Hierarchical Structure from Minimal Observations

Constraint-Driven Entropy Reduction in Algorithms, Inference, and Complex Systems

Flyxion

First Edition

Preface

> Every observation is a constraint. Every constraint is a reduction in uncertainty. What we call knowledge is the residue of that reduction.
>
> —*Flyxion*

This textbook develops a unified mathematical account of a single phenomenon: hierarchical structure can be reconstructed from minimal relational information. The central argument is not merely algorithmic. It is a claim about the nature of structured knowledge itself. When a system of objects possesses latent hierarchical organization, that organization leaves observable traces in the pairwise and triadic relationships among those objects. The task of inference is to reconstruct the global structure from those local witnesses.

The argument unfolds in nine parts. The first provides the prerequisites: discrete mathematics, probability and concentration inequalities, information theory, and the geometry of metric and ultrametric spaces. These tools are not reviewed for their own sake; each one plays a direct role in the main development. The reader who has a background in combinatorics and probability can move quickly through this material, returning to it as a reference.

The second and third parts introduce hierarchical clustering as a formal object. A hierarchy is a laminar family of subsets, and its geometry is captured by an ultrametric. The minimal unit of evidence for branching structure is a rooted triplet relation, and a sufficiently rich collection of such relations uniquely determines the underlying tree. Classical clustering objectives formalize what it means for a hierarchy to represent a dataset well, and classical complexity theory

establishes that optimizing these objectives is computationally hard without additional information.

The fourth and fifth parts introduce the learning-augmented perspective. A splitting oracle provides noisy answers to triplet queries, and the paper of reference in this area shows that such oracle access, even with substantial noise, is sufficient to break the classical hardness barriers. The central construction is the partial hierarchical clustering tree, a data structure that captures the structure of the optimal tree at large scales while deferring resolution at small scales where evidence is thin. Precise approximation guarantees are established for both the Dasgupta and Moseley–Wang objectives.

The sixth part examines how these algorithms extend to streaming and parallel computation models. The seventh part reinterprets the entire framework geometrically and information-theoretically. A triplet observation is equivalent to a local curvature measurement in ultrametric space, and the accumulation of triplet constraints drives an entropy descent on the space of tree topologies. This interpretation connects the algorithmic results to a broader principle: global structure emerges from sparse local measurements when those measurements form a coherent constraint network.

The eighth part extends the framework into the sociology of knowledge. The replication crisis in empirical science is analyzed as a structural consequence of selection mechanisms that suppress disconfirming observations, thereby distorting the constraint network from which scientific knowledge is inferred. The mathematical formalism of constraint operators, Bayesian updating, and entropy reduction provides a precise language for this phenomenon and suggests design principles for more robust inference systems. The value of publishing exploratory, incomplete, and even incorrect reasoning is defended within this framework as a mechanism for preserving the search trajectory that underlies reliable knowledge.

The final part collects the full proofs: the equivalence of trees and ultrametrics, the reconstruction theorem for triplet systems, the noise analysis for oracle aggregation, the loss bounds for collapsed subtrees, and the variational reformulation of Bayesian tree inference.

The mathematical level throughout is graduate. The reader is expected to be comfortable with discrete probability, basic combinatorics, and standard asymptotic notation. Familiarity with informa-

tion theory and metric geometry is helpful but not required, as these topics are developed from first principles in Part I.

A note on notation: throughout the text, T^* denotes the unknown optimal tree, $\widehat{T}$ the tree returned by an algorithm, $\mathcal{H}$ a hypothesis space, $\mathcal{R}$ a set of triplet relations, and $\mathcal{O}$ a splitting oracle. Calligraphic letters generally denote sets or spaces; roman capitals denote random variables or trees; lowercase letters denote vertices or scalars.

Flyxion
First Edition

Contents

List of Figures

Part I

Prerequisites and Foundations

Chapter 1

The Problem of Latent Hierarchy

> Nature does not organize itself in lists. It organizes itself in trees.
>
> —*Flyxion*

This opening chapter situates the central problem of the textbook within the broadest possible scientific context. We begin not with definitions but with a question that any careful observer of natural systems must eventually confront: given only pairwise or local measurements of similarity, how does one recover the branching structure that produced those measurements? The chapter surveys the landscape of systems in which this problem arises—from phylogenetics to linguistics to social organization—and then introduces the conceptual pipeline that the entire book will formalize. Observations become local constraints, constraints accumulate into a network, and the network drives an entropy descent toward the unique hierarchical structure consistent with all the evidence. No formal definitions appear here; those begin in Chapter 2. The purpose of this chapter is to build the reader's intuition for why the problem is deep, why it appears in so many domains, and why the learning-augmented framework developed in Parts III through V represents a genuine advance over classical approaches.

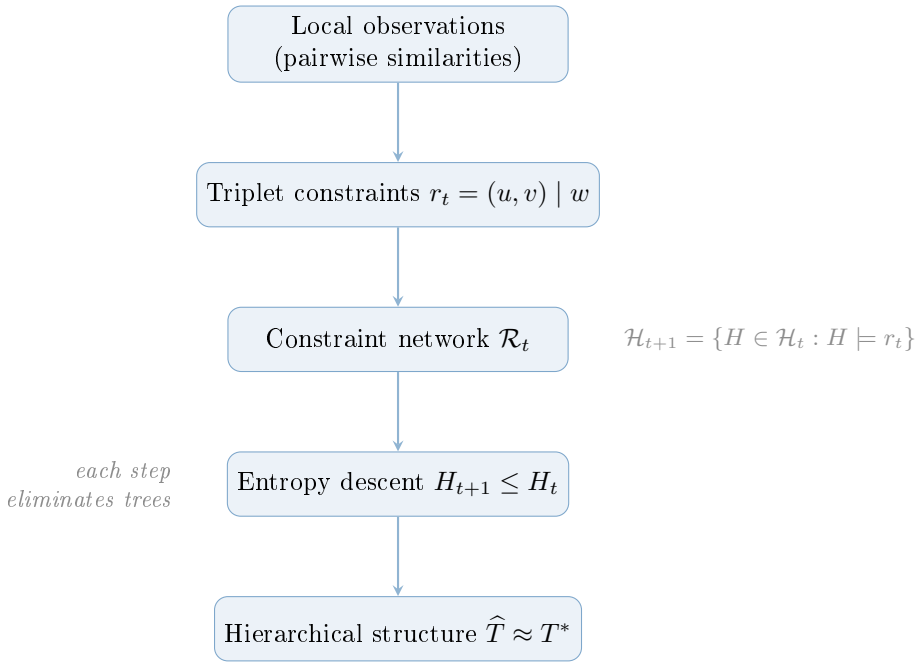


Figure 1.1: The conceptual pipeline of the textbook. Each local observation contributes a constraint that eliminates inconsistent tree topologies, driving entropy monotonically downward until only the true hierarchy remains.

1.1 The Central Question

Many of the most important structures in science, language, and cognition are hierarchical. Biological species are organized by evolutionary branching. Lexical concepts form trees of semantic generality. Grammatical parse structures are rooted labeled trees over symbol sequences. Social organizations decompose into divisions, departments, and teams. Even the physical universe, at sufficiently large scales, exhibits a hierarchical clustering of matter into galaxies, clusters, and superclusters.

What these phenomena share is that the hierarchy is latent. An observer typically cannot directly witness the branching events that produced the structure. Instead, they observe pairwise or local similarities among the objects at the leaves of the tree: genomic distances

between species, semantic proximity between words, social connection strengths among individuals. The challenge is to infer the global branching structure from these local relational measurements.

This textbook is an extended examination of that challenge. The central thesis is the following.

> Hierarchical structure can be reconstructed from minimal relational information when those observations are consistent with a coherent constraint network. The reconstruction process is an instance of constraint-driven entropy reduction: each observation eliminates incompatible tree topologies, driving the residual uncertainty toward the true hierarchy.

The primary technical vehicle is the study of hierarchical clustering algorithms, and in particular the learning-augmented framework in which a splitting oracle provides noisy triplet answers [5, 8]. The conceptual framework reaches further, connecting algorithmic theory to information geometry, Bayesian inference, and the epistemology of knowledge accumulation in scientific communities [7, 9].

1.2 Examples of Latent Hierarchical Organization

Before introducing any formalism it is useful to survey the range of systems in which the problem arises.

1.2.1 Phylogenetics

In evolutionary biology, the taxa of interest are related by a branching history of speciation events. The hierarchy is the phylogenetic tree [11, 12]. It is not directly observable. What is observable is genomic sequence data, from which pairwise similarity scores or distance estimates can be computed.

1.2.2 Linguistic Hierarchies

Natural language exhibits hierarchical structure at multiple levels. Syntactic parse trees decompose sentences into phrases. Semantic

hierarchies organize concepts from general to specific. In each case the hierarchy must be inferred from distributional evidence.

1.2.3 Cognitive Categories

Cognitive scientists have observed that human conceptual organization is approximately hierarchical. The concept "dog" is a subcategory of "mammal," which is a subcategory of "animal." This organization supports inference and knowledge transfer.

1.2.4 Social and Organizational Networks

In social networks, hierarchical community structure must be inferred from the edge structure. The resulting hierarchy is a model of the community decomposition at multiple resolutions.

1.3 The Role of Relational Observations

A key insight, developed in Chapter 9, is that pairwise distances are not the most efficient representation of hierarchical structure. Three objects in a hierarchy can be related in exactly three distinct ways, determined by which pair shares the lowest common ancestor. This triadic information is the minimal unit of evidence for branching order.

A single triplet observation eliminates roughly two-thirds of the possible tree topologies on its three leaves. When many triplet observations are accumulated, the constraint network they define becomes sufficient to determine the entire tree uniquely.

1.4 The Learning-Augmented Perspective

Classical hierarchical clustering algorithms operate without external information. This leads to computational hardness results: optimizing natural objectives over all possible tree topologies is NP-hard under standard assumptions [2, 3].

The learning-augmented framework allows access to a splitting oracle that answers noisy triplet queries. The remarkable result, which this textbook develops in full, is that such noisy oracle access is sufficient to break the classical computational barriers.

1.5 Overview of the Text

Part I develops the mathematical prerequisites. Part II introduces the core combinatorial and geometric objects. Part III establishes the classical hardness setting. Parts IV and V introduce and analyze the learning-augmented algorithms. Parts VI and VII extend to streaming, parallel, and information-theoretic viewpoints. Part VIII connects the framework to inference systems and the epistemology of science. The appendices provide complete proofs.

Chapter 2

Discrete Structures, Trees, and Combinatorics

> Count the shapes before you measure the distances.
>
> —*Flyxion*

This chapter builds the combinatorial foundation for everything that follows. The central objects are rooted trees: finite graphs in which a single distinguished vertex, the root, imposes a directed structure on every edge. We define the key vocabulary—leaves, internal nodes, lowest common ancestors, laminar families—and then turn to a counting problem whose answer is both surprising and foundational. The number of distinct rooted binary tree topologies on n labeled leaves is $(2n-3)!!$, a superexponential quantity that grows faster than any polynomial and faster than simple exponentials. This count defines the size of the hypothesis space over which inference must operate: before any observations are made, the algorithm faces uncertainty over an astronomically large set of candidate trees. Reducing that uncertainty, one constraint at a time, is the core task of the entire book. The chapter closes with the entropy of the uniform prior over tree topologies, which sets the information-theoretic scale for the reconstruction problem.

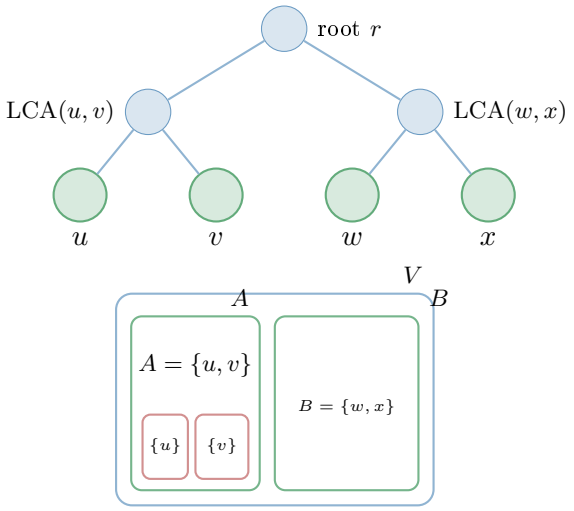


Figure 2.1: *Left*: A rooted binary tree on four leaves. Internal nodes are labeled with their lowest common ancestors. *Right*: The corresponding laminar family of leaf sets. Each internal node of the tree corresponds to a set in the laminar family, and containment of sets mirrors the ancestor relation.

2.1 Graphs and Trees

Definition 2.1 (Graph)**.** A *graph* $G = (V, E)$ consists of a finite set of *vertices* V and a set of *edges* $E \subseteq \binom{V}{2}$. A graph is *connected* if for every pair $u, v \in V$ there exists a path from u to v.

Definition 2.2 (Tree)**.** A *tree* is a connected acyclic graph. A *rooted tree* is a tree $T = (V, E, r)$ with a distinguished vertex $r \in V$ called the *root*. The root induces a natural parent–child relation: u is the *parent* of v if u lies on the unique path from r to v and is adjacent to v.

Definition 2.3 (Leaves and internal nodes)**.** In a rooted tree, a vertex with no children is a *leaf*. The set of all leaves is $L(T)$. A vertex with at least one child is an *internal node*.

Definition 2.4 (Lowest common ancestor)**.** For vertices u, v in a rooted tree T, the *lowest common ancestor* $\mathrm{LCA}_T(u, v)$ is the deepest vertex that is an ancestor of both u and v.

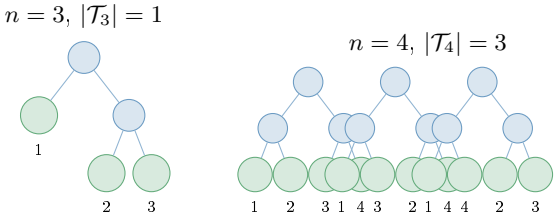


Figure 2.2: All distinct rooted binary tree topologies for $n = 3$ (left) and $n = 4$ (right), illustrating the combinatorial explosion $|\mathcal{T}_n| = (2n-3)!!$. For $n = 5$ there are already 15 topologies, and for $n = 10$ there are 945. This superexponential growth defines the difficulty of the inference problem.

Definition 2.5 (Binary tree)**.** A rooted tree is *binary* if every internal node has exactly two children.

2.2 Subtrees and Induced Structures

Definition 2.6 (Subtree)**.** For a set $S \subseteq L(T)$ of leaves, the *subtree induced by* S, denoted $T[S]$, is the minimal connected subgraph of T containing all vertices of S.

Notation 2.7. We write $T[i \vee j]$ for the subtree rooted at $\mathrm{LCA}_T(i, j)$.

Definition 2.8 (Laminar family)**.** A collection $\mathcal{F}$ of subsets of a ground set V is a *laminar family* if for every $A, B \in \mathcal{F}$,

$$A \cap B \in \{\varnothing, A, B\}.$$

Proposition 2.9. *The family of leaf sets of all subtrees of a rooted binary tree on n leaves forms a laminar family.*

Proof. Let $A = L(T[u])$ and $B = L(T[v])$. If u is an ancestor of v, then $B \subseteq A$. If v is an ancestor of u, then $A \subseteq B$. Otherwise $A \cap B = \varnothing$. □

2.3 Counting Rooted Binary Trees

Theorem 2.10 (Count of rooted binary trees)**.** *The number of distinct rooted binary trees on n labeled leaves is*

$$|\mathcal{T}_n| = (2n-3)!! = 1 \cdot 3 \cdot 5 \cdots (2n-3),$$

for $n \geq 2$, with $|\mathcal{T}_2| = 1$.

Proof. A tree on n leaves is constructed by inserting the n-th leaf into any of the $2n-3$ edges of a tree on $n-1$ leaves (splitting that edge inserts a new internal node). By induction, $|\mathcal{T}_n| = (2n-3) \cdot |\mathcal{T}_{n-1}| = (2n-3)!!$. □

Corollary 2.11 (Entropy of the uniform prior)**.** *The entropy of the uniform distribution over rooted binary tree topologies on n leaves satisfies*

$$H_0 := \log |\mathcal{T}_n| \sim n \log n + O(n) \qquad \text{as } n \to \infty.$$

This corollary is fundamental: the initial uncertainty over tree topologies grows as $n \log n$ bits, setting the information-theoretic scale for reconstruction.

