Taylor & Francis
Taylor & Francis Group

Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: www.tandfonline.com/journals/uasa20

Controlling the False Split Rate in Tree-Based Aggregation

Simeng Shao, Jacob Bien & Adel Javanmard

To cite this article: Simeng Shao, Jacob Bien & Adel Javanmard (24 Sep 2024): Controlling the False Split Rate in Tree-Based Aggregation, Journal of the American Statistical Association, DOI: 10.1080/01621459.2024.2376285

To link to this article: https://doi.org/10.1080/01621459.2024.2376285

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2024, VOL. 00, NO. 0, 1–13: Theory and Methods
https://doi.org/10.1080/01621459.2024.2376285

Taylor & Francis
Taylor & Francis Group

Controlling the False Split Rate in Tree-Based Aggregation

Simeng Shao[a], Jacob Bien[b], and Adel Javanmard[b]

[a]Amazon, Seattle, WA; [b]Data Sciences and Operations, University of Southern California, Los Angeles, CA

ABSTRACT

In many domains, data measurements can naturally be associated with the leaves of a tree, expressing the relationships among these measurements. For example, companies belong to industries, which in turn belong to ever coarser divisions such as sectors; microbes are commonly arranged in a taxonomic hierarchy from species to kingdoms; street blocks belong to neighborhoods, which in turn belong to larger-scale regions. The problem of tree-based aggregation that we consider in this article asks which of these tree-defined subgroups of leaves should really be treated as a single entity and which of these entities should be distinguished from each other. We introduce the *false split rate*, an error measure that describes the degree to which subgroups have been split when they should not have been. While expressible as the false discovery rate in a special case, we show that these measures can be quite different for the general tree structures common in our setting. We then propose a multiple hypothesis testing algorithm for tree-based aggregation, which we prove controls this error measure. We focus on two main examples of tree-based aggregation, one which involves aggregating means and the other hich involves aggregating regression coefficients. Supplementary materials for this article are available online, including a standardized description of the materials available for reproducing the work.

ARTICLE HISTORY
Received August 2021
Accepted June 2024

KEYWORDS
False discovery rate;
Hierarchy; Multiple testing;
Rare features

1. Introduction

A common challenge in data modeling is striking the right balance between models that are sufficiently flexible to adequately describe the phenomenon being studied and those that are simple enough to be easily interpretable. We consider this tradeoff within the increasingly common context in which data measurements can be associated with the leaves of a known tree. Such data structures arise in myriad domains from business to science, including the classification of occupations (US OMB 2018), businesses (US OMB 2017), products, geographic areas, and taxonomies in ecology.

Measurements in low-level branches of the tree may share a lot in common, and so—in the absence of evidence to the contrary—a data modeler would favor a simpler (literally "high-level") description in which distinctions within low-level branches would not be made; on the other hand, when there is evidence of a difference between sibling branches, then modeling them as distinct from each other may be warranted. We use the term *tree-based aggregation* to refer to the process of deciding which branches' leaves should be treated as the same (i.e., aggregated) and which should be treated as different from each other (i.e., split apart).

Tree-based aggregation procedures have been proposed in various contexts, including regression problems, in which features represent counts of rare events (Yan and Bien 2021) or counts of microbial species (Bien et al. 2021), and in graphical modeling (Wilms and Bien 2022). These approaches focus on prediction and estimation but do not address the hypothesis testing question of whether a particular split should occur.

We formulate the general tree-based aggregation problem as a multiple testing problem involving a parameter vector $\boldsymbol{\theta}^*$ whose elements correspond to leaves of a known tree. Our goal is to partition the leaves based on branches of the tree so that the set of parameters in each group share the same value. Every non-leaf node has an associated null hypothesis that states that all of its leaves have the same parameter value. Type I errors correspond to splitting up groups unnecessarily; Type II errors correspond to aggregating groups with different parameter values.

In Section 2, we define an error measure, called the *false split rate* (FSR), that corresponds to the fraction of splits made that were unnecessary. We study the FSR's relationship to the false discovery rate (Benjamini and Hochberg 1995), showing an equivalence in a special case and demonstrating that FDR is not sufficient in the general situations we care about.

In Section 3, we propose a tree-based aggregation procedure that leverages this connection. Our algorithm proceeds in a top-down fashion, only testing hypotheses of nodes whose parents were rejected. Such an approach to hierarchical testing originates with Yekutieli (2008), which lays the foundation for the multiple testing problem on trees. Our procedure is closely related to more recent work by Lynch and Guo (2016), which increases power using carefully chosen node-specific thresholds that depend on where the hypothesis is located in the hierarchy. This work was in turn further developed in Ramdas et al. (2019). Other work involving various forms of a multiple testing problem with tree-structured hypotheses (although not having to do with aggregation in the sense of this article) include Zhong

CONTACT Jacob Bien ✉ jbien@usc.edu 🖃 Marshall School of Business, 3670 Trousdale Parkway, BRI 401W, Los Angeles, CA 90089

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

et al. (2004), Hu, Zhao, and Zhou (2010), Heller et al. (2018), Katsevich and Sabatti (2019), Bogomolov et al. (2021). While these works focus on FDR control, another line of work uses hierarchical testing while controlling the family-wise error rate (Meinshausen 2008; Meijer and Goeman 2015; Guo et al. 2021). Our motivation of finding the proper "resolution" in a tree-structured multiple hypothesis testing context is shared by Katsevich, Sabatti, and Bogomolov (2021). They frame the problem as designing what they call a filter, which simplifies the possibly redundant set of discoveries while preserving FDR control of the final result. Our setting differs from theirs in our focus on aggregation hypotheses, which leads us in a different direction, creating an aggregation-geared error measure and multiple testing procedure.

In Section 4, we consider two concrete scenarios where tree-based aggregation is natural. In the first scenario, the parameter vector $\boldsymbol{\theta}^*$ represents the mean of a scalar signal measured on the leaves of the tree. In the second scenario, $\boldsymbol{\theta}^*$ is a (potentially high-dimensional) vector of regression coefficients where features are associated with leaves of the tree.

Finally, we demonstrate through simulation studies (Section 5) and real data experiments (Section 6) the empirical merits of our framework and algorithm. We consider two applications, corresponding to the two concrete scenarios of tree-based aggregation. The first application involves aggregation of stocks (with respect to the NAICS's sector-industry tree) based on mean log-volatility. The second application aggregates neighborhoods of New York City (with respect to a geographically based hierarchy) based on a regression vector for predicting taxi drivers' monthly total fares based on the frequency of different starting locations.

Notation. For an integer p, we write $[p] = \{1, 2, \dots, p\}$. For $a, b \in \mathbb{R}$, we write $a \wedge b$ and $a \vee b$ for their minimum and maximum, respectively. We use $\boldsymbol{e}_i$ to denote the ith standard basis vector. For $\boldsymbol{x} \in \mathbb{R}^p$, we define $\|\boldsymbol{x}\|_q = \left(\sum_{j=1}^p |x_j|^q\right)^{1/q}$ for $q \geq 0$. For a set $S \subseteq [p]$, $\boldsymbol{x}_S = (x_i)_{i \in S}$ is the vector obtained by restricting the vector $\boldsymbol{x}$ to the indices in set S. We use the term "tree" throughout to denote a rooted directed tree. Given a tree $\mathcal{T}$ with leaf set $\mathcal{L}$, we write $\mathcal{T}_u$ for the subtree rooted at $u \in \mathcal{T}$ and $\mathcal{L}_u$ for its leaf set.

2. Problem Setup

2.1. A Multiple Hypothesis Testing Formulation for Aggregation

Let $\mathcal{T}$ be a known tree with p leaves, each corresponding to a coordinate of the unobserved parameter vector $\boldsymbol{\theta}^* \in \mathbb{R}^p$. We formulate the tree-aggregation task as a multiple hypothesis testing problem: To each internal (non-leaf) node u of the tree we assign a null hypothesis

$$\mathcal{H}_u^0 : \text{ All elements of } \boldsymbol{\theta}^*_{\mathcal{L}_u} \text{ have the same value,} \tag{1}$$

where $\boldsymbol{\theta}^*_{\mathcal{L}_u}$ is the subvector of $\boldsymbol{\theta}^*$ restricted to leaves of the subtree rooted at u. We observe that our choice of null hypothesis follows the usual practice that simpler models correspond to the null. Rejecting the null hypothesis $\mathcal{H}_u^0$ implies that the leaves under u should be further split into smaller groups. Given the way the hypotheses are defined, a logical constraint to impose on the output of a testing procedure is the following:

Constraint 1. The parent of a rejected node must itself be rejected.

By Constraint 1, the set of rejected nodes will then form a subtree $\mathcal{T}_{\text{rej}}$ of $\mathcal{T}$ (sharing the same root as $\mathcal{T}$), and furthermore the subtrees rooted at the leaves of $\mathcal{T}_{\text{rej}}$ represent the aggregated groups. Our goal is to develop testing procedures that result in high quality splits of the parameters. In order to measure the performance of an aggregation (or equivalently a set of splits) we propose a new criterion as follows.

False Split Rate (FSR). Suppose $\widehat{\mathcal{C}} = \{\widehat{C}_1, \dots, \widehat{C}_M\}$ is a splitting of the leaves $[p]$, and $\mathcal{C}^* = \{C_1^*, \dots, C_K^*\}$ is the true splitting. For each true group C_i^*, $i \in [K]$, we count the number of splits of C_i^* by members of $\widehat{\mathcal{C}}$, that is, $\sum_{j=1}^M \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} - 1$. Therefore, the total number of excessive (false) splits of C_i^* is

$$\sum_{i=1}^K \left(\sum_{j=1}^M \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} - 1 \right) = \sum_{i=1}^K \left(\sum_{j=1}^M \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} \right) - K,$$

while the total number of splits is $(M-1) \vee 1$. We define the *false split proportion* (FSP) and true positive proportion (interchanging $\mathcal{C}^*$ and $\widehat{\mathcal{C}}$) as

$$\begin{aligned} \text{FSP} &:= \frac{\sum_{i=1}^K \left(\sum_{j=1}^M \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} \right) - K}{(M-1) \vee 1}, \\ \text{TPP} &:= 1 - \frac{\sum_{i=1}^M \left(\sum_{j=1}^K \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} \right) - M}{K-1}. \end{aligned} \tag{2}$$

The *false split rate* (FSR) and the expected power are defined as $\text{FSR} := \mathbb{E}(\text{FSP})$, $\text{Power} := \mathbb{E}(\text{TPP})$ where the expectation is with respect to the randomness in $\widehat{\mathcal{C}}$, which in our context will depend on the p-values for the hypotheses of the form (1). In the next section we provide another characterization for FSR in the tree-aggregation context, and in Section 3 we develop a testing procedure that controls FSR at a pre-specified level $\alpha < 1$.

