Controlling the False Split Rate in Tree-Based Aggregation

Simeng Shao, Jacob Bien, Adel Javanmard
Data Sciences and Operations, University of Southern California

June 30, 2024

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 paper 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 which involves aggregating regression coefficients.

Keywords: Multiple testing, false discovery rate, rare features, hierarchy

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 & Bien 2020) or counts of microbial species (Bien et al. 2021), and in graphical modeling (Wilms & Bien 2021). 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 & 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 & 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 paper) include **??**Bogomolov et al. (2017), Heller et al. (2018), Katsevich & Sabatti (2019). While these works focus on FDR control, another line of work uses hierarchical testing while controlling the family-wise error rate (Meinshausen 2008, **?**, Guo et al. 2019). Our motivation of finding the proper "resolution" in a tree-structured multiple hypothesis testing context is shared by **?**. 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 $[p] = \{1, 2, \ldots, p\}$. 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 i-th 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 $\widehat{\mathcal{C}} = \{\widehat{C}_1, ..., \widehat{C}_M\}$ is a splitting of the leaves $[p]$, and $\mathcal{C}^* = \{C_1^*, ..., C_K^*\}$ 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 $\widehat{\mathcal{C}}$, i.e., $\sum_{j=1}^{M} \mathbb{1}\{C_i^* \cap \widehat{C}_j \neq \emptyset\} - 1$. 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

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

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, i.e., $\mathcal{B}^*$ 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 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:*

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

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

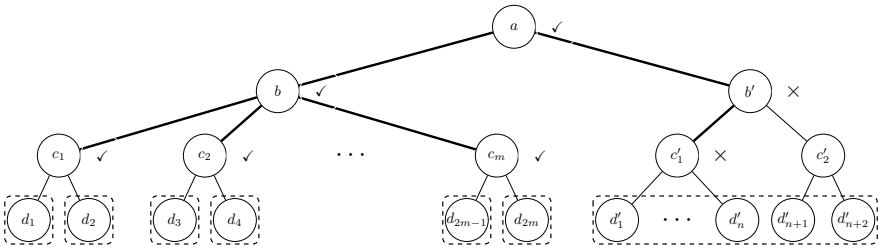


Figure 1: *An example of $\mathcal{T}$ with $2m+n+2$ leaves and $3m+n+7$ nodes in total. The dashed boxes show the true aggregation of the leaves, $\mathcal{C}^*$, into $K = 2m+1$ groups, with $\mathcal{B}^* = \{d_1, \ldots, d_{2m}, b'\}$. The thicker edges and the nodes they connect form $\mathcal{T}_{rej}$, 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: $\{d_1\}, \ldots, \{d_{2m}\}, \{d'_1\}, \ldots, \{d'_n\}, \{d'_{n+1}, d'_{n+2}\}$.*

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 $R = |\mathcal{T}_{\text{rej}}|$. Therefore,* $\text{FSP} = |\mathcal{F}| \,/\, |\mathcal{T}_{\text{rej}}|$ *and* $\text{FSR} = \text{FDR} := \mathbb{E}\left(|\mathcal{F}|/|\mathcal{T}_{\text{rej}}|\right)$.

We defer the proofs for Lemma 2.1 and Lemma 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 non-binary 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 $\mathcal{H}^0_u$. 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 $\text{FDP} = 2/(m+4)$. 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 paper 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 non-binary 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 non-binary 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 $\mathcal{H}_u^0$, i.e.

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

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, i.e.,

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

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

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

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 $1, \ldots, d$.

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.

Algorithm 1 *Hierarchical Aggregation Testing* (HAT) *Algorithm*

Require: : FSR level α, 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 $\mathcal{T}^1_{\text{rej}} = \{\text{root}\}$, $R^{1:1} = \deg_{\mathcal{T}}(\text{root}) - 1$.

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

$$r^*_d = \max\left\{r \geq 0 : \;\; r \leq R^d(r)\right\},$$

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 $\mathcal{T}^d_{\text{rej}} = \left\{u \in \mathcal{T}^d : p_u \leq \alpha_u(r^*_d)\right\}$.

3: Update $\mathcal{T}^{1:d}_{\text{rej}} = \mathcal{T}^{1:(d-1)}_{\text{rej}} \cup \mathcal{T}^d_{\text{rej}}$, and $R^{1:d} = R^{1:(d-1)} + r^*_d$.

4: **until** No node in the current depth has a rejected parent or $d = D$.

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

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

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 $\mathcal{H}^0_u$, 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 $\mathcal{H}^0_u$ 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 & Hochberg (1995), Javanmard & Montanari (2018*b*)). 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 $\mathbb{P}(p_u \leq t) \leq t + \varepsilon_0$ 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*

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

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 $\mathbb{P}(p_u \leq t) \leq t + \varepsilon_0\,, \quad$ 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*

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

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 & Roquain (2008) on dependency control of a pair of non-negative 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 non-binary 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 & 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

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

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 $R^d(r) = \sum_{u \in \widetilde{\mathcal{T}}^d} \mathbb{1}\{p_u \leq \alpha_u(r)\}$.

In our numerical experiment, cf. 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 & Guo 2016, Theorem 1). In the proof of this result, it is shown that for each hypothesis u,

$\mathbb{E}\left(V(\mathcal{T}_u)/R\right) \leq \alpha|\mathcal{L}_u(\widetilde{\mathcal{T}})|/|\mathcal{L}_{\text{root}}(\widetilde{\mathcal{T}})|$ 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 $V(\mathcal{T}_u) = R(\mathcal{T}_u) = |\mathcal{T}_u|$, i.e., 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 & Yekutieli (**?**), 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 (**???**) and supervised settings (**??**).

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 $\sigma > 0$,

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

where $\bar{y}_v = |\mathcal{L}_v|^{-1} \sum_{i \in \mathcal{L}_v} y_i$, and $\text{child}(u)$ 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* $\mathcal{H}_u^0$ *in* (1)*. Furthermore, for any two distinct nodes* $a, b \in \mathcal{T} \setminus \mathcal{L}$*,* p_a *and* p_b *are independent.*

Recall that the nodewise hypotheses $\{\mathcal{H}_u^0\}_{u \in \mathcal{T} \setminus \mathcal{L}}$ 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 $p_{a,\text{Simes}} := \min_{1 \le k \le |\mathcal{T}_a \setminus \mathcal{L}_a|} \left(p_{(k)} \cdot |\mathcal{T}_a \setminus \mathcal{L}_a| \right) / k$, 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., **??**). 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, i.e., X_{ij} records the frequency of an event j occurring in observation i. Yan & Bien (2020) 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 & Bien (2020) focused on predictive performance, here we focus on aggregation recovery itself by controlling FSR. To do so, we use the point estimator of Yan & Bien (2020), along with a debiasing approach to construct the nodewise p-values for our proposed testing procedure.

The Yan & Bien (2020) point estimator is the solution to the optimization problem,

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

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 & Montanari (2014), Zhang & Zhang (2014), van de Geer et al. (2014), Javanmard & Montanari (2018*a*) 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 et al. 2017, 2019) and also general functionals of them (Javanmard & Lee 2020). The original debiasing method can also be used to perform inference on a group of model parameters, e.g. 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 & Montanari 2014, Section 3.4)). More recently, Guo et al. (2019) 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. (2019) 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 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. (2019) 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:

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

where $\mathcal{C}_u = \left\{\boldsymbol{e}_1,...,\boldsymbol{e}_p, \frac{1}{\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u\|_2}[\widehat{\boldsymbol{\theta}}_u^\top\boldsymbol{G}_u\ \ \boldsymbol{0}]^\top\right\}$ 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. 2019, 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

$$\mathbb{P}\left(|\Delta_u| \ge c_1(\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}_u\|_2 + \|\boldsymbol{G}_u\|_2)\frac{s_0\log p}{n}\right) \le 2pe^{-c_2 n}\,, \tag{18}$$

The above bound state that with high probability the bias term Δ_u is of order $s_0(\log p)/n$, while $\mathrm{Var}(\widehat{Q}_u^{\mathrm{d}})$ 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 $\mathrm{Var}(\widehat{Q}_u^{\mathrm{d}})$ 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 $\widehat{Q}_u^{\mathrm{d}}$ is estimated by

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

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 $\widehat{Q}_u^{\mathrm{d}}$. The exact choice of τ does not matter in the large sample limit ($n\to\infty$).

[1] In Guo et al. (2019), the probability bound $pe^{-c_2 n}$ was further simplified to $p^{-c'}$ since $n \gtrsim \log p$ and assuming $n, p\to\infty$.

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

Proposition 4.2. *Consider the asymptotic distributional characterization of $\widehat{Q}_u^{\mathrm{d}}$ given by* (17) *and* (18)*. Let $\widehat{\sigma} = \widehat{\sigma}(\boldsymbol{y}, \boldsymbol{X})$ be an estimator of σ satisfying, for any fixed $\varepsilon > 0$, $\lim_{n\to\infty} \mathbb{P}\Big(\Big|\frac{\widehat{\sigma}}{\sigma} - 1\Big| \geq \varepsilon\Big) = 0$. 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 non-binary 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 non-binary 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 $\mathsf{Unif}([0, 1])$ and the p-values for non-null nodes are simulated independently from $\mathrm{Beta}(1, 60)$. 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 non-binary trees.

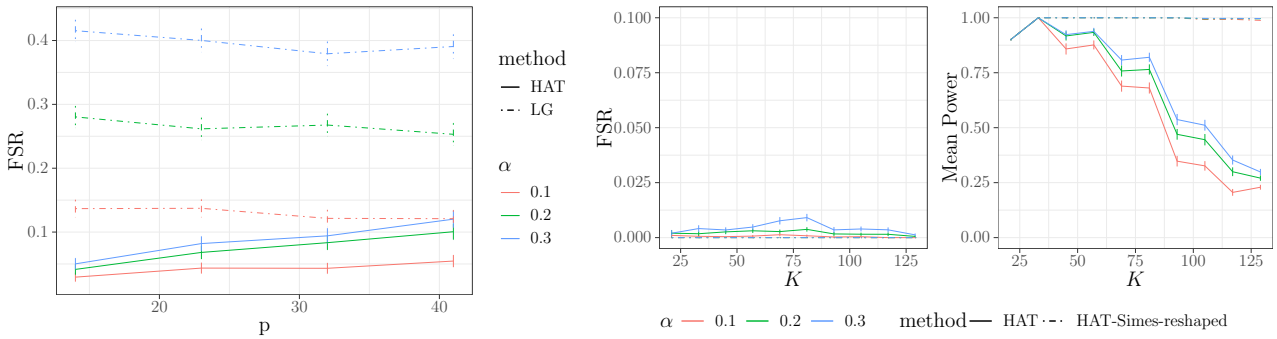


Figure 2: *(Left) Plot of achieved FSR by HAT and LG on a non-binary tree with* $K = 5$ *and independent p-values. LG does not control* FSR *under the target levels. (Center and Right) Plots of achieved FSR and mean power with ANOVA p-values on a 3-regular tree* ($p = 243, \sigma = 0.3$).

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 non-overlapping subgroups of leaves. We assign a value to each leaf as $y_i = \theta^*_{k(i)} + \varepsilon_i, \quad k(i) \in \{1, ..., K\}, i \in \{1, ..., p\}$, where $k(i)$ represents the group of leaf node i and the elements of $\boldsymbol{\theta}$ are independently generated from a $\mathsf{Unif}(1, 1.5)$ 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.

