STATS 300C: Theory of Statistics **Spring 2022**

Lecture 5 — April 12, 2022

Lecturer: Prof. Emmanuel Candès *Editor: Parth Nobel, Scribe: Michael Howes*

Warning: *These notes may contain factual and/or typographic errors. They are based on Emmanuel Candès's course from 2018 to 2022, Lihua Lei's lecture slides from 2022 and scribe notes written by Gene Katsevich, Andy Tsao, Yiguang Zhang and James Yang.*

5.1 The Closure Principle

Last time, we started talking about the *closure principle*. The setting was as follows. Suppose we have a family of hypotheses $\{H_i\}_{i=1}^n$. For a subset $I \subseteq \{1, \dots, n\}$ we defined the intersection hypothesis $H_I = \bigcap_{i \in I} H_i$. We then defined the *closure* of $\{H_i\}_{i=1}^n$ to be the collection $\{H_I : I \subseteq \{1, \dots, n\}\}$ [5]. We then assumed that for each I we had a level-α test φ_I of H_I. We then defined the *closure principle*.

Definition 1 (The Closure Principle)**.** Reject H_I if and only if for all $J \supseteq I$, H_J is rejected by the level-α test φ_I. Mathematically, let $T_I = \min_{J \supseteq I} \varphi_J$, then we reject H_I if and only if $T_I = 1$.

Last time, we saw that if φ_I has level α, then the closure principle controls the FWER at level α. Testing the hypotheses $H_1, \dots, H_n$ with the tests $T_{\{i\}}$ is called a *closed procedure*. One problem with closed procedures is that the computational complexity grows exponentially in the number of hypotheses. In principle, we must run $2^n - 1$ tests (we don't have to perform a test in the degenerate case $I = \emptyset$) to determine which of $H_1, \dots, H_n$ to reject. However, a theme of this lecture is that we can efficiently compute closed tests for some choices of φ_I.

5.1.1 Closing Bonferroni

If we use Bonferroni's procedure to construct the test φ_I, then we have

$$\varphi_I = 1 \iff \min\{p_i : i \in I\} \leq \frac{\alpha}{|I|},$$

where p_i is the p-value for testing H_i and $|I|$ denotes the size of I. In our first lecture, we saw that Bonferroni's procedure tests against the global null at level α. Thus, the closure of Bonferroni's procedure controls that FWER at level α.

We now show that, once we have sorted the p-values, we can avoid running 2^n tests and simply run n tests. Indeed, when we close Bonferroni, we recover a procedure we have already seen.

Proposition 1. The closure of Bonferroni's procedure is Holm's procedure.

Proof. To see why this is, let us first consider when we reject $H_{(1)}$, the hypothesis corresponding to the smallest p-value $p_{(1)}$. Since $p_{(1)}$ is the smallest p-value $\min\{p_i : i \in I\} = p_{(1)}$ for all I such that $(1) \in I$. It follows that, under the closure principle, we reject $H_{(1)}$ if and only if $p_{(1)} \leq \frac{\alpha}{|I|}$ for all I such that $(1) \in I$. The right hand side of this inequality is smallest when $|I| = \{1, \ldots, n\}$. Thus, we reject $H_{(1)}$ if and only if $p_{(1)} \leq \frac{\alpha}{n}$.

Similarly, we can ask when do we reject $H_{(2)}$, the hypothesis corresponding to the second smallest p-value $p_{(2)}$. If $(2) \in I$, then

$$\min\{p_i : i \in I\} = \begin{cases} p_{(1)} & \text{if } (1) \in I, \\ p_{(2)} & \text{if } (1) \notin I. \end{cases}$$

Thus, in order to reject $H_{(2)}$, we must have $p_{(1)} \leq \frac{\alpha}{|I|}$ for all I with $(1), (2) \in I$ and $p_{(2)} \leq \frac{\alpha}{|I|}$ for all I with $(1) \notin I$ and $(2) \in I$. The right hand sides of these inequalities are smallest when I equals $\{1, \ldots, n\}$ and $\{1, \ldots, n\} \setminus \{(1)\}$ respectively. Thus, we reject $H_{(1)}$ if and only if $p_{(1)} \leq \frac{\alpha}{n}$ and $p_{(2)} \leq \frac{\alpha}{n-1}$.

Continuing with this argument, we can see that if we fail to reject $H_{(i)}$ for some $i < j$, then we will not reject $H_{(j)}$. Furthermore, if we have rejected $H_{(1)}, \ldots, H_{(j-1)}$, then we must have $\min\{p_i : i \in I\} \leq \frac{\alpha}{|I|}$ for all I containing at least one of $(1), \ldots, (i-1)$. Thus, to determine if we reject $H_{(j)}$ we need only consider sets I that contain (j) but do not contain $(1), \ldots, (j-1)$. For such a set I, $\min\{p_i : i \in I\} = p_{(j)}$ and the largest such set is $I_{(j)} = \{1, \ldots, n\} \setminus \{(1), \ldots, (i-1)\}$. Thus, we will reject $H_{(j)}$ if and only if we have rejected $H_{(1)}, \ldots, H_{(j-1)}$ and $p_{(j)} \leq \frac{\alpha}{|I_{(j)}|} = p_{(i)} \leq \frac{\alpha}{n-i+1}$.

This argument shows that the closed Bonferroni procedure can be written as the following algorithm.

Algorithm 1 Closed Bonferroni Procedure

```
j ← 0;
while p_{(j+1)} ≤ α/(n−j) do
    j ← j + 1;
end while
Reject H_{(1)}, ..., H_{(j)}
```

Finally, we can realize that we have seen this algorithm before and it is exactly Holm's procedure. □

The fact the closed Bonferroni procedure gives us Holm's procedure is encouraging in that the closure principle appears to yield reasonable results. The reason why the closed Bonferroni procedure was computationally tractable was that for each hypothesis $H_{(j)}$ we could consider a worst case intersection hypothesis $H_{I_{(j)}}$ with $(j) \in I_{(j)}$. This worst case hypothesis $H_{I_{(j)}}$ had the property that, if we have rejected $H_{(1)}, \ldots, H_{(j-1)}$, then

$$\varphi_{I_{(j)}} = 1 \iff \varphi_I = 1 \text{ for all } I \text{ with } (j) \in I.$$

This "worst case analysis" will be useful in the next section where we close Simes procedure.

5.1.2 Closing Simes

For a subset $I \subseteq \{1, \ldots, n\}$, the Simes test statistic for testing against H_I is

$$\varphi_I = \max\left\{\mathbb{I}\left(p_{(1,I)} \leq \frac{\alpha}{|I|}\right), \mathbb{I}\left(p_{(2,I)} \leq \frac{2\alpha}{|I|}\right), \ldots, \mathbb{I}\left(p_{(|I|,I)} \leq \alpha\right)\right\},$$

where $p_{(j,I)}$ is the jth smallest p-value among $\{p_i : i \in I\}$. That is. Simes procedure rejects H_I if and only if there exists $p_{(j,I)} \leq j\alpha/|I|$ for some $j = 1, \ldots, |I|$. We have seen that under independence, the Simes procedure tests against the global null at level α. Thus, under independence, the closure of Simes procedure will control FWER at level α. Since Simes is strictly more powerful than Bonferroni, the closure of Simes will be strictly more powerful than Holm's procedure (the closure of Bonferroni).