2.2. FSR on a Tree

While the FSR metric can be calculated for a general splitting of p objects using definition (2), in this section we focus on splittings that can be expressed as a combination of branches of $\mathcal{T}$ as explained in the previous section. Note that the structure of the tree $\mathcal{T}$ may not be faithful to the true vector $\boldsymbol{\theta}^*$. In that case, the ground truth $\mathcal{C}^*$ may be very large. We will provide an equivalent characterization of FSP in this context in terms of specific structural properties of $\mathcal{T}$.

For a testing procedure satisfying Constraint 1, the rejected nodes form a subtree $\mathcal{T}_{\text{rej}}$ of $\mathcal{T}$. We define $\deg_{\mathcal{T}}(u)$ as the (out) degree of node u on tree $\mathcal{T}$ (the number of children of node u); similarly, $\deg_{\mathcal{T}_{\text{rej}}}(u)$ is the degree of node u on the subtree $\mathcal{T}_{\text{rej}}$. We use $\mathcal{F}$ as the set of false rejections in $\mathcal{T}$. Lastly, we define $\mathcal{B}^*$ as the set of nodes whose leaf sets correspond to the true aggregation, that is, $\mathcal{B}^*$ is such that $\mathcal{C}^* = \{\mathcal{L}_u \mid u \in \mathcal{B}^*\}$. This characterization of $\mathcal{C}^*$ stems from the assumption that the true

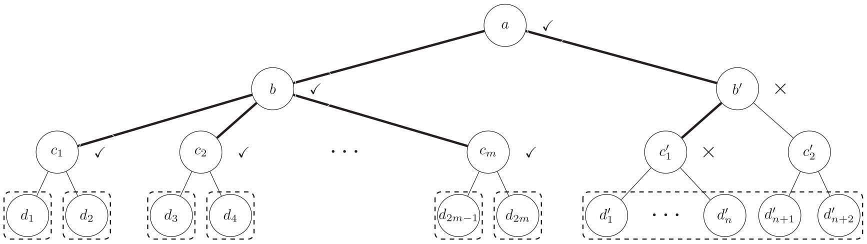


Figure 1. An example of $\mathcal{T}$ with $2m+n+2$ leaves and $3m+n+7$ nodes in total. The dashed boxes show the true aggregation of the leaves, $\mathcal{C}^*$, into $K=2m+1$ groups, with $\mathcal{B}^*=\{d_1,\ldots,d_{2m},b'\}$. The thicker edges and the nodes they connect form $\mathcal{T}_{\text{rej}}$, with ✓'s marking true rejections and ×'s marking false rejections $\mathcal{F}$. The rejections correspond to an estimated aggregation with $M=2m+n+1$ groups: $\{d_1\},\ldots,\{d_{2m}\},\{d_1'\},\ldots,\{d_n'\},\{d_{n+1}',d_{n+2}'\}$.

aggregation is among the partitions allowed by the tree. Figure 1 provides an example showing $\mathcal{T}$, $\mathcal{T}_{\text{rej}}$, $\mathcal{C}^*$, $\mathcal{B}^*$, and $\mathcal{F}$.

Our next lemma characterizes the number of false splits and the total number of splits in terms of the tree $\mathcal{T}$ and its subtree $\mathcal{T}_{\text{rej}}$. By virtue of this lemma we have an alternative characterization of FSP (and FSR), which is more amenable to analysis.

Lemma 2.1. Define V and R as follows:

$$V := \sum_{u\in\mathcal{F}}\left(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u)\right)-\left|\mathcal{B}^*\cap\mathcal{F}\right|,$$
$$R := \max\left\{\sum_{u\in\mathcal{T}_{\text{rej}}}\left(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u)\right)-1,1\right\}. \quad (3)$$

Then V and R quantify the number of false splits and the total number of splits, respectively. Consequently, we have FSP $=V/R$ and FSR $=\mathbb{E}\left(V/R\right)$, where FSP and FSR are defined as in Section 2.1.

The notation of V and R is purposely chosen to match what is commonly used in defining FDR. Indeed, it is natural to ask how the FSR relates to the FDR, and, perhaps most crucially, why one would not simply use the FDR in this situation. The next section addresses these questions and emphasizes why FSR is necessary.

2.3. Why FSR Is Needed

We begin with developing a better understanding of the relationship between the FSR and the FDR. The following lemma establishes that these quantities are in fact identical in the special case that $\mathcal{T}$ is a binary tree.

Lemma 2.2. For a binary tree, the quantities V and R given by (3) can be simplified as $V=|\mathcal{F}|$ and $R=\left|\mathcal{T}_{\text{rej}}\right|$. Therefore, FSP $=|\mathcal{F}|\,/\left|\mathcal{T}_{\text{rej}}\right|$ and FSR $=$ FDR $:=\mathbb{E}\left(|\mathcal{F}|/|\mathcal{T}_{\text{rej}}|\right)$.

We defer the proofs for Lemmas 2.1 and 2.2 to Appendix B. While the above result is conceptually helpful in that it ties the FSR to preexisting work on the FDR, it focuses on a special case that does not represent many common situations we care about in practice. Whether performing aggregation using taxonomic trees in biology (Bien et al. 2021) or using the standard industrial classification system in business (US OMB (2017), considered in Section 6), we are often interested in aggregation on nonbinary trees. The FSR and FDR can in fact be quite different for general trees. In such cases, FSR is precisely tied to the error measure we actually care about in practice, while FDR is not. The key distinction is apparent in the quantity from (3), $\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u)$, which counts the number of additional splits due to rejecting $\mathcal{H}_u^0$. The reason for this difference is that FSR is focused on the clustering that results from an aggregation procedure whereas FDR is focused on the decisions made at the internal nodes of the tree.

To demonstrate how different FSP and FDP can be from each other, we return to the example given in Figure 1. Since two of the $m+4$ rejected nodes are false rejections, we have FDP $=2/(m+4)$. By contrast, the rejections correspond to an estimated aggregation with $M=2m+n+1$ groups, created by $R=2m+n$ splits, and $V=n$ of these splits were false splits, meaning that FSP $=n/(2m+n)$. To understand the practical distinction between a procedure controlling FDP versus FSP, imagine $m=40$ and $n=80$. In such a situation, the FDP ≈ 0.045 while the FSP $=0.5$.

This very large FSP accurately reflects the fact that the estimated aggregation $\widehat{\mathcal{C}}$ with 161 groups is an extreme over-splitting of the true $\mathcal{C}^*$, which has only 81 groups. In particular, the rejection of the c_1' node with its $n=80$ children is a serious error from the standpoint of aggregation accuracy. The FDP, by contrast, ignores the tree structure and rather considers every false rejection as equally bad. While in other problems this may be a sensible assumption, in the aggregation problem considered in this article it is clearly not. This is because a false split at a high-degree node can create a large number of false clusters, which is undesirable in the clustering setting. In Appendix F, we show that a similar distinction between FSP and FDP can occur on trees coming from real data applications.

One might ask whether one can avoid using the FSP by turning a nonbinary tree into a binary one and then simply using FDR (since by Lemma 2.2 it is the same as FSR relative to this new tree). To do so, one would need to take each nonbinary node u and its children and replace this subtree with a binary subtree having u as root and its children as leaves. However, such an approach is problematic as there are many possible binary trees that could be formed, and different choices for this arbitrary tree structure would lead to very different procedures. (This is analogous to attempting to test an ANOVA hypothesis with an arbitrarily-ordered sequence of pairwise t-tests rather than with the standard F test.) Returning to the Figure 1 example, the single p-value at c_1' would have to be replaced with 90 p-values, and the interpretation of each of these p-values and the order in which they are tested would depend on the arbitrary tree structure created. Therefore, we are left with FSR as the target error measure to control. In the next section we introduce a multiple testing procedure for controlling the FSR. In light of Lemma 2.2, in the special case of a binary tree, where FSR = FDR, our procedure also controls the FDR, and we compare our method to other existing methods that control FDR in Section 3.3.

3. Hierarchical Aggregation Testing with FSR Control

So far we have defined the metric FSR to measure the quality of a splitting of leaves and proposed an alternate characterization of it in terms of the structure of the rejected (and false rejected) nodes as in Lemma 2.1. In this section, we introduce a new multiple testing procedure to test the null hypotheses $\mathcal{H}_u^0$, starting from the root and proceeding down the tree. The procedure assumes that each non-leaf node u has a p-value that is super-uniform under $\mathcal{H}_u^0$, that is

$$\mathbb{P}(p_u \le t) \le t \quad \text{for all } t \in [0,1]\,. \tag{4}$$

Later, in Section 4, we discuss how to construct such p-values for two statistical applications.

We call our multiple testing procedure HAT, shorthand for *hierarchical aggregation testing*, as the parameters in the returned splits can be aggregated together to improve model interpretability and in some cases improve the predictive power of the model. The HAT procedure controls the FSR both for independent p-values (Section 3.1) and under arbitrary dependence of the p-values (Section 3.2).

The hypotheses defined in (1) are indeed intersection hypotheses, that is,

$$\mathcal{H}_u^0 \text{ holds } \Rightarrow \mathcal{H}_v^0 \text{ holds for } \forall v \in \mathcal{T}_u, \tag{5}$$

where $\mathcal{T}_u$ is the subtree rooted at node u. In other words, the parent of a non-null node must be non-null, and if a node is null then every child of it is null as well. This property motivates us to use a top-down sequential testing algorithm on the tree that honors Constraint 1.

Before describing the HAT algorithm, we establish some notation. We sometimes write $\mathcal{H}_{d,u}^0$ to make it explicit that node u is at depth d of the tree, where the depth of a node is one plus the length of the unique path that connects the root to that node (the root is at depth 1). We also use $\mathcal{T}^d$ for the set of non-leaf nodes at depth d of $\mathcal{T}$.

The testing procedure runs as follows. Let α be our target FSR level. Starting from the root node, at each level d we only test hypotheses at the nodes whose parents are rejected. The test levels for hypotheses are determined by a step-up threshold function so that the test level at each hypothesis $\mathcal{H}_{d,u}^0$ depends on the number of leaves under this node $|\mathcal{L}_u|$, the target level α, the maximum node degree denoted by Δ, and the number of splits made in previous levels, denoted by $R^{1:(d-1)}$. The details of our HAT procedure are given in Algorithm 1, and depend on node-specific thresholds $\alpha_u(r)$, both explicitly and through the function

$$R^d(r) := \sum_{u \in \mathcal{T}^d} \mathbb{1}\{p_u \le \alpha_u(r)\}(\deg_{\mathcal{T}}(u) - 1). \tag{6}$$

We next give some intuition for the quantity r_d^* that appears in Step 2 of Algorithm 1. In the threshold function $\alpha_u(r)$, r is a free parameter; however, we would like for the argument used in the threshold function to correspond to the actual number of rejections that have occurred previously. The definition of r_d^* ensures this interpretation. To further elaborate, observe that $R^d(r)$ counts the additional splits of the leaves that result due to the rejected nodes in depth d, assuming that the threshold level $\alpha_u(r)$ is used. In our analysis, we prove the following self-consistency property: $R^d(r_d^*) = r_d^*$. In words, using r_d^* to test the nodes in $\mathcal{T}^d$ (node u to be tested at level $\alpha_u(r_d^*)$) gives us r_d^* additional splits of the leaves, and therefore the update rule for $R^{1:d}$ in line 3 of the algorithm ensures that this quantity counts the number of splits formed from testing nodes in depth $1, \ldots, d$.

