MS&E 226: Fundamentals of Data Science

Lecture 12: Beyond single hypothesis tests

Ramesh Johari

Multiple hypothesis testing

An example: Multiple linear regression

Suppose that I have n rows of data with outcomes $\mathbf{Y}$ and corresponding covariates $\mathbf{X}$. Suppose $p = 100$.

I run a linear regression with all the covariates and check statistical significance. I order the resulting covariates in descending order of p-value:

Covariate index	p-value
40	0.0070
58	0.018
93	0.034
69	0.040
57	0.042
10	0.047

You walk away excited: these six coefficients are all significant at the 95% level, and you now have a starting point for building your model.

An example: Multiple linear regression

In fact: *There is no relationship in this data between* $\mathbf{X}$ *and* $\mathbf{Y}$*!*

I used synthetic data to generate this example, with:

- $Y_i \sim \mathcal{N}(0, 1)$ for each i, i.i.d.
- $X_{ij} \sim \mathcal{N}(0, 1)$ for each i, j, i.i.d.

So what happened?

What happened?

Recall the p-value is the answer to the following question:

> *What is the chance I would see an estimated coefficient (from the data) as extreme as what I found, if the true coefficient was actually zero?*

What happened?

Recall the p-value is the answer to the following question:

> *What is the chance I would see an estimated coefficient (from the data) as extreme as what I found, if the true coefficient was actually zero?*

If we use a cutoff of 0.05 to determine whether a coefficient is "statistically significant", then we are willing to accept a 5% rate of *false positives*: coefficients that look large due to random chance, despite the fact that there is really no underlying relationship.

This means with 100 covariates, we should expect 5 of the coefficients to be significant due to random chance alone – even if there is no effect there! (In our case we get slightly more than this.)

Multiple hypothesis testing

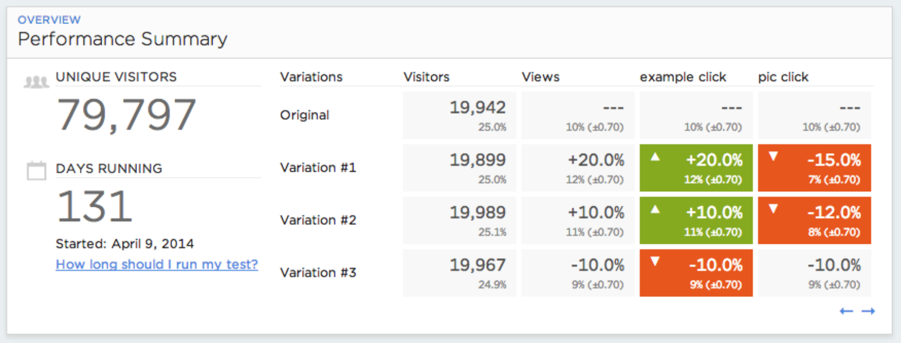
This is a systematic issue with statistical significance based on p-values from individual hypothesis tests:

If you use a cutoff of 0.05 (or 0.01, etc.), you should expect 5% (or 1%, etc.) of your discoveries (rejections) to be false positives.

This applies across all hypothesis tests you do: so for example, if you use a 5% cutoff every day at your job on every test you ever run, you will generate false positives in 5% of your hypothesis tests.

Example: The Optimizely results page

Platforms such as Optimizely enable “A/B testing” (i.e., randomized experimentation) of different website’s designs against each other. Here is a typical results page for an experiment:



OVERVIEW

Performance Summary

UNIQUE VISITORS

79,797

DAYS RUNNING

131

Started: April 9, 2014

How long should I run my test?

Variations	Visitors	Views	example click	pic click
Original	19,942 25.0%	--- 10% (±0.70)	--- 10% (±0.70)	--- 10% (±0.70)
Variation #1	19,899 25.0%	+20.0% 12% (±0.70)	▲ +20.0% 12% (±0.70)	▼ -15.0% 7% (±0.70)
Variation #2	19,989 25.1%	+10.0% 11% (±0.70)	▲ +10.0% 11% (±0.70)	▼ -12.0% 8% (±0.70)
Variation #3	19,967 24.9%	-10.0% 9% (±0.70)	▼ -10.0% 9% (±0.70)	-10.0% 9% (±0.70)

Real dashboards can have many more simultaneous hypothesis tests present.