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

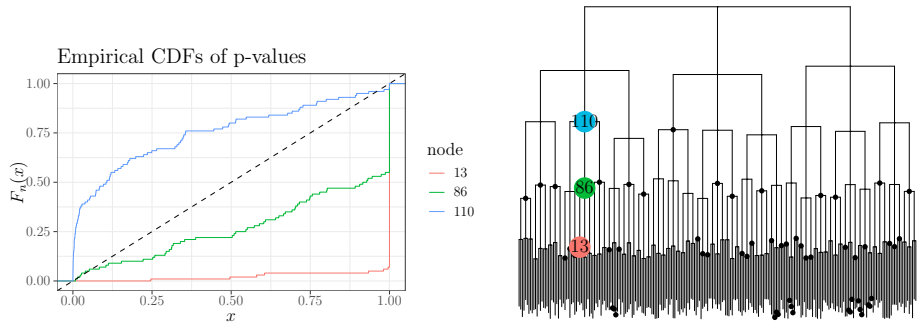


Figure 3: *Plots of empirical CDFs of three nodes under the setting* $n = 100$, $p = 243$, $\beta = 0.6$, $K = 57$, $\rho = 0.2$, $\sigma = 0.6$.

optimization problem (15) is achieved by using the R package `rare` Yan & Bien (n.d.). 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 & 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 subsection, we focus on binary trees and compare HAT with the testing procedure proposed by Lynch & Guo (2016), which controls FDR in the hierarchical testing context. We generate random trees as follows: We randomly generate p points from $\mathsf{Unif}[0, 1]$ 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

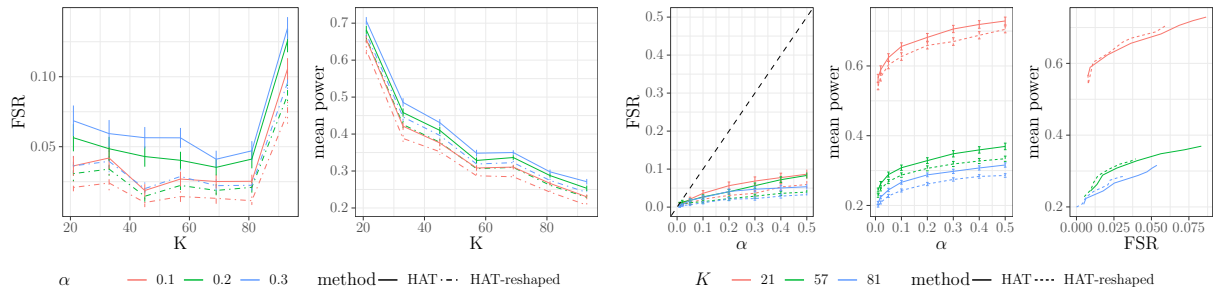


Figure 4: *Plots of the achieved FSR and average power on a 3-regular tree ($n = 100$, $p = 243$, $\beta = 0.6$, $\rho = 0.2$, $\sigma = 0.6$) and p-values generated by the debiasing procedure.*

`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 $\mathsf{Unif}([0, 1])$. All ancestors of $\mathcal{B}^*$ are non-null nodes, with p-values we generate independently from $\text{Beta}(1, 60)$. For each pair of p and K, the set of p-values are 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 right-most 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

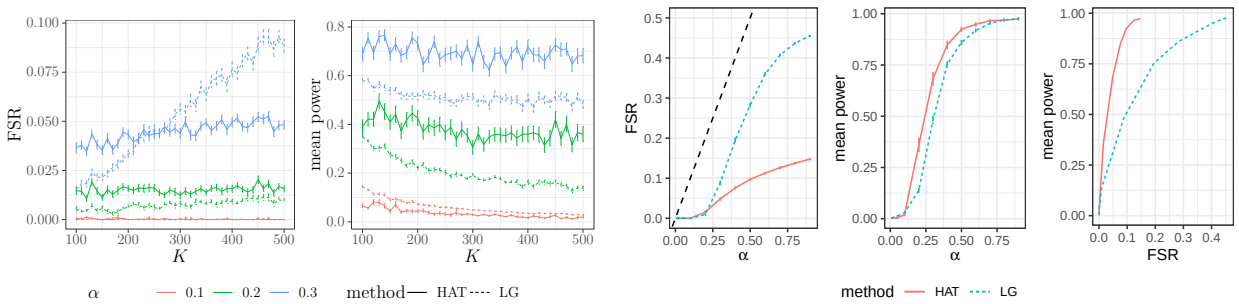


Figure 5: *Plots of achieved FSR and average power by our algorithm (HAT) and Lynch and Guo's algorithm (LG), on a binary tree with* $p = 1000$ *leaves and independent p-values. For the right three panels,* $K = 500$.

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 data set.) 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 non-binary, 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 (Equation 8.4 of Seber & 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 data set. 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

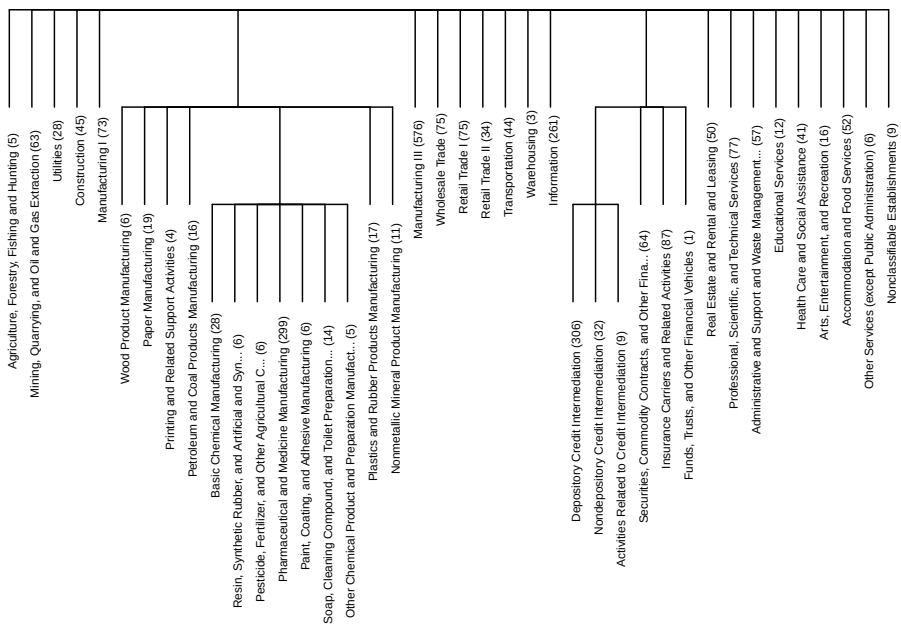


Figure 6: *The aggregation tree that results from applying* HAT *to aggregate* $n = 2538$ *companies based on their volatilities, using the NAICS, a hierarchical categorization of companies based on their sectors. Leaves of this tree represent aggregated clusters (the number of companies within each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

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, i.e. 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 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 $n = 32704$ 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

[2]available at `data.cityofnewyork.us`)

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 9 of Section G of the appendix) and $\boldsymbol{X}$ is a highly sparse matrix.

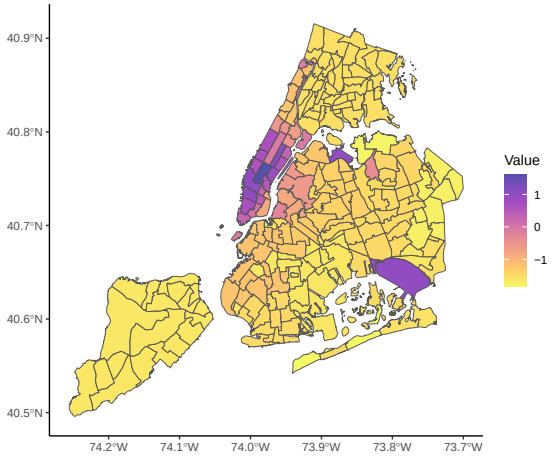
To aggregate neighborhood features, we perform the following procedure: with data $\boldsymbol{X}$ and $\boldsymbol{y}$, as well as the given tree structure, we first fit the penalized regression (15) to construct an initial estimate of the coefficients $\widehat{\boldsymbol{\theta}}$. The estimation is achieved by using the `rare` package with cross-validation across 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 & 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 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 & Bien (2020) (`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



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* 6 *methods with the test data set.*

G of the appendix, we perform an additional experiment with a synthetic response (but with $\boldsymbol{X}$ and $\mathcal{T}$ from this data set) 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 non-overlapping 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 paper 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 paper with classical hierarchical clustering. Firstly, 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. Secondly, 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.

8 Acknowledgments

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.

A Proof of main theorems

A.1 Proof of Theorem 3.1

Recall the definition of the quantities V and R:

$$V := \sum_{u \in \mathcal{F}} \Big(\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\rm rej}}(u) \Big) - |\mathcal{B}^* \cap \mathcal{F}| \,,$$

$$R := \max \left\{ \sum_{u \in \mathcal{T}_{\rm rej}} \Big(\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\rm rej}}(u) \Big) - 1, 1 \right\}.$$

Note that $R > 0$ because $\mathcal{T}_{\rm rej} \subset \mathcal{T}$ (recall that $\mathcal{T}_{\rm rej}$ does not include any leaves of $\mathcal{T}$ as there is no hypothesis associated to those nodes.) As we showed in Lemma 2.1, the false split rate can be written in terms of V and R:

$$\mathrm{FSR} = \mathbb{E}\left[\frac{V}{R \vee 1} \right].$$

For node $a \in \mathcal{B}^*$ let $\mathcal{F}_a = \mathcal{F} \cap \mathcal{T}_a$, and define the quantity V_a as follows:

$$V_a = \begin{cases} \sum_{u \in \mathcal{F}_a} \Big(\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{a,\rm rej}}(u) \Big) - 1 \,, & \text{if } \mathcal{F}_a \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

By definition $V_a \geq 0$. Indeed, from the proof of Lemma 2.1, V_a is the number of false splits in the set $\mathcal{L}_a$. Also it is easy to verify that $V = \sum_{a \in \mathcal{B}^*} V_a$.

We first show that

$$\mathbb{E}\left(\frac{V_a}{R} \right) \leq \frac{\alpha |\mathcal{L}_a|}{p} \,, \quad \text{for } a \in \mathcal{B}^*. \tag{21}$$

Denote by $S(\mathcal{T}_a)$ the set of all nonempty subtrees of $\mathcal{T}_a$ rooted at node a. We also let $V_a(\mathcal{T}')$ be the number of false splits in $\mathcal{L}_a$ when the rejection subtree is $\mathcal{T}'$, i.e.,

$$V_a(\mathcal{T}') = \sum_{u \in \mathcal{T}'} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}'}(u)) - 1\,.$$

Here we used that $a \in \mathcal{B}^*$ and therefore any rejection in $\mathcal{T}'$ is a false rejection and so $\mathcal{F}_a = \mathcal{T}'$. Define $\tilde{R}_{\mathcal{T}'}$ to be the total number of splits when we set $p_u = 0$ for $u \in \mathcal{T}'$ and $p_u = 1$ for $u \in \mathcal{T}_a \backslash \mathcal{T}'$.

Note that $\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = R$ since for $u \in \mathcal{T}_{a,\mathrm{rej}}$ the p-value p_u is already below the threshold at node u and for $u \in \mathcal{T}_a \backslash \mathcal{T}_{a,\mathrm{rej}}$, p_u is already above the threshold at that node u. Therefore, writing $\mathcal{P}^c_{\mathcal{T}_a} = \{p_u : u \notin \mathcal{T}_a\}$, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{V_a}{R}\mathbb{1}(V_a > 0)\middle|\mathcal{P}^c_{\mathcal{T}_a}\right] &= \sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \mathbb{E}\left[\frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}}\mathbb{1}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}')\middle|\mathcal{P}^c_{\mathcal{T}_a}\right] \\
&= \sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}} \cdot \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}')\,, \qquad (22)
\end{aligned}$$

where $S(\mathcal{T}_a)$ denotes the set of all nonempty subtrees of $\mathcal{T}_a$ rooted at node a, and we have used the fact that $V_a(\mathcal{T}')$ is non-random and $\tilde{R}_{\mathcal{T}'}$ is constant conditional on $\mathcal{P}^c_{\mathcal{T}_a}$.

