CHEATBOOK: STAT 24510 STATISTICAL THEORY & METHODS IIA

YIDUAN ZHENG

Contents

Date: Class: January - March, 2023.
First final version: March 14, 2023.

Notes

This cheatbook organizes notes from STAT 24510 Statistical Theory & Methods IIa taught by Professor Chao Gao in Winter 2023.

Several disclaimers:

- The cheatbook's notation does differ from class at times, especially those relating to matrices and vectors in the "Linear Regression" section, where we used copious shorthand.
- The cheatbook's vocabularies also differ at time (for example, we consistently use "normal" instead of "Gaussian" distribution). This was done to better align with the STAT 244 cheatbook and other online resources and is not a result of personal preference.
- I try to keep the cheatbook's materials aligned to the class as close as possible, but I could not resist incorporating some vocabularies from Honors Econometrics and some simplifying notations from Honors Combinatorics in the "Linear Regression" section.

We assume no responsibility for any potential confusion caused by the above items.

For reference,

- 244 cheatbook refers to "Cheatbook: STAT 24410 Statistical Theory & Methods Ia",
- 271 cheatbook refers to "Cheatbook: MATH 27100 Measure & Integration".

For errata or suggestions, please email yzheng27@uchicago.edu.

1. Confidence Interval

In this section, we will first discuss confidence interval construction with respect to Poisson distribution and exponential distribution, though the techniques discussed can sometimes be applied to other distributions as well.

1.1. **Poisson Distribution.**

Let us start with a review on properties of Poission distribution.

Definition 1.1. A random variable (RV) X follows a *Poisson distribution* with *rate* parameter λ if we can write its probability mass function (PMF) as

$$\mathbb{P}(X = k) = e^{-\lambda}\frac{\lambda^k}{k!}$$

for any $k \geq 0$.

Let us first check that this is a well-defined distribution by ensuring the probabilities sum to 1:

$$\sum_{k=0}^{\infty} \mathbb{P}(X = k) = \sum_{k=0}^{\infty} e^{-\lambda}\frac{\lambda^k}{k!} = e^{-\lambda}\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} = e^{-\lambda}\cdot e^{\lambda} = 1.$$

Theorem 1.2. *Let $X \sim Poisson(\lambda)$. Then*

$$\mathbb{E}[X] = \mathbb{V}[X] = \lambda.$$

Proof. For the expectation,

$$\begin{aligned} E[X] &= \sum_{k=0}^{\infty} k \cdot \mathbb{P}(X = k) \\ &= \sum_{k=1}^{\infty} k \cdot e^{-\lambda}\frac{\lambda^k}{k!} \\ &= \lambda e^{-\lambda}\sum_{k=1}^{\infty}\frac{\lambda^{k-1}}{(k-1)!} \\ &= \lambda e^{-\lambda}\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} \\ &= \lambda e^{-\lambda}e^{\lambda} = \lambda. \end{aligned}$$

For the variance,

$$\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

We already know the second term on the RHS is λ^2, so it remains to calculate the first. We will apply an 'add and subtract' trick:

$$\begin{aligned}
E[X^2] &= \sum_{k=0}^{\infty} k^2 \cdot \mathbb{P}(X=k) \\
&= \sum_{k=0}^{\infty} (k^2 - k + k) \cdot \mathbb{P}(X=k) \\
&= \sum_{k=0}^{\infty} (k^2 - k) \cdot \mathbb{P}(X=k) + \sum_{k=0}^{\infty} k \cdot \mathbb{P}(X=k) \\
&= \sum_{k=0}^{\infty} (k^2 - k) \cdot e^{-\lambda} \frac{\lambda^k}{k!} + \mathbb{E}[X] \\
&= \lambda^2 e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^{k-2}}{(k-2)!} + \lambda \\
&= \lambda^2 e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} + \lambda \\
&= \lambda^2 e^{-\lambda} e^{\lambda} + \lambda \\
&= \lambda^2 + \lambda.
\end{aligned}$$

Combined we have

$$\mathbb{V}[X] = \lambda^2 + \lambda - \lambda^2 = \lambda.$$

□

Poisson distribution are used to describe rare events, and we want to formalize this notion with respect to other distributions next. Consider $X_1, ..., X_n \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$. We know $S = \sum X_i = \text{Binomial}(n, p)$, with

$$\mathbb{P}(X=k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

Now let us re-imagine this situation, with the twist that p is not constant but rather a function of n, i.e. $p = p(n)$, such that np approaches a constant λ as $n \to \infty$. This means that $O(p) = 1/n$. In this case, the binomial distribution can be approximated by a Poisson distribution with λ.

Theorem 1.3. *(Law of Small Numbers)*
When $np \to \lambda$ as $n \to \infty$,

$$\binom{n}{k} p^k (1-p)^{n-k} \longrightarrow e^{-\lambda} \frac{\lambda^k}{k!}.$$

Proof. Let us dissect the binomial PMF:

$$\binom{n}{k} p^k (1-p)^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

$$= \frac{1}{k!} \cdot \frac{n!}{(n-k)!n^k} \cdot (np)^k \cdot \left(1 - \frac{np}{n}\right)^{n-k}.$$

As $n \to \infty$, we have

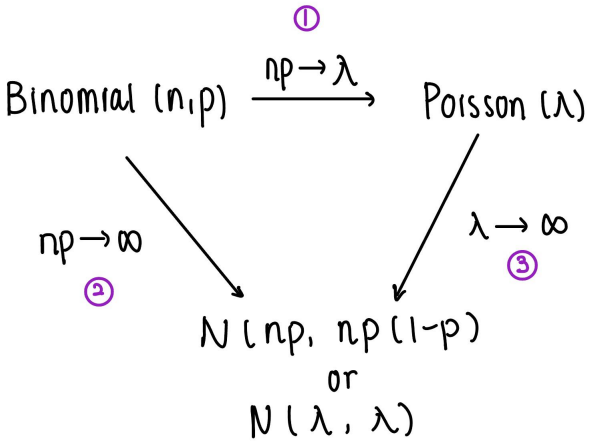
$$\frac{n!}{(n-k)!n^k} = \frac{n \cdot (n-1) \cdots (n-k+1)}{n^k} \to 1,$$
$$(np)^k \to \lambda^k,$$
$$\left(1 - \frac{np}{n}\right)^{n-k} \to \left(1 - \frac{\lambda}{n}\right)^n \to e^{-\lambda},$$

so combined we have

$$\binom{n}{k} p^k (1-p)^{n-k} \to \frac{1}{k!} \cdot \lambda^k \cdot e^{-\lambda} = e^{-\lambda} \frac{\lambda^k}{k!}.$$

□

Note that if $np \to \infty$ instead (for example, the normal case when p is constant), the binomial distribution can be approximated by normal distribution with $\mu = np$, $\sigma^2 = np(1-p)$ by central limit theorem (CLT). In fact, we have the following relation diagram:



where

- (1) follows from law of small numbers,
- (2) follows from CLT,
- (3) follows from additivity of Poisson distribution then CLT.

There are several things that makes Poisson distribution nice. First, it has only one parameter to worry about (versus two for binomial). The (3) above made use of another nice property: the sum of two independent Poisson distribution is still Poisson.

Theorem 1.4. *Suppose $X_1 \sim Poisson(\lambda_1)$ and $X_2 \sim Poisson(\lambda_2)$ are independent. Then*

$$X_1 + X_2 \sim Poisson(\lambda_1 + \lambda_2).$$

Proof. The first sanity check we can perform is that the expectation and variance both match (by linearity of expectation and variance addition for independence), but to show that the resulting distribution is still Poisson needs a bit more work. By law of total probability and independence,

$$\begin{aligned}
\mathbb{P}(X_1 + X_2 = k) &= \sum_{l=0}^{k} \mathbb{P}(X_1 = l, X_2 = k - l) \\
&= \sum_{l=0}^{k} \mathbb{P}(X_1 = l) \cdot \mathbb{P}(X_2 = k - l) \\
&= \sum_{l=0}^{k} e^{-\lambda_1} \frac{\lambda_1^l}{l!} \cdot e^{-\lambda_2} \frac{\lambda_2^{k-l}}{(k-l)!} \\
&= e^{-(\lambda_1+\lambda_2)} \frac{(\lambda_1+\lambda_2)^k}{k!} \sum_{l=0}^{k} \frac{k!}{l!(k-l)!} \cdot \frac{\lambda_1^l \lambda_2^{k-l}}{(\lambda_1+\lambda_2)^k} \\
&= e^{-(\lambda_1+\lambda_2)} \frac{(\lambda_1+\lambda_2)^k}{k!} \sum_{l=0}^{k} \binom{k}{l} \cdot \left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right)^l \cdot \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right)^{k-l} \\
&= e^{-(\lambda_1+\lambda_2)} \frac{(\lambda_1+\lambda_2)^k}{k!}.
\end{aligned}$$

□

Now we are ready to dive into statistical inference.

Theorem 1.5. *Let $X_1, ..., X_n \overset{iid}{\sim} Poisson(\lambda)$. Then the maximum likelihood estimator of λ is the sample mean, i.e.*

$$\widehat{\lambda}_{MLE} = \overline{X}.$$

Proof. First, the likelihood function is

$$L(\lambda) = \prod_{i=1}^{n} e^{-\lambda} \frac{\lambda^{X_i}}{(X_i)!} = e^{-n\lambda} \frac{\lambda^{\sum X_i}}{\prod (X_i)!}.$$

Taking the log yields

$$\log(L(\lambda)) = -n\lambda + \sum X_i \log(\lambda) - \log\left(\prod (X_i)!\right).$$

Setting the derivative to 0 gives

$$-n + \frac{\sum X_i}{\lambda} = 0,$$

or

$$\widehat{\lambda}_{\text{MLE}} = \frac{\sum X_i}{n} = \overline{X}.$$

□

1.2. **Basics of Confidence Interval.**

Compared to point estimators like the MLE estimator in the theorem above, confidence intervals (CIs) seek to put an error bound on the estimation. The goal is to construct instead two estimators $\widehat{\lambda}_L$ and $\widehat{\lambda}_R$ as functions of data. They are still point estimators but not for λ itself. We say that the resulting interval $(\widehat{\lambda}_L, \widehat{\lambda}_R)$ is a $(1-\alpha)$-level CI if

$$\mathbb{P}(\lambda \in (\widehat{\lambda}_L, \widehat{\lambda}_R)) = 1 - \alpha. \tag{1.6}$$

It is important to clarify the interpretation of this statement. At first glance, it looks like the above is saying that the fixed unknown true value of λ lies between the data-dependent random interval with probability of $1-\alpha$, given any single observation. This is **not** true. Think about it: how can you pin down a concrete probability when you do not even know the true value? Rather, the interpretation should be: if we repeat the trials many, many times and calculate the CI for each, we expect a coverage probability of $1-\alpha$, i.e. a proportion of $1-\alpha$ CIs should cover the true value of the parameter. Conversely, α is the non-coverage probability we are willing to pay.

If the above sounds confusing, consider the following scenario: we have a Poisson distribution with unknown parameter value. We ask each of the sixty students in STAT 245 to go home, collect data, and come back with their own CI the next day, say, for $\alpha = 0.95$. We cannot say any of these individual CIs has a 95% chance of covering the true value. What we can expect is that about 57 of these CIs will contain the true value.

The point is, with respect to CIs, probability should always be associated with repeated trials, and we cannot make statements about single observations.

Now let us cover three methods to construct these endpoint estimators.

1.3. **Wald's Method.**

Recall that in CLT we have

$$\sqrt{n} \cdot \frac{\widehat{\lambda} - \lambda}{\sqrt{\lambda}} \rightsquigarrow N(0,1),$$

where '$\rightsquigarrow$' means 'convergence in distribution'. If we use the traditional method to solve for λ (move stuff around inside the parenthesis for probability), things get messy quickly because there are too many λ's, and λ is unknown. So the idea here, attributed to Abraham Wald (credited for survivor's bias too), is to replace the λ in the denominator by $\widehat{\lambda}$, something we do know. This is valid because of the following theorem.

Theorem 1.7. *(Slutsky's Theorem)*
As long as $\widehat{\lambda}$ is consistent,

$$\sqrt{n} \cdot \frac{\widehat{\lambda} - \lambda}{\sqrt{\widehat{\lambda}}} \rightsquigarrow N(0,1).$$

Here we do have $\widehat{\lambda} \to \lambda$ by weak law of large numbers (WLLN). So let us continue the experiment. Let $z_{\alpha/2}$ and $z_{1-\alpha/2}$ be the two numbers such that the probability density beyond them sums up $\alpha/2$, respectively, in a normal distribution. Since $N(0,1)$ is symmetric about 0, we know that $z_{\alpha/2} = -z_{1-\alpha/2}$ By Slutsky's theorem,

$$\mathbb{P}\left(-z_{1-\alpha/2} \leq \sqrt{n} \cdot \frac{\widehat{\lambda} - \lambda}{\sqrt{\widehat{\lambda}}} \leq z_{1-\alpha/2}\right) = \mathbb{P}\left(\widehat{\lambda} - z_{1-\alpha/2} \cdot \sqrt{\frac{\widehat{\lambda}}{n}} \leq \lambda \leq \widehat{\lambda} + z_{1-\alpha/2} \cdot \sqrt{\frac{\widehat{\lambda}}{n}}\right) \to 1 - \alpha,$$

so the corresponding CI would be

$$\left[\widehat{\lambda} - z_{1-\alpha/2} \cdot \sqrt{\frac{\widehat{\lambda}}{n}}, \widehat{\lambda} + z_{1-\alpha/2} \cdot \sqrt{\frac{\widehat{\lambda}}{n}}\right].$$

Although the results are easy to calculate, the drawbacks of this method are also quite apparent. Mainly there exists two approximation error we now need to take into account: aside from the accuracies that may result when we are not using sufficiently large n's, the deviation of $\widehat{\lambda}$ from λ is another thing we need to worry about. Thus the accuracy of this method improves rather slowly as we increase n.

Another drawback is that it is definitely possible for the left point estimate to end up being negative, but $\lambda > 0$ for Poisson distributions. In those cases, we have to remember to adjust the left point estimate to 0 manually.

Before we move on, assuming we do not need to truncate the CI on the left, the length of the CI is

$$\widehat{\lambda}_R - \widehat{\lambda}_L = 2z_{1-\alpha/2}\sqrt{\frac{\widehat{\lambda}}{n}}.$$

We observe that

- as $n \to \infty$, the length decreases at rate $\sqrt{1/n}$ to 0;
- as α increases, the length also decreases.

Both align with intuition.

1.4. **Wilson's Method.**

Given the apparent drawbacks of Wald's method, we want to explore alternative methods. Wilson proposed the following: instead of substituting $\widehat{\lambda}$ for λ, we keep λ as it is and solve the complicated

inequality that would yield the corresponding endpoint estimator. The key is to rewrite

$$
\begin{aligned}
& -z_{1-\alpha/2} \leq \sqrt{n} \cdot \frac{\widehat{\lambda}-\lambda}{\sqrt{\lambda}} \leq z_{1-\alpha/2}, \\
\Longrightarrow & \left| \sqrt{n} \cdot \frac{\widehat{\lambda}-\lambda}{\sqrt{\lambda}} \right| \leq z_{1-\alpha/2}, \\
\Longrightarrow & \frac{n(\widehat{\lambda}-\lambda)^2}{\lambda} \leq z_{1-\alpha/2}^2. \\
\Longrightarrow & (\widehat{\lambda}-\lambda)^2 \leq z_{1-\alpha/2}^2 \cdot \frac{\lambda}{n}, \qquad (*)
\end{aligned}
$$

Plotting the LHS (black) and RHS (blue) of $(*)$ returns Figure 1.

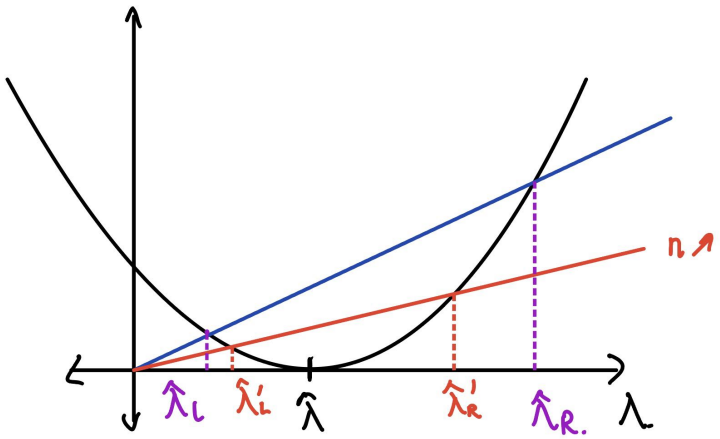


FIGURE 1. LHS & RHS of Inequality from Wilson's Method, Poisson Distribution

We realize that the value of λ for which the curve lies under the line satisfies the inequality; $\widehat{\lambda}_L$ and $\widehat{\lambda}_R$ are solutions to the quadratic equation and act as endpoint estimators for the CI!

We also note that as n increases, the RHS decreases (red line), and the resulting interval becomes smaller.

In general, Wilson's methods works better than Wald's method for small n, but they are asymptotically the same. Another benefit is that as we can see from the graph above, $\widehat{\lambda}_L$ is never negative. This is more reasonable than the symmetric estimators yielded by Wald's method (in the untruncated case), since Poisson distribution is skewed and its error distribution should be skewed as well.

1.5. **Variance Stabilizing Transform (VST).**

This method is attributed to Anscombe, the founding chair of the Yale statistics department. Its performance is comparable to Wilson's method. The main idea is that, given

$$\sqrt{n}(\widehat{\lambda} - \lambda) \rightsquigarrow N(0, \lambda)$$

by CLT, we can stabilize the parameter such that it does not depend on λ anymore, i.e. we can find a function $g(\cdot)$ such that

$$\sqrt{n}(g(\widehat{\lambda}) - g(\lambda)) \rightsquigarrow N(0, 1).$$

Let us provide some intuition (justification) for why such a method can work. First, CLT states that the numerator and denominator go to 0 at the same rate (since $N(0,1)$ is $O(1)$ with no dependence on n), so $(\widehat{\lambda} - \lambda) \to 0$ at $1/\sqrt{n}$. By Taylor's expansion, we have

$$g(\widehat{\lambda}) - g(\lambda) = g'(\lambda)(\widehat{\lambda} - \lambda) + O((\widehat{\lambda} - \lambda)^2) = g'(\lambda)(\widehat{\lambda} - \lambda) + O\left(\frac{1}{n}\right),$$

$$\implies \sqrt{n} \cdot \Big[g(\widehat{\lambda}) - g(\lambda)\Big] = \sqrt{n} \cdot g'(\lambda) \cdot (\widehat{\lambda} - \lambda) + O\left(\frac{1}{\sqrt{n}}\right).$$

Given

$$\sqrt{n} \cdot (\widehat{\lambda} - \lambda) \rightsquigarrow N(0, \lambda),$$

we also have

$$\sqrt{n} \cdot g'(\lambda) \cdot (\widehat{\lambda} - \lambda) \rightsquigarrow N(0, \lambda |g'(\lambda)|^2).$$

Thus

$$\sqrt{n} \cdot \Big[g(\widehat{\lambda}) - g(\lambda)\Big] \rightsquigarrow N(0, \lambda |g'(\lambda)|^2) + 0 = N(0, \lambda |g'(\lambda)|^2).$$

This argument is a specific application of delta method. We can also rephrase what we just said above in English. Picture the normal distribution with respect to λ. Given

$$(\widehat{\lambda} - \lambda) \to N\left(0, \frac{\lambda}{n}\right),$$

the range of λ that we care about becomes smaller and smaller as $n \to \infty$. On the other hand, take any smooth function (not necessarily linear) $g(\lambda)$. By the idea that lies behind the concept of derivative, given a sufficiently small range of λ, we can treat g as (approximately) linear in that range. Linear transformation of normal distributions are normal, and we end up with $g(\widehat{\lambda}) - g(\lambda)$ also normal.

Now the question becomes what kind of $g(\cdot)$ we can pick. The goal is to get

$$\lambda |g'(\lambda)|^2 = 1 \implies g'(\lambda) = \sqrt{\frac{1}{\lambda}} \implies g(\lambda) = \int \sqrt{\frac{1}{\lambda}}\, d\lambda = 2\sqrt{\lambda}.$$

Applying the result gives

$$2\sqrt{n}\left(\sqrt{\widehat{\lambda}} - \sqrt{\lambda}\right) \to N(0, 1),$$

which means

$$\mathbb{P}\left(-z_{1-\alpha/2} \le 2\sqrt{n}\left(\sqrt{\widehat{\lambda}} - \sqrt{\lambda}\right) \le z_{1-\alpha/2}\right) \to 1 - \alpha,$$

and the corresponding CI is

$$\left[\sqrt{\widehat{\lambda}} - \frac{z_{1-\alpha/2}}{2\sqrt{n}}, \sqrt{\widehat{\lambda}} + \frac{z_{1-\alpha/2}}{2\sqrt{n}}\right]$$

for $\sqrt{\lambda}$. We can then determine if we need to truncate the interval if the left side is negative, and then take the square for λ.

Here is another question: what would happen if we apply a function that acts independently of $(\widehat{\lambda} - \lambda)$ without transforming each, i.e. what if we do

$$\sqrt{n} \cdot f(\lambda) \cdot (\widehat{\lambda} - \lambda) \to N(0, \lambda[f(\lambda)]^2) = N(0, 1)?$$

This would give $f(\lambda) = 1/\sqrt{\lambda}$, and we get

$$\frac{\sqrt{n} \cdot (\widehat{\lambda} - \lambda)}{\sqrt{\lambda}} \Longrightarrow N(0, 1),$$

which is exactly Wilson's method!

Now let us apply what we just learned to a new distribution: the exponential distribution, which we will explore in relation to the Poisson distribution in Section 1.8.

1.6. **Exponential Distribution.**

Let $X_1, ..., X_n \overset{\text{iid}}{\sim} \text{Exp}(\lambda)$, i.e.

$$p(x|\lambda) = \lambda e^{\lambda x}, x > 0.$$

Theorem 1.8. *The expectation and variance of an exponential distribution with rate parameter λ is*

$$\mathbb{E}[X] = \frac{1}{\lambda},$$
$$\mathbb{V}[X] = \frac{1}{\lambda^2},$$

respectively.

Proofs are by pure calculus and are left as exercises to the readers.

Theorem 1.9. *Let $X_1, ..., X_n \overset{iid}{\sim} Exp(\lambda)$. Then the maximum likelihood estimator of λ is the inverse of the sample mean, i.e.*

$$\widehat{\lambda}_{MLE} = \frac{1}{\overline{X}}.$$

Proof. First, the likelihood function is

$$L(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda X_i} = \lambda^n \cdot e^{-\lambda \sum X_i}.$$

Taking the log yields

$$\log(L(\lambda)) = n \log(\lambda) - \lambda \sum X_i.$$

Setting the derivative to 0 gives

$$\frac{n}{\lambda} - \sum X_i = 0,$$

or

$$\widehat{\lambda}_{\text{MLE}} = \frac{n}{\sum X_i} = \frac{1}{\overline{X}}.$$

□

Now let us obtain CIs for the MLE estimator. We have a slight dilemma though: the MLE estimator is the inverse of the sample mean and not the sample mean itself, which means applying CLT would not directly yield any results. Nevertheless, let us apply the CLT to the sample mean and see what we get. First, we have

$$\mathbb{E}[\overline{X}] = \frac{1}{\lambda},\ \mathbb{V}[\overline{X}] = \frac{\mathbb{V}[X]}{n} = \frac{1}{n\lambda^2}.$$

So by CLT,

$$\frac{\overline{X} - \frac{1}{\lambda}}{\sqrt{\frac{1}{n\lambda^2}}} = \sqrt{n}\lambda\left(\overline{X} - \frac{1}{\lambda}\right) \rightsquigarrow N(0,1),$$

or

$$\sqrt{n}\left(\overline{X} - \frac{1}{\lambda}\right) \rightsquigarrow N\left(0, \frac{1}{\lambda^2}\right). \tag{1.10}$$

Let us apply the delta method! Given some smooth g, we have

$$\sqrt{n}\left(g(\overline{X}) - g\left(\frac{1}{\lambda}\right)\right) \rightsquigarrow N\left(0, \frac{1}{\lambda^2}|g'(\lambda)|^2\right).$$

Consider $g(t) = 1/t$. Then

$$g'(t) = -\frac{1}{t^2},\ |g'(t)|^2 = \frac{1}{t^4}.$$

Replacing t by $1/\lambda$, we get that the variance on the RHS of 1.10 should be

$$\frac{1}{\lambda^2} \cdot \lambda^4 = \lambda^2,$$

and we have

$$\sqrt{n}(\widehat{\lambda} - \lambda) \rightsquigarrow N(0, \lambda^2), \tag{1.11}$$

the desired asymptotic normal distribution.

There exists an alternative method to get to this answer. We need some new terminology to introduce this second method.

Definition 1.12. Let X be a RV whose distribution depends on θ, and let $p(X|\theta)$ be the corresponding likelihood function. Then the *score function* is

$$S_\theta(X) = \frac{\partial}{\partial\theta}\log[p(X|\theta)].$$

Definition 1.13. Let the set-up be the same as in Definition 1.12. Then the *Fisher information* for parameter θ is

$$I_\theta = \mathbb{E}[S_\theta^2(X)].$$

Theorem 1.14. *The Fisher information admits three alternative expressions:*

$$I_\theta = \mathbb{E}[S_\theta^2(X)] = \mathbb{V}[S_\theta(X)] = -\mathbb{E}\left[\frac{\partial}{\partial\theta} S_\theta^2(X)\right].$$

For proof, please see the 244 cheatbook.

Let us calculate the Fisher information for λ in the exponential distribution. We have

$$\begin{aligned} &\log[p(X|\lambda)] = \log(\lambda) - \lambda X, \\ \implies &S_\lambda(X) = \frac{1}{\lambda} - X, \\ \implies &I_\lambda = \mathbb{E}\left[\left(\frac{1}{\lambda} - X\right)^2\right] = \mathbb{V}[X] = \frac{1}{\lambda^2}. \end{aligned}$$

Now we are ready to introduce Fisher's theorem.

Theorem 1.15. *(Fisher's Theorem)*
Let $X_1, ..., X_n \overset{iid}{\sim} p(X|\theta)$*, and let*

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} p(X_i|\theta)$$

be the MLE estimator. Then as long as $p(X|\theta)$ *is smooth,*

$$\sqrt{n}(\widehat{\theta} - \theta) \rightsquigarrow N\left(0, \frac{1}{I_\theta}\right).$$

We can check that Fisher's theorem immediately gives us the convergence in 1.11 by the above.

1.7. **Confidence Intervals for Exponential Distribution.**

1.7.1. *Wald's Method.*

Let us rewrite (1.11) as

$$\frac{\sqrt{n}}{\lambda}(\widehat{\lambda} - \lambda) \rightsquigarrow N(0, 1), \tag{1.16}$$

then replace λ in the denominator by $\widehat{\lambda}$:

$$\frac{\sqrt{n}}{\widehat{\lambda}}(\widehat{\lambda} - \lambda) \rightsquigarrow N(0, 1),$$

which is still true by Slutsky's theorem. This means

$$\mathbb{P}\left(\left|\frac{\sqrt{n}}{\widehat{\lambda}}(\widehat{\lambda} - \lambda)\right| \leq z_{1-\frac{\alpha}{2}}\right) \longrightarrow 1 - \alpha,$$

and the corresponding CI is

$$\left[\widehat{\lambda} - z_{1-\frac{\alpha}{2}} \cdot \frac{\widehat{\lambda}}{\sqrt{n}}, \widehat{\lambda} + z_{1-\frac{\alpha}{2}} \cdot \frac{\widehat{\lambda}}{\sqrt{n}}\right].$$

1.7.2. *Wilson's Method.*

By (1.16), we have

$$\mathbb{P}\left(\left|\frac{\sqrt{n}}{\lambda}(\widehat{\lambda}-\lambda)\right| \leq z_{1-\frac{\alpha}{2}}\right) \longrightarrow 1-\alpha,$$

where the content inside the bracket is equivalent to

$$|\widehat{\lambda}-\lambda| \leq \frac{z_{1-\frac{\alpha}{2}} \cdot \lambda}{\sqrt{n}}.$$

Graphing out the LHS (black) and RHS (blue) gives Figure 2.

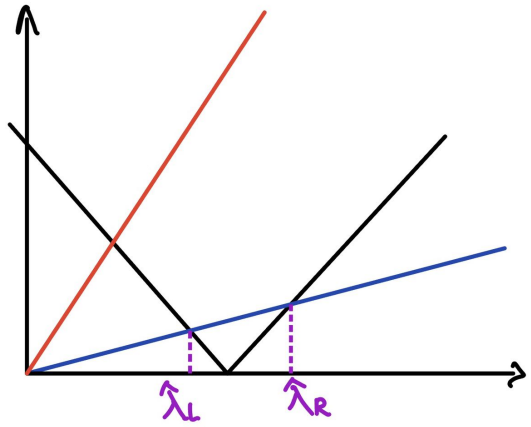


FIGURE 2. LHS & RHS of Inequality from Wilson's Method, Exponential Distribution

Taking away the absolute value and solving the two inequalities gives the CI

$$\left[\frac{\widehat{\lambda}}{1+\frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}}, \frac{\widehat{\lambda}}{1-\frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}}\right].$$

However, we do have a pitfall with Wilson's method here: given that the graph generated by the LHS has slope ± 1, the line generated by the RHS needs to have a slope < 1 in order to have two intersection; otherwise we would only have one intersection, which does not yield a valid CI (see the combination of black and red lines in Figure 2). Thus, for Wilson's method to work in this case, we need to assume

$$\frac{z_{1-\frac{\alpha}{2}}}{n} < 1$$

or

$$n > z^2_{1-\frac{\alpha}{2}}.$$

This turns out to be mostly a trivial assumption, since, for example, if $\alpha = 0.05$, we only need $n \geq 4$. If $n \leq 4$, we probably should not be using this approximation in the first place (recall that CLT only works for large samples, where 'large' is generally taken to be > 30).

