratic discriminant analysis (QDA)**.

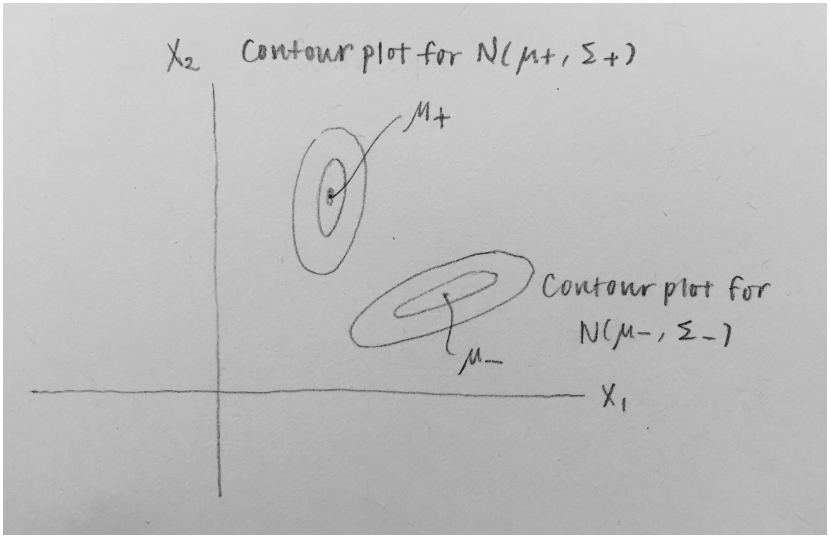
In GDA/QDA, we have

$$\begin{aligned}\mathbf{X}|Y=+1 &\sim N(\mu_+, \Sigma_+) \\ \mathbf{X}|Y=-1 &\sim N(\mu_-, \Sigma_-),\end{aligned}$$

i.e.

$$\begin{aligned}p_+(\mathbf{x}) &= \frac{1}{(2\pi)^{d/2}|\Sigma_+|^{1/2}} \cdot \exp\left(-\frac{(\mathbf{x}-\mu_+)^T\Sigma_+^{-1}(\mathbf{x}-\mu_+)}{2}\right) \\ p_-(\mathbf{x}) &= \frac{1}{(2\pi)^{d/2}|\Sigma_-|^{1/2}} \cdot \exp\left(-\frac{(\mathbf{x}-\mu_-)^T\Sigma_-^{-1}(\mathbf{x}-\mu_-)}{2}\right),\end{aligned}$$

where $\mathbf{X} \in \mathbb{R}^d$ and $|\Sigma|$ is the determinant of the covariate matrix. Assume that Σ is invertible whenever we use a multivariate Gaussian. See the following figure for contour plots of $N(\mu_+, \Sigma_+)$ and $N(\mu_-, \Sigma_-)$.



Note that we are not assuming the true conditional distributions are Gaussian; we are using a simplified model for the sake of regularization. If this does not fit the data well, then we can always fit a new model. Even if it does not fit the data well, though, this model still has some predictive power.

Definition 3.100. Note that we can write Bayes rule under specific models. **Bayes rule under GDA** can be rewritten as follows. First note that

$$\begin{aligned} P(Y=+1|\mathbf{X}=\mathbf{x}) > P(Y=-1|\mathbf{X}=\mathbf{x}) &\implies \frac{p_+(\mathbf{x})\eta}{p_+(\mathbf{x})\eta + p_-(\mathbf{x})(1-\eta)} > \frac{p_-(\mathbf{x})(1-\eta)}{p_+(\mathbf{x})\eta + p_-(\mathbf{x})(1-\eta)} \\ &\implies p_+(\mathbf{x})\eta > p_-(\mathbf{x})(1-\eta) \\ &\implies \log\frac{p_+(\mathbf{x})}{p_-(\mathbf{x})} + \log\frac{\eta}{1-\eta} > 0. \end{aligned}$$

Note that

$$\begin{aligned} \frac{p_+(\mathbf{x})}{p_-(\mathbf{x})} &= \frac{|\Sigma_-|^{1/2}}{|\Sigma_+|^{1/2}} \cdot \exp\left(-\frac{(\mathbf{x}-\mu_+)^T\Sigma_+^{-1}(\mathbf{x}-\mu_+)}{2} + \frac{(\mathbf{x}-\mu_-)^T\Sigma_-^{-1}(\mathbf{x}-\mu_-)}{2}\right) \\ \implies \log\frac{p_+(\mathbf{x})}{p_-(\mathbf{x})} &= \frac{1}{2}\log\frac{|\Sigma_-|}{|\Sigma_+|} + \frac{1}{2}(\mathbf{x}-\mu_-)^T\Sigma_-^{-1}(\mathbf{x}-\mu_-) - \frac{1}{2}(\mathbf{x}-\mu_+)^T\Sigma_+^{-1}(\mathbf{x}-\mu_+). \end{aligned}$$

We define the **Mahalanobis distances** to be

$$r_+^2(\mathbf{x}) = (\mathbf{x}-\mu_+)^T\Sigma_+^{-1}(\mathbf{x}-\mu_+),$$

$$r_-^2(\mathbf{x}) = (\mathbf{x}-\mu_-)^T\Sigma_-^{-1}(\mathbf{x}-\mu_-),$$

so we have that the **Bayes classifier of GDA** is

$$h^*(\mathbf{x}) = \begin{cases} +1 \text{ if } P(Y=+1|\mathbf{X}=\mathbf{x}) > P(Y=-1|\mathbf{X}=\mathbf{x}) \\ -1 \text{ otherwise.} \end{cases}$$

$$\implies h^*(\mathbf{x}) = \begin{cases} +1 \text{ if } \frac{1}{2}r_-^2(\mathbf{x}) - \frac{1}{2}r_+^2(\mathbf{x}) + \frac{1}{2}\log\frac{|\Sigma_-|}{|\Sigma_+|} + \log\frac{\eta}{1-\eta} > 0 \\ -1 \text{ otherwise.} \end{cases}$$

The decision boundary is

$$\left\{\mathbf{x} : r_-^2(\mathbf{x}) - \frac{1}{2}r_+^2(\mathbf{x}) + \frac{1}{2}\log\frac{|\Sigma_-|}{|\Sigma_+|} + \log\frac{\eta}{1-\eta} = 0\right\}.$$

Note that the decision boundary is a quadratic form of $\mathbf{x}$—it is of the form $\mathbf{x}^TA\mathbf{x}+b^T\mathbf{x}+c$, hence the name "quadratic" discriminant analysis (QDA).

Remark 3.101. Given random samples $(Y_i, \mathbf{X}_i), ..., (Y_n, \mathbf{X}_n)$, how shall we estimate Σ_+, Σ_-, μ_+, μ_-, and η? Answer: MLE. See Homework 6.

Recall that we defined $n_+ = \sum_{i=1}^n I(Y_i = +1)$ and $n_- = \sum_{i=1}^n I(Y_i = -1)$.

$$\hat{\mu}_+ = \frac{1}{n_+}\sum_{i:Y_i=+1} X_i,$$

$$\hat{\mu}_- = \frac{1}{n_-}\sum_{i:Y_i=-1} X_i,$$

$$\hat{\eta} = \frac{n_+}{n},$$

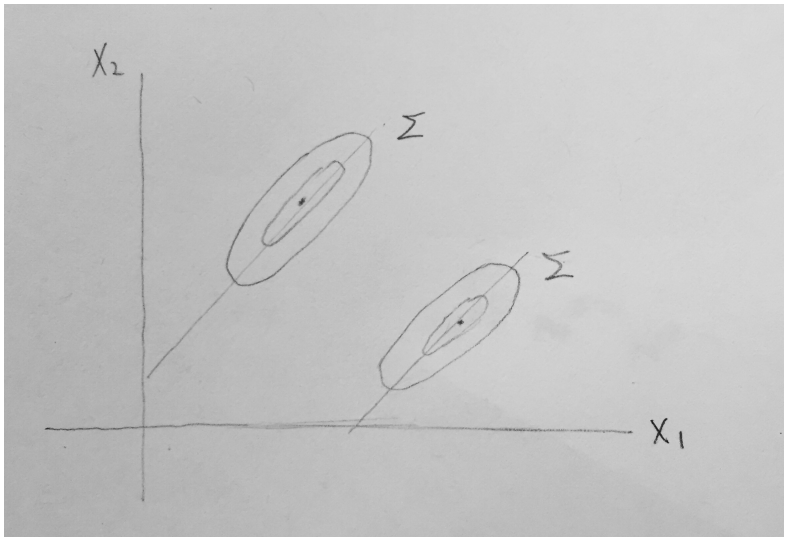
$$\hat{\Sigma}_+ = \frac{1}{n_+} \sum_{i:Y_i=+1} (\mathbf{X}_i - \hat{\mu}_+)(\mathbf{X}_i - \hat{\mu}_+)^T,$$

$$\hat{\Sigma}_- = \frac{1}{n_-} \sum_{i:Y_i=-1} (\mathbf{X}_i - \hat{\mu}_-)(\mathbf{X}_i - \hat{\mu}_-)^T.$$

Definition 3.102. Linear Discriminant Analysis (LDA). QDA with the regularization

$$\Sigma_+ = \Sigma_- = \Sigma.$$

See the following figure for the contour plots showing the regularization.