2.4 Paths, Distances, and Depth

Definition 2.12 (Node height function)**.** A *node height function* assigns a nonnegative real value to each internal node, strictly decreasing toward the leaves, with $h(v) = 0$ for all leaves.

Node height functions connect tree structure to ultrametric geometry in Chapter 5.

2.5 Exercises

1. Prove that a tree on n vertices has exactly $n-1$ edges.
2. Verify the recurrence $|\mathcal{T}_n| = (2n-3)|\mathcal{T}_{n-1}|$ for $n = 3, 4, 5$.
3. Prove that the leaf sets of a rooted tree form a laminar family, and show by example that not every laminar family arises from a rooted binary tree.

4. Draw all 15 rooted binary tree topologies on $n = 5$ labeled leaves.

5. Compute $H_0 = \log |\mathcal{T}_n|$ for $n = 5, 10, 20$ and comment on the rate of growth.

Chapter 3

Probability, Concentration, and Noisy Observations

> Noise is not the enemy of inference. It is the cost of sampling.
>
> —*Flyxion*

This chapter develops the probabilistic tools needed to work with the splitting oracle, which is the central device of Parts IV and V. The oracle answers queries correctly with probability $p > 1/2$—providing a weak but consistent signal in favor of the truth. The challenge is to aggregate many such weak signals into a reliable conclusion. We derive Hoeffding's inequality, which quantifies how rapidly the probability of a majority-vote error decays as the number of queries grows, and we compute the sample complexity: the minimum number of queries needed to achieve any desired level of reliability. We also introduce the union bound, which allows us to extend per-triplet guarantees to uniform guarantees over all triplets simultaneously. The chapter closes with the Bayesian framing of oracle aggregation, which connects the algorithmic oracle model to the inference framework developed in Part VIII.

3.1 Probability Spaces and Random Variables

We assume familiarity with the standard axioms of probability. A *probability space* is a triple $(\Omega, \mathcal{F}, \mathbb{P})$ where Ω is a sample space, $\mathcal{F}$ a σ-algebra of events, and $\mathbb{P}$ a probability measure.

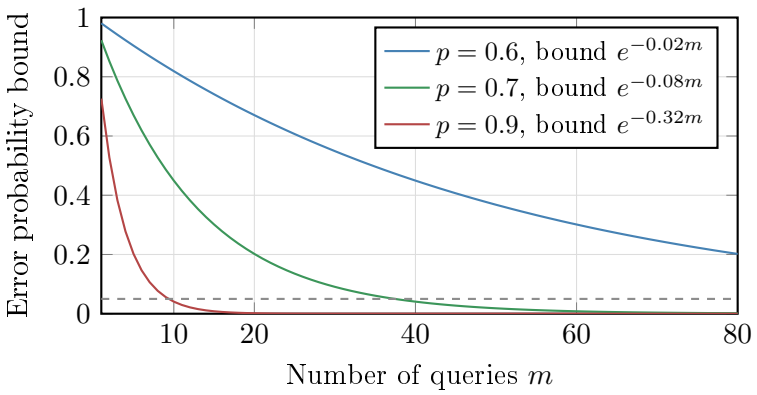


Figure 3.1: Hoeffding's bound $\exp(-2m(p-\frac{1}{2})^2)$ on the majority-vote error probability as a function of the number of queries m, for three oracle accuracies $p \in \{0.6, 0.7, 0.9\}$. Even a weakly biased oracle ($p = 0.6$) achieves error below 5% with around 150 queries; a stronger oracle ($p = 0.9$) achieves the same with fewer than 20.

3.2 The Noisy Oracle Model

Definition 3.1 (Biased Bernoulli)**.** A random variable $X \sim \mathrm{Ber}(p)$ satisfies $\mathbb{P}(X = 1) = p$ and $\mathbb{P}(X = 0) = 1 - p$.

When the oracle is queried m times on the same triplet, each query produces an independent $\mathrm{Ber}(p)$ outcome. The majority vote is correct whenever more than $m/2$ responses are correct.

3.3 Concentration Inequalities

Theorem 3.2 (Hoeffding's inequality [6])**.** *Let $X_1, \dots, X_m$ be independent with $X_i \in [0, 1]$ and $\mathbb{E}[X_i] = p$. Then*

$$\mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} X_i \le \frac{1}{2}\right) \le \exp\!\left(-2m(p-\tfrac{1}{2})^2\right).$$

Corollary 3.3 (Sample complexity for reliable oracle aggregation)**.** *To reduce the probability of a majority-vote error to at most $\delta > 0$, it*

suffices to make

$$m \geq \frac{\log(1/\delta)}{2(p-\frac{1}{2})^2}$$

queries per triplet.

3.4 Union Bounds

Lemma 3.4 (Union bound). $\mathbb{P}\left(\bigcup_{i=1}^{k} A_i\right) \leq \sum_{i=1}^{k} \mathbb{P}(A_i)$.

Corollary 3.5 (Uniform concentration). *With* $m = O\left(\frac{\log(n^3/\delta)}{(p-\frac{1}{2})^2}\right)$ *queries per triplet, all* $\binom{n}{3}$ *triplets are answered correctly by majority vote simultaneously with probability* $\geq 1-\delta$.

3.5 Bayesian Updating

The Bayesian update formula connects oracle aggregation to inference over tree topologies. Given a prior $P(T)$ over tree topologies and a triplet observation r_t,

$$P(T \mid r_t) \propto P(r_t \mid T)\, P(T).$$

After t conditionally independent observations,

$$P(T \mid r_1, \ldots, r_t) \propto P(T) \prod_{i=1}^{t} P(r_i \mid T).$$

This Bayesian framing connects directly to the inference framework developed in Part VIII.

3.6 Chernoff Bounds

Theorem 3.6 (Chernoff bound). *Let* $X = \sum_{i=1}^{m} X_i$ *with* $X_i \sim \mathrm{Ber}(p_i)$ *independent and* $\mu = \mathbb{E}[X]$. *For* $\delta \in (0,1)$,

$$\mathbb{P}(X \leq (1-\delta)\mu) \leq \exp\left(-\frac{\delta^2 \mu}{2}\right).$$

3.7 Exercises

1. Use Corollary 3.3 to compute the number of queries needed per triplet when $p \in \{0.6, 0.75, 0.9\}$, at confidence level $\delta = 0.01$.

2. Let $p = 0.51$. How many queries are needed to achieve failure probability at most n^{-3} for $n = 1000$ leaves where $N = \binom{n}{3}$ triplets are queried?

3. Write out the Bayesian update for a uniform prior over the $|\mathcal{T}_3| = 3$ topologies on three leaves when the oracle returns $(u, v)|w$ with probability p if that relation holds and $(1-p)/2$ for each incorrect answer.

4. Prove the union bound from the axioms of probability.

5. Show that majority vote of any number of queries still gives error probability $1/2$ when $p = 1/2$, and explain why Hoeffding's inequality does not apply.

Chapter 4

Entropy and Information Theory

> Information is not what you know. It is what you did not know before.
>
> —*Flyxion*

This chapter develops the information-theoretic language that frames the entire reconstruction enterprise. The central object is Shannon entropy, which measures the uncertainty of a probability distribution and therefore quantifies the difficulty of the inference problem at any stage of the algorithm [4]. We begin with the entropy of the uniform distribution over tree topologies—the initial uncertainty before any observations are made—and derive how each triplet observation reduces this entropy. The key monotonicity result, Proposition 4.5, shows that entropy can only decrease as constraints accumulate: observations never increase uncertainty, though some may be redundant. We then derive a lower bound on the total number of queries needed for reconstruction, establishing that the algorithm's query complexity is fundamentally information-theoretically optimal up to logarithmic factors. The chapter closes with mutual information and KL divergence, tools that will reappear in the variational reformulation of tree inference in Appendix E.

4.1 Shannon Entropy

Definition 4.1 (Shannon entropy)**.** Let X be a discrete random variable on a finite set $\mathcal{X}$ with mass function $P(x)$. The *Shannon*

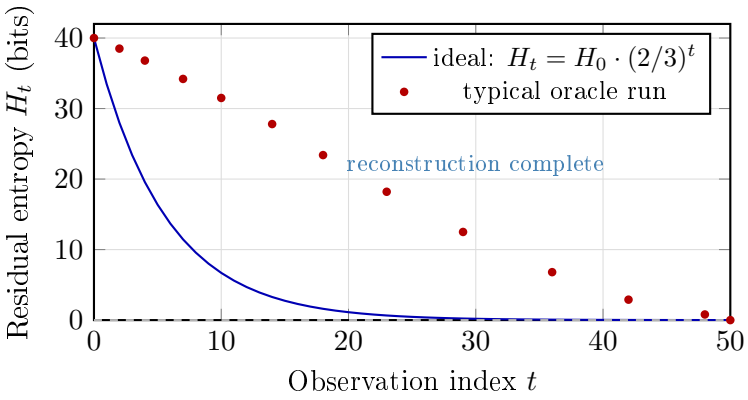


Figure 4.1: Entropy of the hypothesis space $H_t = \log |\mathcal{T}_n^{\mathcal{R}_t}|$ as a function of the number of triplet observations. Each observation reduces H_t monotonically. The ideal curve assumes each triplet eliminates exactly $2/3$ of remaining trees; the dotted points show a typical oracle run with variable information gain per query.

entropy is

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x),$$

with convention $0 \log 0 = 0$. Logarithms are in base 2 unless noted.

Proposition 4.2 (Bounds on entropy)**.** *For $|\mathcal{X}| = k$: (1) $H(X) \geq 0$, with equality iff X is deterministic; (2) $H(X) \leq \log k$, with equality iff X is uniform.*

4.2 Joint Entropy, Conditional Entropy, and Mutual Information

Definition 4.3 (Conditional entropy and mutual information)**.**

$$H(X \mid Y) = H(X,Y) - H(Y), \qquad I(X;Y) = H(X) - H(X \mid Y).$$

Proposition 4.4. *$I(X;Y) \geq 0$, with equality iff X and Y are independent.*

4.3 Entropy as Uncertainty over Tree Topologies

Let T be uniform over $\mathcal{T}_n$. Then

$$H_0 := H(T) = \log |\mathcal{T}_n| \sim n \log n + O(n)$$

by Corollary 2.11. After observing $\mathcal{R}_t = \{r_1, \ldots, r_t\}$, the residual uncertainty is

$$H_t = \log |\mathcal{T}_n^{\mathcal{R}_t}|.$$

Proposition 4.5 (Monotone entropy descent [4]). *$H_{t+1} \leq H_t$ for all $t \geq 0$.*

Proposition 4.6 (Information per triplet). *Each consistent triplet observation eliminates approximately two-thirds of remaining trees, contributing approximately $\log(3/2) \approx 0.585$ bits of information under the uniform prior.*

4.4 Information-Theoretic Lower Bounds on Query Complexity

Theorem 4.7 (Query complexity lower bound). *Any algorithm that reconstructs the tree topology with high probability must make at least $\Omega(n \log n)$ queries to a noiseless triplet oracle.*

Proof. The total information to determine is $H_0 = \Theta(n \log n)$ bits. Each query reveals at most $\log 3 < 2$ bits. Therefore at least $H_0 / \log 3 = \Omega(n \log n)$ queries are required. □

4.5 KL Divergence and Model Comparison

Definition 4.8 (KL divergence).

$$\mathrm{KL}(P \| Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}.$$

$\mathrm{KL}(P \| Q) \geq 0$ with equality iff $P = Q$. KL divergence appears in the variational reformulation of tree inference (Appendix E), where the inference objective decomposes into an expected log-likelihood and a KL regularization term.

4.6 Exercises

1. Compute $H(T)$ for $n = 4, 5, 6$ under the uniform distribution over rooted binary tree topologies.

2. Show that observing a consistent triplet reduces $H(T)$ by exactly $\log(3/2)$ bits when the prior is uniform and the three triplet relations are equally probable a priori.

3. Show that the total number of triplet queries sufficient for noiseless reconstruction is $O(n \log n)$ by arguing each query can be chosen to eliminate at least a constant fraction of remaining consistent trees.

4. Compute $\mathrm{KL}(P \| Q)$ when $P = \mathrm{Ber}(p)$ and $Q = \mathrm{Ber}(q)$, and show it is zero iff $p = q$.

Chapter 5

Metric Spaces and Ultrametric Geometry

> In an ultrametric world, every triangle is isosceles. This is not a limitation. It is the signature of a tree.
>
> —*Flyxion*

This chapter introduces the geometric framework that underlies hierarchical clustering. Ordinary metric spaces capture notions of distance in Euclidean and more general spaces, but hierarchical structure corresponds to a strictly stronger geometric condition: the ultrametric inequality. Whereas the ordinary triangle inequality permits triangles of any shape, the ultrametric inequality forces every triangle to be isosceles—the two longer sides must be equal. This isosceles property is not an incidental curiosity; it is the precise geometric signature of a tree. We prove that every rooted tree with a height function induces an ultrametric on its leaves, and we state the converse—that every finite ultrametric arises from a rooted tree—which is proved in full in Appendix A. Understanding this equivalence is the key to interpreting the splitting oracle geometrically in Chapter 20: each oracle response reveals which of the three ultrametric inequality patterns holds on a given triple, effectively measuring the branching curvature of the hidden tree.

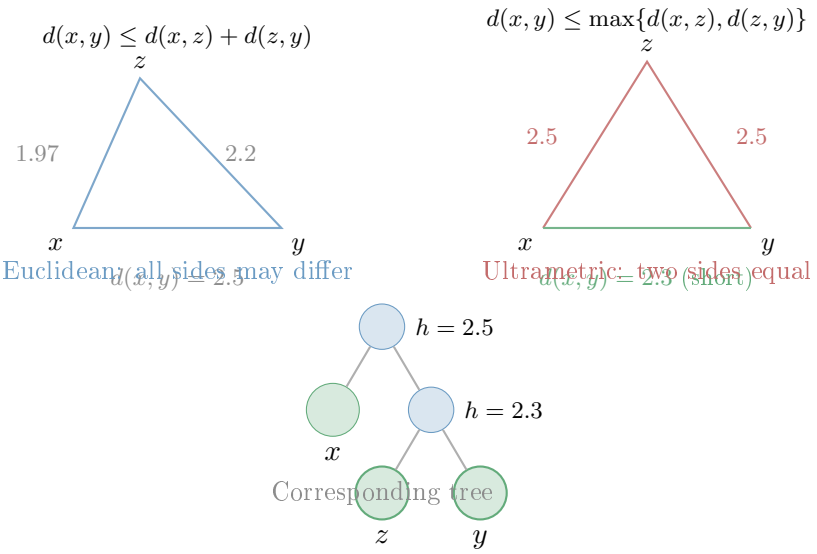


Figure 5.1: Left: A Euclidean triangle permits all three sides to have different lengths. Center: An ultrametric triangle is always isosceles—the two longest sides are equal. Right: The corresponding rooted tree, where $d_T(u,v) = h(\mathrm{LCA}_T(u,v))$. The short side of the ultrametric triangle corresponds to the pair sharing the deeper LCA.

5.1 Metric Spaces

Definition 5.1 (Metric)**.** A *metric* on X is a function $d : X \times X \to \mathbb{R}_{\ge 0}$ satisfying non-negativity, identity of indiscernibles ($d(x,y) = 0 \iff x = y$), symmetry, and the triangle inequality $d(x,y) \le d(x,z) + d(z,y)$.