1.7.3. *VST.*

Given (1.11), we want a transformation g such that

$$\sqrt{n}\big(g(\widehat{\lambda}) - g(\lambda)\big) \rightsquigarrow N(0, \lambda^2 |g'(\lambda)|^2) = N(0,1).$$

Then one of the solutions would be

$$|g'(\lambda)|^2 = \frac{1}{\lambda^2} \Longrightarrow g'(\lambda) = \frac{1}{\lambda} \Longrightarrow g(\lambda) = \log(\lambda).$$

Applying the result gives

$$\sqrt{n}\big(\log(\widehat{\lambda}) - \log(\lambda)\big) \rightsquigarrow N(0,1),$$

and

$$\mathbb{P}\left(\left|\sqrt{n} \cdot \log\left(\frac{\widehat{\lambda}}{\lambda}\right)\right| \leq z_{1-\frac{\alpha}{2}}\right) \longrightarrow 1 - \alpha.$$

Taking away the absolute value and solving the two inequalities gives the CI

$$\left[\widehat{\lambda} e^{-\frac{z_{1-\frac{\alpha}{2}}}{n}}, \widehat{\lambda} e^{\frac{z_{1-\frac{\alpha}{2}}}{n}}\right].$$

Disregarding the caveat in Wilson's method, here we still have Wilson and VST outperforming Wald. In fact, taking the first-order Taylor expansion of either Wilson or VST would recover Wald, so they are first-order equivalent.

We conclude with a discussion on optimality. A procedure is *inadmissible* if there exists another procedure that works better for any parameter and for any n. (Fun fact: if a procedure is Bayesian, then it is always admissible.) None of the three methods we have covered is inadmissible. There are cases, however few, where Wald outperforms the other two.

1.8. **Poisson Process.**

Let us start with the following scenario: suppose we are recording volcano eruptions in a certain period of time and attempting to model them according to some distribution. The scenario seems to satisfy the situation for Poisson, so let us, say, record the total number of volcano eruptions in the first century, the number in the second century, etc., and model this using a Poisson distribution with appropriate λ.

There are, however, obvious caveats here: first, the length of period to use is rather subjective. Why are we measuring things only every century and not every decade, or even every twenty-seven years if I feel like it? There is simply no good rule of thumb. More importantly though, by recording the eruptions by sums, we are intentionally deleting a lot of data. When we record the volcano eruptions, we are not only recording the number but also the *time* at which these eruptions took place. Information is precious, and we should really be using all that we have to generate a model

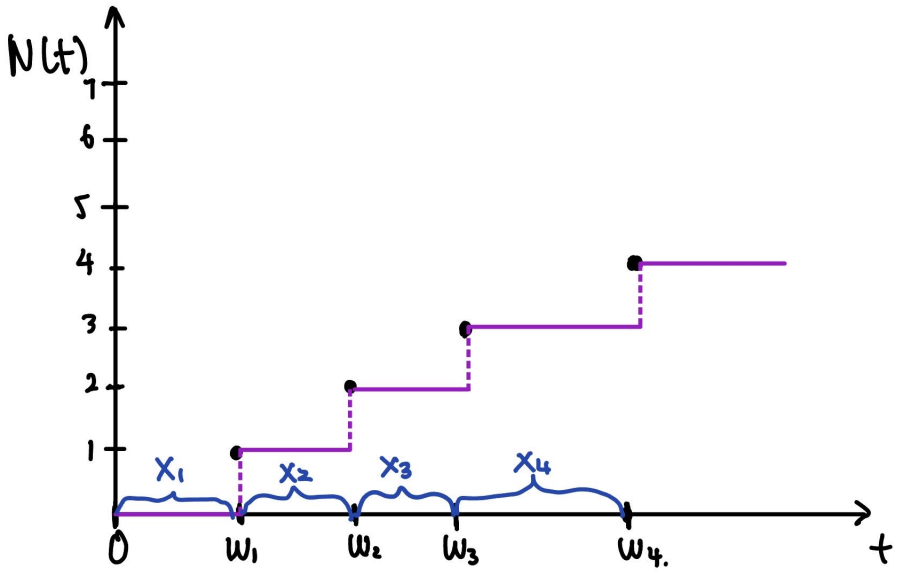
as accurate as possible. This motivates a different approach to model this situation, for which we need some new terminologies.

Definition 1.17. A *random vector* $[X_1, ..., X_k]'$ is a k-dimensional vector of k RVs, for which there exists a joint distribution between them.

Definition 1.18. A *random function* (aka. *stochastic process*) $X(t)$ is a function that returns a RV for each t, such that for all $t_1, t_2, ..., t_k$, it defines a random vector $[X(t_1), X(t_2), ..., X(t_k)]$.

This is a valid definition because if a function defines the joint distribution for any finite-dimensional realization, then there exists a joint distribution underlying the entire function. This existence is guaranteed by Kolmogorov's consistency theorem (interested readers can find more information on the Wikipedia page or the lecture notes here, the latter of which is slightly easier to parse). The comments in this StackExchange post provides a good example of stochastic process.

Let us return to our volcano eruption recording scenario. Suppose, after measurement, we end up with the following graph:



where we have

- time t on the x-axis,
- $N(t)$, the number of events happened before (and including) t, on the y-axis,
- W_k, *waiting time* for the kth event occurred,
- X_k, the time passed between the $k-1$th and the kth event.

This model motivates the following definition.

Definition 1.19. A random function $(N(t) : t \geq 0)$ follows a *Poisson process* with rate parameter λ (denoted $(N(t) : t \geq 0) \sim \text{PP}(\lambda)$) if it satisfies the following three criteria:

(1) $N(0) = 0$;
(2) for any s and t, $N(t+s) - N(s) \perp\!\!\!\perp N(s)$, i.e. the number of events taking place in a given interval must be independent of the past (conversely, past realizations do not change future patterns);
(3) for any s and t, $N(t+s) - N(s) \sim \text{Poisson}(\lambda t)$, i.e. the number of events taking place in a given interval must follow a Poisson distribution with a rate parameter proportional to the length of the interval.

In other words, if there exists a unique stochastic process that satisfies these three conditions, then it is a Poisson process. This is a very elegant way to model event occurrences over a long period of time.

We have a few remarks: first, it immediately follows that

$$N(t) = N(t) - N(0)$$

must also be Poisson with rate parameter λt. Secondly, observe that

$$W_1 = X_1,\ W_2 = X_1 + X_2,\ W_3 = X_1 + X_2 + X_3,\ W_4 = X_1 + X_2 + X_3 + X_4, ...$$

so if we can understand the distribution of waiting times, we can also analyze the distribution of X_i's. Our goal is to show the following.

Theorem 1.20. *Let $X_1, X_2, ..., X_n$ be defined as above. Then $X_1, X_2, ..., X_n \overset{iid}{\sim} Exp(\lambda)$.*

In particular, given $\mathbb{E}[X] = 1/\lambda$, we have this inverse dependence on the parameter. This should make some intuitive sense since the larger the λ, the more frequently the events occur and the shorter the waiting time.

Proof. We start with X_1:

$$\mathbb{P}(X_1 \leq t) = 1 - \mathbb{P}(X_1 \geq t) = 1 - \mathbb{P}(N(t) = 0),$$

where the second equality follows from the fact that $X_1 > t$ implies zero events occurred as of time t, and vice versa. However, we know $N(t) \sim \text{Poisson}(\lambda t)$, so

$$\mathbb{P}(X_1 \leq t) = 1 - e^{-\lambda t} \cdot \frac{(\lambda t)^0}{0!} = 1 - e^{-\lambda t},$$

which is exactly the CDF of an exponential distribution with parameter λ! (If you do not have that memorized, we can just take the derivative to get the more recognizable PDF $e^{-\lambda t}$.) Hence $X_1 \sim \text{Exp}(\lambda)$.

Now we proceed to X_2. We cannot directly calculate X_2, for we have no guarantee that the X_i's are independent, so let us instead examine the joint distribution of X_1 and X_2 and then use the marginal of X_1 if needed. Thus, for some arbitrary s,

$$\mathbb{P}(X_2 \leq t \mid X_1 = s) = 1 - \mathbb{P}(X_2 > t \mid X_1 = s)$$

$$\begin{aligned}
&= 1 - \mathbb{P}\big(N(t+s) - N(s) = 0 \mid X_1 = s\big) \\
&= 1 - \mathbb{P}\big(N(t+s) - N(s) = 0\big) \\
&= 1 - e^{-\lambda t},
\end{aligned}$$

where the second equality follows from similar reasoning to above and the third by independence of history. The result is again the exponential CDF. However, since we are able to eliminate the dependence of X_2 on s and thus on X_1, this is exactly the distribution of X_2, and we have again $X_2 \sim \text{Exp}(\lambda)$.

Finally, by mathematical induction, we conclude that $X_1, X_2, ..., X_n \overset{\text{iid}}{\sim} \text{Exp}(\lambda)$. □

As a bonus, per the fact that $\text{Exp}(\lambda) = \text{Gamma}(1, \lambda)$ and the Gamma addition rule

$$\sum_i \text{Gamma}(\alpha_i, \lambda) = \text{Gamma}\left(\sum_i \alpha_i, \lambda\right),$$

we conclude

$$W_k = \sum_{i=1}^{k} X_i = \text{Gamma}(k, \lambda),$$

i.e. the RV for waiting times follows a Gamma distribution. (Note that we wrote '$1/\lambda$' instead of 'λ' as the second parameter in class.)

Let us pause for a moment and reflect. The Poisson process model we established here uses a constant λ throughout time, which seems reasonable for our volcano eruption measurement purposes. However, it may no longer be sufficient in scenarios where λ varies over time. For example, if we model the distribution of Bus 6 arrival times, every single UChicago student who have struggled to go Downtown from Hyde Park with that bus would tell you that is a horrible model. This motivates a more flexible model than the one in Definition 1.19.

Definition 1.21. An *inhomogeneous Poisson process* (denoted $(N(t) : t \geq 0) \sim \text{PP}(\lambda(\cdot))$, where $\lambda(\cdot)$ is the *rate function*) is a random function that satisfies the following three criteria:

(1) $N(0) = 0$;
(2) for any s and t, $N(t+s) - N(s) \perp\!\!\!\perp N(s)$;
(3) for any s and t,

$$N(t+s) - N(s) \sim \text{Poisson}\left(\int_s^{t+s} \lambda(x)\, dx\right).$$

This model allows us to do non-parametric estimations. Note the specific case where $\lambda(\cdot)$ is constant, i.e. $\lambda(x) = \lambda$ for all $x \geq 0$. Then

$$\int_t^{t+s} \lambda(x)\, dx = \lambda s,$$

and we recover the *homogeneous* case above.

We can even play around with higher generalizations. We have been working with a 1-dimensional space so far ($t \in [0, \infty) \subset \mathbb{R}$). With a little more abstraction, there is really nothing to stop us from applying the model to 2- or even $\geq$ 3-dimensional spaces (for example, the position of stars in the universe).

Definition 1.22. An *Poisson point process* (denoted ($N(B) : B$ is any measurable set) $\sim \text{PPP}(\lambda(\cdot))$, where $\lambda(\cdot)$ is the *rate measure*) is a random function that satisfies the following three criteria:

(1) $N(\emptyset) = 0$;
(2) for any pair of disjoint measurable sets A and B (i.e. $A \cap B = \emptyset$), $N(A) \perp\!\!\!\perp N(B)$;
(3) for any measurable set B,

$$N(B) \sim \text{Poisson}(\lambda(B)).$$

(The concept of measure and measurable set is not too important here. In fact, this cheatbook is written by someone who had taken an entire course in measure theory, and the only non-measurable set I have come across is the Vitali set. For all purposes and intent, if you are not studying a field that uses measure theory, all sets you will ever interact with are measurable with probability 1. So relax. If you are interested, however, feel free to consult 271 cheatbook.)

Note that the model we have in Definition 1.19 is just a special case of the model here where sets are intervals. Also remark the difference between (3) here and (3) in the previous two definitions: the interval $t = t + s - s$ is equivalent to B, not $B \backslash A$. Given $N(A) \sim \text{Poisson}(\lambda(A))$, $N(B) \sim \text{Poisson}(\lambda(B))$, $N(B) - N(A)$ is not Poisson. Even though the sum of two Poisson distributions is Poisson (as we have shown in Theorem 1.4), their difference is not. (In fact, the difference follows a Skellam distribution.)

1.9. **Normal Distribution.**

Let $X_1, ..., X_n \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$. Let us recall some basic facts:

$$(1)\, \widehat{\mu}_{\text{MLE}} = \overline{X},$$
$$(2)\, \mathbb{E}[\overline{X}] = \mu,$$
$$(3)\, \mathbb{V}[\overline{X}] = \frac{\mathbb{V}[X]}{n} = \frac{\sigma^2}{n},$$

which means

$$\overline{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right),$$

or

$$\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sigma} \sim N(0, 1). \tag{1.23}$$

Note that the result here is exact (we do not have convergence in distribution).

The goal is to find CIs for μ. We have two cases.

Case 1. σ^2 is known.
This case is straightforward. Per (1.23),

$$\mathbb{P}\left(\left|\frac{\sqrt{n}\cdot(\overline{X}-\mu)}{\sigma}\right| \leq z_{1-\frac{\alpha}{2}}\right) = 1-\alpha.$$

This gives the CI

$$\left[\overline{X} - z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}},\ \overline{X} + z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right].$$

Case 2. σ^2 is unknown.
Let us first reflect which of the three methods above we can use. Since MLE estimators are consistent, Slutsky's theorem holds and we have at least one choice for $\widehat{\sigma}^2$ to ensure Wald's method works. Note that, however, neither Wilson's method nor VST is applicable here. The key is that the normal distribution has two parameters, compared to the one-parameter Poisson and exponential distributions. If we perform the procedure for Wilson, we would end up with one equation with two unknowns, which is unsolvable. Similarly, it would not be possible to determine a $g(\cdot)$ that only takes one parameter for stabilization. Thus only Wald remains.

To use Wald's method, we need to determine an estimator for σ^2, among which we have the following choices:

$$\widehat{\sigma}_1^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2,$$
$$\widehat{\sigma}_2^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X})^2,$$
$$\widehat{\sigma}_3^2 = \frac{1}{n+1}\sum_{i=1}^{n}(X_i-\overline{X})^2,$$

where $\widehat{\sigma}_1^2$ is unbiased, $\widehat{\sigma}_2^2$ is the MLE estimator, and $\widehat{\sigma}_3^2$ has minimal MSE (the verification of each we left as exercises to the reader). All estimators here are consistent. Small MSE is good in statistics, so let us go along with $\widehat{\sigma}^2 = \widehat{\sigma}_3^2$. Wald's method gives

$$\mathbb{P}\left(\left|\frac{\sqrt{n}(\overline{X}-\mu)}{\widehat{\sigma}^2}\right| \leq z_{1-\frac{\alpha}{2}}\right) \to 1-\alpha.$$

The corresponding CI is

$$\left[\overline{X} - z_{1-\frac{\alpha}{2}}\frac{\widehat{\sigma}}{\sqrt{n}},\ \overline{X} + z_{1-\frac{\alpha}{2}}\frac{\widehat{\sigma}}{\sqrt{n}}\right].$$

So this approximation gives us something to work with. It turns out even when σ^2 is unknown, we can have something exact. This requires a set of new terminologies and concepts.

1.10. **Multivariate Normal Distribution.**

Although multivariate normal distribution is interesting in itself, here we will only briefly cover its key properties as tools for Section 1.11. Also note that it is more often referred as 'multivariate Gaussian distribution' in class.

Definition 1.24. A random vector $X = (X_1, ..., X_p)^T \sim N(\mu, \Sigma)$ in $\mathbb{R}^p$ follows a *multivariate normal distribution* (MVN) if its density is

$$p(x) \propto \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right),$$

such that

$$\mu = \mathbb{E}[X] = (\mathbb{E}[X_1], ..., \mathbb{E}[X_p])^T \in \mathbb{R}^p$$

and the *covariance matrix*

$$\Sigma = \begin{pmatrix} \mathbb{V}[X_1] & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_p) \\ \mathrm{Cov}(X_2, X_1) & \mathbb{V}[X_2] & \cdots & \mathrm{Cov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_p, X_1) & \mathrm{Cov}(X_p, X_2) & \cdots & \mathbb{V}[X_p] \end{pmatrix} \in \mathbb{R}^{p\times p}$$

is positive semidefinite.

The normalizing constant is technically $1/\sqrt{(2\pi)^p|\Sigma|}$, though we can always figure that out later in actual computation.

Fun aside before we get into serious business: even though normal distributions also go by 'Gaussian distribution,' the major work was not done by Gauss. Instead, the initial discovery was by de Moivre, and then Laplace conducted rigorous investigation by first calculating the normalizing constant

$$\int e^{-\frac{x^2}{2}}\, dx = \sqrt{2\pi},$$

and secondly, verifying CLT in the case of Bernoulli distribution. Apparently, for a list of very mis-credited deeds, see Stigler's law of eponymy.

There are three important properties to keep in mind.

Theorem 1.25. *(Properties of MVN)*
Let $X \sim N(\mu, \Sigma)$.

1. *Linear combinations of X is still normal.*
2. *Given $A \in \mathbb{R}^{q\times p}, b \in \mathbb{R}^{q\times 1}$,*
$$AX + b \sim N\left(A\mu + b, A\Sigma A^T\right).$$
3. *Let $X \in \mathbb{R}^p$, $Y \in \mathbb{R}^q$ such that $(X\, Y)^T$ is jointly normal (here $(X\, Y)^T$ just means X stacked on top of Y). If $Cov(X, Y) = 0$, then $X \perp\!\!\!\perp Y$.*

A few remarks before the proof. Property (1) follows directly from the density function of normal distribution and linear transformation of RVs, and the proof is left to readers as an exercise. Also note that the converse of (3) is true for any distribution (i.e. if X and Y are independent, then their covariance is 0). The significance of this statement lies exactly in the fact that we can reverse the independence-covariance relation as long as the distribution is normal.

Proof. We start with (2). By (1), $AX + b$ is also normal, so we only need to compute the mean and the covariance. By linearity of expectation,

$$\mathbb{E}[AX + b] = A\mathbb{E}[X] + b = A\mu + b.$$

The convariance part is slightly tricker:

$$\begin{aligned}\mathrm{Cov}(AX + b) &= \mathbb{E}\left[(AX + b - A\mu - b)(AX + b - A\mu - b)^T\right] \\ &= \mathbb{E}[A(x - \mu)(x - \mu)^T A^T] \\ &= A\mathbb{E}[(x - \mu)(x - \mu)^T]A^T = A\Sigma A^T.\end{aligned}$$

For (3), the goal is to show that the joint density can be factorized, i.e. $p(x, y) \propto p(x)p(y)$. We start with

$$\begin{pmatrix} X \\ Y \end{pmatrix} = N\left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}\right).$$

Two things: first, $\Sigma_{xy} = \Sigma_{yx}$ by symmetry, and since $\mathrm{Cov}(X, Y) = 0$ by given, $\Sigma_{xy} = \Sigma_{yx} = 0$, and we can rewrite the distribution as

$$\begin{pmatrix} X \\ Y \end{pmatrix} = N\left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{pmatrix}\right).$$

Now let us expand the density:

$$\begin{aligned}p(x, y) &\propto \exp\left(-\frac{1}{2}\big((x - \mu_x)^T(y - \mu_y)^T\big)\begin{pmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{pmatrix}^{-1}\big((x - \mu_x)(y - \mu_y)\big)\right) \\ &= \exp\left(-\frac{1}{2}\big((x - \mu_x)^T(y - \mu_y)^T\big)\begin{pmatrix} \Sigma_{xx}^{-1} & 0 \\ 0 & \Sigma_{yy}^{-1} \end{pmatrix}\big((x - \mu_x)(y - \mu_y)\big)\right) \\ &= \exp\left(-\frac{1}{2}\left[\big((x - \mu_x)^T\Sigma_{xx}^{-1}(x - \mu_x)\big) + \big((y - \mu_y)^T\Sigma_{yy}^{-1}(y - \mu_y)\big)\right]\right) \\ &= \exp\left(-\frac{1}{2}\big((x - \mu_x)^T\Sigma_{xx}^{-1}(x - \mu_x)\big)\right) \cdot \exp\left(-\frac{1}{2}\big((y - \mu_y)^T\Sigma_{yy}^{-1}(y - \mu_y)\big)\right) \\ &\propto p(x)p(y).\end{aligned}$$

□

1.11. t & Chi-Squared Distributions.

Definition 1.26. Let $X, Z_1, ..., Z_n \overset{\text{iid}}{\sim} N(0,1)$. Then

(1)
$$Y = \sum_{i=1}^{n} Z_i^2 \sim \chi_n^2,$$
where χ_n^2 is a *chi-squared distribution* with n dof.

(2) for $n \geq 2$,
$$\frac{X}{\sqrt{Y/n}} \sim t_n$$
is a *t-distribution* (aka. *Student's distribution*) with n dof.

Remarks 1.27.

(1) $\chi_n^2 \sim \text{Gamma}(n/2, 2)$.
(2) As $n \to \infty$, $Y/n \to \mathbb{E}[Z_i^2] = 1$ by WLLN, so $t_n \rightsquigarrow N(0,1)$, and thus we usually only use t_n for small n.
(3) The discovery of t-distribution is credited to William Sealy Gosset. Given (1-α)-confidence level, we can determine the values $t_{n-1,\,\alpha/2}$ and $t_{n-1,\,1-\alpha/2}$ such that the probability density beyond the thresholds sum to $\alpha/2$ respectively for each n (note the similarity of this definition with $z_{\alpha/2}$ and $z_{1-\alpha/2}$ in Section 1.3). It turns out t-distribution is also symmetric, so $t_{n-1,\,\alpha/2} = -t_{n-1,\,1-\alpha/2}$, and
$$\mathbb{P}\left(\left|\sqrt{n}\cdot\frac{\overline{X}-\mu}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2}}\right| \leq t_{n-1,\,1-\frac{\alpha}{2}}\right) = 1-\alpha.$$
The corresponding CI is
$$\left[\overline{X} - t_{n-1,\,1-\frac{\alpha}{2}}\cdot\sqrt{\frac{\sum_i(X_i-\overline{X})^2}{n(n-1)}},\ \overline{X} + t_{n-1,\,1-\frac{\alpha}{2}}\cdot\sqrt{\frac{\sum_i(X_i-\overline{X})^2}{n(n-1)}}\right].$$
This result is exact for small n's, which is immensely useful.
(4) For $n = 1$, $X/|Z_1|$ follows a *Cauchy distribution*. For more on Cauchy distribution, see the next section.

Lemma 1.28. *(Mean Pythagorean Theorem)*
Let $\{y_i\}_{i=1}^n$ be any sequence of numbers. Then
$$\sum_{i=1}^{n}(y_n - a)^2 = \sum_{i=1}^{n}(y_i - \overline{y})^2 + n(\overline{y} - a)^2.$$

Proof. The proof follows from the ancient wisdom of add and subtract:

$$\begin{aligned}
\sum_{i=1}^{n}(y_i-a)^2 &= \sum_{i=1}^{n}\big((y_i-\overline{y})+(\overline{y}-a)\big)^2 \\
&= \sum_{i=1}^{n}(y_i-\overline{y})^2+\sum_{i=1}^{n}(\overline{y}-a)^2+2\sum_{i=1}^{n}(y_i-\overline{y})(\overline{y}-a) \\
&= \sum_{i=1}^{n}(y_i-\overline{y})^2+n(\overline{y}-a)^2+2(\overline{y}-a)\sum_{i=1}^{n}(y_i-\overline{y}) \\
&= \sum_{i=1}^{n}(y_i-\overline{y})^2+n(\overline{y}-a)^2+2(\overline{y}-a)\left(\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}y_i\right) \\
&= \sum_{i=1}^{n}(y_i-\overline{y})^2+n(\overline{y}-a)^2.
\end{aligned}$$

□

Next comes about the most important theorem in this section.

Theorem 1.29. *Let* $X_1,...,X_n \overset{iid}{\sim} N(\mu,\sigma^2)$. *Then*

(1)
$$\sqrt{n}\cdot\frac{\overline{X}-\mu}{\sigma}\sim N(0,1).$$

(2)
$$\frac{\sum_{i=1}^{n}(X_i-\overline{X})^2}{\sigma^2}\sim\chi^2_{n-1}.$$

(3)
$$\overline{X}\perp\!\!\!\perp\sum_{i=1}^{n}(X_i-\overline{X})^2.$$

(4)
$$\sqrt{n}\cdot\frac{\overline{X}-\mu}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2}}\sim t_{n-1}.$$

Several remarks: first, note that all the relations are exact. Secondly, the reduction in the dof in statements (2) and (4) ($n-1$ instead of n) is due to the fact that X_i and $\overline{X}$ are not independent and thus loses information. This remark will become clearer in the proof. Lastly, statement (3) immediately implies that under the iid assumption, sample mean and sample variance are independent despite being drawn from the same distribution.

We will show things in order. Given that the proofs of (2) and (3) are quite long, we will separate the proofs into four different blocks.

Proof. **Statement 1.**
First, by Theorem 1.25 Property (1), $\sqrt{n} \cdot (\overline{X} - \mu)/\sigma$ is normal, and we only need to check the expectation and variance.

$$\mathbb{E}\left[\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sigma}\right] = \frac{\sqrt{n}}{\sigma} \cdot (0 - 0) = 0,$$
$$\mathbb{V}\left[\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sigma}\right] = \frac{n}{\sigma^2}\mathbb{V}[\overline{X} - \mu] = \frac{n}{\sigma^2}\mathbb{V}[\overline{X}] = \frac{n}{\sigma^2} \cdot \frac{\sigma^2}{n} = 1.$$

□

Proof. **Statement 2.**
First, let us reformulate the statement. Let $Z_i \sim N(0, 1)$. Then

$$X_i = \mu + \sigma Z_i \text{ and } \overline{X} = \mu + \sigma\overline{Z}.$$

Thus

$$\frac{\sum_{i=1}^n (X_i - \overline{X})^2}{\sigma^2} = \frac{\sum_{i=1}^n (\sigma Z_i - \sigma\overline{Z})^2}{\sigma^2} = \sum_{i=1}^n (Z_i - \overline{Z})^2,$$

and it suffices to show that the latter follows a χ^2_{n-1} distribution. We proceed by induction. When $n = 1$,

$$\sum_{i=1}^n (Z_i - \overline{Z})^2 = 0.$$

When $n = 2$,

$$\begin{aligned}\sum_{i=1}^n (Z_i - \overline{Z})^2 &= \left(Z_1 - \frac{Z_1 + Z_2}{2}\right)^2 + \left(Z_2 - \frac{Z_1 + Z_2}{2}\right)^2 \\ &= \left(\frac{Z_1 - Z_2}{2}\right)^2 + \left(\frac{Z_2 - Z_1}{2}\right)^2 \\ &= \frac{(Z_1 - Z_2)^2}{2} \\ &= \left(\frac{Z_1 - Z_2}{\sqrt{2}}\right)^2.\end{aligned}$$

Now, we know $Z_1 - Z_2$ is normal, with

$$\mathbb{E}[Z_1 - Z_2] = 0,\ \mathbb{V}[Z_1 - Z_2] = \mathbb{V}[Z_1] + \mathbb{V}[Z_2] = 2,$$

so $Z_1 - Z_2 \sim N(0, 2)$. Then

$$\frac{Z_1 - Z_2}{\sqrt{2}} \sim N(0, 1),$$

and

$$\left(\frac{Z_1 - Z_2}{\sqrt{2}}\right)^2 \sim \chi^2_1$$

as desired.