Define $R^d(r) := \sum_{u \in \mathcal{T}^d} \mathbb{1}\{p_u \le \alpha_u(r)\}(\deg_{\mathcal{T}}(u) - 1)$. Observe that $R^d(r^*_d)$ is the additional number of splits made by the rejected nodes in depth d, going from depth $d-1$ to depth d, because the hypotheses $\mathcal{H}^0_u$ in depth d are tested at level $\alpha_u(r^*_d)$. Using our notation this can be written as the identity $R^{1:d} = R^{1:(d-1)} + R^d(r^*_d)$.

We argue that $r^*_d = R^d(r^*_d)$. To see why, note that by definition

$$r^*_d = \max\left\{0 \le r \le \sum_{u \in \mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d| : \quad r \le R^d(r)\right\}.$$

Hence, $r^*_d \le R^d(r^*_d)$ and $r^*_d + 1 > R^d(r^*_d + 1)$. Since $R^d(r)$ is an integer valued function, the fact that $R^d(r^*_d + 1) < r^*_d + 1$ implies $R^d(r^*_d + 1) \le r^*_d$. Thus, $r^*_d \le R^d(r^*_d) \le R^d(r^*_d + 1) \le r^*_d$, which gives $r^*_d = R^d(r^*_d)$, and consequently

$$R^{1:d} = R^{1:(d-1)} + r^*_d\,. \qquad (23)$$

We next continue by upper bounding the right hand side of (22). Based on our testing methodology, described in Algorithm 1, a typical node u at depth d is tested at level $\alpha_u(r^*_d)$ given by (7). We have

$$\begin{aligned}
\alpha_u(r^*_d) &= \mathbb{1}\{\mathrm{parent}(u) \in \mathcal{T}^{d-1}_{\mathrm{rej}}\}\ \frac{1}{\Delta}\ \frac{\alpha|\mathcal{L}_u|(R^{1:(d-1)} + r^*_d)}{p(1 - \frac{1}{\Delta^2})\hbar_{d,r} + \alpha|\mathcal{L}_u|(R^{1:(d-1)} + r^*_d)} \\
&= \mathbb{1}\{\mathrm{parent}(u) \in \mathcal{T}^{d-1}_{\mathrm{rej}}\}\ \frac{1}{\Delta}\ \frac{\alpha|\mathcal{L}_u|R^{1:d}}{p(1 - \frac{1}{\Delta^2})\hbar_{d,r} + \alpha|\mathcal{L}_u|R^{1:d}} \\
&= \mathbb{1}\{\mathrm{parent}(u) \in \mathcal{T}^{d-1}_{\mathrm{rej}}\}\ \frac{1}{\Delta}\ \frac{\gamma_u}{p(1 - \frac{1}{\Delta^2}) + \gamma_u}\,, \qquad (24)
\end{aligned}$$

with

$$\gamma_u := \frac{\alpha}{\hbar_{d,r}} |\mathcal{L}_u| R^{1:d} .$$

Note that $\alpha_u(r_d^*)$ is increasing in γ_u.

Lemma A.1. *Suppose that $u \in \mathcal{T}_a$ and the node a is at level d_a. Then, on the event $\{\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}'\}$ we have*

$$\gamma_u \le \frac{\alpha}{\hbar_{d,r}} |\mathcal{L}_a| \tilde{R}_{\mathcal{T}'} . \tag{25}$$

The proof of Lemma A.1 follows readily from the fact that on the event $\{\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}'\}$, we have $R^{1:d} \le \tilde{R}_{\mathcal{T}'}$. Also, since $u \in \mathcal{T}_a$ we have $|\mathcal{L}_u| \le |\mathcal{L}_a|$.

We next provide an upper bound for the thresholds $\alpha_u(r_d^*)$ for all nodes $u \in \mathcal{T}_{a,\mathrm{rej}}$, which will be useful in controlling FSR. For positive integer m, define

$$\tilde{\gamma}_{a,m} := \frac{\alpha}{\hbar_{d,r}} |\mathcal{L}_a| m . \tag{26}$$

Using Lemma A.1 and the fact $\alpha_u(r_d^*)$ is increasing in γ_u, we obtain that on the event $\{\tilde{R}_{\mathcal{T}'} = m\}$, the following holds:

$$\alpha_u(r_d^*) \le \tilde{\alpha}_{a,m} := \frac{1}{\Delta} \frac{\tilde{\gamma}_{a,m}}{p(1 - \frac{1}{\Delta^2}) + \tilde{\gamma}_{a,m}} . \tag{27}$$

We are now ready to upper bound the right hand side of (22).

Proposition A.2. *Let $a \in \mathcal{B}^*$ and assume that the null p-values are mutually independent, and independent from the non-null p-values. For our testing procedure described in Algorithm 1, the following holds true:*

$$\mathbb{E}\left[\sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}} \cdot \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \right] \le \alpha \frac{|\mathcal{L}_a|}{p} . \tag{28}$$

The proof of Proposition A.2 uses the equality (24) and the structural properties of the tree $\mathcal{T}$ tree. Its proof is deferred to Section C of the appendix. The bound (21) now follows readily by applying iterative expectation to (28).

Proof of Theorem 3.1. By using the bound (21) and noting that $V = \sum_{a \in \mathcal{B}^*} V_a$, we have

$$\mathrm{FSR} = \sum_{a \in \mathcal{B}^*} \mathbb{E}\left[\frac{V_a}{R \vee 1} \right] = \sum_{a \in \mathcal{B}^*} \mathbb{E}\left[\frac{V_a \, \mathbb{1}(V_a > 0)}{R} \right] \le \sum_{a \in \mathcal{B}^*} \frac{|\mathcal{L}_a|}{p} \alpha = \alpha.$$

The result follows. □

A.2 Proof of Theorem 3.2

Theorem 3.2 can be proved by following similar lines of the proof of Theorem 3.1 and so we omit a detailed proof here. The main difference is that in this case, the quantity $\tilde{\alpha}_{a,m}$ should be defined as

$$\tilde{\alpha}_{a,m} := \frac{1}{\Delta} \frac{\tilde{\gamma}_{a,m}}{p(1-\frac{1}{\Delta^2}) + \tilde{\gamma}_{a,m}} - \varepsilon_0 \,. \tag{29}$$

Also, the bound (53) is updated as

$$q_{u,m} = \mathbb{P}(u \in \mathcal{T}^m_{a,\mathrm{rej}}) \leq (\tilde{\alpha}_{a,m} + \varepsilon_0)^{\mathrm{depth}(u)-\mathrm{depth}(a)+1} \,, \tag{30}$$

and therefore similar to (54) we have

$$\sum_{u \in \mathcal{T}_a} q_{u,m} \leq \frac{1}{\Delta} \cdot \frac{\Delta(\tilde{\alpha}_{a,m} + \varepsilon_0)}{1 - \Delta(\tilde{\alpha}_{a,m} + \varepsilon_0)} = \frac{1}{\Delta p(1-\frac{1}{\Delta^2})} \tilde{\gamma}_{a,m} \,, \tag{31}$$

which is the same bound as in (54), albeit via a slightly different derivation and choice of threshold levels $\alpha_u(r)$. The rest of the proof would be identical to the proof of Theorem 3.1.

A.3 Proof of Theorem 3.3

Let $a \in \mathcal{B}^*$, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{V_a}{R}\cdot \mathbb{1}\{V_a>0\}\right] &= \mathbb{E}\left[\sum_{\mathcal{T}'\in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{R}\cdot \mathbb{1}\left\{\mathcal{T}_{a,\mathrm{rej}}=\mathcal{T}'\right\}\right] \\
&\le (\Delta-\frac{1}{\Delta})\sum_{\mathcal{T}'\in S(\mathcal{T}_a)} \mathbb{E}\left[\frac{|\mathcal{T}'|}{R}\cdot \mathbb{1}\left\{\mathcal{T}_{a,\mathrm{rej}}=\mathcal{T}'\right\}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{\mathcal{T}'\in S(\mathcal{T}_a)}\sum_{u\in\mathcal{T}'} \mathbb{E}\left[\frac{\mathbb{1}\left\{\mathcal{T}_{a,\mathrm{rej}}=\mathcal{T}'\right\}}{R}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a}\sum_{\mathcal{T}'\in S(\mathcal{T}_a):u\in\mathcal{T}'} \mathbb{E}\left[\frac{\mathbb{1}\left\{\mathcal{T}_{a,\mathrm{rej}}=\mathcal{T}'\right\}}{R}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{u\in\mathcal{T}_{a,\mathrm{rej}}\right\}}{R}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{p_u\le\alpha_u(r_d^*)\right\}}{R}\right] \\
&\le (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{p_u\le\alpha_u(r_d^*)\right\}}{R^{1:(d-1)}+r_d^*}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{p_u\le\frac{\alpha|\mathcal{L}_u|\beta_d(R^{1:(d-1)}+r_d^*)}{p(\Delta-\frac{1}{\Delta})(D-1)}-\varepsilon_0\right\}}{R^{1:(d-1)}+r_d^*}\right] \\
&= (\Delta-\frac{1}{\Delta})\sum_{u\in\mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{p_u+\varepsilon_0\le\frac{\alpha|\mathcal{L}_u|\beta_d(R^{1:(d-1)}+r_d^*)}{p(\Delta-\frac{1}{\Delta})(D-1)}\right\}}{R^{1:(d-1)}+r_d^*}\right], \qquad (32)
\end{aligned}$$

where the first inequality follows from Lemma D.1; the second inequality is because $R \ge R^{1:d} = R^{1:(d-1)} + r_d^*$; and, in the second-to-last line, we used the definition of threshold function $\alpha_u(r_d^*)$ given by (12). Also by the theorem assumption $p_u + \varepsilon_0$ is super-uniform because

$$\mathbb{P}(p_u+\varepsilon_0\le t) = \mathbb{P}(p_u\le t-\varepsilon_0) \le (t-\varepsilon_0)+\varepsilon_0 = t\,.$$

Next, we will use the following proposition by Blanchard & Roquain about super-uniform random variables.

Proposition A.3 (Blanchard & Roquain (2008))**.** *A couple (U, V) of possibly dependent nonnegative random variables such that U is superuniform, i.e., $\forall t \in [0,1], \mathbb{P}(U \le t) \le t$, satisfy the following inequalities*

$$\forall c>0, \mathbb{E}\left[\frac{\mathbb{1}\left\{U\le c\beta(V)\right\}}{V}\right] \le c,$$

if $\beta(\cdot)$ is a shape function of the following form

$$\beta(x) = \int_0^x t d\nu(t),$$

where ν is an arbitrary probability distribution on $(0, \infty)$, and V is arbitrary.

