6.S951 Modern Mathematical Statistics Fall 2024

Lecture 17 — November 14, 2024

Prof. Stephen Bates

Scribe: Chenyu Zhang

1 Outline

Agenda:

1. Setup for multiple hypothesis testing
2. Bonferroni Algorithm
3. Connection to sparsity

Last Time:

1. Setup for prediction inference
2. Conformal prediction
3. Risk-controlling prediction sets

2 Multiple Hypothesis Testing

Similar to the setup for single hypothesis testing, we consider data $\mathcal{X} \sim P_\theta$, where $\theta \in \Theta$. However, here we test multiple null hypotheses: $H_{0,i} : \theta \in \Theta_{0,i}$ for $i = 1, \ldots, n$. Here n is the number of hypotheses, and $\Theta_{0,i}$ are general subsets of Θ that may overlap. A straightforward example is that $H_{0,i}$ is a null hypothesis about the i-th coordinate of X, and $\Theta_{0,i}$ contains parameters with their i-th coordinate satisfying the null condition. We ask the following questions.

Questions

1. (**Global null testing**). Is $\cap_i H_{0,i}$ true? For example, when the null hypothesis is $H_{0,i} : \theta_i = 0$, the global null testing asks whether $\theta = 0$ holds.
2. Which $H_{0,i}$ are not true? We ask this because the effects of the alternative hypotheses are often of greater interest. Let the output of a multiple hypothesis test be the rejection set $R \subset \{1, \ldots, n\}$. We assess the following two error metrics:
    1. (**Family-wise error rate; FWER**). We want to return an R such that $\mathbb{P}(R \text{ contains any null}) \leq \alpha$
    2. **(False discovery rate)**. We want to return an R such that at most an α-fraction of the rejected hypotheses are null.

Example 1 (Gaussian)**.** *Consider Gaussian data $X \sim \mathcal{N}(\theta, I)$ with $\theta \in \mathbb{R}^d$. One example multiple testing problem is to test against $H_{0,i} : \theta_i = 0$.*

Example 2 (Genome-wide association studies)**.** *Given some disease status variable Y, we want to study the association of each gene with the disease. Consider $X \in \mathbb{R}^n$ with $n \approx 20,000$, where X_i is some gene. We test against $H_{0,i} : X_i$ is independent of Y.*

A key question in designing a multiple testing algorithm is how to combine the results of individual hypothesis tests to produce a coherent output. p-values serve as a convenient object to work with for this purpose. We denote p_i as the p-value for $H_{0,i}$, i.e., $P_\theta(p_i \leq t) \leq t$ for all $\theta \in \Theta_{0,i}$ and $t \in [0,1]$. Note that we only need the p-value to be super uniform; but sometimes we assume they are exactly uniform to obtain tight results. Figure 1 plots the sorted p-values for different signals. Specifically, when the null hypothesis is true, the p-values are uniformly distributed (no interesting signal); when the sorted p-values deviate significantly from the line $y = x$, it presents a clear signal. However, this signal does not directly translate into that all p-values below a certain threshold are significant. Because when there are many true nulls, some of their p-values will be small by chance. Thus, the observed signal suggests only a systematic departure from the null hypothesis rather than significance for each individual p-value.

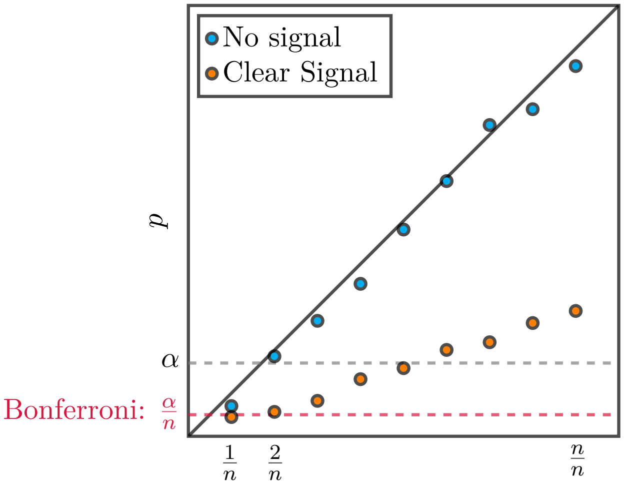

Figure 1: Sorted p-values

A multiple testing algorithm using p-values is of the form

$$A : [0,1]^n \to 2^{\{1,\dots,n\}} \cong \{0,1\}^n, \quad \vec{p} \mapsto R.$$

One simple algorithm of this kind is the Bonferroni algorithm.

Definition 3 (Bonferroni)**.** *Let $\alpha \in (0,1)$ be the family-wise error rate (FWER). Let $\{p_i\}_{i=1}^n$ be the p-values of individual tests. The Bonferroni algorithm returns*

$$A(\vec{p}) = \{i : p_i \leq \alpha/n\}.$$

Proposition 4 (FWER control for Bonferroni)**.** *The Bonferroni algorithm controls the FWER at level α.*

Proof. By definition, the FWER is

$$\mathbb{P}(\exists i : p_i \leq \alpha/n) = \mathbb{P}(\cup_{i\in\mathcal{N}}\{p_i \leq \alpha/n\}),$$

where $\mathcal{N} = \{1 \leq i \leq n : H_{0,i} \text{ is true}\}$. Then, by the union bound,

$$\mathbb{P}(\cup_{i\in\mathcal{N}}\{p_i \leq \alpha/n\}) \leq \sum_{i\in\mathcal{N}} \mathbb{P}(p_i \leq \alpha/n) \leq \sum_{i\in\mathcal{N}} \alpha/n \leq \alpha. \qquad \square$$

We remark that the Bonferroni algorithm works for dependent tests. Nonetheless, the following example on independent Gaussian helps us understand the algorithm.