5.2 Ultrametric Spaces

Definition 5.2 (Ultrametric)**.** A metric d is an *ultrametric* if it satisfies the *strong triangle inequality*:

$$d(x,y) \le \max\{d(x,z),\, d(z,y)\} \qquad \forall x, y, z \in X.$$

Proposition 5.3 (Isosceles property)**.** *In an ultrametric space, every triangle is isosceles: the two largest pairwise distances in any triple are equal.*

Proof. Let $d(x,y) \le d(x,z) \le d(y,z)$. The strong triangle inequality applied at x gives $d(y,z) \le \max\{d(y,x), d(x,z)\} = d(x,z)$, contradicting $d(x,z) \le d(y,z)$ unless $d(x,z) = d(y,z)$. □

5.3 Trees Define Ultrametrics

Definition 5.4 (Tree metric)**.** Let T be a rooted tree with node height function h. The *tree metric* on $L(T)$ is

$$d_T(u,v) = h(\mathrm{LCA}_T(u,v)).$$

Theorem 5.5 (Trees induce ultrametrics)**.** *For any rooted tree T with height function h, the function d_T is an ultrametric on $L(T)$.*

Proof. Non-negativity and symmetry are immediate. Identity of indiscernibles follows because $h(u) = 0$ for leaves and $h(\mathrm{LCA}(u,v)) > 0$ for $u \neq v$. For the strong triangle inequality, the three LCA values $\mathrm{LCA}(u,v)$, $\mathrm{LCA}(u,w)$, $\mathrm{LCA}(v,w)$ admit one that is an ancestor of the other two. If $\mathrm{LCA}(u,w)$ is the highest, then $d_T(u,v) \le d_T(u,w) = \max\{d_T(u,w), d_T(v,w)\}$. □

The full converse—every ultrametric arises from a rooted tree—is proved in Appendix A.

5.4 Ultrametric Inequalities and Branching Order

Proposition 5.6 (Ultrametric and LCA)**.** *For leaves u, v, w in a rooted tree T,*

$$d_T(u,v) < d_T(u,w) = d_T(v,w) \quad \Longleftrightarrow \quad \mathrm{LCA}_T(u,v) \text{ is a strict descendant of } \mathrm{LC}$$

This connects the metric formalism directly to the triplet relation framework: the triplet $(u,v) \mid w$ is equivalent to $d_T(u,v) < d_T(u,w) = d_T(v,w)$.

5.5 Exercises

1. Verify that the p-adic metric on the integers is an ultrametric, and describe the hierarchical structure it encodes.

2. Prove the isosceles property (Proposition 5.3) from the strong triangle inequality.

3. Let T be the tree on $\{1, 2, 3, 4\}$ with $\mathrm{LCA}(1, 2)$ at height 1, $\mathrm{LCA}(3, 4)$ at height 1, and root at height 2. Write the 4×4 ultrametric distance matrix.

4. Show that Euclidean distance on $\mathbb{R}^2$ is not an ultrametric by exhibiting a violating triple.

5. Reconstruct the rooted tree corresponding to the ultrametric: $d(1, 2) = 1$, $d(1, 3) = d(2, 3) = 2$, $d(1, 4) = d(2, 4) = d(3, 4) = 3$.

Chapter 6

Constraint Systems and Emergent Global Structure

> Global structure is not assembled. It is carved.
>
> —*Flyxion*

This chapter formalizes the constraint-accumulation process that is the organizing principle of the entire textbook. We define a constraint operator as a function that maps the hypothesis space to the subset consistent with a given observation, and we study how a sequence of such operators progressively narrows the space of viable hypotheses. The key theorem is that a sufficiently rich set of consistent triplet relations uniquely determines the underlying tree—global structure emerges from local relational observations through the accumulated pressure of constraint. We also introduce the probabilistic (soft-constraint) version of this framework, in which observations carry likelihoods rather than hard compatibility requirements, connecting the constraint view to Bayesian inference. The chapter ends by making precise the analogy between constraint propagation in tree reconstruction and constraint propagation in other complex systems, a theme that will recur in Part VII and Part VIII.

6.1 Constraint Satisfaction

A *constraint satisfaction problem* consists of a hypothesis space $\mathcal{H}$, a domain, and a set of constraints $\mathcal{C}$ specifying valid assignments. In

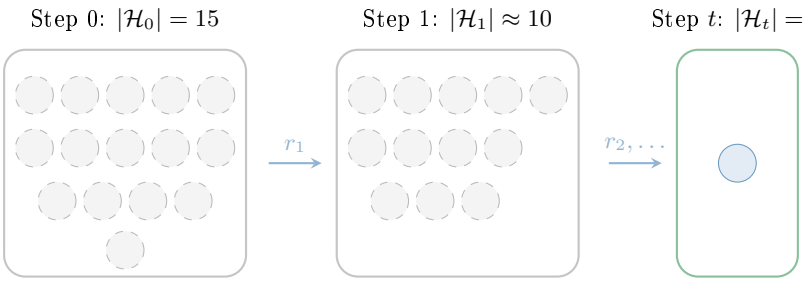


Figure 6.1: Constraint accumulation progressively narrows the hypothesis space $\mathcal{H}_t$. At step 0, all tree topologies are viable. Each triplet observation r_i eliminates inconsistent hypotheses, reducing $|\mathcal{H}_t|$ monotonically. At step t, a single tree remains: the reconstruction is complete.

the tree reconstruction setting, $\mathcal{H} = \mathcal{T}_n$ and each observed triplet relation r_i imposes a constraint.

Definition 6.1 (Constraint operator)**.** For hypothesis space $\mathcal{H}$ and observation o, the *constraint operator* is

$$C_o : \mathcal{H} \to 2^{\mathcal{H}}, \qquad C_o(\mathcal{H}) = \{H \in \mathcal{H} : H \text{ is consistent with } o\}.$$

Definition 6.2 (Accumulated constraint space)**.** After observations $o_1, \ldots, o_t$, the *accumulated constraint space* is

$$\mathcal{H}_t = C_{o_t} \circ \cdots \circ C_{o_1}(\mathcal{H}_0).$$

Proposition 6.3 (Monotone reduction)**.** $\mathcal{H}_{t+1} \subseteq \mathcal{H}_t$ *for all* $t \geq 0$*; constraint operators are commutative* ($C_a \circ C_b = C_b \circ C_a$)*.*

6.2 Propagation of Local Constraints

Proposition 6.4 (Triplet constraints propagate)**.** *A triplet observation* $(u, v)|w$ *constrains not only the local branching of* $\{u, v, w\}$ *but also the placement of every other leaf relative to this triple.*

This propagation is why $O(n \log n)$ judiciously chosen triplets suffice to reconstruct the entire tree. Each triplet contributes local information that, in combination with others, propagates to constrain the global topology.

6.3 Soft Constraints and Probabilistic Inference

When observations are noisy, hard elimination is replaced by likelihood weighting. The posterior distribution after t observations is

$$P_t(H) \propto P_0(H) \prod_{i=1}^{t} P(o_i \mid H).$$

Under the noisy triplet oracle model, $P(r_t \mid T) = p$ if $T \models r_t$ and $(1-p)/2$ otherwise.

6.4 Global Structure from Minimal Local Information

Theorem 6.5 (Convergence of constraint accumulation)**.** *A consistent set $\mathcal{R}$ of rooted triplet relations uniquely determines a rooted binary tree topology if and only if for every bipartition $\{A, B\}$ of the leaf set, there exists a triplet in $\mathcal{R}$ witnessing the partition.*

The proof is in Appendix B.

6.5 Exercises

1. Write out the constraint operators C_r for each of the three possible triplet relations on $\{1, 2, 3\}$ in a tree on four leaves. How many of the $|\mathcal{T}_4| = 3$ topologies does each operator eliminate?

2. Show the accumulated constraint space is independent of the order of observations.

3. Define a *redundant* observation as one that does not reduce $|\mathcal{H}_t|$. Give an example on four leaves.

4. Show that maximizing expected information gain per query is equivalent to choosing triplets that eliminate the largest expected fraction of consistent trees under the current posterior.

Part II

Hierarchical Structure and Minimal Relational Information

Chapter 7

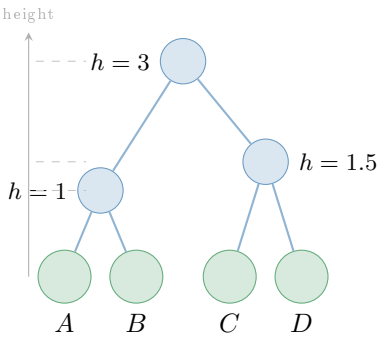
Hierarchical Clustering as a Branching Object

> A dendrogram is a hypothesis about the history of similarity.
>
> —*Flyxion*

This chapter introduces dendrograms—the primary data structure of hierarchical clustering—and shows how they can be interpreted both combinatorially, as laminar families of subsets, and geometrically, as ultrametric spaces. The key object is the merge height: the value attached to each internal node of a dendrogram that records the scale at which two clusters merged. Reading the dendrogram at a fixed height threshold produces a flat clustering; varying the threshold produces the full hierarchy. We present two classical algorithms for constructing dendrograms—agglomerative (bottom-up) and divisive (top-down)—and note that neither optimizes any explicitly stated global objective. This motivates the formal objectives introduced in Chapter 11 and the oracle-augmented methods of Part IV. The visual comparison between a dendrogram and its corresponding ultrametric distance matrix—shown side by side in Figure 7.1—illustrates concretely the equivalence proved in Appendix A.

7.1 Dendrograms and Laminar Families

Definition 7.1 (Dendrogram)**.** A *dendrogram* over a ground set V is a rooted binary tree T with $L(T) = V$ equipped with a node height



Ultrametric distance matrix d_T

	A	B	C	D
A	0	1	3	3
B	1	0	3	3
C	3	3	0	1.5
D	3	3	1.5	0

merge at $h = 1$
merge at $h = 1.5$
merge at $h = 3$

Figure 7.1: A dendrogram on four leaves (left) and its corresponding ultrametric distance matrix (right). Each internal node at height h contributes the value h to the distance between any pair of leaves whose LCA it is. The matrix encodes exactly the same information as the tree.

function $h : V(T) \to \mathbb{R}_{\geq 0}$.

The height function encodes the scale at which clusters merge. Reading the tree at threshold θ produces a flat clustering.

Definition 7.2 (Horizontal cut)**.** For a dendrogram T and threshold $\theta \geq 0$, the *θ-cut clustering* partitions V into parts that are maximal subtrees whose root has height $\leq \theta$.

Proposition 7.3. *The collection of all parts of all θ-cut clusterings forms a laminar family on V.*

7.2 Cluster Merges and the UPGMA Algorithm

Algorithm 7.4 (UPGMA / Average-linkage)**. Input**: Similarity matrix w_{ij}. **Initialize**: Clusters $\mathcal{C} = \{\{i\} : i \in V\}$. **Repeat** until $|\mathcal{C}| = 1$: find $A, B \in \mathcal{C}$ maximizing average inter-cluster similarity, merge into $A \cup B$ at height $h = \bar{w}(A, B)$, update $\mathcal{C}$. **Output**: the sequence of merge events defines a dendrogram T.

UPGMA runs in $O(n^2 \log n)$ but does not optimize any global objective.

7.3 Divisive Clustering

Divisive methods begin with the full dataset and recursively partition it. The splitting oracle model (Part IV) provides a principled way to perform the partition step using external information.

7.4 Exercises

1. Given $w(1,2) = 5$, $w(1,3) = 3$, $w(2,3) = 4$, apply UPGMA and draw the resulting dendrogram.

2. Show that single and complete linkage can produce different dendrograms on the same similarity matrix by exhibiting an example on four points.

3. From the dendrogram in Figure 7.1, read off the flat clustering at threshold $\theta = 2$.

Chapter 8

Ultrametric Geometry and Hierarchical Trees

> A finite ultrametric is a tree wearing a distance costume.
>
> —*Flyxion*

Chapter 5 showed that every rooted tree induces an ultrametric. This chapter proves the converse: every finite ultrametric space arises from a unique rooted tree. The proof is constructive—given an ultrametric, we build the corresponding tree by applying single-linkage clustering at every scale—and it illuminates why the tree–ultrametric equivalence is more than a formal curiosity. It means that the entire apparatus of tree algorithms can be brought to bear on any problem formulated in terms of ultrametric distances, and vice versa. A full proof with uniqueness is given in Appendix A; this chapter sketches the construction and then focuses on the geometric consequences: how ultrametric inequalities encode branching order, and how the oracle response to a triplet query is precisely a local ultrametric measurement.

8.1 The Equivalence of Trees and Ultrametrics

Theorem 8.1 (Ultrametrics correspond to trees)**.** *Let (X, d) be a finite ultrametric space. Then there exists a rooted tree T with leaf*

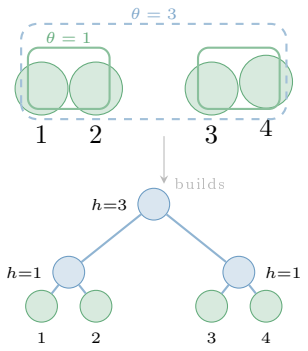


Figure 8.1: Threshold clustering at $\theta = 1$ reveals clusters $\{1, 2\}$ and $\{3, 4\}$; at $\theta = 3$ the whole set merges. The sequence of threshold clusterings is equivalent to the dendrogram shown below.

set X and height function h such that $d(x, y) = h(\mathrm{LCA}_T(x, y))$ for all $x, y \in X$.

Sketch. Apply single-linkage clustering. For each threshold θ, define $x \sim_\theta y$ if $d(x, y) \leq \theta$. The ultrametric property ensures these are genuine equivalence relations. The cluster hierarchy as θ varies defines the tree, with internal nodes assigned height equal to the threshold at which their cluster first appears. Full proof in Appendix A. □

8.2 Ultrametric Inequalities as Branching Signatures

The triplet relation $(u, v)|w$ is equivalent to the strict ultrametric inequality $d(u, v) < d(u, w) = d(v, w)$ by Proposition 5.6. This correspondence is the geometric heart of the triplet oracle model: the oracle reveals which ultrametric inequality holds on each triple, effectively measuring the branching geometry of the hidden tree.

8.3 Exercises

1. Prove that the single-linkage construction produces a valid rooted tree.

2. Show that the tree is unique when all distance values are distinct.

3. Given the ultrametric on four points with $d(1,2) = 1$, $d(3,4) = 2$, and $d(i,j) = 3$ for $i \in \{1,2\}$, $j \in \{3,4\}$, draw the corresponding dendrogram.

Chapter 9

Rooted Triplets and Minimal Topology

> Two points tell you a distance. Three points tell you a branch.
>
> —*Flyxion*

This chapter identifies the minimal unit of relational information capable of revealing hierarchical structure. A pair of leaves can be compared by their distance, but distance alone does not determine the branching order—two pairs at the same distance may sit in very different positions relative to the rest of the tree. Three leaves, by contrast, contain enough relational information to specify a branching event unambiguously: for any triple $\{u, v, w\}$, exactly one of the three pairs shares the deeper lowest common ancestor, and this determines the rooted triplet relation. We prove that the set of all triplet relations on n leaves distinguishes any two distinct tree topologies, establish the consistency conditions for triplet sets, and show that pairwise distance information alone cannot always determine topology. The figure below makes the key point visually: three leaves admit exactly three branching configurations, and the oracle's job is to identify which one holds.

9.1 Triplet Relations

Definition 9.1 (Rooted triplet)**.** For three distinct leaves u, v, w in a rooted tree T, the *rooted triplet relation* is

$$(u, v) \mid w \quad \Longleftrightarrow \quad \mathrm{LCA}_T(u, v) \text{ is a strict descendant of } \mathrm{LCA}_T(u, w).$$

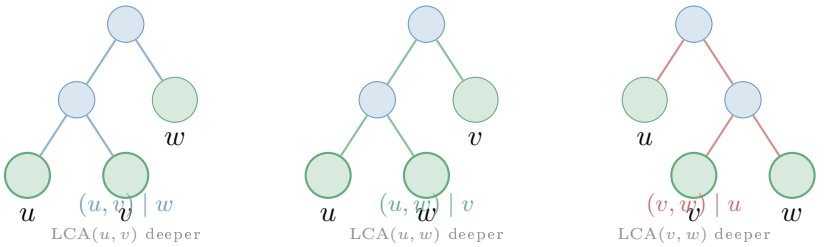


Figure 9.1: The three possible rooted triplet relations on leaves $\{u, v, w\}$. In each case exactly one pair shares the deeper LCA, which corresponds to the shorter ultrametric distance: $d(u, v) < d(u, w) = d(v, w)$ for $(u, v) \mid w$. Exactly one of these three configurations holds in any rooted binary tree.