Definition 3.103. (Bayes classification rule for LDA).

We first need to calculate

$$\log \frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{P(Y=-1|\mathbf{X}=\mathbf{x})} = \log \frac{p_+(\mathbf{x})}{p_-(\mathbf{x})} + \log \frac{\eta}{1-\eta} = \frac{1}{2} r_-^2(\mathbf{x}) - \frac{1}{2} r_+^2(\mathbf{x}) + \log \frac{\eta}{1-\eta}.$$

Note that

$$\begin{aligned} r_-^2(\mathbf{x}) - r_+^2(\mathbf{x}) &= (\mathbf{x}-\mu_+)^T \Sigma^{-1} (\mathbf{x}-\mu_+) - (\mathbf{x}-\mu_+)^T \Sigma^{-1} (\mathbf{x}-\mu_+) \\ &= \mathbf{x}^T \Sigma^T \mathbf{x} - 2\mu_-^T \Sigma^{-1} \mathbf{x} + \mu_-^T \Sigma^{-1} \mu_- - \mathbf{x}^T \Sigma^T \mathbf{x} + 2\mu_+^T \Sigma^{-1} \mathbf{x} - \mu_+^T \Sigma^{-1} \mu_+ \\ &= -2\mu_-^T \Sigma^{-1} \mathbf{x} + \mu_-^T \Sigma^{-1} \mu_- + 2\mu_+^T \Sigma^{-1} \mathbf{x} - \mu_+^T \Sigma^{-1} \mu_+. \end{aligned}$$

If we define

$$\beta := \Sigma^{-1}(\mu_+ - \mu_-)$$

$$\beta_0 := \frac{1}{2} \mu_-^T \Sigma^{-1} \mu_- - \frac{1}{2} \mu_+^T \Sigma^{-1} \mu + \log \frac{\eta}{1-\eta},$$

we can write

$$h^*(\mathbf{x}) = \begin{cases} 1 \text{ if } \beta^T\mathbf{x} + \beta_0 > 0 \\ -1 \text{ otherwise.} \end{cases}$$

Thus, the LDA decision boundary is

$$\{\beta^T\mathbf{x} + \beta_0 = 0\}.$$

The decision boundary is linear in $\mathbf{x}$—hence the name "linear" discriminant analysis. Since

$$\log \frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{P(Y=-1|\mathbf{X}=\mathbf{x})} = \log \frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{1-P(Y=+1|\mathbf{X}=\mathbf{x})} = \beta^T\mathbf{x} + \beta_0,$$

we can now write

$$P(Y=+1|\mathbf{X}=\mathbf{x}) = \frac{1}{1+e^{-\beta^T\mathbf{x}-\beta_0}}.$$

Remark 3.104. Recall the linear logistic regression model:

$$\log\left(\frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{P(Y=-1|\mathbf{X}=\mathbf{x})}\right) = f(\mathbf{X}) = \beta^T\mathbf{x} + \beta_0.$$

Note that for any β and β_0 of LLR, we can choose $\Sigma = I$, $\mu_+ = \beta$, and $\mu_- = 0$ for LDA. Then, we can tune η to make sure that

$$\frac{1}{2}\beta^T\beta + \log\frac{\eta}{1-\eta} = \beta_0.$$

The conditional distribution $Y|X$ under the LDA framework appears to be precisely the conditional distribution in the setting of linear logistic regression. However, LDA can be viewed as a more regularized version of the linear logistic regression model! Why? With LDA, the extra constraints on β (and β_0) (in terms of parameters of the Gaussian distributions) constrict the model space since they cannot be arbitrary d-dimensional vectors anymore (as they can be under logistic regression). Thus, LDA is a special case of linear logistic regression.

In fact, the LR model space is strictly bigger than LDA model space. The key intuition lies in examining the joint distribution instead of the conditional distribution (after all, the joint distribution is what we care about). In the generative model (LDA), we factorize the joint distribution as $p(y,x) = p(x|y)p(y)$ and model both $p(x|y)$ and $p(y)$. In the discriminative model (LR), we factorize the joint distribution as $p(y,x) = p(y|x)p(x)$ and only model $p(y|x)$. Note that with LDA, $p(y,x)$ is finite-dimensional (LDA requires $p(x)$ to be a mixture of two Gaussian distributions), but with LR, $p(y,x)$ is semiparametric since we do not specify the distribution $p(x)$! Thus, the set for LDA is strictly smaller than the set for LR.

Another key intuition: The bigger the model is, the easier it is to fit the data, but there might be higher variance. Thus, in real applications, LDA or LR might be better. LDA is more efficient, while it is easier to fit data for LR. But since LR can be more flexible, it is possible to overfit.

Remark 3.105. MLE under LDA model? See Homework 6. $\hat{\mu}_+$ and $\hat{\mu}_-$ are the same as under QDA.

$$\hat{\Sigma} = \frac{n_+\hat{\Sigma}_+ + n_-\hat{\Sigma}_-}{n_+ + n_-}.$$

Remark 3.106. (Implementation in R). Package e1071.

out = lda(x, y).

y.hat = predict(out)$class.

Question 3.107. Is QDA a more regularized version of linear logistic model? No. QDA's decision boundary is quadratic.

Question 3.108. Is QDA a more regularized version of quadratic logistic model?

$$P(Y=+1|\mathbf{X}=\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{x}^T A\mathbf{x} - b^T\mathbf{x} - c)}.$$

Definition 3.109. (Naive Bayes regularization). Generative modeling with an extra "class conditional independence" regularization. Naive Bayes classifiers form a family of generative classification methods that solve the issue of high dimensionality by exploiting the regularization

$$p(\mathbf{x}|Y) = \prod_{j=1}^{d} p(x_j|Y).$$

When $Y \in \{+1, -1\}$, we use the regularization

$$p(\mathbf{x}|Y=+1) = \prod_{j=1}^{d} p(x_j|Y=+1),$$

$$p(\mathbf{x}|Y=-1) = \prod_{j=1}^{d} p(x_j|Y=-1),$$

$$P(Y=+1) = \eta.$$

Thus, we only need to model $x_j|Y=+1$ and $x_j|Y=-1$. In other words, naive Bayes regularization gives us a channel to go from a multivariate model to a simpler univariate model.

Note that

$$\log\frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{P(Y=-1|\mathbf{X}=\mathbf{x})} = f(\mathbf{x})$$

is a general model, where $f(\mathbf{x})$ is a d-variate function.