Letting $U = p_u + \varepsilon_0$, $V = R^{1:(d-1)} + r_d^*$, and $c = \frac{\alpha|\mathcal{L}_u|}{p(\Delta - \frac{1}{\Delta})(D-1)}$, we have

$$\begin{aligned}
(\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \mathbb{E}\left[\frac{\mathbb{1}\left\{ p_u + \varepsilon_0 \leq \frac{\alpha|\mathcal{L}_u|\beta_d(R^{1:(d-1)} + r_d^*)}{p(\Delta - \frac{1}{\Delta})(D-1)} \right\}}{R^{1:(d-1)} + r_d^*} \right] &\leq (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \frac{\alpha|\mathcal{L}_u|}{p(\Delta - \frac{1}{\Delta})(D-1)} \\
&= \frac{\alpha}{p}\left[\sum_{u \in \mathcal{T}_a} \frac{|\mathcal{L}_u|}{D-1} \right] \\
&\leq \frac{\alpha|\mathcal{L}_a|}{p},
\end{aligned}$$

where the last inequality follows from

$$\sum_{u \in \mathcal{T}_a} |\mathcal{L}_u| \leq \sum_{d=2}^{D} \sum_{u \in \mathcal{T}^d \cap \mathcal{T}_a} |\mathcal{L}_u| = \sum_{d=2}^{D} |\mathcal{L}_a| = (D-1)|\mathcal{L}_a|.$$

It is reasonable to use a measure ν that puts mass proportional to $\frac{1}{x}$ only on the values that its arguments could possibly take. By the design of the tree, we have

$$R^{1:(d-1)} + r_d^* \geq (d-1)(\delta - 1) + \delta - 1 = d(\delta - 1),$$

since at least one node has to be rejected on each depth from 1 to $d-1$ for the algorithm to carry on to depth d, and

$$R^{1:(d-1)} + r_d^* \leq \sum_{u \in \mathcal{T}^{d-1}} \deg_{\mathcal{T}}(u) - 1.$$

Therefore,

$$\beta_d(R^{1:(d-1)} + r_d^*) = \frac{R^{1:(d-1)} + r_d^*}{\sum_{k=d(\delta-1)}^{\sum_{u \in \mathcal{T}^{d-1}} \deg_{\mathcal{T}}(u) - 1} \frac{1}{k}}.$$

The rest of the proof is identical to the proof of Theorem 3.1.

B Proof of technical lemmas

B.1 Proof of Lemma 2.1

We will prove the lemma by showing that

$$\max\left\{\sum_{u\in\mathcal{T}_{\text{rej}}}(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u))-1,1\right\}=M-1. \tag{33}$$

and

$$\sum_{u\in\mathcal{F}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u)\Big)-|\mathcal{B}^*\cap\mathcal{F}|=\sum_{i=1}^{K}\left(\sum_{j=1}^{M}\mathbb{1}\{C_i^*\cap\widehat{C}_j\neq\emptyset\}\right)-K, \tag{34}$$

The proof is based on induction on the depth of the tree D.

We first prove the induction basis when $D=2$. In this case, $\mathcal{T}$ consists in one root node u_0 and its children as leaves. We therefore have only one hypothesis, $\mathcal{H}_{u_0}^0$.

- If $\mathcal{H}_{u_0}^0$ fails to be rejected, then $\mathcal{F}=\mathcal{T}_{\text{rej}}=\emptyset$, $M=1$. Both left hand side and right hand side of equation (33) are 0. For equation (34), the left hand side is clearly 0, and the right hand side is also 0 since $\sum_{i=1}^{K}\left(\sum_{j=1}^{M}\mathbb{1}\{C_i^*\cap\widehat{C}_j\neq\emptyset\}\right)-K=(\sum_{i=1}^{K}1)-K=0$.

- If $\mathcal{H}_{u_0}^0$ is rejected, we will have $M=\deg_{\mathcal{T}}(u_0)$ and $\mathcal{T}_{\text{rej}}=\{u_0\}$. Equation (33) holds because $\sum_{u\in\mathcal{T}_{\text{rej}}}(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u))-1=\deg_{\mathcal{T}}(u_0)-\deg_{\mathcal{T}_{\text{rej}}}(u_0)-1=M-1$ since $\deg_{\mathcal{T}_{\text{rej}}}(u_0)=0$. For equation (34), we consider two scenarios:

  - If $\mathcal{H}_{u_0}^0$ is true, then $K=|\mathcal{B}^*|=1$ and $\mathcal{F}=\mathcal{T}_{\text{rej}}=\{u_0\}$. So the left hand side of (34) becomes $\deg_{\mathcal{T}}(u_0)-1=M-1$, and the right hand side becomes $M-K=M-1$, hence the equality holds.
  - Otherwise $\mathcal{H}_{u_0}^0$ is false and $K=|\mathcal{B}^*|=\deg_{\mathcal{T}}(u_0)=M$, and $\mathcal{F}=\emptyset$. So the left hand side of (34) becomes 0, and the right hand side becomes $M-K=0$, hence the equality holds.

Next we proceed by proving the induction step for equation (33). Let $D>2$ be an arbitrary integer. We assume for a tree with maximum depth $\leq D-1$, identity (33) holds. We want to show that it holds for a tree with maximum depth D.

Clearly, this equation holds when the root node is not rejected, i.e., $\mathcal{T}_{\text{rej}}=\emptyset$ and $M=1$. We henceforth discuss the case that the root node is rejected. In this case, equation (33) can be simplified as

$$\sum_{u\in\mathcal{T}_{\text{rej}}}(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u))=M.$$

For a tree $\mathcal{T}$ with maximum depth D, if we remove the root node, we will be left with a forest where each tree is of maximum depth less than D. Within each tree, we have that

identity (33) holds by the induction hypothesis. We refer to the set of trees in the forest as S_{root}. Furthermore, we use $M_{\mathcal{T}'}, \mathcal{T}' \in S_{\text{root}}$ for the number of achieved groups in each such tree. Obviously,

$$\sum_{\mathcal{T}' \in S_{\text{root}}} M_{\mathcal{T}'} = M. \tag{35}$$

Therefore,

$$\begin{aligned}
&\sum_{u \in \mathcal{T}_{\text{rej}}} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\text{rej}}}(u)) \\
&= \sum_{u \in \mathcal{T}_{\text{rej}} \setminus \text{root}} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\text{rej}}}(u)) + \deg_{\mathcal{T}}(\text{root}) - \deg_{\mathcal{T}_{\text{rej}}}(\text{root}) \\
&= \sum_{\mathcal{T}' \in S_{\text{root}}} \sum_{u \in \mathcal{T}_{\text{rej}} \cap \mathcal{T}'} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\text{rej}}}(u)) + \deg_{\mathcal{T}}(\text{root}) - \deg_{\mathcal{T}_{\text{rej}}}(\text{root}) \\
&= \sum_{\substack{\mathcal{T}' \in S_{\text{root}} \\ \mathcal{T}' \cap \mathcal{T}_{\text{rej}} \neq \emptyset}} \sum_{u \in \mathcal{T}_{\text{rej}} \cap \mathcal{T}'} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}_{\text{rej}}}(u)) + \deg_{\mathcal{T}}(\text{root}) - \deg_{\mathcal{T}_{\text{rej}}}(\text{root}) \\
&= \sum_{\substack{\mathcal{T}' \in S_{\text{root}} \\ \mathcal{T}' \cap \mathcal{T}_{\text{rej}} \neq \emptyset}} M_{\mathcal{T}'} + \deg_{\mathcal{T}}(\text{root}) - \deg_{\mathcal{T}_{\text{rej}}}(\text{root}) \\
&= \sum_{\substack{\mathcal{T}' \in S_{\text{root}} \\ \mathcal{T}' \cap \mathcal{T}_{\text{rej}} \neq \emptyset}} M_{\mathcal{T}'} + \sum_{\substack{\mathcal{T}' \in S_{\text{root}} \\ \mathcal{T}' \cap \mathcal{T}_{\text{rej}} = \emptyset}} M_{\mathcal{T}'} \\
&= M,
\end{aligned}$$

where the fourth equality is by the induction hypothesis; the fifth equality holds because there are $\deg_{\mathcal{T}}(\text{root}) - \deg_{\mathcal{T}_{\text{rej}}}(\text{root})$ subtrees $\mathcal{T}' \in S_{\text{root}}$ such that $\mathcal{T}' \cap \mathcal{T}_{\text{rej}} = \emptyset$, and their $M_{\mathcal{T}'} = 1$; the last equality follows from (35). This proves the induction step and hence completes the proof of identity (33).

We next proceed to prove (34). Suppose that the induction hypothesis holds for trees with depth at most $D-1$. We want to prove it for trees of depth D. Note that this identity trivially holds when the root is not rejected, and therefore we focus on the case where the root is rejected. There are two scenarios: (1) the root is a true rejection, or (2) the root is a false rejection.

We first assume the root is a true rejection. Then we have

$$K = \sum_{\mathcal{T}' \in S_{\text{root}}} K_{\mathcal{T}'}, \tag{36}$$

where $K_{\mathcal{T}'} \geq 1$ is defined as the number of true groups in each $\mathcal{T}' \in S_{\text{root}}$.

Then the left hand side of (34) becomes

$$\begin{aligned}
&\sum_{u\in\mathcal{F}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\rm rej}}(u)\Big)-|\mathcal{B}^*\cap\mathcal{F}|\\
&=\sum_{\mathcal{T}'\in S_{\rm root}}\left(\sum_{u\in\mathcal{F}\cap\mathcal{T}'}\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\rm rej}}(u)-|\mathcal{B}^*\cap\mathcal{F}\cap\mathcal{T}'|\right)\\
&=\sum_{\mathcal{T}'\in S_{\rm root}}\left[\sum_{1\le i\le K_{\mathcal{T}'}}\left(\sum_{1\le j\le M_{\mathcal{T}'}}\mathbb{1}\{C_i^*\cap\widehat{C}_j\neq\emptyset\}\right)-K_{\mathcal{T}'}\right]\\
&=\sum_{1\le i\le K}\left(\sum_{1\le j\le M}\mathbb{1}\{C_i^*\cap\widehat{C}_j\neq\emptyset\}\right)-K,
\end{aligned}$$

where the first equality holds because $\mathcal{T}'\in S_{\rm root}$ are disjoint from each other and the root is not in $\mathcal{B}^*\cap\mathcal{F}$; the second equality follows from the induction hypothesis, and the last equality follows from (35) and (36).

For the case where the root is a false rejection, we have $K=1$, $\mathcal{B}^*=\{\text{root}\}$ and any rejection is a false rejection ($\mathcal{F}=\mathcal{T}_{\rm rej}$). We write

$$\sum_{u\in\mathcal{F}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\rm rej}}(u)\Big)-|\mathcal{B}^*\cap\mathcal{F}|=\sum_{u\in\mathcal{T}_{\rm rej}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\rm rej}}(u)\Big)-1=M-1\,,$$

where in the last step we used identity (33). On the other hand, in this case there is only one true group ($K=1$) which consists of all leaves. Therefore, any returned group $\widehat{C}_j$ will intersect with it and we get

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

Comparing the previous two equations implies that identity (34) holds for the tree $\mathcal{T}$. This completes the induction step and hence proves identity (34).