Proposition 9.2 (Trichotomy)**.** *For any three distinct leaves u, v, w in a rooted binary tree, exactly one of $(u, v) \mid w$, $(u, w) \mid v$, or $(v, w) \mid u$ holds.*

9.2 Why Three Leaves Suffice

Proposition 9.3 (Pairwise distance is insufficient)**.** *The complete matrix of pairwise distances determines the tree topology if and only if all inter-leaf distances are distinct.*

Proposition 9.4 (Triplets determine topology)**.** *For any two distinct rooted binary trees $T_1 \neq T_2$ on the same leaf set, there exist three leaves u, v, w such that the triplet relation on $\{u, v, w\}$ differs between T_1 and T_2.*

9.3 Triplet Consistency

Definition 9.5 (Consistent triplet set)**.** A set $\mathcal{R}$ of triplet relations is *consistent* if there exists a rooted binary tree T with $T \models r$ for all $r \in \mathcal{R}$.

Proposition 9.6 (Incompatible triplets)**.** *The triplets $(1, 2) \mid 3$ and $(1, 3) \mid 2$ and $(2, 3) \mid 1$ on three leaves are mutually incompatible.*

9.4 Exercises

1. List all rooted triplet relations for the tree on four leaves where $\mathrm{LCA}(1,2)$ is at depth 2 and $\mathrm{LCA}(3,4)$ is at depth 2 and the root is at depth 1.

2. Show that a triplet set containing all three relations on some triple is inconsistent.

3. Describe an efficient algorithm for checking whether a given triplet set is consistent, and state its time complexity.

4. Give an example of a pairwise distance matrix on four points that is consistent with two different rooted binary tree topologies, demonstrating that pairwise distances alone are insufficient.

Chapter 10

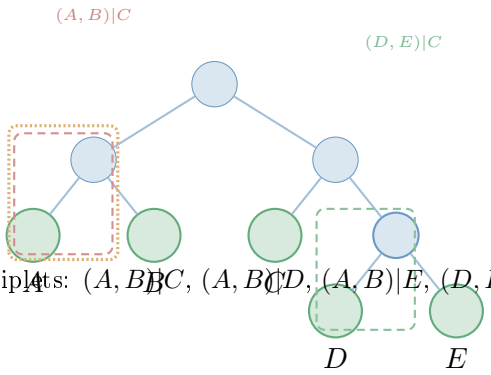
Reconstruction from Triplets

> A jigsaw puzzle is solvable even without seeing the box, provided enough pieces constrain each other.
>
> —*Flyxion*

This chapter addresses the algorithmic core of Part II: given a set of observed triplet relations, how does one efficiently reconstruct the underlying tree? The BUILD algorithm of Aho et al. [1] solves this problem in linear time by reducing it to a sequence of graph connectivity queries—at each level, the separation graph encodes which leaves can share a subtree, and its connected components identify the partition. We prove that BUILD succeeds if and only if the input triplet set is consistent, and we prove the fundamental reconstruction theorem: a complete, consistent triplet set uniquely determines the tree topology. We close with the sparse reconstruction result—that $O(n \log n)$ triplets suffice—which anticipates the oracle-based query strategies of Part IV.

10.1 The Reconstruction Problem

Definition 10.1 (Triplet reconstruction problem)**.** Given $\mathcal{R}$ on leaf set V, find a rooted binary tree T with $T \models r$ for all $r \in \mathcal{R}$, or determine none exists.



Five-leaf tree determined by triplets: $(A, B)|C$, $(A, B)|D$, $(A, B)|E$, $(D, E)|A$, $(D, E)|B$, $(D$

Figure 10.1: A rooted binary tree on five leaves $\{A, B, C, D, E\}$. The dashed rectangles highlight two of the six triplet relations that collectively determine the topology uniquely. No single triplet suffices; together they constrain every internal branch point.

10.2 The BUILD Algorithm

Algorithm 10.2 (BUILD [1]). **Input**: Leaf set V, triplet set $\mathcal{R}$. **If** $|V| = 1$: return a leaf. **Construct** the *separation graph*: vertices V; edge (u, v) iff $(u, v)|w \in \mathcal{R}$ for some w. **Find** connected components $V_1, \ldots, V_k$. **If** $k = 1$: return FAIL (inconsistency detected). **Recurse** on each V_i with restricted triplet set. **Return** tree rooted at new node with k subtrees.

Theorem 10.3 (Correctness of BUILD). *BUILD returns a consistent tree iff $\mathcal{R}$ is consistent.*

10.3 Sufficiency for Reconstruction

Theorem 10.4 (Dense triplets determine the tree). *If $\mathcal{R}$ contains a consistent triplet for every triple $\{u, v, w\} \subseteq V$, then $\mathcal{R}$ uniquely determines the tree topology.*

Theorem 10.5 (Sparse reconstruction). *There exists a set $\mathcal{R}$ of $O(n \log n)$ rooted triplets that uniquely determines any rooted binary tree on n leaves.*

10.4 Exercises

1. Apply BUILD to $\{(1,2)|3,\ (1,2)|4,\ (3,4)|1\}$ on four leaves.
2. Construct a minimum triplet set uniquely determining the balanced binary tree on $n = 8$ leaves.
3. Show $\Omega(n \log n)$ triplets are necessary for reconstruction by an information-theoretic argument.

Part III

Classical Objectives and Hardness

Chapter 11

Similarity-Based Hierarchical Clustering Objectives

> Before you can say an algorithm is good, you need to say what good means.
>
> —*Flyxion*

The algorithms of Chapter 7 construct hierarchies by greedy local merges, without optimizing any explicit global criterion. This chapter introduces two formal objectives that make precise what it means for a hierarchy to faithfully represent a given similarity structure. The first, due to Dasgupta [5], defines a cost function that penalizes merging dissimilar pairs at low levels of the tree. The second, due to Moseley and Wang [8], is a reward formulation that incentivizes merging similar pairs at low levels. We prove that the two objectives are equivalent up to a constant, explain their graph-theoretic interpretation in terms of weighted cuts, and illustrate with Figure 11.1 how the cost is distributed across different merge events in a simple example. The hardness results of Chapter 12 will show that optimizing these objectives is computationally difficult without the oracle access introduced in Part IV.

11.1 Formalizing the Quality of a Hierarchy

Given a similarity matrix $w : V \times V \to \mathbb{R}_{\geq 0}$, two natural objectives formalize what it means for a hierarchy to represent the similarity structure.

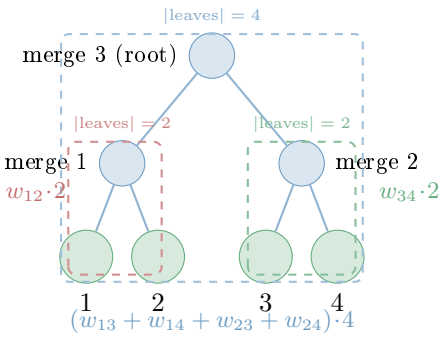


Figure 11.1: The Dasgupta cost $\text{cost}_D(T) = \sum_{i<j} w_{ij}|\text{leaves}(T[i \vee j])|$ for a tree on four leaves. Each pair $\{i, j\}$ contributes w_{ij} multiplied by the number of leaves in the subtree rooted at their LCA. Similar pairs (large w_{ij}) should merge early (small subtree size) to minimize cost.

11.2 The Dasgupta Objective

Definition 11.1 (Dasgupta cost [5])**.**

$$\text{cost}_D(T) = \sum_{\{i,j\} \subseteq V} w_{ij} \cdot |\text{leaves}(T[i \vee j])|.$$

The intuition: similar pairs (large w_{ij}) should merge at low levels (small $|\text{leaves}(T[i \vee j])|$).

11.3 The Moseley–Wang Objective

Definition 11.2 (Moseley–Wang reward [8])**.**

$$\text{reward}_{MW}(T) = \sum_{\{i,j\} \subseteq V} w_{ij} \cdot (n - |\text{leaves}(T[i \vee j])|).$$

Proposition 11.3. $\text{cost}_D(T) + \text{reward}_{MW}(T) = n \sum_{i<j} w_{ij}$ *is constant over all trees. Minimizing* cost_D *is equivalent to maximizing* reward_{MW}*.*

11.4 Relationship to Graph Cuts

Proposition 11.4 (Cut representation). $\text{cost}_D(T) = \sum_{e \in E(T)} w(e_{\text{cut}})$, *where* $w(e_{\text{cut}})$ *is the total similarity weight cut by removing internal edge* e.

11.5 Exercises

1. Compute $\text{cost}_D(T)$ for both binary tree topologies on four leaves with uniform similarities $w_{ij} = 1$.

2. For path-graph similarities (only adjacent leaves have $w_{ij} > 0$), show the optimal Dasgupta tree is the balanced binary tree.

3. Derive the cut representation of $\text{cost}_D(T)$ from Definition 11.1.

Chapter 12

Approximation Barriers and Complexity

> Hardness is not a failure of algorithms. It is information about the problem.
>
> —*Flyxion*

This chapter establishes the computational barriers that motivate the learning-augmented approach. Optimizing the Dasgupta objective is NP-hard, and under the Small Set Expansion (SSE) hypothesis [2, 3], no polynomial-time algorithm can achieve a constant-factor approximation using only the pairwise similarity matrix. The best oblivious approximation achieves ratio $O(\sqrt{\log n})$ via semidefinite programming relaxations. For the Moseley–Wang objective, APX-hardness is known: no PTAS exists unless P = NP. These results are not simply limitations to be complained about; they reveal that the optimization landscape is genuinely complex, with many locally good solutions far from the global optimum. The figure below illustrates intuitively how the graph partition structure that drives hardness manifests in the clustering objective: a wrong top-level split affects all cross-cluster pairs and accumulates a large penalty.

12.1 NP-Hardness of Optimal HC

Theorem 12.1 (Hardness of Dasgupta objective [5])**.** *Optimizing* $\mathrm{cost}_D(T)$ *over all rooted binary tree topologies is NP-hard. Under the SSE hypothesis, no polynomial-time algorithm achieves a constant-factor approximation.*

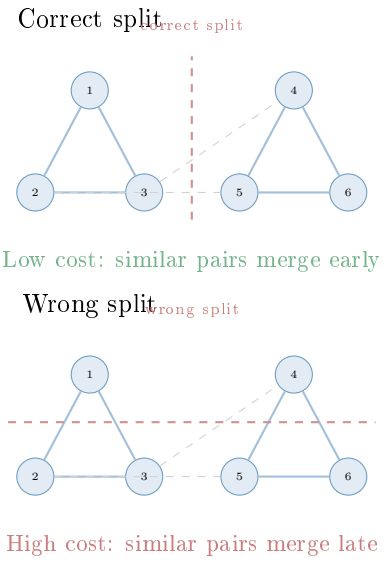


Figure 12.1: A graph with two dense communities. The correct split (left) separates communities at the top level, so similar pairs merge in small subtrees—low Dasgupta cost. The wrong split (right) cuts across communities, forcing similar pairs to merge late in large subtrees—high cost. The difficulty of finding the correct split is the source of hardness.

12.2 Known Approximation Results

The best polynomial-time oblivious approximation for cost_D achieves ratio $O(\sqrt{\log n})$ via semidefinite programming [10]. For reward_{MW}, APX-hardness holds: no PTAS exists unless P = NP [3].

12.3 Small Set Expansion and Its Role

Definition 12.2 (SSE hypothesis)**.** It is NP-hard to distinguish between graphs with a set of δn vertices of expansion $< \varepsilon$ and graphs where all such sets expand by $> 1 - \varepsilon$, for every $\delta > 0$ and some $\varepsilon(\delta) > 0$.

Under SSE, the $O(\sqrt{\log n})$ approximation for Dasgupta is essentially optimal for oblivious algorithms. The learning-augmented algo-

rithms of Part IV break this barrier by exploiting oracle information.

12.4 Exercises

1. Describe the reduction from graph partitioning to the Dasgupta objective optimization problem.

2. Explain why APX-hardness of reward_{MW} does not contradict the $(1-o(1))$-approximation achieved in the learning-augmented setting.

3. What feature of the oracle makes it possible to break the SSE-based lower bound?

Part IV

Learning-Augmented Clustering and the Splitting Oracle

Chapter 13

The Splitting Oracle Model

> A weak signal, repeated enough times,
> becomes certain knowledge.
>
> —*Flyxion*

This chapter introduces the splitting oracle, the central device that enables the learning-augmented algorithms to break the classical hardness barriers. The oracle is modeled as an imperfect external source of information that can answer questions about the structure of the unknown optimal tree T^*: given a triplet (u, v, w), the oracle returns the leaf that splits away from the other two earliest in T^*. The answer is noisy—correct with probability $p > 1/2$—but the noise is handled by majority vote over repeated queries, using the concentration inequalities of Chapter 3. We derive the per-triplet sample complexity, apply the union bound to obtain uniform correctness over all triplets simultaneously, and observe that the oracle's response is, geometrically, a local measurement of the ultrametric structure of T^*—a point elaborated in Chapter 20.

13.1 Definition of the Splitting Oracle

Definition 13.1 (Splitting oracle)**.** The *splitting oracle* $\mathcal{O}$ takes a triplet (u, v, w) and returns the leaf $x \in \{u, v, w\}$ such that $\mathrm{LCA}_{T^*}(u, v, w)$ separates x from the other two first. Equivalently, $\mathcal{O}$ returns the triplet relation that holds in T^*.

Definition 13.2 (Noisy oracle)**.** The oracle is (p)-*noisy* if it returns

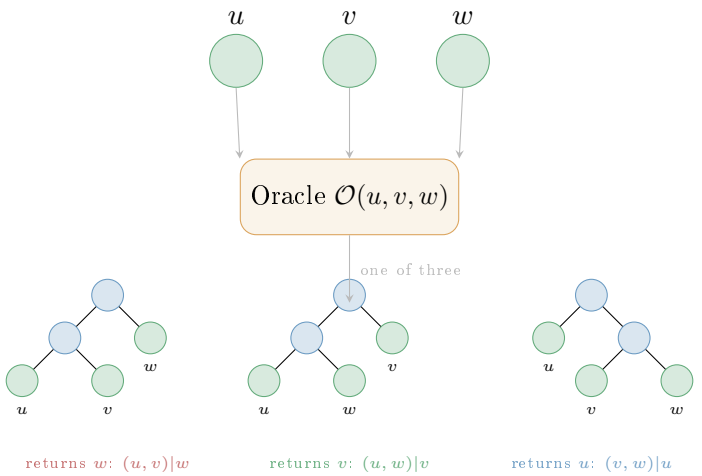


Figure 13.1: The splitting oracle $\mathcal{O}(u, v, w)$ receives a triplet query and returns one of three possible answers, each corresponding to a different rooted tree topology on the three leaves. The correct answer is returned with probability $p > 1/2$; incorrect answers are returned with probability $1-p$ (split uniformly between the two wrong options). Majority vote over m repeated queries yields the correct answer with high probability.

the correct answer with probability $p > 1/2$ and an incorrect answer with probability $1 - p$, independently across queries.

13.2 Oracle Aggregation

Proposition 13.3 (Reliable aggregation)**.** *Using* $m = O\left(\frac{\log(n^3/\delta)}{(p-1/2)^2}\right)$ *repeated queries per triplet, the plurality vote is correct for all* $O(n^3)$ *triplet queries simultaneously with probability* $\geq 1 - \delta$*.*

Proof. Apply Hoeffding's inequality (Theorem 3.2) per triplet, then the union bound over $\binom{n}{3}$ triplets. □

13.3 What the Oracle Reveals

The oracle reveals the ultrametric branching structure of T^* locally. Each response specifies which of the three ultrametric inequalities holds on the queried triple, providing direct geometric information about T^*'s structure without revealing the full tree. This geometric interpretation is elaborated in Chapter 20.

13.4 Exercises

1. Show that a single oracle query reduces uncertainty about a specific triple from $\log 3$ bits to at most $H(p, (1-p)/2, (1-p)/2)$ bits.

2. Derive the minimum number of queries per triplet needed to achieve error probability $\leq 1/n^3$ as a function of p.

3. Argue why the oracle model is more powerful than having access to a noisy estimate of the pairwise similarity matrix.

Chapter 14

Partial Hierarchical Clustering Trees

> The honest answer to an unanswerable question is a placeholder.
>
> —*Flyxion*

This chapter introduces the partial HC tree: a data structure that represents what the algorithm has reliably inferred about T^* while honestly acknowledging the regions that remain unresolved. The key idea is the super-vertex: a leaf of the partial tree that represents an entire cluster of original leaves whose internal topology the algorithm cannot yet determine reliably. Super-vertices arise naturally when a cluster shrinks below the logarithmic threshold $c \log n$, at which point the oracle's evidence becomes insufficient to justify further resolution without exceeding the query budget. We define two notions of consistency—strong and weak—and show that weakly consistent partial trees retain enough structural information to yield strong approximation guarantees for both the Dasgupta and Moseley–Wang objectives. The comparison in Figure 14.1 between a full tree and its partial counterpart is the most important single figure in the algorithmic development of this book.