Under naive Bayes regularization, we have

$$\begin{aligned}
\log\frac{P(Y=+1|\mathbf{X}=\mathbf{x})}{P(Y=-1|\mathbf{X}=\mathbf{x})} &= \log\frac{p(\mathbf{x}|Y=+1)P(Y=+1)}{p(\mathbf{x}|Y=-1)P(Y=-1)} \\
&= \log\frac{p(\mathbf{x}|Y=+1)}{p(\mathbf{x}|Y=-1)} + \log\frac{\eta}{1-\eta} \\
&= \log\prod_{j=1}^{d}\frac{p(x_j|Y=+1)}{p(x_j|Y=-1)} + \log\frac{\eta}{1-\eta} \\
&= \sum_{j=1}^{d}\log\frac{p(x_j|Y=+1)}{p(x_j|Y=-1)} + \log\frac{\eta}{1-\eta} \\
&=: \sum_{j=1}^{d} f_j(x_j) + \log\frac{\eta}{1-\eta}.
\end{aligned}$$

Definition 3.110. (LDA model under naive Bayes regularization (DLDA)).

$$\mathbf{X}|Y=+1 \sim N(\mu_+, \mathbf{\Sigma}),$$

$$\mathbf{X}|Y=-1 \sim N(\mu_-, \mathbf{\Sigma}),$$

where

$$\mathbf{\Sigma} = \begin{pmatrix} \Sigma_1^2 & & \\ & \ddots & \\ & & \sigma_d^2 \end{pmatrix},$$

$$X_j|Y=+1 \sim N(\mu_{+j}, \sigma_j^2),$$

$$X_j|Y=-1 \sim N(\mu_{-j}, \sigma_j^2),$$

so

$$p(\mathbf{x}|Y=+1) = \prod_{j=1}^{d} p(x_j|Y=+1) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_j-\mu_j)^2/2\sigma_j^2}.$$

Note that Σ is not invertible.

Definition 3.111. (MLE of DLDA). μ_+ and μ_- are the same as in LDA, since they are not affected by Σ.

$$\hat{\sigma}_j^2 = \frac{n_+\hat{\Sigma}_{+j} + n_-\hat{\Sigma}_{-j}}{n},$$

where

$$\hat{\Sigma}_{+j} = \frac{1}{n_+} \sum_{i:Y_i=+1} (X_{ij} - \hat{\mu}_{+j})^2,$$

$$\hat{\Sigma}_{-j} = \frac{1}{n_-} \sum_{i:Y_i=-1} (X_{ij} - \hat{\mu}_{-j})^2.$$

The likelihood function decouples. Ex: If X_j is a categorical variable, we can use a discrete model (ex: Bernoulli, etc.).

Question 3.112. How many free parameters are in the Naive Bayes model?

Full QDA: We have $\Sigma_+, \Sigma_-, \mu_+, \mu_-, \eta$, so the total number of parameters is $\frac{d(d+1)}{2} + \frac{d(d+1)}{2} + d + d + 1 = d(d+1) + 2d + 1$.

LDA: We have $\Sigma, \mu_+, \mu_-, \eta$, so the total number of parameters is $\frac{d(d+1)}{2} + d + d + 1 = \frac{d(d+1)}{2} + 2d + 1$.

DLDA: Same as LDA, but now we have that Σ is diagonal, so we have $\sigma_1^2, ..., \sigma_d^2, \mu_+, \mu_-, \eta$, and the number of parameters is $d + d + d + 1 = 3d + 1$.

The above covers the Gaussian case—make sure to understand this concept applied to other cases, as Naive Bayes does not have to be applied to Gaussian distributions.

Question 3.113. Y, X_1, X_2, X_3 are all binary (take values +1 and -1).

If we use the full generative model, how many parameters do we have? 15: 1 for $P(Y = +1) = \eta$, 7 for $P(X_1, X_2, X_3|Y = +1)$, 7 for $P(X_1, X_2, X_3|Y = -1)$.

If we use the Naive Bayes model, how many parameters do we have? 7: $P(Y = +1)$, $X_1|Y = +1$ and $X_1|Y = -1$, $X_2|Y = +1$ and $X_2|Y = -1$, and $X_3|Y = +1$ and $X_3|Y = -1$.

4 Exploratory analytics (unsupervised learning)

4.1 Graphical models

Refer to Homework 7 for additional coverage of graphical models.

Let us start by motivating our study of graphical models. We wish to provide a formal mechanism to represent complex phenomena. Applications range from advanced association rule mining and the study of AlphaGo, to advanced social network analysis and scientific data visualization and graph analytics.

Definition 4.1. (Graphical model). A statistical model—a set of probability distributions that are characterized by an undirected graph.

4.1.1 Gaussian graphical models

Remark 4.2. Problem setup: Let $\mathbf{X}_1, ..., \mathbf{X}_n \sim N(\mu, \Sigma), \mathbf{X} \in \mathbb{R}^d$, be random samples from a multivariate Gaussian $N(\mu, \Sigma)$. Our goal is to estimate $\Theta = \Sigma^{-1}$ (the **precision matrix**). From Homework 7, we have the theorem that

$$\Theta_{jk} = 0 \iff \mathbf{X}_j \perp\!\!\!\perp \mathbf{X}_k \mid \mathbf{X}_{\setminus\{j,k\}},$$

where $\mathbf{X}_{\setminus A} := \{\mathbf{X}_j : j \notin A\}$. We can then define an undirected graph $G = (V, E)$ based on the sparsity pattern of Θ, where $G = (V, E)$, $V \in \{1, ..., d\}$, $E \in V \times V$, and $(j, k) \in E$ if and only if $\Theta_{jk} \neq 0$. Before we introduce why this graph is interesting, we first explain how to estimate the graph based on data.

Question 4.3. How do we estimate the graph based on data? The key idea is MLE. In the following, we use the notation $\hat{\mu} = \overline{\mathbf{x}}$ and $\hat{\Sigma}_n$ to represent the Gaussian MLE.

$$\begin{aligned}
p(\mathbf{x}) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \\
&= \frac{1}{(2\pi)^{d/2}}|\Theta|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Theta(\mathbf{x}-\mu)\right) \\
\implies L_n(\Theta) &= (2\pi)^{-nd/2}|\Theta|^{n/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^T\Theta(\mathbf{x}_i-\overline{\mathbf{x}})\right) \\
\implies \ell_n(\Theta) &= \log(L_n(\Theta)) = -\frac{nd}{2}\log(2\pi) + \frac{n}{2}\log|\Theta| - \frac{\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^T\Theta(\mathbf{x}_i-\overline{\mathbf{x}})}{2} \\
&= \frac{n}{2}\log|\Theta| - \frac{n}{2}\mathrm{Tr}((\mathbf{x}_i-\overline{\mathbf{x}})^T\Theta(\mathbf{x}_i-\overline{\mathbf{x}})) + \text{constant} \\
&= \frac{n}{2}\log|\Theta| - \frac{n}{2}\mathrm{Tr}(\Theta(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^T) + \text{constant} \\
&= \frac{n}{2}\log|\Theta| - \frac{n}{2}\mathrm{Tr}(\Theta\hat{\Sigma}_n) + \text{constant}.
\end{aligned}$$

To achieve sparsity, we add L_1-penalty to get the **graphical lasso (glasso)**.

$$\hat{\Theta}^{\lambda} = \arg\min_{\Theta}\left\{\mathrm{Tr}(\Theta\hat{\Sigma}_n) - \log|\Theta| + \lambda||\Theta||_1\right\}_{\Sigma_{j,k}|\Theta_{j,k}|}.$$

Note: Why didn't we just estimate Σ, then invert it to find Θ? Σ is not invertible, and even if Σ were invertible, obtaining Θ from Σ is not sparse.

Remark 4.4. Note that Σ and Θ encode different things!

$$\Sigma_{jk} = 0 \iff \mathbf{X}_j \perp\!\!\!\perp \mathbf{X}_k.$$

$$\Theta_{jk} = 0 \iff \mathbf{X}_j \perp\!\!\!\perp \mathbf{X}_k \mid \mathbf{X}_{\setminus\{j,k\}},$$

or *conditional* independence.

Definition 4.5. (Independence). $\mathbf{X}_A$ and $\mathbf{X}_B$ are independent if, for any realizations $\mathbf{x}_A, \mathbf{x}_B$, we have

$$p(\mathbf{x}_A, \mathbf{x}_b) = p(\mathbf{x}_A)p(\mathbf{x}_B).$$

Definition 4.6. (Conditional independence). $\mathbf{X}_A$ and $\mathbf{X}_B$ are conditionally independent given $\mathbf{X}_C$ if, for any realizations $\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C$, we have

$$p(\mathbf{x}_A, \mathbf{x}_b|\mathbf{x}_C) = p(\mathbf{x}_A|p(\mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C).$$

Proposition 4.7. $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B \mid \mathbf{X}_C$ if and only if $p(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_C) = p(\mathbf{x}_A \mid \mathbf{X_C})$.