B.2 Proof of Lemma 2.2

The proof of Lemma 2.2 follows from Lemma 2.1 and that $\deg_{\mathcal{T}}(u)=2$, for all non-leaf nodes $u\in\mathcal{T}$. It suffices to show

$$\sum_{u\in\mathcal{F}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\rm rej}}(u)\Big)-|\mathcal{B}^*\cap\mathcal{F}|=|\mathcal{F}|, \tag{37}$$

and

$$\max\left\{\sum_{u\in\mathcal{T}_{\text{rej}}}(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u))-1,1\right\}=|\mathcal{T}_{\text{rej}}|. \tag{38}$$

To prove equation (37), note that if a node is falsely rejected all of its rejected children are also false rejections. Therefore, $\sum_{u\in\mathcal{F}}\deg_{\mathcal{T}_{\text{rej}}}(u)$ counts the total number of edges where both nodes of it are in $\mathcal{F}$. Hence,

$$\begin{aligned}
&\sum_{u\in\mathcal{F}}\Big(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u)\Big)-|\mathcal{B}^*\cap\mathcal{F}|\\
&=2|\mathcal{F}|-|\left\{u:u\in\mathcal{F},\text{parent}(u)\in\mathcal{F}\right\}|-|\mathcal{B}^*\cap\mathcal{F}|\\
&=2|\mathcal{F}|-|\left\{u:u\in\mathcal{F},\text{parent}(u)\in\mathcal{F}\right\}|-|\left\{u:u\in\mathcal{F},\text{parent}(u)\notin\mathcal{F}\right\}|\\
&=2|\mathcal{F}|-|\mathcal{F}|\\
&=|\mathcal{F}|.
\end{aligned}$$

Equation (38) holds trivially when $|\mathcal{T}_{\text{rej}}|=0$. When $|\mathcal{T}_{\text{rej}}|>0$, the root node is rejected, and we write

$$\begin{aligned}
&\sum_{u\in\mathcal{T}_{\text{rej}}}(\deg_{\mathcal{T}}(u)-\deg_{\mathcal{T}_{\text{rej}}}(u))-1\\
&=2|\mathcal{T}_{\text{rej}}|-\sum_{u\in\mathcal{T}_{\text{rej}}}\deg_{\mathcal{T}_{\text{rej}}}(u)-1\\
&=2|\mathcal{T}_{\text{rej}}|-(|\mathcal{T}_{\text{rej}}|-1)-1\\
&=|\mathcal{T}_{\text{rej}}|.
\end{aligned}$$

This completes the proof.

B.3 Proof of Lemma 4.1

We use the shorthand $\ell_u:=|\mathcal{L}_u|$ for a node u. Define the random vector $\boldsymbol{w}\in\mathbb{R}^{\Delta_a}$ with elements $w_u=\ell_u^{1/2}\bar{y}_u$ and the fixed unit vector $\boldsymbol{r}\in\mathbb{R}^{\Delta_a}$ with elements $r_u=(\ell_u/\ell_a)^{1/2}$. We have

$$\boldsymbol{r}^\top\boldsymbol{w}=\sum_{u\in\text{child}(a)}(\ell_u/\ell_a)^{1/2}(\ell_u^{\frac{1}{2}}\bar{y}_u)=\ell_a^{-1/2}\sum_{u\in\text{child}(a)}\ell_u\bar{y}_u=\ell_a^{1/2}\bar{y}_a, \tag{39}$$

from which it follows that

$$\sum_{u\in\text{child}(a)}\ell_u(\bar{y}_u-\bar{y}_a)^2=\sum_{u\in\text{child}(a)}(\ell_u^{1/2}(\bar{y}_u-\bar{y}_a))^2=\sum_{u\in\text{child}(a)}(w_u-r_u\boldsymbol{r}^\top\boldsymbol{w})^2=\|(\boldsymbol{I}_{\Delta_a}-\boldsymbol{r}\boldsymbol{r}^\top)\boldsymbol{w}\|^2.$$

The random vector $\boldsymbol{w}$ is multivariate normal with $\mathbb{E}[w_u]=\ell_u^{1/2}\bar{\theta}_u$, where $\bar{\theta}_u=\frac{1}{|\mathcal{L}_u|}\sum_{i\in\mathcal{L}_u}\theta_i$ is the average of parameters on the leave nodes $\mathcal{L}_u$. In addition, $\text{Cov}(\boldsymbol{w})=\sigma^2\boldsymbol{I}_{\Delta_a}$. Taking the expectation of (39) establishes that

$$\mathbb{E}[(\boldsymbol{r}\boldsymbol{r}^\top\boldsymbol{w})_u]=\mathbb{E}[r_u(\boldsymbol{r}^\top\boldsymbol{w})]=(\ell_u/\ell_a)^{1/2}\ell_a^{1/2}\mathbb{E}[\bar{y}_a]=\ell_u^{1/2}\bar{\theta}_a.$$

Under $\mathcal{H}_a$, we have $\bar{\theta}_u = \bar{\theta}_a$ and thus

$$(\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top)\boldsymbol{w} \sim \mathsf{N}\left(\boldsymbol{0}, \sigma^2(\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top)\right) ,$$

where we use the fact that $\|\boldsymbol{r}\|_2 = 1$ and so $\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top$ is a projection matrix. This establishes that

$$\sum_{u \in \text{child}(a)} \ell_u(\bar{y}_u - \bar{y}_a)^2 \sim \sigma^2 \chi^2_{\Delta_a - 1}$$

under $\mathcal{H}_a$, meaning that p_a is uniform. Now consider some node $b \neq a$. If $\mathcal{L}_a \cap \mathcal{L}_b = \emptyset$, then p_a and p_b are clearly independent (because they depend only on $\boldsymbol{y}_{\mathcal{L}_a}$ and $\boldsymbol{y}_{\mathcal{L}_b}$, respectively). Thus, it remains to consider the case that $\mathcal{L}_a \subset \mathcal{L}_b$ (i.e., a is a descendant of b). There must exist $v \in \text{child}(b)$ with $\mathcal{L}_a \subseteq \mathcal{L}_v \subset \mathcal{L}_b$. From (14), $p_b = f(\bar{y}_v, \boldsymbol{y}_{\mathcal{L}_b \setminus \mathcal{L}_v})$. Since $(\mathcal{L}_b \setminus \mathcal{L}_v) \cap \mathcal{L}_a = \emptyset$, we know that p_a is independent of $\boldsymbol{y}_{\mathcal{L}_b \setminus \mathcal{L}_v}$. It therefore remains to show that p_a is also independent of $\bar{y}_v$. To do so, observe that

$$\ell_v \bar{y}_v = \sum_{i \in \mathcal{L}_a} y_i + \sum_{i \in \mathcal{L}_v \setminus \mathcal{L}_a} y_i = \ell_a^{1/2} \boldsymbol{r}^\top \boldsymbol{w} + \sum_{i \in \mathcal{L}_v \setminus \mathcal{L}_a} y_i. \tag{40}$$

Thus,

$$\begin{aligned}
\text{Cov}\left([\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^T]\boldsymbol{w}, \bar{y}_v\right) &= \ell_v^{-1}\text{Cov}\left([\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top]\boldsymbol{w}, \ell_v \bar{y}_v\right) \\
&= \ell_a^{1/2}\ell_v^{-1}\text{Cov}\left([\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top]\boldsymbol{w}, \boldsymbol{r}^\top \boldsymbol{w}\right) \\
&= \sigma^2 \ell_a^{1/2}\ell_v^{-1}[\boldsymbol{I}_{\Delta_a} - \boldsymbol{r}\boldsymbol{r}^\top]\boldsymbol{r} \\
&= 0,
\end{aligned}$$

where the first equality follows from observing that the second term in (40) is independent of $\boldsymbol{w}$ (which depends only on $\boldsymbol{y}_{\mathcal{L}_a}$) and the second inequality uses that $\text{Cov}(\boldsymbol{w}) = \sigma^2 \boldsymbol{I}_{\Delta_a}$. This establishes that p_a is independent of p_b.

B.4 Proof of Proposition 4.2

Note that for any node u, we have $\|\boldsymbol{G}_u\|_2 = 1$ since $\boldsymbol{G}_u$ is a projection matrix. Also, by using (Guo et al. 2019, Lemma 2) (which itself follows from (Cai et al. 2019, Lemma 1)), we have

$$\|\boldsymbol{G}_u \widehat{\boldsymbol{\theta}}\|_2 \leq c_0 (\widehat{\boldsymbol{b}}^\top \widehat{\boldsymbol{\Sigma}} \widehat{\boldsymbol{b}})^{1/2} , \tag{41}$$

for some constant $c_0 > 0$. This inequality follows by analyzing the optimization (16) which is used to define the direction $\widehat{\boldsymbol{b}}$. Therefore, for any node u, we obtain

$$\begin{aligned}\frac{\|\boldsymbol{G}_u\widehat{\boldsymbol{\theta}}\|_2 + \|\boldsymbol{G}_u\|_2}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u^{\mathrm{d}})}} &\leq \frac{c_0(\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}})^{1/2}+1}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u^{\mathrm{d}})}} \\ &= \frac{c_0(\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}})^{1/2}+1}{\sqrt{\frac{4\widehat{\sigma}^2}{n}\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}}+\frac{\tau}{n}}} \\ &\leq \frac{c_0\sqrt{n}}{2\widehat{\sigma}} + \sqrt{\frac{n}{\tau}}\,. \end{aligned} \tag{42}$$

Define $x := \Phi^{-1}(1-\frac{t}{2})$ and

$$\eta_n := c_1\left(\frac{c_0}{2\widehat{\sigma}} + \sqrt{\frac{1}{\tau}}\right)\frac{s_0\log p}{\sqrt{n}}\,, \tag{43}$$

with c_1 given in (18). Under the null hypothesis $\widetilde{\mathcal{H}}_{0,u}$ (or equivalently $\mathcal{H}_{0,u}$), we have for all $t \in [0,1]$,

$$\begin{aligned}\mathbb{P}(p_u \leq t) &= \mathbb{P}\left(\Phi^{-1}(1-\frac{t}{2}) \leq \frac{|\widehat{Q}_u^{\mathrm{d}}|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right) \\ &= \mathbb{P}\left(x \leq \frac{|\widehat{Q}_u^{\mathrm{d}}|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right) \\ &= \mathbb{P}\left(x \leq \frac{|Z_u+\Delta_u|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right) \\ &\leq \mathbb{P}\left(x-\eta_n \leq \frac{|Z_u|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right) + \mathbb{P}\left(\eta_n \leq \frac{|\Delta_u|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right). \end{aligned} \tag{44}$$

By using the bias bound (18), together with (42) and definition of η_n given by (43), we have

$$\mathbb{P}\left(\eta_n \leq \frac{|\Delta_u|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right) \leq 2pe^{-c_2 n}\,, \tag{45}$$

for all nodes u. In addition,

$$\begin{aligned}
&\mathbb{P}\left(x-\eta_n \le \frac{|Z_u|}{\sqrt{\widehat{\mathrm{Var}}_\tau(\widehat{Q}_u)}}\right)\\
&\le \mathbb{P}\left(x-\eta_n \le \frac{|Z_u|}{\sqrt{\frac{4\widehat{\sigma}^2}{n}\widehat{\boldsymbol{b}}^\top\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{b}}}}\right) = \mathbb{P}\left(x-\eta_n \le \frac{\sigma|Z_u|}{\widehat{\sigma}\sqrt{\mathrm{Var}(\widehat{Q}^{\mathrm{d}}_u)}}\right)\\
&\le \mathbb{P}\left((x-\eta_n)(1-\varepsilon) \le \frac{|Z_u|}{\sqrt{\mathrm{Var}(\widehat{Q}^{\mathrm{d}}_u)}}\right) + \mathbb{P}\left(\left|\frac{\widehat{\sigma}}{\sigma}-1\right| \ge \varepsilon\right)\\
&= 2\Phi(\varepsilon x - x + \eta_n - \varepsilon\eta_n) + \mathbb{P}\left(\left|\frac{\widehat{\sigma}}{\sigma}-1\right| \ge \varepsilon\right) \qquad (46)
\end{aligned}$$

Combining (44), (45) and (46) we obtain

$$\mathbb{P}(p_u \le t) \le 2\Phi(\varepsilon x - x + \eta_n - \varepsilon\eta_n) + \mathbb{P}\left(\left|\frac{\widehat{\sigma}}{\sigma}-1\right| \ge \varepsilon\right) + 2pe^{-c_2 n}\,.$$

Note that the right-hand side of the above equation does not depend on the node u. In other words, it is a uniform bound for all nodes. Under the condition $s_0(\log p)/\sqrt{n} \to 0$, we have $\eta_n \to 0$ as $n \to \infty$. Therefore, for any fixed $\varepsilon_0 > 0$, by choosing $\varepsilon > 0$ small enough and $n_0 = n_0(\varepsilon)$ large enough we can ensure that for all $n \ge n_0$

$$\mathbb{P}(p_u \le t) \le 2\Phi(-x) + \varepsilon_0 = 2(1-\Phi(x)) + \varepsilon_0 = t + \varepsilon_0\,,$$

for all nodes u.