Now, suppose the statement is true for $n = m$, i.e.

$$\sum_{i=1}^{m}(Z_i - \overline{Z}_m)^2 \sim \chi^2_{m-1},$$

where $\overline{Z}_m$ is the mean taken over m Z_i's. We want to establish the case for $n = m + 1$, i.e.

$$\sum_{i=1}^{m+1}(Z_i - \overline{Z}_{m+1})^2 \sim \chi^2_m.$$

Decomposition gives

$$\begin{aligned}\sum_{i=1}^{m+1}(Z_i - \overline{Z}_{m+1})^2 &= \sum_{i=1}^{m}(Z_i - \overline{Z}_{m+1})^2 + (Z_{m+1} - \overline{Z}_{m+1})^2 \\ &= \underbrace{\sum_{i=1}^{m}(Z_i - \overline{Z}_m)^2}_{(a)} + \underbrace{\sum_{i=1}^{m}(\overline{Z}_m - \overline{Z}_{m+1})^2}_{(b)} + \underbrace{(Z_{m+1} - \overline{Z}_{m+1})^2}_{(c)}.\end{aligned}$$

where the second equality follows from the mean Pythagorean theorem. By induction assumption, the part in (a) follows a χ^2_{m-1} distribution, so it remains to show (b) + (c) follows a χ^2_1 distribution. Note that

$$\overline{Z}_{m+1} = \frac{1}{m+1}\sum_{i=1}^{m+1} Z_i = \frac{m}{m+1}\cdot\left(\frac{1}{m}\sum_{i=1}^{m} Z_i\right) + \frac{1}{m+1}Z_{m+1} = \frac{m}{m+1}\overline{Z}_m + \frac{1}{m+1}Z_{m+1},$$

and we can really view $\overline{Z}_{m+1}$ as the weighted average of $\overline{Z}_m$ and Z_{m+1}. So, for (b),

$$\sum_{i=1}^{m}(\overline{Z}_m - \overline{Z}_{m+1})^2 = m(\overline{Z}_m - \overline{Z}_{m+1})^2 = m\left(\frac{1}{m+1}\overline{Z}_m - \frac{1}{m+1}Z_{m+1}\right)^2 = \frac{m}{(m+1)^2}\cdot(\overline{Z}_m - Z_{m+1})^2.$$

For (c),

$$(Z_{m+1} - \overline{Z}_{m+1})^2 = \left(\frac{m}{m+1}Z_{m+1} - \frac{m}{m+1}\overline{Z}_m\right) = \frac{m^2}{(m+1)^2}(Z_{m+1} - \overline{Z}_m)^2.$$

Thus

$$(b) + (c) = \frac{m^2 + m}{(m+1)^2}(Z_{m+1} - \overline{Z}_m)^2 = \frac{m}{m+1}(Z_{m+1} - \overline{Z}_m)^2 = \left(\sqrt{\frac{m}{m+1}}(Z_{m+1} - \overline{Z}_m)\right)^2.$$

But then

$$\begin{aligned}\mathbb{E}\left[\sqrt{\frac{m}{m+1}}(Z_{m+1} - \overline{Z}_m)\right] &= \sqrt{\frac{m}{m+1}}\big(\mathbb{E}[Z_{m+1}] - \mathbb{E}[\overline{Z}_m]\big) = 0, \\ \mathbb{V}\left[\sqrt{\frac{m}{m+1}}(Z_{m+1} - \overline{Z}_m)\right] &= \frac{m}{m+1}\cdot\big(\mathbb{V}[\overline{Z}_m] + \mathbb{V}[Z_{m+1}]\big) = \frac{m}{m+1}\left(\frac{1}{m} + 1\right) = 1.\end{aligned}$$

So

$$\left(\sqrt{\frac{m}{m+1}}(Z_{m+1}-\overline{Z}_m)\right)^2 \sim \chi_1^2.$$

We are not entirely done yet. By Definition 1.26, we need the two components

$$\sum_{i=1}^{m}(Z_i-\overline{Z}_m)^2 \text{ vs } \left(\sqrt{\frac{m}{m+1}}(\overline{Z}_m-Z_{m+1})\right)^2$$

to be independent to sum their dofs. Observe that (a) and $\overline{Z}_m$ are independent by (3) (which we will verify the validity in the next proof), and (a) and Z_{m+1} are independent by construction. Hence independence holds and

$$\sum_{i=1}^{m+1}(Z_i-\overline{Z}_{m+1})^2 \sim \chi_m^2.$$

□

Proof. **Statement 3.**
We will need Theorem 1.25 Property (3) to keep simplifying the problem at hand. First, it suffices to show

$$\overline{X} \perp\!\!\!\perp \begin{pmatrix} X_1-\overline{X} \\ X_2-\overline{X} \\ \vdots \\ X_n-\overline{X} \end{pmatrix},$$

since if this is true, functions of both sides would be independent as well. Next, we observe that

$$\begin{pmatrix} \overline{X} \\ X_1-\overline{X} \\ X_2-\overline{X} \\ \vdots \\ X_n-\overline{X} \end{pmatrix}$$

is normal in $\mathbb{R}^{n+1}$. The key here is the iid property: the independence of the X_i's guarantee that any linear combinations of them is also normal, hence joint normality. Usually the fact that X and Y are normal does not imply $(X\,Y)^T$ is jointly normal (for a concrete counterexample, see the

second answer of this post), which we need for Property (3) to work. Thus it remains to show that

$$\text{Cov}\left(\overline{X}, \begin{pmatrix} X_1 - \overline{X} \\ X_2 - \overline{X} \\ \vdots \\ X_n - \overline{X} \end{pmatrix}\right) = 0_n,$$

or

$$\text{Cov}(\overline{X}, X_i - \overline{X}) = 0$$

for all $i = 1, ..., n$. But then, by bilinearity of covariance and independence of X_i's,

$$\begin{aligned} \text{Cov}(\overline{X}, X_i - \overline{X}) &= \text{Cov}(\overline{X}, X_i) - \text{Cov}(\overline{X}, \overline{X}) \\ &= \frac{1}{n}\text{Cov}(X_i, X_i) - \mathbb{V}[\overline{X}] \\ &= \frac{\mathbb{V}[X_i]}{n} - \frac{\sigma^2}{n} \\ &= \frac{\sigma^2}{n} - \frac{\sigma^2}{n} = 0 \end{aligned}$$

for all i. Hence statement (3) holds. □

Proof. **Statement 4.**
Statement (4) follows immediately from (1), (2), (3). Manipulating the expression gives

$$\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sqrt{\frac{1}{n-1}\sum_{i=1}^n (X_i - \overline{X})^2}} = \frac{\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sigma}}{\sqrt{\frac{1}{n-1}\left(\sum_{i=1}^n \frac{(X_i - \overline{X})^2}{\sigma^2}\right)}},$$

where we have the numerator $\sim N(0, 1)$ by (1), the denominator $\sim \chi^2_{n-1}$ by (2), and the independence of the two by (3). By Definition 1.26,

$$\sqrt{n} \cdot \frac{\overline{X} - \mu}{\sqrt{\frac{1}{n-1}\sum_{i=1}^n (X_i - \overline{X})^2}} \sim t_{n-1}$$

as desired. □

1.12. **Cauchy Distribution.**

Here are some important facts about Cauchy distribution:

- A Cauchy RV does not have mean nor variance, as $x \cdot p(x)$ is not integrable;
- WLLN does not hold for Cauchy distribution. Recall that for $X_1, ..., X_n \overset{\text{iid}}{\sim} N(0, 1)$, $\overline{X} \sim N(0, 1/n)$, and $\overline{X} \to 0$ as $n \to \infty$. However, for $X_1, ..., X_n \overset{\text{iid}}{\sim}$ Cauchy, $\overline{X} \sim$ Cauchy.

This is because Cauchy has heavy tails, which are commonly seen in financial data. In fact, Cauchy distribution represents the boundary with respect to the effect of averaging: if a distribution has a lighter tail than Cauchy (for example, normal distribution and t-distribution), then larger samples would reduce the variance of the average; conversely, if a distribution has an even heavier tail than Cauchy (these are known as stable distributions), taking large samples would actually increase the variance of the average.

2. Hypothesis Testing

2.1. **Set-up.**

This section serves as a simplified review on the set-up and notation related to hypothesis testing (HT). For a more comprehensive introduction, see 244 cheatbook.

There are several components to a HT:

- Two hypotheses: the *null hypothesis* H_0 and the *alternative hypothesis* H_1. The treatment of the two is asymmetric, i.e. we take the null as the default and we do not reject it unless there is 'sufficient' evidence to do so (we will make concrete what that means soon).
- A *test-statistic*: $T = T(X_1, ..., X_n)$.
- A *rejection region* (RR) R.

Given these components, we have

- a testing procedure: we reject H_0 when $T \in R$;
- two types of errors:
  - *Type I error*, or $\mathbb{P}(T \in R \mid H_0)$,
  - *Type II error*, or $\mathbb{P}(T \notin R \mid H_1)$.
- requirement: under the constraint that the type I error
$$\mathbb{P}(T \in R \mid H_0) \leq \alpha$$
for some given $\alpha > 0$, we want the type II error $\mathbb{P}(T \in R \mid H_1)$ to be as small as possible.

Fixing the level of type I error makes sense (we are defining the maximum tolerance for error given H_0), but what is the motivation for minimizing type II error? This can be traced to the concept of power.

Definition 2.1. The *power* of a HT is
$$\mathbb{P}(T \in R \mid H_1) = 1 - \mathbb{P}(T \notin R \mid H_1).$$

In other words, the power of a HT is a measure of the accuracy of the test given H_1. Given a reasonably designed HT, there is usually a trade-off between type I and type II error (see Figure 3).

Example 2.2. Let $X_1, ..., X_n \overset{\text{iid}}{\sim} N(\mu, 1)$. Suppose $H_0 : \mu = 0$ and $H_1 : \mu > 0$. Since we are testing the mean, the sample mean $\overline{X}$ should be a good test-statistic. Intuitively, we want to reject the null when the test-statistic is 'significantly' larger than 0, i.e. $\overline{X} > c$ for some c. Given α, we have
$$\alpha = \mathbb{P}\left(\overline{X} > c \mid H_0\right) = \mathbb{P}\left(\sqrt{n}\overline{X} > \sqrt{n}c \mid H_0\right) = \mathbb{P}(N(0,1) > \sqrt{n}c \mid H_0).$$
since under H_0, $\overline{X} \sim N(0, 1/n)$. Finding the corresponding quantile gives
$$c = \frac{z_{1-\alpha}}{\sqrt{n}}.$$
Usually, in a HT, we would try our best to reject the null, which means we want this threshold to be as low as possible. From the expression, there are two ways to do so: either we can increase the

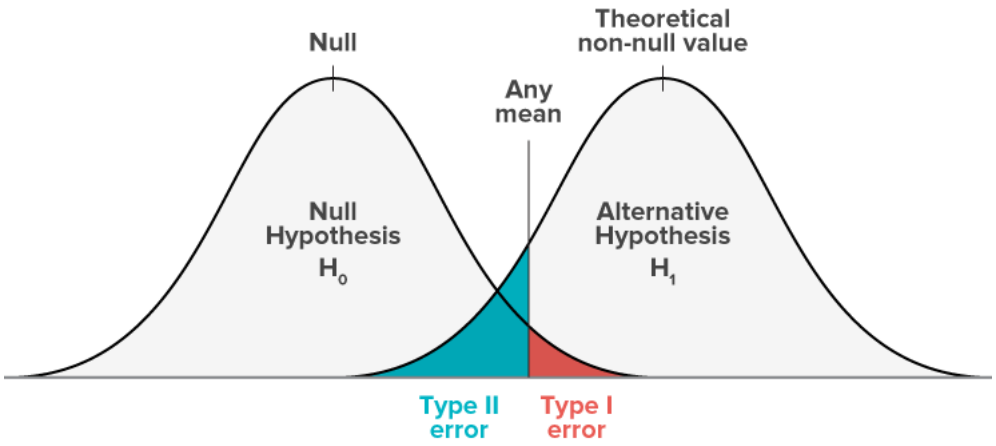


FIGURE 3. Relation between type I & type II error

sample size n or we can increase α such that $1-\alpha$ is smaller and the quantile visually shifts to the left. Note that in real life, changing α is less feasible due to research community guidelines.

Now let us try to analyze the power. Define the power function to be

$$g(\mu) = \mathbb{P}_\mu\left(\overline{X} > \frac{z_{1-\alpha}}{\sqrt{n}}\right),$$

where the subscript is present to emphasize the dependence on μ. Also note that

$$g(0) = \mathbb{P}\left(\overline{X} > \frac{z_{1-\alpha}}{\sqrt{n}} \;\middle|\; \mu = 0\right) = \alpha$$

by our construction of type I error. The power function represents the probability that we reject the null given the actual value is μ. Now, it is crucial not to confuse this function with the concept of power in Definition 2.1: even though the two are closely related, $g(\cdot)$ is a more general construction defined in relation to μ and the threshold we are working with. In other words, if the usual concept of power is associated with the notion of 'correctness', the power function here focuses on presenting the pure magnitude of probability: the correctness part only comes into being when we interpret things w.r.t. specific μ values. We can even take μ to be negative if we want to, though that would be an artificial construct and does not have meaning w.r.t. the original set-up. If this does not make sense, there will be more explanation afterward.[1]

Manipulating the expression of the function gives

$$g(\mu) = \mathbb{P}_\mu\left(\overline{X} > \frac{z_{1-\alpha}}{\sqrt{n}}\right)$$

[1]Another resource that may help immensely in understanding the power function is Section 8.3 in Castella's *Statistical Inference*.

$$= \mathbb{P}_\mu \left(\sqrt{n}(\overline{X} - \mu) > \sqrt{n} \left(\frac{z_{1-\alpha}}{\sqrt{n}} - \mu \right) \right)$$
$$= \mathbb{P}(N(0,1) > z_{1-\alpha} - \sqrt{n}\mu),$$

where the subscript is taken out at the end because $N(0,1)$ no longer depends on μ. Below is a visualization of the power function.

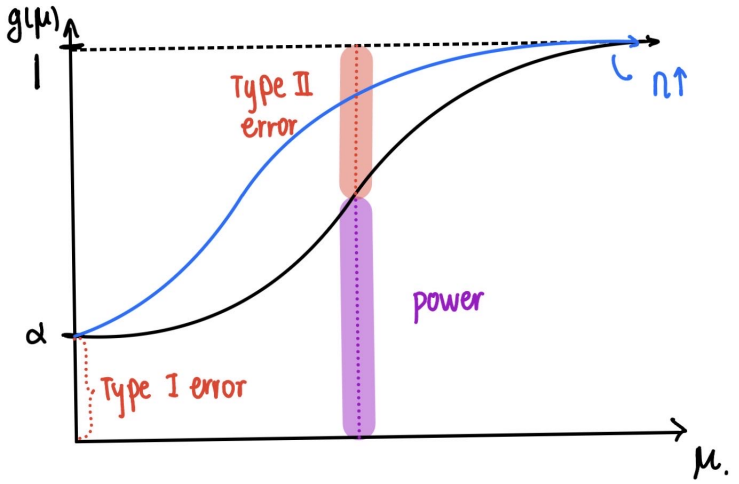


FIGURE 4. Power Function for Example 2.2

Several remarks:

(1) $g(\mu) \to 1$ either when $n \to \infty$ or $\mu \to \infty$. We show the first case with the blue curve (increased n). The latter also makes intuitive sense because the further way the 'true' value of μ is from 0, the more likely our rejection of the null is correct.
(2) Note that in alignment with above, we have the intercept as α. This would not make sense w.r.t. to the original set-up (we cannot have $\mu = 0$ in both the null and the alternative), but this is alright with the general construction. If you must make reference to the original set-up, think about this intercept as a limit, i.e. 'the power as $\mu \to 0$'.
(3) If we focus on a specific value of $\mu > 0$, we recover the 'power = 1 - type II error' interpretation.

Example 2.3. In Example 2.2, our null is set to be $H_0 : \mu = 0$. This is a *simple* null. Let us instead consider *composite* null, where μ can take on multiple values under the null. Let us take the same set-up as in Example 2.2, except now the null is composite:

- $H_0 : \mu \leq 0$, $H_1 : \mu > 0$;
- $X_1, ..., X_n \overset{\text{iid}}{\sim} N(\mu, 1)$;

- we reject H_0 when $\overline{X} > c$, where c is the threshold chosen such that the worst-case type I error is controlled by α.

Per the last item,

$$\begin{aligned}\sup_{\mu\leq 0}\mathbb{P}_\mu(\overline{X}>c) &= \sup_{\mu\leq 0}\mathbb{P}_\mu\big(\sqrt{n}(\overline{X}-\mu)>\sqrt{n}(c-\mu)\big)\\ &= \sup_{\mu\leq 0}\mathbb{P}\big(N(0,1)>\sqrt{n}c-\sqrt{n}\mu\big)\\ &= \mathbb{P}\big(N(0,1)>\sqrt{n}c\big)=\alpha,\end{aligned}$$

where the last equality follows from the fact that we want the probability to be as large as possible and that μ is maximized at 0. Hence we get

$$c=\frac{z_{1-\alpha}}{\sqrt{n}},$$

exactly the same as the simple null case.

Example 2.4. Let us alter the set-up to Examples 2.2 and 2.3 a bit. Suppose $X_1,...,X_n \overset{\text{iid}}{\sim} N(\mu,\sigma^2)$ for some unknown σ^2. The hypotheses are $H_0:\mu=0$, $H_1:\mu\neq 0$. The goal is to find a test-statistic whose distribution does not depend on σ^2 under H_0. Since X_i's are normal, an obvious choice here is the t-statistic

$$T(X)=\frac{\sqrt{n}\overline{X}}{\sqrt{\frac{\sum_{i=1}^n(X_i-\overline{X})^2}{n-1}}}=\frac{\sqrt{n}\overline{X}}{S}\overset{H_0}{\sim} t_{n-1}.$$

We still want to reject H_0 when the test statistic deviates too much from 0, i.e.

$$\left|\frac{\sqrt{n}\overline{X}}{S}\right|>c$$

or

$$|\overline{X}|>\frac{cS}{\sqrt{n}}$$

for some c. It turns out the threshold is $c=t_{n-1,\ 1-\alpha/2}$, and we get

$$|\overline{X}|>\frac{t_{n-1,\ 1-\alpha/2}\cdot S}{\sqrt{n}}.$$

Example 2.5. Let us take the same set-up as Example 2.4 with the tests changed: now $H_0:\mu=\mu^*$, $H_1:\mu\neq\mu^*$. We use the t-statistic again to get

$$|\overline{X}-\mu^*|>\frac{t_{n-1,\ 1-\alpha/2}\cdot S}{\sqrt{n}}.$$

This means

$$\mathbb{P}_{\mu^*}\left(|\overline{X}-\mu^*|>\frac{t_{n-1,\ 1-\alpha/2}\cdot S}{\sqrt{n}}\right)=\alpha,$$

or

$$\mathbb{P}_{\mu^*}\left(|\overline{X}-\mu^*|<\frac{t_{n-1,\ 1-\alpha/2}\cdot S}{\sqrt{n}}\right)=1-\alpha.$$

Looks familiar? This should remind you of CIs! This is only one example of the *duality* between HT and CI: they are two sides of the same coin, only that HT cares about rejection probability α and CI the coverage probability $1-\alpha$. A HT automatically produces a CI if we take the complement of the rejection region. On the other hand, when constructing a CI, we can pretend there exists a HT in question, for which failure to situate inside the interval would result in rejection.

Recall that we covered three methods of constructing CIs: Wald, Wilson and VST. Do there exist corresponding methods in the construction of HT? Let us look at an example with our old friend, the Poisson distribution.

Example 2.6. Let $X_1, ..., X_n \overset{\text{iid}}{\sim} \text{Poisson}(\lambda)$. Consider $H_0 : \lambda = \lambda^*$, $H_1 : \lambda \neq \lambda^*$. Recall under the null, CLT gives

$$\sqrt{n} \cdot \frac{\overline{X} - \lambda^*}{\sqrt{\lambda^*}} \to N(0,1).$$

So let us take the LHS to be the test-statistic. In alignment with previous examples, we reject when

$$\left| \sqrt{n} \cdot \frac{\overline{X} - \lambda^*}{\sqrt{\lambda^*}} \right| > z_{1-\alpha/2}$$

for which we can then find the corresponding CI. Why is this so straightforward? Recall that we developed the three methods for CI construction incorporating the fact that we do not know μ^* (or rather λ^* here). However, λ^* is known here: we need to designate a specific value for the HT to make sense in the first place. We do not need to do anything except do the calculations and reject when we should. Again, if we take the complement of the event:

$$\mathbb{P}\left(\left| \sqrt{n} \cdot \frac{\overline{X} - \lambda^*}{\sqrt{\lambda^*}} \right| \leq z_{1-\alpha/2} \right) \to 1 - \alpha.$$

This gives a CI corresponding to Wilson's method.

2.2. **P-Value.**

In this section, we want to address common issues with α. First, which α to choose always provoke debates. Current research community put the standard as $\alpha = 0.05$, but that is an historical artifact (credit to Fisher) rather than logical reasoning. Another more important issue, however, is the difficulty of conversion between α and $z_{1-\alpha}$, $t_{1-\alpha}$, or whatever other quantile depending on the distribution used. A good remedy, instead, is to find a uniformly-distributed test-statistic such that its threshold exactly correspond to α, and we can reject the null when the test-statistic is small. This is the main idea that lies behind *p-value*, but for the further exploration, we need the tool of *CDF transform*.

Theorem 2.7. *Let $F(t) = \mathbb{P}(X \leq t)$ be the CDF of a continuous RV X. Then $Y = F(X) \sim$ Uniform$(0,1)$.*

Proof.

$$\mathbb{P}(Y \leq t) = \mathbb{P}(F(X) \leq t) = \mathbb{P}\left(X \leq F^{-1}(t)\right) = F\left(F^{-1}(t)\right) = t.$$

One thing to note is that the proof requires existence of F^{-1}, so the theorem does not apply to all continuous RVs. □

Let us look back to Example 2.2 quickly to apply this technique to the normal distribution. We have $\sqrt{n}\overline{X} \sim N(0,1)$ under H_0. Note that by symmetry of normal distribution, we also have $-\sqrt{n}\overline{X} \sim N(0,1)$ under H_0, and we have two options for the transform. Remember our goal is to reject the null when the test-statistic is small. If we go along with the first option, then we would only reject when the test-statistic is large. So we use the second option, which gives

$$\Phi(-\sqrt{n}\overline{X}) \sim \text{Uniform}(0,1),$$

and we reject when $p(x) = \Phi(-\sqrt{n}\overline{X}) \leq \alpha$. The test-statistic defined by $\Phi(-\sqrt{n}\overline{X})$ is the p-value statistic. Note that p-value is a RV, with realizations dependent on data.

In relation to the first issue we brought up (which α to choose), let us consider the following: for scientific discovery, can we change the α we use for significance after determining the p-value based on data? The answer is that we should not. Note that we only have the incentive to increase α if the p-value falls outside the rejection region, which is a random event. On the other hand, α as a representation of the probability of erroneous rejection is not random. So the post-adjustment would render the entire experiment meaningless and does not make sense in itself. Such action is called *p-hacking* and is generally considered unethical for this and other practical purposes.

Next, let us consider p-value for two-sided tests.

Example 2.8. Let us continue with Example 2.2. We have $X_1, ..., X_n \overset{\text{iid}}{\sim} N(\mu, 1)$, $H_0 : \mu = 0$, $H_1 : \mu \neq 0$, and we reject H_0 when $\sqrt{n}|\overline{X}| > z_{1-\frac{\alpha}{2}}$. The CDF transform tool would not be convenient here, so let us work directly with area:

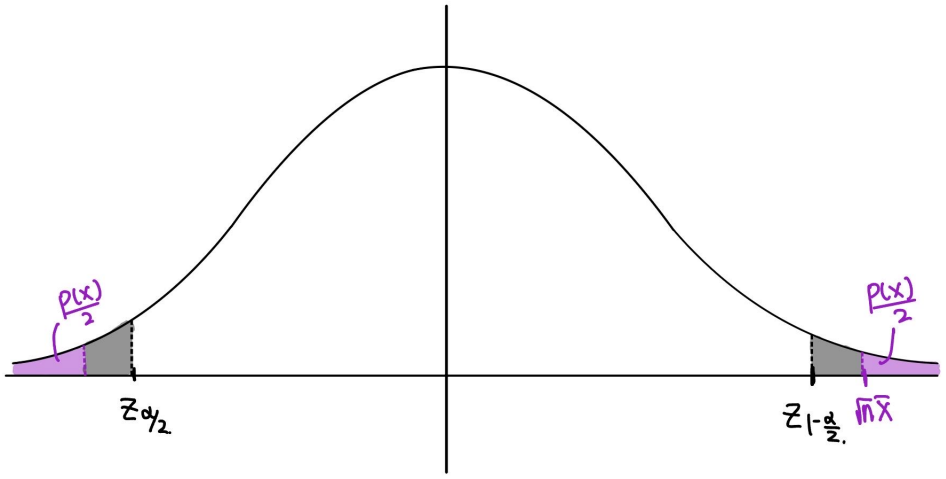


FIGURE 5. Two-Sided P-Value Graph

Basically, in accordance with two-sided test, we only have density $p(x)/2$ on each side, and we reject if and only if $p(x)/2 < \alpha/2$, or $p(x) < \alpha$. This is a manually-constructed definition, and we still need to make sure that it follows a Uniform(0,1) distribution. So

$$\mathbb{P}_0(p(x) < \alpha) = \mathbb{P}_0\left(-\sqrt{nX} < z_{\frac{\alpha}{2}} \mid \sqrt{nX} > z_{1-\frac{\alpha}{2}}\right) = \mathbb{P}_0\left(|\sqrt{nX}| > z_{1-\frac{\alpha}{2}}\right) = \alpha$$

under H_0, and we are done.

To conclude this section, we note that p-value is nothing new; it is just another method of reporting our conclusion based on data.

2.3. **Multiple Testing.**

True to its name, *multiple testing* (aka. multiple hypothesis test (MHT), multiple comparison or simultaneous inference) occurs when we test several HTs, usually closely related to one another, at the same time. Multiple testing has frequent applications in real life: testing for significant genes associated with a certain disease, identifying relevant factors for salary levels, and so on. We will take genetic association detection as an example throughout this section.

Here is the general set-up of a MHT: we have data in the form of a sequence of p-values $p_1, p_2, ..., p_n$ (allows for easy interpretation) with the associated HTs $H_{01}, H_{02}, ..., H_{0n}$, where $H_{0i} : p_i \sim \text{Uniform}(0, 1)$, for all $i \in \{1, ..., n\}$. In other words, p_i's should behave like noise under the H_{0i}'s. We are mainly concerned with two questions:

(1) Does there exist $p_i \not\sim \text{Uniform}(0, 1)$ (does there exist significant genes)?
(2) If yes, identify all i's for which it is so.

Let us start with the first question. We still want to perform rejections based on α, but things are not as straightforward as single HTs. Even if all p_i's do follow Uniform(0,1) in reality, we can still have quite a few $p_i < \alpha$ by pure chance if n is sufficiently large. Another way to look at this is that if we still set type I error to be α for each individual test, then the probability that we reject at least one test by mistake is

$$1 - \mathbb{P}(\text{no test is rejected}) = 1 - (1 - \alpha)^n,$$

assuming the tests are independent. This can quickly deviate from α when n grows large. For example, if we take $\alpha = 0.01$, this probability equals $1 = 0.366 = 0.634$. For a more well-known example, see here. This does not look ideal. Hence we need much more stringent benchmark (i.e. stronger threshold) for the *global null*.

2.3.1. *Global Null.*

First, the point of MHT is working with one overall test instead of n individual HTs separately. To this purpose, we define the *global null*

$$H_0 := \bigcap_{i=1}^{n} H_{0i},$$

i.e. all $p_i \sim \text{Uniform}(0, 1)$. The alternative H_1 is then that at least one H_{0i} is rejected.

To address the issue with controlling the type I error at α, we introduce the *Bonferroni (correction) test*, which rejects H_0 when

$$\min_i p_i \leq \frac{\alpha}{n}.$$

The last term is known as the *Bonferroni correction factor*. To show that the Bonferroni test indeed controls the error rate at α, note that the *union bound*

$$\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$$

holds for any measurable A, B. It follows that

$$\begin{aligned}
\text{Type I error} &= \mathbb{P}\left(\min_i p_i \leq \frac{\alpha}{n} \;\middle|\; H_0\right) \\
&= \mathbb{P}\left(p_1 \leq \frac{\alpha}{n} \text{ or } p_2 \leq \frac{\alpha}{n} \text{ or ... or } p_n \leq \frac{\alpha}{n} \;\middle|\; H_0\right) \\
&\leq \sum_{i=1}^{n} \mathbb{P}\left(p_i \leq \frac{\alpha}{n} \;\middle|\; H_0\right) \\
&= n \cdot \frac{\alpha}{n} = \alpha.
\end{aligned}$$

An upside of Bonferroni test is that it poses no assumption on independence (so the genes in question can be correlated). The downside is that since it is based on an inequality, the actual type I error (especially if p_i's are independent of each other) can be much smaller than α, and the method is rather conservative. Recall from Section 2.1 that we always to want to make full use of the given error allowance to increase power. This motivates some improvements.

Theorem 2.9. *When the p_i's are independent, Simes' test is an improvement of Bonferroni's, i.e.*

(1) if Bonferroni rejects the null, so will Simes,
(2) the Bonferroni rejections always form a subset of Simes rejections.

It follows that Simes test is more aggressive and yields more power. The details of the test are beyond the scope of the class and can be found here, but the rough procedure involves ordering p_i's in order of increasing magnitude then cleverly assign thresholds based on the order and rejects by order-statistics.

2.3.2. *Familywise Error Rate.*

Let us now investigate the second question of compiling a list of i's where H_{0i} is rejected. To this end, let us decompose $\{1, ..., n\}$ into I_0 and I_1, where

$$I_0 := \{i : H_{0i} \text{ is true}\}$$

is the null set, and we denote its size by n_0. Then I_1 is the set of i's for which H_{0i} is not true, and we denote its size by n_1. It follows that $n = n_0 + n_1$.

Just as we need a global test for error in investigating the first problem, here we need a global measure for error.

Definition 2.10. The *familywise error rate* (FWER) is the probability of making at least one false rejection, i.e. the probability that there exists $i \in I_0$ with H_{0i} rejected.

In the genetic testing setting, the FWER can be interpreted as "the maximum probability for which the list given contains useless genes".

There are several tests that are constructed based on the concept of FWER. The first, Bonferroni test, has the same name but different usage to the test above. The Bonferroni test here checks each H_{0i} and rejects when

$$p_i \leq \frac{\alpha}{n}$$

for each i. The FWER, by the union bound,

$$\text{FWER} = \mathbb{P}\left(\bigcup_{i \in I_0} p_i < \frac{\alpha}{n}\right) \leq \sum_{i \in I_0} \mathbb{P}\left(p_i < \frac{\alpha}{n}\right) = \frac{n_0}{n}\alpha \leq \alpha,$$

and is thus controlled by α. Note that we only sum over the null set.

Another remark is that since Bonferroni examines each individual hypothesis (local application), it is forgoing information derived from the family of hypotheses as a whole and thus sacrificing more power than necessary. This gives way to immediate free improvements.

Remarks 2.11.

(1) Holm test always improves Bonferroni.
(2) When p_i's are independent, Hochberg test offers a further improvement.

Potential improvements, however, do not stop here: there exists strong objection to using FWER as the basis for MHT, for the simple reason that it is too good at eliminating errors. In the genetic association tests, researchers do not need every single gene on the rejection list to be truly significant; doing that would usually leave them with a list way too short. Instead, they are happy as long as the majority of them, say 90%, is actually significant. Herein lies the motivation for a new measure of error.

Notation 2.12. Given a family of hypotheses $H_{01}, ..., H_{0n}$, we denote $R :=$ the total number of rejections and $V :=$ the number of false rejections.

Definition 2.13. The *false discovery proportion* (FDP) is

$$\begin{cases} \frac{V}{R}, & R > 0, \\ 0, & R = 0, \end{cases} \quad = \quad \frac{V}{\max(R, 1)}.$$

Definition 2.14. The *false discovery rate* (FDR) is

$$\mathbb{E}[\text{FDP}] = \mathbb{E}\left[\frac{V}{\max(R, 1)}\right].$$

Note that FDP is a RV by virtue of p_i's being random while FDR is deterministic.

Theorem 2.15. *FDR* $\leq$ *FWER.*

Proof. Note that by definition, FWER $= \mathbb{P}(V > 0)$. Then by law of iterated expectation,

$$\begin{aligned}
\text{FDR} &= \mathbb{E}\left[\frac{V}{\max(R,1)}\right] \\
&= \mathbb{E}\left[\frac{V}{\max(R,1)} \cdot \mathbb{1}[V>0]\right] + \mathbb{E}\left[\frac{V}{\max(R,1)} \cdot \mathbb{1}[V=0]\right] \\
&= \mathbb{E}\left[\frac{V}{R} \cdot \mathbb{1}[V>0]\right] \\
&\leq \mathbb{E}\big[\mathbb{1}[V>0]\big] = \mathbb{P}(V>0) = \text{FWER}.
\end{aligned}$$

□

Now let us introduce tests that is designed w.r.t. FDR, the most important of which is the *Benjamini-Hochberg procedure*. First, we order $p_1, ..., p_n$ from smallest to largest such that

$$p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(n)}.$$

For each $p_{(i)}$, we assign the rejection threshold $\alpha \cdot i/n$. Then the algorithm searches from right to left; we stop if $p_{(j)} \leq \alpha \cdot j/n$ and reject all $p_{(1)}, p_{(2)}, ..., p_{(j)}$.