Proof. We first prove the forward direction:

$$p(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_C) = \frac{p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C)}{p(\mathbf{x}_B|\mathbf{x}_C)} = p(\mathbf{x}_A|\mathbf{x}_C).$$

Then we prove the backward direction:

$$p(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C) = p(\mathbf{x}_A|\mathbf{x}_C)p(\mathbf{x}_B|\mathbf{x}_C).$$

□

Remark 4.8. The motivating application of the Gaussian graphical model is to be able to automatically read out complex conditional independence relations based on graph separation.

Definition 4.9. (Graph separation). Recall that $G = (V, E)$ is an undirected graph, with $V = \{1, ..., d\}$ and $E \in V \times V$.

Let $A, B, C \in V$. We say C separates A and B if for all paths between nodes in A and nodes in B contain at least one node in C. In other words, removal of C makes the nodes in A and B disconnected. We denote this as $A \perp B \mid C$.

Definition 4.10. (Markov property). Our motivation: connect probability theory with graph theory. Note some notation differences used in probability theory vs. graph theory. In probability theory, we use $p(\mathbf{x})$ and denote the Markov property by $A \perp B \mid C$, and in graph theory, we use $G = (V, E)$ and denote the Markov property by $\mathbf{X}_A \perp \mathbf{X}_B \mid \mathbf{X}_C$.

Definition 4.11. (Pairwise Markov property (PMP).

$$\mathbf{X}_j \perp \mathbf{X}_k \mid \mathbf{X}_{\setminus\{j,k\}} \iff (j, k) \notin E.$$

Theorem 4.12. *The Gaussian graphical model satisfies PMP.*

Definition 4.13. (Global Markov property (GMP)). $\forall\ A, B, C, \in V$,

$$\mathbf{X}_A \perp \mathbf{X}_B \mid \mathbf{X}_C \Leftrightarrow A \perp B \mid C.$$

Theorem 4.14. *(Lauritzen 1996). If $p(\mathbf{x}) > 0\ \forall\ x$, then PMP implies GMP.*

Corollary 4.15. *The Gaussian graphical model satisfies GMP.*

Remark 4.16. GMP implies PMP.

We can observe a deeper relationship between the Gaussian graphical model and lasso regression, which we study further through the neighborhood pursuit algorithm. By GMP, we have

$$p(\mathbf{x}_1|\mathbf{x}_{\setminus\{1\}}) = p(\mathbf{x}_1|\mathbf{x}_{\text{nb}(1)}),$$

where nb(1) are all the neighbor nodes of 1. This motivates a "discriminative" approach for graphical model inference. In contrast, glasso models the full likelihood, a "generative" approach.

Question 4.17. How do we find the neighborhood of $\mathbf{X}_1$ (the nodes directly connected to $\mathbf{X}_1$)?

Recall that

$$\begin{pmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}\right).$$

$$\mathbf{X}_1|\mathbf{X}_2 \sim N(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{X}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) = N(\beta^T\mathbf{X}_2 + \beta_0, \sigma^2).$$

Thus,

$$\mathbf{X}_1 = \beta_0 + \beta^T\mathbf{X}_2 + \varepsilon,\ \varepsilon \sim N(0, \sigma^2).$$

Theorem 4.18. *(Proved in Homework 7).*

$$Var(\varepsilon_1) = \Theta_{11}^{-1},\ \beta = \frac{\Theta_{21}}{\Theta_{11}} = \frac{\Theta_{12}^T}{\Theta_{11}} \Longrightarrow supp(\beta) = nb(\mathbf{X}_1).$$

To estimate β_1, we regress $\mathbf{X}_1$ on $\mathbf{X}_2$ using lasso:

$$\hat{\beta}^{\text{Lasso}} := \arg\min_{\beta} \sum_{i=1}^{n}\left(\mathbf{X}_{i1} - \beta_0 - \sum_{j=2}^{d}\beta_j\mathbf{X}_{ij}\right)^2 + \lambda\sum_{j=2}^{d}|\beta_j|.$$

$$\text{nb}(\mathbf{X}_1) = \text{supp}(\hat{\beta}^{\text{Lasso}}).$$

We iterate through to get $\text{nb}(1), ..., \text{nb}(d)$ and merge using "AND" or "OR" rule.

Definition 4.19. (Neighborhood pursuit algorithm).
For $j = 1, ..., d$,

Regress X_j on $X_{\setminus\{j\}}$ using lasso to get $\hat{\text{nb}}(j)$.

We can then merge these neighborhoods to obtain the estimated graph.

4.1.2 Ising graphical models

Now we move to discrete data. A natural idea to estimate the graph modeling $\mathbf{X}_1, ..., \mathbf{X}_n \in \{+1, -1\}^d$ follows.

For $j = 1, ..., d$,

Regress $\mathbf{X}_j$ on $\mathbf{X}_{\setminus\{j\}}$ to get $\hat{\text{nb}}(j)$ using sparse logistic regression.

We can then merge $\hat{\text{nb}}(1), ..., \hat{\text{nb}}(d)$ using either "AND" or "OR" rule.

This motivates the Ising graphical model.

Definition 4.20. (Ising model).

$$p(\mathbf{x}) = \frac{\exp\left(\sum_{j=1}^d \beta_j \mathbf{x}_j + \sum_{k<l} \beta_{kl} \mathbf{x}_k \mathbf{x}_l\right)}{\sum_z \exp\left(\sum_{j=1}^d \beta_j \mathbf{x}_j + \sum_{k<l} \beta_{kl} \mathbf{x}_k \mathbf{x}_l\right)}.$$

The denominator arises from normalization.

Define a graph $G = (V, E)$ where $(j, k) \notin E$ if and only if $\beta_{jk} = 0$.

Theorem 4.21. *(Proved in Homework 7). $\beta_{kl} = 0$ if and only if $\mathbf{X}_k \perp\!\!\!\perp \mathbf{X}_l \mid \mathbf{X}_{\setminus\{k,l\}}$ (the pairwise Markov property).*

Therefore, the graph can be inferred using nodewise sparse logistic regression.

Corollary 4.22. *If $p(\mathbf{x}) > 0$, then the Ising graphical model satisfies GMP.*

4.2 Clustering

Definition 4.23. (Clustering). A classification problem with hidden/unobserved/latent class labels. Our goal is to recover the latent class variables, based on the input/features.

$X_1, ..., X_n$ are features (input). $Z_1, ..., Z_n$ are labels (output)—these are latent.

4.2.1 Latent variable models and mixture models

Definition 4.24. (Latent variable model). Latent variable modeling serves as a foundational role in modern statistics (deep hierarchical modeling) and machine learning (mixture modeling), allowing us to stack component models together to make them "deep." Relationship with deep learning.

Example 4.25. (Mixture of two Gaussians). Flip a coin with outcome Z: $Z = 1$ if heads, $Z = 2$ if tails. $Z \sim Ber(\eta)$, i.e. $P(Z = 1) = \eta$.

$$X|Z = 1 \sim N(\mu_1, 1)$$

$$X|Z = 2 \sim N(\mu_2, 1).$$

Only given $X_1, ..., X_n$ (observed random samples), how do we infer η, μ_1, μ_2 and $Z_1, ..., Z_n$ (latent variables)? Note that if $Z_1, ..., Z_n$ were given, then this is simply a classification problem, and $N(\mu_1, 1)$ and $N(\mu_2, 1)$ would be the generative models for the population.

Clustering lets us infer $P(Z_i = 1|X_1, ..., X_n)$—the deterministic probability that the ith data is sampled from the subpopulation 1. This is referred to as **soft clustering**. This problem is strictly more challenging than classification: first, we do not know $Z_1, ..., Z_n$. We still need to infer parameters η, μ_1, μ_2. Then, we actually have to infer $Z_1, ..., Z_n$—what we care about.