Example 5 (Gaussian)**.** *Consider data $X \sim \mathcal{N}(\theta, I)$, null hypotheses $H_{0,i} : \theta_i = 0$, and one-sided p-values $p_i = 1 - \Phi(X_i)$. Then, the Bonferroni algorithm rejects $H_{0,i}$ if $p_i \le \alpha/n$, which is equivalent to $X_i \ge -\Phi^{-1}(\alpha/n)$. See Figure 2 for a visualization of how the Bonferroni algorithm controls the cumulative tail probability.*

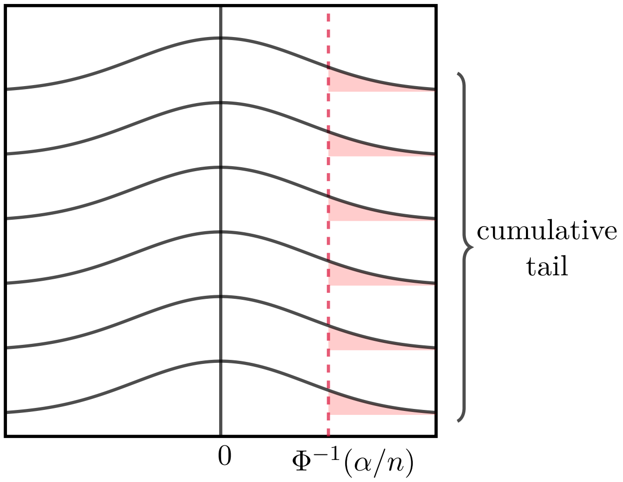

Figure 2: The Bonferroni algorithm on Gaussian data

The following proposition gives an approximation of $\Phi^{-1}(-\alpha/n)$ for any $\alpha \in (0, 1)$.

Proposition 6. *Let $Z_i \overset{iid}{\sim} \mathcal{N}(0,1), i = 1, \dots, n$. We have*

$$\frac{\max_i Z_i}{\sqrt{2 \log n}} \xrightarrow{P} 1, \quad \text{as } n \to \infty.$$

By the max-central limit theorem, we have

$$-\Phi^{-1}(\alpha/n) = \sqrt{2 \log n}(1 + o(1)),$$

where the asymptotic holds as $n \to \infty$.

Figure 3 gives an illustration of how $\sqrt{2 \log n}$ approximates $\Phi^{-1}(-\alpha/n)$ when $\alpha = 0.05$.

3 Sparsity Connection

The previous proposition already connects the threshold of the Bonferroni algorithm to the max statistic, which is good for detecting sparse signals. The following propositions further formalize the connection between the Bonferroni algorithm and sparsity, suggesting that the Bonferroni algorithm is good at detecting sparse signals and dealing with sparse alternatives.

Proposition 7. *If $\theta_1 = (1 + \epsilon)\sqrt{2 \log n}$ with $\epsilon \in (0, 1)$ and $\theta_i = 0$ for $i \ge 2$, then the Bonferroni algorithm has power*

$$\mathbb{P}(\underbrace{1 \in R = A(X)}_{\text{reject } H_{0,1}}) \to 1, \ \text{as } n \to \infty.$$

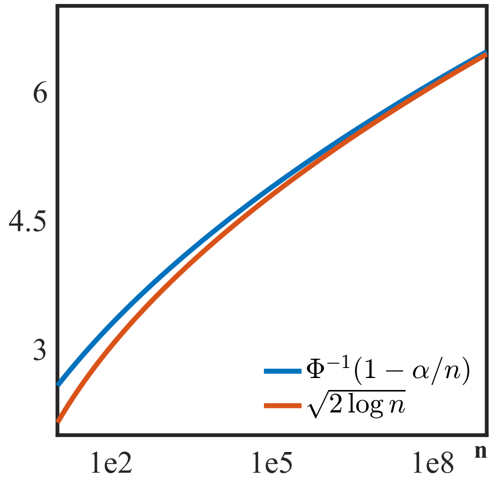

Figure 3: Approximation of $\Phi^{-1}(-\alpha/n)$.

Proof. First, by the definition of the Bonferroni algorithm, we have

$$\mathbb{P}(1 \in R) = \mathbb{P}(X_1 \geq -\Phi^{-1}(\alpha/n)) = \mathbb{P}(Z \geq -\Phi^{-1}(\alpha/n) - \theta_1),$$

where $Z \sim \mathcal{N}(0,1)$. Then, by Proposition 6, we get

$$\mathbb{P}(Z \geq -\Phi^{-1}(\alpha/n) - \theta_1) = \mathbb{P}(Z \geq (1 + o(1) - 1 - \epsilon)\sqrt{2\log n}).$$

Letting $n \to \infty$ gives

$$\mathbb{P}(1 \in \mathbb{R}) \to \mathbb{P}(Z \geq -\infty) = 1.$$

□

The following proposition can be obtained by a similar argument.

Proposition 8. *If $\theta_1 = (1-\epsilon)\sqrt{2\log n}$ with $\epsilon \in (0,1)$ and $\theta_i = 0$ for $i \geq 2$, then the Bonferroni algorithm has power approaching 0 as $n \to \infty$.*