Theorem 2.16. *If* p_i*'s are independent, then*

$$FDR(BH) = \frac{n_0}{n}\alpha \leq \alpha.$$

Two important remarks: first, for FDR, α is defined as the proportion of false discovery tolerated and is generally set to be 20% (rather than 0.05). Secondly, even though the procedure technically requires independence, it usually yields good approximations even when p_i's are not. If we want to be rigorous, we can consider the *Benjamini-Hoshberg-Yekutieli procedure* instead, which employs the exact method, except the thresholds can reduced by a factor of

$$s(n) = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \sim \log(n).$$

The BHY procedure controls the FDR regardless in independence. In real life, however, when n is large, $\log(n)$ can still be large, and researchers are generally unwilling to make such sacrifice, sticking to the BH approximation instead.

3. Linear Regression

3.1. Univariate.

The goal of this section is to solve the following problem: given a sequence of **data** $\{(x_i, y_i)\}_{i=1}^n$, we want to fit a linear model that would help **predict** any new data points with the greatest efficiency (i.e. low bias and low variance) as possible. This is a common procedure in all areas of research and is used to investigate the relationship between two factors, for (a boring) example, housing price with respect to the size of the house, etc. This process is called *linear regression* and for the univariate case, we will use the model

$$y_i = \beta_0 + \beta_1 x_i + \sigma z_i, \tag{3.1}$$

where z_i is the error term. There exists several distinct mainstream interpretations of z_i as the measurement error, as any unobservable characteristics, etc.

Throughout the section, we will denote scalars or single data points with normal letters and vectors with bolded ones.

we will denote RVs with uppercase letters and their realizations with lowercase ones.

Assumption 3.2. We assume the following for our linear regression model:

(1) The errors are normally distributed, i.e. $z_1, ..., z_n \overset{\text{iid}}{\sim} N(0, 1)$. This allows us to reframe

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2) \tag{3.3}$$

independently for $i = 1, ..., n$, i.e. y_i is a sequence of normally-distributed RV with the aforementioned specification.

(2) x_i's are fixed, i.e. not random. This does not mean they only take on one value (in that case we would say x_i's are constant), but that their values are set prior to data collection. This is called *fixed design* (in the sense that we manually design the *Gram matrix* - a concept that we will discuss more later) versus *random design*, where both x_i's and y_i's are assumed to be random (as in the case of field experiments or observational data). All the characteristics we cover for the fixed design can be naturally extended to random design as well.

To find an appropriate estimator of β_0 and β_1, let us calculate the MLE as our guess. We start with the likelihood function:

$$\begin{aligned} p(y) = \prod_{i=1}^n p(y_i) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right) \\ &= \prod_{i=1}^n \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2\right), \end{aligned}$$

which means

$$\log(p(y)) = -n\log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2.$$

The first term on the RHS does not depend on β_0 or β_1 and so can be treated as a constant. Thus, the likelihood maximization problem can be reduced to minimizing the second term on the RHS, i.e.

$$\max_{\beta_0,\beta_1} \log(p(y)) = \min_{\beta_0,\beta_1} S(\beta_0,\beta_1), \;\; S(\beta_0,\beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2.$$

In other words, MLE estimators also minimize MSE, and thus they are the desired estimators.

Before we proceed, let us introduce some simplifying notations we will use throughout the section.

Notation 3.4. We denote

$$\begin{aligned}
\overline{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i,\ \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \\
\overline{x^{(j)}} &= \frac{1}{n}\sum_{i=1}^{n} x_i^j, \\
\overline{xy} &= \frac{1}{n}\sum_{i=1}^{n} x_i y_i, \\
V &= \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2, \\
C &= \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}).
\end{aligned}$$

(As an aside, the notations V and C are inspired by the fact that the sums are sample analogs of the variance of x_i's and covariance between x_i and y_i.

To find a concrete expression for the estimators in terms of data, we take the partial derivatives

$$\begin{aligned}
\frac{\partial S}{\partial \beta_0} &= 2\sum_{i=1}^{n} (\beta_0 + \beta_1 x_i - y_i) = 0, \\
\frac{\partial S}{\partial \beta_1} &= 2\sum_{i=1}^{n} (\beta_0 + \beta_1 x_i - y_i) x_i = 0.
\end{aligned}$$

Dividing both sides by n, we get

$$\beta_0 + \beta_1 \overline{x} - \overline{y} = 0, \tag{3.5}$$

$$\beta_0 \overline{x} + \beta_1 \left(\overline{x}\right)^2 - \overline{xy} = 0. \tag{3.6}$$

Multiplying (3.5) on both sides by $\overline{x}$ gives

$$\beta_0 \overline{x} + \beta_1 \overline{x}_1 - \overline{x} \cdot \overline{y} = 0. \tag{3.7}$$

Subtracting (3.7) from (3.6) gives

$$\widehat{\beta}_1 \left(\overline{x^{(2)}} - \left(\overline{x}\right)^2\right) - \left(\overline{xy} - \overline{x} \cdot \overline{y}\right) = 0.$$

This gives the first major result of the section.

Theorem 3.8. *The univariate linear regression slope coefficient is*

$$\widehat{\beta}_1 = \underbrace{\frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^{(2)}} - (\overline{x})^2}}_{(1)} = \underbrace{\frac{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2}}_{(2)} = \underbrace{\frac{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x}) y_i}{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2}}_{(3)} = \underbrace{\frac{\frac{1}{n}\sum_{i=1}^n x_i (y_i - \overline{y})}{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2}}_{(4)}.$$

Proof. Definitions (1) and (2) follows directly from above. It remains to discuss why (3) and (4) are true. WLOG examine (3). Note that

$$\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y}) = \frac{1}{n}\sum_{i=1}^n (x_i - \overline{x}) y_i - \frac{\overline{y}}{n}\sum_{i=1}^n (x_i - \overline{x}).$$

But then

$$\sum_{i=1}^n (x_i - \overline{x}) = 0,$$

so

$$\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y}) = \frac{1}{n}\sum_{i=1}^n (x_i - \overline{x}) y_i$$

as desired. Definition (4) follows from a similar logic and is left to readers as an exercise. □

Remarks 3.9.

(1) Using our notation, we can write $\widehat{\beta}_1 = C/V$, which we will use sometimes. In particular, $\widehat{\beta}_1$ can be interpreted as the ratio between the joint variability of x_i and y_i and the variability of x_i itself.
(2) From (3.5), we have

$$\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}.$$

Next, let us calculate the expectation, variance, and covariance of our estimators, and we will formulate the result over several theorems.

Theorem 3.10. *The expectations of $\widehat{\beta}_1$ and $\widehat{\beta}_0$ are*

$$\mathbb{E}\left[\widehat{\beta}_1\right] = \beta_1, \mathbb{E}\left[\widehat{\beta}_0\right] = \beta_0.$$

It follows that both estimators are **unbiased**. Even though bias is usually considered a secondary measurement of efficiency (compared to MSE), having no bias is still a desirable property. We only intentionally want biased estimators when there exists additional assumptions. Examples include ridge regressions (when highly correlated regressors suffer from multicollinearity in a multivariate setting) and lasso regressions (which purposefully reduce some coefficients to 0 to render a smaller model for better interpretation). For more details, see Sections 3.5 and 3.8.

Proof. Let us calculate $\mathbb{E}\left[\widehat{\beta}_1\right]$ first. By linearity of expectations,

$$\begin{aligned}\mathbb{E}\left[\widehat{\beta}_1\right] &= \mathbb{E}\left[\frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})(y_i-\overline{y})}{V}\right] \\ &= \frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})\,\mathbb{E}\left[y_i-\overline{y}\right]}{V} \\ &= \frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})\left[(\beta_0+\beta_1 x_i)-(\beta_0+\beta_1\overline{x})\right]}{V} \\ &= \frac{\beta_1\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})^2}{V} = \beta_1.\end{aligned}$$

For $\widehat{\beta}_0$,

$$\mathbb{E}\left[\widehat{\beta}_0\right] = \mathbb{E}\left[\overline{y}-\widehat{\beta}_1\overline{x}\right] = (\beta_0+\beta_1\overline{x})-\beta_1\overline{x} = \beta_0.$$

□

Theorem 3.11. *The variances of $\widehat{\beta}_1$ and $\widehat{\beta}_0$ are*

$$\mathbb{V}\left[\widehat{\beta}_1\right] = \frac{\sigma^2}{nV},\ \mathbb{V}\left[\widehat{\beta}_0\right] = \frac{\sigma^2}{n}\left(1+\frac{\overline{x}^2}{V}\right).$$

Proof. To calculate the variance of $\widehat{\beta}_1$, we use Definition (3):

$$\mathbb{V}\left[\widehat{\beta}_1\right] = \mathbb{V}\left[\frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})y_i}{V}\right] = \frac{\frac{1}{n^2}\mathbb{V}\left[\sum_{i=1}^n (x_i-\overline{x})y_i\right]}{V^2} = \frac{V\cdot n\mathbb{V}[y_i]}{V^2} = \frac{\sigma^2}{nV}.$$

For $\widehat{\beta}_0$,

$$\begin{aligned}\mathbb{V}\left[\widehat{\beta}_0\right] &= \mathbb{V}\left[\overline{y}-\widehat{\beta}_1\overline{x}\right] \\ &= \mathbb{V}\left[\overline{y}\right] + \mathbb{V}\left[\widehat{\beta}_1\overline{x}\right] - 2\ \mathrm{Cov}\left(\overline{y},\widehat{\beta}_1\overline{x}\right) \\ &= \frac{\sigma^2}{n} + \overline{x}^2\cdot\frac{\sigma^2}{nV} - 2\overline{x}\ \mathrm{Cov}\left(\overline{y},\widehat{\beta}_1\right),\end{aligned}$$

so it remains to determine $\mathrm{Cov}\left(\overline{y},\widehat{\beta}_1\right)$:

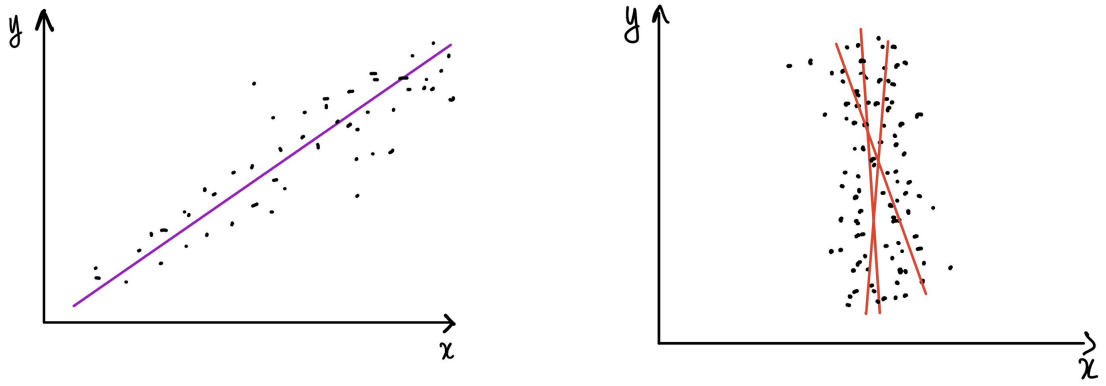
$$\mathrm{Cov}\left(\overline{y},\widehat{\beta}_1\right) = \mathrm{Cov}\left(\overline{y},\frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})(y_i-\overline{y})}{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})^2}\right) = \frac{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})\mathrm{Cov}(\overline{y},y_i-\overline{y})}{\frac{1}{n}\sum_{i=1}^n (x_i-\overline{x})^2} = 0.$$

Hence

$$\mathbb{V}\left[\widehat{\beta}_0\right] = \frac{\sigma^2}{n}\left(1+\frac{\overline{x}^2}{V}\right).$$

□

Remark 3.12. The expressions for the variances of the coefficient estimators give some insight into the desired properties of the experimental set-up. First, for $\widehat{\beta}_1$:



(A) Case 1: Large V (B) Case 2: Small V

FIGURE 6. Effect of Variability of x_i on Variance of $\widehat{\beta}_1$

(1) σ^2 is the 'noise level'. This is predetermined in the model, and there is not much we can do about it.
(2) n is the sample size. We can see that increasing the sample size helps reduce variance of $\widehat{\beta}_1$.
(3) V represents the spread of x and inversely controls for the variability of the design. To see this, consider the two cases in Figure 6. We can see that, when the spread of x_i is small, it is harder to pin down the exact slope of the linear regression. Thus during the experimental set-up, if the costs of small and large spreads are similar, then the researcher should employ the design with the larger spread.

For $\widehat{\beta}_0$:

(1) The $\overline{x}^2$ factor is easier to interpret if we rewrite it as $\overline{x}^2 = (\overline{x} - 0)^2$, or the squared distance between the mean and the y-axis. If this is not immediately apparent, consider the two cases in Figure 7. Intuitively, if $\overline{x}$ is close to the y-axis, the location of y-intercept is fairly determined; if $\overline{x}$ is far, however, even a small change in the slope could result in a decently large change in the y-intercept, so there is a larger variability in the possible values of $\widehat{\beta}_0$.

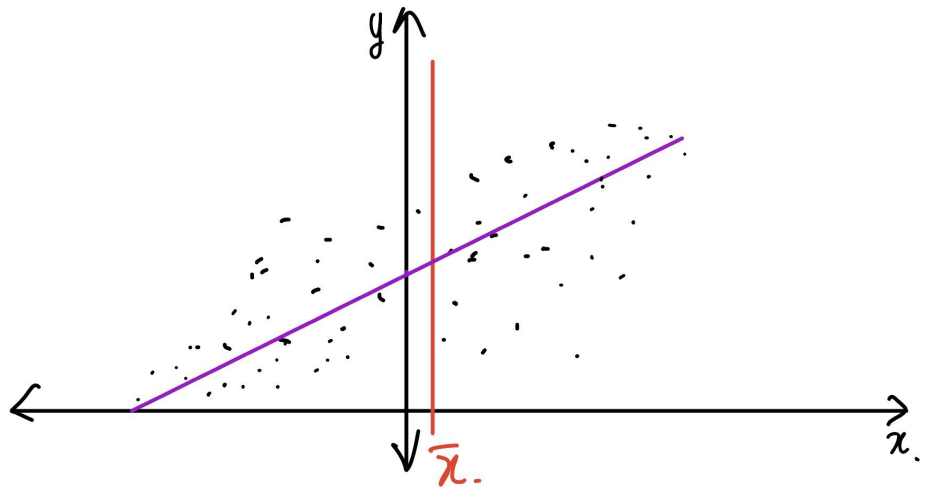
Lastly, we calculate the covariance.

Theorem 3.13. *The covariance of $\widehat{\beta}_1$ and $\widehat{\beta}_0$ is*

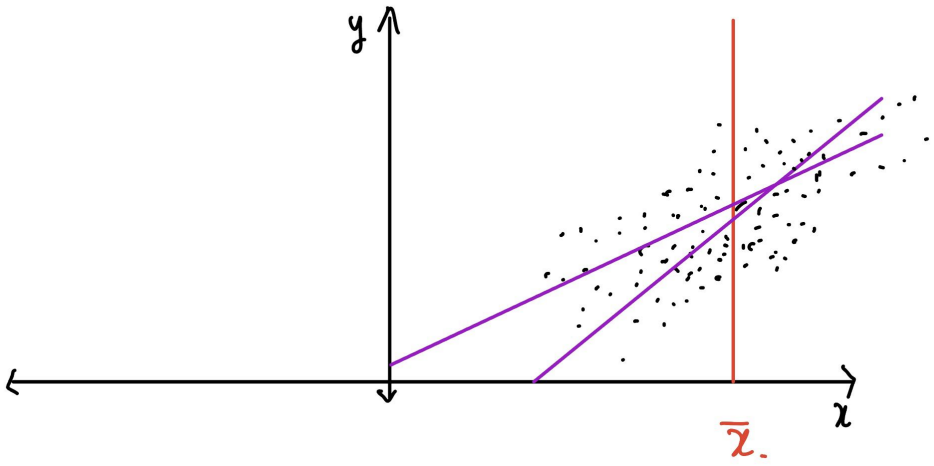
$$Cov\left(\widehat{\beta}_0, \widehat{\beta}_1\right) = -\frac{\sigma^2 \overline{x}}{nV}.$$

Proof. The result follows directly from previous calculations and bilinearity of covariance:

$$\text{Cov}\left(\widehat{\beta}_0, \widehat{\beta}_1\right) = \text{Cov}\left(\overline{y} - \widehat{\beta}_1 \overline{x}, \widehat{\beta}_1\right)$$



(A) Case 1: Small $\overline{x}$



(B) Case 2: Large $\overline{x}$

FIGURE 7. Effect of Mean of x_i on Variance of $\widehat{\beta}_0$

$$
\begin{aligned}
&= \operatorname{Cov}\left(\overline{y}, \widehat{\beta}_1\right) - \operatorname{Cov}\left(\widehat{\beta}_1 \overline{x}, \widehat{\beta}_1\right) \\
&= 0 - \overline{x} \cdot \mathbb{V}\left[\widehat{\beta}_1\right]
\end{aligned}
$$

$$= -\overline{x} \cdot \frac{\sigma^2}{nV} = -\frac{\sigma^2 \overline{x}}{nV}.$$

□

With these, we claim we have the joint distribution of the MLE estimators. Why? Well, first, we know their joint distribution is bivariate because with x_i's fixed, the two estimators end up both being functions of y_i's, which means they are linear functions of normal distributions and thus must also be normal (examine the expressions in Theorem 3.8 and Remarks 3.9). Then the expectation vector and the covariance matrix follow directly from Theorems 3.10, 3.11, 3.12 and 3.13. We also get some other results, which we will state here without proof because they turn out to be special cases of a more general theorem that we will cover later in the multivariate set-up.

Theorem 3.14.

(1) The joint distribution of the MLE is

$$\begin{pmatrix} \widehat{\beta}_1 \\ \widehat{\beta}_0 \end{pmatrix} \sim N\left(\begin{pmatrix} \beta_1 \\ \beta_0 \end{pmatrix}, \begin{pmatrix} \frac{\sigma^2}{nV} & -\frac{\sigma^2 \overline{x}}{nV} \\ -\frac{\sigma^2 \overline{x}}{nV} & \frac{\sigma^2}{n}\left(1 + \frac{\overline{x}^2}{V}\right) \end{pmatrix} \right).$$

(2) The standardized residual distribution follows a chi-squared distribution with dof $n-2$, i.e.

$$\frac{\sum_{i=1}^n (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2}{\sigma^2} \sim \chi^2_{n-2}.$$

(Note that we lose two dofs here and not one because there are two parameters being estimated.)

(3) The MLE estimators and the residuals are independent, i.e.

$$\begin{pmatrix} \widehat{\beta}_1 \\ \widehat{\beta}_0 \end{pmatrix} \perp\!\!\!\perp \sum_{i=1}^n (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2.$$

(4) All kinds of t-statistics, i.e. we can freely combine results from (1), (2), and (3) to design t-tests with different dofs and find their corresponding quantiles.

In particular, (4) would be especially useful when we construct all kinds of CI and HTs for the resulting estimators. Since σ^2 is not assumed to be known, most of the time we cannot directly use the normal distribution and have to rely on (2), (3), and (4) to cancel out σ^2 and estimate the variance using the sample. With this in mind, let us see two applications.

Example 3.15. (Univariate Model Selection)
Suppose we are given a HT with the hypotheses $H_0 : \beta_1 = 0$, $H_1 : \beta_1 \neq 0$. This is the HT used for *model selection*, where the null dictates a constant model $y_1, ..., y_m \overset{\text{iid}}{\sim} N(\beta_0, \sigma^2)$ and the alternative suggests a linear regression model where $y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$ independently. Then by Theorem 3.14, we can use an appropriate t-distribution to build a test-statistic and RR for β_1.

First, by results above, we have

$$\widehat{\beta}_1 - \beta_1 \sim N\left(0, \frac{\sigma^2}{nV}\right)$$

or

$$\widehat{\beta}_1 \cdot \frac{\sqrt{nV}}{\sigma} \overset{H_0}{\sim} N(0,1).$$

Then by (2) and (3) from Theorem 3.14,

$$\frac{\widehat{\beta}_1 \cdot \frac{\sqrt{nV}}{\sigma}}{\sqrt{\frac{1}{n-2}\frac{\sum_{i=1}^n \left(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i\right)^2}{\sigma^2}}} = \frac{\widehat{\beta}_1 \sqrt{nV}}{\sqrt{\frac{1}{n-2}\sum_{i=1}^n \left(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i\right)^2}} \overset{H_0}{\sim} t_{n-2}.$$

Thus we can simply take the above as the test-statistic T and reject when

$$|T| > t_{n-2,\, 1-\frac{\alpha}{2}}.$$

Example 3.16. (Univariate Prediction)
Suppose we are given new data point x^* and we want to predict the corresponding response variable. Technically we want $\beta_0 + \beta_1 x^*$, which we can estimate by $\widehat{\beta}_0 + \widehat{\beta}_1 x^*$. To say anything meaningful about this estimator, we need, of course, its expectation and variance:

$$\mathbb{E}\left[\widehat{\beta}_0 + \widehat{\beta}_1 x^*\right] = \beta_0 + \beta_1 x^*,$$

so the estimator is unbiased. Next,

$$\begin{aligned}
\mathbb{V}\left[\widehat{\beta}_0 + \widehat{\beta}_1 x^*\right] &= \mathbb{V}\left[\widehat{\beta}_0\right] + (x^*)^2\, \mathbb{V}\left[\widehat{\beta}_1\right] + 2x^* \mathbb{C}\left[\widehat{\beta}_0, \widehat{\beta}_1\right] \\
&= \frac{\sigma^2}{n}\left(1 + \frac{\overline{x}^2}{V} + \frac{(x^*)^2}{V} - \frac{2x^*\overline{x}}{V}\right) \\
&= \frac{\sigma^2}{n}\left[1 + \frac{(x^* - \overline{x})^2}{V}\right],
\end{aligned}$$

and therefore the accuracy of the estimator depends on the (squared) distance between the average observation and the new data point. Combined, we have

$$\widehat{\beta}_0 + \widehat{\beta}_1 x^* \sim N\left(\beta_0 + \beta_1 x^*,\, \frac{\sigma^2}{n}\left[1 + \frac{(x^* - \overline{x})^2}{V}\right]\right).$$

Moving things around gives

$$\frac{\widehat{\beta}_0 + \widehat{\beta}_1 x^* - \beta_0 - \beta_1 x^*}{\sqrt{\frac{\sigma^2}{n}\left[1 + \frac{(x^* - \overline{x})^2}{V}\right]}} \sim N(0,1).$$

Let us now construct a CI for our predicted value. Given the above is again a standard normal RV, we apply (2), (3), (4) of Theorem 3.14 to get

$$\frac{\frac{\widehat{\beta}_0+\widehat{\beta}_1 x^*-\beta_0-\beta_1 x^*}{\sqrt{\frac{1}{n}\left[1+\frac{(x^*-\overline{x})^2}{V}\right]}}}{\sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\left(y_i-\widehat{\beta}_0-\widehat{\beta}_1 x_i\right)^2}} \overset{H_0}{\sim} t_{n-2}.$$

It follows that the CI is bounded by

$$\widehat{\beta}_0+\widehat{\beta}_1 x^* \pm t_{n-2,\,1-\frac{\alpha}{2}} \cdot \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\left(y_i-\widehat{\beta}_0-\widehat{\beta}_1 x_i\right)^2} \cdot \sqrt{\frac{1}{n}\left[1+\frac{(x^*-\overline{x})^2}{V}\right]}.$$

In particular, this means that the length of CI will increase as x^* deviates further from $\overline{x}$. A typical CI graph for linear regression would typically look like Figure 8.

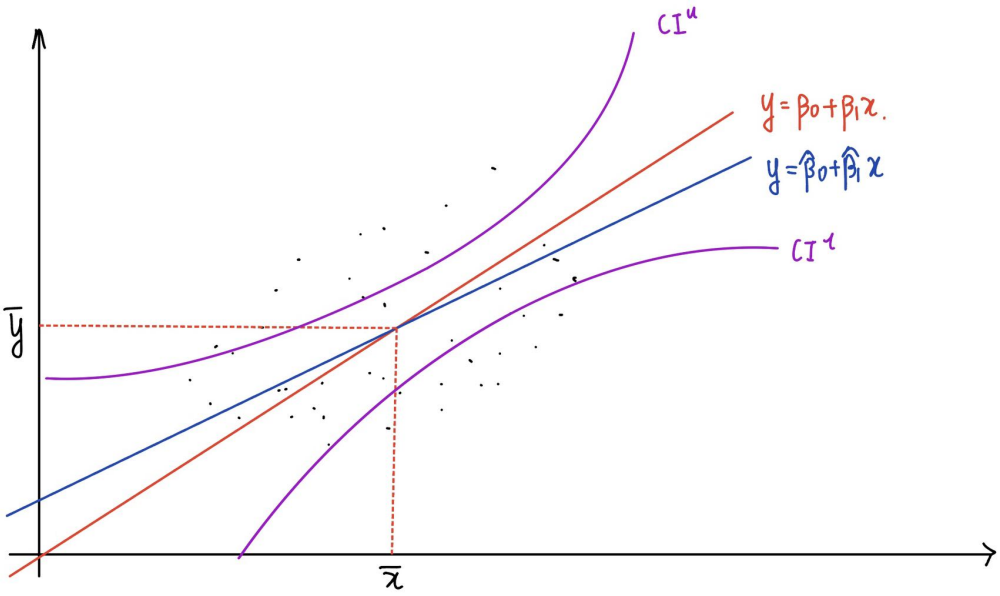


FIGURE 8. CI for Univariate Linear Regression

I cannot resist inserting an example done on real data from my metrics class (see Figure 9). The mean age $\overline{x}$ is around 43. Here the line is curved because the linear regression was done with respect to a quadratic function of age, but it did not change the trait of larger CI towards the extreme ends.

Now we move on to the multivariate case.

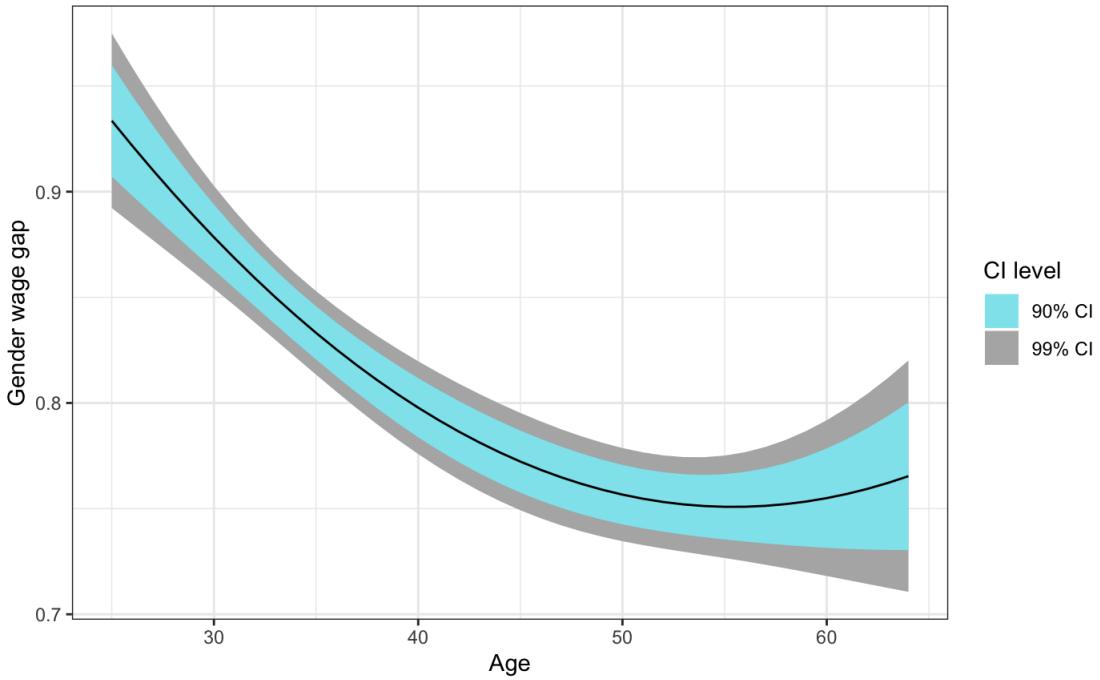


FIGURE 9. CI for Linear Regression with CPS Data

3.2. **Multivariate.**

Even though we are working with multiple dimensions in this section, the results will actually be a lot clearer than the univariate case thanks to linear algebra. Let us start by considering

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{p-1} x_{ip-1} + \sigma z_i$$

for $i = 1, ..., n$, $z_i \overset{\text{iid}}{\sim} N(0,1)$. This is a system of n linear equations and thus can be rewritten in matrix form:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p-1} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p-1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np-1} \end{pmatrix}}_{\text{design matrix}} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{pmatrix} + \sigma \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}. \tag{3.17}$$

In particular, the design matrix (it is called the design matrix because in fixed design, the entry values are determined prior to data collection) is of dimension $n \times p$. Usually $n >> p$, so it is **not** a square matrix, despite looking very much like one here. This fact will be important soon.

We can also summarize (3.17) as

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta} + \sigma\boldsymbol{z} \sim N(\boldsymbol{x}\boldsymbol{\beta}, \sigma^2 I_n). \tag{3.18}$$

To find an appropriate estimator, we start with the MLE, and we observe, yet again,

$$p(\boldsymbol{y}) = \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2\right).$$

Thus

$$\max_{\boldsymbol{\beta}} p(\boldsymbol{y}) = \min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2,$$

and the MLE is the LSE (least-squares estimator).

Now we proceed to find $\widehat{\boldsymbol{\beta}}$ in the following way: we first posit a reasonable guess $\widehat{\boldsymbol{\beta}}_{\text{guess}} = \widehat{\boldsymbol{\beta}}_g$ and then show it is correct. For our guess, we look to the simplest possible case: the special case with no noise $\sigma^2 = 0$. Then we just have $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta}$, so taking $\widehat{\boldsymbol{\beta}}_g = \boldsymbol{x}^{-1}\boldsymbol{y}$ seems like a decent guess.