Multiple hypothesis testing

Is this a problem? Perhaps not: if you understand that false positives are generated in this way, you can be wary of overinterpreting significance with many hypothesis tests.

The problem is that interpretation of the results becomes much harder: which results are “trustworthy”, and which are “spurious”?

Multiple testing corrections

Multiple testing corrections provide a systematic approach to identifying "meaningful" effects when dealing with many simultaneous hypothesis tests.

This has been an active area of work in the last several decades in statistics, as the range of applications where many hypothesis tests are possible has increased.

Notation

We will discuss multiple testing corrections in the context of the OLS regression example.

As before, we will assume that Assumptions (A1)-(A3) hold. Let $H_0^{(j)} : \beta_j = 0$ denote the null hypothesis that coefficient j is zero.

If n is "large", then the p-value on feature j in the regression table is the chance of seeing a corresponding t statistic as extreme as observed, if $H_0^{(j)}$ is true.

Bonferroni correction

The simplest example of a multiple testing correction is the *Bonferroni* correction.

This approach tries to ensure that the probability of declaring even one false positive across all the hypothesis tests (also called the *familywise error rate*, FWER) is no more than, e.g., 5%.

Bonferroni correction

The simplest example of a multiple testing correction is the *Bonferroni* correction.

This approach tries to ensure that the probability of declaring even one false positive across all the hypothesis tests (also called the *familywise error rate*, FWER) is no more than, e.g., 5%.

The Bonferroni correction declares as significant (rejects the null) any coefficient j where the p-value is $\leq \alpha/p$, where p is the number of hypothesis tests being carried out.

In our example, $p = 100$ and $\alpha = 0.05$, so only coefficients with p-values ≤ 0.0005 are declared significant – *none* in the example I showed!

Bonferroni correction

Why does the Bonferroni correction work?

- For a collection of events $A_1, \ldots, A_p$, we have the following bound (called the *union bound*):

$$\mathbb{P}(A_1 \text{ or } A_2 \text{ or } \cdots A_p) \leq \sum_{j=1}^{p} \mathbb{P}(A_j).$$

Bonferroni correction

Why does the Bonferroni correction work?

- For a collection of events $A_1, \dots, A_p$, we have the following bound (called the *union bound*):

$$\mathbb{P}(A_1 \text{ or } A_2 \text{ or } \cdots A_p) \leq \sum_{j=1}^{p} \mathbb{P}(A_j).$$

- So now let A_j be the event that the p-value for coefficient j is less than or equal to α/p. Then if $H_0^{(j)}$ is true, we have:

$$\mathbb{P}(A_j | \beta_j = 0) \leq \frac{\alpha}{p}.$$

Bonferroni correction

Why does the Bonferroni correction work?

- For a collection of events $A_1, \ldots, A_p$, we have the following bound (called the *union bound*):

$$\mathbb{P}(A_1 \text{ or } A_2 \text{ or } \cdots A_p) \le \sum_{j=1}^{p} \mathbb{P}(A_j).$$

- So now let A_j be the event that the p-value for coefficient j is less than or equal to α/p. Then if $H_0^{(j)}$ is true, we have:

$$\mathbb{P}(A_j | \beta_j = 0) \le \frac{\alpha}{p}.$$

- Finally, suppose that all $H_0^{(j)}$ are true, i.e., the β_j's are in fact *all* zero. Then the probability at least one of the A_j's is true (i.e., at least one false positive) is $\le p \times \alpha/p = \alpha$.

Bonferroni correction in R

In R, the Bonferroni correction is easy to implement using the `p.adjust` function:

```
# Suppose you have a vector of p-values
pvalues <- c(0.001, 0.045, 0.0004, 0.025, 0.15)

# Apply Bonferroni correction
adjusted_pvalues <- p.adjust(pvalues, method = "bonferroni")

# Reject at level alpha = 0.05
alpha <- 0.05
reject <- adjusted_pvalues <= alpha
```

The function multiplies each unadjusted p-value by p (the number of tests), capping at 1.

Benjamini-Hochberg procedure

The Bonferroni correction works by essentially forcing your attention only on the smallest p-values (most significant results).