C Proof of Proposition A.2

For depth d we define the quantities $L_d := R^{1:(d-1)} + r_d^*$ and $U_d := p - 1 - \big(\sum_{u \in \mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d| - r_d^*\big)$. For node $a \in \mathcal{B}^*$ with $\mathrm{depth}(a) = d$, we write

$$
\begin{aligned}
&\sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}} \cdot \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \\
&\overset{(a)}{\leq} (\Delta - \frac{1}{\Delta}) \sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \frac{|\mathcal{T}'|}{\tilde{R}_{\mathcal{T}'}} \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \\
&= (\Delta - \frac{1}{\Delta}) \sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \sum_{u \in \mathcal{T}'} \frac{1}{\tilde{R}_{\mathcal{T}'}} \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \\
&= (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \sum_{\mathcal{T}' \in S(\mathcal{T}_a): u \in \mathcal{T}'} \frac{1}{\tilde{R}_{\mathcal{T}'}} \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \\
&\overset{(b)}{\leq} (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \sum_{m=L_d}^{U_d} \sum_{\mathcal{T}' \in S(\mathcal{T}_a): u \in \mathcal{T}'} \frac{1}{\tilde{R}_{\mathcal{T}'}} \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \mathbb{1}(\tilde{R}_{\mathcal{T}'} = m) \\
&= (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \sum_{m=L_d}^{U_d} \frac{1}{m} \sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}^c_{\mathcal{T}_a}) \mathbb{1}(\tilde{R}_{\mathcal{T}'} = m, u \in \mathcal{T}') \\
&= (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \sum_{m=L_d}^{U_d} \frac{1}{m} \mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}_{a,\mathrm{rej}} | \mathcal{P}^c_{\mathcal{T}_a}) \,. \qquad (47)
\end{aligned}
$$

Here (a) follows from Lemma D.2, and (b) holds since for $\mathcal{T}' \in S(\mathcal{T}_a)$, the number of total rejections $\tilde{R}_{\mathcal{T}'}$ satisfies

$$L_d \leq \tilde{R}_{\mathcal{T}'} \leq U_d \,.$$

The lower bound holds trivially since $\mathcal{T}' \in S(\mathcal{T}_a)$ and $\mathrm{depth}(a) = d$. The number of splits made by algorithm up to level d is $R^{1:d} = R^{1:(d-1)} + r_d^*$ by using Equation (23). For the upper bound, note that one can split the p leaves at most $p - 1$ times. Now focusing on nodes in depth d, rejecting a node u results in $\deg_{\mathcal{T}}(u) - 1$ additional splits. So the nodes in depth d can make up to $\sum_{u \in \mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d|$ additional splits, while the algorithm makes r_d^* additional splits as we discussed in Equation (23). So the difference between these two quantities, $\sum_{u \in \mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d| - r_d^*$, is the number of potential splits that the testing rule has missed while testing nodes at depth d. This argument implies that the total number of splits can go up to $U_d = p - 1 - (\sum_{u \in \mathcal{T}^d} \deg_{\mathcal{T}}(u) - |\mathcal{T}^d| - r_d^*)$.

Now by using bound (27), on the event $\{\tilde{R}_{\mathcal{T}'} = m\}$ we have $\alpha_u(r_d^*) \leq \tilde{\alpha}_{a,m}$. Define $\mathcal{T}^m_{a,\mathrm{rej}}$ as the rejection subtree as if the test levels $\alpha_u(r_d^*)$ are replaced by $\tilde{\alpha}_{a,m}$. Therefore $\mathcal{T}_{a,\mathrm{rej}} \subseteq \mathcal{T}^m_{a,\mathrm{rej}}$, which implies

$$\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}_{a,\mathrm{rej}} | \mathcal{P}^c_{\mathcal{T}_a}) \leq \mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}^m_{a,\mathrm{rej}} | \mathcal{P}^c_{\mathcal{T}_a}).$$

Combining this inequality with (47) and taking the expectation gives

$$\mathbb{E}\left[\sum_{\mathcal{T}'\in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}} \cdot \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}'|\mathcal{P}^c_{\mathcal{T}_a})\right] \leq (\Delta - \frac{1}{\Delta}) \sum_{u\in\mathcal{T}_a} \sum_{m=L_d}^{U_d} \frac{1}{m}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}^m_{a,\mathrm{rej}}). \tag{48}$$

Focusing on the innermost summation, we have

$$\begin{aligned}
&\sum_{m=L_d}^{U_d} \frac{1}{m}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}^m_{a,\mathrm{rej}}) \\
&= \sum_{m=L_d}^{U_d} \frac{1}{m}\left[\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^m_{a,\mathrm{rej}}) - \mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m+1, u \in \mathcal{T}^m_{a,\mathrm{rej}})\right] \\
&= \sum_{m=L_d}^{U_d} \frac{1}{m}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^m_{a,\mathrm{rej}}) - \sum_{m'=L_d+1}^{U_d+1} \frac{1}{m'-1}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m', u \in \mathcal{T}^{m'-1}_{a,\mathrm{rej}}) \\
&= \sum_{m=L_d+1}^{U_d} \left[\frac{1}{m}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^m_{a,\mathrm{rej}}) - \frac{1}{m-1}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^{m-1}_{a,\mathrm{rej}})\right] \\
&\quad + \frac{1}{L_d}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq L_d, u \in \mathcal{T}^{L_d}_{a,\mathrm{rej}}) - \frac{1}{U_d}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq U_d+1, u \in \mathcal{T}^{U_d}_{a,\mathrm{rej}}) \\
&\leq \sum_{m=L_d+1}^{U_d} \left[\frac{1}{m}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^m_{a,\mathrm{rej}}) - \frac{1}{m-1}\mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^{m-1}_{a,\mathrm{rej}})\right] \\
&\quad + \frac{1}{L_d}\mathbb{P}(u \in \mathcal{T}^{L_d}_{a,\mathrm{rej}}) \\
&\leq \sum_{m=L_d+1}^{U_d} \frac{1}{m}\mathbb{P}\left(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} \geq m, u \in \mathcal{T}^m_{a,\mathrm{rej}} \backslash \mathcal{T}^{m-1}_{a,\mathrm{rej}}\right) + \frac{1}{L_d}\mathbb{P}(u \in \mathcal{T}^{L_d}_{a,\mathrm{rej}}) \\
&\leq \sum_{m=L_d+1}^{U_d} \frac{1}{m}\mathbb{P}\left(u \in \mathcal{T}^m_{a,\mathrm{rej}} \backslash \mathcal{T}^{m-1}_{a,\mathrm{rej}}\right) + \frac{1}{L_d}\mathbb{P}(u \in \mathcal{T}^{L_d}_{a,\mathrm{rej}}), \qquad (49)
\end{aligned}$$

where in the last equality we used the observation $\mathcal{T}^{m-1}_{a,\mathrm{rej}} \subseteq \mathcal{T}^m_{a,\mathrm{rej}}$, since $\alpha_{u,m}$ is increasing in m.

For exposition purposes, we define the shorthand $q_{u,m} = \mathbb{P}(u \in \mathcal{T}^m_{a,\mathrm{rej}})$ for $u \in \mathcal{T}_a$ and

$m \geq 1$. Then, from the chain of inequalities in (49) we get

$$\begin{aligned}
&\sum_{m=L_d}^{U_d} \frac{1}{m} \mathbb{P}(\tilde{R}_{\mathcal{T}_{a,\mathrm{rej}}} = m, u \in \mathcal{T}_{a,\mathrm{rej}}^m) \\
&\leq \sum_{m=L_d+1}^{U_d} \frac{1}{m} \mathbb{P}\left(u \in \mathcal{T}_{a,\mathrm{rej}}^m \backslash \mathcal{T}_{a,\mathrm{rej}}^{m-1}\right) + \frac{1}{L_d} \mathbb{P}(u \in \mathcal{T}_{a,\mathrm{rej}}^{L_d}) \\
&\leq \sum_{m=L_d+1}^{U_d} \frac{1}{m} (q_{u,m} - q_{u,m-1}) + \frac{1}{L_d} q_{u,L_d} \\
&= \sum_{m=L_d}^{U_d} \frac{1}{m} q_{u,m} - \sum_{m=L_d}^{U_d-1} \frac{1}{m+1} q_{u,m} \\
&= \frac{1}{U_d} q_{u,U_d} + \sum_{m=L_d}^{U_d-1} \left(\frac{1}{m} - \frac{1}{m+1}\right) q_{u,m}. \qquad (50)
\end{aligned}$$

By deploying (50) in the bound (48), we get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{\mathcal{T}' \in S(\mathcal{T}_a)} \frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}} \cdot \mathbb{P}(\mathcal{T}_{a,\mathrm{rej}} = \mathcal{T}' | \mathcal{P}_{\mathcal{T}_a}^c)\right] \\
&\leq (\Delta - \frac{1}{\Delta}) \sum_{u \in \mathcal{T}_a} \left(\frac{1}{U_d} q_{u,U_d} + \sum_{m=L_d}^{U_d-1} \left(\frac{1}{m} - \frac{1}{m+1}\right) q_{u,m}\right) \\
&= (\Delta - \frac{1}{\Delta}) \left(\frac{1}{U_d} \sum_{u \in \mathcal{T}_a} q_{u,U_d} + \sum_{m=L_d}^{U_d-1} \frac{1}{m(m+1)} \sum_{u \in \mathcal{T}_a} q_{u,m}\right). \qquad (51)
\end{aligned}$$

Our next step is to upper bound $\sum_{u \in \mathcal{T}_a} q_{u,m}$ which is the subject of the following lemma.

Lemma C.1. *For any integer $m \geq 1$ we have*

$$\sum_{u \in \mathcal{T}_a} q_{u,m} \leq \frac{\tilde{\gamma}_{a,m}}{p(\Delta - \frac{1}{\Delta})},$$

where $\tilde{\gamma}_{a,m}$ is given by (26).

The proof of Lemma C.1 is deferred to Section C.1.

By virtue of Lemma C.1 and (51), we have

$$\begin{aligned}\mathbb{E}\left[\sum_{\mathcal{T}'\in S(\mathcal{T}_a)}\frac{V_a(\mathcal{T}')}{\tilde{R}_{\mathcal{T}'}}\cdot\mathbb{P}(\mathcal{T}_{a,\mathrm{rej}}=\mathcal{T}'|\mathcal{P}^c_{\mathcal{T}_a})\right] &\le \frac{1}{p}\left(\frac{1}{U_d}\tilde{\gamma}_{a,U_d}+\sum_{m=L_d}^{U_d-1}\frac{1}{m(m+1)}\tilde{\gamma}_{a,m}\right)\\ &= \frac{\alpha|\mathcal{L}_a|}{p\hbar_{d,r}}\left(1+\sum_{m=L_d}^{U_d-1}\frac{1}{m+1}\right)\\ &= \frac{\alpha|\mathcal{L}_a|}{p\hbar_{d,r}}\left(1+\sum_{m=L_d+1}^{U_d}\frac{1}{m}\right)\\ &= \frac{\alpha|\mathcal{L}_a|}{p}\,. \end{aligned} \tag{52}$$

C.1 Proof of Lemma C.1

Since $a\in\mathcal{B}^*$, any node $u\in\mathcal{T}_a$ is a true null and hence it has a super uniform p-value, i.e. for any $x\in[0,1]$ we have $\mathbb{P}(p_u\le x)\le x$. In addition, by our assumption the null p-values are independent and if a node u is rejected so are the nodes on the path from node a to it. Therefore,

$$q_{u,m}=\mathbb{P}(u\in\mathcal{T}^m_{a,\mathrm{rej}})\le\tilde{\alpha}_{a,m}^{\mathrm{depth}(u)-\mathrm{depth}(a)+1}\,. \tag{53}$$

Here we used the fact that the rejection thresholds in $\mathcal{T}^m_{a,\mathrm{rej}}$ are set to $\tilde{\alpha}_{a,m}$.