14.1 Motivation: Representing Partial Knowledge

The oracle provides reliable information about the structure of T^* at large scales but becomes unreliable when a cluster shrinks below

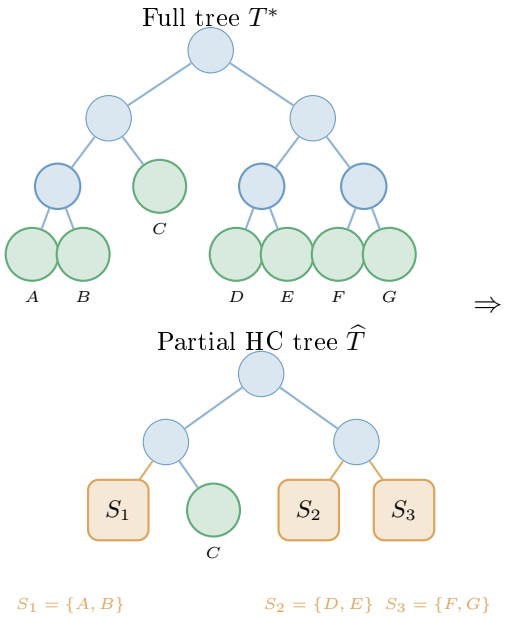


Figure 14.1: Left: the full optimal tree T^* on seven leaves. Right: the partial HC tree $\widehat{T}$ produced by the algorithm. Internal structure above the threshold is resolved (black nodes); clusters of size $\leq c \log n$ are collapsed into super-vertices (orange boxes). The top-level branching structure is correctly inferred; the unresolved structure within super-vertices is deferred.

$c \log n$ leaves. The partial HC tree captures the reliable portion while deferring resolution of ambiguous small clusters.

14.2 Definition of Partial HC Trees

Definition 14.1 (Super-vertex)**.** A *super-vertex* $\mathbf{v}$ is a leaf of the partial tree representing an unresolved cluster $S_{\mathbf{v}} \subseteq V$ with $|S_{\mathbf{v}}| \leq c \log n$.

Definition 14.2 (Partial HC tree)**.** A *partial HC tree* $\widehat{T}$ on leaf set V is a rooted binary tree whose leaves are either original vertices or super-vertices, together with a collapse map $\pi : V \to L(\widehat{T})$.

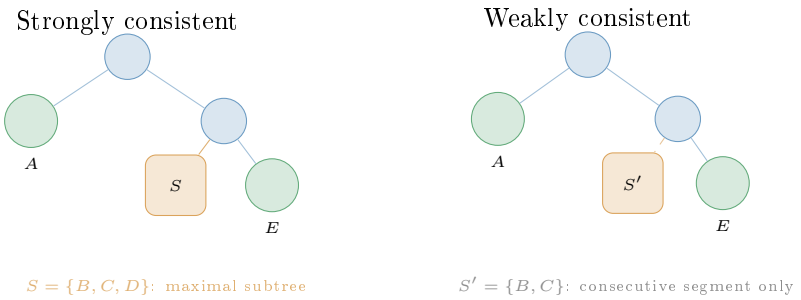


Figure 14.2: Strong consistency (left) requires each super-vertex to be the leaf set of a maximal subtree of T^*. Weak consistency (right) only requires the super-vertex's leaves to form a consecutive segment in T^*. Both notions preserve sufficient structure for the approximation guarantees of Chapter 16.

14.3 Strong and Weak Consistency

Definition 14.3 (Strongly consistent partial HC tree)**.** $\widehat{T}$ is *strongly consistent* with T^* if every super-vertex $\mathbf{v}$ satisfies: $S_{\mathbf{v}}$ is the leaf set of a maximal subtree of T^*.

Definition 14.4 (Weakly consistent partial HC tree)**.** $\widehat{T}$ is *weakly consistent* with T^* if every super-vertex $\mathbf{v}$ satisfies: $S_{\mathbf{v}}$ forms a consecutive segment in T^*.

Every strongly consistent tree is weakly consistent, but not vice versa.

14.4 Construction

Algorithm 14.5 (Partial HC tree construction sketch)**. Procedure** BuildPartial(S): (1) If $|S| \leq c \log n$: return super-vertex with $S_{\mathbf{v}} = S$. (2) Query oracle to determine top-level split $S = A \sqcup B$. (3) Recursively apply BuildPartial to A and B. (4) Return tree rooted at new internal node.

14.5 Exercises

1. Draw an example of a strongly consistent partial HC tree on $n = 8$ leaves with two super-vertices.

2. Show weak consistency is strictly weaker than strong consistency.

3. Explain why the threshold $c \log n$ is necessary: what goes wrong at $O(1)$ size?

Chapter 15

Top-Down Recursive Partitioning

> Divide until you can divide no further. Then stop.
>
> —*Flyxion*

This chapter details the algorithmic engine that builds the partial HC tree: the top-down recursive partitioning procedure. Starting from the full leaf set, the algorithm repeatedly identifies the correct top-level split of the current cluster by aggregating oracle responses over many sampled triplets, recurses on each half, and terminates by collapsing clusters below the $c \log n$ threshold into super-vertices. We prove the split identification theorem: for any cluster of size $\geq c \log n$, the correct partition is identified with high probability using $O(|S|^2 \log n)$ oracle queries. We then derive the total query count by summing over all levels of the recursion tree, establishing the $O(n^2 \operatorname{polylog} n)$ and $O(n^3)$ bounds that appear in the main results.

15.1 The Recursive Structure

At each level, the current cluster S is split into two subclusters by querying the oracle on triplets drawn from S and aggregating the responses.

Definition 15.1 (Correct split)**.** A partition $S = A \sqcup B$ is *correct* if it matches the top-level partition induced by T^* restricted to S.

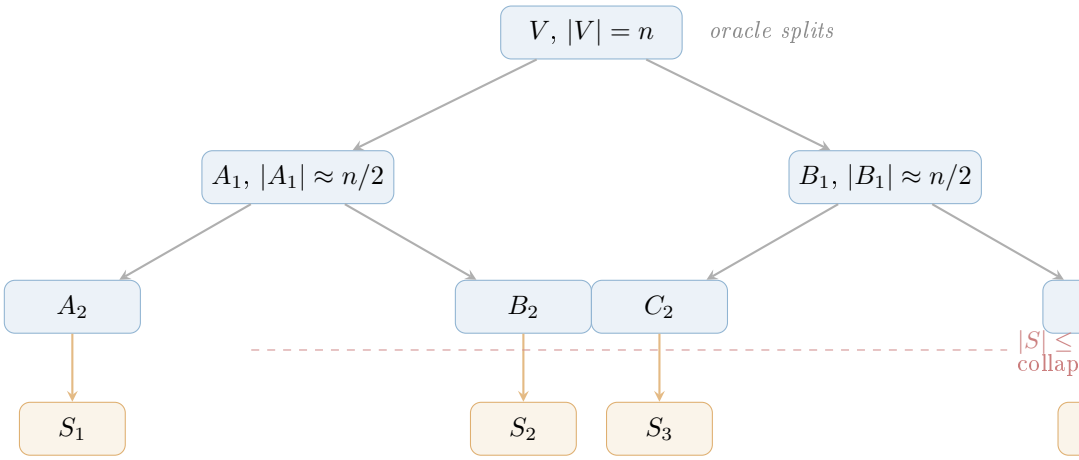


Figure 15.1: Recursive top-down partitioning. Each cluster is split by querying the oracle on sampled triplets. The recursion terminates when a cluster reaches size $\leq c \log n$, at which point it is collapsed into a super-vertex (orange boxes). The recursion depth is $O(\log n)$; the total work is dominated by the per-level oracle query cost.

15.2 Identifying the Split from Oracle Queries

Fix a pivot $u \in S$ and query (u, v, w) for sampled pairs $v, w \in S \setminus \{u\}$. Each query reveals whether v and w are on the same side of the split, and aggregating $O(|S|^2 \log n)$ queries identifies the correct partition with high probability.

Theorem 15.2 (Split identification)**.** *For any cluster S with $|S| \geq c \log n$, after $O(|S|^2 \log n)$ oracle queries (with $m = O(\log n/(p - 1/2)^2)$ repetitions each), the correct top-level split is identified with probability $\geq 1 - n^{-3}$.*

15.3 Recursion and Total Query Count

The recursion terminates at depth $O(\log n)$. At each level, clusters have total size n, and each cluster of size k uses $O(k^2 \log n)$ queries. The total is

$$T_q(n) = O(n^2 \log^2 n)$$

for the efficient variant, and $O(n^3)$ for the Dasgupta-objective algorithm.

15.4 Exercises

1. Solve the recursion $T(n) \leq 2T(n/2) + cn^2 \log n$ and compute the total.

2. Why is it critical that splits reduce cluster sizes by a constant factor at each level?

3. Describe how the pivot strategy can be made adaptive to minimize query count.

Part V

Approximation Guarantees and Algorithmic Consequences

Chapter 16

Approximation Guarantees

> Breaking a barrier means understanding why it was there in the first place.
>
> —*Flyxion*

This chapter presents the main approximation results of the book. The partial HC tree construction of Part IV yields surprising guarantees: a constant-factor approximation for the Dasgupta objective using $O(n^3)$ queries, and a near-optimal $(1-o(1))$-approximation for the Moseley–Wang reward using only $O(n^2 \operatorname{polylog} n)$ queries. These results break the classical barriers established in Chapter 12—barriers that hold for any algorithm using only the pairwise similarity matrix, without oracle access. The key mechanism is that weakly consistent partial trees preserve enough of T^*'s structure to control the objective error, with the remaining error coming only from pairs whose LCA falls inside a super-vertex. Appendix D proves that this error is small because super-vertices are small.

Objective	Approximation	Query complexity	Significance
Dasgupta	$O(1)$	$O(n^3)$	breaks SSE barrier
Dasgupta	$O(\sqrt{\log \log n})$	$O(n^3 \log n)$	beats oblivious $O(\sqrt{\log n})$
MW reward	$(1 - o(1))$	$O(n^2 \cdot \operatorname{polylog} n)$	breaks APX-hardness

Table 16.1: Main approximation results under the splitting oracle model [5, 8].

16.1 Approximation for the Dasgupta Objective

Theorem 16.1 (Constant-factor Dasgupta approximation)**.** *There exists an algorithm using $O(n^3)$ oracle queries returning $\widehat{T}$ with*

$$\mathrm{cost}_D(\widehat{T}) \le C \cdot \mathrm{cost}_D(T^*)$$

for an absolute constant C, with probability $\ge 1 - 1/n$.

Sketch. The partial HC tree is strongly consistent with T^* at all scales above $c \log n$. Error comes only from pairs whose LCA is inside a super-vertex. By Appendix D, this contribution is bounded by a constant times the optimal cost. □

16.2 Near-Optimal Moseley–Wang Reward

Theorem 16.2 (Near-optimal MW reward)**.** *There exists an algorithm using $O(n^2 \cdot \mathrm{polylog}\, n)$ queries returning $\widehat{T}$ with*

$$\mathrm{reward}_{MW}(\widehat{T}) \ge (1 - o(1)) \cdot \mathrm{reward}_{MW}(T^*),$$

with probability $\ge 1 - 1/n$.

16.3 Exercises

1. Verify $\mathrm{cost}_D(T) + \mathrm{reward}_{MW}(T) = \text{const}$ and explain why near-optimality for one implies near-optimality for the other.

2. Why does the constant-factor Dasgupta algorithm use more queries than the near-optimal MW algorithm?

Chapter 17

Query and Time Complexity

> Count what you compute before you compute what you count.
>
> —*Flyxion*

This chapter derives the query and time complexity of the learning-augmented algorithms from the recursion structure established in Chapter 15. The analysis proceeds by summing work across all levels of the recursion tree: at each level, clusters of total size n are processed, and each cluster of size k incurs $O(k^2 \log n)$ oracle queries. The total follows from the recursion $T(n) \leq 2T(n/2) + cn^2 \log n$, which solves to $O(n^2 \log^2 n)$ for the efficient variant. The computational time beyond oracle queries is dominated by aggregation and data structure maintenance, adding logarithmic factors. The lower bound from Chapter 4 confirms that these algorithms are near information-theoretically optimal.

17.1 Query Complexity

Theorem 17.1 (Total query count)**.** *The top-down partial HC tree construction uses $O(n^3)$ oracle queries for the Dasgupta objective and $O(n^2 \cdot \mathrm{polylog}\, n)$ for the Moseley–Wang objective.*

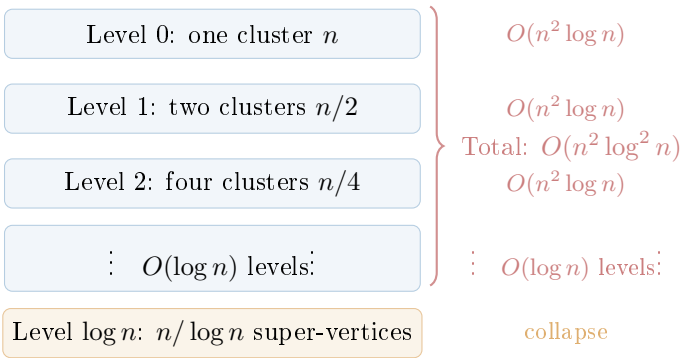


Figure 17.1: Work distribution across levels of the recursion tree. Each level processes clusters of total size n; the sum-of-squares property ensures the per-level work is $O(n^2 \log n)$. With $O(\log n)$ levels, the total is $O(n^2 \log^2 n)$ for the efficient algorithm.

17.2 Time Complexity

Theorem 17.2 (Time complexity)**.** *The algorithm runs in $O(n^3 \log n)$ time for Dasgupta and $O(n^2 \cdot \operatorname{polylog} n)$ time for Moseley–Wang, assuming $O(1)$ time per oracle query.*

17.3 Exercises

1. Solve $T(n) = 2T(n/2) + n^2$ and compare with $T(n) = 2T(n/2) + n^2 \log n$.
2. Explain why the $O(n^2 \operatorname{polylog} n)$ MW algorithm cannot obviously improve the Dasgupta guarantee.

Part VI

Streaming, Parallelism, and Compressed Inference

Chapter 18

Streaming Algorithms for Hierarchical Clustering

> Memory is the bottleneck. Use it wisely.
>
> —*Flyxion*

In many practical settings the similarity matrix is too large to store in memory, arriving instead as a stream of individual entries (i, j, w_{ij}) in arbitrary order. This chapter adapts the partial HC tree construction to this demanding setting, showing that a single-pass semi-streaming algorithm can maintain a compressed representation of the evolving partial tree using only $O(n \log^3 n)$ bits—far less than the $\Theta(n^2)$ bits needed to store the full matrix. The key insight is that the oracle's responses provide a compact summary of the relevant structure: instead of storing all pairwise similarities, the algorithm stores only the inferred partial tree and a buffer of recent oracle answers sufficient to update it. The streaming result is particularly striking because previous algorithms achieving $O(1)$ approximation required exponential time; the oracle model enables polynomial time within the streaming memory constraint.

18.1 The Streaming Model

Definition 18.1 (Semi-streaming algorithm)**.** A *semi-streaming algorithm* for hierarchical clustering is a single-pass streaming algorithm using $O(n \cdot \operatorname{polylog} n)$ bits of memory.

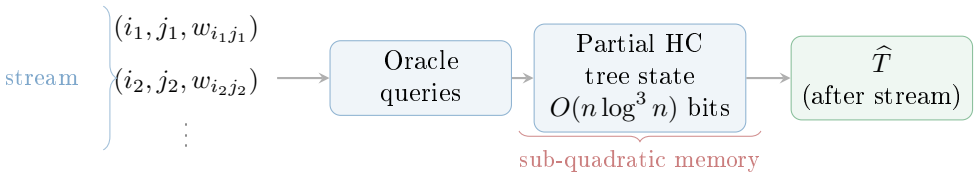


Figure 18.1: The streaming algorithm processes pairs (i, j, w_{ij}) in a single pass, querying the oracle as needed and maintaining a compact partial HC tree representation. The final tree $\widehat{T}$ is produced after the stream ends. Total memory is $O(n \log^3 n)$ bits, far below the $\Omega(n^2)$ needed to store the full similarity matrix.