In practice, though, it can be too conservative, especially as the number of hypotheses (e.g., coefficients) increases.

Other methods have emerged to deal with this issue, to allow valid inference while being somewhat less conservative. We consider one, the *Benjamini-Hochberg (BH)* procedure.

Benjamini-Hochberg procedure: False discovery proportion

Suppose that S_0 is the set of coefficients where the null is in fact true:

$$S_0 = \{j : H_0^{(j)} \text{ is true}\} = \{j : \beta_j = 0\}.$$

Suppose that under a given decision procedure, you reject the null for the set of hypotheses in R.

The *false discovery proportion* (FDP) is the fraction of your rejections that were also in the null set:[1]

$$\text{FDP} = \frac{|S_0 \cap R|}{|R|}.$$

[1] FDP is defined to be zero if you make no rejections.

Benjamini-Hochberg procedure: False discovery rate

The *false discovery rate* (FDR) is the expected value of this fraction over the randomness of the data ("parallel universes"), given the true coefficients:

$$\mathrm{FDR} = \mathbb{E}_{\mathscr{D}}[\mathrm{FDP} | \beta_1, \dots, \beta_p],$$

where $\beta_j = 0$ for $j \in S_0$. (Here $\mathscr{D}$ represents your data sample used for hypothesis testing.)

The BH procedure ensures FDR is less than or equal to α – regardless of which coefficients are null or not!

Intuition for false discovery rate

Consider the Optimizely results page again.

Suppose we use BH with $\alpha = 0.05$. This ensures that on average, *of those cells that are declared significant*, we will have made mistakes on at most 5% of them.

The criterion is stronger than just controlling each individual test at $\alpha = 0.05$, but weaker than controlling the familywise error rate at $\alpha = 0.05$.

Benjamini-Hochberg procedure

The BH procedure at level α is simple to implement:

1. Compute p-values for each of your hypothesis tests, and order them in *increasing* order. Denote these by $q_{(1)}, q_{(2)}, \dots, q_{(p)}$.
2. Find the *largest* j such that: $q_{(j)} \leq \alpha j/p$.
3. Reject all hypotheses $1, \dots, j$.

As long as all hypothesis tests are independent of each other, this procedure ensures FDR $\leq \alpha$.[2]

Note: The Bonferroni correction would reject only those where $q_{(j)} \leq \alpha/p$ (a constant threshold).

[2] If the hypotheses are not independent, the same result holds if we change the right hand side to $\alpha j/(p \log p)$.

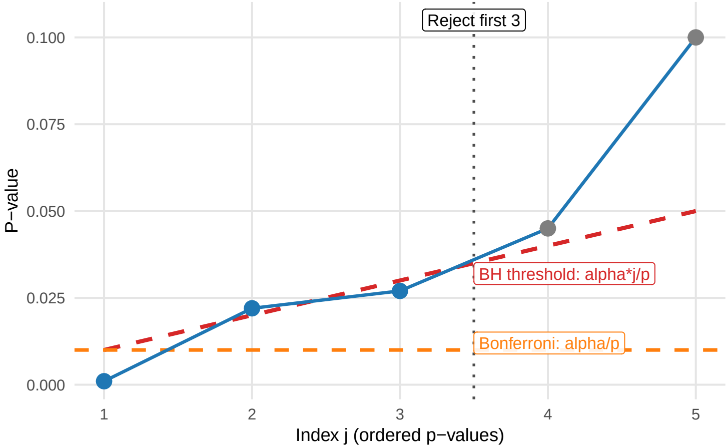
Benjamini-Hochberg: Numerical example

As a numerical example, suppose that we run BH at level $\alpha = 0.05$ with 5 hypothesis tests, where we assume the tests are independent, and we receive p-values $0.001, 0.045, 0.027, 0.022, 0.10$.

We first order the p-values from lowest to highest: $0.001, 0.022, 0.027, 0.045, 0.10$.

Since $\alpha/5 = 0.01$, we look for the largest j such that the j'th p-value in the ordered list is $\leq 0.01j$. This is 0.027, so we reject the three hypotheses with p-values $0.001, 0.022$, and 0.027.