But wait! This operation is completely illegal! Remember that the design matrix $\boldsymbol{x}$ is not a square matrix, so we cannot just apply the inverse to it. That problem can be resolved if we multiply the transpose in front:

$$\boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{x}\widehat{\boldsymbol{\beta}}_g,$$

and we get

$$\widehat{\boldsymbol{\beta}}_g = \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\left(\boldsymbol{x}^T\boldsymbol{y}\right).$$

Note that this operation would still require $\boldsymbol{x}$ to have full rank, otherwise there is perfect multicollinearity, i.e., there exists an infinite number of minimizers, and $\widehat{\boldsymbol{\beta}}$ would not be well-defined. Thus we take for granted for $\boldsymbol{x}$ has full rank.

Next, we want to show that for any $\beta \in \mathbb{R}^p$,

$$\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 \geq \|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}_g\|^2.$$

We need a little tool from linear algebra.

Lemma 3.19. *The norm of the sum of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ can be decomposed as*

$$\|\boldsymbol{a} + \boldsymbol{b}\|^2 = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 + 2\langle a, b\rangle = \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2 + 2\boldsymbol{a}^T\boldsymbol{b}.$$

With the help of the lemma, we can rewrite

$$\begin{aligned}\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 &= \|\left(\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}_g\right) + \left(\boldsymbol{x}\widehat{\boldsymbol{\beta}}_g - \boldsymbol{x}\boldsymbol{\beta}\right)\|^2 \\ &= \underbrace{\|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}_g\|^2}_{(1)} + \underbrace{\|\boldsymbol{x}\widehat{\boldsymbol{\beta}}_g - \boldsymbol{x}\boldsymbol{\beta}\|^2}_{(2)} + \underbrace{2\left(\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}_g\right)^T\left(\boldsymbol{x}\widehat{\boldsymbol{\beta}}_g - \boldsymbol{x}\boldsymbol{\beta}\right)}_{(3)}.\end{aligned}$$

Now we claim that (3) vanishes. Why? Because

$$(3) = 2\left(\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}_g\right)^T\boldsymbol{x}\left(\widehat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}\right)$$

$$
\begin{aligned}
&= 2\left(\boldsymbol{y} - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\left(\boldsymbol{x}^T\boldsymbol{y}\right)\right)^T \boldsymbol{x}\left(\widehat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}\right) \\
&= 2\left[\left(I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\boldsymbol{y}\right]^T \boldsymbol{x}\left(\widehat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}\right) \\
&= 2\boldsymbol{y}^T\left(I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\boldsymbol{x}\left(\widehat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}\right) \\
&= 2\boldsymbol{y}^T \underbrace{\left(\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{x}\right)}_{=\mathbf{0}_n}\left(\widehat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}\right) = 0.
\end{aligned}
$$

Since $(2) \geq 0$, we conclude our guess is correct, and

$$
\widehat{\boldsymbol{\beta}}_{\text{LSE}} = \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\left(\boldsymbol{x}^T\boldsymbol{y}\right). \tag{3.20}
$$

Another thing to note is that without (3), the rest of the terms above looks like Pythagorean theorem, and they should! As a preview of Section 3.3, the matrix

$$
\boldsymbol{p}_{\boldsymbol{x}} = \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T \tag{3.21}
$$

is called the *projection matrix*, and it projects $\boldsymbol{y}$ onto the column space of $\boldsymbol{x}$ (the linear subspace of dimension p spanned by $\boldsymbol{x}$) (see Figure 10).

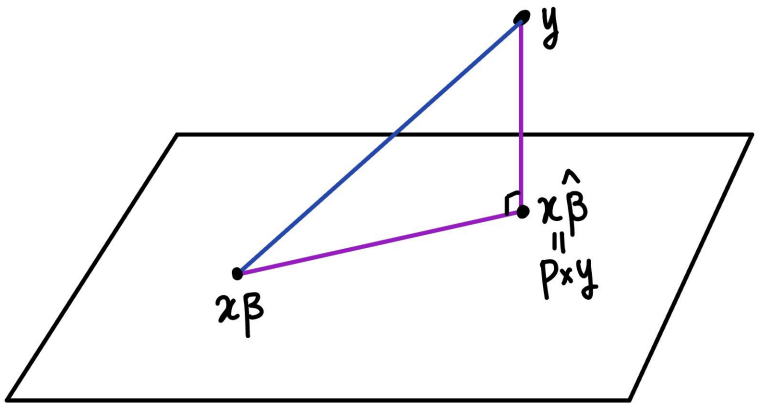


FIGURE 10. Projection onto 2-Dimensional Plane

This should make intuitive sense: to minimize the distance between the predicted and the actual values, the vector connecting them should be orthogonal to the column space, which is exactly the Pythagorean theorem.

As an aside, suppose Z is a RV and a any constant. Recall the variance-bias decomposition of MSE:

$$\mathbb{E}\left[(Z-a)^2\right] = \mathbb{V}[Z]+\mathbb{E}[(Z-a)]^2 = \mathbb{E}\left[(Z-\mathbb{E}[Z])^2\right]+(\mathbb{E}[Z]-a)^2 = \mathbb{E}\left[(Z-\mathbb{E}[Z])^2\right]+\mathbb{E}\left[(\mathbb{E}[Z]-a)^2\right],$$

which is also exactly the Pythagorean theorem! According to the result above, we can interpret

$$\mathbb{E}[Z] = \operatorname*{argmin}_a \mathbb{E}\left[(Z-a)^2\right],$$

which is why regressing a RV on a constant exactly recovers the mean.

Now we calculate some properties of LSE:

$$\mathbb{E}\left[\widehat{\boldsymbol{\beta}}\right] = \mathbb{E}\left[\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\left(\boldsymbol{x}^T\boldsymbol{y}\right)\right] = \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\mathbb{E}[\boldsymbol{y}] = \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{\beta} = \boldsymbol{\beta},$$

and the estimator is unbiased. For the variance, we calculate the covariance matrix:

$$\begin{aligned}\operatorname{Cov}\left[\widehat{\boldsymbol{\beta}}\right] &= \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\mathbb{C}[\boldsymbol{y}]\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\\ &= \sigma^2\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T I_n\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\\ &= \sigma^2\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}.\end{aligned}$$

Combined, we have

$$\widehat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta},\, \sigma^2\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\right). \tag{3.22}$$

Theorem 3.23. *We basically rewrite Theorem 3.14 with linear algebra notation:*

(1)

$$\widehat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta},\, \sigma^2\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\right).$$

(2)

$$\frac{\|\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2}{\sigma^2} \sim \chi^2_{n-p}.$$

(3)

$$\widehat{\boldsymbol{\beta}} \perp\!\!\!\perp \|\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2.$$

(4) All kinds of t-statistics.

Proof. It remains to show (2) and (3). We will postpone the proof for (2) to the next section as it requires new tools, so we focus on (3). First, we know the two quantities in question are jointly normal as linear transformations of $\boldsymbol{y}$:

$$\begin{pmatrix}\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\beta}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{y}-\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{y}\\ \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{y}\end{pmatrix} = \begin{pmatrix}\left(I_n-\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\\ \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\end{pmatrix}\boldsymbol{y}.$$

This is just a matrix transformation of $\boldsymbol{y}$, and thus the two quantities have to be jointly normal. This means, by Theorem 1.25, we only have to show that the covariance is 0:

$$\operatorname{Cov}(\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}}) = \operatorname{Cov}\left(\boldsymbol{y}-\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{y}, \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{y}\right)$$

$$
\begin{aligned}
&= \operatorname{Cov}\left(\left(I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\boldsymbol{y}, \left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{y}\right) \\
&= \left(I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\operatorname{Cov}(\boldsymbol{y},\boldsymbol{y})\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1} \\
&= \sigma^2 \underbrace{\left(\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{x}\right)}_{\mathbf{0}_{n\times p}}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1} = \mathbf{0}_{n\times p}.
\end{aligned}
$$

□

3.3. **Projection Matrix.**

We have mentioned, in the passing, the projection matrix for our linear regression (see (3.21)). This is only one example out of the class of projection matrices, all of which satisfy certain criteria. Now it is time to introduce the notion of projection more formally.

Definition 3.24. A matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is a *projection matrix* if it is

(1) symmetric, i.e. $\boldsymbol{P} = \boldsymbol{P}^T$,
(2) idempotent, i.e. $\boldsymbol{P}^2 = \boldsymbol{P}$.

A direct result is that projective matrices must be square matrices. Let's see some examples.

Examples 3.25. **(1)** Let $\boldsymbol{u}$ be any vector of unit length, i.e.

$$\boldsymbol{u} = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix}^T \in \mathbb{R}^{n\times 1}$$

such that

$$\|u\| = \sqrt{\sum_{i=1}^{n} u_i^2} = 1.$$

Then $\boldsymbol{u}\boldsymbol{u}^T$ is a $n \times n$-dimensional matrix. We can verify that this is a valid projection matrix:

$$\left(\boldsymbol{u}\boldsymbol{u}^T\right)^T = \left(\boldsymbol{u}^T\right)^T\boldsymbol{u}^T = \boldsymbol{u}\boldsymbol{u}^T,$$

and

$$\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{u}\boldsymbol{u}^T = \boldsymbol{u}\left(\boldsymbol{u}^T\boldsymbol{u}\right)\boldsymbol{u}^T = \boldsymbol{u}\|\boldsymbol{u}\|^2\boldsymbol{u}^T = \boldsymbol{u}\boldsymbol{u}^T.$$

Projection of another $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ onto $\boldsymbol{u}$ would look something like in Figure 11.

Here, the projection coordinate is

$$\boldsymbol{P}_{\boldsymbol{u}}(\boldsymbol{y}) = \left(\boldsymbol{u}\boldsymbol{u}^T\right)\boldsymbol{y} = \langle \boldsymbol{y}, \boldsymbol{u}\rangle\boldsymbol{u} = \frac{\langle \boldsymbol{y}, \boldsymbol{u}\rangle}{\langle \boldsymbol{u}, \boldsymbol{u}\rangle}\boldsymbol{u},$$

since $\boldsymbol{u}$ has unit length.

(2) Let us define $\mathbb{1}_n$ to be the column vector of 1's. Then we take $\boldsymbol{u}$ to be

$$\boldsymbol{u} = \frac{1}{\sqrt{n}}\mathbb{1}_n = \begin{pmatrix} \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}} & \cdots & \frac{1}{\sqrt{n}} \end{pmatrix}^T.$$

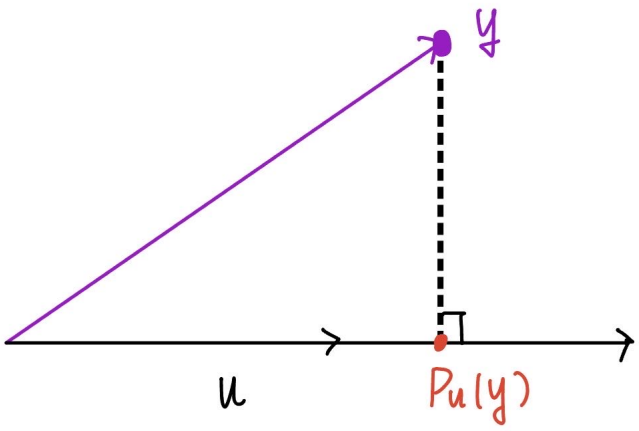


FIGURE 11. Projection onto 1-Dimensional Subspace

This is just a special case of (1), since

$$\|\boldsymbol{u}\|^2 = \sum_{i=1}^{n} u_i^2 = \sum_{i=1}^{n} \frac{1}{n} = 1.$$

So

$$\boldsymbol{P}_{\mathbb{1}} = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$$

is a projection matrix, and a very useful one at that. Observe that

$$P_{\mathbb{1}}(\boldsymbol{y}) = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\boldsymbol{y} = \frac{1}{n}\mathbb{1}_n\sum_{i=1}^{n} y_i = \overline{y}\mathbb{1}_n,$$

which is just the constant vector of $\overline{y}$. So $\boldsymbol{P}_{\mathbb{1}}$ is the projection matrix that produces the mean.

(3) Following the work we did in the previous section, for any full rank $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, the matrix $\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is a projection. We can check symmetry

$$\left(\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\right)^T = \left(\boldsymbol{X}^T\right)^T\left((\boldsymbol{X}^T\boldsymbol{X})^{-1}\right)^T\boldsymbol{X}^T = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$$

and idempotency

$$\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T.$$

(4) Per (3), the identity matrix I_n is also a projection matrix. We can interpret this projection as follows: I_n projects vectors just like any other projection matrices, but since the subspace spanned happens to be the entire space, we observe no changes as a result of the projection.

Why are we covering projection matrices? It turns out symmetry and idempotency combined can bring forth very nice properties.

Theorem 3.26. *(Properties of Projection Matrices)*
Let $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ be a projection matrix.

(1) Its orthogonal complement $I_n - \boldsymbol{P}$ is also a projection matrix.
(2) $\boldsymbol{P}(I_n - \boldsymbol{P}) = \boldsymbol{0}_{n\times n}$, i.e. $\boldsymbol{P}$ and $(I_n - \boldsymbol{P})$ are orthogonal, and projection onto either leaves nothing for the complement.
(3) Let $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ be another projection matrix such that $\boldsymbol{PQ} = \boldsymbol{0}$ (i.e. the projection subspaces are orthogonal). Then $\boldsymbol{P} + \boldsymbol{Q}$ is another projection matrix.
(4) The eigenvalues of $\boldsymbol{P}$ are either 0 or 1.

Before we start the proof, a few remarks: first, the notion of orthogonal complement and Property (2) should not be unfamiliar: it has already appeared once in the proof for Theorem 3.23 Property 3, when we claimed that

$$(I_n - \boldsymbol{p_x})\boldsymbol{x} = \left(I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\right)\boldsymbol{x} = \left(\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T\boldsymbol{x}\right) = \boldsymbol{0}_{n\times p}.$$

There the matrix

$$I_n - \boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T$$

is exactly the orthogonal complement of $\boldsymbol{p_x}$.

Secondly, (2) gives a trivial example of (3). Examine $\boldsymbol{P}$ and $\boldsymbol{Q} = I_n - \boldsymbol{P}$. Then the combined subspace span the entire $\mathbb{R}^{n\times n}$, which correspond to the identity matrix.

Proof. We will proceed in order. Property (1) follows directly from idempotency:

$$(I_n - \boldsymbol{P})(I_n - \boldsymbol{P}) = I_n - 2\boldsymbol{P} + \boldsymbol{P}^2 = I_n - 2\boldsymbol{P} + \boldsymbol{P} = I_n - \boldsymbol{P}.$$

Property (2) follows similarly:

$$\boldsymbol{P}(I_n - \boldsymbol{P}) = \boldsymbol{P} - \boldsymbol{P}^2 = \boldsymbol{P} - \boldsymbol{P} = \boldsymbol{0}_{n\times n}.$$

Property (3) needs a bit more maneuvering. First, note that

$$(\boldsymbol{PQ})^T = \boldsymbol{Q}^T\boldsymbol{P}^T = \boldsymbol{QP} = 0.$$

Then the symmetry of the sum follows directly from the symmetry of each:

$$(\boldsymbol{P} + \boldsymbol{Q})(\boldsymbol{P} + \boldsymbol{Q}) = \boldsymbol{P}^2 + \boldsymbol{PQ} + \boldsymbol{QP} + \boldsymbol{Q}^2 = \boldsymbol{P} + \boldsymbol{Q}.$$

Lastly, for Property (4), real symmetric matrices are always diagonalizable, so we know we can decompose $\boldsymbol{P}$ into $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U}$ is full rank and satisfies

$$\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = I_n$$

(i.e. the n rows of $\boldsymbol{U}$ are orthogonal to each other and the n columns of $\boldsymbol{U}$ are orthogonal, and for each row and column of the inner product, the $\mathcal{L}^2$ norm is exactly 1) and

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

is a diagonal matrix. Idempotency means

$$\boldsymbol{U\Lambda U}^T\boldsymbol{U\Lambda U}^T = \boldsymbol{U\Lambda}^2\boldsymbol{U}^T = \boldsymbol{U\Lambda U}^T \iff \boldsymbol{UU}^T\boldsymbol{\Lambda}^2\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{UU}^T\boldsymbol{\Lambda U}^T\boldsymbol{U} \iff \boldsymbol{\Lambda}^2 = \boldsymbol{\Lambda}.$$

Since $\boldsymbol{\Lambda}$ is diagonal, this happens if and only if $\lambda_i^2 = \lambda_i$ for all i, so it must be the case that $\lambda_i = 0$ or $\lambda_i = 1$ for all i. □

The eigenvalue brings forth another interpretation of projection matrices: given $\boldsymbol{y}$, the eigenvectors associated with the projection first rotate it to some proper basis, and then keep or delete each component based on whether the eigenvalue is 0 or 1.

Let us go further. We can always write

$$\boldsymbol{P} = \boldsymbol{U}\begin{pmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{pmatrix}\boldsymbol{U}^T = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n \end{pmatrix}\begin{pmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{pmatrix}\begin{pmatrix} \boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \vdots \\ \boldsymbol{u}_n^T \end{pmatrix},$$

where $\boldsymbol{u}_i$ is the ith column of $\boldsymbol{U}$. This is always obtainable through rowwise operations. Here r as the number of non-zero eigenvalues is simply the rank of $\boldsymbol{P}$. But then the above can be further rewritten as

$$\boldsymbol{P} = \boldsymbol{u}_1\boldsymbol{u}_1^T + \boldsymbol{u}_2\boldsymbol{u}_2^T + \cdots + \boldsymbol{u}_r\boldsymbol{u}_r^T.$$

We have recognized previously that the norm of each component is 1, so $\boldsymbol{u}_i\boldsymbol{u}_i^T$ are all projection matrices with the subspaces orthogonal to each other (since the eigenvectors $\boldsymbol{u}_i$'s are orthogonal to each other, see here for proof). Thus we can always decompose $\boldsymbol{P}$ into r unit projections corresponding to the basis given by $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, ..., $\boldsymbol{u}_r$. Note that if all eigenvalues are 1, we recover

$$\boldsymbol{u}_1\boldsymbol{u}_1^T + \boldsymbol{u}_2\boldsymbol{u}_2^T + \cdots + \boldsymbol{u}_n\boldsymbol{u}_n^T = \boldsymbol{UU}^T = I_n,$$

which aligns with previous discussions.

Graphs are always good for intuition, so here is one. Consider the very innocent cuboid in Figure 12. Examine $\boldsymbol{v}$ defined by the vector (v_1, v_2, v_3), and consider projection onto standard basis e_1, e_2, e_3.

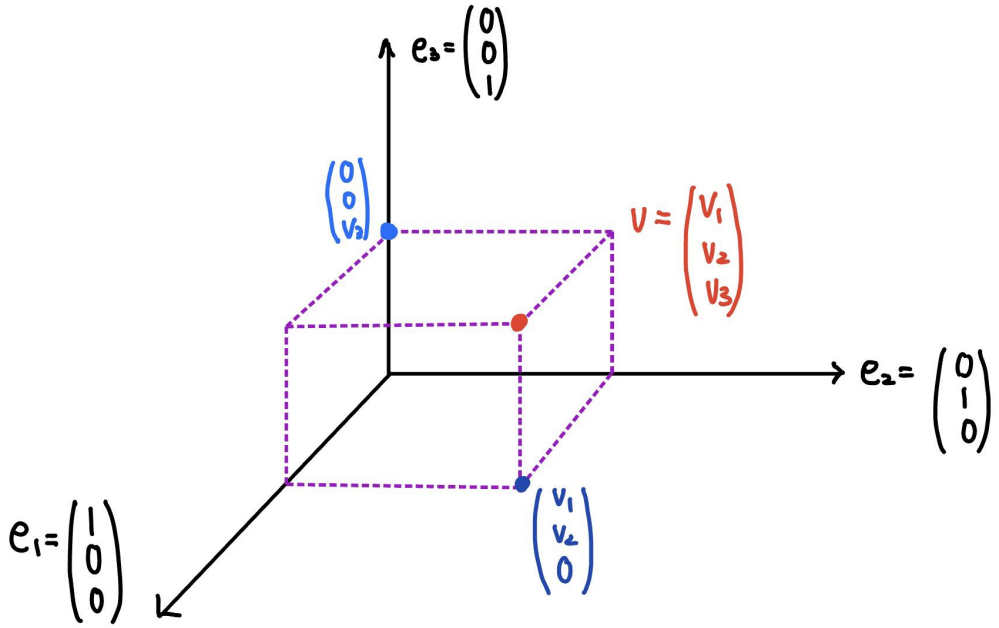


FIGURE 12. 3-Diemnsional Projection onto Standard Basis

Then we can interpret $(0, 0, v_3)$ as the projection of $\boldsymbol{v}$ onto the subspace spanned by e_3:

$$e_3 e_3^T \boldsymbol{v} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v_3 \end{pmatrix},$$

and $(v_1, v_2, 0)$ as the projection of v onto the subspace spanned by e_1 and e_2:

$$\left(e_1 e_1^T + e_2 e_2^T\right) \boldsymbol{v} = \begin{pmatrix} v_1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ v_2 \\ 0 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ 0 \end{pmatrix}.$$

It should be evident that the two subspaces are orthogonal complements of each other, and in alignment with what we discussed above, we recover

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ v_3 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

The concept of projection, however, is useful exactly because it applies to more complicated situations. Let us now apply projection to $\boldsymbol{v}$ onto a non-standard basis.

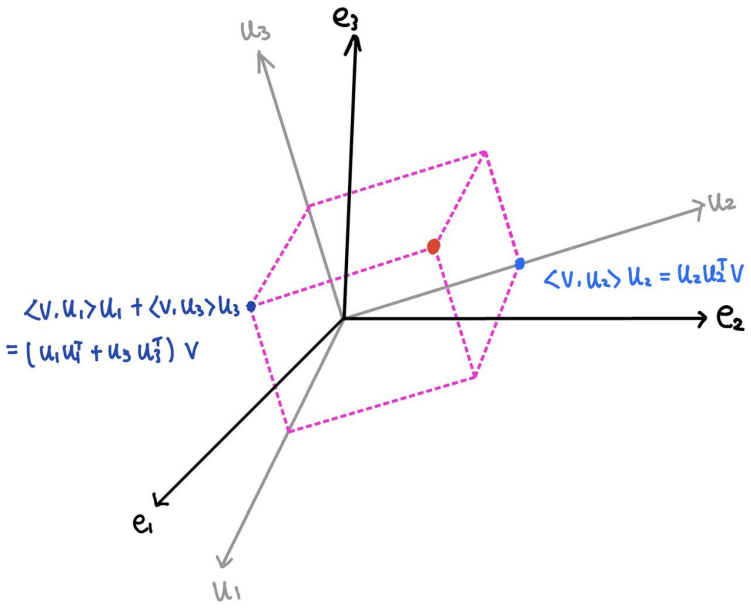


FIGURE 13. 3D Projection onto Rotated Basis

In Figure 13, we are working with the basis defined by the unit vectors $\boldsymbol{u}_1, \boldsymbol{u}_2$ and $\boldsymbol{u}_3$. The exact same ideas apply: if we combine the projection of $\boldsymbol{v}$ onto the subspace spanned by u_2 and the subspace spanned by $\boldsymbol{u}_1$ and $\boldsymbol{u}_3$, we still recover

$$\boldsymbol{v} = \left(\boldsymbol{u}_1\boldsymbol{u}_1^T + \boldsymbol{u}_2\boldsymbol{u}_2^T + \boldsymbol{u}_3\boldsymbol{u}_3^T\right)\boldsymbol{v}$$

due to the orthogonality of the subspaces.

Now we introduce another nice property of projection matrices. For this, we need the concept of trace.

Definition 3.27. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a square matrix. Then the *trace* of $\boldsymbol{A}$ is the sum of all of its diagonal entries, i.e.

$$\mathrm{Tr}(\boldsymbol{A}) = \sum_{i=1}^{n} A_{ii}.$$

Theorem 3.28. *(Properties of Trace)*

(1) Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$. Then $Tr(A+B) = Tr(A) + Tr(B)$.
(2) Let $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, $\boldsymbol{B} \in \mathbb{R}^{m\times n}$. Then $Tr(AB) = Tr(BA)$.

Proof. The first is left to readers as an exercise. We will focus on the second property. Suppose

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix}.$$

Direct observation gives

$$\mathrm{Tr}(AB) = \sum_{j=1}^{n}\sum_{i=1}^{m} a_{ji}b_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}a_{ji} = \mathrm{Tr}(BA).$$

□

Theorem 3.29. *Let $\boldsymbol{P}$ be a projection matrix. Then*

$$\mathit{rank}(\boldsymbol{P}) = \mathit{Tr}(\boldsymbol{P}) = \sum_{j=1}^{n} P_{ii}.$$

This implies that rank, which is usually a non-linear function of matrices, is linear for projection matrices.

Proof. Suppose $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. Thus by Property (2) of Theorem 3.28,

$$\begin{aligned} \mathrm{Tr}(\boldsymbol{P}) &= \mathrm{Tr}(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T) = \mathrm{Tr}(\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{U}) = \mathrm{Tr}(\boldsymbol{\Lambda}) \\ &= \mathrm{Tr}\left(\begin{pmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{pmatrix}\right) \\ &= \#\text{ of non-zero eigenvalues} = \mathrm{rank}(\boldsymbol{P}). \end{aligned}$$

□

Now we are ready to start proving Property (2) of Theorem 3.23. We need the following lemma.

Lemma 3.30. *Assume $Z \sim N(0, I_n)$ and let $\boldsymbol{P}$ be a projection matrix with $\mathit{rank}(\boldsymbol{P}) = r$. Then $\|\boldsymbol{P}\boldsymbol{Z}\|^2 \sim \chi^2_r$.*

To get some intuition as to what this lemma is saying, first consider the case $n = 2$, or the bivariate normal case

$$\boldsymbol{Z} = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim N(0, I_2).$$

When the components are uncorrelated, the density is perfectly symmetric about 0.

(The graph is taken from this website. I strongly encourage the reader to read the whole thing, as it furnishes some very nice graphics which builds intuition for MVN.)

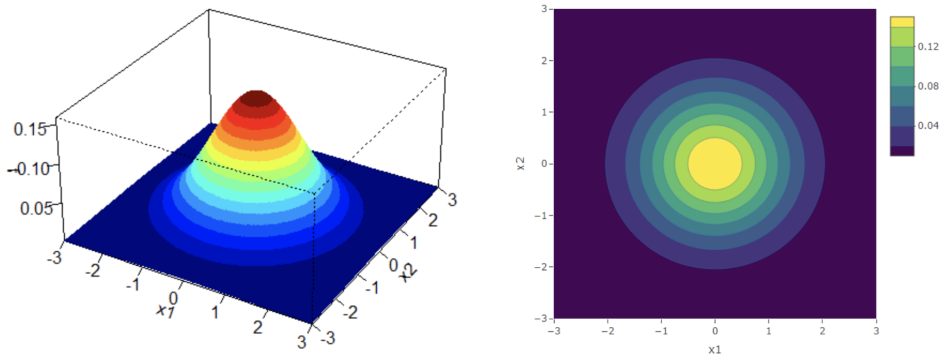


FIGURE 14. Contour Density of Bivariate Normal Distribution

Let us first examine the density w.r.t. the standard basis. We know

$$e_1 e_1^T \boldsymbol{Z} = \begin{pmatrix} Z_1 \\ 0 \end{pmatrix},$$

which means

$$\|e_1 e_1^T \boldsymbol{Z}\|^2 \sim \chi_1^2.$$

However, this is not unique to the standard basis. If we take any other set of unit vectors u_1 and u_2 such that they span $\mathbb{R}^2$ together, we also have

$$\|u_1 u_1^T \boldsymbol{Z}\|^2 \sim \chi_1^2.$$

If we expand this scenario to higher dimensions and project $\boldsymbol{Z}$ onto a r-dimensional subspace, we can simply take the sum over the coordinates to have a χ_r^2-distribution by Property (3) of Theorem 3.26.

Proof. Suppose

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \boldsymbol{U} \begin{pmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{pmatrix} \boldsymbol{U}^T.$$

Then $\|\boldsymbol{P}\boldsymbol{Z}\| = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{Z}$, and

$$\|\boldsymbol{P}\boldsymbol{Z}\|^2 = \boldsymbol{Z}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{Z} = \boldsymbol{Z}^T\boldsymbol{P}\boldsymbol{Z} = \boldsymbol{Z}^T\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{Z} = (\underbrace{\boldsymbol{U}^T\boldsymbol{Z}}_{\boldsymbol{W}})^T\boldsymbol{\Lambda}(\boldsymbol{U}^T\boldsymbol{Z}) = \boldsymbol{W}^T\boldsymbol{\Lambda}\boldsymbol{W}.$$

Now we examine

$$\boldsymbol{W} = \boldsymbol{U}^T\boldsymbol{Z} \sim N(0, \boldsymbol{U}^T\boldsymbol{U}) = N(0, I_n),$$

which is MVN! Hence we can write

$$\|\boldsymbol{P}\boldsymbol{Z}\|^2 = \boldsymbol{W}^T \begin{pmatrix} I_r & 0 \\ 0 & 0_{n-r} \end{pmatrix} \boldsymbol{W} = W_1^2 + \cdots + W_r^2 \sim \chi_r^2.$$

□

Now we are officially ready for Property (2) of Theorem 3.23.