Also, since the node degrees in $\mathcal{T}$ are at most Δ, the number of nodes in subtree $\mathcal{T}_a$ that are depth d of the tree $\mathcal{T}$ is at most $\Delta^{d-\mathrm{depth}(a)}$. We therefore have

$$\begin{aligned}\sum_{u\in\mathcal{T}_a}q_{u,m} &\le \sum_{d=\mathrm{depth}(a)}^{D}\Delta^{d-\mathrm{depth}(a)}\tilde{\alpha}_{a,m}^{d-\mathrm{depth}(a)+1}\\ &\le \sum_{d=1}^{\infty}\Delta^{d-1}\tilde{\alpha}_{a,m}^{d}\\ &= \frac{1}{\Delta}\frac{\Delta\tilde{\alpha}_{a,m}}{1-\Delta\tilde{\alpha}_{a,m}}\\ &= \frac{\tilde{\gamma}_{a,m}}{p(\Delta-\frac{1}{\Delta})}\,, \end{aligned} \tag{54}$$

which completes the proof.

D Some useful lemmas

Lemma D.1. *Consider a tree $\mathcal{T}$ with maximum degree Δ. Denote by $\mathcal{L}$ the set of leaf nodes in $\mathcal{T}$. We then have*

$$|\mathcal{L}|\le\frac{(\Delta-1)|\mathcal{T}|+1}{\Delta},$$

where $|\mathcal{T}|$ denotes the number of nodes in $\mathcal{T}$.

Proof. Recall that the degree of a node u is the number of its children in the tree. The leaves are of zero degree and the other nodes are of maximum degree Δ. Therefore,

$$(|\mathcal{T}| - p)\Delta \geq \sum_{u \in \mathcal{T}} \deg_{\mathcal{T}}(u) = |\mathcal{T}| - 1\,.$$

By rearranging the terms we get

$$p \leq \frac{(\Delta - 1) \cdot |\mathcal{T}| + 1}{\Delta}.$$

□

Lemma D.2. *Consider a tree $\mathcal{T}$ with maximum degree Δ. For $\mathcal{T}'$, a subtree of $\mathcal{T}$, define*

$$V(\mathcal{T}') = \sum_{u \in \mathcal{T}'} (\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}'}(u)) - 1\,.$$

We then have the following bound on $V(\mathcal{T}')$:

$$V(\mathcal{T}') \leq \frac{(\Delta^2 - 1) \cdot |\mathcal{T}'| + 1}{\Delta} - 1 \leq \left(\Delta - \frac{1}{\Delta}\right) |\mathcal{T}'|\,,$$

where $|\mathcal{T}'|$ denotes the number of nodes in $\mathcal{T}'$.

Proof. If node $u \in \mathcal{T}'$ is not a leaf of $\mathcal{T}'$, we have $\deg_{\mathcal{T}'}(u) \geq 1$ and so

$$\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}'}(u) \leq \deg_{\mathcal{T}}(u) - 1 \leq \Delta - 1\,.$$

If $u \in \mathcal{T}'$ is a leaf of $\mathcal{T}'$, we have

$$\deg_{\mathcal{T}}(u) - \deg_{\mathcal{T}'}(u) = \deg_{\mathcal{T}}(u) \leq \Delta\,.$$

We therefore have

$$\begin{aligned}
V(\mathcal{T}') &= \sum_{u \in \mathcal{T}'} (\deg_{\mathcal{T}'}(u) - \deg_{\mathcal{T}'}(u)) - 1 \\
&\leq |\mathcal{L}_{\mathcal{T}'}| \cdot \Delta + (|\mathcal{T}'| - |\mathcal{L}_{\mathcal{T}'}|)(\Delta - 1) - 1 \\
&= |\mathcal{T}'| \cdot (\Delta - 1) + |\mathcal{L}_{\mathcal{T}'}| - 1 \\
&\leq \frac{(\Delta^2 - 1) \cdot |\mathcal{T}'| + 1}{\Delta} - 1 \\
&\leq \left(\Delta - \frac{1}{\Delta}\right) |\mathcal{T}'|, \qquad (1)
\end{aligned}$$

where the second inequality follows from Lemma D.1. □

E Data generating process for regression simulation

We first form a balanced 3-regular tree with $p = 243$ leaves. We express the tree by a binary matrix $\boldsymbol{A} \in \{0,1\}^{p\times|\mathcal{T}|}$ with rows corresponding to features and columns corresponding to nodes. Each entry $A_{ju} = 1$ if node u is an ancestor of leaf j or if $u = j$, and $A_{ju} = 0$ otherwise. For a given K, we cut the tree into K subtrees. The roots of the subtrees form $\mathcal{B}^*$. We want to set the coefficients corresponding to the leaves within each subtree to the same value. To achieve this, we generate a vector of length K, denoted as $\widetilde{\boldsymbol{\theta}}^*$, with the first $(1-\beta)K$ elements set to 0; the other βK elements of of $\widetilde{\boldsymbol{\theta}}^*$ are independently drawn from $\mathsf{N}(0, 0.5^2)$. Then we set $\boldsymbol{\theta}^* = \boldsymbol{A}_{\mathcal{B}^*}\widetilde{\boldsymbol{\theta}}^*$, where $\boldsymbol{A}_{\mathcal{B}^*}$ is matrix $\boldsymbol{A}$ restricted to columns that correspond to the nodes in $\mathcal{B}^*$. Note that the columns of $\boldsymbol{A}_{\mathcal{B}^*}$ have disjoint supports as no two nodes in $\mathcal{B}^*$ can share a same descendant. Parameter β controls the sparsity of $\widetilde{\boldsymbol{\theta}}^*$, and therefore sparsity of $\boldsymbol{\theta}^*$.

To simulate a setting with rare feature, we consider a design matrix $\boldsymbol{X} := \widetilde{\boldsymbol{X}} \odot \boldsymbol{W} \in \mathbb{R}^{n\times p}$ from a Bernoulli-Gaussian distribution. The entries $\widetilde{X}_{ij}$ are generated i.i.d from standard normal distribution. The entries W_{ij} are drawn i.i.d from Bernoulli(ρ). The Bernoulli parameter ρ determines the level of rareness in the design matrix. Also $\odot$ represents the entry-wise product of two matrices. Finally, the high-dimensional linear model is generated by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta}^* + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathsf{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n), \tag{55}$$

where $\sigma = c\frac{\|\boldsymbol{X}\boldsymbol{\theta}^*\|_2}{\sqrt{n}}$. We fix the parameters as $n = 100, p = 243, \beta = 0.6, \rho = 0.2, \sigma = 0.6$, and vary K from 21 to 93.

F More information about the stocks example

In this appendix, we provide further information about the NAICS stocks example.

F.1 Determination of a stock's average daily volatility

We use daily stock price data from January 1, 2015 to December 31, 2019,[3] (CRSP Stocks 2015-2019). Specifically, we wish to aggregate stocks in a similar sector unless their volatility levels are significantly different. We use several criteria for screening stocks of interest: We only keep common stocks that are publicly traded throughout this entire period; we also avoid penny stocks that have prices under \$0.01 per share. After pre-screening, we have $n = 2538$ stocks in total. Following Parkinson (1980) and Martens & van Dijk (2007), we use the high-low range estimator for the daily variance $v_t = \frac{1}{4\log(2)}(\log(H_t) - \log(L_t))^2$, where H_t and L_t are day t's highest and lowest prices, respectively. Finally, for each stock we take the average of v_t throughout the 5-year period and log-transform to reduce skewness.

[3]derived from the US Stock Database ©2021 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business

F.2 More displays of HAT aggregation results

Table 1: *The table shows* 40 *clusters that are aggregated by our procedure. The rightmost column is the mean log-volatility. The column Company/Number of Companies shows how many companies are in each cluster (or the name of the company when there is only one). The columns from Sector to Industry National show the classification of the clusters that are selected.*

Sector	Subsector	Industry Group	Industry	Industry National	Company/Number of Companies	Mean
Agriculture, Forestry, Fishing and Hunting					5	-7.69
Mining, Quarrying, and Oil and Gas Extraction					63	-6.51
Utilities					28	-8.4
Construction					45	-7.55
Manufacturing I					73	-8.03
Manufacturing III					576	-7.48
Wholesale Trade					75	-7.62
Retail Trade I					75	-7.56
Retail Trade II					34	-7.5
Transportation					44	-7.81
Warehousing					3	-8.31
Information					261	-7.5
Real Estate and Rental and Leasing					50	-7.34
Professional, Scientific, and Technical Services					77	-7.55
Administrative and Support and Waste Management and Remediation Services					57	-7.65
Educational Services					12	-7.38
Health Care and Social Assistance					41	-7.36
Arts, Entertainment, and Recreation					16	-7.75
Accommodation and Food Services					52	-7.86
Other Services (except Public Administration)					6	-8.08
Nonclassifiable Establishments					9	-7.28
Manufacturing II	Wood Product Manufacturing				6	-7.55
	Paper Manufacturing				19	-8.14
	Printing and Related Support Activities				4	-7.91
	Petroleum and Coal Products Manufacturing				16	-7.93
	Plastics and Rubber Products Manufacturing				17	-7.65
	Nonmetallic Mineral Product Manufacturing				11	-7.44
	Chemical Manufacturing	Basic Chemical Manufacturing			28	-7.18
		Resin, Synthetic Rubber, and Artificial and Synthetic Fibers and Filaments Manufacturing			6	-7.91
		Pesticide, Fertilizer, and Other Agricultural Chemical Manufacturing			6	-7.14
		Pharmaceutical and Medicine Manufacturing			299	-6.37
		Paint, Coating, and Adhesive Manufacturing			6	-8.06
		Soap, Cleaning Compound, and Toilet Preparation Manufacturing			14	-8.18
		Other Chemical Product and Preparation Manufacturing			5	-8.35
Finance and Insurance	Securities, Commodity Contracts, and Other Financial Investments and Related Activities				64	-7.95
	Insurance Carriers and Related Activities				87	-8.3
	Funds, Trusts, and Other Financial Vehicles				MANHATTAN BRIDGE CAPITAL INC	-7.64
	Credit Intermediation and Related Activities	Depository Credit Intermediation			306	-8.27
		Nondepository Credit Intermediation			32	-7.67
		Activities Related to Credit Intermediation			9	-7.59

In the main paper, the aggregation tree from applying HATis shown. Here, we include Table 1, which shows this same aggregation together with the value of the estimated volatility for each aggregated cluster.

To get a further sense of the results, Figure 8 focuses on the 347 companies in the subsector "Credit Intermediation and Related Activities". Each point represents the log-volatility of a company. The three facets correspond to three industry groups within the subsector and eight levels on the y-axis correspond to the eight industries nested in the industry groups. As can be observed in the plot, the industry group "Depository Credit Intermediation" has significantly lower mean (around -8.27) compared to the other two industry groups in the subsector (around -7.67 and -7.59 respectively). Therefore, the null hypothesis that the three industry groups have similar mean volatility is rejected. On the contrary, within each industry group, there are no noticeable differences among different industries, leading none of the null hypotheses at the industry group level to be rejected.