Algorithm 1 *Hierarchical Aggregation Testing (HAT) Algorithm*

```
Require: : FSR level $\alpha$, Tree $\mathcal{T}$, p-values $p_u$ for $u \in \mathcal{T} \setminus \mathcal{L}$.
Ensure: : Aggregation of leaves such that the procedure controls
    FSR.
    initialize $\mathcal{T}_{\text{rej}}^1 = \{\text{root}\}$, $R^{1:1} = \deg_{\mathcal{T}}(\text{root}) - 1$.
1: repeat
2:     From depth $d = 2$ to maximum depth $D$ of the tree $\mathcal{T}$,
       perform hypothesis testing on each node in $\mathcal{T}^d$. Compute
       $r_d^*$ as
           $r_d^* = \max\left\{r \ge 0 :\ \ r \le R^d(r)\right\}$,
       where $R^d(r)$ is defined in (6), with threshold function
       $\alpha_u(r)$ given by (7) (for case of independent p-values) or
       (10) (under general dependence among p-values). Reject
       the nodes in the set $\mathcal{T}_{\text{rej}}^d = \{u \in \mathcal{T}^d : p_u \le \alpha_u(r_d^*)\}$.
3:     Update $\mathcal{T}_{\text{rej}}^{1:d} = \mathcal{T}_{\text{rej}}^{1:(d-1)} \cup \mathcal{T}_{\text{rej}}^d$, and $R^{1:d} = R^{1:(d-1)} + r_d^*$.
4: until No node in the current depth has a rejected parent or
    $d = D$.
```

3.1. Testing with Independent p-values

While in general one might expect the p-values in a tree-structured setting to be dependent, in Section 4.1 we consider a statistical application where the p-values are independent. For this reason, and for the sake of simplicity, we start by considering the case in which the p-values are independent.

Assuming that the node p-values p_u are independent, the threshold function $\alpha_u(r)$ used for testing $\mathcal{H}_{d,u}^0$ is defined as

$$\begin{aligned}\alpha_u(r) = {} & \mathbb{1}\{\text{parent}(u) \in \mathcal{T}_{\text{rej}}^{d-1}\}\,\frac{1}{\Delta} \\ & \times \frac{\alpha|\mathcal{L}_u|(R^{1:(d-1)} + r)}{p(1 - \frac{1}{\Delta^2})\hbar_{d,r} + \alpha|\mathcal{L}_u|(R^{1:(d-1)} + r)}\,,\end{aligned} \tag{7}$$

where $\hbar_{d,r}$ is the partial harmonic sum given by

$$\hbar_{d,r} = 1 + \sum_{m=R^{1:(d-1)}+r+1}^{p-1-\left(\sum_{u\in\mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d| - r\right)} \frac{1}{m}\,. \tag{8}$$

To understand the lower and upper bounds in the summation that defines $\hbar_{d,r}$, consider the case when $r = r_d^*$. The lower bound corresponds to (one more than) the number of splits that have occurred so far in the algorithm; likewise, the upper bound corresponds to the maximal possible increase in the number of splits at this level. For more on $\hbar_{d,r}$, we refer the reader to the proof of Proposition A.2 in Section C of the appendix.

Theorem 3.1. Consider a tree with maximum node degree Δ and suppose that for each node u in the tree, under the null hypothesis $\mathcal{H}_u^0$, the p-value p_u is super-uniform (see (4)). Further, assume that the p-values for the null nodes are independent from each other and from the non-null p-values. Then using Algorithm 1 with threshold function (7) to test intersection hypotheses $\mathcal{H}_u^0$ controls FSR under the target level α.

The proof of Theorem 3.1 is given in Section A.1 of the appendix and uses a combination of different ideas. At the core of the proof is a "leave-one-out" technique to decouple the quantities V and R. Using this technique together with the self-consistency property discussed after (6) and intricate probabilistic bounds in terms of structural properties of $\mathcal{T}$, we prove that FSR is controlled at the pre-assigned level α.

A few remarks are in order regarding the testing threshold $\alpha_u(r)$. From its definition, we have $\alpha_u(r) = 0$ if the parent hypothesis of u is not rejected. Also note that since the testing is done in a downward manner, the event $\{\text{parent}(u) \in \mathcal{T}_{\text{rej}}^{d-1}\}$ is observed by the time the node u is tested. Also note that as we reject more hypotheses early on, the burden of proof reduces for the subsequent hypotheses, because $\alpha_u(r)$ is increasing in $R^{1:(d-1)}$. This trend is similar to FDR control methods (e.g., Benjamini and Hochberg 1995; Javanmard and Montanari 2018b). We also observe that $\alpha_u(r)$ is increasing in $|\mathcal{L}_u|$. For the nodes at upper levels of the tree, this is crucially useful as $R^{1:(d-1)}$ is small for these nodes, while $|\mathcal{L}_u|$ is large and compensates for it in the threshold function.

Our next theorem is a generalization of Theorem 3.1 to the case that the null p-values distribution deviates from a super-uniform distribution. We will use Theorem 3.2 to control FSR in Section 4.2 where we aim to aggregate the features in a linear regression setting. As we will discuss, for this application we suggest to construct the p-values using a debiasing approach, which results in p-values that are asymptotically super-uniform (as the sample size n diverges).

Theorem 3.2. Consider a tree with maximum node degree Δ and suppose that for each non-leaf node u in the tree, under the null hypothesis $\mathcal{H}_u^0$, the p-value p_u satisfies $\mathbb{P}(p_u \le t) \le t + \varepsilon_0$ for all $t \in [0,1]$, for a constant $\varepsilon_0 > 0$. Further, assume that the p-values for the null nodes are independent from each other and from the non-null p-values. Consider running Algorithm 1 to test intersection hypotheses $\mathcal{H}_u^0$ with the threshold function

$$\alpha_u(r) = \mathbb{1}\{\text{parent}(u) \in \mathcal{T}_{\text{rej}}^{d-1}\} \times \left\{ \frac{1}{\Delta} \, \frac{\alpha|\mathcal{L}_u|(R^{1:(d-1)}+r)}{p(1-\frac{1}{\Delta^2})\hbar_{d,r} + \alpha|\mathcal{L}_u|(R^{1:(d-1)}+r)} - \varepsilon_0 \right\}. \tag{9}$$

Then, FSR is controlled under the target level α.

3.2. Testing with Arbitrarily Dependent p-values

Theorems 3.1 and 3.2 assume that the null p-values are independent from each other and from the non-null p-values. To handle arbitrarily dependent p-values, we propose a modified threshold function:

$$\alpha_u(r) = \mathbb{1}\{\text{parent}(u) \in \mathcal{T}_{\text{rej}}^{d-1}\} \, \frac{\alpha|\mathcal{L}_u| \cdot \beta_d(R^{1:(d-1)}+r)}{p(\Delta - \frac{1}{\Delta})(D-1)}, \tag{10}$$

where $\beta_d(\cdot)$ is a reshaping function of the form

$$\beta_d(R^{1:(d-1)}+r) = \frac{R^{1:(d-1)}+r}{\sum_{k=d(\delta-1)}^{\sum_{u\in\mathcal{T}^d}\deg_{\mathcal{T}}(u)} \frac{1}{k}}, \tag{11}$$

and δ is the minimum node degree in $\mathcal{T}\setminus\mathcal{L}$. It is straightforward to see that the reshaping function is lowering the test thresholds compared to the independent p-values case, making the testing procedure more conservative to handle general dependence among p-values. In the next theorem, we show that with the reshaped testing threshold FSR is controlled for arbitrarily dependent p-values. In addition, we prove the next result in the more general case in which the p-values may be approximately super-uniform (as in Theorem 3.2).

Theorem 3.3. Consider a tree with maximum node degree Δ and minimum node degree δ, and suppose that for each non-leaf node u in the tree, under the null hypothesis $\mathcal{H}_u^0$, the p-value p_u satisfies $\mathbb{P}(p_u \le t) \le t + \varepsilon_0\,, \quad$ for all $t \in [0,1]\,$, for a constant $\varepsilon_0 > 0$. The p-values for the nodes can be arbitrarily dependent. Consider running Algorithm 1 to test the hypotheses $\mathcal{H}_u^0$ with threshold function given by

$$\alpha_u(r) = \mathbb{1}\{\text{parent}(u) \in \mathcal{T}_{\text{rej}}^{d-1}\} \times \left\{ \frac{\alpha|\mathcal{L}_u| \cdot \beta_d(R^{1:(d-1)}+r)}{p(\Delta - \frac{1}{\Delta})(D-1)} - \varepsilon_0 \right\}, \tag{12}$$

with the reshaping function $\beta_d(\cdot)$ of (10). Then, FSR is controlled under the target level α.

For the special case of exact super-uniform p-values (i.e., $\varepsilon_0 = 0$), this theorem can be perceived as a generalization of Theorem 3.1 to the case of arbitrarily dependent p-values.

The proof of Theorem 3.3 builds upon a lemma from Blanchard and Roquain (2008) on dependency control of a pair of nonnegative random variables. We refer to Section A.3 of the appendix for further details and the complete proof.

3.3. A Few Remarks on HAT

In Section 2.3 we discussed the relevance of the proposed FSR metric to assess the quality of an aggregation, compatible with the given tree structure. We also discussed that for nonbinary trees, FSR and FDR could be very different measures. Nonetheless, for the special case of a binary tree, we showed in Lemma 2.2 that FSR and FDR are equivalent. In this section, we would like to understand how well HAT performs as an FDR control method on binary trees. To this end, we compare HAT with a testing procedure proposed by Lynch and Guo (2016) to control FDR in the hierarchical testing context. Their method, which we refer to as LG, corresponds to Algorithm 1 with some modifications. First, their thresholds are

$$\alpha_u(r) = \alpha \frac{|\mathcal{L}_u(\widetilde{\mathcal{T}})|}{|\mathcal{L}_{\text{root}}(\widetilde{\mathcal{T}})|} \frac{m_u(\widetilde{\mathcal{T}}) + R^{1:(d-1)} + r - 1}{m_u(\widetilde{\mathcal{T}})}, \qquad (13)$$

where $\widetilde{\mathcal{T}}$ is the tree in which we take $\mathcal{T}$ and remove the leaves, $m_u(\widetilde{\mathcal{T}})$ is the number of descendants of node u in $\widetilde{\mathcal{T}}$, $|\mathcal{L}_u(\widetilde{\mathcal{T}})|$ is the number of leaves in $\widetilde{\mathcal{T}}$ that descend from u. Also, they initialize $R^{1:1} = 1$ and, instead of (6), they take $R^d(r) = \sum_{u \in \widetilde{\mathcal{T}}^d} \mathbb{1}\left\{p_u \le \alpha_u(r)\right\}$.