The closure of Simes can be computed in linear time [6]. The resulting method is called Hommel's procedure and is a bit complicated. We will instead discuss a more conservative procedure called *Hochberg's* procedure. While more conservative than Hommel's procedure, this procedure is simpler to describe and still more powerful than Holm's. To summarize, we will show the following relationship

$$\text{Closure of Bonferroni (Holm's)} \prec \text{Hochberg} \prec \text{Closure of Simes (Hommel's)},$$

where $A \prec B$ means that A is a more conservative procedure than B. We will now describe Hochberg's procedure. Note however that the Hochberg and Simes procedures require assumptions about the dependencies between the p-values. Bonferroni's test and hence Holm's procedure makes no assumptions.

Definition 2 (Hochberg's procedure)**.** Given sorted p-values $p_{(1)} \leq \ldots \leq p_{(n)}$ and corresponding hypothesis $H_{(1)}, \ldots, H_{(n)}$, Hochberg's procedure is given by

$$\text{Reject } H_{(j)} \iff \text{there exists } j' \geq j \text{ such that } p_{(j')} \leq \frac{\alpha}{n - j' + 1}.$$

We know that the closure of Simes controls FWER at level α under independence. Thus, to conclude that Hochberg's procedure also controls FWER at level α, it suffices to show whenever Hochberg's procedure rejects a hypothesis, then the closure of Simes also rejects.

Lemma 1. If $H_{(j)}$ is rejected under Hochberg's procedure, then $H_{(j)}$ is also rejected under the closure of Simes.

Proof. Suppose that $H_{(j)}$ is rejected by Hochberg's procedure. Then, by definition, there exists $j' \geq j$ such that $p_{(j')} \leq \frac{\alpha}{n-j'+1}$. To show that $H_{(j)}$ is rejected by the closure of Simes, we need to prove that for all I with $(j) \in I$, H_I is rejected by Simes. Thus, fix such a set I and let $h = |I|$. Define

$$K_h = \begin{cases} \{(j), (n), (n-1), \ldots, (n-h+2)\} & \text{if } j \leq n-h+1, \\ \{(n), (n-1), \ldots, (n-h+1)\} & \text{if } j > n-h+1. \end{cases}.$$

Note that $|K_h| = h$ and $(j) \in K_h$ always. Furthermore, if Simes procedure rejects H_{K_h}, then Simes procedure will also reject H_I. This is because $|K_h| = |I|$ and the p-values indexed in

K_h are larger than than the p-values indexed in I. Since, the thresholds in Simes procedure only depend on the size of the set I, this shows that $\varphi_{K_h} = 1$ implies $\varphi_I = 1$.

By considering cases we will now show that $\varphi_{K_h} = 1$. First suppose that $h \leq n - j' + 1$. Then $p_{(j)}$ is the smallest p-value in K_h and so,

$$p_{(1,K_h)} \leq p_{(j)} \leq p_{(j')} \leq \frac{\alpha}{n - j + 1} \leq \frac{\alpha}{|K_h|}.$$

Thus, Simes procedure rejects H_{K_h}. Now suppose that $h \geq n - j' + 2$ so that $j' \geq n - h + 2$ which implies $(j') \in K_h$. Note that we must have $p_{(h-n+j',K_h)} \leq p_{(j')}$. This is because K_h contains h p-values and the $n - j'$ p-values $p_{(n)}, p_{(n-1)}, \ldots, p_{(j'+1)}$ are all in K_h and are all larger than $p_{(j')}$. Thus,

$$p_{(h-n+j',K_h)} \leq p_{(j')} \leq \frac{\alpha}{n - j' + 1} \leq \frac{(h - n + j')\alpha}{h}. \tag{5.1}$$

To see why the last inequality holds, let $a = n - j' + 1$ and $b = h - n + j'$. It suffices to show that $h \leq ab$. But note that $a + b - 1 = h$ and thus $ab - h = (a - 1)(b - 1)$. Since a and b are both integers greater than or equal zero we must have $ab - h = (a - 1)(b - 1) \geq 0$ and hence $ab \geq h$. By equation (5.1) and the definition of Simes procedure we can conclude that $\varphi_{K_h} = 1$. □

5.2 Step-Down vs. Step-Up Procedures

We claimed that Holm's procedure was more conservative than Hochberg's procedure. This can be seen when we write Holm's and and Hochberg's procedures side by side

Algorithm 2 Holm's Procedure

$j \leftarrow 0$;
while $p_{(j+1)} \leq \frac{\alpha}{n-j}$ **do**
    $j \leftarrow j + 1$;
end while
Reject $H_{(1)}, \ldots, H_{(j)}$

Algorithm 3 Hochberg's procedure

$j \leftarrow n$;
while $p_{(j)} > \frac{\alpha}{n-j+1}$ **do**
    $j \leftarrow j - 1$;
end while
Reject $H_{(1)}, \ldots, H_{(j)}$

Note that both procedures use the same thresholds and compare $p_{(j)}$ to $\frac{\alpha}{n-j+1}$. However,

- Holm's scans *forward* and stops at the first p-value that is **larger** than the corresponding threshold. As discussed last lecture, this is called a **step-down** procedure.
- Hochberg's scans *backwards* and stops at the first p-value that is **smaller** than the corresponding threshold. Likewise, this is called a **step-up** procedure.

In general, step-up procedures can be substantially more powerful than step-down procedures. In an extreme case, if $p_1 = \ldots = p_n = \alpha$, then Holm's rejects *nothing* whereas Hochberg's rejects *everything*. A less extreme example is shown in Figure 5.1. Here, by scanning forward, Holm's can only reject 3 hypotheses whereas Hochberg's can reject 8 hypotheses because $p_{(8)} \leq \frac{\alpha}{n-8+1}$.