18.2 Main Result

Theorem 18.2 (Streaming algorithm for Dasgupta [5])**.** *There exists a single-pass semi-streaming algorithm producing a partial HC tree achieving an $O(1)$-approximation to the Dasgupta objective, using $O(n \log^3 n)$ bits of space and polynomial time.*

18.3 Exercises

1. Show that any streaming algorithm for Dasgupta achieving a constant-factor approximation must use $\Omega(n)$ bits.

2. Describe how the partial HC tree can be updated incrementally as new similarity entries arrive.

3. Explain why the oracle model is especially powerful in the streaming setting: what information can be compressed that the similarity matrix cannot?

Chapter 19

Parallel Algorithms for Hierarchical Clustering

> Depth is the enemy of speed. Flatten the recursion.
>
> —*Flyxion*

The top-down recursive algorithm of Chapter 15 has $O(\log n)$ levels of depth, but within each level the split identification tasks for different clusters are entirely independent and can be parallelized. This chapter formalizes the PRAM model and shows that the oracle-augmented partial HC tree construction achieves near-linear total work $W(n) = n^{1+o(1)}$ with only polylogarithmic depth $D(n) = \text{polylog}(n)$. The key observation is that within each cluster, the $O(|S|^2)$ oracle queries are mutually independent and can be distributed across $O(|S|^2)$ processors, collapsing each level's contribution to $O(\log n)$ depth. Over $O(\log n)$ levels the total depth is $O(\log^2 n)$.

19.1 The PRAM Model

We measure *work* $W(n)$ (total operations) and *depth* $D(n)$ (parallel time, i.e., the longest sequential dependency chain).

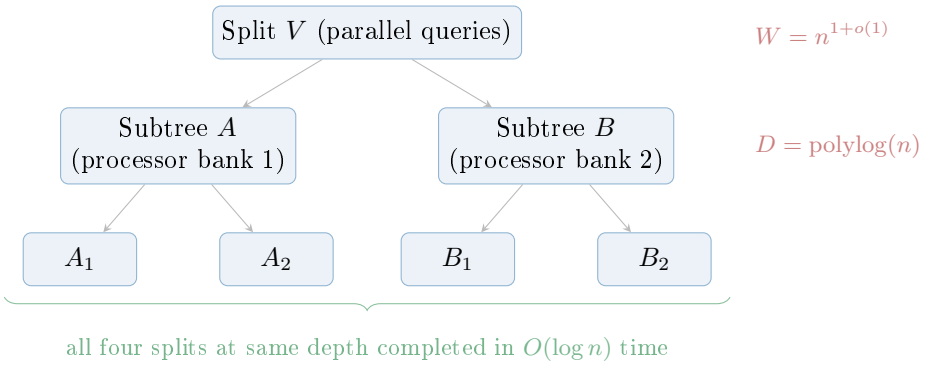


Figure 19.1: Parallel reconstruction: different subtrees are processed by independent processor banks simultaneously. Within each subtree, oracle queries are also parallelized. The result is a polylogarithmic-depth algorithm with near-linear total work.

19.2 Main Result

Theorem 19.1 (Parallel MW approximation)**.** *There exists a PRAM algorithm for the Moseley–Wang objective with*

$$W(n) = n^{1+o(1)}, \qquad D(n) = \text{polylog}(n),$$

achieving a $(1 - o(1))$*-approximation.*

19.3 Exercises

1. Show the top-down split identification is embarrassingly parallelizable and compute the depth per level.

2. Compare the work of the parallel algorithm with the sequential algorithm.

3. Describe how the parallel algorithm handles load balancing for unequal cluster sizes.

Part VII

Geometric and Information-Theoretic Reinterpretations

Chapter 20

Triplets as Ultrametric Curvature Observations

> To measure a tree, you need not see the whole tree. You need only see which paths branch.
>
> —*Flyxion*

This chapter develops the geometric interpretation of the splitting oracle. We showed in Chapter 5 that every hierarchical tree corresponds to an ultrametric, and in Chapter 9 that every triplet relation corresponds to a specific ultrametric inequality. Combining these two observations gives a clean geometric interpretation: each oracle query is a local measurement of the curvature structure of the ultrametric space defined by T^*. The oracle does not reveal the actual distance values—only the ordering of distances for each triple—and this minimal information is precisely what is needed to determine the tree topology, as Theorem 10.4 established. The analogy with curvature is not merely metaphorical: in negatively curved spaces, geodesics diverge in a tree-like pattern, and the branching structure of that pattern is what the oracle measures.

20.1 The Oracle as a Geometric Instrument

Theorem 20.1 (Oracle as curvature measurement)**.** *The splitting oracle, queried on (u, v, w), reveals which of the three strict ultrametric*

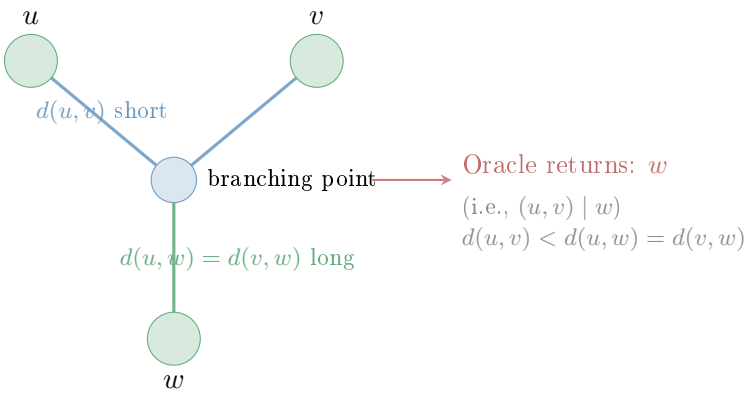


Figure 20.1: The oracle query on (u, v, w) reveals the branching geometry of the ultrametric space: it identifies which of the three pairs shares the lower branching point (shorter ultrametric distance). The Y-shaped tree makes this geometric: the oracle identifies the pair $\{u, v\}$ whose geodesics share the most path before diverging.

inequality patterns holds:

$$\begin{aligned}(u,v) \mid w &\iff d_{T^*}(u,v) < d_{T^*}(u,w) = d_{T^*}(v,w),\\ (u,w) \mid v &\iff d_{T^*}(u,w) < d_{T^*}(u,v) = d_{T^*}(v,w),\\ (v,w) \mid u &\iff d_{T^*}(v,w) < d_{T^*}(u,v) = d_{T^*}(u,w).\end{aligned}$$

Proof. Immediate from Proposition 5.6 and the isosceles property (Proposition 5.3). □

20.2 Sparse Curvature Observations

Corollary 20.2. *The splitting oracle provides exactly the information needed to reconstruct the ultrametric tree topology, and no more: it reveals the branching structure but not the merge heights.*

Knowing all triplet relations for n leaves is equivalent to knowing the tree topology but not the height function.

20.3 Exercises

1. Describe the oracle's output in terms of the ultrametric distance matrix for a specific tree on five leaves.

2. Show that knowing distance orderings (not values) suffices for topology reconstruction.

3. Discuss in what sense the oracle measures "discrete curvature" and how this relates to curvature in Riemannian geometry.

Chapter 21

Entropy Descent on Tree Space

> Certainty is what remains when all alternatives have been eliminated.
>
> —*Flyxion*

This chapter returns to the information-theoretic perspective introduced in Chapter 4 with the full algebraic formalism now in place. The residual hypothesis set $\mathcal{T}_n^{\mathcal{R}_t}$ shrinks monotonically as triplet observations accumulate, and its logarithm $H_t = \log|\mathcal{T}_n^{\mathcal{R}_t}|$ is a non-increasing entropy sequence that reaches zero exactly when reconstruction is complete. We compute the initial entropy $H_0 \sim n \log n$, derive the per-step reduction under the ideal strategy, and show that the partial HC tree construction drives entropy from $\Theta(n \log n)$ down to $O(n \log \log n)$—a nearly complete resolution. The residual entropy is concentrated in the super-vertices, each contributing $O(|S_{\mathbf{v}}| \log |S_{\mathbf{v}}|)$ bits, and their small sizes make the total residual negligible.

21.1 Residual Tree Set and Combinatorial Entropy

Definition 21.1 (Residual tree set)**.** Let $\mathcal{R}_t = \{r_1, \ldots, r_t\}$ be the observed triplet relations. The *residual tree set* is

$$\mathcal{T}_n^{\mathcal{R}_t} = \{T \in \mathcal{T}_n : T \models r_i \text{ for all } i \leq t\}, \qquad H_t = \log|\mathcal{T}_n^{\mathcal{R}_t}|.$$

By Proposition 4.5, H_t is non-increasing.

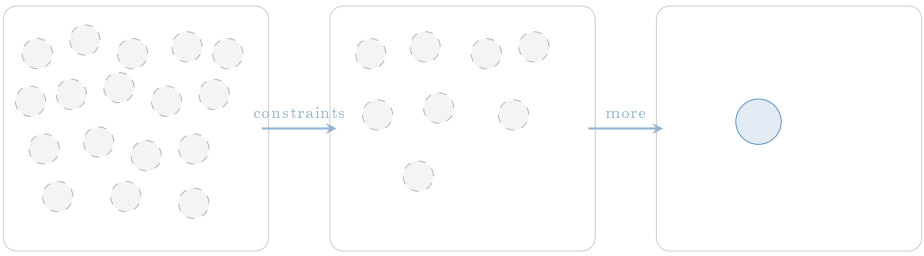


Figure 21.1: The hypothesis space at three stages of the oracle algorithm. Initially all tree topologies are viable. As triplet constraints accumulate, the feasible set shrinks monotonically. When a single topology remains, reconstruction is complete and $H_t = 0$.

21.2 Entropy in the Partial HC Tree

Proposition 21.2. *At the conclusion of the partial HC tree construction, the total residual entropy is*

$$H_t = \sum_{\text{super-vertices } \mathbf{v}} O(|S_{\mathbf{v}}| \log |S_{\mathbf{v}}|) = O(n \log \log n),$$

since each $|S_{\mathbf{v}}| = O(\log n)$ *and* $\sum_{\mathbf{v}} |S_{\mathbf{v}}| = n$.

This shows the algorithm drives entropy from $\Theta(n \log n)$ to $O(n \log \log n)$: a nearly complete resolution.

21.3 Exercises

1. Compute the entropy remaining in a partial HC tree where all super-vertices have size exactly $c \log n$.

2. Show k independent maximally informative triplet queries drive entropy to $H_0 - k \log(3/2)$.

3. Discuss why $O(n \log n / \log(3/2))$ queries are both necessary and sufficient for reconstruction.

Chapter 22

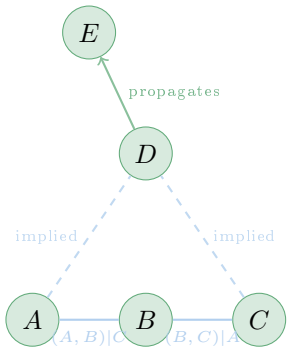
Constraint Propagation and Sparse Inference

> Local constraints speak globally through the language of consistency.
>
> —*Flyxion*

This chapter formalizes the mechanism by which local triplet observations determine global tree structure. A triplet observation concerns only three leaves, yet it constrains the relative placement of every other leaf—because any consistent tree must extend the local branching decision globally. We formalize this as constraint propagation in a network: the nodes are the $\binom{n}{3}$ possible triplets, and the edges represent logical implications (if this triplet relation holds, those relations must hold). Sufficiently many observations create a propagation cascade that determines the entire topology. We connect this mechanism to sparse inference in graphical models and to the broader principle of RSVP-style constraint propagation: local compatibility conditions, accumulated densely enough, drive an entropy descent toward a unique global structure [12].

22.1 Sparse Inference as Constraint Satisfaction

Definition 22.1 (Maximum consistency inference)**.** Given observed triplet set $\mathcal{R}$ and noisy oracle responses, the *maximum consistency*



Two observed triplet relations (solid)
propagate to constrain the whole tree (dashed)

Figure 22.1: Local triplet observations (solid edges) propagate through the consistency network to constrain distant branching events (dashed edges). The directed arrow to E indicates that the observed relations, combined with consistency requirements, restrict where E can be placed in any consistent tree topology.

tree is

$$\widehat{T} = \arg\max_{T \in \mathcal{T}_n} \sum_{r \in \mathcal{R}} \mathbf{1}[T \models r] = \arg\max_{T} \log P(T) + \sum_{r \in \mathcal{R}} \log P(r \mid T).$$

22.2 Constraint Propagation Bound

Theorem 22.2. *Let $\mathcal{R}$ be a consistent set of k triplet observations. Under the uniform prior and maximally informative query strategy,*

$$|\mathcal{T}_n^{\mathcal{R}}| \leq |\mathcal{T}_n| \cdot (2/3)^k.$$

22.3 Connection to Field Theories

The propagation mechanism described here—local observations determining global structure through consistency—is structurally analogous to constraint propagation in field theories such as the RSVP framework, where local gradient constraints propagate through a scalar field to determine global thermodynamic structure. Both are

systems in which local compatibility constraints, accumulated in sufficient density, drive entropy descent toward a unique macroscopic configuration.

22.4 Exercises

1. Formalize: how many other triplet relations does a single triplet observation imply, as a function of n?

2. Show the maximum consistency inference problem is NP-hard in general but tractable under the noisy oracle model.

3. Describe the analogy between the constraint propagation algorithm and belief propagation in graphical models.

Part VIII

Inference Systems, Replication, and Knowledge Accumulation

Chapter 23

Bayesian and Variational Reformulations

This chapter recasts hierarchical reconstruction as a problem of Bayesian inference over tree topologies and develops the variational approximation that connects oracle-augmented algorithms to probabilistic reasoning. In the Bayesian framing, oracle responses are observations that update a prior distribution over $\mathcal{T}_n$, and the accumulated posterior concentrates on trees consistent with the majority vote of all oracle queries. The partial HC tree emerges as a natural compressed representation of this posterior: the resolved portion carries nearly all the posterior probability, while super-vertices represent regions where the posterior remains diffuse. We then introduce the variational free energy, which provides a tractable lower bound on the log-evidence and connects inference over discrete tree spaces to continuous optimization. The ELBO decomposition—expected log-likelihood minus KL divergence from the prior—reappears in Appendix E.

23.1 Bayesian Inference over Tree Metrics

The hierarchical clustering problem has a natural Bayesian formulation. Let $P_0(T)$ be a prior distribution over tree topologies and let the observed triplet relations $\mathcal{R} = \{r_1, \ldots, r_t\}$ be conditionally independent given T with likelihoods $P(r_i \mid T)$. The posterior is

$$P(T \mid \mathcal{R}) \propto P_0(T) \prod_{r \in \mathcal{R}} P(r \mid T).$$

Under the noisy oracle model, $P(r \mid T) = p$ if $T \models r$ and $P(r \mid T) = (1-p)/2$ otherwise. The posterior concentrates on trees consistent with the majority vote of all oracle responses.

23.2 The Partial HC Tree as a Belief State

The partial HC tree is a compressed representation of the posterior $P(T \mid \mathcal{R})$. The structure above the super-vertex level is essentially certain (probability approaching 1 for large n). The uncertainty is concentrated within the super-vertices, where the local posterior is approximately uniform over the $(2|S_{\mathbf{v}}| - 3)!!$ local topologies.

23.3 Variational Inference

When the exact posterior is intractable to compute, variational inference approximates it with a tractable distribution Q.

Definition 23.1 (Variational free energy)**.** For a variational distribution Q over $\mathcal{T}_n$, the *free energy* is

$$\mathcal{F}(Q) = \mathbb{E}_Q[-\log P(\mathcal{R} \mid T)] + \mathrm{KL}(Q \| P_0(T)).$$

Proposition 23.2 (Free energy as upper bound)**.** $\mathcal{F}(Q) \geq -\log P(\mathcal{R})$ *for all Q, with equality when $Q = P(T \mid \mathcal{R})$.*

Proof. This is a standard result in variational Bayes, following from the non-negativity of KL divergence. □

Minimizing $\mathcal{F}(Q)$ over a tractable family of distributions Q provides a principled approximate inference procedure for tree structure.