Proof. (Property (2) of Theorem 3.23)
Recall the objective of proof

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2}{\sigma^2} \sim \chi^2_{n-p}.$$

By our set-up, $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta} + \sigma\boldsymbol{z}$, where $\boldsymbol{z} \sim N(0, I_n)$. We also know that

$$\begin{aligned}
\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}} &= \big(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}\big)\boldsymbol{y} \\
&= \big(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}\big)(\boldsymbol{x}\boldsymbol{\beta} + \sigma\boldsymbol{z}) \\
&= (\boldsymbol{x} - \boldsymbol{x})\boldsymbol{\beta} + \sigma\big(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}\big)\boldsymbol{z} \\
&= \sigma\big(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}\big)\boldsymbol{z}.
\end{aligned}$$

Therefore,

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2}{\sigma^2} = \frac{\sigma^2\|(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x})\boldsymbol{z}\|^2}{\sigma^2} = \|\underbrace{\big(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}\big)}_{\boldsymbol{M}}\boldsymbol{z}\|^2,$$

and we know the RHS follows a $\chi^2_{\text{rank}(\boldsymbol{M})}$ distribution. It remains to compute

$$\begin{aligned}
\text{rank}(\boldsymbol{M}) &= \text{Tr}(I_n - \boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T) \\
&= \text{Tr}(I_n) - \text{Tr}(\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T) \\
&= \text{Tr}(I_n) - \text{Tr}((\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{x}) \\
&= \text{Tr}(I_n) - \text{Tr}(I_p) \quad \text{since } \boldsymbol{x} \text{ is full rank} \\
&= n - p.
\end{aligned}$$

Hence

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2}{\sigma^2} \sim \chi^2_{n-p}.$$

□

Lastly, for the sake of intellectual exercise, let us return to a theorem we have proven long ago, namely Property (2) of Theorem 1.29: if $X_1, ..., X_n \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$, then

$$\frac{\sum_{i=1}^n (X_i - \overline{X})^2}{\sigma^2} \sim \chi^2_{n-1}.$$

If you somehow miraculously recite this whole cheatbook, you will remember that our old method was a painful proof by induction which spanned over two pages. Let us substitute that painful

memory by a new proof using matrices which only takes a few lines. First, we recognize that

$$\sum_{i=1}^{n}(X_i-\overline{X})^2=\left\|\begin{pmatrix}(X_1-\overline{X})\\ \vdots \\ (X_n-\overline{X})\end{pmatrix}\right\|=\|X-\overline{X}\mathbb{1}_n\|^2=\left\|X-\left(\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)X\right\|=\left\|\left(I_n-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)X\right\|,$$

where $1/n\mathbb{1}_n\mathbb{1}_n^T$ is the rank-one projection. Next, since we can write $X=\mu\mathbb{1}_x+\sigma Z$,

$$\begin{aligned}\sum_{i=1}^{n}(X_i-\overline{X})^2&=\left\|\left(I_n-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)(\mu\mathbb{1}_n+\sigma Z)\right\|\\&=\left\|\sigma\left(I_n-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)Z\right\|\\&=\sigma^2\left\|\left(I_n-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)Z\right\|\sim\sigma^2\chi^2_{n-1},\end{aligned}$$

by Lemma 3.30, since $I_n-1/n\mathbb{1}_n\mathbb{1}_n^T$ has trace $n-1$.

3.4. **Model Selection.**

Let us return to the question of linear regressions. Recall that the goal is to use a set of data (x_i,y_i) to predict future observations, and in reality, the given $\{x_i\}$ may contain a lot more information than we 'need'. For example, if we are trying to predict wages, traits like height or weight should not add much to our predictive power, even though they may be in the data we collected. Then the question becomes: *what is the appropriate subset of variables to choose when running the regression*?

The first possibility to consider is, well, why don't we just use all the variables that are given? Even if height and weight is not associated with wage as much, adding them to the set certainly does not hurt, right? Well, the problem is, it does. We would find that, if we use all the variables, then

$$\min_{\widehat{\boldsymbol{\beta}}}\|\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}\|=0,$$

but the corresponding minimizer $\widehat{\boldsymbol{\beta}}$ has a definite form that depends on all the existing data, and it cannot be used to do prediction at all. It is known as the *binning estimator*

$$\widehat{\boldsymbol{\beta}}:=\frac{\sum_{i=1}^{n}y_i\mathbb{1}[x_i=x]}{\sum_{i=1}^{n}\mathbb{1}[x_i=x]},$$

and it stands at one of the extremes of the *bias-variance tradeoff*, risking very high variance (and sometimes not being well-defined) for zero bias. The phenomenon itself is called *overfitting*.

Even if we do not go as far as to include every single variable we have, including more variables than 'needed' would necessarily introduce more variance. Consider the following options in the

univariate set-up:

$$\begin{cases} \text{Model 1: } y_i \overset{\text{iid}}{\sim} N(\beta_0, \sigma^2), \\ \text{Model 2: } y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2), \beta_1 = 0. \end{cases}$$

Thus in Model 2 we are intentionally overfitting. Then under Model 1,

$$\mathbb{V}\left[\widehat{\beta}_0\right] = \frac{\sigma^2}{n}.$$

While under Model 2,

$$\mathbb{V}\left[\widehat{\beta}_0\right] = \frac{\sigma^2}{n}\left(1 + \frac{\overline{x}^2}{V}\right).$$

Hence a more noisy estimator is the cost of introducing more variables.

Another potential problem that may follow from including too many variables is the issue of *perfect multicollinearity*. Recall that the existence of a LSE estimator depends on the fact that the design matrix has full rank. Thus technicality requires us to exclude redundant variables that can be written as linear combinations of others. We will not go into details, but the most common scenario with perfect multicollinearity is we include a set of mutually exclusive, mutually exhaustive indicator variables and a constant.

However, as bad as overfitting sounds, *underfitting* is even worse. If we fail to include a variable that should have been included, we would have *omitted variable bias*, and the resulting estimator can be drastically biased.

To see if we have the appropriate set of variables then, we need HT. The univariate case is already covered in Example 3.15, which we encourage the readers to reread. Here we focus on the multivariate case. There are tests to evaluate the significance of single coefficients, but here we will only introduce the test for overall significance. Consider the hypotheses

$$\begin{cases} H_0: \ y_i \overset{\text{iid}}{\sim} N(\beta_0, \sigma^2), \\ H_1: \ y_i \sim N(\boldsymbol{x}\boldsymbol{\beta}, \sigma^2 I_n), \end{cases}$$

i.e. the null model versus the full model. An equivalent way to write this is

$$\begin{cases} H_0: \ \beta_1 = \cdots = \beta_{p-1} = 0, \\ H_1: \ \text{otherwise}. \end{cases}$$

In other words, we are essentially comparing the two fit

$$\begin{pmatrix} \overline{y} \\ \vdots \\ \overline{y} \end{pmatrix} = \left(\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\right)\boldsymbol{y} \text{ vs. } \widehat{\boldsymbol{y}} = \boldsymbol{x}\widehat{\boldsymbol{\beta}} = \underbrace{\boldsymbol{x}\left(\boldsymbol{x}^T\boldsymbol{x}\right)^{-1}\boldsymbol{x}^T}_{\boldsymbol{H}}\boldsymbol{y}$$

and examining the distance between the two. Intuitively, a very large difference would signal rejection. This is the idea behind the *analysis of variance* (ANOVA):

$$\underbrace{\sum_{i=1}^{n}(y_i-\overline{y})^2}_{\text{TSS}}=\underbrace{\sum_{i=1}^{n}(y_i-\widehat{y}_i)^2}_{\text{RSS}}+\underbrace{\sum_{i=1}^{n}(\widehat{y}_i-\overline{y})^2}_{\text{MSS}}, \tag{3.31}$$

where TSS (MSS, RSS) stands for total (model, residual) sum of squares. The idea is to use MSS as the test-statistic and reject H_0 if MSS is large.

Before we apply ANOVA to anything, however, we need to show (3.31) is true. To see this, we first rewrite (3.31) in matrix form. Let $J_n = 1/n\mathbb{1}_n\mathbb{1}_n^T$. Then (3.31) is equivalent to

$$\|(I_n-J_n)\boldsymbol{y}\|^2=\|(I_n-\boldsymbol{H})\boldsymbol{y}\|^2+\|(\boldsymbol{H}-J_n)\boldsymbol{y}\|^2. \tag{3.32}$$

It should not be hard to see that $I_n - J_n$ is the projection onto the orthogonal complement of column space of $\mathbb{1}_n$, and $I_n - \boldsymbol{H}$ onto that of $\boldsymbol{x}$. If we can show (i) $\boldsymbol{H} - J_n$ is also a projection and (ii) $I_n - \boldsymbol{H}$ and $\boldsymbol{H} - J_n$ are orthogonal, then (3.32) is nothing but a Pythagorean identity of projection matrices.

First, to see why $\boldsymbol{H} - J_n$ is a projection matrix, we need the following.

Lemma 3.33. *Let* $\boldsymbol{H}=\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T$. *Then* $\boldsymbol{H}\mathbb{1}_n=\mathbb{1}_n$.

Proof. First, we can write

$$\boldsymbol{x}=\begin{pmatrix}\mathbb{1}_n & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{p-1}\end{pmatrix}.$$

Then

$$\boldsymbol{H}\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{x}=\boldsymbol{x},$$

which means

$$\boldsymbol{H}\begin{pmatrix}\mathbb{1}_n & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{p-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{H}\mathbb{1}_n & \boldsymbol{H}\boldsymbol{x}_1 & \boldsymbol{H}\boldsymbol{x}_2 & \cdots & \boldsymbol{H}\boldsymbol{x}_{p-1}\end{pmatrix}=\begin{pmatrix}\mathbb{1}_n & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{p-1}\end{pmatrix}.$$

So $\boldsymbol{H}\mathbb{1}_n = \mathbb{1}_n$. It also follows that

$$(\boldsymbol{H}\mathbb{1}_n)^T=\mathbb{1}_n^T\boldsymbol{H}^T=\mathbb{1}_n^T\boldsymbol{H}=\mathbb{1}_n^T.$$

□

Therefore, $\boldsymbol{H} - J_n$ is idempotent because

$$\begin{aligned}(\boldsymbol{H}-J_n)(\boldsymbol{H}-J_n)&=\boldsymbol{H}+\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T-\boldsymbol{H}\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\boldsymbol{H}\\&=\boldsymbol{H}+\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\\&=\boldsymbol{H}-\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T=\boldsymbol{H}-J_n.\end{aligned}$$

It is also symmetric per the fact that both $\boldsymbol{H}$ and J are symmetric. Hence (i) is proven.

Next, to see why $I_n - \boldsymbol{H}$ and $\boldsymbol{H} - J_n$ are orthogonal, we just need to check that their inner product is $\mathbf{0}$:

$$(I_n - \boldsymbol{H})(\boldsymbol{H} - J_n) = \boldsymbol{H} - \boldsymbol{H}\boldsymbol{H} - J_n + \boldsymbol{H}\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T = \boldsymbol{H} - \boldsymbol{H} - J_n + J_n = \mathbf{0}.$$

Hence (ii) is proven. We are one step away from showing that the decomposition is true:

$$\begin{aligned}\|(I_n - J_n)\boldsymbol{y}\|^2 &= \|(I_n - \boldsymbol{H})\boldsymbol{y} + (\boldsymbol{H} - J_n)\boldsymbol{y}\|^2 \\ &= \|(I_n - \boldsymbol{H})\boldsymbol{y}\|^2 + \|(\boldsymbol{H} - J_n)\boldsymbol{y}\|^2 + 2\boldsymbol{y}^T(I_n - \boldsymbol{H})^T(\boldsymbol{H} - J_n)\boldsymbol{y} \\ &= \|(I_n - \boldsymbol{H})\boldsymbol{y}\|^2 + \|(\boldsymbol{H} - J_n)\boldsymbol{y}\|^2.\end{aligned}$$

Now it turns out that if you have followed the materials closely and truly understand the concept of intuition, there is a much quicker proof for the ANOVA decomposition. This originates from a previously established Pythagorean identity in Section 3.2, namely, if

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2,$$

then

$$\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 = \|\boldsymbol{y} - \boldsymbol{x}\widehat{\boldsymbol{\beta}}\|^2 + \|\boldsymbol{x}\widehat{\boldsymbol{\beta}} - \boldsymbol{x}\boldsymbol{\beta}\|^2$$

for all $\boldsymbol{\beta} \in \mathbb{R}^p$. The key here is that this works for **any** $\boldsymbol{\beta}$, so we can just set $\boldsymbol{\beta}$ to be the one such that $J_n\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta}$ (see Figure 15). It turns out

$$\boldsymbol{\beta} = \begin{pmatrix} \overline{y} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

With this choice of $\boldsymbol{\beta}$, the ANOVA decomposition immediately follows.

With the ANOVA decomposition proven, let us return to HT. With MSS as the intended test-statistic, we need to find its distribution. For this we need a new tool.

Definition 3.34. Let $Y_1 \sim \chi^2_{d_1}$, $Y_2 \sim \chi^2_{d_2}$, with $Y_1 \perp\!\!\!\perp Y_2$. Then

$$\frac{Y_1/d_1}{Y_2/d_2} \sim F_{d_1,d_2},$$

where F_{d_1,d_2} is a *F-distribution* with two different dofs d_1 and d_2.

Theorem 3.35. *Let $\boldsymbol{y} \sim N(\boldsymbol{x}\boldsymbol{\beta}, \sigma^2 I_n)$. Then the following holds true in general:*

(1) TSS = RSS + MSS.
(2) RSS $\perp\!\!\!\perp$ MSS.
(3) $RSS/\sigma^2 \sim \chi^2_{n-p}$.

The following holds true under the null:

(4) $TSS/\sigma^2 \sim \chi^2_{n-1}$.

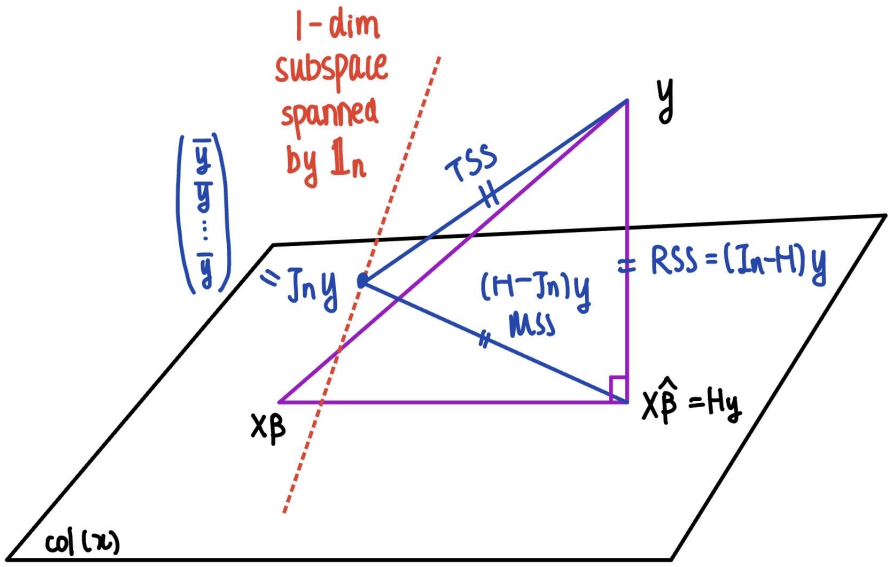


FIGURE 15. The Pythagorean Theorem on Column Space of $\boldsymbol{x}$

(5) $MSS/\sigma^2 \sim \chi^2_{p-1}$.
(6)

$$\frac{MSS/p-1}{RSS/n-p} \sim F_{p-1,n-p}.$$

The theorem should give some insight into why we introduce the new distribution. If σ^2 is known, we can just use (5) and call it a day. However, in our set-up, σ^2 is assumed to be unknown, so we need a distribution to help us cancel that parameter out. One might consider t-distribution, but it is hard to find something that follows a $N(0,1)$ distribution. With the F-distribution, however, (6) follows naturally from (2), (3) and (5), and we can just reject the null model when

$$\frac{\text{MSS}/p-1}{\text{RSS}/n-p} > F_{p-1,n-p,1-\alpha}.$$

Theorem 3.35 may look like a handful, but the truth is we have already done the majority of the work: (1) is the ANOVA decomposition, (3) is Property (2) of Theorem 3.23, (4) follows from Property (2) of Theorem 1.29, and (6) is just by definition. So the only ones that we truly need to work on are (2) and (5), and they are quick derivations from previous conclusions.

Proof. Let us start with (2). Recall

$$\text{RSS} = \|(I_n - \boldsymbol{H})\boldsymbol{y}\|^2, \text{MSS} = \|(\boldsymbol{H} - J_n)\boldsymbol{y}\|^2.$$

Note that both are linear transformations of $\boldsymbol{y}$ and thus jointly normal. To show independence, we only need to check that the covariance is 0:

$$\operatorname{Cov}\big((I_n - \boldsymbol{H})\boldsymbol{y}, (\boldsymbol{H} - J_n)\boldsymbol{y}\big) = (I_n - \boldsymbol{H})\mathbb{V}[\boldsymbol{y}](\boldsymbol{H} - J_n) = \sigma^2(I_n - \boldsymbol{H})(\boldsymbol{H} - J_n) = 0$$

due to them being orthogonal.

Next, to show (5), recall that under the null, $y_i \overset{\text{iid}}{\sim} N(\beta_0, \sigma^2)$, which can be rewritten as

$$\boldsymbol{y} = \beta_0 \mathbb{1}_n + \sigma^2 \boldsymbol{z},\ \boldsymbol{z} \sim N(0, I_n),$$

so

$$(\boldsymbol{H} - J_n)\boldsymbol{y} = \beta_0(\boldsymbol{H} - J_n)\mathbb{1}_n + \sigma^2(\boldsymbol{H} - J_n)\boldsymbol{z}.$$

However, we have previously shown $\boldsymbol{H} - J_n$ is orthogonal to the column space of $\mathbb{1}_n$, so the first term on the RHS is $\mathbf{0}$. Therefore,

$$\frac{\|(\boldsymbol{H} - J_n)\boldsymbol{y}\|^2}{\sigma^2} = \|(\boldsymbol{H} - J_n)\boldsymbol{z}\|^2 \sim \chi^2_{\operatorname{rank}(\boldsymbol{H} - J_n)}$$

by Lemma 3.30. But then

$$\operatorname{rank}(\boldsymbol{H} - J_n) = \operatorname{Tr}(\boldsymbol{H} - J_n) = \operatorname{Tr}(\boldsymbol{H}) - \operatorname{Tr}(J_n) = p - 1.$$

Hence (5). □

All this work gives us our first model selection test by F-distribution. However, we should not be too satisfied: this HT, though useful, only compares two models. It offers no guidance on how to select the variables in the first place, nor on identifying the irrelevant variables or how to remove them. It also offers no solution to one of earlier dilemmas we mentioned, about how to work with situations where the design matrix does not have full rank. These all motivates us to look at several other techniques, all related to penalized regression.

3.5. **Ridge Regression.**

The idea behind ridge regression is that, instead of minimizing $\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2$, we define the estimator to be

$$\widehat{\boldsymbol{\beta}}_\lambda = \underset{\beta}{\operatorname{argmin}} \left[\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda\|\beta\|^2\right]$$

for some $\lambda > 0$, i.e. we penalize any deviation of β from 0. The penalty term (second on RHS) is called *ℓ_2-penality* or *ℓ_2-regularization*. It should be apparent that the estimator approaches the LSE as $\lambda \to 0$. When $\lambda \to \infty$, however, the first term becomes negligible, and $\beta = 0$ for minimization.

One immediate question that should come to mind is, well, what is the appropriate choice of λ? This by itself is not apparent at all and requires a more complicated process to determine (which we will discuss soon). It also, like LSE, offers no explicit guidance on variable selection.

Ridge regression, however, does have its advantages. One advantage is that it is just as easily computable as LSE; in fact, in terms of notational stability, it performs even better than LSE

because of strong convexity. Another advantage of ridge regression is that it admits a closed form solution:

$$\begin{aligned}\|\boldsymbol{y}-\boldsymbol{x}\boldsymbol{\beta}\|^2+\lambda\|\beta\|^2 &= \|\boldsymbol{y}\|^2+\boldsymbol{\beta}^T\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{\beta}-2\boldsymbol{\beta}^T\boldsymbol{x}^T\boldsymbol{y}+\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}\\ &= \|\boldsymbol{y}\|^2+\boldsymbol{\beta}^T(\boldsymbol{x}^T\boldsymbol{x}+\lambda I_p)\boldsymbol{\beta}-2\boldsymbol{\beta}^T\boldsymbol{x}^T\boldsymbol{y}.\end{aligned}$$

Then, analogous to the ordinary LSE, the solution turns out to be

$$\widehat{\boldsymbol{\beta}}=(\boldsymbol{x}^T\boldsymbol{x}+\lambda I_p)^{-1}\boldsymbol{x}^T\boldsymbol{y}, \tag{3.36}$$

from which we can another benefit of ridge regression: due to the addition of I_p, there are no longer zero eigenvalues, and we can relax the requirement of $\boldsymbol{x}$ being full rank.

The extra term deserves attention, because it brings forth another cost and another benefit: $\widehat{\boldsymbol{\beta}}$ is now biased, with the bias getting progressively worse as $\lambda\to\infty$, but it also suffers from smaller variance (this is the easiest to see as $\lambda\to\infty$, because then $\widehat{\boldsymbol{\beta}}$ is deterministically 0).

Let us now return to the question of how to pick λ. The problem is if we examine the data as a whole and attempt to minimize $\|\boldsymbol{y}-\boldsymbol{x}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}\|^2$, it always returns $\lambda=0$ (this has something to do with Gauss-Markov theorem and LSE being the BLUE estimator, which is not covered in this course). Thus we need to rely on a procedure called *data splitting*.

The data splitting procedure is as follows: we examine the data

$$\boldsymbol{y}=\begin{pmatrix}y_1\\ \vdots\\ y_{n_1}\\ y_{n_1+1}\\ \vdots\\ y_n\end{pmatrix},\ \boldsymbol{x}=\begin{pmatrix}x_1^T\\ \vdots\\ x_{n_1}^T\\ x_{n_1+1}^T\\ \vdots\\ x_n^T\end{pmatrix}$$

and then split them into two groups

$$\boldsymbol{y}^{(1)}=\begin{pmatrix}y_1\\ \vdots\\ y_{n_1}\end{pmatrix},\ \boldsymbol{y}^{(2)}=\begin{pmatrix}y_{n_1+1}\\ \vdots\\ y_n\end{pmatrix},\ \boldsymbol{x}^{(1)}=\begin{pmatrix}x_1^T\\ \vdots\\ x_{n_1}^T\end{pmatrix},\ \boldsymbol{x}^{(2)}=\begin{pmatrix}x_{n_1+1}^T\\ \vdots\\ x_n^T\end{pmatrix}$$

such that there are n_1 observations in the first group and n_2 in the second, with $n_1+n_2=n$. (A technical detail: we always shuffle before splitting to ensure that the traits are balanced across the two groups.) Then we compute

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}=\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left[\|\boldsymbol{y}^{(1)}-\boldsymbol{x}^{(1)}\boldsymbol{\beta}\|^2+\lambda\|\boldsymbol{\beta}\|\right]^2=\left(\left(\boldsymbol{x}^{(1)}\right)^T\boldsymbol{x}^{(1)}+\lambda I_p\right)^{-1}\boldsymbol{x}^{(1)}\boldsymbol{y}^{(1)},$$

then examine

$$\widehat{\lambda} = \underset{\lambda>0}{\operatorname{argmin}} \|\boldsymbol{y}^{(2)} - \boldsymbol{x}^{(2)}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}\|^2.$$

Here $(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)})$ is the *training data* and $(\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)})$ the *testing data*. How to allocate the data depends mostly on our intention, but the rule of thumb is 80%-20% or 70%-30%, since we do want a good training model. The two together gives the final estimator $\widehat{\beta}_{\widehat{\lambda}}$.

To obtain a more intuitive understanding, let us examine the relationship between error levels and λ w.r.t. to the two sets of data (see Figure 16). First, in the training data, we know as λ increases, the magnitude of the estimated coefficient must decrease and increasingly deviate from LSE, so the training data must be strictly increasing in λ. On the other hand, per above, we know that the testing error levels have to be large as $\lambda \to 0$ (LSE, overfitting) and $\lambda \to \infty$ (underfitting), so that would follow a quadratic pattern. Thus the testing error curve summarizes the statistical benefits of ridge regression: we would get an estimator with good generalization behavior (λ not too small) but also not too biased (λ not too large).

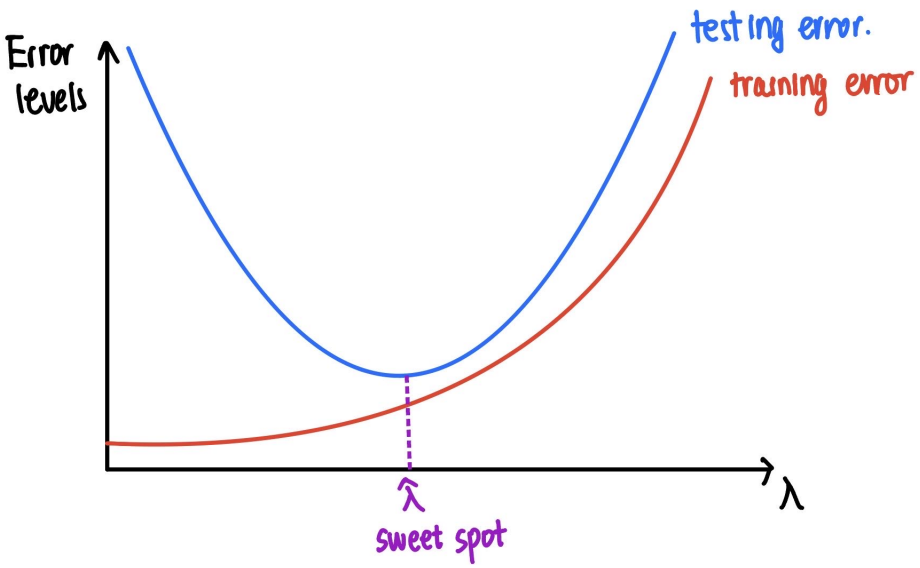


FIGURE 16. Training and Testing Error Levels in Ridge Regression

Let us conclude this section by summarizing the advantages of ridge regression:

(1) Ridge regression frees us from the concern of ensuring the design matrix to be full rank. In fact, it can be used in high-dimensional data, where $n < p$, which is by definition low-rank.
(2) Ridge regression yields a closed form solution and is computationally efficient due to strong convexity, i.e. all eigenvalues of the Hessian is bounded away from 0 (another benefit per the extra I_p term). The ordinary least-squares problem (without $\lambda\|\boldsymbol{\beta}\|^2$) only has convexity.

On the other hand, its disadvantage is that it does not offer explicit variable selection. While some set-ups can work with this, others cannot: imagine the significant genes identifications. Out of the tens of thousands of genes we are examining, we only want to select a few to concentrate on, and we also need to know which ones they are. This demand gives us the occasion to introduce other techniques.

3.6. **Best Subset Selection.**

To compensate for the missing feature in both OLS and ridge regressions, here we prioritize variable selection. Let $S \subseteq \{1, 2, ..., p\}$ (note that in applications we do not usually include intercepts, but we include it for generality), with $\boldsymbol{x} \in \mathbb{R}^{n \times p}$. Then we can divide $\boldsymbol{x}$ into

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_S & \boldsymbol{x}_{S^c} \end{pmatrix},$$

where $\boldsymbol{x}_S$ contains all the columns of $\boldsymbol{x}$ indexed by S, and $\boldsymbol{x}_{S^c}$ for the complement of S. It is not too hard to derive the corresponding closed form solution

$$\widehat{\boldsymbol{\beta}}_S = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{x}_S\boldsymbol{\beta}\| = \left(\boldsymbol{x}_S^T\boldsymbol{x}_S\right)^{-1}\left(\boldsymbol{x}_S^T\boldsymbol{y}\right).$$

It is also not too hard to see that this method shares the same caveat as ridge regression: if we just minimize $\|\boldsymbol{y} - \boldsymbol{x}_S\widehat{\boldsymbol{\beta}}_S\|^2$ w.r.t. S, it is always going to return the full set of variables, because the errors is a non-decreasing function of the number of variables we have. This serves as the basis for - drum roll, please - a penalty term. The data splitting technique introduced in the previous subsection is a possibility works the exact same way here, but there are other techniques available as well.

3.6.1. *AIC & BIC.*

The *Akaike information criterion* (AIC) proposes to minimize the following:

$$\|\boldsymbol{y} - \boldsymbol{x}_S\widehat{\boldsymbol{\beta}}_S\|^2 + 2\sigma^2|S|,$$

where $|S|$ is the size of S. (As an aside, the name 'Akaike' refers to Hirotugu Akaike, a famous Japanese statistician who made significant contributions to time-series analysis and information theory.)

The *Bayesian information criterion* (BIC) proceeds along similar lines, with the difference being the weight of penalty used:

$$\|\boldsymbol{y} - \boldsymbol{x}_S\widehat{\boldsymbol{\beta}}_S\|^2 + \log(n) \cdot \sigma^2|S|.$$

(Another side: the BIC was not derived by Bayes but by Gideon M. Schwarz, who used a Bayesian formulation for his argument of adoption.)

There are other information criteria (for instance, the residual information criterion (RIC), the focused information criterion (FIC), etc.) but the AIC and BIC are the most frequently used.

There is one downside we can already see: if we naively try all combinations of S, we would end up with 2^p trials, which is completely infeasible for large p's. Thus to implement either the AIC or the BIC, we need to make educated guesses to reduce the potential S's to a reasonable range

(interestingly, this does sound like the Bayesian framework, where you have to have some sort of priors). This is both risky and annoying: risky because there is a chance to leave out significant variables that we are previously not aware of, and annoying because pre-screening usually only identify marginal importance, not joint ones. To perform the procedure properly, we have to go through several stages, including performing a marginal screening to calculate the marginal correlation of variables w.r.t. $\boldsymbol{y}$, etc.

Before moving on, here are some intuition for why the variance σ^2 matters in AIC and BIC. Recall that the goal of adding the penalty term in the first place is to control the variance, which directly determines the predictive power. If $\sigma^2 = 0$, the second term vanishes and it becomes the normal least-squares procedure. This aligns with the fact that, when $\sigma^2 = 0$, the data is perfectly accurate and we should just use it directly.

Note that even though they look similar, AIC and BIC are derived from very different assumptions. We won't go into details, but AIC is derived by assuming none of the sub-models is true (basically the truth is some arbitrary vector out there), while BIC assumes one of the sub-models is true, and so

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta}_S + \sigma Z$$

for some S (and the choice of which S's to test is heavily prior-dependent). The point of all this is that generally we should not expect the two to output the same model, and the convention is to make adjustments after seeing the results of both.

There is a reason though why the two minimization problems look similar: they are both specific instances of ℓ_0-penalty / regularization, which we will examine next.

3.7. ℓ_0 Regularization.

ℓ_0 regularization works with the following set-up:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_0 \right),$$

where we define

$$\|\boldsymbol{\beta}\|_0 = \sum_{i=1}^{p} \mathbb{1}[\beta_j \neq 0] = |\operatorname{supp}(\boldsymbol{\beta})|,$$

with $\operatorname{supp}(\beta) = \{j : \beta_j \neq 0\}$. Then λ is just the penalty for the size of the support.

We can see the remark about computational infeasibility problem also carries over to the general framework. The term $\|\beta\|_0$ does not allow convex optimization anymore.