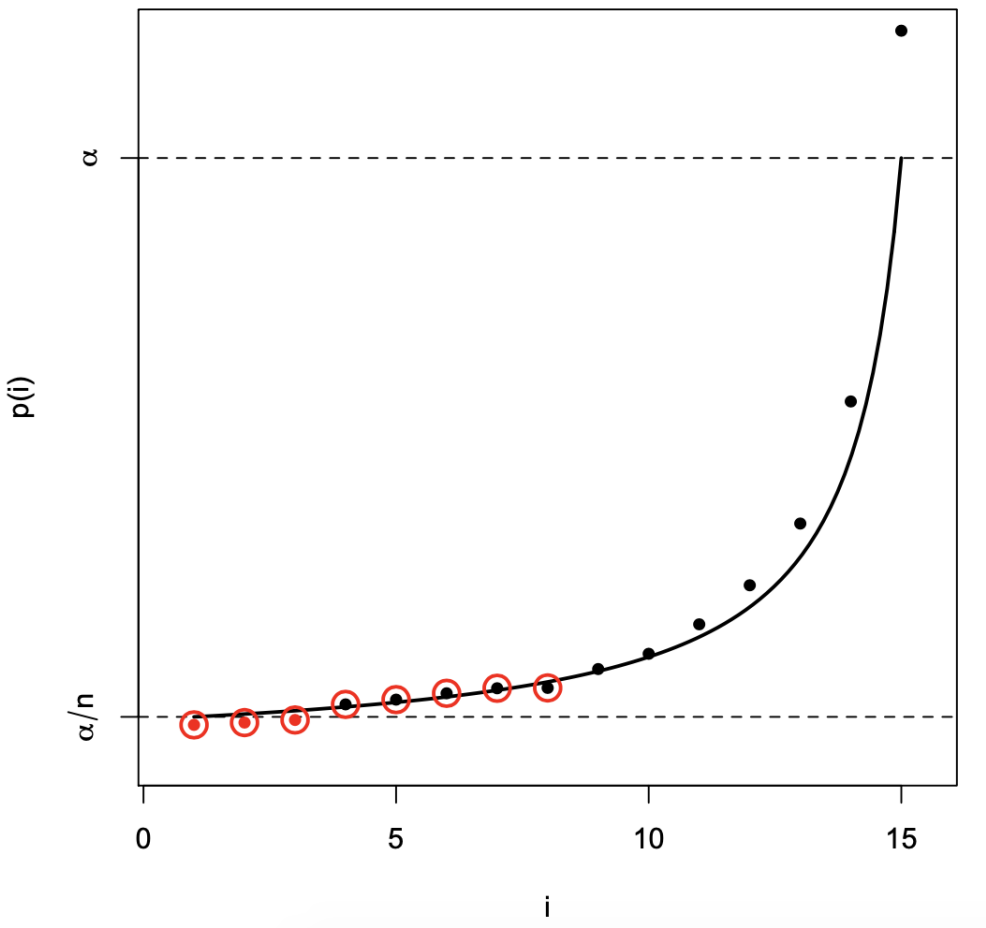

Figure 5.1. A comparison of Holm's step-down procedure and Hochberg's step-up procedure. Under independence, both control the FWER, but Hochberg's procedure is more powerful as it makes more rejections. Solid red points are hypotheses rejected by Holm's (3 rejections). Points circled in read are rejected by Hochberg's procedure (8 rejections)

As presented here, the names step-up and step-down may seem counter-intuitive and presented in the wrong order. For step-up procedures we start at the largest p-value and decrease but for step-down procedures we start at the smallest p-value and increase. The names make sense if we think about the procedures in terms of z-scores. That is if our p-values are of the form $p_i = \mathbb{P}(Z > z_i)$ where $Z \sim \mathcal{N}(0, 1)$, then the smallest p-values corresponds to the largest z-score. In this case, a step-up procedure would start at the smallest z-score and increase. A step-down procedure would start at the largest z-score and decrease.

In the next section, we will shift our focus and discuss how problem specific knowledge can be used to create a range of procedures that all control FWER.

5.3 Graphical procedures

Bonferonni, Holm's and Hochberg's procedures are all symmetric procedures. By this, we mean that if we permute the p-values between the hypotheses, then the rejected hypotheses will likewise be permuted. This approach is desirable if we don't have any prior knowledge about the hypotheses we are testing. This might be the case in a genome wide association study where we have no information about which genes are most likely to be associate with a gene.

There are other situations where the hypotheses are asymmetric and we would like to incorporate this asymmetry into our procedures [4]. A common example would be in clinical trials. Clinical trials tend to have multiple stages and a significant result in the first stage is often more important than a significant result in later stages. Indeed, a significant result in the earliest stage is often necessary for FDA approval. That is regardless of the significance and effect sizes of later stages, we need a significant result in the first stage. Another source of asymmetry may come from prior knowledge about effect sizes. If one alternative is expected to have a large effect size, then we may prioritize testing that hypothesis.

Graphical procedures are a family of procedures that use a variety prior knowledge to distributing the "α-budget" of the procedure unevenly and adaptively across the n hypotheses.

5.3.1 Sequential Testing

Before defining graphical procedures in general, we will discuss two examples.

Fixed sequence testing

Suppose that, prior to running our experiments, we have ordered our hypotheses $H_1, \dots, H_n$ where H_1 is the "most promising" hypothesis. We then perform our experiment and calculate a p-value p_i for each hypothesis H_i. The **fixed sequence test** first compares p_1 to α. If $p_1 > \alpha$, then the procedure terminates. If $p_1 \leq \alpha$, then we reject H_1 and compare p_2 to α. If $p_2 > \alpha$, then the procedure terminates. Otherwise, we reject H_2 and next compare p_3 to α. This procedure continues until either $p_j > \alpha$ for some j or until we reject every hypothesis. An example is shown in Figure 5.2.

As an algorithm, the fixed sequence procedure can be written as,

Algorithm 4 Fixed Sequence Test

1: **Inputs:** Ordered hypotheses $H_1, \dots, H_n$;
2: $j \leftarrow 1$;
3: **while** $p_j \leq \alpha$ **do**
4: $\quad j \leftarrow j + 1$;
5: **end while**
6: Reject $H_1, \dots, H_j$.

We next show that this procedure controls the FWER. Recall that controlling the FWER strongly means that the FWER is controlled under all possible arrangements of null and

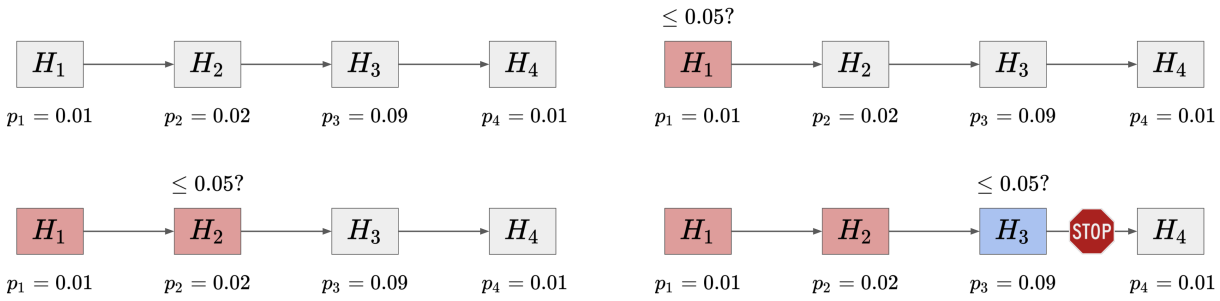

Figure 5.2. Above we apply the fixed sequence test to four hypotheses with $\alpha = 0.5$. Since $p_1, p_2 \leq 0.05$ we reject H_1 and then also reject H_2. However, $p_3 > \alpha$ and so the procedure terminates and we do not reject H_3. We do have $p_4 \leq \alpha$ but we do not reject H_4 because H_3 comes before H_4.

alternative hypotheses. This is in contrast to controlling the FWER weakly in which the FWER is only controlled under the global null.

Theorem 1. The fixed sequence test controls the FWER strongly.