In our numerical experiment, see Figure 5 (right three panels), we observe that on deep binary trees HAT achieves higher power than LG, while being more conservative and achieving lower FDR. This observation can be explained by going over the details of the proof technique used for showing the FDR control for the LG method (Lynch and Guo 2016, Theorem 1). In the proof of this result, it is shown that for each hypothesis u, $\mathbb{E}\left(V(\mathcal{T}_u)/R\right) \le \alpha|\mathcal{L}_u(\widetilde{\mathcal{T}})|/|\mathcal{L}_{\text{root}}(\widetilde{\mathcal{T}})|$ where $V(\mathcal{T}_u)$ is the number of false rejections in $\mathcal{T}_u$, the subtree rooted at node u. In deriving this bound, a chain of inequalities is used which becomes tight only if $V(\mathcal{T}_u) = R(\mathcal{T}_u) = |\mathcal{T}_u|$, that is, all the hypotheses in the subtree $\mathcal{T}_u$ are falsely rejected. Obviously this becomes a very loose bound for nodes far from the leaves, which explains why the LG method can be at a disadvantage for deep trees. In contrast, in the analysis of HAT we use a leave-one-out technique and for every fixed subtree of $\mathcal{T}_u$ we bound the probability of rejecting that tree, which is tighter than assuming all nodes of $\mathcal{T}_u$ are rejected.

The other remark we would like to make is on the harmonic term $\hbar_{d,r}$ in the expression of thresholds, given by (7). Its justification is different from that of the common adjustment factor in FDR control methods, such as Benjamini and Yekutieli (2001), which accounts for general dependency among p-values. For HAT, the harmonic term is needed even in the case of independent p-values. The reason is due to the proof technique, which we briefly explain here, and we refer to Section A.1 for more details. In our proof, we write FSR as a summation over nodes $a \in \mathcal{B}^*$. We then treat each of the summands separately via a leave-one-out technique, where we set the p-values on the rejected subtree $\mathcal{T}_{a,\text{rej}}$ of $\mathcal{T}_a$ to zero and to one on $\mathcal{T}_a \backslash \mathcal{T}_{a,\text{rej}}$. We then bound the corresponding summand conditional on $\mathcal{P}^c_{\mathcal{T}_a} = \{p_u : u \notin \mathcal{T}_a\}$. When we calculate the expectation with respect to $\mathcal{P}^c_{\mathcal{T}_a}$ at the last step, we will have dependency between $\mathcal{T}_{a,\text{rej}}$ and $\tilde{R}_{\mathcal{T}_{a,\text{rej}}}$ (the total number of splits after the leave-one-out step), since they both depend on the rejections in the previous levels of the tree. The harmonic term is needed to deal with this dependency, which exists even in the case of independent p-values.

4. Two Statistical Applications

Here we consider two statistical applications of tree-based aggregation. In Section 4.1, we study the problem of pruning a fixed tree based on measurements associated with its leaves. In this context, nodewise p-values are formed by one-way ANOVA tests. In Section 4.2, we study how to aggregate features with the same coefficients in a linear regression setting.

4.1. Testing Equality of Means

In this section, we consider the situation where we are given a tree $\mathcal{T}$ and a vector of measurements y_i on its leaves. The goal is to prune $\mathcal{T}$, using the variability in the y_i to guide this process. The goal of the pruning process is to make the tree as small as possible by aggregating branches whose y_i are not significantly different from each other. In our setting, the tree $\mathcal{T}$ is thought of as fixed and therefore is not dependent on y_i. This is in contrast to approaches where the data used to form the tree is also used to perform pruning, which has been considered both in unsupervised Langfelder, Zhang, and Horvath (2007); Gao, Bien, and Witten (2022); Ge and Tibshirani (2022) and supervised settings (Breiman et al. 1984; Neufeld, Gao, and Witten 2022).

In this application, we imagine that $\boldsymbol{\theta}^*$ is a vector of unknown means and that at each leaf node i of a tree $\mathcal{T}$ there is a noisy observation of the corresponding mean: $y_i = \theta^*_i + \varepsilon_i$, where the $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$ are independent. Given the y_i, we want to aggregate the leaves by testing the equality of their means. For each node $u \in \mathcal{T}$, we construct a p-value based on a one-way ANOVA test with known $\sigma > 0$,

$$p_u = 1 - F_{\chi^2_{\Delta_u - 1}}\left(\sigma^{-2} \sum_{v \in \text{child}(u)} |\mathcal{L}_v|(\bar{y}_v - \bar{y}_u)^2\right), \qquad (14)$$

where $\bar{y}_v = |\mathcal{L}_v|^{-1} \sum_{i \in \mathcal{L}_v} y_i$, and child$(u)$ is the set of children of u. Also $\Delta_u := \deg_{\mathcal{T}}(u) = |\text{child}(u)|$ and $F_{\chi^2_{\Delta_u - 1}}$ is the cdf of a $\chi^2_{\Delta_u - 1}$ random variable. We show in the following lemma that the above construction gives bona fide p-values for our testing procedure.

Lemma 4.1. The p-value defined in (14) is uniform under $\mathcal{H}^0_u$ in (1). Furthermore, for any two distinct nodes $a, b \in \mathcal{T} \backslash \mathcal{L}$, p_a and p_b are independent.

Recall that the nodewise hypotheses $\{\mathcal{H}^0_u\}_{u \in \mathcal{T} \backslash \mathcal{L}}$ are intersection hypotheses as in (5), and therefore one can apply Simes' procedure to form bona fide intersection p-values.

The Simes' p-value at node a is given by $p_{a,\text{Simes}} := \min_{1 \le k \le |\mathcal{T}_a \backslash \mathcal{L}_a|} \left(p_{(k)} \cdot |\mathcal{T}_a \setminus \mathcal{L}_a|\right)/k$, where $p_{(k)}$ is the kth smallest p-value in $\mathcal{T}_a \setminus \mathcal{L}_a$. As shown by Simes (1986), as the original p-values are independent (as per Lemma 4.1), the Simes' p-values constructed as above are super-uniform, and hence can be used to test the nodewise hypotheses. However, note that the Simes' p-values are not independent anymore, so when applying the HAT procedure, we need to use the reshaped threshold function (10).

4.2. Testing Equality of Regression Coefficients

In the regression setting, many authors have considered approaches for quantifying and controlling the error associated with variable selection (see, e.g., G'Sell, Hastie, and Tibshirani 2013; Barber and Candès 2015). However, we consider here the related challenge of *aggregating* rather than selecting features. Consider a linear model where the response variables are generated as $\boldsymbol{y} \sim \mathsf{N}(\boldsymbol{X}\boldsymbol{\theta}^*, \sigma^2\boldsymbol{I}_n)$. In many applications the features are counts data, that is, X_{ij} records the frequency of an event j

occurring in observation i. Yan and Bien (2021) note that when events rarely occur, a common practice is to remove the rare features in a pre-processing step; however, they show that when a tree is available, rare features can instead be aggregated to create informative predictors that count the frequency of tree-based unions of events. While Yan and Bien (2021) focused on predictive performance, here we focus on aggregation recovery itself by controlling FSR. To do so, we use the point estimator of Yan and Bien (2021), along with a debiasing approach to construct the nodewise p-values for our proposed testing procedure.

The Yan and Bien (2021) point estimator is the solution to the optimization problem,

$$\widehat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{2n}\left\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\right\|_2^2 + \min_{\boldsymbol{\gamma}\in\mathbb{R}^{|\mathcal{T}|}} \lambda\left(\nu\sum_{u\in\mathcal{T}\backslash\text{root}}|\gamma_u| + (1-\nu)\sum_{j=1}^{p}|\theta_j|\right) \quad \text{s.t. } \boldsymbol{\theta}=\boldsymbol{A}\boldsymbol{\gamma}\,, \tag{15}$$

where $\boldsymbol{A}\in\mathbb{R}^{p\times|\mathcal{T}|}$ encodes the tree structure with A_{ij} indicating whether leaf i is a descendant of node j. The resulting $\widehat{\boldsymbol{\theta}}$ tends to be constant on branches of the tree, leading to aggregated features.

4.2.1. Constructing p-values for the Null Hypotheses

A challenge in constructing p-values for the null hypotheses $\mathcal{H}_u^0$ given in (1) is that the distribution of the estimator $\widehat{\boldsymbol{\theta}}$ is not tractable. Moreover, due to the regularization term, this estimator is biased. We therefore use a debiasing approach.

The debiasing approach was pioneered in Javanmard and Montanari (2014), Zhang and Zhang (2014), van de Geer et al. (2014), and Javanmard and Montanari (2018a) for statistical inference in high-dimensions where the sample size is much smaller than the dimension of the features (i.e., $n \ll p$). Regularized estimators such as the lasso (Tibshirani 1996) are popular point estimators in these regimes however they are biased. The focus of the debiasing work has been on statistical inference on individual model parameters, namely constructing p-values for null hypotheses of the form $\mathcal{H}_{0,i}:\theta_i^*=0$. The debiasing approach has been extended for inference on linear functions of model parameters (Cai and Guo 2017; Cai, Cai, and Guo 2021) and also general functionals of them (Javanmard and Lee 2020). The original debiasing method can also be used to perform inference on a group of model parameters, for example constructing valid p-values for null hypothesis $\mathcal{H}_0:\boldsymbol{\theta}_A=0$ where the group size $|A|$ is fixed as $n,p\to\infty$ (see e.g., Javanmard and Montanari 2014, sec. 3.4). More recently, Guo et al. (2021) have studied the group inference problem for linear regression model by considering sum-type statistics. Namely, by considering quadratic form hypotheses, $\mathcal{H}_0:\boldsymbol{\theta}_A^\top\boldsymbol{G}\boldsymbol{\theta}_A=0$, for a positive definite matrix $\boldsymbol{G}$. They propose a debiasing approach to directly estimate the quadratic form $\boldsymbol{\theta}_A^\top\boldsymbol{G}\boldsymbol{\theta}_A$ and to provide asymptotically valid p-values for the corresponding hypotheses. The constructed p-values are valid for any group size in terms of Type-I error control. This work also discusses how by a direct application of the methodology developed in Meinshausen (2008), one can test significance of multiple groups, where the groups are defined by a tree structure. The method of Meinshausen (2008) is based on a hierarchical approach to test variables' importance. At the core, it constructs hierarchical adjusted p-values to account for the multiplicity of testing problems and controls the family wise error rate at the prespecified level. At every level of the tree, the p-value adjustment is a weighted Bonferroni correction and across different levels it is a sequential procedure with no correction but with the constraint that if a parent hypothesis is not rejected then the procedure does not go further down the tree. By comparison, our HAT algorithm controls the FSR, a very different criterion than the family wise error rate. Also HAT does not do any adjustment to p-values, and instead chooses the threshold levels in a sequential manner depending on the previous rejections and the structural properties of the tree.