To see how this translates to AIC/BIC, observe that we can convert this problem to a 2-step process:

$$\begin{aligned}
\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_0 &= \min_{S} \min_{\operatorname{supp}(\beta)=S} \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_0 \\
&= \min_{S} \min_{\operatorname{supp}(\beta)=S} \|\boldsymbol{y} - \boldsymbol{x}_S\boldsymbol{\beta}_S\|^2 + \lambda |S|,
\end{aligned}$$

and setting $\lambda = 2\sigma^2$ ($\lambda = \sigma^2 \log(n)$) recovers AIC (BIC).

Thus ℓ_0 regularization makes the process more interpretable at the cost of computational power. To comfort our genetic scientists, we need to find an approach that incorporates both the shrinkage (render the problem computationally feasible) and the selection features, which is the motivation for ℓ_1-regularization.

3.8. **Lasso Regression.**

The ℓ_1 regularization process, a.k.a. least absolute value shrinkage selection operation (Lasso), works with the following:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 \right),$$

where

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

First good news: this thing is convex optimizable by software.

Lasso regression offers interesting solutions structurally. To demonstrate what we mean by that, we start with an example.

Example 3.37. (1-Dimensional Lasso Example)
Suppose we are facing the following problem:

$$\min f(\beta) = \min_{\beta} \left[(y - \beta)^2 + \lambda |\beta| \right].$$

Taking the derivative yields

$$\begin{cases} 2(\beta - y) + \lambda, & \beta > 0, \\ 2(\beta - y) - \lambda, & \beta < 0. \end{cases}$$

Setting each to 0 gives

$$\widehat{\beta} = \begin{cases} y + \frac{\lambda}{2}, & y < -\frac{\lambda}{2}, \\ y - \frac{\lambda}{2}, & y > \frac{\lambda}{2}. \end{cases}$$

We are obviously missing a case, when $y \in [-\lambda/2, \lambda/2]$. The key here is to split into subcases:

- If $\beta > 0$,
$$f'(\beta) = 2(\beta - y) + \lambda = 2\beta - 2\left(y - \frac{\lambda}{2}\right) > 0.$$
- If $\beta < 0$,
$$f'(\beta) = 2(\beta - y) - \lambda = 2\beta - 2\left(y + \frac{\lambda}{2}\right) < 0.$$

The two combined means that $\widehat{\beta}$ is optimized at 0 when $y \in [-\lambda/2, \lambda/2]$, so the complete solution is

$$\widehat{\beta} = \begin{cases} y + \frac{\lambda}{2}, & y < -\frac{\lambda}{2}, \\ 0, & y \in \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right], \\ y - \frac{\lambda}{2}, & y > \frac{\lambda}{2}. \end{cases}$$

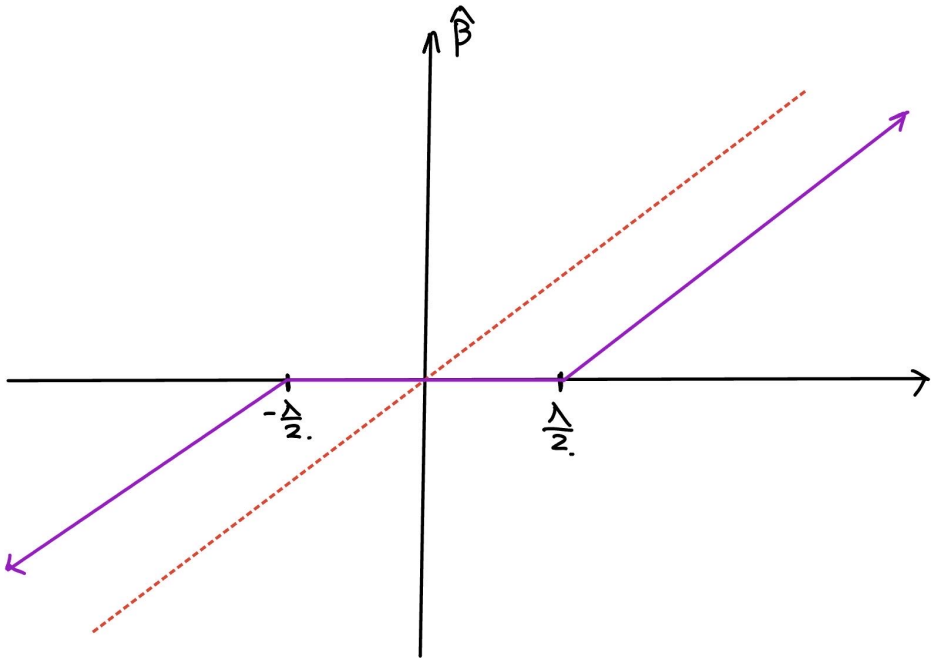


FIGURE 17. Lasso Regression 1-Dimensional Optimizer

The shape of the solution gives us a first glimpse into the power of ℓ_1 regularization. First, when the data y is small (recall that λ in general should not be too large to avoid bias), the minimizer is 0, which corresponds to selection. For y sufficiently large in magnitude, there is a tendency to reduce y towards zero, thus the effect of shrinkage. Hence the minimizer inherits features from both ℓ_2 and ℓ_0 regularization.

Example 3.38. (2-Dimensional Lasso Example)
Let us extend the observation to higher dimensions. For ease of exposition, we note that for each ℓ_1-optimization problem, there exists a *penalty form*

$$\min_{\boldsymbol{\beta}} \left(\|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 \right)$$

and an equivalent *constraint form*

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 \text{ s.t. } \|\boldsymbol{\beta}\|_1 < C,$$

where C is a constant that is determined by the value of λ (the latter should look somewhat familiar to those have learned Lagrange multipliers before).

Drawing out the constraints (diamond shape) and the contours of $\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}$ (recall the shape of MVN contour density), we get something like Figure 18. Intuitively, the point on the sample point that

happens to have the smallest ℓ_2-distance (i.e. closest) to $\boldsymbol{y}$ and satisfies the constraint is the desired minimizer, which also is the first point where the contour line touches the constraint area. Here, interestingly, the point of contact lies exactly on the β_2-axis, which yields the solution $(0, C)$. This is variable selection at work, and we keep only one out of the two variables.

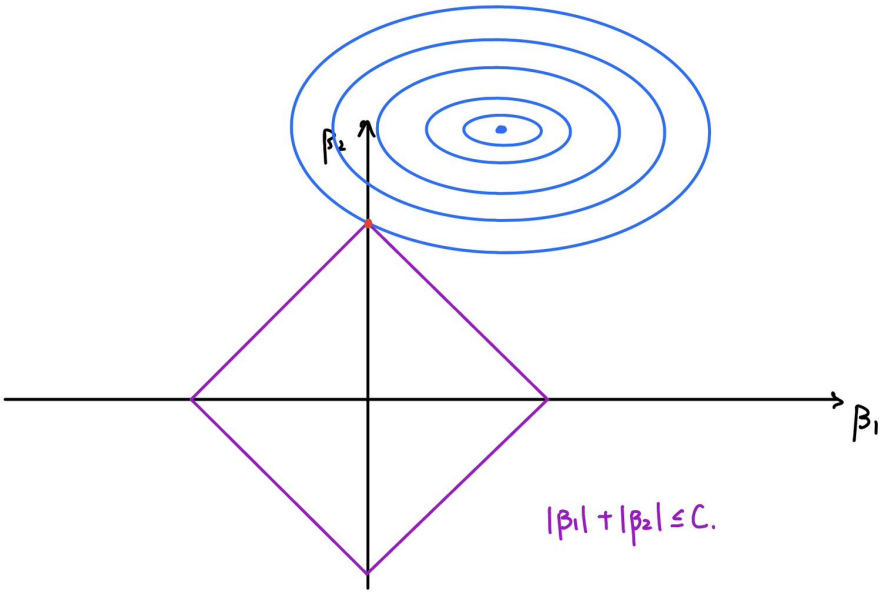


FIGURE 18. Lasso 2-Dimensional Optimizer with Selection

The above is in contrast to Figure 19, where both $\beta_1, \beta_2 > 0$ and we need both coefficients to stay.

We have noted that ℓ_1 regularization preserves convexity, so such solution is unique. Lastly, note that if $\boldsymbol{y}$ is situated within the constraint, then there exists $\boldsymbol{\beta}$ such that $\boldsymbol{x\beta} = \boldsymbol{y}$ while already satisfying the constraint, and there is no need for the minimization problem.

One downside of Lasso is that there is no closed solution - we simply rely on convex programming and algorithms. One of the most commonly used methods is called *coordinate descent*, where we solve for the minimizer coordinate by coordinate. To do coordinate descent, we optimize $\boldsymbol{\beta}_i$ while keeping all others fixed, in order. To see why this solution works, let us verify the process for one coordinate. First, return to Example 3.37 and define

$$S_{\frac{\lambda}{2}}(y) = (y - \beta)^2 + \lambda|\beta|,$$

to be the *soft-threshold operator* (here we call $\lambda/2$ the *threshold level*). (As an aside, 'soft' just means the solution performs shrinkage; 'hard' does not.)

Let us try to upgrade this concept to higher dimensions. First, note that

$$\|\boldsymbol{y} - \boldsymbol{x\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 = \|\boldsymbol{y} - \boldsymbol{x}_j\beta_j - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\|^2 + \lambda|\beta_j| + \lambda \sum_{i:i\neq j} |\beta_i|,$$

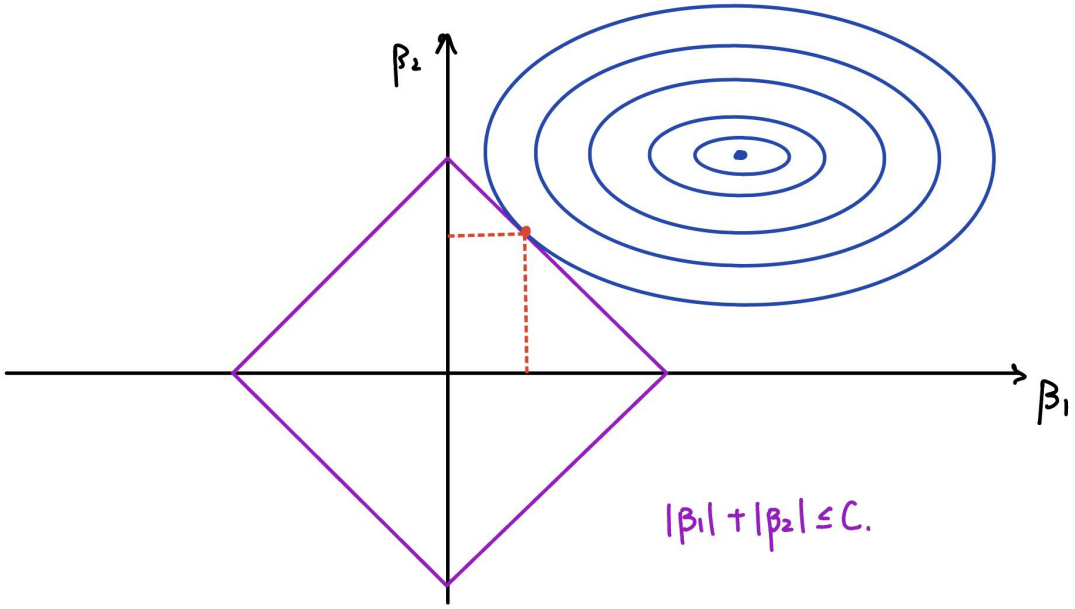


FIGURE 19. Lasso 2-Dimensional Optimizer without Selection

where $\boldsymbol{x_j}$ is the jth column extracted from $\boldsymbol{x}$. The goal is to find the minimizer for β_j. The last term does not depend on β_j, so we can write the objective as

$$\min_{\beta_j} \|\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j} - \boldsymbol{x}_j\beta_j\|^2 + \lambda|\beta_j|.$$

Expanding the objective function gives

$$\|\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j} - \boldsymbol{x}_j\beta_j\|^2 = \|\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\|^2 + \beta_j^2\|\boldsymbol{x}_j\|^2 - 2\beta_j\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right) + \lambda|\beta_j|.$$

The first term on the RHS again does not depend on β_j, so we focus on optimizing the rest. But then they can be rewritten as

$$\begin{aligned}
&\beta_j^2\|\boldsymbol{x}_j\|^2 - 2\beta_j\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right) + \lambda|\beta_j| \\
=&\|\boldsymbol{x}_j\|^2\left(\beta_j^2 - 2\beta_j \cdot \frac{\boldsymbol{x_j}^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2} + \frac{\lambda}{\|\boldsymbol{x}_j\|^2}|\beta_j|\right) \\
=&\|\boldsymbol{x}_j\|^2\left(\left(\beta_j - \frac{\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2}\right)^2 + \frac{\lambda}{\|\boldsymbol{x}_j\|^2}|\beta_j| - \left(\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2}\right)^2\right) \\
=&\|\boldsymbol{x}_j\|^2\left(\left(\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2} - \beta_j\right)^2 + \frac{\lambda}{\|\boldsymbol{x}_j\|^2}|\beta_j| - \left(\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y} - \boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2}\right)^2\right).
\end{aligned}$$

The last term and the coefficient, for the third time, does not depend on β_j, so we can focus on minimizing

$$\left(\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y}-\boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2}-\beta_j\right)^2+\frac{\lambda}{\|\boldsymbol{x}_j\|^2}|\beta_j|.$$

This is by no means evident, but if we set

$$y'=\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y}-\boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2},\ \text{threshold}=\frac{\lambda}{\|\boldsymbol{x}_j\|^2},$$

then we recover the soft-threshold operator for β_j:

$$\widehat{\beta}_j=S_{\frac{\lambda}{2\|\boldsymbol{x}_j\|^2}}\left(\frac{\boldsymbol{x}_j^T\left(\boldsymbol{y}-\boldsymbol{x}_{-j}\boldsymbol{\beta}_{-j}\right)}{\|\boldsymbol{x}_j\|^2}\right).$$

This serves as the algorithm for cyclic descent. The default is *cyclic coordinate descent*, where we perform the procedure for $j=1,...,m$ in order. This may have some benefits depending on the exact set-up but is generally not required, as shown as by the algorithm.

Before we move on, note that even though here we categorize the regressions w.r.t. the type of penalties, they are by no means mutually exclusive: in real applications, we usually use combinations of several methods. For example, elastic net regression employs both ℓ_1 and ℓ_2 penalties. Some procedures also incorporates Lasso and AIC/BIC when the shrinkage effect is not desirable.

3.9. **Post Selection Inference.**

It turns out that performing statistical inference when there is explicit variable selection (e.g. in ℓ_0 regularization or Lasso regression) is much more complicated (here I mean conceptually rather than practically) compared to all the previous scenarios we have seen. One issue is that of selection bias: intuitively, we select one certain specification over another because we have more confidence in it, which means we should also choose more stringent criteria for rejection. More importantly, however, we may run into trouble with the hypothesis tests not being well-defined themselves.

Let $\widehat{S}$ contain the index of the subset of the variables we have chosen. Suppose $7\in\widehat{S}$, and the goal is to provide a CI for β_7. However, we have an incentive to do so if and only if it is selected in the first place, so whether the hypotheses themselves exist is a random event. If we continue this line of thought, all the other artifacts dependent on the hypothesis, for instance the p-value, the decision rule, are also random ... which does not make sense at all.

If the above sounds abstract, let us return to the univariate selection model for a more concrete example.

Example 3.39. Suppose we have two models

$$M_0=y_i\sim N(\beta_0,1),$$
$$M_1=u_i\sim N(\beta_0+\beta_1x_i,1)$$

for $i=1,...,n$ independently. The goal is to construct a CI for β_0 for a given confidence level α. There are three possibilities here.

(1) Suppose we trust M_0. Then the CI is just
$$\left[\overline{y} \pm \frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}\right].$$
This gives the correct coverage under M_0.

(2) Suppose we trust M_1. Then
$$\widehat{\beta}_0 \sim N\left(\beta_0, \frac{1}{n}\left(1 + \frac{\overline{x}^2}{V}\right)\right).$$
We can just use
$$\left[\widehat{\beta}_0 \pm z_{1-\frac{\alpha}{2}}\sqrt{\frac{1}{n}\left(1 + \frac{\overline{x}^2}{V}\right)}\right].$$
This gives the correct coverage under M_1.

(3) We are unsure and would like to make a decision based on information from data. Intuitively, we reject M_0 when $\widehat{\beta}_1$ is large (so there is a nested hypothesis testing going on with $H_0 : \beta_1 = 0$). If we borrow the appropriate quantile from Example 3.15, which we denote as c here, then the CI is
$$\begin{cases} \left[\overline{y} \pm \frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}\right], & |\widehat{\beta}_1| < c, \\ \left[\widehat{\beta}_0 \pm z_{1-\frac{\alpha}{2}}\sqrt{\frac{1}{n}\left(1 + \frac{\overline{x}^2}{V}\right)}\right], & |\widehat{\beta}_1| \geq c. \end{cases}$$
But then this is a random event that depends on data, since $\widehat{\beta}_1$ is a RV itself. Thus in either case, the coverage cannot be correct (in fact, it is usually undercovered).

This existence of this issue is actually quite reasonable: the decision to choose a certain subset of variables over another is usually driven by prior information or existing data. When we perform statistical inference, we are intentionally ignoring the fact that we have extra information, and in statistics, it is generally a sin to not use all that we already have.

One proposal to remedy the issues above would be to use conditioning. For instance, in the initial example, we can just condition everything on $7 \in \widehat{S}$, whose probability can be calculated. However, this also means that we would be examining $Y_i|7 \in \widehat{S}$ rather than Y_i's, which we have no guarantee to be normally distributed (in fact, the conditional distribution is most likely not normally distributed). Again, we end up with something we do not know. This is a working field, and currently such problems are only solvable by Monte Carlos methods.

However, notice that in both Example 3.39 and above, it seems to be the unknown conditional distribution that messes things up. Thus one correct solution to would be to use this but per a procedure through which the original distribution would not be affected. Data splitting is a good candidate here, because it gives us independence between the two sets. If we split the data into halves, what we do to the first half does not affect the second, and
$$Y|7 \in \widehat{S} = Y \sim N(X_S\beta_S, \sigma Z).$$

3.10. **Logistic Regression.**

The goal of this section is to work with a particular constraint on y_i. Recall the original framework $y_i \sim N(\mu_i, \sigma^2)$ independently, where $\mu_i = x_i^T \beta$. Suppose now $y_i \in \{0, 1\}$ only. This may look like a ridiculous constraint, but there are numerous examples in life where y_i follows a binary decision rule:

(1) spam email detection: the algorithm takes 'sensitive words' as data and then decides whether to put the email into the spam folder,
(2) fraud transaction: the algorithm takes location and type (amount) of transaction into account to decide whether to stop a certain transaction or send an alert message,
(3) significant gene expression detection,
(4) college application decision (though not machined-based),

and so on.

Let us try to set up the framework. Let $y_i \sim \text{Bernoulli}(p_i)$ independently, where we want the identification relation $p_i \sim x_i^T \beta$. This, however, does give us a problem: we must have $p_i \in (0, 1)$ as an input. Thus we need an invertible function from $\mathbb{R} \to (0, 1)$. Strictly monotone functions, like the one in Figure 20 are the easiest to maneuver. The normal CDF is a good candidate, which corresponds to the *probit* model. Here we will focus on the *sigmoid* function

$$\sigma(t) = \frac{1}{1 + e^{-t}},$$

which corresponds to the *logit* model.

To find a good candidate for the estimator, we start again with the MLE. The likelihood function is

$$p(y) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i}.$$

Taking log gives

$$\log(p(y)) = \sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i).$$

Substituting the sigmoid function yields

$$L(\beta) = \sum_{i=1}^{n} y_i \log\left(\sigma(x_i^T \beta)\right) + (1 - y_i) \log\left(1 - \sigma(x_i^T \beta)\right).$$

This is a concave function, so solving for

$$\widehat{\beta} = \underset{\beta}{\text{argmax}}\, L(\beta) = \underset{\beta}{\text{argmin}}(-L(\beta))$$

is just a convex optimization problem. To delve deeper though, we first need to understand what exactly are convex functions. Intuitively, a function is *convex* if it always lies below the segment connecting any two points on itself (Figure 21). Otherwise it is non-convex (Figure 22).

Here is a mathematical definition (not covered in class, credit to Economics Analysis II handout).

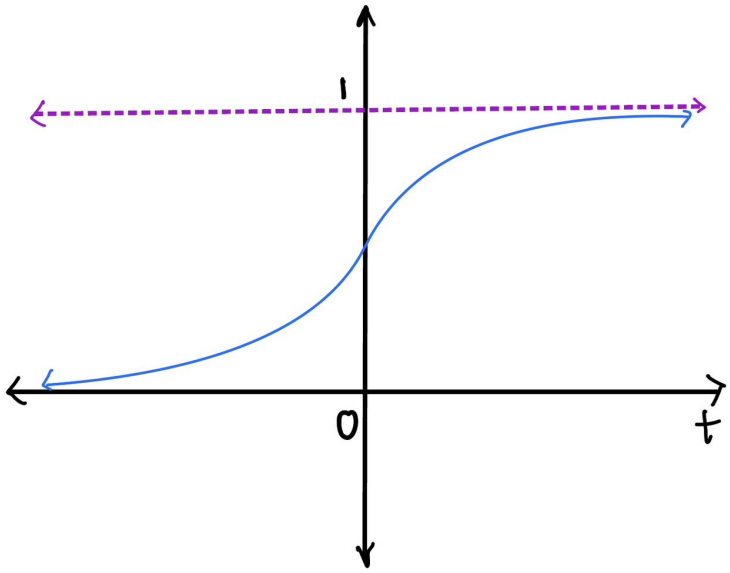


FIGURE 20. Transformation Function for Probit & Logit Models

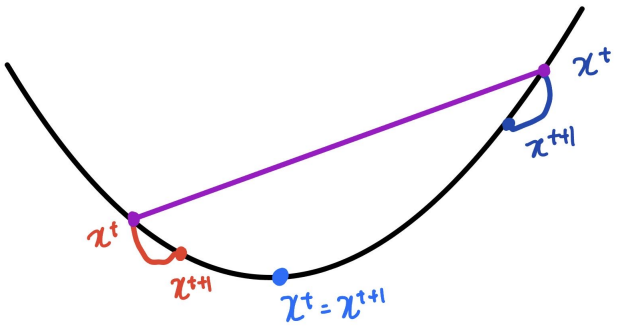


FIGURE 21. A Convex Function

Definition 3.40. A function $f : \mathbb{R} \to \mathbb{R}$ is *convex* if for all $x, x' \in \mathbb{R}$ and $\alpha \in [0, 1)$,

$$f\big(\alpha x + (1-\alpha)x'\big) \leq \alpha f(x) + (1-\alpha) f(x').$$

3.10.1. *Gradient Descent.*

To do convex optimization, *gradient descent* is the simplest and one of the most popular algorithms as of right now. We start with the univariate case. Let x be the variable we are trying to optimize.

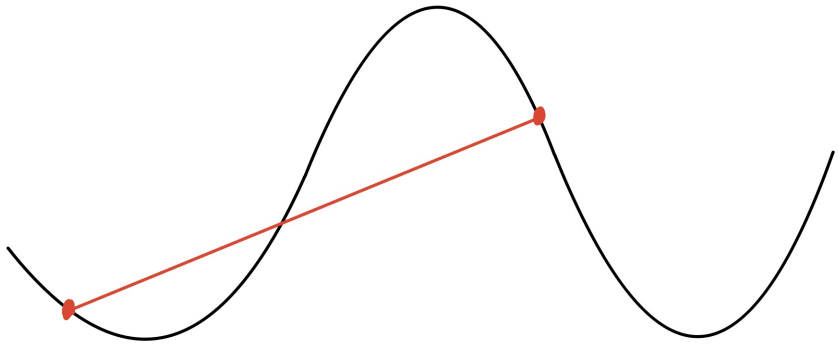


FIGURE 22. A Non-Convex Function

The idea is to examine the derivative at x^t, where x^t is the value at the t-th iteration. Given $f'(x^t)$, we choose the next update

$$x^{t+1} = x^t - \eta f'(x^t),$$

where η is called the *step size* or the *learning rate*. Several remarks:

- We stop when $f'(x^t) = 0$, or $x^t = x^{t+1}$, when coincides with the usual notion of convergence.
- When the step size is chosen 'wisely', this procedure applies regardless of whether $f'(x^t) < 0$ or $f'(x^t) > 0$. In the former case, $x^{t+1} > x^t$, tending towards the optimal point (red portion on the left of Figure 21). In the latter case, $x^{t+1} < x^t$, and x moves to the left (blue portion on the right of Figure 21).
- The step size must be chosen wisely: if it is too large, x will end up oscillating between the two sides; if it is too small, we would get a very slow convergence. Thus the optimal choice of η depends on the local curvature and is usually chosen by a process called *line search*.

With a sufficiently small step size and a sufficient number of iterations, we are guaranteed to have convergence, i.e. $x^t \to x^*$.

The multivariate case uses essentially the same formula except we exchange the derivative for the gradient (hence the name):

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t).$$

Now, to solve for β in the logistics regression, we simply apply the framework:

$$\beta^{t+1} = \beta^t + \eta \nabla(\beta^t),$$

where we add instead of subtract the second term due to maximization. This is also a generic algorithm that can be used in any situation. As a demonstration, we will use the gradient descent to solve for $\widehat{\boldsymbol{\beta}}_{\text{LSE}}$ below.

Example 3.41. Recall the objective is to minimize

$$L(\beta) = \|\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}\|^2 = (\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{x}\boldsymbol{\beta}).$$

Per some matrix calculus rules,

$$\nabla L(\boldsymbol{\beta}) = 2\boldsymbol{x}^T(\boldsymbol{x}\boldsymbol{\beta} - \boldsymbol{y}).$$

Substitute the above into the algorithm gives

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - 2\eta\boldsymbol{x}^T(\boldsymbol{x}\boldsymbol{\beta}^t - \boldsymbol{y}),$$

and we repeat until convergence. Note that even though gradient descent is technically not needed due to the existence of a closed-form solution, the procedure does bring benefit in the sense that there is no matrix inversion involved (no $(\boldsymbol{x}^T\boldsymbol{x})^{-1}$ term), thus potentially easier to solve computationally as dimensions get large.

Even though the gradient descent algorithm may liberate us from computationally-expensive procedures, it itself can get vexing as the dimension of $\boldsymbol{\beta}$ gets large. This serves as the motivation for its variant, the *stochastic gradient descent*, which we calculate the gradient on a small subset or even one sample (called *min-batch*) for each update. In other words, in the original stochastic descent, we use a subset of the i's in

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} L_i(\boldsymbol{\beta}).$$

The resulting update is noisy but moves in the right direction on average, since we are not deleting data but saving them for later updates. Compared to the regular stochastic descent, it saves time when the two algorithms converge around the same steps, since each step of the latter is simpler. The stochastic version is also particularly useful as the only method available in non-convex optimizations.

This concludes the discussion of logistics regression. In the next section, we will see how applications of logistics regression motivate the development of neural networks.

4. Neural Networks

(Remark: this section is to meant to be an overview of the topic and is thus by no means accurate or comprehensive. There may exist conceptual mistakes. Readers are encouraged to be cautious. Furthermore, there will be no distinction marked by bolded versus regular letters for simplicity.)

We start with specific applications that involve regression with binary outputs.

4.1. **Classification Model.**

Example 4.1. (Image classification)
Handwritten digit recognition started as early as 1960s, when the US postal office posed a challenge, asking for algorithms to recognize handwritten digits. WLOG, we limit the digits to 0 and 1. Let x_i each represent an image for identification with resolution 16×16. Then for each pixel, we record 1 if there exists something and 0 otherwise. To perform the logistic regression, we proceed to stack the columns of each x_i into one 256-dimensional vector of 0's and 1's. Thus the classical model would resemble Figure 23. However, this is not an efficient approach. Intuitively, algorithms work best when they use some heuristics like humans implicitly do.

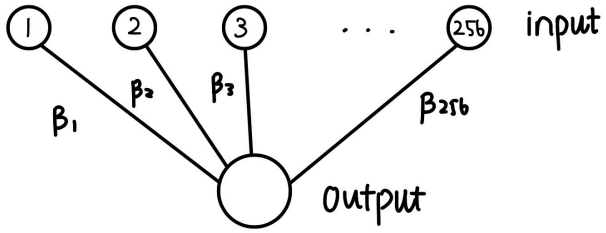


Figure 23. Image Classification: Classical Approach

The generalized version of this idea is to build some structure on the search, or rather on the information we are looking for. For example, in spam email detection, instead of taking every letter or every word into account, we can assemble the 'likely' words, such as 'prize', 'congratulations', 'lottery', etc., and then perform a logistic regression on those. Similarly, in our digit recognition setting, instead of examining what pixel $(3, 3) = 1$ implies about the digit, we can engineer different *feature vectors*, which gives rise to an entire area known as *computer vision*. For example, to identify '1', we can design matrices with a column of 1's in the middle

$$\begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix},$$

then take the inner product. This is called a *line detector*. Similarly, to identify 0, we can design a small-window 3×3 submatrix, e.g.

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

(where the '-1' serves as penalty since we know there should be an empty area in the middle), then slide it around to detect if there exists an edge in the corresponding places. This is known as an *edge detector*. Thus, for any $x \in \{0,1\}^{256}$, we first compute the q-dimensional feature vector

$$h := \begin{pmatrix} h_1 & h_2 & \cdots & h_q \end{pmatrix}^T = \begin{pmatrix} \langle x, w_1 \rangle & \langle x, w_2 \rangle & \cdots & \langle x, w_q \rangle \end{pmatrix},$$

where $w_1, w_2, ..., w_q$ are the various detectors we designed. Then we run the logistic regression on features (instead of pixels):

$$L(\beta) = \sum_{i=1}^{n} \left[y_i \log \left(\sigma(h_i^T \beta) \right) + (1 - y_i) \log \left(1 - \sigma(h_i^T \beta) \right) \right].$$

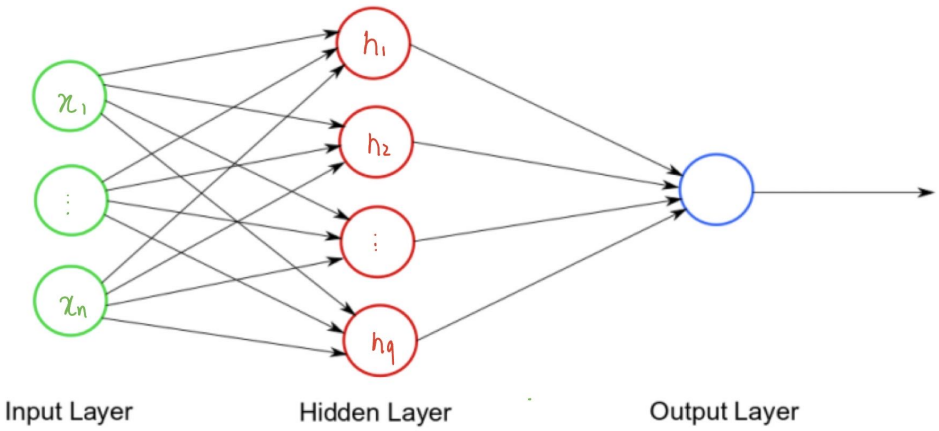


FIGURE 24. Neural Network: One Hidden Layer

(Image credit to this website.)