Proof. Let H_{i^*} be the first true null hypothesis. That is i^* is the smallest index i such that H_i is true. Then,

$$\begin{aligned}
\text{FWER} &= \mathbb{P}(V \geq 1) \\
&= \mathbb{P}(\text{reject } H_{i^*}) \\
&= \mathbb{P}(p_1 \leq \alpha, p_2 \leq \alpha, \ldots, p_{i^*} \leq \alpha) \\
&\leq \mathbb{P}(p_{i^*} \leq \alpha) \\
&\leq \alpha.
\end{aligned}$$

□

In the above proof it was important that the order of the hypotheses was chosen in advance and independently of the p-values. Indeed, if the order depended on the p-values, then i^* would be a random variable and we may no longer have $\mathbb{P}(p_{i^*} \leq \alpha) \leq \alpha$ even if H_{i^*} is true.

Fallback procedure

In Figure 5.2 we did not reject H_4 even though $p_4 = 0.01 < 0.5$. This is because we failed to reject H_3 and thus never considered p_4. **Fallback procedures** provide a way of making sure we at least consider each hypothesis.

Like fixed sequence testing, a fallback procedure requires that an ordered list of hypotheses $H_1, \ldots, H_n$ must be chosen before the experiment. Unlike fixed sequence testing, a set of thresholds $\alpha_1, \ldots, \alpha_n \geq 0$ such that $\sum_{i=1}^n \alpha_i = \alpha$ must also be fixed in advance.

Once the p-values have been calculated, the fallback procedure does the following:

- First we compare p_1 to α_1. If $p_1 \leq \alpha_1$, then we reject H_1 and change α_2 to $\alpha_2 + \alpha_1$. If $p_1 > \alpha_1$, then we fail to reject H_1 and we leave α_2 unchanged.
- We then reject H_2 if and only if $p_2 \leq \alpha_2$. If we reject H_2, then we change α_3 to $\alpha_3 + \alpha_2$.

- In general, assuming we have tested the hypotheses $H_1, \ldots, H_{j-1}$, we next test H_j. We reject H_j if and only $p_j \leq \alpha_j$. If we reject H_j and $j < n$, then we change α_{j+1} to $\alpha_{j+1} + \alpha_j$.
- The procedure terminates once we have tested every hypothesis.

An example is shown in Figure 5.3.

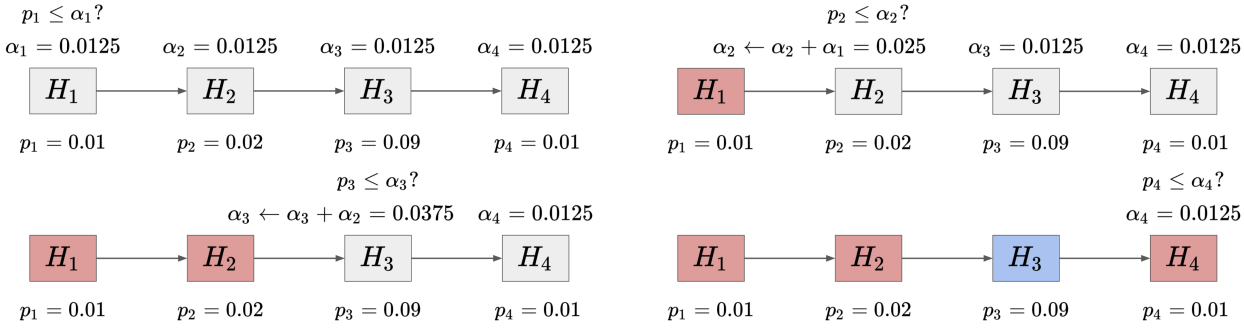

Figure 5.3. Above we apply a fallback procedure to four hypotheses with $\alpha_j = 0.0125 = 0.05/4$. We first compare p_1 to α_1. Since $0.01 \leq 0.0125$, we reject H_1 and change α_2 to $\alpha_2 + \alpha_1 = 0.025$. We then compare p_2 to α_2. Since $0.02 \leq 0.025$, we reject H_1 and change α_3 to $\alpha_3 + \alpha_2 = 0.0375$. Since $0.09 > 0.035$ we do not reject H_3 and we do not change α_4. Since $0.01 \leq 0.0125$, we do reject H_4. Note that the p-values are the same as in Figure 5.2. In contrast to the fixed sequence test in Figure 5.2, the fallback procedure rejects H_4.

We can write the fallback procedure as an algorithm.

Algorithm 5 Fallback procedure

Inputs: Ordered hypotheses $H_1, \ldots, H_n$, thresholds $\alpha_1, \ldots, \alpha_n \geq 0$, $\sum_{i=1}^n \alpha_i = \alpha$;
$\mathcal{R} \leftarrow \emptyset$;
for $j = 1, \ldots, n$ **do**
    if $p_j \leq \alpha_j$ **then**
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{j\}$;
        $\alpha_{j+1} \leftarrow \alpha_{j+1} + \alpha_j$;
    end if
end for
Reject $\mathcal{R}$.

Note that the fixed sequence test is a fallback procedure with $\alpha_1 = \alpha$ and $\alpha_i = 0$ for $i > 1$. We next show that fallback procedures control the FWER strongly.

Theorem 2. Given any ordering of the hypotheses $H_1, \ldots, H_n$ and any thresholds $\alpha_1, \ldots, \alpha_n$ with $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i = \alpha$, the corresponding fallback procedure controls FWER strongly.

Proof. Let $j_1 < j_2 < \ldots < j_{n_0}$ be the indexes of the true nulls. Note that,

$$\begin{aligned}
\text{FWER} &= \mathbb{P}(H_{j_k} \text{ rejected for some } k = 1, \ldots, n_0) \\
&= \mathbb{P}(H_{j_1} \text{ rejected}) + \mathbb{P}(H_{j_2} \text{ rejected and } H_{j_1} \text{ not rejected}) \\
&\quad + \ldots + \mathbb{P}(H_{j_{n_0}} \text{ rejected and } H_{j_k} \text{ not rejected for } k < n_0) \\
&\leq \mathbb{P}(p_{j_1} \leq \alpha_1 + \ldots + \alpha_{j_1}) + \mathbb{P}(p_{j_2} \leq \alpha_{j_1+1} + \ldots + \alpha_{j_2}) \\
&\quad + \ldots + \mathbb{P}(p_{j_{n_0}} \leq \alpha_{j_{n_0-1}+1} + \ldots + \alpha_{j_{n_0}}) \\
&\leq \sum_{i=1}^{j_{n_0}} \alpha_i \\
&\leq \alpha
\end{aligned}$$

□

We have a couple of remarks about the above proof.

- As with the fixed sequence test, the important assumption was that the order of hypotheses and the thresholds α_i were specified independently of the p-values.
- The proof makes no assumptions about the dependencies between the p-values.
- The proof is similar to the proof that closed procedures control the FWER. In both proofs we consider the subset of $\{1, \ldots, n\}$ containing the true nulls. This connection is explored further in the following sections.

5.3.2 General Graphical Procedures

We are now ready to describe graphical procedures in full generality. To specify a graphical procedure, the following must be given

- An error level $\alpha \in (0, 1)$.
- A set of initial thresholds $\alpha_1, \ldots, \alpha_n \geq 0$ such that $\sum_{i=1}^n \alpha_i = \alpha$
- A weighted directed graph $(w_{ij})_{i,j=1}^n$ with $w_{ij} \geq 0$, $w_{ii} = 0$ and $\sum_{j=1}^n w_{ij} \leq 1$ for all $i = 1, \ldots, n$.