Remark 4.26. (Statistical model). Let $\theta = (\eta, \mu_1, \mu_2)^T$.

$$\begin{aligned} p_\theta(x) &= p_\theta(x, Z = 1) + p_\theta(x, Z = 2) \\ &= p_\theta(x|Z = 1)\eta + p_\theta(x|Z = 2)(1 - \eta) \\ &= \eta p_{\mu_1}(x) + (1 - \eta)p_{\mu_2}(x), \end{aligned}$$

where $p_{\mu_1} \sim N(\mu_1, 1)$ and $p_{\mu_2} \sim N(\mu_2, 1)$, so $p_\theta(x)$ is a mixture of densities. η is the mixing coefficient. The model (a family of density functions):

$$\left\{p_\theta(x) = \eta p_{\mu_1}(x) + (1 - \eta)p_{\mu_2}(x)\right\}_{\eta,\mu_1,\mu_2}.$$

Remark 4.27. Estimation: MLE.

$$\begin{aligned} \ell(\mathbf{\Psi}) &= \sum_{i=1}^{n} \log p_{\mathbf{\Psi}}(X_i) \\ &= \sum_{i=1}^{n} \log[\eta p_{\mu_1}(X_i) + (1 - \eta)p_{\mu_2}(X_i)] \\ &= \sum_{i=1}^{n} \log\left[\eta \frac{1}{\sqrt{2\pi}} e^{-(X_i-\mu_1)^2/2} + (1 - \eta)\frac{1}{\sqrt{2\pi}} e^{-(X_i-\mu_2)^2/2}\right]. \end{aligned}$$

Note that this is nonconvex due to the "log-sum" structure. This unfortunately shows up whenever we deal with latent variables.

Computational solutions include:

- Gradient-based method (ex: stochastic gradient descent). Performance is not good since it does not find the true structure (it is too generic and doesn't utilize the problem structure enough).
- Convex/SDP relaxation (see ORF 363).
- Expectation-Maximization (EM) algorithm—the focus of our class. The more general version is the Minorization-Maximization strategy. This method fully utilizes problem structure and is scalable (1K, 10K parameters—still very fast).

Definition 4.28. (Mixture model). A set of densities that can be represented as a convex combination of a set of component densities.

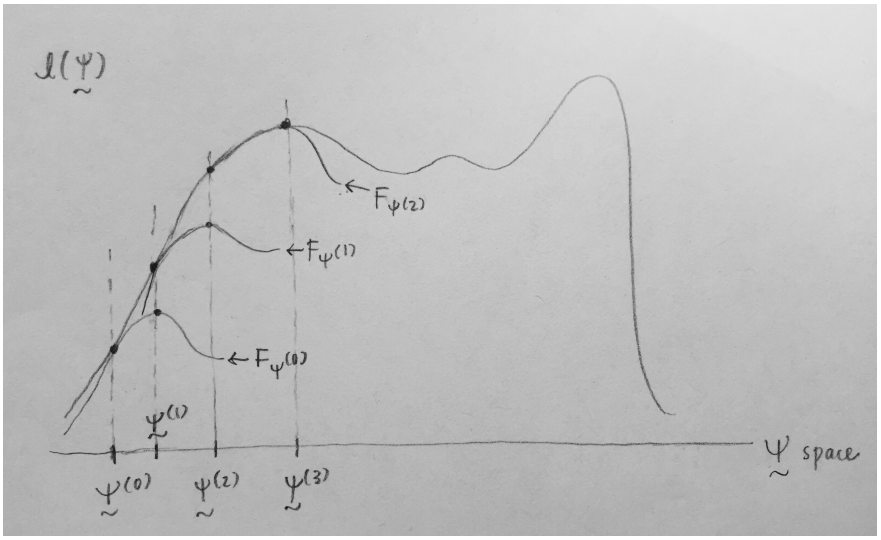
Convex: $\eta \geq 0, 1 - \eta \geq 0$. $\eta + 1 - \eta = 1$—lies on simplex.

Finite combination $\rightarrow$ **Finite mixture model**.

Infinite combination $\rightarrow$ **Infinite mixture model**.

4.2.2 EM algorithm

Definition 4.29. (Expectation-maximization (EM) algorithm). The main idea: Minorization-Maximization. See the following figure for the geometric intuition behind the EM algorithm and how EM can find one local optimum.



Key: Given a current parameter $\mathbf{\Psi}^{(\text{old})}$, find a lower bound function $F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi})$ such that (lower bound)

$$F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi}) \leq \ell(\mathbf{\Psi}) \ \forall \ \mathbf{\Psi},$$

(tight at $\mathbf{\Psi}^{(\text{old})}$)

$$F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi}^{(\text{old})}) = \ell(\mathbf{\Psi}^{(\text{old})}).$$

$F_{\mathbf{\Psi}^{(\text{old})}}(\cdot)$ is easy to be optimized:

$$\mathbf{\Psi}^{(\text{new})} \leftarrow \arg\max_{\mathbf{\Psi}} F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi}).$$

We repeatedly apply the above strategy to get a solution sequence $\mathbf{\Psi}^{(0)}, \mathbf{\Psi}^{(1)}, ..., \mathbf{\Psi}^{(k)}, ...$, and we monitor for convergence of $\ell(\mathbf{\Psi}^{(0)}), \ell(\mathbf{\Psi}^{(1)}), ..., \ell(\mathbf{\Psi}^{(k)}), ...$ We try multiple initializations and pick the best one.

Theorem 4.30. *(Convergence of EM).*

$$\ell(\mathbf{\Psi}^{(0)}) \leq \ell(\mathbf{\Psi}^{(1)}) \leq \ell(\mathbf{\Psi}^{(2)}) \leq ...$$

In other words, if $\ell(\mathbf{\Psi}) \leq C \ \forall \ \mathbf{\Psi}$, then the EM algorithm converges to a local maximum.

Proof. For all t,

$$\ell(\mathbf{\Psi}^{(t)}) = F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}^{(t)}) \leq F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}^{(t+1)}) \leq \ell(\mathbf{\Psi}^{(t+1)}).$$

Thus, since $\ell(\mathbf{\Psi}^{(0)}) \leq \ell(\mathbf{\Psi}^{(1)}) \leq ...$ is a nondecreasing sequence and $\ell(\mathbf{\Psi}) \leq C$, $\{\ell(\mathbf{\Psi}^{(t)})\}_{t=0,1,2,...}$ converges by the monotone convergence theorem.

The stopping criterion is

$$|\ell(\mathbf{\Psi}^{(t+1)}) - \ell(\mathbf{\Psi}^{(t)})| < \varepsilon.$$

□

Remark 4.31. Given the current parameter configuration $\mathbf{\Psi}^{(\text{old})}$, how shall we construct a lower bound function $F_{\mathbf{\Psi}^{(\text{old})}}(\cdot)$ such that $F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi}) \leq \ell(\mathbf{\Psi})$ and $F_{\mathbf{\Psi}^{(\text{old})}}(\mathbf{\Psi}^{(\text{old})}) = \ell(\mathbf{\Psi}^{(\text{old})})$?

The key idea is to use Jensen's inequality:

$$\log E[X] \geq E[\log X].$$

The geometric intuition lies in the concavity of the log function.

Definition 4.32. (EM algorithm cont.).

Initialize $\mathbf{\Psi}^{(0)}$.

For $t = 0, 1, 2, ...,$

- E-step: Construct $F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi})$.
- M-step: Maximize $F_{\mathbf{\Psi}}(\mathbf{\Psi})$. $\mathbf{\Psi}^{(t+1)} = \arg\max_{\mathbf{\Psi}} F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi})$.