23.4 Mean-Field Approximation for Trees

A tractable mean-field approximation assumes that the internal node assignments are independent. Under this assumption, the free energy decomposes into a sum over internal nodes, making optimization tractable. The resulting approximation is related to the partial HC tree construction: the high-confidence resolved nodes correspond to the converged mean-field assignments.

23.5 Exercises

1. Write out the log-posterior $\log P(T \mid \mathcal{R})$ as a function of the number of consistent and inconsistent triplets in $\mathcal{R}$ under the noisy oracle model.

2. Show that minimizing free energy is equivalent to maximizing the evidence lower bound (ELBO).

3. Describe a mean-field approximation for trees on n leaves and identify which parameters the variational distribution must specify.

Chapter 24

The Replication Crisis as Constraint Distortion

> A scientific literature that systematically suppresses negative results is a constraint network with missing edges. The structure it implies is not the structure of nature.
>
> —*Flyxion*

This chapter applies the constraint accumulation framework to one of the most consequential problems in contemporary science: the replication crisis [7, 9]. We argue that the crisis is not merely a statistical pathology but a structural consequence of selection mechanisms that corrupt the constraint network from which scientific knowledge is inferred. The formalism is identical to that developed in Chapter 6: each experiment contributes a constraint operator on the hypothesis space, and accurate inference requires the full set of operators—including those from failed, null, or unexpected results. When selection mechanisms systematically suppress disconfirming operators, the accumulated constraint set becomes incomplete, and the hypothesis space it implies is inflated and biased. The analogy to the BUILD algorithm receiving only a biased subset of triplets—and converging to the wrong tree—is precise and predictive.

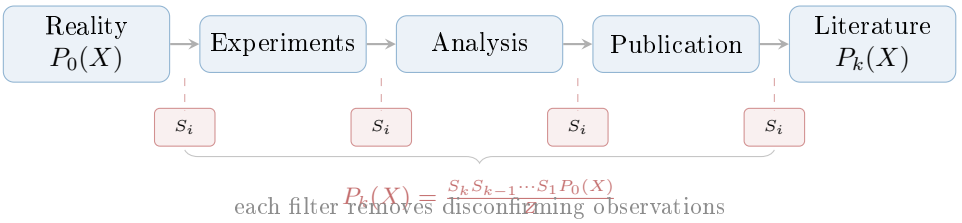


Figure 24.1: The scientific selection pipeline. Each stage—experiment design, analysis, peer review, editorial decisions—applies a selection function S_i that may systematically favor statistically significant or confirmatory outcomes. The observed literature $P_k(X)$ is a distorted version of the true outcome distribution $P_0(X)$.

24.1 Selection as a Distribution Transformation

Definition 24.1 (Selection mechanism)**.** A *selection mechanism* is $S : \mathcal{X} \to [0, 1]$ assigning each outcome x the probability of being reported. The *selected distribution* is

$$P_{\text{obs}}(X) = \frac{S(X)P(X)}{\int S(x)P(x)\,\mathrm{d}x}.$$

Proposition 24.2 (Selection bias is structural)**.** *Even when each individual experiment is unbiased, a selection mechanism correlated with the outcome produces a biased observed distribution.*

24.2 Statistical Power and the False Discovery Rate

Consider a research community testing many hypotheses, of which fraction π_1 are true. With significance threshold α and Type II error rate β, the false discovery rate among published results is

$$\text{FDR} = \frac{\alpha\pi_0}{\alpha\pi_0 + (1-\beta)\pi_1}. \tag{24.1}$$

Proposition 24.3 (High FDR under low power [7])**.** *When $\pi_1 \ll 1$ and $1 - \beta \approx \alpha$,* FDR $\approx \pi_0 \approx 1$*: almost all published findings are false positives.*

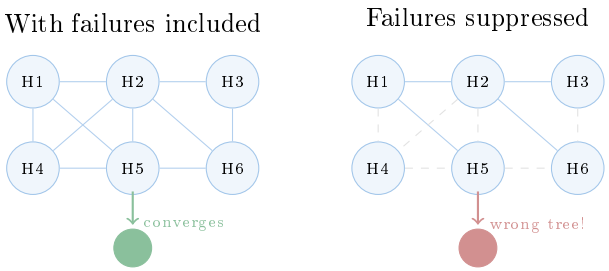


Figure 24.2: Left: when all experimental results are recorded—including failures (gray dashed edges)—the constraint network is dense and the hypothesis space collapses to the correct theory. Right: when failures are suppressed, the sparse constraint network is underdetermined, and inference converges to a hypothesis consistent with only the positive evidence—which may be far from the truth.

24.3 Multiscale Selection

Scientific results pass through k selection stages. The distribution visible after stage k is

$$P_k(X) = \frac{S_k \circ S_{k-1} \circ \cdots \circ S_1(P_0)(X)}{Z}. \tag{24.2}$$

Small biases at each stage compound multiplicatively.

24.4 Missing Constraints and the Distorted Posterior

Let R be reported results and F the unreported failures. The community's posterior is $P(H \mid R) \propto P(H)P(R \mid H)$, whereas the correct posterior is $P(H \mid R, F) \propto P(H)P(R, F \mid H)$.

Theorem 24.4 (Distortion from missing constraints)**.** $\mathcal{H}^{\mathcal{C}_R} \supseteq \mathcal{H}^{\mathcal{C}_R \cup \mathcal{C}_F}$. *Inference using only $\mathcal{C}_R$ yields an inflated feasible set and overconfident posterior.*

24.5 Exercises

1. Compute FDR (equation 24.1) for $\pi_1 = 0.1$, $\alpha = 0.05$, power $1 - \beta \in \{0.8, 0.3\}$.

2. Give an example of a biased triplet set $\mathcal{R}' \subsetneq \mathcal{R}$ that causes BUILD to reconstruct a wrong tree.

3. Prove that funnel plot asymmetry in meta-analysis is a direct consequence of the selection mechanism defined here.

Chapter 25

Observations as Operators on Hypothesis Space

This chapter develops the formal proposal that scientific communication can be made more epistemically robust by representing experimental outcomes as mathematical operators on hypothesis space, rather than as narrative claims of success or failure. The evidence content of an observation—its action on $\mathcal{H}$—is determined entirely by its logical relationship to hypotheses and is independent of the social context in which it was produced. If experimental outcomes were submitted as anonymous constraint operators, the reputational incentive to suppress negative results would be severed: there is no such thing as a failed operator, only one that eliminates certain hypotheses. We formalize this proposal, show its compatibility with Bayesian inference, and draw the analogy to the triplet oracle: just as oracle responses are aggregated without regard to individual correctness, an epistemic infrastructure based on operators would aggregate evidence without regard to its social valence.

25.1 From Narratives to Operators

The previous chapter identified the root cause of the replication crisis as the social filtering of constraint operators: negative results are suppressed because they carry reputational costs. This chapter develops the proposal that scientific communication can be reformed by abstracting experimental outcomes from their social context.

The central idea is to represent each experiment not as a narrative of success or failure but as a mathematical operator on the hypothesis

space.

Definition 25.1 (Evidence operator)**.** For a hypothesis space $\mathcal{H}$ and an observation o, the *evidence operator* is

$$C_o : \mathcal{H} \to \mathcal{H}, \qquad C_o(H) = \begin{cases} H & \text{if } H \text{ is consistent with } o, \\ \varnothing & \text{otherwise.} \end{cases}$$

In this formulation, there is no concept of a "failed" experiment. An experiment that rules out a hypothesis removes it from the feasible set. An experiment that confirms a hypothesis leaves only those consistent with the observation. Both operations reduce uncertainty.

25.2 Anonymization as Structural Preservation

The reputational dimension of scientific communication arises because experimental outcomes are attributed to individual researchers. Positive results earn credit; negative results risk embarrassment.

If experimental outcomes were instead represented as anonymous constraint contributions – formal operators on $\mathcal{H}$ with no authorship information – the reputational incentive to suppress negative results would be removed.

Proposition 25.2 (Anonymization preserves information)**.** *The evidence operator C_o is fully determined by the logical content of observation o, independent of who produced it. Anonymizing authorship does not affect the operator's action on $\mathcal{H}$.*

This observation has a practical implication: scientific infrastructure that treats observations as formal mathematical objects can preserve the full constraint network regardless of social incentives, as long as researchers submit their operators even when the outcome is unexpected.

25.3 Bayesian Interpretation

The operator model is directly compatible with Bayesian inference. The Bayesian update for a prior P_0 over $\mathcal{H}$ under observation o is

$$P(H \mid o) \propto P(o \mid H) P_0(H).$$

Both positive and negative results update the posterior: a negative result assigns low likelihood $P(o \mid H)$ to the hypotheses it contradicts and high likelihood to the hypotheses it confirms.

Proposition 25.3. *The information contributed by a negative result equals the information contributed by any other result with the same likelihood ratio. The distinction between positive and negative is epistemically irrelevant; only the likelihood function matters.*

25.4 Implications for Scientific Infrastructure Design

The mathematical framework points to specific features that a replication-resistant scientific infrastructure should possess.

The system should accept submissions that consist of the evidence operator alone, without requiring a narrative claim about what the operator "proves."

The system should record negative results with the same fidelity as positive ones, so that the accumulated constraint network is complete.

The system should provide meta-analytic tools that aggregate operators across studies, computing the posterior distribution $P(H \mid \mathcal{R})$ for any specified prior.

These requirements are not merely social; they are mathematical necessities for robust inference from a distributed constraint network.

25.5 Exercises

1. Describe the evidence operator C_o for a null hypothesis significance test that fails to reject the null at $\alpha = 0.05$. Which hypotheses does it eliminate?

2. Show that the cumulative evidence operator $C_{\mathcal{R}} = C_{o_k} \circ \cdots \circ C_{o_1}$ is commutative: the order of observations does not affect the accumulated constraint space.

3. Propose a formal representation format for evidence operators that would allow automatic aggregation across independent studies.

Chapter 26

The Epistemic Value of Exploratory Reasoning

This chapter defends, on mathematical grounds, the epistemic value of publishing exploratory, incomplete, and even incorrect scientific work. The argument is informational rather than social: the search trajectory that led to an incorrect result contains information about the hypothesis space, because it documents which branches were explored and found barren. An incorrect derivation that assumed conditions $\mathcal{A}$ and reached a contradiction proves $\neg\mathcal{A}$—a constraint operator on $\mathcal{H}$ just as valid as any positive result. We illustrate the principle with the essay "Three Reasons Why 17 Is the Best Possible Number," which functions as a probe proposing candidate operators on a conceptual hypothesis space: not claiming to determine the structure of that space but highlighting regions worth examining. We formalize the notion of a constraint refinement—a later, tighter operator that supersedes an earlier approximate one—and show that publishing early approximate operators is epistemically valuable as a scaffold for refinement.

26.1 Incomplete Work as Search Trajectory Preservation

The previous chapters established that complete inference requires complete constraints. This chapter argues for a complementary principle: the search *trajectory* that generated the constraints is itself an informational object whose preservation is epistemically valuable.

Scientific progress rarely proceeds by direct logical deduction from axioms to theorems. It proceeds through exploration: conjecture, partial formalization, failed approaches, and gradual refinement. If only the final correct results are recorded, later researchers must reconstruct the search process independently, repeatedly rediscovering the same dead ends.

Definition 26.1 (Search trajectory)**.** Let $\mathcal{H}$ be a hypothesis space and $o_1, o_2, \ldots, o_t$ a sequence of observations. The *search trajectory* is the sequence of accumulated constraint spaces $\mathcal{H}_0 \supseteq \mathcal{H}_1 \supseteq \cdots \supseteq \mathcal{H}_t$ together with the observations that produced each reduction.

Proposition 26.2 (Trajectory preserves more than endpoint)**.** *The trajectory $(\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_t)$ contains strictly more information than the endpoint $\mathcal{H}_t$ alone, because it encodes which observations produced which reductions.*

26.2 Flawed Work as Constraint

An incorrect derivation or failed simulation still contributes information. If a derivation assumes conditions $\mathcal{A}$ and reaches a contradiction, the derivation proves $\neg\mathcal{A}$. If a simulation breaks down at a boundary condition, the simulation reveals that the model is inconsistent with behavior at that boundary.

Proposition 26.3 (Incorrect results are constraints)**.** *Let D be a derivation that assumes $\mathcal{A}$ and produces a contradiction. Then D contributes the constraint $C_D : \mathcal{H} \to \mathcal{H}$ that eliminates all hypotheses consistent with $\mathcal{A}$.*

A derivation that appears to "fail" is therefore not an absence of information. It is a constraint operator that eliminates a region of $\mathcal{H}$.

26.3 An Illustrative Example: Heuristic Argumentation

The essay "Three Reasons Why 17 Is the Best Possible Number" illustrates the practice of exploratory reasoning through a deliberately

playful case. The essay draws connections between the ten decimal digits, the seven days of the week, geometric interpretations involving cubes and orthodromes, and a classification of mathematical disciplines. None of these connections is proposed as a formal theorem.

Within the operator framework, such an essay functions as a probe of the hypothesis space: it proposes candidate correspondences

$$C_1, C_2, C_3 : \mathcal{H} \to \mathcal{H}$$

that restrict attention to regions of $\mathcal{H}$ where certain numerical and structural patterns coexist. The operators do not claim to determine $\mathcal{H}$'s correct structure; they highlight regions worth examining further.

Remark 26.4. The epistemic contribution of a playful or conjectural argument is not its truth value but its role in directing subsequent inquiry. An incorrect operator that eliminates a productive subspace is harmful; an incorrect operator that eliminates an unproductive one is beneficial. The utility of exploratory reasoning must therefore be evaluated prospectively, in terms of the search it enables.

26.4 Refactoring as Constraint Revision

Exploratory work supports what might be called *epistemic refactoring*: the revision of earlier approximate constraints as better ones become available. Early-stage research produces loose constraints that are later replaced by tighter ones derived from more careful analysis.

Definition 26.5 (Constraint refinement)**.** A constraint C' is a *refinement* of C if $\mathcal{H}^{C'} \subseteq \mathcal{H}^{C}$. The constraint network benefits from refinements: they tighten the feasible set without erasing previous progress.

Publishing exploratory work creates a scaffold of approximate constraints that later researchers can refine without starting from scratch.

26.5 Connection to the HC Reconstruction Framework

The partial HC tree is an exact formal analogue of this principle. The algorithm publishes a partial structure – a tree with unresolved super-

vertices – rather than waiting until complete resolution is achieved. This partial result is genuinely useful: it captures the structure that has been reliably inferred and clearly marks the regions of remaining uncertainty.

A scientist who publishes a partial derivation, clearly marking what is established and what remains conjectural, provides the same service. The partial result contributes reliable constraints to the collective hypothesis space while honestly representing the state of remaining uncertainty.

26.6 Exercises

1. Formalize the value of a published exploratory argument as a function of its expected reduction in collective search time, and describe what properties of the argument maximize this value.

2. Describe a scenario in which publishing an incorrect result is more epistemically valuable than publishing no result, using the constraint operator framework.

3. Explain how the partial HC tree construction models the practice of publishing interim scientific results: what is the analogue of a super-vertex in scientific inquiry?

4. Discuss the ethics of publishing exploratory or speculative arguments. When does the epistemic value outweigh the risk of misleading readers?

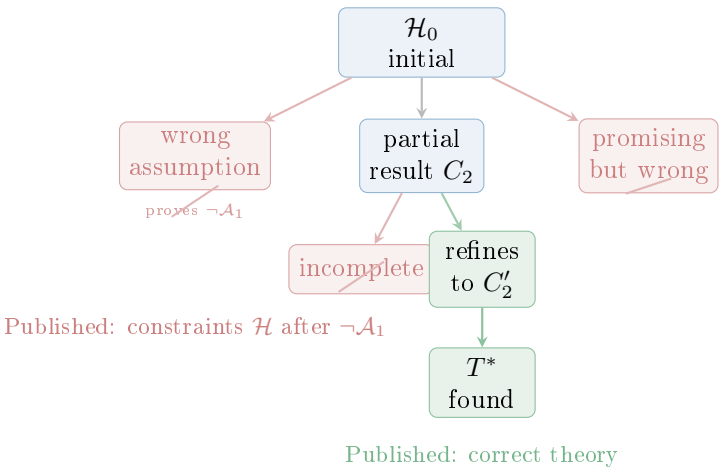


Figure 26.1: The scientific search tree. Successful results (green) and failures (red) both contribute constraint operators that narrow the hypothesis space. When dead ends are published, future researchers avoid re-exploring them. When only successes are published, the search tree must be re-traversed independently by each generation.

