Power Analysis

G. de los Campos

11/16/2022

1) Hypothesis Testing

Any Hypothesis testing problem involves the following elements:

- A statistical model.
- A null (H_0) and an alternative (Ha) hypothesis.
- A test statistic.
- A decision rule. This rule establishes when to reject/do not reject H0.

The test statistic is any function of the data that that allows us to quantify evidence against H_0.

Examples of test statistics we have considered are:

- F-statistic.
- Chi-square statistic (we used it in likelihood ratio test and in Wald's test).
- The t-statistic.
- pvalues (pvalues quantify the probabillity of obtaining a test statistic as extreme or more extreme than the one that we obtained if H_0 holds).

Note: For any power/type-I erro rate analys, be sure you identify each of the elements above described, this will be critical for any analysis. Take time to identify each of those elements carefully.

2) Error types and error rates in hypothesis testing

In hypothesis testing we use a decision rule (e.g., reject if |t-statistic|>1.96) to reject/do-not reject H_0. In the simplest case we have two possible states of nature (H_0 and H_a) and two possible decisions (reject/do not reject); the table below classifies each of these cases

	Do not reject H_0	Reject H_0
H_0 holds	True Negative	False Positive
H_a holds	False Negative	True positive

Types of error: In the table above there are two decisions that are incorrect: the False Positives (also called Type-I errors) the False Negatvies (called the Type-II errors).

Distribution of the Decision Rule over conceptual repeated sampling

Suppose we repeat the experiment a large number of times, each time collecting data, evaluating the test-statistic, and rejecting or not. Imagine we know whether H_0 or H_a holds and we count how many TN (N1), FP (N2), FN (N3) and TP (N4) we get,

	Do not reject H_0	Reject H_0
H_0	N1	N2
H_a	N3	N4

If H_0 holds (first row in the above-table), the false discovery proportion is: N2/(N1+N2)

The **type-I error rate** is the probability of rejecting the null given that the null is true, that is Type-I Error Rate= $p(rejecting\ H_0|H_0$ holds)= E[N2/(N1+N2)].

The **power** of an experiment is the probability of rejecting the null given that the alternative hypothesis holds, that is Power= $p(rejecting\ H_0|H_a$ holds)= E[N4/(N3+N4)].

The **type-II error** is the probability of failing to reject H_0 given that H_a holds.

3.1) Why using p-values?

P-values quantify the probability of rejecting the null when the null holds, that is

- $pvalue = P(reject\ H_0|H_0$ holds)

Since the $pvalue$ quantifies the type-I error rate, it follows that if we use pvalues as our test-statistic and we reject if $pvalue < \alpha$, then, if the pvalues are correct, the type-I error rate will be controlled at a level $< \alpha$. A decision rule that rejects if $pvalue < 0.05$ aims at controlling the type-I error rate at a low level (< 0.05).

It turns out that, if the pvalues are correct, using pvalues also maximizes power among the decision rules that control the type-I error rate at the $\alpha-$level.

3.2) Estimating Type-I erro rate using Monte Carlo Methods

Consider a linear model of the form $y = \mu + x\beta + \varepsilon$ and let's assume that $H_0 : \beta = 0$ holds.

How often do we expect to reject the null if our decision rule is reject if $pvalue < 0.05$?

Because p-values quantify the expected Type-I error rate, if w reject whenever $pvalue < 0.05$ the expected type-I erro rate is expected to be smaller than 5%.

This is illustrated in the following example:

Example 1: Estimating Type-I Error rate using Monte Carlo Methods

```r
set.seed(12345)
n=10
mu=10

nReps=10000 # number of Monte Carlo replicates
reject=rep(NA,nReps)

for(i in 1:nReps){
  y=rnorm(n,mean=mu,sd=1)
  x=runif(n)
  fm=lm(y~x)
  pvalue=summary(fm)$coef[2,4]
  reject[i]=pvalue<0.05
}

table(reject)
```

```
## reject
## FALSE  TRUE
##  9484   516
```

```r
mean(reject)
```

```
## [1] 0.0516
```

Above, the pvalues are *correct* because the assumptions used to derive it (errors are iid normal) hold.

3.3) Type-I error rate when the assumptions used to derive p-values do not hold

What if the assumptions do not hold?

For instance, what would be the rejection rate if the errors are exponentially distributed?