Here we follow the methodology of Guo et al. (2021) to construct valid p-values for the HAT procedure, using the point estimator (15). We write $\mathcal{H}_u^0$ equivalently as $\widetilde{\mathcal{H}}_u^0:Q_u:=\boldsymbol{\theta}_{\mathcal{L}_u}^{*\top}\boldsymbol{G}_u\boldsymbol{\theta}_{\mathcal{L}_u}^*=0$, where $\boldsymbol{G}_u$ is the centering matrix and we use the shorthand $\boldsymbol{\theta}_u:=\boldsymbol{\theta}_{\mathcal{L}_u}$. To make inference on the quadratic form Q_u, we first consider the point estimator $\widehat{Q}_u:=\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u$, where $\widehat{\boldsymbol{\theta}}$ is the estimator given by (15). To debias $\widehat{Q}_u$ we first decompose the error term into $\widehat{Q}_u-Q_u=\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^{*\top}\boldsymbol{G}_u\boldsymbol{\theta}_u^*=2\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\left(\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^*\right)-\left(\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^*\right)^\top\boldsymbol{G}_u\left(\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^*\right)$. The dominating term in this decomposition is $2\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u(\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^*)$. The approach in Guo et al. (2021) is to develop an *unbiased* estimate of this term and then subtract this estimate from $\widehat{Q}_u$. Given a projection direction $\widehat{\boldsymbol{b}}$, the unbiased estimate is of the form

$$\frac{1}{n}\widehat{\boldsymbol{b}}^\top\boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{\theta}})=\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}^*-\widehat{\boldsymbol{\theta}})+\frac{1}{n}\widehat{\boldsymbol{b}}^\top\boldsymbol{X}^\top\boldsymbol{\varepsilon},$$

where $\widehat{\boldsymbol{\Sigma}}:=\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$. The idea is to find a projection direction $\widehat{\boldsymbol{b}}$ such that $\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}(\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*)$ is a good estimate for $\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u(\widehat{\boldsymbol{\theta}}_u-\boldsymbol{\theta}_u^*)$. The projection direction $\widehat{\boldsymbol{b}}$ is constructed by solving the following optimization problem:

$$\widehat{\boldsymbol{b}}=\arg\min_{\boldsymbol{b}}\ \boldsymbol{b}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{b}\quad\text{s.t.}\ \max_{\boldsymbol{\omega}\in\mathcal{C}_u}\left|\langle\boldsymbol{\omega},\widehat{\boldsymbol{\Sigma}}\boldsymbol{b}-[\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\ \ \boldsymbol{0}]^\top\rangle\right|\le\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u\|_2\lambda_n\,, \tag{16}$$

where $\mathcal{C}_u=\left\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_p,\frac{1}{\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u\|_2}[\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\ \ \boldsymbol{0}]^\top\right\}$ and λ_n is chosen to be of order $\sqrt{\log(p)/n}$. Finally the debiased estimator for Q_u is constructed as $\widehat{Q}_u^{\mathrm{d}}:=\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u+\frac{2}{n}\widehat{\boldsymbol{b}}^\top\boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{\theta}})$. Suppose that the true model $\boldsymbol{\theta}^*$ is s_0 sparse (i.e., it has s_0 nonzero entries). As shown in Guo et al. 2021 (Theorem 2) under the condition $s_0(\log p)/\sqrt{n}\to 0$, and assuming that the initial estimator $\widehat{\boldsymbol{\theta}}$ satisfies $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\|_2\le C\sqrt{s_0(\log p)/n}$ and $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\|_1\le Cs_0\sqrt{(\log p)/n}$ for some constant $C>0$, then the residual $\widehat{Q}_u^{\mathrm{d}}-Q_u$ asymptotically admits a Gaussian distribution. More specifically, $\widehat{Q}_u^{\mathrm{d}}-Q_u=Z_u+\Delta_u$ where

$$Z_u\sim\mathsf{N}(0,\mathrm{var}(\widehat{Q}_u^{\mathrm{d}})),\quad \mathrm{var}(\widehat{Q}_u^{\mathrm{d}})=\frac{4\sigma^2}{n}\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}}. \tag{17}$$

In addition, for any constant $c_1>0$, there exists a constant $c_2>0$ depending on c_1 such that

$$\mathbb{P}\left(|\Delta_u|\ge c_1(\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u\|_2+\|\boldsymbol{G}_u\|_2)\frac{s_0\log p}{n}\right)\le 2pe^{-c_2n}. \tag{18}$$

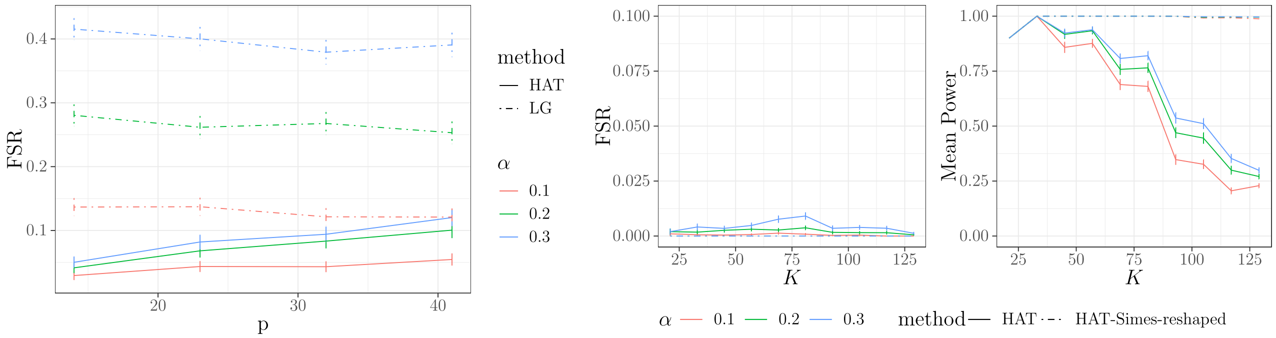


Figure 2. (Left) Plot of achieved FSR by HAT and LG on a nonbinary tree with $K = 5$ and independent p-values. LG does not control FSR under the target levels. (Center and Right) Plots of achieved FSR and mean power with ANOVA p-values on a 3-regular tree ($p = 243, \sigma = 0.3$).

The above bound state that with high probability the bias term Δ_u is of order $s_0(\log p)/n$, while $\text{var}(\widehat{Q}_u^{\text{d}})$ is of order $1/n$. Therefore, under the condition $s_0(\log p)/\sqrt{n} \to 0$ the noise term Z_u dominates the bias term Δ_u.[1]

Note that $\text{var}(\widehat{Q}_u^{\text{d}})$ involves the noise variance σ^2 (which is the same for all nodes u). Let $\widehat{\sigma}$ be a consistent estimate of σ. Then the variance of the debiased estimator $\widehat{Q}_u^{\text{d}}$ is estimated by

$$\widehat{\text{var}}_\tau(\widehat{Q}_u^{\text{d}}) = \frac{4\widehat{\sigma}^2}{n}\widehat{\boldsymbol{b}}^\top \widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}} + \frac{\tau}{n}, \tag{19}$$

for some positive fixed constant τ. The term τ/n is just to ensure that the estimated variance is at least of order $1/n$ (in the case of $\widehat{\boldsymbol{b}}^\top \widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}} = 0$), and so it dominates the bias component of $\widehat{Q}_u^{\text{d}}$. The exact choice of τ does not matter in the large sample limit ($n \to \infty$).

Using this result, we construct the two-sided p-value for the null hypothesis $\widetilde{\mathcal{H}}_u^0$ as follows: $p_u = 2\left[1 - \Phi\left(\frac{|\widehat{Q}_u^{\text{d}}|}{\sqrt{\widehat{\text{var}}_\tau(\widehat{Q}_u^{\text{d}})}}\right)\right]$, where Φ is the cdf of the standard normal distribution.

Proposition 4.2. Consider the asymptotic distributional characterization of $\widehat{Q}_u^{\text{d}}$ given by (17) and (18). Let $\widehat{\sigma} = \widehat{\sigma}(\boldsymbol{y}, \boldsymbol{X})$ be an estimator of σ satisfying, for any fixed $\varepsilon > 0$, $\lim_{n\to\infty} \mathbb{P}\left(\left|\frac{\widehat{\sigma}}{\sigma} - 1\right| \geq \varepsilon\right) = 0$. Under the condition $s_0(\log p)/\sqrt{n} \to 0$, for any fixed arbitrarily small constant ε_0 (say 0.001), there exists $n_0 > 0$ such that for all $n > n_0$, $\mathbb{P}(p_u \leq t) \leq t + \varepsilon_0$, for all $t \in [0, 1]$.

We refer to Appendix B.4 for the proof of Proposition 4.2. By virtue of Proposition 4.2, the constructed p-values satisfy the assumption of Theorem 3.3 and therefore by running the HAT procedure we are able to control FSR under the target level.

5. Simulations

In this section, we conduct simulation studies (using the `simulator` R package, Bien 2016) to understand the performance of HAT in different settings.

5.1. Testing on a Nonbinary Tree with Idealized p-values

The LG algorithm is guaranteed to control FSR due to the equivalence between FSR and FDR in the special case of a binary tree (see Lemma 2.2). However, for a nonbinary tree, the LG algorithm does not have a theoretical guarantee on FSR control. We generate a tree where the root has degree 5, and each child of the root is either a non-leaf node with degree 10 or is a leaf node; we vary the number of non-root non-leaf nodes from 1 to 4, which results in p ranging from 14 to 41. The number of true groups is fixed at 5, therefore, the root is the only non-null node. We simulate p-values for the interior nodes in the same fashion as in Section 5.3: the p-values for null nodes are simulated independently from Unif([0, 1]) and the p-values for non-null nodes are simulated independently from Beta(1, 60). An estimate of FSR is obtained by averaging FSP over 100 runs. The achieved FSR is shown in the leftmost panel of Figure 2. As expected, we observe that the HAT procedure controls FSR under each target α for all values of p, whereas the LG algorithm does not. Therefore, for aggregating leaves in general settings where the tree can be beyond binary, only our algorithm provably controls FSR under the pre-specified level. This highlights the importance of using our approach, which has guaranteed FSR control for tree-based aggregation problems with nonbinary trees.