Part IX

Full Derivations and Advanced Appendices

Appendix A

Proof That Trees and Ultrametrics Are Equivalent

A.1 Statement of the Theorem

Theorem A.1 (Tree–ultrametric equivalence)**.** *There is a bijection between:*

1. *Rooted trees (T, h) on a leaf set X with a strictly decreasing node height function h taking distinct values at distinct internal nodes, and*

2. *Ultrametric spaces (X, d) with distinct pairwise distances.*

The bijection sends T to $d_T(x, y) = h(\mathrm{LCA}_T(x, y))$ and sends d to the tree constructed by the single-linkage procedure.

A.2 Proof: Trees Induce Ultrametrics

This was established in Chapter 5 (Theorem 5.5). Distinctness of distances follows from distinct height values.

A.3 Proof: Ultrametrics Induce Trees

Proof. Let (X, d) be a finite ultrametric with distinct distance values. We construct a rooted tree T as follows.

Step 1: Sort the distinct distance values as $\theta_1 < \theta_2 < \cdots < \theta_m$.

Step 2: For each threshold θ_k, define the θ_k-partition of X by: $x \sim_k y$ if $d(x, y) \leq \theta_k$. By the ultrametric property, each $\sim_k$ is an equivalence relation. The classes are the clusters at scale θ_k.

Step 3: The cluster hierarchy is the set of all clusters at all scales, ordered by inclusion. This forms a laminar family and thus defines a rooted tree.

Step 4: Assign height $h(v) = \theta_k$ to the internal node corresponding to a cluster first appearing at scale θ_k. The leaves have height 0.

Verification: By construction, $d(x, y) = h(\mathrm{LCA}_T(x, y))$: the LCA of x and y is the internal node at height $d(x, y)$, which is the smallest θ_k at which x and y join the same cluster.

Uniqueness: If two distinct trees T_1 and T_2 induced the same ultrametric, they would have the same clusters at every scale, hence the same laminar family, hence $T_1 = T_2$. □

A.4 The Case of Non-Distinct Distances

When distance values are not all distinct, the tree is not unique: multiple internal nodes may have the same height. The correspondence becomes one-to-many. The textbook focuses on the generic case of distinct distances for clarity.

Appendix B

Proof That Triplets Determine Topology

B.1 Consistency and Compatibility

Definition B.1. A set of rooted triplet relations $\mathcal{R}$ is *compatible* if there exists a rooted binary tree T such that $T \models r$ for all $r \in \mathcal{R}$.

B.2 The Reconstruction Theorem

Theorem B.2 (Triplet determination theorem)**.** *Let $\mathcal{R}$ be a consistent, complete set of triplet relations on leaf set V (i.e., for every triple $\{u, v, w\} \subseteq V$, exactly one of the three relations is in $\mathcal{R}$). Then $\mathcal{R}$ uniquely determines the rooted binary tree topology T with $L(T) = V$.*

Proof. We proceed by induction on $|V| = n$.

Base case: $n = 3$. There is exactly one triplet relation in $\mathcal{R}$, say $(u, v)|w$. There are three rooted binary tree topologies on three leaves, each corresponding to one of the three relations. The given relation selects a unique topology.

Inductive step: Assume the result holds for all leaf sets of size $< n$. For leaf set V with $|V| = n$, consider any internal node v of the true tree T with children v_L and v_R. Let $A = L(T[v_L])$ and $B = L(T[v_R])$.

Identifying the top-level split: The bipartition $\{A, B\}$ can be identified from $\mathcal{R}$ as follows. For any $a \in A$ and $b \in B$, every leaf $x \in V$ either:

- satisfies $(a, b)|x$ (meaning x splits before a and b merge, i.e., $x \notin \{\text{subtree rooted at } \mathrm{LCA}(a, b)\}$), which is impossible since $\mathrm{LCA}(a, b)$ is the root, or

- satisfies $(a, x)|b$ or $(b, x)|a$ depending on whether $x \in A$ or $x \in B$.

Therefore: $x \in A$ if and only if $(a, x)|b$ for any fixed $b \in B$ and $a \in A$. This recovers A (and hence B) from the triplets.

Recursive reconstruction: Having identified A and B, restrict $\mathcal{R}$ to triplets within A and within B. By induction, each restricted triplet set uniquely determines the subtree on A and the subtree on B.

Uniqueness: The split $\{A, B\}$ is unique (being the top-level split of the true tree), and the subtrees on A and B are uniquely determined by induction. Therefore T is unique. □

Corollary B.3 (Sufficiency of dense triplets)**.** *If $\mathcal{R}$ contains at least one triplet relation for every triple in V, then $\mathcal{R}$ uniquely determines the tree.*

B.3 Closure of Triplet Systems

Definition B.4 (Triplet closure)**.** The *closure* $\overline{\mathcal{R}}$ of a triplet set $\mathcal{R}$ is the maximal consistent triplet set containing $\mathcal{R}$. It is the set of all triplet relations implied by $\mathcal{R}$ under the tree axioms.

Computing the closure can be done via the BUILD algorithm applied iteratively. If $\mathcal{R}$ is inconsistent, its closure is undefined.

Appendix C

Noise Analysis for Oracle Aggregation

C.1 Setup

Let the oracle answer each query correctly with probability $p > 1/2$, independently across queries. We want to use majority vote of m queries to achieve failure probability δ.

C.2 Single Triplet Analysis

Theorem C.1 (Majority vote error bound)**.** *Let* $X_1, \ldots, X_m \sim_{\text{iid}} \text{Ber}(p)$ *with* $p > 1/2$*. Then*

$$\mathbb{P}\left(\sum_{i=1}^{m} X_i \leq m/2\right) \leq \exp\bigl(-2m(p-1/2)^2\bigr).$$

Proof. This is Hoeffding's inequality (Theorem 3.2) applied directly. □

Corollary C.2 (Required sample size)**.** *To achieve* $\mathbb{P}(\textit{error}) \leq \delta$*, it suffices to use*

$$m \geq \frac{\log(1/\delta)}{2(p-1/2)^2}.$$

queries per triplet.

C.3 Uniform Bound over All Triplets

Let $N = \binom{n}{3}$ be the number of distinct triplets queried.

Theorem C.3 (Uniform correctness)**.** *Setting $\delta' = \delta/N$ in the single-triplet bound, and using*

$$m = O\left(\frac{\log(N/\delta)}{(p-1/2)^2}\right) = O\left(\frac{\log n + \log(1/\delta)}{(p-1/2)^2}\right)$$

queries per triplet, all N triplets are answered correctly simultaneously with probability $\geq 1 - \delta$.

Proof. Apply the union bound over N events, each with probability δ'. □

C.4 Adaptive Query Strategies

When the algorithm queries triplets adaptively (choosing the next query based on previous answers), the analysis remains valid because each individual query error event is independent. The union bound applies to any fixed set of at most N queries regardless of the selection order.

Appendix D

Loss Bounds from Collapsing Unresolved Subtrees

D.1 The Contribution of Super-Vertex Pairs

Let $\widehat{T}$ be the partial HC tree and T^* the optimal tree. For pairs $\{i, j\}$ whose LCA in both trees is above all super-vertices, the Dasgupta cost contribution is matched exactly. The error comes from pairs whose LCA lies inside a super-vertex.

Theorem D.1 (Collapse loss bound)**.** *Let* $\mathbf{v}$ *be a super-vertex with cluster* $S_{\mathbf{v}}$*. The loss in the Dasgupta objective incurred by collapsing* $S_{\mathbf{v}}$ *satisfies*

$$\sum_{\{i,j\} \subseteq S_{\mathbf{v}}} w_{ij} \cdot |\textit{merge-level distortion}| \leq |S_{\mathbf{v}}|^2 \cdot \max_{i,j \in S_{\mathbf{v}}} w_{ij}.$$

Proof. The merge-level distortion for any pair $\{i, j\}$ is at most n (the maximum possible subtree size). However, since $|S_{\mathbf{v}}| = O(\log n)$, the actual distortion is at most $O(\log n)$ for each pair, and there are $O(\log^2 n)$ pairs. The total distortion per super-vertex is $O(\log^3 n \cdot \max w_{ij})$.

Summing over all $n/O(\log n) = O(n/\log n)$ super-vertices, the total loss is $O(n \log^2 n \cdot \max w_{ij})$, which is $o(\text{cost}_D(T^*))$ for similarity matrices where the objective is $\Omega(n^2)$. □

D.2 Implications for Approximation Ratio

Corollary D.2. *For similarity matrices where* $\text{cost}_D(T^*) = \Omega(n^2 \cdot \text{avg}(w))$*, the relative loss from collapsing super-vertices is* $o(1)$*, giving a* $(1 + o(1))$*-approximation for the contribution of resolved pairs.*

The overall approximation ratio combines this with the quality of the split identification at each level.

Appendix E

Bayesian and Variational Reformulations

E.1 Full Posterior over Tree Metrics

The complete Bayesian model for hierarchical reconstruction is:

$$P(T \mid \mathcal{R}) \propto P(T) \prod_{r \in \mathcal{R}} P(r \mid T),$$

where $P(T)$ is the prior over tree topologies and $P(r \mid T) = p \cdot \mathbf{1}[T \models r] + (1-p)/2 \cdot \mathbf{1}[T \not\models r]$.

E.2 Free Energy and ELBO

Theorem E.1. *For any distribution Q over $\mathcal{T}_n$,*

$$\log P(\mathcal{R}) \geq \mathbb{E}_Q[\log P(\mathcal{R} \mid T)] - \mathrm{KL}(Q \| P(T)).$$

Equality holds when $Q = P(T \mid \mathcal{R})$.

Proof.

$$\begin{aligned}
\log P(\mathcal{R}) &= \log \mathbb{E}_{P(T)}[P(\mathcal{R} \mid T)] \\
&= \log \mathbb{E}_Q\left[\frac{P(\mathcal{R} \mid T)P(T)}{Q(T)}\right] \\
&\geq \mathbb{E}_Q\left[\log \frac{P(\mathcal{R} \mid T)P(T)}{Q(T)}\right] \quad \text{(Jensen's inequality)} \\
&= \mathbb{E}_Q[\log P(\mathcal{R} \mid T)] + \mathbb{E}_Q\left[\log \frac{P(T)}{Q(T)}\right] \\
&= \mathbb{E}_Q[\log P(\mathcal{R} \mid T)] - \mathrm{KL}(Q \| P(T)).
\end{aligned}$$

□

E.3 Connection to the Partial HC Tree

The partial HC tree corresponds approximately to the MAP estimate of $P(T \mid \mathcal{R})$: the tree of highest posterior probability given the oracle observations. The super-vertices represent regions where the posterior is locally diffuse – where the evidence does not strongly favor any particular sub-topology – so the MAP estimate collapses them.

E.4 Stochastic Variational Inference for Trees

A full variational inference scheme would parameterize Q as a distribution over tree topologies and optimize $\mathcal{F}(Q)$ by gradient methods. For trees this is technically challenging due to the discrete, combinatorial nature of the space. Current approaches include:

- *Mean-field approximations*: factorize Q over independent edge or node assignments.
- *Sampling-based methods*: approximate the ELBO gradient by Monte Carlo sampling from Q.
- *Structured variational families*: exploit the laminar structure of dendrograms to define tractable distributions.

The partial HC tree algorithm can be understood as a deterministic approximation to the variational inference problem, in which the oracle evidence is used to greedily maximize the ELBO at each level of the recursion.

Synthesis: Structure from Constraint

> What we have called inference is really a kind of erosion. Each observation removes a possibility. What remains is the shape of the world.
>
> —*Flyxion*

This textbook has traced a single thread from the combinatorics of rooted trees to the epistemology of scientific knowledge. The thread is the principle of constraint-driven structure emergence: global organization arises when local observations form a coherent constraint network that systematically reduces the entropy of a hypothesis space.

The mathematical argument was developed in three registers. The first was geometric: every hierarchy corresponds to an ultrametric, and every triplet observation reveals which ultrametric inequality holds for a given triple. The oracle, in this light, is an instrument for measuring the local curvature of an ultrametric space from which the global geometry can be inferred.

The second register was algorithmic: the partial HC tree construction demonstrates that a noisy oracle providing $O(n^2 \operatorname{polylog} n)$ bits of information is sufficient to reconstruct a hierarchy well enough to achieve near-optimal objective values for both the Dasgupta and Moseley–Wang formulations. This result breaks classical hardness barriers by transforming structural information into computational leverage.

The third register was epistemological: the same constraint-accumulation mechanism that drives algorithmic reconstruction also describes how scientific communities build knowledge from distributed experimental evidence. The replication crisis is not merely a statistical pathology;

it is what happens when the constraint network is systematically corrupted by selection mechanisms that suppress disconfirming observations. Restoring the full constraint network – by treating experimental outcomes as operators on hypothesis space rather than narratives about personal success – is a mathematical necessity for reliable inference, not merely a normative aspiration.

Several implications follow from the unified perspective.

Minimal information suffices when structure is present. The tree reconstruction results show that $O(n \log n)$ bits of triplet information are sufficient to determine a tree topology on n leaves, even though the naive description of the topology requires $\Theta(n \log n)$ bits. There is no slack: the information content of the triplet basis exactly matches the complexity of the structure being recovered. This tight correspondence suggests that hierarchical structures are, in a precise sense, maximally compressible into their minimal relational witnesses.

Noise tolerance reflects structural regularity. The oracle is reliable only with probability $p > 1/2$. Yet this weak reliability is sufficient, because the tree's structure is globally consistent: every correct triplet relation is coherent with every other. An adversarial configuration of triplet responses would be self-contradictory, and the contradiction would be detectable. It is the global consistency of the true hierarchy that allows noisy local observations to be aggregated into reliable global inferences.

Partial knowledge is not ignorance. The partial HC tree makes explicit what is known and what is not. The resolved portion of the tree is correct with high probability; the super-vertices honestly represent the frontier of reliable inference. This is the model that scientific communication should aspire to: not premature certainty, and not false modesty about what has been reliably established, but an accurate representation of the structure inferred from available evidence with the unresolved regions clearly marked.

The constraint network is the epistemic object. In the end, what the algorithm maintains, what scientific literature accumulates, and what Bayesian inference updates, is not a single best hypothesis but a constraint network from which hypotheses can be evaluated. The quality of knowledge is determined by the completeness

and consistency of this network. Strategies that increase completeness – registering experiments before they run, publishing negative results, representing outcomes as formal operators – are not bureaucratic formalities. They are maintenance operations on the inferential substrate from which reliable knowledge is extracted.

The learning-augmented hierarchical clustering framework is, in this light, a small but precise model of a large and general principle. The tree reconstruction algorithm is a controlled laboratory in which the abstract principle of constraint-driven entropy reduction can be studied exactly. The objective functions have precise definitions. The oracle model is explicit. The approximation ratios are proven. The connection to replication and scientific inference, while more qualitative, is mathematically grounded in the same formalism.

This combination – a rigorous algorithmic theory with a broad interpretive reach – is the contribution the textbook has aimed to make. The reader who has followed the argument to this point now has both the technical apparatus to study hierarchical structure precisely and a framework for thinking about how structure emerges from constraint in systems far beyond the computational domain.

The field has further to go. Questions remain about optimal query strategies, the geometry of hypothesis spaces beyond trees, the design of constraint-preserving scientific infrastructure, and the connections between ultrametric geometry and the large-scale structure of physical space. The partial HC tree is not the end of the story; it is a super-vertex in a larger hierarchy that future work will resolve.

Bibliography

[1] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.

[2] Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. pages 841–854, 2017.

[3] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM*, 66(4), 2019.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[5] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. pages 118–127, 2016.

[6] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[7] John P. A. Ioannidis. Why most published research findings are false. *PLOS Medicine*, 2(8):e124, 2005.

[8] Benjamin Moseley and Joshua Wang. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. 30, 2017.

[9] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251), 2015.

[10] Aurko Roy and Sebastian Pokutta. Hierarchical clustering via spreading metrics. 30, 2017.

[11] Naruya Saitou and Masatoshi Nei. *The neighbor-joining method: A new method for reconstructing phylogenetic trees*, volume 4. 1987.

[12] Charles Semple and Mike Steel. *Phylogenetics*. Oxford University Press, 2003.