The graph in a graphical procedure can be thought of as having vertices $\{H_1, \ldots, H_n\}$, and edges from H_i to H_j weighted by w_{ij}. We interpret $w_{ij} = 0$ as the absence of an edge between H_i and H_j. Thus the condition $w_{ii} = 0$ means that we do not allow self-loops. Cycles of length greater than 1 are allowed.

Given the above inputs, the graphical procedure is an iterative procedure that does the following. At each time step, we have a set of current hypotheses $\{H_i\}_{i \in I}$. This collection is the set of hypotheses we are yet to reject. First we check if there exists an index $i \in I$ with $p_i \leq \alpha_i$. If no such index exists, then procedure terminates. If such an $i \in I$ does exist, then we reject H_i and performing the following update to our set of hypotheses, thresholds and

weights,

$$\begin{aligned} I &\leftarrow I \setminus \{i\} \\ \alpha_j &\leftarrow \alpha_j + \alpha_i w_{ij}, \\ w_{jk} &\leftarrow \begin{cases} \frac{w_{jk}+w_{ji}w_{ik}}{1-w_{ji}w_{ij}} & \text{if } j \neq k \\ 0 & \text{if } j = k. \end{cases} \end{aligned} \tag{5.2}$$

We then repeat this procedure with the new set of hypotheses. Written as an algorithm, the graphical procedure is

Algorithm 6 Graphical Procedure

Inputs: Thresholds $(\alpha_i)_{i=1}^n$, weights $(w_{ij})_{i,j=1}^n$;
$\mathcal{R} \leftarrow \emptyset$;
$I \leftarrow \{1, \ldots, n\}$
while There exists $i \in I$ with $p_i \leq \alpha_i$ **do**
    Find i such that $p_i \leq \alpha_i$;
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$;
    $I \leftarrow I \setminus \{i\}$;
    for $j \in I$ **do**
        $\alpha_j \leftarrow \alpha_j + \alpha_i w_{ij}$;
        for $k \in I$ **do**
            if $k \neq j$ **then**
                $w_{jk} \leftarrow \frac{w_{jk}+w_{ji}w_{ik}}{1-w_{ji}w_{ij}}$
            end if
        end for
    end for
end while
Reject $\mathcal{R}$.

We can see that when we reject a hypothesis H_i, we pass H_i's "budget" α_i to the remain hypotheses according to the weights w_{ij}. The updated weights in equation (5.2) can be given the following interpretation. The previous w_{ij} can be roughly thought of as the transition probabilities for a random walk on $\{H_i\}_{i \in I}$. Once we reject H_i we construct a new random walk on $\{H_j\}_{j \in I \setminus \{i\}}$. In the new random walk we can either transition from H_j to H_k directly or we can transition via the deleted state H_i. This gives the numerator $w_{jk} + w_{ji}w_{ik}$ in (5.2). The denominator in (5.2) comes because we do not allow the random walk to stay in the same state. Thus, we must correct for the possibility of a transition from H_j back to H_j via H_i.

Many of the procedures we have seen so far fall under the umbrella of graphical procedures. For example, if we apply Bonferroni and reject all hypotheses H_i with $p_i \leq \alpha/n$, then we are performing a graphical procedure with $\alpha_i = \alpha/n$ and $w_{ij} = 0$ for all i and j. Holm's procedure is also a graphical procedure with $\alpha_i = \alpha/n$ and $w_{ij} = \frac{1}{n-1}$ for all i and j with $i \neq j$. Figure 5.4 shows an example of using Holm's as a graphical procedure.

The fixed sequence procedure is a graphical procedure with $\alpha_1 = \alpha$, $\alpha_i = 0$ for $i = 2, \ldots, n$ and $w_{i,i+1} = 1$ for $i = 1, \ldots, n-1$ and $w_{ij} = 0$ for all other i, j. Likewise, the fallback

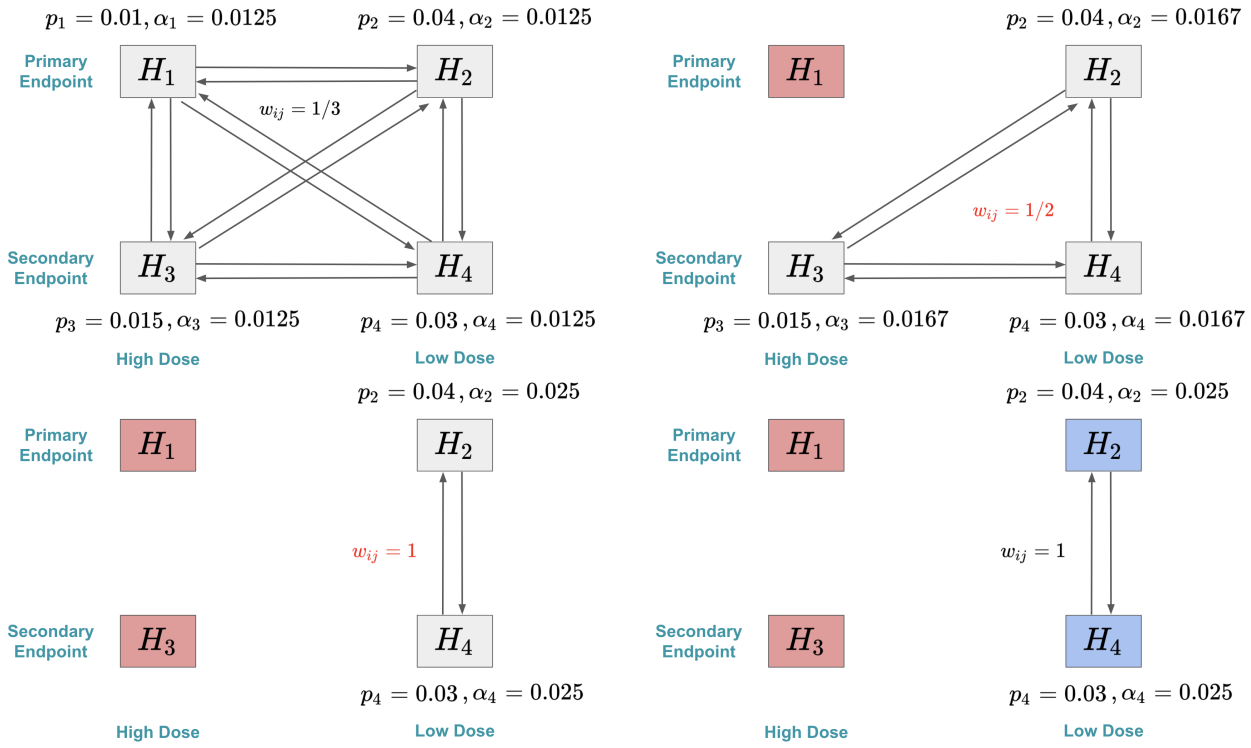

Figure 5.4. Using Holm's procedure as a graphical procedure. There are $n = 4$ tests and thus the initial thresholds are $0.05/4 = 0.0125$ and the initial weights are $1/3 = 1/(4-1)$. Since $p_1 \leq \alpha_1$, we reject H_1. We then update the thresholds and weights so that we have thresholds of $0.05/3 \approx 0.0167$ and weights $1/2 = 1/(4-2)$. Of the remaining hypotheses we have $p_3 \leq \alpha_3$ and thus we reject H_3. The thresholds are updated to $0.05/2 = 0.025$ and the weights are updated to equal 1. The procedure then terminates since $p_2 > \alpha_2$ and $p_4 > \alpha_4$. The labels "Primary/Secondary Endpoint" and "High/Low Dose" refer to a hypothetical clinical trial. In such a trial the primary endpoint is important and we would expect an large effect under the high dose. Thus, an asymmetric graphical procedure like the one shown in Figure 5.5 may be more powerful than Holm's procedure.

procedure is a graphical procedure with the same weights as the fixed sequence procedure but with arbitrary thresholds. Figure 5.5 shows a new graphical procedure that is asymmetric and incorporates prior information about the hypotheses.