Construction/E-step. We start by defining

$$\gamma_i(z_i) = P_{\mathbf{\Psi}^{(t)}}(Z_i = z_i | \mathbf{X_i}),$$

the probability that, given $\mathbf{X_i}$ and $\mathbf{\Psi}^{(t)}$, $\mathbf{X_i}$ belongs to the cluster z_i. Fix $\mathbf{X_i}$, $\gamma_i(z_i) \geq 0$, and $\sum_{z_i} \gamma_i(z_i) = 1$.

$$\begin{aligned}
\ell(\mathbf{\Psi}) &= \sum_{i=1}^{n} \log p_{\mathbf{\Psi}}(\mathbf{X_i}) \\
&= \sum_{i=1}^{n} \log \left[\sum_{z_i} p_{\mathbf{\Psi}}(\mathbf{X_i}, Z_i = z_i) \right] \\
&= \sum_{i=1}^{n} \log \left[\sum_{z_i} \gamma_i(z_i) \cdot \frac{p_{\mathbf{\Psi}}(\mathbf{X_i}, Z_i = z_i)}{\gamma_i(z_i)} \right] \\
&\geq \sum_{i=1}^{n} \sum_{z_i} \gamma_i(z_i) \cdot \log \left[\frac{p_{\mathbf{\Psi}}(\mathbf{X_i}, Z_i = z_i)}{\gamma_i(z_i)} \right] =: F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}).
\end{aligned}$$

The name "expectation" comes in because the key to the construction step is to evaluate

$$\gamma_i(z_i) = P_{\mathbf{\Psi}}(Z_i = z_i | \mathbf{X_i}) = E[I(Z_i = z_i) | \mathbf{X_i}].$$

Note that, by construction, $F_{\mathbf{\Psi}^{(t)}} \leq \ell(\mathbf{\Psi})$. Also,

$$\begin{aligned}
F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}^{(t)}) &:= \sum_{i=1}^{n} \sum_{z_i} \gamma_i(z_i) \left(\log \frac{p_{\mathbf{\Psi}^{(t)}}(\mathbf{X_i}, Z_i = z_i)}{\gamma_i(z_i)} \right) \\
&= \sum_{i=1}^{n} \sum_{z_i} \gamma_i(z_i) \cdot \log \frac{P_{\mathbf{\Psi}^{(t)}}(Z_i = z_i | \mathbf{X_i}) p_{\mathbf{\Psi}^{(t)}}(\mathbf{X_i})}{\gamma_i(z_i)} \\
&= \sum_{i=1}^{n} \log p_{\mathbf{\Psi}^{(t)}}(\mathbf{X_i}) = \ell(\mathbf{\Psi}^{(t)}).
\end{aligned}$$

Maximization/M-step. Once $F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi})$ is constructed, optimize it with respect to $\mathbf{\Psi}$:

$$\mathbf{\Psi}^{(t+1)} \leftarrow \arg\max_{\mathbf{\Psi}} F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}).$$

Example 4.33. (EM algorithm for finite mixture model).

$$p_{\mathbf{\Psi}}(\mathbf{x}) = \sum_{j=0}^{K-1} p_{\theta_j}(x_j) \cdot \eta_j$$

where $\eta_i \geq 0$ and $\sum_{j=0}^{k-1} \eta_j = 1$. The EM algorithm proceeds as follows:

Initialize

$$\mathbf{\Psi}^{(0)} = \left(\{\eta_j\}_{j=0}^{K-1},\ \{\theta_j\}_{j=0}^{K-1} \right).$$

In total, there are K clusters. Note that η_{K-1} is effectively 1 since $\sum_{j=0}^{K-1} \eta_j = 1$. γ_{ij} represents that the ith data belongs to the jth cluster.

For $t = 0, 1, 2, ...,$

E-step:

$$\gamma_{ij}^{(t+1)} := P_{\mathbf{\Psi}^{(t)}}(Z_i = j | \mathbf{X_j}) = \frac{P_{\theta_j^{(t)}}(\mathbf{X_i} | Z_i = j) \cdot \eta_j^{(t)}}{\sum_{j=0}^{k-1} P_{\theta_j^{(t)}}(\mathbf{X_i} | Z_i = j) \cdot \eta_j^{(t)}}.$$

M-step:

$$\eta_j^{(t+1)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)},$$

$$\theta_j^{(t+1)} \leftarrow \arg\max_{\theta_j} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \log P_{\theta_j}(\mathbf{X_i}).$$

Proof. E-step is obvious. We analyze M-step. Recall that $\eta_j = P(Z = j)$ and that there are K clusters in total (i.e., K possible values that Z can equal).

$$F_{\mathbf{\Psi}^{(t)}}(\mathbf{\Psi}) = \sum_{i=1}^{n} \sum_{j=0}^{K-1} \gamma_{ij}^{(t+1)} \log \left[\frac{P_{\theta_j}(\mathbf{X_i} | Z_i = j) \eta_j}{\gamma_{ij}^{(t+1)}} \right].$$

First, we optimize θ_j:

$$\begin{aligned}
\theta_j^{(t+1)} &:= \arg\max_{\theta_j} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \log p_{\theta_j, \eta_j}(\mathbf{X_i} | Z_i = j) \eta_j \\
&= \arg\max_{\theta_j} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \log p_{\theta_j}(\mathbf{X_i} | Z_i = j) \\
&= \arg\max_{\theta_j} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \log p_{\theta_j}(\mathbf{X_i}).
\end{aligned}$$

Next, we examine η:

$$\eta^{(t+1)} = \arg\max_{\eta_0,...,\eta_{K-1}} \sum_{i=1}^{n}\sum_{j=0}^{K-1} \gamma_{ij}^{(t+1)} \log \eta_j \text{ s.t. } \sum_{j=0}^{K-1} \eta_j = 1.$$

We optimize η using the Lagrangian form:

$$\mathcal{L}(\eta, \alpha) = \sum_{i=1}^{n}\sum_{j=0}^{K-1} \gamma_{ij}^{(t+1)} \log \eta_j - \alpha \left(\sum_{j=0}^{K-1} \eta_j - 1 \right).$$

$$0 = \frac{\partial \mathcal{L}(\eta, \alpha)}{\partial \eta_j} = \frac{1}{\hat{\eta}_j} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)} - \alpha.$$

Since

$$1 = \sum_{j=0}^{K-1} \hat{\eta}_j = \frac{1}{\alpha} \sum_{j=0}^{K-1}\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} = \frac{n}{\alpha},$$

$$\eta_j = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)}.$$

□

Example 4.34. (EM algorithm for mixture of K Gaussians).

$$Z \sim \text{Multi}(1, \eta_0, ..., \eta_{K-1}).$$

$$X|Z = j \sim N(\mu_j, \Sigma_j) \text{ for } j = 0, 1, ..., K-1.$$

We want to infer $\theta := (\eta_0, ..., \eta_{K-1}, \mu_0, ..., \mu_{K-1}, \Sigma_0, ..., \Sigma_{K-1})$.

Initialize

$$\theta^{(0)} = (\eta_0^{(0)}, ..., \eta_{K-1}^{(0)}, \mu_0^{(0)}, ..., \mu_{K-1}^{(0)}, \Sigma_0^{(0)}, ..., \Sigma_{K-1}^{(0)}).$$

For $t = 0, 1, 2, ...,$

E-step:

$$\gamma_{ij}^{(t+1)} = \frac{\eta_j^{(t)} p_{\mu_j^{(t)}, \Sigma_j^{(t)}}(\mathbf{X_i})}{\sum_{l=0}^{K-1} \eta_l^{(t)} p_{\mu_l^{(t)}, \Sigma_l^{(t)}}(\mathbf{X_i})}, \ i = 1, ..., n, \ j = 0, ..., K-1.$$

Here, $p_{\mu_j^{(t)}, \Sigma_j^{(t)}} \sim N(\mu_j^{(t)}, \Sigma_j^{(t)})$.

M-step:

$$\eta_j^{(t+1)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \gamma_{ij}^{(t+1)},$$

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \mathbf{X_i}}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}},$$

$$\Sigma_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} (\mathbf{X_i} - \mu_j^{(t+1)})(\mathbf{X_i} - \mu_j^{(t+1)})^T}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}}.$$

Repeat until convergence.

4.2.3 K-means algorithm

Definition 4.35. (K-means algorithm). The limiting procedure by applying the EM algorithm on a sequence of degenerate (variance shrinks to zero) mixtures of K isotropic/spherical Gaussians. An **isotropic (spherical)** Gaussian distribution is a Gaussian distribution with covariance matrix in the form $\sigma^2 I_d$ (regularized model).

$$\lim_{\sigma^2 \to 0} \mathcal{A}_{\sigma^2} = \mathcal{A},$$

where $\mathcal{A}_{\sigma^2}$ is EM, and $\mathcal{A}$ is K-means.

Example 4.36. (EM for mixture of K-spherical Gaussians (with known σ^2)).