Example 2: Type-I Error rate when some assumptions are violated

```r
set.seed(12345)
nReps=50000 # number of Monte Carlo replicates
n=10
reject=rep(NA,nReps)

for(i in 1:nReps){
  y=rexp(n,rate=1/mu)
  x=runif(n)

  fm=lm(y~x)
  pvalue=summary(fm)$coef[2,4]
  reject[i]=pvalue<0.05
}

table(reject)
```

```
## reject
## FALSE  TRUE
## 47526  2474
```

```r
mean(reject)
```

```
## [1] 0.04948
```

We can see that the linear model is quite robust even under an important violoation of assumptions about the distribution of the error terms.

However,other violations are more consequential.

3.4) Model miss-specification

Suppose that you want to test whether x has an effect on y using a model

$$y_i = \mu + x_i\beta + \varepsilon_i$$

Now, assume that the true model

$$y_i = \mu + z_i\beta + \varepsilon_i$$

where z_i is a covariate that is correlated with x_i; however, you may have not measure that covariate or you don't know that z_i affects y_i; therefore, you test for the effect of x_i on y_i using

$$y_i = \mu + x_i\beta + \varepsilon_i$$

The following code shows that the error rate is much higher than the significance used for rejection. This happens because of model-miss-specification, when z_i is not included in the model, x_i captures partially the effect of the left-out variable.

Example 3: Type-I Error rate when the model is misspecified

```
N=100
nRep=10000 # number of Monte Carlo replicates

pValues=rep(NA,nRep)
N0=floor(N/2)
bZ=.1

for(i in 1:nRep){
    error=rnorm(n=N)
    z=rnorm(N)
    x=rnorm(N)+z
    signal=z*bZ
    y=signal + error
    fm=lm(y~x)
    pValues[i]=summary(fm)$coef[2,4]

}

reject=pValues<.05 # decision rule
mean(reject,na.rm=T)
```

```
## [1] 0.1052
```

Including z_i in the model fixes the problem.

Example 4: Type-I Error rate when the model is correctly specified

```
N=50
nRep=10000 # number of Monte Carlo replicates

pValues=rep(NA,nRep)
N0=floor(N/2)
bZ=.1


for(i in 1:nRep){
    error=rnorm(n=N)  # this violates the normality assumption
    z=rnorm(N)
    x=rnorm(N)+z
    signal=z*bZ
    y=signal + error
    fm=lm(y~z+x)
```

```
      pValues[i]=summary(fm)$coef[3,4]

  }

  reject=pValues<.05 # decision rule
  mean(reject,na.rm=T)
```

```
## [1] 0.0516
```

Note:

- Violations about the distribution of error terms are usually not that serious and the consequences of such violations of assumptions diminish with sample size (this is an intance of the Cetnral Limit Theorem).
- However model miss-specification (e.g., omitting relevant covariates) can lead to bias in estimates and inadequate error control, and this consequences do not vanish with large sample size. Indeed, the consequences of such misspecification are even more important with large sample size.

4) Estimating Power using Monte Carlo Methods

The structure of a Monte Carlo simulation to estimate power or type-I erro rate is very similar:

(i) We simulate data
(ii) We fit the model and extract the relevant test-statistic
(iii) We decide whether to reject or not.

We repeate steps (i)-(iii) a large number of times and keep track of how many times we rejected or don't rejected.

The estimated rejection rate is the proportion of times we rejected.

If in (i) we simulated data under H_0, then the proportion of times we rejected is an estimate of the Type-I erro rate.

On the other hand, if in (i) data was simualted under H_a, the empricial rejection rate is an estimate of the Power of the analysis.

Example 5: Estimating Power using Monte Carlo Methods

The following code was used in Example 1 to estimate type-I error rate, here we modify one line in the code to simulate data for H_a instead of H_0; therfore, the empirical rejection rate is an estimate of the power of the study.

```
set.seed(12345)
n=10
mu=10

nReps=10000 # number of Monte Carlo replicates
reject=rep(NA,nReps)

# size of effect

b=1

for(i in 1:nReps){
  y=rnorm(n,mean=mu,sd=1)

  x=runif(n)
  y=x*b+y # Adding the signal....

  fm=lm(y~x)
  pvalue=summary(fm)$coef[2,4]
  reject[i]=pvalue<0.05
}

table(reject)
```

```
## reject
## FALSE  TRUE
##  8789  1211
```

```
mean(reject)
```

```
## [1] 0.1211
```

The estimated power is 0.12.

The tow major factors affecting power are:

- Sample size (power increases with sample size)
- The signal-to-noise ratio, or, more specifically the size of effect relative to the SD of the errors.

In the aobve-example, the effect was sizable (1 which was equal to the SD of the errors); however, sample size is too small to achieve high power.

In the INCLASS-assigment we will explore the effect of sample size and effect size on power.