5.2. Two Statistical Applications

5.2.1. Testing Equality of Means

In this section we apply the HAT procedure to the problem of testing equality of means. To simulate this setting, we form a balanced 3-regular tree with $p = 243$ leaves. For each K, we cut the tree into K disjoint subtrees, which leads to K nonoverlapping subgroups of leaves. We assign a value to each leaf as $y_i = \theta^*_{k(i)} + \varepsilon_i,\ \ k(i) \in \{1, \ldots, K\}, i \in \{1, \ldots, p\}$, where $k(i)$ represents the group of leaf node i and the elements of $\boldsymbol{\theta}$ are independently generated from a Unif(1, 1.5) distribution multiplied by random signs, and ε_i's from a $\mathsf{N}(0, \sigma^2)$ distribution. We simulate 100 runs by generating 100 independent $\boldsymbol{\varepsilon}$'s with the noise level set to $\sigma = 0.3$. The p-values are calculated as in (14).

By Lemma 4.1, the ANOVA p-values are independent. Thus, by Theorem 3.1, we can perform HAT using the using threshold function (7). Alternatively, we can form the bona fide p-value using Simes' procedure, and test with the reshaped threshold function that is designed for arbitrarily dependent p-values.

[1] In Guo et al. (2021), the probability bound $pe^{-c_2 n}$ was further simplified to $p^{-c'}$ since $n \gtrsim \log p$ and assuming $n, p \to \infty$.

We calculate FSR and average power by taking the average of the FSP and power over 100 runs. The center and right plots of Figure 2 demonstrate how FSR and average power change with K. Using Simes' p-values together with the reshaped thresholds achieves both lower FSR and higher power, which makes sense in this context because large effect sizes low in the tree may not translate to large effect sizes high in the tree.

5.2.2. Testing Equality of Regression Coefficients

We apply HAT to the application of testing equality of regression coefficients. We assume a high-dimensional linear model as described in Section 4.2 and generate p coefficients that take K unique values. This partition comes from leaves of disjoint subtrees of $\mathcal{T}$. We compute the p-values using the debiased method on each node as in Section 4.2.1. The details of the data generating process are described in Section E of the appendix.

For each K, we simulate 100 independent $\boldsymbol{\varepsilon}$'s. The initial estimator $\widehat{\boldsymbol{\theta}}$ that solves the optimization problem (15) is achieved by using the R package `rare` Yan and Bien (2018). The tuning parameters λ and ν are chosen by cross-validation over a 2×10 grid. We then follow the steps described in Section 4.2.1 to compute the p-values at each node. The positive constant τ in (19) is set to one and the noise level estimate $\widehat{\sigma}$ is obtained using the scaled lasso Sun and Zhang (2012) (R package `scalreg`). Figure 3 shows the empirical cdf of the p-values, obtained from the 100 realizations of the noise, at three representative nodes when $K = 57$. Among the three nodes, node #110 is a non-null node, which means $\boldsymbol{\theta}^*_{\mathcal{L}_{110}}$ contains at least two distinct values. Nodes #13 and #86 are both null nodes but at different depths on the tree. Node #86 is one of the $\mathcal{B}^*$ nodes and node #13 is a descendant of node #86. The curve of p-values at node #110 is above the diagonal line, which means the distribution has a higher density at small values than uniform distribution. On the contrary, the distribution of p-values at nodes #13 and #86 are super-uniform. The curve for a deeper level node seems to be further away from the diagonal line than its ancestor node.

The p-values generated are not necessarily independent, so we use the reshaped threshold function (10), which we have shown in theory controls FSR with arbitrarily dependent p-values. We also test with the threshold function (7), which we have not proven FSR control when the p-values are dependent. In Figure 4, we demonstrate the result for both threshold functions, varying K and α. We observe from the plots that testing with both threshold functions control FSR below each target level α. The reshaping function makes the threshold more conservative, hence, the power of the HAT test with the reshaping function is generally lower.

5.3. Testing on a Binary Tree with Idealized p-values

As we proved in Lemma 2.2, on binary trees FSR and FDR metrics become equivalent. In this section, we focus on binary trees and compare HAT with the testing procedure proposed by Lynch and Guo (2016), which controls FDR in the hierarchical testing context. We generate random trees as follows: We randomly generate p points from Unif[0, 1] and form a binary tree structure among them using hierarchical clustering. We let $K = |\mathcal{B}^*|$ be the number of true groups by cutting the tree into K disjoint subtrees with the R function `cutree`. The nodes that are the roots of the subtrees form $\mathcal{B}^*$. All non-leaf nodes in $\mathcal{B}^*$ and their non-leaf descendants are null nodes, and we generate their p-values independently from Unif([0, 1]). All ancestors of $\mathcal{B}^*$ are non-null nodes, with p-values we generate independently from Beta(1, 60). For each pair of p and K, the set of p-values are

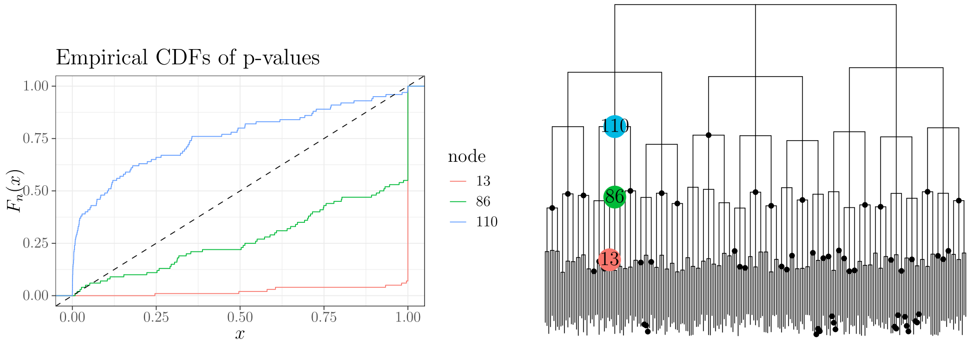


Figure 3. Plots of empirical CDFs of three nodes under the setting $n = 100, p = 243, \beta = 0.6, K = 57, \rho = 0.2, \sigma = 0.6$.

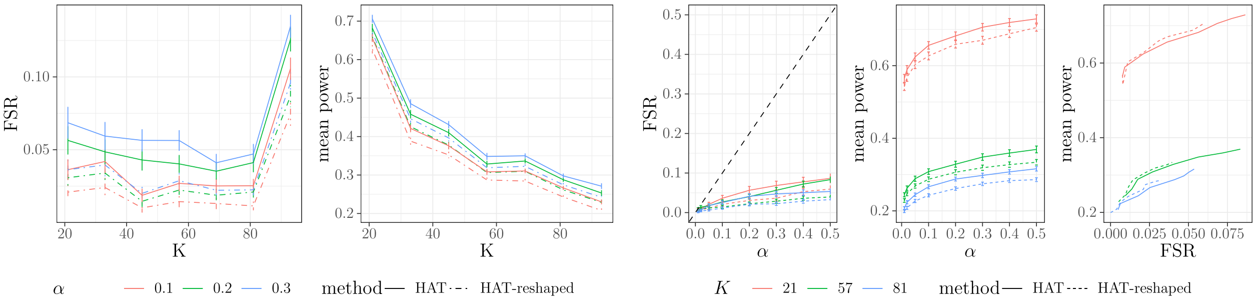


Figure 4. Plots of the achieved FSR and average power on a 3-regular tree ($n = 100, p = 243, \beta = 0.6, \rho = 0.2, \sigma = 0.6$) and p-values generated by the debiasing procedure.

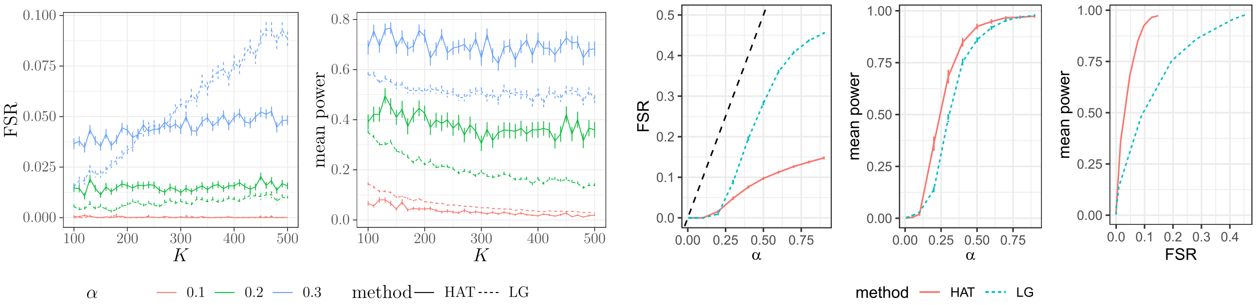


Figure 5. Plots of achieved FSR and average power by our algorithm (HAT) and Lynch and Guo's algorithm (LG), on a binary tree with $p = 1000$ leaves and independent p-values. For the right three panels, $K = 500$.

simulated independently for 100 repetitions as described above. We calculate FSP and TPP based on the aggregation of leaves that results and average over the 100 values to estimate FSR and the mean power.

The left two panels of Figure 5 show how FSR and average power change with K when p is fixed at 1000. We can see that both methods control FSR under the target α's. In terms of power, when $\alpha = 0.1$, the LG method enjoys slightly higher power. For larger α, however, the average power achieved by our HAT method is higher; the gap in power enlarges as K increases. When K is large with the tree fixed, meaning that the $\mathcal{B}^*$ nodes are at deeper levels, LG's power drops at a faster rate than ours. Indeed, for these α values, our method shows a substantial advantage when we have a deep tree and the non-null nodes appear at deeper levels of the tree.

The right three panels of Figure 5 show how achieved FSR and average power change with α in the setting where $p = 1000, K = 500$. We observe again that HAT achieves higher power than LG when α is above 0.1. From the left panel, we see that both methods are conservative in that the achieved FSR is lower than the target level α, but as evident from the rightmost panel, HAT showcases a better tradeoff between FSR and the mean power.

6. Data Examples

6.1. Application to Stocks Data

The North American Industry Classification System (NAICS; Compustat Industrial Annual Data 2015–2019) arranges companies in a hierarchy of sectors, subsectors, industry groups, industries, and national industries. This tree structure provides a principled and interpretable way of organizing a large number of companies, and it is natural to ask in what way an attribute that one can measure across individual companies may be related to this multi-level classification system. One might expect companies that are similar to each other according to NAICS to have similar values of the attribute while those that are in very different parts of the tree to have different values of the attribute. Tree-based aggregation provides a convenient approach to investigating such a question: it identifies branches of the tree whose companies could be thought of as having the same value of the attribute (in population). Doing so may provide an analyst with a simple summary of the association between the attribute and the tree structure.