E-step:

$$\gamma_{ij} \leftarrow \frac{\eta_j p_{\mu_j,\sigma^2}(\mathbf{x})}{\sum_{l=0}^{K-1} \eta_l p_{\mu_l,\sigma^2}(\mathbf{x})} = \frac{\eta_j \exp\left(-\frac{1}{2\sigma^2}||\mathbf{X}_i - \mu_j||_2^2\right)}{\sum_{l=0}^{k-1} \eta_l \exp\left(-\frac{1}{2\sigma^2}||\mathbf{X}_i - \mu_l||_2^2\right)}.$$

Assuming no ties, when $\sigma^2 \to 0$,

$$\gamma_{ij}^{(t+1)} = \begin{cases} 1 \text{ if } ||\mathbf{X}_i - \mu_j||_2^2 < ||\mathbf{X}_i - \mu_l||_2^2 \ \forall\ l \neq j \\ 0 \text{ otherwise} \end{cases}$$

Thus, $P(Z_i = j|\mathbf{X}_i)$ is either 0 or 1, so we can define

$$\gamma_{ij} = I(Z_i = j).$$

This is **hard assignment** of data i to class j, as opposed to the soft assignment of the EM algorithm.
M-step:

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{\gamma} \gamma_{ij}^{(t+1)} \mathbf{X}_i}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}} = \frac{\sum_{i=1}^{\gamma} I(Z_i = j)\mathbf{X}_i}{\sum_{i=1}^{\gamma} I(Z_i = j)}.$$

Remark 4.37. In the E-step, we only need information on $\mu_0, ..., \mu_{K-1}$. We do not need information on $\eta_0, ..., \eta_{K-1}$. In the M-step, we only need to update $\mu_0, ..., \mu_{K-1}$ based on the current γ_{ij}. We calculate μ_j by averaging all the data points assigned to this particular class in the previous E-step.

Remark 4.38. (Formal description of K-means algorithm).
(Randomly) initialize $\mu_0^{(0)}, ..., \mu_{k-1}^{(0)}$.

For $t = 0, 1, 2, ...,$

E-step:

$$\gamma_{ij}^{(t+1)} = \begin{cases} 1 \text{ if } ||\mathbf{X}_i - \mu_j||_2^2 < ||\mathbf{X}_i - \mu_l||_2^2 \ \forall\ l \neq j \\ 0 \text{ otherwise.} \end{cases}$$

M-step:

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)} \mathbf{X}_i}{\sum_{i=1}^{n} \gamma_{ij}^{(t+1)}} \text{ for } j = 0, ..., K-1.$$

Repeat until (guaranteed) convergence.

Remark 4.39. Thus far, we have covered the model-based approach to the K-means algorithm. We now summarize a risk minimiation/model-free approach. The key intuition is that the K-means algorithm can be viewed as a **block coordinate minimization** algorithm of an empirical risk function.

Definition 4.40. (Block coordinate minimization). The risk minimization view of K-means is to view it as a block coordinate minimization of an empirical risk function. The intuition of block coordinate minimization follows:

Objective:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}, \mathbf{y}).$$

Initialize $\mathbf{x}^{(0)}, \mathbf{y}^{(0)}$.

For $t = 0, 1, 2, ...,$

$$\mathbf{x}^{(t+1)} \leftarrow \arg\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^{(t)},$$

$$\mathbf{y}^{(t+1)} \leftarrow \arg\min_{\mathbf{y}} f(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t)}).$$

Remark 4.41. The goal of K-means is to choose K clustering centers to minimize

$$\hat{R}(\mu_0, ..., \mu_{K-1}) = \frac{1}{n} \sum_{i=1}^{n} \min_{0 \le j \le K-1} ||\mathbf{X}_i - \mu_j||_2^2.$$

The K-means empirical risk function is very similar to the squared loss function, but now with an extra minimization step. Note that this extra minimization is a computational challenge.

The population risk is

$$R(\mu_0, ..., \mu_{K-1}) := E\left(\min_{0 \le j \le K-1} ||\mathbf{X}_i - \mu_j||_2^2 \right).$$

Remark 4.42. An equivalent formulation of K-means is

$$\min_{A_i \in \{0,...,K-1\},\ \mu_0,...,\mu_{K-1}} \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{X}_i - \mu_{A_i}||_2^2.$$

K-means is the block coordinate descent algorithm for the above function.
Loop:

Step 1: Provide $\mu_0, ..., \mu_{K-1}$. Update $A_1, ..., A_n$. (A_i is just like γ: $A_i = j$ means that the ith data belongs to the jth cluster.)

Step 2: Given $A_1, ..., A_n$, update $\mu_0, ..., \mu_{K-1}$.

Remark 4.43. A more compact form of the K-means algorithm follows.
(Random) initialization.

For $t = 0, 1, 2, ...,$

$$Z_i^{(t+1)} \leftarrow \arg\min_{0 \le j \le K-1} ||\mathbf{X}_i - \mu_j^{(t)}||_2^2 \text{ for } i = 1, ..., n.$$

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^n I(Z_i^{(t+1)} = j)\mathbf{X}_i}{\sum_{i=1}^n I(Z_i^{(t+1)} = j)} \text{ for } j = 0, ..., K-1.$$

Note that previously, we derived the K-means algorithm from EM, so it looked very similar to EM. Now, we have derived the full-fledged K-means algorithm.

4.3 Tree, bagging, and random forest

4.3.1 Tree-based method

Definition 4.44. (Tree-based regression method). The population L_2-risk is

$$R(f) = E_{\mathbf{X},Y}(Y - f(\mathbf{X}))^2,$$

where $Y \in \mathbb{R}$, $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^d$.

$$\min_{f \in \mathcal{F}} R(f),$$

$$\mathcal{F} = \left\{ f(\mathbf{x}) = \sum_{j=1}^M \beta_j I(\mathbf{x} \in \mathbb{R}_j) \right\},$$

where $R_1, ..., R_M$ form a "tree partition" of the input space $\mathcal{X}$.

Definition 4.45. (Tree partition). A partition of the input space $\mathcal{X}$ that can be formed by recursively applying the following two rules:

- Choose a cell of the current partition.
- Split the chosen cell into two cells by binary splitting along one dimension (or variable).

Remark 4.46. Note that we cannot find a finite parameter to represent the true model $\mathcal{F}$, since M can be arbitrarily large, and we can do binary splits arbitrarily (make the model arbitrarily refined). Thus, the tree model is nonparametric. We take on a perspective similar to how we treated graphical models. Even though we seemingly are using a new function class, we still go through a similar path of regression, regularization, etc.

Moving to the sample version, given random samples $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n) \sim \mathcal{P}$, how do we fit a tree model? Empirical risk minimization.

The empirical risk is

$$\hat{R}(f) := \frac{1}{n} \sum_{i=1}^n |Y_i - f(\mathbf{X}_i)|^2.$$

We seek to minimize $R(f)$:

$$\hat{f} := \arg\min_{f \in \mathcal{F}} \hat{R}(f).$$

However, we can perfectly fit all the data via arbitrary splitting that perfectly fits all n data! Thus, we need to regularize:

$$\mathcal{F}_{K_{\min}} := \left\{ f(\mathbf{x}) = \sum_{j=1}^M \beta_j I(\mathbf{x} \in R_j) \in \mathcal{F}, \text{ and any } R_j \text{ contains at least } K_{\min} \text{ data points} \right\}.$$

For instance, we can have $K_{\min} = 5$.

Since this optimization is combinatoric (NP-hard), we discuss the greedy algorithm.

Definition 4.47. (Greedy algorithm for tree partitioning). Grow a tree by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $K_{\min}$ is reached:

- Pick a variable/split-point that decreases $\hat{R}(f)$ the most.
- Split the node into two child nodes.

How do we choose $K_{\min}$? Cross-validation. In applications, we generally choose $K_{\min} = 5$. However, overfitting still occurs. Thus, we perform the following steps to regularize/prune the tree:

- Given a fully grown tree T_0, find an internal node (or non-terminal node) that, after collapsing the subtree into itself, will increase $\hat{R}(f)$ the least.
- Collapse the subtree into the chosen internal node. We get a new tree T_1. Repeating this process, we get a sequence of trees $T_0, T_1, T_2, ..., T_K$. (The last tree has 1 node.)
- Pick one tree by minimizing
$$\hat{R}(\hat{f}_T) + \lambda|T|,$$
where $\hat{f}_T$ is the regression function fitted on T, λ is a tuning parameter chosen by cross-validation, and $|T|$ is the number of nodes in T.

$$\hat{T}^* = \arg \min_{T \in \{T_0,...,T_K\}} \hat{R}(\hat{f}_T) + \lambda|T|,$$

where $\hat{T}^*$ is the chosen tree.