Importantly, all graphical procedures control the FWER.

Theorem 3. Given any thresholds $\alpha_1, \ldots, \alpha_n$ and weights $(w_{ij})_{i,j=1}^n$, the corresponding graphical procedure described in Algorithm 6 controls the FWER strongly under arbitrary dependencies between the p-values. Furthermore, the set of rejected hypotheses $\mathcal{R}$ does not depend on the order in which the hypotheses are rejected.

This theorem is proved in Appendix A of [2] (see also [3]). We will sketch the proof and discuss some related concepts in the next section. Theorem 3 follows by showing that every graphical procedure is the closure of a *weighted Bonferroni procedure*. This will show that graphical procedures are a large class of computationally tractable closed procedures. Indeed, in algorithm 6 the number of operations is $O(n^3)$. Thus, with graphical procedure we avoid the exponential complexity of general closed procedures.

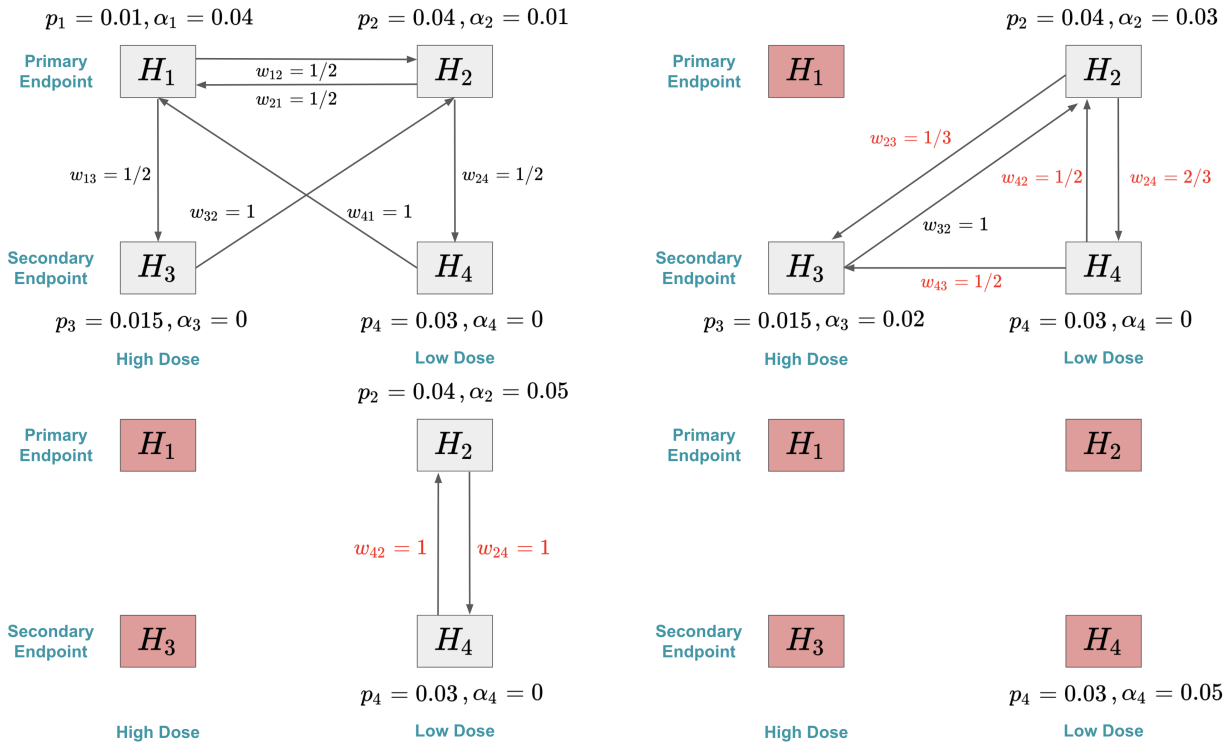

Figure 5.5. An asymmetric graphical procedure with the same hypotheses and p-values as in Figure 5.4. The thresholds α_i and weights w_{ij} are chosen to reflect the greater importance of hypotheses H_1, H_2 and the stronger signal strength under the high dose. As in Figure 5.4 we first reject H_1. We then update the thresholds and weights and reject H_3. We then update the thresholds and weights again and reject H_2 and H_4. Note that by using a problem specific structure we get a more powerful test.

One limitation of graphical procedures is their flexibility. A choice of $\alpha_1, \ldots, \alpha_n$ and $(w_{ij})_{i,j=1}^n$ that performs well in one setting need not perform well in another setting. In [7], it is shown that graphical procedures (and, more generally, *consonant procedures*) are optimal in specific settings. However, knowing how to choose optimal thresholds and weights in a given setting is an open problem.

5.3.3 Closing Weighted Bonferroni and Consonance

In this section we will see that graphical procedures are the closures of weighted Bonferroni tests. Theorem 3 will thus be a consequence of this connection. Along the way we will define consonant procedures and examine an example from [7]. We first define weighted Bonferroni tests.

Definition 3. Given $\alpha_1, \ldots, \alpha_n \geq 0$ with $\sum_{i=1}^n \alpha_i \leq \alpha$, the *weighted Bonferroni test* rejects the global null if $p_i \leq \alpha_i$ for some i.

Weighted Bonferroni tests control the type one error of rejecting the global null at level α. This holds without any dependence assumptions. Bonferroni's test is the special case when $\alpha_i = \alpha/n$ for every i. Since we are interested in closing such tests, we need to have thresholds $\alpha_i(I)$ for every intersection hypothesis H_I. Thus, suppose that we have thresholds $\alpha_i(I)$ for

every $I \subseteq \{1, \ldots, n\}$ and $i \in I$ such that $\sum_{i \in I} \alpha_i(I) \leq \alpha$. The weighted Bonferroni's test rejects H_I if and only if $p_i \leq \alpha_i(I)$ for some i. That is,

$$\varphi_I = \max\{\mathbb{I}(p_i \leq \alpha_i(I)) : i \in I\}.$$

We would like to impose a relationship between $\alpha_i(I)$ and $\alpha_i(J)$ for $i \in J \subseteq I$. Recall that φ_I is a test of the global null of $\{H_i : i \in I\}$. If $i \in J \subseteq I$, then p_i is a p-value that can be used to test both H_I and H_J. The set I is bigger than the set J. This means that when we test H_I we have a larger multiple testing problem than when we test H_J. Thus, to control probability of falsely rejecting a null hypothesis, the threshold $\alpha_i(I)$ should be smaller than $\alpha_i(J)$. This property is called *monotonicity* and can be written as

$$\alpha_i(I) \leq \alpha_i(J) \quad \text{for all } i \in J \subseteq I. \tag{5.3}$$

Monotonicity has the following important consequence.