F.3 The difference between FSR and FDR control with this tree

We noted in the main paper that had we instead 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.

We emphasize the practical impact that this distinction between FDR and FSR can have by considering what would happen if we falsely reject a high degree node. In particular, "Commercial Banking" is the highest degree node in the NAICS tree. It is at the "industry" level and has 254 companies (banks) as its children. Suppose the volatility within this industry were effectively constant. A false rejection of this node would wrongly increase the number of clusters by 253. As an example, suppose $\mathcal{B}^*$ consists of five randomly chosen nodes along with the commercial banking node. That is, the true aggregation would be into these six clusters (in which one of the true clusters consists of all the commercial banks). Imagine a rejection procedure that gets all the proper rejections to create these six clusters except that it accidentally rejects the "commercial banking" node. We carried out this scenario in a simulation (choosing the five other nodes uniformly at random). The FDP is about 4.5% while the FSP is about 80%. Given how different these two values are, we see that an FSR controlling aggregation policy like HAT will be extremely careful before splitting a high degree node compared to an FDR controlling aggregation policy.

G NYC taxi data

The availability of taxis is not uniformly distributed across the city (see Figure 9), and $\boldsymbol{X}$ is a highly sparse matrix: Most areas had fewer than 10% of the drivers starting their trips there during that month, and in fact 109 out of the 194 neighborhoods have seen less than 1% of the drivers.

We study how the aggregation results vary with sample size. To do so, we randomly subset the original dataset to different sizes, and perform the above-mentioned procedure

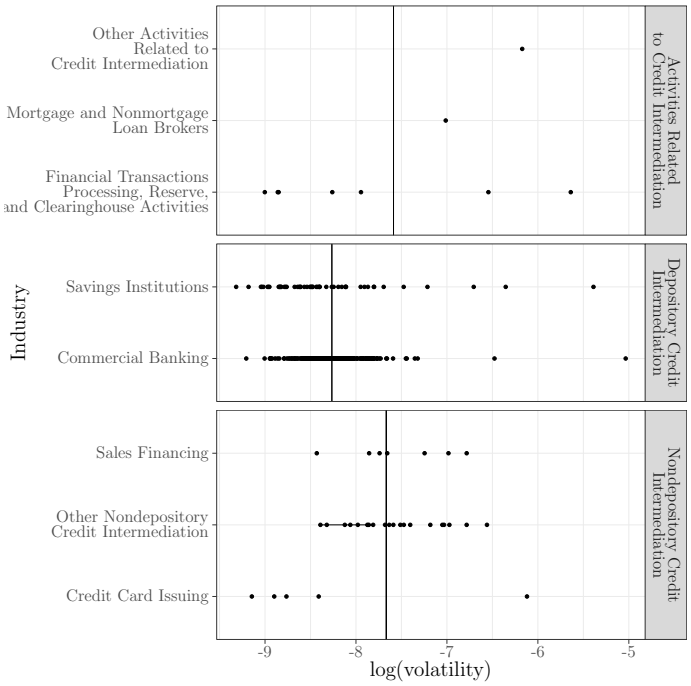


Figure 8: *The subsector "Credit Intermediation and Related Activities" consists of* 347 *companies, represented as points. These fall into 3 industry groups and 8 industries. Applying* HAT *rejects the null hypothesis that the 3 industry groups have the same mean* log*-volatility, but does not reject this within each industry group.*

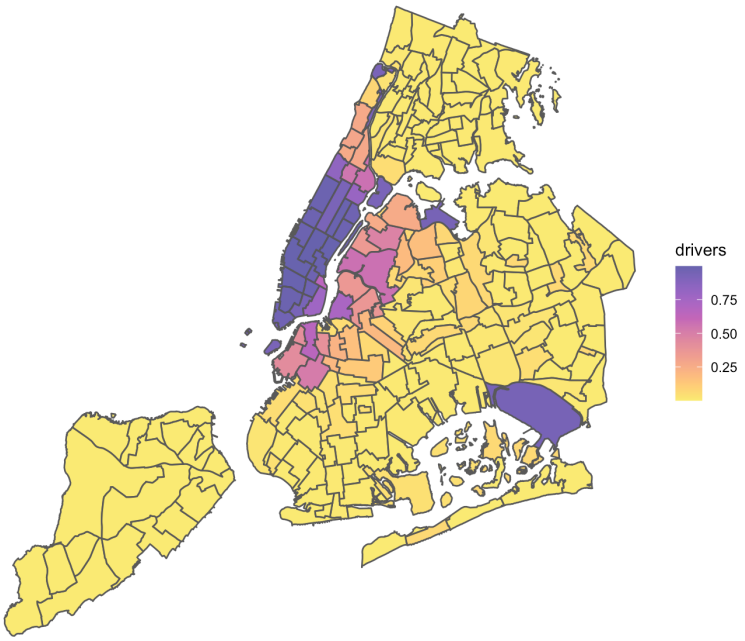


Figure 9: *Map of neighborhoods colored with percentage of drivers who have started a trip from there. Most neighborhoods have fewer than 10% of the drivers starting their trips there in the month of December 2013.*

on each sample. The number of achieved groups for each sample size is shown in Table 2. As expected, reduced sample sizes leads to fewer rejections and therefore fewer aggregated groups.

G.1 FSR with synthetic data

To directly evaluate the aggregation recovery performance of HAT, we create a synthetic response based on the tree structure $\mathcal{T}$ constructed by the neighborhoods, as well as the observed trip counts data $\boldsymbol{X}$. In addition, we take the aggregation result and fitted coefficients from Section 6.2.1 as the true aggregation and true vector $\boldsymbol{\theta}^*$. We simulate the

Table 2: *Achieved number of groups with decaying sample size.*

Sample Size	p	Number of Groups
$n = 32704$	194	45
$n/2 = 16352$	194	42
$n/4 = 8176$	194	34
$n/8 = 4088$	194	29
$n/16 = 2044$	194	21
$n/32 = 1022$	194	17

Table 3: *Achieved FSR and average power by our algorithm with synthetic data where noise level is* $\sigma = 15$.

Target Level	FSR	Average Power
0.01	0.000	0.546
0.02	0.000	0.559
0.05	0.002	0.580
0.10	0.003	0.598
0.20	0.004	0.615
0.30	0.007	0.624
0.40	0.009	0.633
0.50	0.011	0.643

response 100 times independently according to (55) with $\sigma = 15$. We use the same debiased method to calculate the node-wise p-values and perform our testing procedure with target FSR levels varying from $\alpha = 0.01$ to $\alpha = 0.3$. We compare the aggregation results with the true aggregation and compute FSR and average power over the 100 runs. The results are shown in Table 3.

References

Benjamini, Y. & Hochberg, Y. (1995), 'Controlling the false discovery rate: A practical and powerful approach to multiple testing', *Journal of the Royal Statistical Society. Series B (Methodological)* **57**(1), 289–300.

Bien, J. (2016), 'The simulator: an engine to streamline simulations', *arXiv preprint arXiv:1607.00021* .

Bien, J., Yan, X., Simpson, L. & Müller, C. L. (2021), 'Tree-aggregated predictive modeling of microbiome data', *Scientific Reports* **11**(1), 1–13.

Blanchard, G. & Roquain, E. (2008), 'Two simple sufficient conditions for fdr control', *Electronic Journal of Statistics* **2**(0), 963–992.

Bogomolov, M., Peterson, C. B., Benjamini, Y. & Sabatti, C. (2017), 'Testing hypotheses on a tree: new error rates and controlling strategies', *arXiv preprint arXiv:1705.07529* .

Cai, T., Cai, T. & Guo, Z. (2019), 'Optimal statistical inference for individualized treatment effects in high-dimensional models', *arXiv preprint arXiv:1904.12891 (To appear in Journal of the Royal Statistical Society: Series B)* .

Cai, T. T., Guo, Z. et al. (2017), 'Confidence intervals for high-dimensional linear regression: Minimax rates and adaptivity', *The Annals of statistics* **45**(2), 615–646.

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

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.

Guo, Z., Renaux, C., Bühlmann, P. & Cai, T. T. (2019), 'Group inference in high dimensions with applications to hierarchical testing', *arXiv preprint arXiv:1909.01503* .

Heller, R., Chatterjee, N., Krieger, A. & Shi, J. (2018), 'Post-selection inference following aggregate level hypothesis testing in large-scale genomic data', *Journal of the American Statistical Association* **113**(524), 1770–1783.

Javanmard, A. & Lee, J. D. (2020), 'A flexible framework for hypothesis testing in high dimensions', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **82**(3), 685–718.

Javanmard, A. & Montanari, A. (2014), 'Confidence intervals and hypothesis testing for high-dimensional regression', *The Journal of Machine Learning Research* **15**(1), 2869–2909.

Javanmard, A. & Montanari, A. (2018*a*), 'Debiasing the lasso: Optimal sample size for gaussian designs', *The Annals of Statistics* **46**(6A), 2593–2622.

Javanmard, A. & Montanari, A. (2018*b*), 'Online rules for control of false discovery rate and false discovery exceedance', *The Annals of statistics* **46**(2), 526–554.

Katsevich, E. & Sabatti, C. (2019), 'Multilayer knockoff filter: Controlled variable selection at multiple resolutions', *The annals of applied statistics* **13**(1), 1.

Lynch, G. & Guo, W. (2016), 'On procedures controlling the fdr for testing hierarchically ordered hypotheses', *arXiv preprint arXiv:1612.04467* .

Martens, M. & van Dijk, D. (2007), 'Measuring volatility with the realized range', *Journal of Econometrics* **138**(1), 181–207. 50th Anniversary Econometric Institute.

Meinshausen, N. (2008), 'Hierarchical testing of variable importance', *Biometrika* **95**(2), 265–278.

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

Parkinson, M. (1980), 'The extreme value method for estimating the variance of the rate of return', *The Journal of Business* **53**(1), 61–65.

Ramdas, A., Chen, J., Wainwright, M. J. & Jordan, M. I. (2019), 'A sequential algorithm for false discovery rate control on directed acyclic graphs', *Biometrika* **106**(1), 69–86.

Seber, G. A. F. & Lee, A. J. (2012), *Linear regression analysis*, Wiley.

Simes, R. J. (1986), 'An improved bonferroni procedure for multiple tests of significance', *Biometrika* **73**(3), 751–754.

Sun, T. & Zhang, C.-H. (2012), 'Scaled sparse linear regression', *Biometrika* **99**(4), 879–898.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288.

US OMB (2017), 'North american industry classification system', *Executive Office of the President; Office of Management and Budget* .
URL: *https://www.census.gov/naics/reference_files_tools/2017_NAICS_Manual.pdf*

US OMB (2018), 'Standard occupational classification manual', *Executive Office of the President; Office of Management and Budget* .
URL: *https://www.bls.gov/soc/2018/soc_2018_manual.pdf*

van de Geer, S., Bühlmann, P., Ritov, Y. & Dezeure, R. (2014), 'On asymptotically optimal confidence regions and tests for high-dimensional models', *The Annals of Statistics* **42**(3), 1166–1202.

Wilms, I. & Bien, J. (2021), 'Tree-based node aggregation in sparse graphical models', *arXiv preprint arXiv:2101.12503* .

Yan, X. & Bien, J. (2020), 'Rare feature selection in high dimensions', *Journal of the American Statistical Association* **0**(0), 1–14.

Yan, X. & Bien, J. (n.d.), *rare: Linear Model with Tree-Based Lasso Regularization for Rare Features*. R package version 0.1.0.
URL: *https://github.com/yanxht/rare*

Yekutieli, D. (2008), 'Hierarchical false discovery rate-controlling methodology', *Journal of the American Statistical Association* **103**(481), 309–316.

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 (Statistical Methodology)* **76**(1), 217–242.