There are several potential issues with this method. First, we observe that the output is still a linear combination of h_j's, which is in turn a linear combination of x_i's and thus not desirable. So instead of directly using the feature, we can apply another sigmoid non-linear function, i.e. set

$$h = \begin{pmatrix} \sigma(\langle x, w_1 \rangle) & \sigma(\langle x, w_2 \rangle) & \cdots & \sigma(\langle x, w_q \rangle) \end{pmatrix}^T.$$

Then h is no longer a linear combination of the x_i's.

Another major issue is that the solution is heavily dependent on the engineering part. Thus a further improvement we can consider (which also results in the decline of computer vision groups) is to optimize over both β and w first instead of fixing w first, i.e. we change the problem to maximizing

$$L(\beta; w) = \sum_{i=1}^{n} \left[y_i \log \left(\sigma(\beta^T \sigma(w x_i)) \right) + (1 - y_i) \log \left(1 - \sigma(\beta^T \sigma(w x_i)) \right) \right].$$

In other words, we let data tell us what features are the best instead of picking them ourselves. This is the idea behind *neural networks*. It is no longer convex optimization but it also eliminates the need for unreliable human ingenuity.

As an aside, the idea of neural networks appeared as early as 1950s/60s, but useful applications would have to wait until much later for developments in computational power and data storage.

For more advanced improvements, we can keep adding the hidden layers to form *deep neural networks* (see Figure 25 for demonstration):

$$\beta^T \sigma(w_L \sigma(w_{L-1} \cdots \sigma(w_1 x))).$$

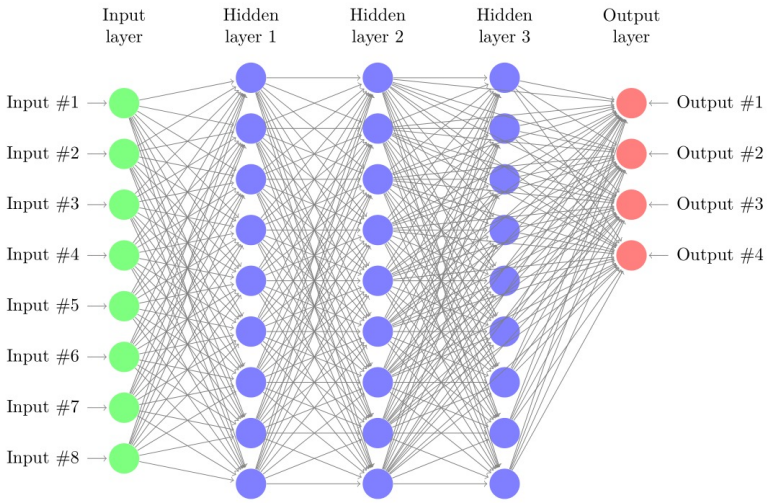


FIGURE 25. Neural Network: Multiple Hidden Layers

(Image credit to this website.)

It suffers from expensive training and retrieves only local minima as solutions (non-convex optimization), but it works.

There is one more improvement to take into account, due to the realization that the sigmoid transformation does not work that well in practice. Recall that

$$\sigma(t) = \frac{1}{1+e^{-t}},$$

so the function is flat for very small or very large values of t, which means the derivative is approximately 0. The chain rule caused by multiple hidden layers only exacerbate this reduction effect. Thus the motivation to replace $\sigma(\cdot)$ by the ReLU, or *rectified linear unit*, function:

$$\mathrm{ReLU}(t) = \max\{0, t\}.$$

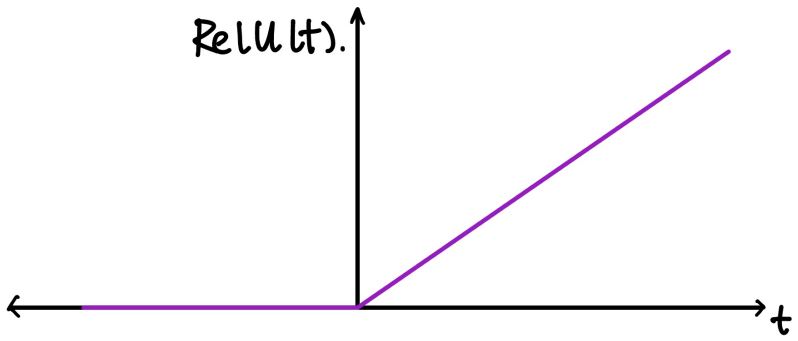


FIGURE 26. ReLU Function

By its form, the ReLU function solves the problem of gradient vanishing but still gives feature extraction: it deselects the feature when the values are negative, and when it learns the features, the values are pushed towards the positive region. Thus the deep neural network would be formed in this way:

$$\begin{aligned} x_i &\mapsto w_1 x_i \mapsto \mathrm{ReLU}(w_1, x_i) \mapsto w_2 \mathrm{ReLU}(w_1, x_i) \mapsto \cdots \mapsto \mathrm{ReLU}(w_L \cdots w_2 \mathrm{ReLU}(w_1, x_i)) \\ &\mapsto \beta^T \mathrm{ReLU}(w_L \cdots w_2 \mathrm{ReLU}(w_1, x_i)) =: D_{\beta,w}(x_i). \end{aligned}$$

We assume

$$y_i \sim \mathrm{Bernoulli}(\sigma(D_{\beta,w}(x_i)))$$

independently, and the optimization problem is

$$L(\beta; w) = \sum_{i=1}^{n} \left[y_i \log\left(\sigma(D_{\beta,w}(x_i))\right) + (1-y_i) \log\left(1 - \sigma(D_{\beta,w}(x_i))\right) \right].$$

A natural question that follows is, well, how should we determine the size of the neural network, i.e. the number of hidden layers and their widths respectively? The following theorem is a good starting point.

Theorem 4.2. *(Universality)*
Any non-linear function can be approximated by a neural network with one hidden layer.

Therefore, a neural network with a single hidden layer can reproduce any non-linear relation between covariates and output. However, in practice, we usually employ more than one layer because there exists a tradeoff between depth and width: with just one layer, we need an extreme width to yield a sufficiently good approximation. The exact size, however, depends on this tradeoff between depth and width, which in turn is very task-dependent. As a rule of thumb, for a fixed width, we should go deeper as long as computational resources allow the operation. To see this, we re-examine the relationship between training error, testing error and model complexity. Recall in Section 3.5, when we first discussed these concepts, we claimed the errors and model complexity have the relationships described by the dark blue lines in Figure 27.

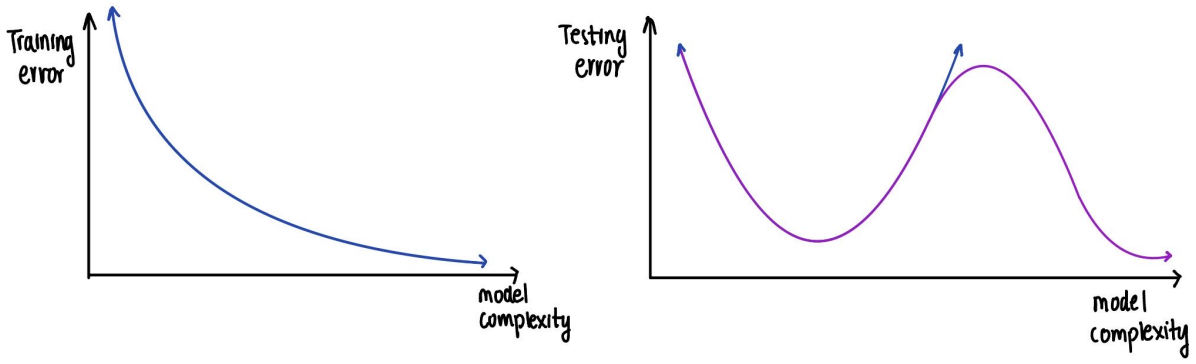


FIGURE 27. Neural Network: Training and Testing Error

(Note that the graphs in Figure 16 are reversed because higher λ's lead to simpler models.)

We now claim that, as we increase the size of the neural network, the inverse relationship for the training error still holds but not the parabolic relationship for the testing error. Recall that the testing error would eventually increase after the optimal point due to an overfitting problem. In this setting, however, overfitting no longer hurts: if the testing error does increase after a certain point, it would start to decrease again upon adding in sufficient layers. Why? The intuition is that we are working with a highly non-convex structure which admits many local maximizers found by stochastic gradient descent. As the size of the neural network increases, the landscape of the objective function becomes better. In other words, overparametrization makes the model flatter, and we are less likely to 'get stuck' in a bad local. Thus overfitting actually facilitates the convergence procedure, and this phenomenon is known as *benign overfitting (benign overparametrization)*. There are still a lot of local optimizers left, but it happens that solutions found by gradient descent has the property of *implicit regularization*, i.e. among the local optimizers that converge, the algorithm would always try to output the model with the simplest structure. Let us see this in an example.

Example 4.3. Let us consider a problem we have encountered before in ridge regression:

$$\min_{\beta} \|y - x\beta\|^2,\ x \in \mathbb{R}^{n\times p}, p > n.$$

There exists infinitely many β's such that $y = x\beta$. Which one should we choose? It turns out if we use gradient descent

$$\beta^{t+1} = \beta^t - \eta x^T(x\beta^t - y)$$

with the initial condition $\boldsymbol{\beta}^0 = 0$, we have, for a sufficiently small step size,

$$\beta^t \overset{t\to\infty}{\longrightarrow} \operatorname*{argmin}_{\beta}\{\|\beta\|^2 : y = x\beta\},$$

the optimizer with the least complexity in terms of ℓ_2-norm. We denote this optimizer as $\tilde{\beta}$.

(As an aside, the initial condition we posed above is not entirely accurate. Neural networks have a specific way of initialization, starting not at 0 but somewhere close, with each parameter sampled from a normal distribution with extremely low variance.)

Next, let us turn to the explicit regularization in ridge regression through adding the ℓ_2-penalty:

$$\widehat{\beta}_\lambda = \operatorname*{argmin}_{\beta} \left(\|y - x\beta\|^2 + \lambda\|\beta\|^2\right) = \left(x^T x + \lambda I_p\right)^{-1} x^T y.$$

It turns out that

$$\tilde{\beta} = \lim_{\lambda\to 0} \widehat{\beta}_\lambda,$$

or the optimizer for the 'ridgeless regression'. Hence there exists a relation (correspondence) between the implicit regularization in gradient descent versus the explicit regularization in ridge regression.

Note that the concept of implicit regularization is not unique to gradient descent. It is applicable to other algorithms like the coordinate descent that we mentioned before. In fact, we can show that similar to the above example, the implicit regularization in coordinate descent matches the solution implied by the ℓ_1-norm.

Now we transition from classification models to generative models.

4.2. **Generative Models.**

Let us start with some examples of generative modelling to see its difference from classification models:

- compose music;
- text $\to$ text (e.g. ChatGPT, Google translation, etc.);
- text $\to$ image (e.g. DALL-E).

To get an idea about how generative modeling work, we start with a simple one that we have worked with previously: $N(\theta, 1)$. The task is that, given a set of training data $x_1, ..., x_n \overset{\text{iid}}{\sim} N(\theta, I_p)$, we want to generate more similar data points. In our previous approaches, we first estimate θ^* and then take samples from $N(\theta^*, I_p)$. However, this method is not so readily generalizable to other

scenarios, since we may be working with many parameters in a complicated model, which makes even rough estimation difficult. Thus we need a new approach, which proceeds roughly as follows:

(1) We propose a value for the parameter $\theta \in \mathbb{R}^p$ and use the resulting distribution $N(\theta, I_p)$ to generate $\tilde{x}_1$, $\tilde{x}_2$, ..., $\tilde{x}_n$.
(2) We try to distinguish the training data $x_1, ..., x_n$ from the generated data $\tilde{x}_1$, ..., $\tilde{x}_n$.
(3) If we can indeed tell them apart, this means the proposed estimator is not a sufficiently good approximation of the original one, and our attempt was a failure. We then propose a different θ and repeat steps (1) and (2). We terminate the process when differentiation is no longer possible.

Steps (1) and (2) imply the need for two models: a generative one and a discriminative one. For our simple model, we can simply do the following:

(1) Let $z_1, ..., z_n \overset{\text{iid}}{\sim} N(0, I_p)$. Then define
$$\tilde{x}_i = G_\theta(z_i) = z_i + \theta.$$
(2) We use logistic regression with training data in the first sum and generated data the in the second:
$$\min_\theta \max_\beta V(\theta, \beta) = \min_\theta \max_\beta \left[\sum_{i=1}^n \log\left(\sigma(\beta^T x_i)\right) + \sum_{i=1}^n \log\left(1 - \sigma(\beta^T G_\theta(z_i))\right) \right].$$

The two steps combined yields an estimator $\widehat{\theta}$, which in this case equals $\overline{x}$, since we know it has to match the MLE. To solve for it by computation, however, we can just use the *gradient descent ascent* algorithm, first maximizing β by
$$\beta^{t+1} = \beta^t + \eta \nabla_p V(\theta^t, \beta^t),$$
then minimizing θ by
$$\theta^{t+1} = \theta^t - \alpha \nabla_\theta V(\theta^t, \beta^{t+1})$$
at each step. This converges with certainty for normal distribution, but the situation is much more complicated for other scenarios.

If we substitute in the neural networks, this is also a very simplified example of *GAN*, or *generative adversial network*, one of the most famous generative procedure currently invented. by Ian Goodfellow in 2014. The general idea is to replace the logistic regression with a neural network, the generative model with another, and then have the two compete against each other. A solution can be found by the *gradient descent ascent* algorithm, first maximizing β by
$$\beta^{t+1} = \beta^t + \eta \nabla_p V(\theta^t, \beta^t),$$
then minimizing θ by
$$\theta^{t+1} = \theta^t - \alpha \nabla_\theta V(\theta^t, \beta^{t+1})$$
at each step. When minimizing θ, the algorithm is implicitly trying to determine the coefficients in the entire structure. This converges with certainty for normal distribution, but the situation is much more complicated for other scenarios.

For more complex inputs and outputs, for example, images $\rightarrow$ images of cats, we can use a convolutional neural network (such that the output is not 1-dimensional) with similar procedure to above. The resulting network would still resemble the one in Figure 25.

5. Appendix I: Frequently Seen Distributions

Warning: Although double-checked with online resources, these results were manually calculated and potential errors exist. Readers should always verify before using.

5.1. Discrete.

Note: $\overline{X} = \sum X_i/n$.

1. **Bernoulli**(p).
PMF:
$$\begin{cases} \mathbb{P}(X=1) = p, \\ \mathbb{P}(X=0) = 1-p. \end{cases}$$
Expectation & Variance:
$$\mathbb{E}[X] = p,\ \mathbb{V}[X] = p(1-p).$$
MoM & MLE:
$$\hat{p}_{\text{MoM}} = \overline{X},\ \hat{p}_{\text{MLE}} = \overline{X}.$$
Fisher information:
$$I(p) = \frac{1}{p(1-p)}.$$

2. **Binomial**(m, p).
PMF:
$$\mathbb{P}(X=k) = \binom{m}{k} p^k (1-p)^{m-k}.$$
Expectation & Variance:
$$\mathbb{E}[X] = mp,\ \mathbb{V}[X] = mp(1-p).$$
MoM & MLE:
$$\hat{p}_{\text{MoM}} = \frac{\overline{X}}{m},\ \hat{p}_{\text{MLE}} = \frac{\overline{X}}{m}.$$
Fisher information:
$$I(p) = \frac{m}{p(1-p)}.$$

3. **Poisson**(λ).
PMF:
$$\mathbb{P}(X=k) = e^{-\lambda}\frac{\lambda^k}{k!}.$$
Expectation & Variance:
$$\mathbb{E}[X] = \lambda,\ \mathbb{V}[X] = \lambda.$$
MoM & MLE:
$$\hat{p}_{\text{MoM}} = \overline{X},\ \hat{p}_{\text{MLE}} = \overline{X}.$$
Fisher information:
$$I(p) = \frac{1}{\lambda}.$$

4. **Geometric**(p).
PMF:
$$\mathbb{P}(X=k) = p(1-p)^{k-1}.$$
Expectation & Variance:
$$\mathbb{E}[X] = \frac{1}{p},\ \mathbb{V}[X] = \frac{1-p}{p^2}.$$
MoM & MLE:
$$\hat{p}_{\text{MoM}} = \frac{1}{\overline{X}},\ \hat{p}_{\text{MLE}} = \frac{1}{\overline{X}}.$$
Fisher information:
$$I(p) = \frac{1}{p^2(1-p)}.$$

5. **Negative Binomial**(r, k).
PMF:
$$\mathbb{P}(X=k) = \binom{k-1}{r-1} p^r (1-p)^{k-r}.$$
Expectation & Variance:
$$\mathbb{E}[X] = \frac{r}{p},\ \mathbb{V}[X] = \frac{r(1-p)}{p^2}.$$

5.2. **Continuous.**
Note: $\overline{X} = \sum X_i/n$, $\overline{X^{(2)}} = \sum X_i^2/n$.

1. **Uniform**(a, b).
PDF:
$$f_X(x) = \frac{1}{b-a}\mathbf{1}_{(a,b)}.$$
CDF:
$$F_X(x) = \frac{x-a}{b-a}\mathbf{1}_{(a,b)} + 1 \cdot \mathbf{1}_{(b,\infty)}.$$
Expectation & Variance:
$$\mathbb{E}[X] = \frac{a+b}{2},\ \mathbb{V}[X] = \frac{(b-a)^2}{12}.$$
MoM:
$$\hat{a}_{\text{MoM}} = \overline{X} - \sqrt{3(\overline{X^{(2)}} - (\overline{X})^2)},$$
$$\hat{b}_{\text{MoM}} = \overline{X} + \sqrt{3(\overline{X^{(2)}} - (\overline{X})^2)}.$$
MLE:
$$\hat{a}_{\text{MLE}} = \min_i\{X_i\},\ \hat{b}_{\text{MLE}} = \max_i\{X_i\}.$$
Fisher information: DNE.[a]

2. **Gaussian**(μ, σ^2).
PDF:
$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$
CDF:
$$F_X(x) = \Phi(x) \text{ or } \operatorname{erf}(x).$$
Expectation & Variance:
$$\mathbb{E}[X] = \mu,\ \mathbb{V}[X] = \sigma^2.$$
MoM:
$$\hat{\mu}_{\text{MoM}} = \overline{X},\ \hat{\sigma^2}_{\text{MoM}} = \overline{X^{(2)}} - (\overline{X})^2.$$
MLE:
$$\hat{\mu}_{\text{MLE}} = \overline{X},\ \hat{\sigma^2}_{\text{MLE}} = \overline{X^{(2)}} - (\overline{X})^2.$$
Fisher information:
$$I(\mu) = \frac{1}{\sigma^2},\ I(\sigma^2) = \frac{1}{2(\sigma^2)^2}.$$

Other facts: If $Z \sim N(0,1)$,
$$\mathbb{E}[Z^i] = \begin{cases} 0, & i \text{ odd}, \\ (2m-1)(2m-3)\cdots, & i = 2m \text{ even}. \end{cases}$$

3. **Exponential**(λ).
PDF:
$$f_X(x) = \lambda e^{-\lambda x}, x \geq 0.$$
CDF:
$$F_X(x) = 1 - e^{-\lambda x}, x \geq 0.$$
Expectation & Variance:
$$\mathbb{E}[X] = \frac{1}{\lambda},\ \mathbb{V}[X] = \frac{1}{\lambda^2}.$$
MoM & MLE:
$$\hat{\lambda}_{\text{MoM}} = \frac{1}{\overline{X}},\ \hat{\lambda}_{\text{MLE}} = \frac{1}{\overline{X}}.$$
Fisher information:
$$I(\lambda) = \frac{1}{\lambda^2}.$$

4. **Gamma**(α, λ).
PDF:
$$f_X(x) = \begin{cases} \frac{\lambda e^{-\lambda x}(\lambda x)^{\alpha-1}}{\Gamma(\alpha)}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$
where $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$.
CDF:
$$F_X(x) = \frac{\gamma(\alpha, \lambda x)}{\Gamma(\alpha)},$$
where γ is the incomplete gamma function.
Expectation & Variance:
$$\mathbb{E}[X] = \frac{\alpha}{\lambda},\ \mathbb{V}[X] = \frac{\alpha}{\lambda^2}.$$
MoM:
$$\hat{\lambda}_{\text{MoM}} = \frac{\overline{X}}{\overline{X^{(2)}} - (\overline{X})^2},\ \hat{\alpha}_{\text{MoM}} = \frac{(\overline{X})^2}{\overline{X^{(2)}} - (\overline{X})^2}.$$
MLE: if α known,
$$\hat{\lambda}_{\text{MLE}} = \frac{\alpha}{\overline{X}}.$$
Fisher information:
$$I(\lambda) = \frac{\alpha}{\lambda^2}.$$

[a] In fact, any distribution whose support depends on the estimated parameters does not have an associated $I(\theta)$.

$\hat{\alpha}_{\text{MLE}}$ and $I(\alpha)$ involves digamma function.
Other facts:

$$\sum \text{Gamma}(\alpha_i, \lambda) = \text{Gamma}\left(\sum \alpha_i, \lambda\right).$$

If $X \sim \text{Gamma}(\alpha, \lambda)$,

$$\mathbb{E}\left[\frac{1}{X}\right] = \frac{\lambda}{\alpha - 1}, \mathbb{E}\left[\frac{1}{X^2}\right] = \frac{\lambda^2}{(\alpha-2)(\alpha-1)}.$$

5. **Beta**(α, β).
PDF:

$$f_X(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \ 0 \leq x \geq 1.$$

CDF:

$$F_X(x) = I_x(\alpha, \beta),$$

the regularized incomplete beta function.
Expectation & Variance:

$$\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}, \ \mathbb{V}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

MoM:

$$\hat{\alpha}_{\text{MoM}} = \overline{X}\left[\frac{\overline{X}(1-\overline{X})}{\overline{X^{(2)}}} - 1\right],$$

$$\hat{\beta}_{\text{MoM}} = (1-\overline{X})\left[\frac{\overline{X}(1-\overline{X})}{\overline{X^{(2)}}} - 1\right].$$

MLE: involves digamma function.

5.3. Multidimensional/Others.

1. **Chi-Square**(n).
Definition: if $Z_1, ..., Z_n \overset{\text{iid}}{\sim} N(0,1)$, then

$$\sum_{i=1}^{n} Z_i^2 \sim \chi_n^2 = \text{Gamma}\left(\frac{n}{2}, \frac{1}{2}\right).$$

Expectation & Variance:

$$\mathbb{E}[X] = n, \ \mathbb{V}[X] = 2n.$$

Other facts:

$$(n-1)\frac{S^2}{\sigma^2} \sim \chi_{n-1}^2.$$

2. t**-Distribution**(n).
Definition: if $Z_1, ..., Z_n \overset{\text{iid}}{\sim} N(0,1)$, then

$$t_n = Z/\sqrt{V/n},$$

where $V \sim \chi_n^2$.
Expectation: $\mathbb{E}[X] = 0$ if $n > 1$, else undefined.
Variance:

$$\mathbb{V}[X] = \frac{n}{n-2}.$$

Other facts:

$$\sqrt{n}\frac{\overline{X} - \mu}{S^2} t_{n-1}.$$

3. **Dirichlet**$(\vec{\alpha})$.
PDF:

$$f(\vec{x}; \vec{\alpha}) = \frac{1}{B(\vec{\alpha})} \prod_{i=1}^{n} x_i^{\alpha_i - 1}.$$

Expectation & Variance:

$$\mathbb{E}[X_i] = \frac{\alpha_i}{\alpha_0}, \ \mathbb{V}[X_i] = \frac{\tilde{\alpha}_i(1-\tilde{\alpha}_i)}{\alpha_0 + 1},$$

where $\tilde{\alpha}_i = \alpha_i/\alpha_0$, $\alpha_0 = \sum \alpha_k$.

4. **Hypergeometric**(n).
PMF:

$$\mathbb{P}(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}.$$

Expectation & Variance:

$$\mathbb{E}[X] = n \cdot \frac{K}{N}, \ \mathbb{V}[X] = n \cdot \frac{K}{N} \cdot \frac{N-K}{N} \cdot \frac{N-n}{N-1}.$$

5. **Multinomial**$(n, \vec{p})$.
PMF:

$$\mathbb{P}(X_1 = k_1, ..., X_n = k_n) = \binom{n}{k_1, ..., k_n} p_1^{k_1} \cdots p_n^{k_n}.$$

Expectation & Variance:

$$\mathbb{E}[X_i] = np_i, \ \mathbb{V}[X_i] = np_i(1-p_i).$$

6. **Multivariate Normal**$(\vec{\mu}, \vec{\Sigma})$.

PDF:

$$f_X(\vec{x}) = \frac{1}{\sqrt{(2\pi)^n |\vec{\Sigma}|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T \vec{\Sigma}^{-1}(\vec{x}-\vec{\mu})\right).$$

Linear transformation:

$$Y = \vec{A}\vec{X} \sim \text{MVN}(\vec{A}\vec{\mu}, \vec{A}\vec{\Sigma}\vec{A}^T).$$

Condition distribution: if

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy} & \Sigma_{yy} \end{pmatrix}\right),$$

then

$$X|Y \sim N\left(\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(Y-\mu_y),\, \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}\right).$$

6. Appendix II: CI - Wald, Wilson & VST

(For Wilson, we only put the inequality that needs to be solved to find the CI. For VST, we try to put both the transformation function $g(\cdot)$ and the CI when applicable. but only the former when things are messy.)

1. **Bernoulli**(p).

Wald:

$$\left[\overline{X} - z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\overline{X}(1-\overline{X})}{n}},\, \overline{X} + z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\overline{X}(1-\overline{X})}{n}}\right]$$

Wilson: solve

$$\left(\overline{X} - p\right)^2 \leq \frac{z_{1-\frac{\alpha}{2}}^2}{n} \cdot p(1-p)$$

VST: $g(p) = 2\arcsin(\sqrt{p})$.

$$\left[\sin^2\left(\arcsin\left(\sqrt{\overline{X}}\right) - \frac{z_{1-\frac{\alpha}{2}}}{2\sqrt{n}}\right),\, \sin^2\left(\arcsin\left(\sqrt{\overline{X}}\right) + \frac{z_{1-\frac{\alpha}{2}}}{2\sqrt{n}}\right)\right]$$

2. **Poisson**(λ).

Wald:

$$\left[\overline{X} - z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\overline{X}}{n}},\, \overline{X} + z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\overline{X}}{n}}\right]$$

Wilson: solve

$$\left(\overline{X} - \lambda\right)^2 \leq z_{1-\frac{\alpha}{2}}^2 \cdot \frac{\lambda}{n}$$

VST: $g(\lambda) = 2\sqrt{\lambda}$.

$$\left[\overline{X} - \frac{z_{1-\frac{\alpha}{2}}}{2\sqrt{n}},\, \overline{X} + \frac{z_{1-\frac{\alpha}{2}}}{2\sqrt{n}}\right]$$

3. **Geometric**(p). $(\widehat{p} = \widehat{p}_{\text{MLE}})$

Wald:

$$\left[\widehat{p} - z_{1-\frac{\alpha}{2}}\widehat{p} \cdot \sqrt{\frac{1-\widehat{p}}{n}},\, \widehat{p} + z_{1-\frac{\alpha}{2}}\widehat{p} \cdot \sqrt{\frac{1-\widehat{p}}{n}}\right]$$

Wilson: solve

$$(\widehat{p}-p)^2 \le z_{1-\frac{\alpha}{2}} \cdot \frac{p^2(1-p)}{n}$$

VST:

$$g(p) = \ln\left(\left|\sqrt{1-p}-1\right|\right) - \ln\left(\sqrt{1-p}+1\right).$$

4. **Gaussian**(σ^2).

Wald:

$$\left[S^2\left(1 - z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{2}{n-1}}\right),\, S^2\left(1 + z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{2}{n-1}}\right)\right]$$

Wilson: solve

$$\left(S^2-\sigma^2\right)^2 \le \frac{2z_{1-\frac{\alpha}{2}}^2\sigma^4}{n-1}$$

VST: $g(\sigma^2) = \sigma^4/\sqrt{2}$.

$$\left[\left(S^2\right)^2 - z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{2}{n}},\, \left(S^2\right)^2 - z_{1+\frac{\alpha}{2}} \cdot \sqrt{\frac{2}{n}}\right]$$

5. **Exponential**(λ). $(\widehat{\lambda} = \widehat{\lambda}_{\mathrm{MLE}})$

Wald:

$$\left[\widehat{\lambda} - z_{1-\frac{\alpha}{2}} \cdot \frac{\widehat{\lambda}}{\sqrt{n}},\, \widehat{\lambda} + z_{1-\frac{\alpha}{2}} \cdot \frac{\widehat{\lambda}}{\sqrt{n}}\right]$$

Wilson: solve

$$\left|\widehat{\lambda}-\lambda\right| \le z_{1-\frac{\alpha}{2}} \cdot \frac{\lambda}{\sqrt{n}}$$

or

$$\left[\frac{\widehat{\lambda}}{1+\frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}},\, \frac{\widehat{\lambda}}{1-\frac{z_{1-\frac{\alpha}{2}}}{\sqrt{n}}}\right]$$

VST: $g(\lambda) = \log(\lambda)$.

$$\left[\widehat{\lambda}e^{-\frac{z_{1-\frac{\alpha}{2}}}{n}},\, \widehat{\lambda}e^{\frac{z_{1-\frac{\alpha}{2}}}{n}}\right]$$