Benjamini-Hochberg: Visual demonstration



The blue line shows ordered p-values. The red line is the BH threshold $\alpha j/p$. We reject hypotheses 1, 2, and 3 (blue dots below the BH threshold). (The orange dashed line is the Bonferroni threshold α/p.)

Benjamini-Hochberg in R

In R, the BH procedure is also easy to implement using `p.adjust`:

```
# Suppose you have a vector of p-values
pvalues <- c(0.001, 0.045, 0.027, 0.022, 0.10)

# Apply Benjamini-Hochberg correction
adjusted_pvalues <- p.adjust(pvalues, method = "BH")

# Reject at level alpha = 0.05
alpha <- 0.05
reject <- adjusted_pvalues <= alpha

# Which hypotheses are rejected?
which(reject)  # Returns: 1 3 4
```

Why use Benjamini-Hochberg?

The BH procedure is much less conservative than the Bonferroni correction, while still providing useful inference when many hypothesis tests are run.

You should have the habit of always using a procedure like BH when you run many hypothesis tests (e.g., testing many coefficients at once, or testing many exploratory hypotheses with the same data), to validate that your findings are actually meaningful.

Post-selection inference

Hypothesis testing in practice

Multiple testing corrections address one problem: choosing significant results among *many simultaneous tests*.

But this is not the only problem that can arise...In practice, testing many coefficients is often only the first step in a common data science pipeline:

- *Model selection*: Determine *which* features are "significant", i.e., which features to keep.
- *Inference*: Report standard errors, p-values, and confidence intervals for the resulting model.

Unfortunately, this practice is problematic if the *same* data is used for selection and inference!

Post-selection inference: The problem

Example: Suppose you:

1. Fit a regression with 10 variables
2. Keep only the variables with p-value < 0.10
3. Refit the model with just those variables
4. Report p-values and confidence intervals from the final model

Post-selection inference: The problem

Example: Suppose you:

1. Fit a regression with 10 variables
2. Keep only the variables with p-value < 0.10
3. Refit the model with just those variables
4. Report p-values and confidence intervals from the final model

What goes wrong? By selecting variables based on significance, you've *cherry-picked* a favorable sample. The p-values in step 4 are biased downward (i.e., too likely to lead you to reject the null) because they don't account for the selection in step 2.

Post-selection inference: The problem

Example: Suppose you:

1. Fit a regression with 10 variables
2. Keep only the variables with p-value < 0.10
3. Refit the model with just those variables
4. Report p-values and confidence intervals from the final model

What goes wrong? By selecting variables based on significance, you've *cherry-picked* a favorable sample. The p-values in step 4 are biased downward (i.e., too likely to lead you to reject the null) because they don't account for the selection in step 2.

This practice is also sometimes called "p-value hacking".

Sample splitting: A practical solution

The simplest solution to post-selection bias is *sample splitting*:

1. *Split your data* into two independent sets:
   - *Exploration set* (e.g., 50% of data): Use for model selection
   - *Confirmation set* (e.g., 50% of data): Use for inference
2. *Selection step*: Use the exploration set to choose your model (e.g., which variables to include, which transformations to use).
3. *Inference step*: Fit the selected model on the **confirmation set only**, and report p-values and confidence intervals from this fit.

Sample splitting: A practical solution

The simplest solution to post-selection bias is *sample splitting*:

1. *Split your data* into two independent sets:
   - *Exploration set* (e.g., 50% of data): Use for model selection
   - *Confirmation set* (e.g., 50% of data): Use for inference
2. *Selection step*: Use the exploration set to choose your model (e.g., which variables to include, which transformations to use).
3. *Inference step*: Fit the selected model on the **confirmation set only**, and report p-values and confidence intervals from this fit.

The confirmation set is *independent* of the selection process, so p-values and confidence intervals remain valid as long as the selected model satisfies the assumptions needed for valid inference; e.g., (A1)-(A3) for OLS.

Sample splitting: Connection to cross-validation

Sample splitting should feel familiar from our discussion of prediction:

- In *prediction*, we split data into training/test sets to evaluate model performance on unseen data.
- In *inference*, we split data into exploration/confirmation sets to ensure valid p-values and confidence intervals.

The underlying principle is the same: *data used for selection should be independent from data used for evaluation or inference*.

On the problem set, you will see post-selection inference in action, and use sample splitting to correct it.