Lemma 2. Suppose that the thresholds $\alpha_i(I)$, $I \subseteq \{1, \ldots, n\}$, $i \in I$, satisfy equation (5.3) and that φ_I are the corresponding weighted Bonferroni tests. Then, for all non-empty $I \subseteq \{1, \ldots, n\}$, if H_I is rejected, then there exists $i \in I$ such that for all J such that $i \in J \subseteq I$, H_J is also rejected.

Proof. If H_I is rejected, then $p_i \leq \alpha_i(I)$ for some i. If $i \in J \subseteq I$, then by equation (5.3)

$$p_i \leq \alpha_i(I) \leq \alpha_i(J).$$

Thus, H_J is rejected. □

Lemma 2 proves that, provided (5.3) holds, the closure of the weighted Bonferroni tests satisfies a property named *consonance*.

Definition 4. Let $\{\varphi_I\}$ be a family of tests for $\{H_I : I \subseteq \{1, \ldots, n\}\}$. The family $\{\varphi_I\}$ is *consonant* if for all $I \subseteq \{1, \ldots, n\}$ if $\varphi_I = 1$, then there exists $i \in I$ such that for all J with $i \in J \subseteq I$, $\varphi_J = 1$.

Note that step-up and step-down procedures are always consonant. This is because if such a procedure rejects an intersection hypothesis H_I, then the procedure must have rejected H_J for all J containing (i, I), where (i, I) is the index of the smallest p-value indexed in I.

Without consonance one could be in a situation where the global null has been rejected but none of the individual nulls have been rejected. An example of a nonconsonant test is given in [7] and discussed below.

Example 1 (A nonconsonant procedure)**.** Suppose that for $i = 1, 2$ we have $X_i \overset{\text{ind}}{\sim} \mathcal{N}(\theta_i, 1)$. For $i = 1, 2$, let H_i be the hypothesis that $\theta_i = 0$. Since X_1 and X_2 are independent, $X_1^2 + X_2^2 \sim \chi_2^2$ under the global null and $X_i^2 \sim \chi_1^2$ under H_i. One may then propose the following local tests,

$$\text{Reject the global null } H_{\{1,2\}} \iff X_1^2 + X_2^2 \geq \chi_2^2(1-\alpha),$$
$$\text{Reject the null } H_i \iff X_i^2 \geq \chi_1^2(1-\alpha).$$

We could then close the above tests to obtain a procedure that control the FWER at α. However, the result tests will be nonconsant. This is because $2\chi_1^2(1-\alpha) > \chi_2^2(1-\alpha)$. Thus, one can construct a scenario where $X_1^2, X_2^2 < \chi_2^2(1-\alpha)$ and $X_1^2 + X_2^2 \geq \chi_2^2(1-\alpha)$. This would lead to a scenario where the global null is rejected but neither H_1 nor H_2 is rejected.

One solution is to consider the step-down procedure,

$$\text{Reject the global null } H_{\{1,2\}} \iff \max\{X_1^2, X_2^2\} \geq m(1-\alpha),$$
$$\text{Reject the null } H_i \iff X_i^2 \geq \chi_1^2(1-\alpha),$$

where $m(1-\alpha)$ denotes the $1-\alpha$ quantile of the maximum of two independent χ_1^2 random variables. Since $m(1-\alpha) > \chi_1^2(1-\alpha)$, the closure of this procedure is consonant. A similar argument shows that the closure of Fisher's combination test is also nonconsonant.

In our setting, consonance allows us to perform a "greedy" algorithm that efficiently performs the closure of the weighted Bonferroni test.

Algorithm 7 Closed weighted Bonferroni procedure

Inputs: A function that can compute $\alpha_i(I)$ satisfying (5.3);
$\mathcal{R} \leftarrow \emptyset$;
$I \leftarrow \{1, \ldots, n\}$
for $j \in I$ **do**
    Compute $\alpha_j(I)$;
end for
while There exists $i \in I$ with $p_i \leq \alpha_i(I)$ **do**
    Find i such that $p_i \leq \alpha_i(I)$;
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$;
    $I \leftarrow I \setminus \{i\}$;
    for $j \in I$ **do**
        Compute $\alpha_j(I)$;
    end for
end while
Reject $\mathcal{R}$.

Notice the similarities between the above greedy algorithm and algorithm 6 for graphical procedures. Algorithm 7 for closed weight Bonferroni avoids the exponential complexity of the closure principle by only computing the thresholds $\alpha_i(I)$ for a small number of subsets I. Indeed, the number of such subset is at most n. Thus, if $\{\alpha_i(I) : i \in I\}$ can be calculated in polynomial time in $|I|$, then algorithm 7 runs in polynomial time in n.

5.3.4 Graphical Procedures as Weighted Bonferroni Procedures

We now show that graphical procedures can be described as the closure of a weighted Bonferroni procedure. Thus, suppose we are given thresholds $\alpha_1, \ldots, \alpha_n \geq 0$ and weights $(w_{ij})_{i,j=1}^n$ as in Section 5.3.2.

We will define the local thresholds $\alpha_i(I)$ and local weights $w_{ij}(I)$ by backwards induction on the size of I. If $I = n$, then we simply define $\alpha_i(I) = \alpha_i$ and $w_{ij}(I) = w_{ij}$. If $I = J \setminus \{i\}$ for some i, then for all $j, k \in I$

$$\begin{aligned} \alpha_j(I) &= \alpha_j(J) + \alpha_i(J) w_{ij}(J), \\ w_{jk}(I) &= \begin{cases} \frac{w_{jk}(J) + w_{ji}(J) w_{ik}(J)}{1 - w_{ji}(J) w_{ij}(J)} & \text{if } j \neq k \\ 0 & \text{if } j = k. \end{cases} \end{aligned} \tag{5.4}$$

In Appendix A of [2], it is shown that the above definition does not depend on the order in which indices are removed from $\{1, \ldots, n\}$ to produce I. One can also prove this by considering the random walk on $\{H_i : i \in I\}$ with weights $w_{ij}(I)$ as described after equation (5.2). From equation (5.4), we can immediately see that the thresholds $\alpha_i(I)$ satisfy monotonicity (5.3). Finally, by comparing algorithms 6 and 7 one can see that the graphical procedure does indeed compute the closure of the weighted Bonferroni procedure with thresholds given by (5.4).

Since closed procedures always control the FWER, we can conclude that graphical procedures also control the FWER. Furthermore, since graphical procedures are consonant, they satisfy the optimality condition described in Section 4 of [7].