To demonstrate, we consider the average daily volatility of $n = 2538$ companies' stock price computed over a five-year period, using data from the US Stock Database ©2021 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business (CRSP Stocks 2015–2019). (Appendix F provides details on preparation of this dataset.) It is plausible to imagine that companies in a shared branch of the NAICS tree may have similar volatility; however, there is no reason to think that there is a single aggregation level (such as industry group) that would apply across all companies. Aggregation provided by HAT is well suited for this goal. The tree is nonbinary, with more than 20% of nodes having at least 5 children and 10 nodes having more than 30 children, thus, as described in Section 2.3, using an FDR controlling method would not be appropriate.

To apply HAT, we first compute a p-value at every interior node of the tree by performing an F-test (eq. 8.4 of Seber and Lee 2012), for testing equality of the log-volatilities of all stocks within the subtree defined by this node. We further apply Simes' procedure to the p-values. We use HAT with the reshaped thresholds and $\alpha = 0.05$. The aggregated tree that results is shown in Figure 6 (Table 1 in Appendix F provides an alternate view). The aggregation represents a substantial simplification of the information contained in this dataset. To see this, consider that the full tree contains 702 interior nodes and 2538 leaves (which is too large to be clearly displayed in a plot). By contrast, the HAT aggregation delivers to us a great simplification: a tree with only 40 leaves. Each leaf represents an aggregated cluster of companies whose volatility is being deemed homogenous. Looking at the leaves of this aggregation tree provides a multi-level summary of the main trends of volatility across relevant sectors: 21 of the leaves are at the sector level, 8 at the subsector level, 10 at the industry group level, and one is at the company level. Two sectors are split into further clusters while other sectors remain undivided.

In looking at such a tree, one might be concerned that some of these 40 leaves actually should have been aggregated together, that is their companies appeared to have different volatilities from each other but in truth they are the same. The fact that HAT controls FSR tells us that we would only expect at most $39\alpha \approx 2$ false splits like this. By contrast, if we had used a procedure that controlled the FDR (rather than the FSR), we could end up with *many* more clusters that should not have been separated from each other. The reason, as described in Section 2.3, is that the FDR does not take into account the effect

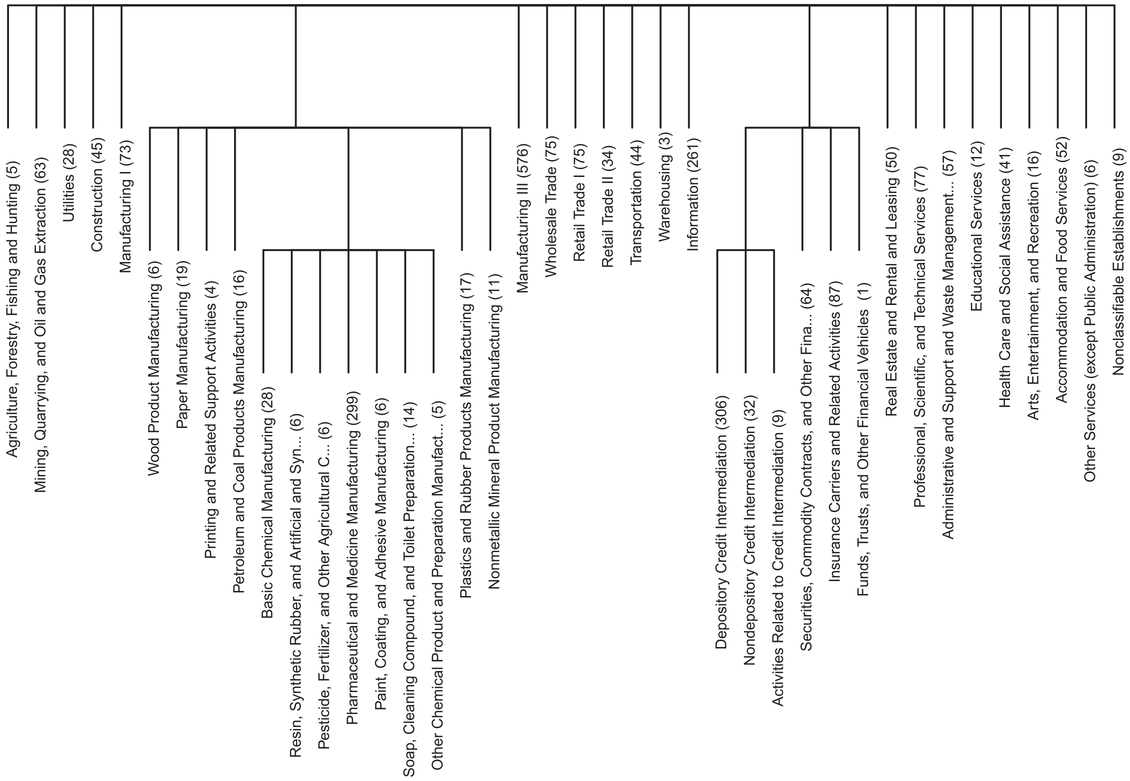

Agriculture, Forestry, Fishing and Hunting (5)
Mining, Quarrying, and Oil and Gas Extraction (63)
Utilities (28)
Construction (45)
Manufacturing I (73)
Wood Product Manufacturing (6)
Paper Manufacturing (19)
Printing and Related Support Activities (4)
Petroleum and Coal Products Manufacturing (16)
Basic Chemical Manufacturing (28)
Resin, Synthetic Rubber, and Artificial and Syn... (6)
Pesticide, Fertilizer, and Other Agricultural C... (6)
Pharmaceutical and Medicine Manufacturing (299)
Paint, Coating, and Adhesive Manufacturing (6)
Soap, Cleaning Compound, and Toilet Preparation... (14)
Other Chemical Product and Preparation Manufact... (5)
Plastics and Rubber Products Manufacturing (17)
Nonmetallic Mineral Product Manufacturing (11)
Manufacturing III (576)
Wholesale Trade (75)
Retail Trade I (75)
Retail Trade II (34)
Transportation (44)
Warehousing (3)
Information (261)
Depository Credit Intermediation (306)
Nondepository Credit Intermediation (32)
Activities Related to Credit Intermediation (9)
Securities, Commodity Contracts, and Other Fina... (64)
Insurance Carriers and Related Activities (87)
Funds, Trusts, and Other Financial Vehicles (1)
Real Estate and Rental and Leasing (50)
Professional, Scientific, and Technical Services (77)
Administrative and Support and Waste Management... (57)
Educational Services (12)
Health Care and Social Assistance (41)
Arts, Entertainment, and Recreation (16)
Accommodation and Food Services (52)
Other Services (except Public Administration) (6)
Nonclassifiable Establishments (9)

Figure 6. The aggregation tree that results from applying HAT to aggregate $n = 2538$ companies based on their volatilities, using the NAICS, a hierarchical categorization of companies based on their sectors. Leaves of this tree represent aggregated clusters (the number of companies within each cluster is given in parentheses after the name of the cluster).

that a falsely rejected node has on the clustering result. This point is underscored by a numerical experiment based on the NAICS tree given in Appendix F.

6.2. Application to New York City (NYC) Taxi Data

We apply our method of aggregating features to the NYC Yellow Taxi Trip data,[2] restricting attention to taxi trips made in December 2013. After cleaning the data, we have 13.5 million trips made by $n = 32{,}704$ taxi drivers. We take the total fare each taxi driver earned as the response variable and take the number of rides starting from each of $p = 194$ neighborhood tabulation areas (NYC Planning 2020) as the features. We form a tree with NTAs as leaves, by connecting the root to five nodes, representing the boroughs of NYC. Within each borough, we apply hierarchical clustering to the NTAs based on their geographical coordinates. This results in a tree with depth 10. The availability of taxis is not uniformly distributed across the city (see Figure 2 of Section G of the appendix) and $\boldsymbol{X}$ is a highly sparse matrix.

To aggregate neighborhood features, we perform the following procedure: with data $\boldsymbol{X}$ and $\boldsymbol{y}$, as well as the given tree structure, we first fit the penalized regression (15) to construct an initial estimate of the coefficients $\widehat{\boldsymbol{\theta}}$. The estimation is achieved by using the `rare` package with cross-validation for choosing the regularization parameters ν and λ across a grid of 5×50 values. Next, we carry out the debiasing step by solving the optimization problem (16), with the R package `quadprog`. Note that the noise level σ is unknown, which we estimate by using the scaled lasso ([Sun and Zhang 2012]; R package `scalreg`). Moreover, the positive constant τ in (19) is set to one. After constructing the p-values for each non-leaf node of the tree, we run HAT with $\alpha = 0.05$.

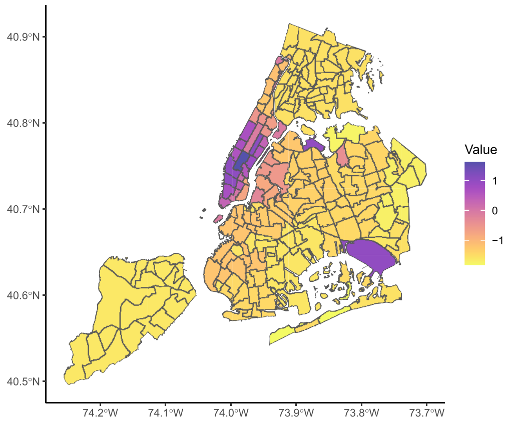
6.2.1. Aggregation Results

Our procedure results in 45 aggregated clusters, with the boroughs of Bronx and Staten Island remaining undivided. Brooklyn, Queens, and Manhattan are divided into 7, 14, and 22 subgroups, respectively. The left panel of Figure 7 shows the coefficients from performing least squares on these 45 aggregated features. Trips starting from Manhattan and parts of Queens, especially the airports, have higher coefficient values. Within Manhattan, Hell's kitchen, Times Square, and Penn Station have higher coefficient values. In Section G.1 of the appendix we show, by taking subsamples of different sizes, that reducing sample size leads to fewer rejections and therefore fewer aggregated groups.

6.2.2. Comparing Prediction Performance

To assess prediction performance achieved by our aggregated features, we hold out a random sample of 20% of the drivers as the test set, and train with the remaining 80%. We compare to the following models (each tuned via 10-fold cross validation): (i) Lasso with the original variables (`L1`); (ii) Lasso with only dense features (`L1-dense`): We drop features with $<$ 0.5% nonzeros then fit a lasso on the remaining 99 features; (iii) Least squares with clusters aggregated to the five

[2] Available at *data.cityofnewyork.us*.