An overview of the algorithm follows: first, we grow the tree–an exploratory, myopic process. Then, we prune the tree back and pick the best tree from the sequence, using $\hat{R}(\hat{f}_T) + \lambda|T|$. These two processes work in opposite directions.

Remark 4.48. What are some pros and cons of the tree-based method? Pros: Simple and interpretable. Cons: Fitted functions are non-smooth, so the greedy algorithm is hard to analyze. We address this by discussing two extensions—bagging and random forest.

To address the drawbacks of trees, our first extension is bagging (bootstrap aggregation). This does not work particularly well in applications because of the lack of diversity involved in data. Imagine an analogy with a dictator versus a mixture of experts. Generally, the latter will experience better perofrmance with the assumption of diversity (if there is not enough diversity, then we might experience selection bias, in which case, the dictator might actually be better).

Thus, this brings us to our second extension—the random forest. An industrial standard, it is one of the best off-the-shelf solvers/predictors.

4.3.2 Extension 1: Bagging/bootstrapping

Definition 4.49. (Bagging (boostrap aggregation) algorithm). The key to bootstrapping is uniform sampling with replacement.

Bagging involves using the tree-based method on the first, second, ..., Bth datasets, then averaging the respective regression functions.

For $b = 1, ..., B$:

- Draw a boostrap sample $Z_{1:n}^{*b}$ of size n from $Z_{1:n}$.
- Fit a regression tree on the bootstrapped data $\hat{f}^b$ (with minimum node size K_{min} and no pruning).

Output:

$$\hat{f}(\mathbf{x})^{\text{Bagging}} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(\mathbf{x}).$$

Note that we can have a very large value of B (for instance, $B = 20,000$). So, essentially, we now can work with 20,000 datasets that are different because they are randomly generated (note that they are still not i.i.d., but rather, conditionally independent, given their uniform sampling without replacement from $Z_{1:n}$). With $K \cdot B$ number of splits, we can achieve a very fine regression function through bagging.

Remark 4.50. Bagging vs. original tree.

- $\hat{f}^{\text{Bagging}}(\mathbf{x})$ has the same bias as $\hat{f}^b(\mathbf{x})$ but potentially smaller variance. (Some situations have smaller variances, while others do not.)
- The larger B is, the better, but with diminishing returns. In real applications, $B = 20,000$ is the rule of thumb.
- Bagging works well if $\hat{f}^1(\mathbf{x}), \hat{f}^2(\mathbf{x}), ..., \hat{f}^B(\mathbf{x})$ are decorrelated, i.e. if the tree structures are very diversified. This motivates the random forest.

4.3.3 Extension 2: Random forest

Definition 4.51. (Random forest algorithm).
For $b = 1, ..., B$:

- Draw a bootstrap sample $Z_{1:n}^{*b}$ of size n from $Z_{1:n}$.
- Fit a regression tree to the bootstrapped data by recursively repeating the following steps for each terminal node of the tree, until the minimum node size K_{min} is reached.
  - Select m variables at random from d variables. (This step adds a lot of randomness and allows $\hat{f}^1, \hat{f}^2, ..., \hat{f}^B$ to be much more diversified and for the average to be "good.")
  - Pick the best variable/split-point among the m.

– Split the node into two daughter nodes.

Output the ensemble of fitted tree functions, $\hat{f}^1, ..., \hat{f}^B$. The final random forest estimator is

$$\hat{f}^{\text{RF}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x).$$

4.4 Neural networks and deep learning

Definition 4.52. (Neural computation unit). A linear function with a possibly nonlinear transformation

$$f(\mathbf{x}) = \sigma(\beta^T \mathbf{x} + \beta_0).$$

Some choices of σ include:

- Linear computation unit: $\sigma(t) = t$.
- Thresholded linear computation unit: $\sigma(t) = \begin{cases} +1 \text{ if } t \geq 0 \\ -1 \text{ otherwise.} \end{cases}$
  Note that this is similar to an actual neuron (it either fires or does not fire).
- Rectified linear unit (ReLU): $\sigma(t) = \begin{cases} t \text{ if } t \geq 0 \\ 0 \text{ otherwise} \end{cases}$
- Sigmoid computation unit: $\sigma(t) = \frac{1}{1+e^{-t}}$.

In practice today, ReLU is the industrial standard due to its convex nature.

Definition 4.53. (Neural network model). A statistical model that depends on the funciton class

$$\mathcal{F} := \{f : \mathcal{X} \to \mathbb{R}, f \text{ can be represented by connecting different neural computation units}\}.$$

(For instance, the input can be another unit.) If the number of units is finite, then we refer to the neural network as a **parametric neural network**.

Remark 4.54. A neural network can be represented as a recursive composition of the basic computation units.

Definition 4.55. (Depth of neural network model/function class). The number of layers of recursive composition.

Definition 4.56. (Deep neural network). A neural network with depth of at least 3.

Definition 4.57. (Deep learning). Machine learning methods using the deep neural network. For instance, in logistic regression, we model

$$P_f(Y = +1 | \mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}}.$$

If f belongs to a deep network class, then this is deep logistic regression/deep learning.

Definition 4.58. (Deep feature). The output layer of a neural network can always be represented by a function

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{k} \beta_j \phi_j(\mathbf{x}).$$

$\phi_1(\mathbf{x}), ..., \phi_k(\mathbf{x})$ are deep features.

Remark 4.59. How do we fit the deep network to training data? Consider an example of a classification problem with $(Y_1, \mathbf{X}_1), ..., (Y_n, \mathbf{X}_n)$, $Y_i \in \{+1, -1\}$.

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \log(1 + e^{-Y_i f(\mathbf{x}_i)}),$$

where $\mathcal{F}$ is the deep network model.

Model fitting with MLE poses a computational challenge since it is a highly nonconvex, nonlinear optimization with potentially billions of decision variables/parameters. The solution is stochastic gradient descent and "algorithmic regularization" (similar intuition as used in random forest).

Deep learning is a big deal now that big training data is available. (Otherwise, with smaller datasets, it would be very easy to overfit the neural network since it is so flexible.) Stochastic gradient descent and algorithmic regularization lead to applications related to transfer learning and information retrieval.

Definition 4.60. (Connectivism). Different ways to design networks—an area of research. Once we have a network model, we can seek optimization, for the purpose of generalizing well. Applications include computer vision and text analysis.

Remark 4.61. Note that deep learning is mainly useful for zero-noise applications, such as image classification.

4.5 Kernel method

Definition 4.62. (Kernel method). A nonparametric extension of (more flexible version of) SVM.

$$\min_{\beta} \sum_{i=1}^{n} (1 - Y_i f(X_i))_+ + ||f||_H,$$

where H is the reproducing kernel Hilbert space. Recall that with SVM, $f(X_i) = \beta^T X_i$ and $||f||_H$ is replaced with $\lambda||\beta||_2^2$.

Definition 4.63. (Reproducing kernel Hilbert space (RKHS)). Hilbert space with a "reproducing kernel" property. (An infinite-dimensional extension of Euclidean space $\mathbb{R}^d \to \mathbb{R}^\infty$, constrained with the reproducing kernel property.) We can also replace $(1 - Y_i f(X_i))_+$ with a different loss function. For instance, with logistic loss, this becomes

$$\min_{\beta} \sum_{i=1}^{n} \ell(f(X_i), Y_i) + ||f||_H.$$

Remark 4.64. The kernel method is useful for high-noise applications. Thus, neural networks and the kernel method are both great for big datasets and are actually complementary to each other.

5 Analysis of small data: Bayesian inference

We cover the fundamental difference between Bayesian methods and non-Bayesian methods, which are not "competing ideas" but are just solving very different problems. We illustrate this concept with a story of two consulting firms: Company 1 offers frequentist guarantees of a procedure, while Company 2 offers subject belief manipulation (Company 2 asks the client what their belief is, then uses Bayes formula to offer the best possible belief update, given the specific data and their belief). These two approaches—frequentist inference and Bayesian inference—are like apples and oranges: we should not compare them.