5.4 Multiple testing in machine learning

The lecture finished with an overview of [1]. This recent paper includes an application of multiple testing in machine learning. The machine learning problem is multi-label classification. The algorithm learns to take pictures as inputs and outputs a list of labels of objects in the picture. See Figure 5.4 for some examples.

The machine learning algorithm works by learning a map that takes in an image and assigns a probability to each possible labels. To use this map to perform multi-label classification, a practitioner must choose a threshold $\lambda \in (0, 1)$. When an image is given to the algorithm, the labels with probability at least λ are said to be in the picture. The practitioner would like to choose λ so that the algorithm returns as many labels as possible while controlling the number of spurious labels.

This can phrased as a multiple testing problem by defining the hypotheses H_i to be that the image does not contain object i. Each rejection is an object claimed to be in the image. Each false positive is an object said to be in the image that is not present. The type of error control used is the *false discovery rate* (see next lecture). In this setting the false discovery rate is a function of λ and will be written as $\mathrm{FDR}(\lambda)$.

The paper provides a method for choosing an estimate $\widehat{\lambda}$ based on a calibration set, which is a random sample of images and true labels independent of the training images and labels. Given $\alpha, \delta \in (0, 1)$, the estimate $\widehat{\lambda}$ can be chosen such that,

$$\mathbb{P}(\mathrm{FDR}(\widehat{\lambda}) \leq \alpha) \geq 1 - \delta. \tag{5.5}$$

The construction of $\widehat{\lambda}$ is similar to the graphical and sequence procedures we have seen today. The interval $[0, 1]$ is discretized into finitely many values $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. For each value

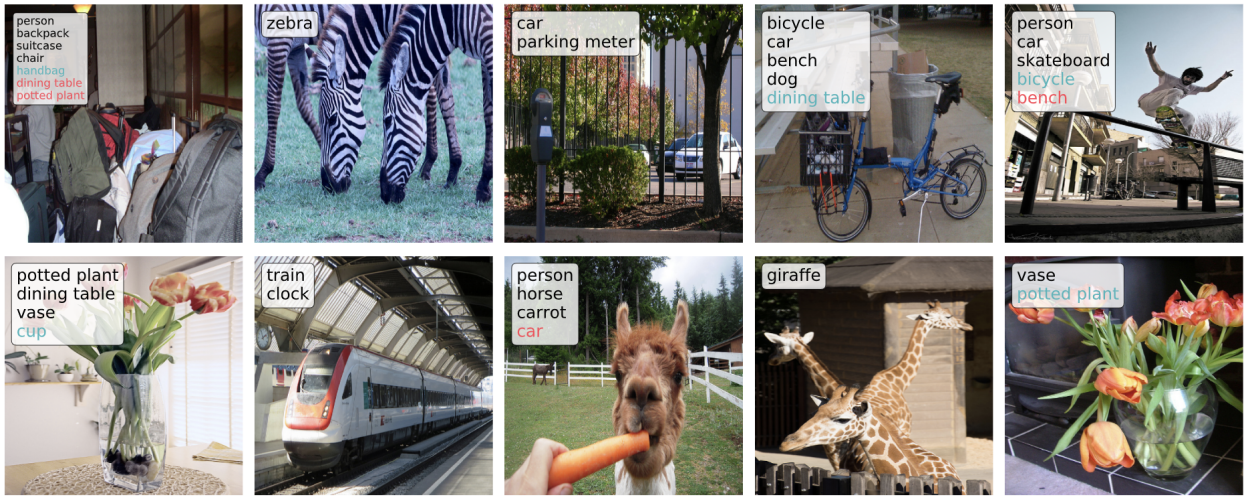

Figure 2: **Multi-label prediction set examples on MS COCO.** Black classes are correctly identified (true positives), blue ones are spurious (false positives), and red ones are missed (false negatives). The sets are produced with parameters $\alpha = 0.2$ and $\delta = 0.1$ using the fixed sequence testing procedure.

Figure 5.6. Image from [1] with permission.

λ_i we have a null hypothesis H_{λ_i} which is that $\text{FDR}(\lambda_i) > \alpha$. Equation 5.5 then relates to the FWER of testing $H_1, \ldots, H_n$. The testing procedure applied to $H_1, \ldots, H_n$ is a sequential procedure that exploits the fact that we expect $\text{FDR}(\lambda)$ to be a smooth function of λ and thus monotone on small intervals, see Figure 5.7

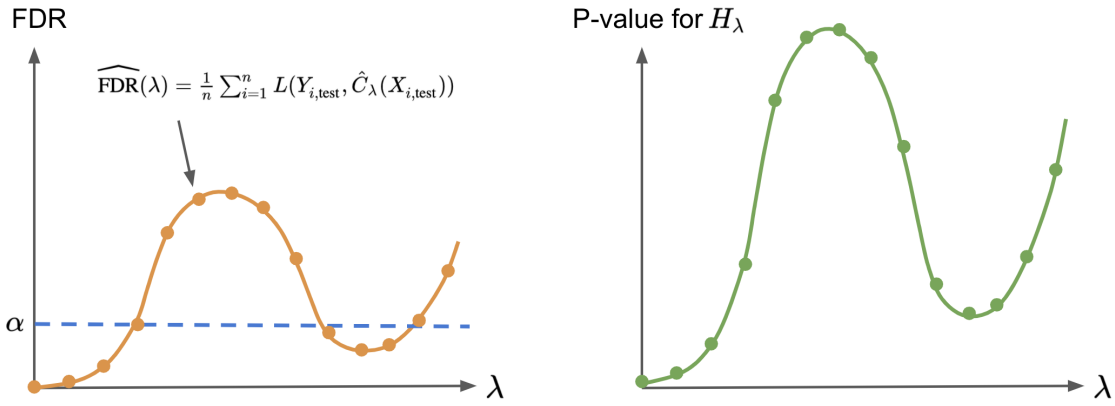

Figure 5.7. In [1], the value $\widehat{\text{FDR}}(\lambda_i)$ gives a p-value for H_{λ_i}. The value $\widehat{\lambda}$ is then chosen using a sequential testing procedure that control the FWER.

This procedure can also be applied in other domains such as tumor detection, hierarchical classification and protein structure prediction.

Bibliography

[1] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *CoRR*, abs/2110.01052, 2021.

[2] Frank Bretz, Willi Maurer, Werner Brannath, and Martin Posch. A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine*, 28, 2009.

[3] Carl-Fredrik Burman, Christian Sonesson, and Olivier J M Guilbaud. A recycling framework for the construction of bonferroni-based multiple tests. *Statistics in Medicine*, 28, 2009.

[4] Alex Dmitrienko and Ralph B. D'Agostino. Multiplicity considerations in clinical trials. *New England Journal of Medicine*, 378(22):2115–2122, 2018. PMID: 29847757.

[5] Ruth Marcus, Peritz Eric, and K. R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660, 12 1976.

[6] Rosa J. Meijer, Thijmen J. P. Krebs, and Jelle J. Goeman. Hommel's procedure in linear time. *Biometrical Journal*, 61(1):73–82, 2019.

[7] Joseph Romano, Azeem Shaikh, and Michael Wolf. Consonance and the closure method in multiple testing. *Institute for Empirical Research in Economics - IEW, IEW - Working Papers*, 7, 09 2009.