Method	MSPE
`L1`	95.780
`L1-dense`	95.864
`ls-boro`	147.438
`L1-agg-h`	102.593
`Rare`	95.725
`HAT` (our method)	95.443

Figure 7. Left: Map colored with log-transformed least square coefficients from regressing fare on features from HAT's aggregation of neighborhoods of New York City. There are 45 aggregated clusters out of the 194 neighborhoods. Darker colors correspond to higher fitted coefficients. Right: Prediction performance of the six methods with the test dataset.

boroughs (`ls-boro`); (iv) Lasso with clusters aggregated at optimized height (`L1-agg-h`), and we tune (over a grid of 5 values) an extra parameter h that determines the aggregation height in the tree; (v) Rare regression proposed by Yan and Bien (2021) (`Rare`). We compute the mean squared prediction error (MSPE) of each method on the test set (see right panel of Figure 7). The `L1` and `L1-dense` methods are not aggregation-related and achieve similar performance. Both `ls-boro` and `L1-agg-h` achieve some level of aggregation but the aggregations are determined at certain heights. `L1-agg-h` has an additional tuning parameter and is therefore advantageous. Lastly, both `Rare` and our method achieve aggregation in a flexible way, and the prediction results are comparable. `Rare` selects 43 aggregation clusters while our method achieves 45 groups in total. In Section G of the appendix, we perform an additional experiment with a synthetic response (but with $\boldsymbol{X}$ and $\mathcal{T}$ from this dataset) to measure the FSR and power.

7. Conclusion

In many application domains, ranging from business and e-commerce, to computer vision and image processing, biology and ecology, the data measurements are naturally associated with the leaves of a tree which represents the data structure. Motivated by these applications, in this work we studied the problem of splitting the measurements into nonoverlapping subgroups which can be expressed as a combination of branches of the tree. The subgroups ideally express the leaves that should be aggregated together, and perceived as single entities. We formulate the task of tree-based aggregation/splitting as a multiple testing problem and introduced a novel metric called false split rate which corresponds to the fraction of splits made that were unnecessary. In addition, we proposed a procedure call HAT (and a few variants of it) to return a splitting of leaves, which is guaranteed to control the false split rate under the target level. In this article we have thought of the tree as given. However, in some cases one might be interested in learning the tree from the same data that would be used in inference. In such a case, one would need to make use of post-selection inference techniques to account for the data-driven nature of the hypotheses.

It is worth noting some of the salient distinctions of the setup considered in this article with classical hierarchical clustering. First, in hierarchical clustering the tree is cut at a fixed level, while our framework allows for more flexible summarization of the tree, with different branches cut at different depths. That is, our framework yields multi-scale resolution of the data. Second, clustering is often formulated as an unsupervised problem. In contrast, our framework can be perceived as a supervised clustering problem where labeled data are used to group the leaves by combining branches of the tree.

Supplementary Materials

The supplementary material includes all appendices, including proofs of the theoretical results and additional information about the numerical experiments.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

A. Javanmard is partially supported by the Sloan Research Fellowship in mathematics, an Adobe Data Science Faculty Research Award and the NSF CAREER Award DMS-1844481. J. Bien was supported in part by NIH Grant R01GM123993 and NSF CAREER Award DMS-1653017.

References

Barber, R. F., and Candès, E. J. (2015), "Controlling the False Discovery Rate via Knockoffs," *The Annals of Statistics*, 43, 2055–2085. [6]

Benjamini, Y., and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society*, Series B, 57, 289–300. [1,5]

Benjamini, Y., and Yekutieli, D. (2001), "The Control of the False Discovery Rate in Multiple Testing Under Dependency," *Annals of Statistics*, 29, 1165–1188. [6]

Bien, J. (2016), "The Simulator: An Engine to Streamline Simulations," arXiv preprint arXiv:1607.00021. [8]

Bien, J., Yan, X., Simpson, L., and Müller, C. L. (2021), "Tree-Aggregated Predictive Modeling of Microbiome Data," *Scientific Reports*, 11(1):1–13. [1,3]

Blanchard, G., and Roquain, E. (2008), "Two Simple Sufficient Conditions for FDR Control," *Electronic Journal of Statistics*, 2, 963–992. [5]

Bogomolov, M., Peterson, C. B., Benjamini, Y., and Sabatti, C. (2021), "Hypotheses On A Tree: New Error Rates and Testing Strategies," *Biometrika*, 108, 575–590. [2]

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984), *Classification and Regression Trees*, Boca Raton, FL: CRC Press. [6]

Cai, T., and Guo, Z. (2017), "Confidence Intervals for High-Dimensional Linear Regression: Minimax Rates and Adaptivity," *The Annals of Statistics*, 45, 615–646. [7]

Cai, T., Cai, T. T., and Guo, Z. (2021), "Optimal Statistical Inference for Individualized Treatment Effects in High-Dimensional Models," *Journal of the Royal Statistical Society*, Series B, 83, 669–719. [7]

Compustat Industrial Annual Data. (2015-2019), "Available: Standard & Poor's Compustat [01/26/2021]," Retrieved from Wharton Research Data Service. [10]

CRSP Stocks. (2015–2019), "Available: Center For Research in Security Prices," Graduate School of Business. University of Chicago [01/26/2021]. Retrieved from Wharton Research Data Service. [10]

Gao, L. L., Bien, J., and Witten, D. (2022), "Selective Inference for Hierarchical Clustering," *Journal of the American Statistical Association*, 119, 332–342. [6]

Ge, J., and Tibshirani, R. (2022), "Weakest Link Pruning of a Dendrogram," arXiv preprint arXiv:2212.05367. [6]

G'Sell, M. G., Hastie, T., and Tibshirani, R. (2013), "False Variable Selection Rates in Regression," arXiv preprint arXiv:1302.2303. [6]

Guo, Z., Renaux, C., Bühlmann, P., and Cai, T. (2021), "Group Inference in High Dimensions with Applications to Hierarchical Testing," *Electronic Journal of Statistics*, 15, 6633–6676. [2,7,8]

Heller, R., Chatterjee, N., Krieger, A., and Shi, J. (2018), "Post-selection Inference Following Aggregate Level Hypothesis Testing in Large-Scale Genomic Data," *Journal of the American Statistical Association*, 113, 1770–1783. [2]

Hu, J. X., Zhao, H., and Zhou, H. H. (2010), "False Discovery Rate Control with Groups," *Journal of the American Statistical Association*, 105, 1215–1227. [2]

Javanmard, A., and Lee, J. D. (2020), "A Flexible Framework for Hypothesis Testing in High Dimensions," *Journal of the Royal Statistical Society*, Series B, 82, 685–718. [7]

Javanmard, A., and Montanari, A. (2014), "Confidence Intervals and Hypothesis Testing for High-Dimensional Regression," *The Journal of Machine Learning Research*, 15, 2869–2909. [7]

——— (2018a), "Debiasing the Lasso: Optimal Sample Size for Gaussian Designs," *The Annals of Statistics*, 46, 2593–2622. [7]

——— (2018b), "Online Rules for Control of False Discovery Rate and False Discovery Exceedance," *The Annals of Statistics*, 46, 526–554. [5]

Katsevich, E., and Sabatti, C. (2019), "Multilayer Knockoff Filter: Controlled Variable Selection at Multiple Resolutions," *The Annals of Applied Statistics*, 13, 1–33. [2]

Katsevich, E., Sabatti, C., and Bogomolov,M. (2021), "Filtering the Rejection Set While Preserving False Discovery Rate Control," *Journal of the American Statistical Association*, 118, 165–176. [2]

Langfelder, P., Zhang, B., and Horvath, S. (2007), "Dynamic Tree Cut: In-depth Description, Tests and Applications," *Bioinformatics*, 24. [6]

Lynch, G., and Guo, W. (2016), "On Procedures Controlling the FDR for Testing Hierarchically Ordered Hypotheses," arXiv preprint arXiv:1612.04467. [1,5,6,9]

Meijer, R. J., and Goeman, J. J. (2015), "A Multiple Testing Method for Hypotheses Structured in a Directed Acyclic Graph," *Biometrical Journal*, 57, 123–143. [2]

Meinshausen, N. (2008), "Hierarchical Testing of Variable Importance," *Biometrika*, 95, 265–278. [2,7]

Neufeld, A. C., Gao, L. L., and Witten, D. M. (2022), "Tree-Values: Selective Inference for Regression Trees," *Journal of Machine Learning Research*, 23, 1–43. [6]

NYC Planning. (2020), "Available: "Neighborhood Tabulation Areas (Formerly "Neighborhood Projection Areas")," Retrieved from September 22, 2020. [11]

Ramdas, A., Chen, J., Wainwright, M. J., and Jordan, M. I. (2019), "A Sequential Algorithm for False Discovery Rate Control on Directed Acyclic Graphs," *Biometrika*, 106, 69–86. [1]

Seber, G. A. F., and Lee, A. J. (2012), *Linear Regression Analysis*, Hoboken NJ: Wiley. [10]

Simes, R. J. (1986), "An Improved Bonferroni Procedure for Multiple Tests of Significance," *Biometrika*, 73, 751–754. [6]

Sun, T., and Zhang, C.-H. (2012), "Scaled Sparse Linear Regression," *Biometrika*, 99, 879–898. [9,11]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [7]

US OMB. (2017), "North American Industry Classification System," *Executive Office of the President; Office of Management and Budget*. [1,3]

——— (2018), "Standard Occupational Classification Manual," *Executive Office of the President; Office of Management and Budget*. [1]

van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014), "On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models," *The Annals of Statistics*, 42, 1166–1202. [7]

Wilms, I., and Bien, J. (2022), "Tree-based Node Aggregation in Sparse Graphical Models," *The Journal of Machine Learning Research*, 23, 11078–11113. [1]

Yan, X., and Bien, J. (2018), rare: Linear model with tree-based lasso regularization for rare features. R package version 0.1.0. [9]

——— (2021), "Rare Feature Selection in High Dimensions," *Journal of the American Statistical Association*, 116, 887–900. [1,7,12]

Yekutieli, D. (2008), "Hierarchical False Discovery Rate-Controlling Methodology," *Journal of the American Statistical Association*, 103, 309–316. [1]

Zhang, C.-H., Zhang, S. S. (2014), "Confidence Intervals for Low Dimensional Parameters in High Dimensional Linear Models," *Journal of the Royal Statistical Society*, Series B, 76, 217–242. [7]

Zhong, S., Tian, L., Li, C., Storch, K.-F., and Wong, W. H. (2004), "Comparative Analysis of Gene Sets in the Gene Ontology Space Under the Multiple Hypothesis Testing Framework," in *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, CSB '04, pp. 425–435, USA, 2004. IEEE Computer Society. [2]